



SCRIPTS E ISTRUZIONI EDEN EDITOR ARMA 3

INDICE

1	INTRODUZIONE.....	7
2	MODIFICARE FILE MISSIONE SCARICATI DAL WEB.....	7
3	SALVATAGGIO MISSIONE	7
4	DISABILITARE IL SALVATAGGIO IN MULTIPLAYER.....	8
5	DISABILITARE LA CHAT	8
6	VIRTUAL ARSENAL.....	8
7	COME TROVERE LE CLASSNAMES PER GLI ADDON (MOD).....	8
8	SELEZIONARE LOADOUT E FREQUENZE RADIO PREDEFINITI DALL'ARSENAL	8
8.1	SELEZIONARE LOADOUT PREDEFINITI	8
8.2	IMPOSTARE LA RADIO DIRETTAMENTE DALL'ARSENAL.....	11
8.3	SCRIPT PER ARSENAL (ESEMPIO).....	11
8.3.1	<i>Alternativa con file esterno.....</i>	<i>12</i>
8.4	SELEZIONARE DEI FILE SQF RANDOM PER UN DETERMINATO RUOLO.....	13
8.5	ASSEGNARE L'ABILITÀ DI MEDICO O INGEGNERE (SPECIALISTA ESPLOSIVI) AD UN'UNITÀ DA SCRIPT.....	14
9	RIPRODURRE UN SUONO ESTERNO IN UN OGGETTO O IN UN TRIGGER	15
9.1	CODICE DA INSERIRE (FUNZIONA).....	18
9.2	RIPRODURRE DELLA MUSICA ALL'INTERNO DI UN VEICOLO.....	19
10	ANIMAZIONI CIVILI.....	23
10.1	CODICE DA INSERIRE (FUNZIONA).....	24
11	MODULO ZEUS – COME IMPOSTARE ZEUS ED INTEREAGIRE CON OGGETTI GIÀ PRESENTI...24	
11.1	ALTRO MATERIALE.....	25
12	RICHIAMARE IL SODDISFACIMENTO DI UN TRIGGER.....	26
13	COME ATTIVARE UN TRIGGER DA PARTE DI UNA SPECIFICA UNITÀ (THISLIST COMMAND)..27	
13.1	TRIGGER RADIO ALPHA, BRAVO.....	27
14	DEAD BODIES - POSTURE.....	28
15	END MISSION.....	29
15.1	END MISSION SCRIPT.....	30
15.2	ESEMPIO FUNZIONANTE.....	33
16	VISUALIZZARE UN'IMMAGINE SU SCHERMO	34
16.1	IMMAGINE A SCHERMO	34
16.2	IMMAGINE A SCHERMO INTERNO	35
16.3	CREARE UNA SCHERMATA FINE MISSIONE A TUTTO SCHERMO CON SUONO PERSONALIZZATO	38
17	HOW TO ADD AN IMAGE TO A HINT	40



18	AGGIUNGERE UN'INSEGNA PERSONALIZZATA SULL'UNIFORME DELL'UNITÀ	41
19	ACTIVATE A TRIGGER BY LANDING THE HELICOPTER IN A CERTAIN AREA/WITHIN THE TRIGGER ZONE	45
20	CREARE UN OBIETTIVO	46
20.1	UCCIDI IL VIP (O DISTRUGGERE UN OGGETTO)	46
20.2	LIBERA L'OSTAGGIO	48
20.2.1	Primo modo:	48
20.2.2	Secondo modo con soddisfacimento obiettivo.....	50
20.2.3	Terzo modo sempre con soddisfacimento obiettivo (Da verificare).....	53
21	RECUPERO INFORMAZIONI (INTEL).....	55
22	FAR COMPARIRE UN MARKER IN MAPPA CON TRIGGER	58
23	CREARE UNA SCHERMATA DI CARICAMENTO PERSONALIZZATA DELLA MISSIONE	60
23.1	MISSION CONFIGURATION	60
23.2	WORLD CONFIGURATION	61
23.3	FULL SCREEN MISSION LOADING SCREEN.....	61
24	DISABILITAZIONE DELLE AI AD INIZIO PARTITA (SOLO POSTI PER PLAYERS) E ABILITAZIONE DELLA CONSOLE PER L'ADMIN.....	62
25	IMPOSTARE LE POSIZIONI DI RESPAWN.....	62
25.1	IMPOSTARE MANUALMENTE L'ALTEZZA DEL RESPAWN (O UNA COORDINATA)	63
26	ESEMPIO DI FORMA BASE DI UN FILE DESCRIPTION.EXT	63
27	INSERIRE UN TESTO/FRASE ALL'INIZIO DELLA MISSIONE	64
28	CREARE UN VIDEO INTRO DELLA MISSIONE	66
28.1	PRIMO METODO	66
28.2	SECONDO METODO	69
28.3	TERZO METODO	71
28.4	QUARTO METODO (ATTIVABILE ANCHE CON TRIGGER IN GIOCO).....	71
28.5	QUINTO METODO	73
28.6	ALTRO	74
29	MOVIE MAKER SCRIPT.....	75
30	COME RIPRODURRE UN VIDEO DURANTE IL GIOCO.....	75
31	CREARE UNA BANDIERA CON LOGO PERSONALIZZATO	75
32	CHIAMATE IN GLOBALE (REMOTEEEXEC E BIS_FNC_MP)	75
33	CREARE UN IED.....	76
33.1	PRIMO METODO (NON DISINNECABILE)	76
33.2	SECONDO METODO	76
33.3	TERZO METODO	78
33.4	QUARTO METODO	79
34	CREARE UN EFFETTO TORMENTA.....	81
34.1	TORMENTA DI NEVE.....	81
34.2	TORMENTA DI POLVERE O SABBIA	84
34.3	TEMPESTA DI MONSONI.....	86
35	CREARE (SPAWNARE) UN VEICOLO IN UNA DATA POSIZIONE	88



35.1	CREARE UN CARTELLO PER SPAWNARE VEICOLI.....	90
36	AGGIUNGERE UN'IMMAGINE AD UN OGGETTO.....	92
37	CAMBIARE IL VOLTO AI PLAYERS (CAMO FACE) (DA TESTARE).....	93
38	FUOCO DI SOPPRESSIONE	94
38.1	CREARE UN FUOCO DI SOPPRESSIONE.....	94
38.2	TERMINARE (ABORTIRE) UNO SPAWNED OR EXECVMED SCRIPT.....	98
38.3	ALTRA PROCEDURA PER FUOCO DI SOPPRESSIONE	99
38.4	ESEMPIO DI MOVIMENTO A SBALZI CON FUOCO DI SOPPRESSIONE.....	100
39	SUPPORT (ARTILLERY, TRANSPORT ETC.).....	101
39.1	ARTILLERY	101
39.2	HELICOPTER TRANSPORT TAXI.....	101
39.3	ADDING AN ARTILLERY PROVIDER.....	102
39.4	ADDING AN AIRCRAFT CLOSE AIR SUPPORT PROVIDER.....	102
39.5	ADDING A HELICOPTER CLOSE AIR SUPPORT PROVIDER	102
40	AGGIUNGERE CIVILI IN MAPPA.....	103
41	PLAY RADIO MESSAGE.....	104
42	EFFETTI AMBIENTE.....	104
43	WAYPOINTS.....	104
44	HELICOPTER INSERTION/EXTRACTION	106
44.1	INSERTION/EXTRACTION	106
44.1.1	<i>Inserzione.....</i>	<i>106</i>
44.1.2	<i>Estrazione</i>	<i>109</i>
44.1.3	<i>Estrazione 2 (da testare).....</i>	<i>111</i>
44.1.4	<i>Estrazione 3 (via moduli) - Aggiungere un trasporto con elicottero virtuale o fisico</i>	<i>113</i>
44.2	ATTIVAZIONE VIA RADIO	114
45	CREARE UN EFFETTO DI UNO SCHIANTO D'ELICOTTERO	115
46	ATTACCO AD ONDATE SEQUENZIALI.....	118
47	NASCONDERE UNITÀ E FARLE APPARIRE CON TRIGGER	120
48	CREARE UN ATTACCO SUICIDA CON ESPLOSIVO.....	121
48.1	PRIMO METODO	121
48.2	SECONDO METODO	121
48.3	TERZO METODO	122
49	CREARE UN ANIMALE CHE ATTACCA (DA TESTARE)	126
49.1	PRIMO METODO (ATTACCO E PATTUGLIAMENTO CON CANE)	126
49.2	SECONDO METODO	127
50	ESEMPI DI TITOLI DI TESTO A SCOMPARSA - TITLE TEXT EXAMPLES.....	128
51	TELEPORT.....	128
51.1	TELEPORT CON CLICK SU MAPPA	128
51.2	TELEPORT SU UN PUNTO CON ATTIVAZIONE DI UN TRIGGER.....	128
51.3	TELEPORT SU UN PUNTO IDENTIFICATO DA UN OGGETTO O MARKER	128
52	TENERE FERMA UNA IA	129



53	SPAWN AND RESPAWN INTO A PARACHUTE	129
54	MASS AI PARADROP	130
54.1	PRIMO METODO	130
54.2	SECONDO METODO	131
54.3	TERZO METODO	131
55	SUPPLY DROP	132
55.1	PRIMO METODO	132
55.2	SECONDO METODO	133
55.3	TERZO METODO	135
56	SPAWNING UNITS	136
57	COPY POS TO CLIPBOARD.....	138
58	INIZIARE LA PARTITA A BORDO DI UN ELICOTTERO	138
59	SIMPLEX SUPPORT SERVICE MOD – MODULI DA POSIZIONARE	139
60	ADDACTION.....	139
60.1	ADDACTION COMMAND	139
60.2	REMOVE ADDACTION	141
61	OTHER SCRIPTS	143
61.1	COMMON SCRIPTS.....	143
61.1.1	<i>Teleport to empty marker.....</i>	<i>143</i>
61.1.2	<i>Trigger to create a marker - Also creates a hint.....</i>	<i>143</i>
61.1.3	<i>Only fire a trigger when Multiple Triggers Have fired.....</i>	<i>143</i>
61.1.4	<i>Fire a trigger when either Triggers Have fired</i>	<i>143</i>
61.1.5	<i>Create a Virtual Arsenal Object</i>	<i>143</i>
61.1.6	<i>Add BI and ACE arsenal to the same box.....</i>	<i>143</i>
61.1.7	<i>Group patrol.....</i>	<i>143</i>
61.1.8	<i>Delete an Object</i>	<i>143</i>
61.1.9	<i>Set Flying height for chopper.....</i>	<i>143</i>
61.2	MISSION STORY TELLING	144
61.2.1	<i>AI chat.....</i>	<i>144</i>
61.2.2	<i>Set a hint – Inserire un testo a video.....</i>	<i>144</i>
61.3	CONDIZIONI.....	144
61.3.1	<i>Use speed as a condition.....</i>	<i>144</i>
61.3.2	<i>Hold at waypoint until trigger is activated</i>	<i>144</i>
61.3.3	<i>This group activates trigger.....</i>	<i>144</i>
61.3.4	<i>How much does unit know about this?.....</i>	<i>144</i>
61.3.5	<i>Is unit dead?.....</i>	<i>144</i>
61.3.6	<i>Check if engine is on.....</i>	<i>144</i>
61.3.7	<i>Is unit present inside trigger radius?</i>	<i>144</i>
61.3.8	<i>Is unit a human player?</i>	<i>145</i>
61.3.9	<i>What is the damage level of a unit?</i>	<i>145</i>
61.3.10	<i>Is unit in this vehicle?</i>	<i>145</i>
61.3.11	<i>Check number of units in vehicle</i>	<i>145</i>
61.3.12	<i>Distance between two objects/units</i>	<i>145</i>
61.3.13	<i>Check the damage level of unit/object</i>	<i>145</i>
61.3.14	<i>Check time of day.....</i>	<i>145</i>
61.4	MISSION MANIPULATION.....	145
61.4.1	<i>Disable user input.....</i>	<i>145</i>
61.4.2	<i>Rain/fog on command or random</i>	<i>146</i>
61.4.3	<i>Add sounds.....</i>	<i>146</i>
61.4.4	<i>Slow or speed up time</i>	<i>146</i>
61.4.5	<i>Skip Time</i>	<i>146</i>



61.4.6	Place random civilian vehicles around map.....	146
61.4.7	Respawning empty vehicles.....	146
61.4.8	Unit respawns.....	147
61.4.9	Add environmental effects.....	147
61.4.10	Adding medic and drag capability.....	147
61.4.11	End a mission as win/lose.....	147
61.4.12	Send Radio Message.....	147
61.4.13	Create AddAction.....	148
61.4.14	Set date and time of day.....	148
61.4.15	Add score to players.....	148
61.4.16	Disable/ Enable radio ALPHA, BRAVO, etc.....	148
61.4.17	Add on screen hint.....	148
61.4.18	Make helicopter land and drop off troops.....	148
61.4.19	Create mission name and image in menu.....	148
61.4.20	Zeus player respawns in user made mission.....	149
61.4.21	Create Floating Camera.....	149
61.5	OBJECT MANIPULATION.....	149
61.5.1	Set unit/object height.....	149
61.5.2	Set flying height.....	149
61.5.3	Change an objects angle.....	149
61.5.4	Set fuel capacity.....	149
61.5.5	Set Unit Damage.....	150
61.5.6	Light/put out campfire on command.....	150
61.5.7	Units teleport into vehicle.....	150
61.5.8	Do not allow fleeing of a unit.....	150
61.5.9	Create a smoke grenade xx meters above unit.....	150
61.5.10	Accendere il motore di un veicolo.....	150
61.5.11	Change speed of a unit.....	151
61.5.12	Remove all weapons and add weapons to unit.....	151
61.5.13	Adding or removing weapons and ammo from an ammo box.....	151
61.5.14	Not allow unit to enter vehicle.....	151
61.5.15	Attach/detach an object to another.....	152
61.5.16	Display the speed of a unit.....	152
61.5.17	Make a unit prisoner.....	152
61.5.18	Name a group.....	152
61.5.19	Unit joins group.....	152
61.5.20	Teleport a unit or object to any position.....	152
61.5.21	Imposta la postura di un'unità.....	152
61.5.22	Create a name for nearest building (for destroying particular buildings).....	152
61.5.23	Impostare il blocco di un veicolo.....	152
61.5.24	Randomize units/object start location.....	153
61.5.25	Move marker to another marker.....	153
61.5.26	Make a unit move to any object/marker.....	153
61.5.27	Mettere un'unità all'interno di un veicolo.....	153
61.5.28	Moving objects, triggers or markers.....	154
61.5.29	Make object invisible.....	154
61.5.30	Affect all units in a given parameter.....	154
61.5.31	Show classname of unit/object.....	154
61.5.32	Set a waypoint to a AI unit to get a weapon from an ammo boxes.....	155
61.5.33	Creating Vehicles.....	155
61.5.34	Create Dogs.....	155
61.5.35	Map teleport.....	155
61.5.36	God Mode.....	155
61.5.37	Kills Player.....	155
61.5.38	Heal Player, or Player(s).....	155
61.5.39	Destroy Targets.....	155
61.5.40	Disable fatigue (without ACE loaded).....	155
61.5.41	Repair Vehicle.....	155
61.5.42	Remove Fuel from player, or player(s) vehicles.....	156



61.5.43	Add weapon to player or player(s).....	156
61.5.44	Set Player Ammo.....	156
61.5.45	See all people on the map	156
61.5.46	Show all Vehicles on the map with markers.....	156
61.5.47	Attaching a Marker to a Unit.....	157
61.6	COMPLEX FUNCTIONS	157
61.6.1	Unit sets pipebomb.....	157
61.6.2	Improvised explosive device.....	157
61.6.3	Mass paradrop.....	158
61.6.4	Call air support.....	158
61.6.5	HALO Jump	158
61.6.6	Create support options.....	158
61.6.7	Create playable UAV.....	159
61.6.8	Helicopter airlifts object.....	159
61.6.9	MC-130 Refueler.....	159
61.6.10	Add loadouts to aircraft.....	160
61.6.11	Record path for AI to follow	160
62	CORSO EDITOR	162
62.1	GESTIONE FILE MISSIONE.....	162
62.2	ALCUNE FUNZIONI BASE DI SCRIPT	170
62.3	FILES DI LOADOUT	177
62.4	RECUPERO INFORMAZIONI (INTEL)	182
62.5	ENEMY SPAWINING SYSTEM (ESS)	184
63	ENEMY OCCUPATION SYSTEM (EOS)	189
63.1	MODALITÀ BASTION.....	193
64	EAGLE WING EFFECTS	196
65	ATTRIBUTI DEL TRIGGER.....	198
66	EQUIPAGGIAMENTO STANDARD PER GLI OPERATORI.....	201



1 INTRODUZIONE

Questo documento è una raccolta di procedure e scripts che possono tornare utili per un mission maker di Arma 3. Tutto il materiale proviene da varie fonti e tutorials sull'argomento, che si trovano sparsi nel vasto mondo del web. Molte delle procedure sono state riorganizzate, tradotte (lì dove serviva), dettagliate e testate. Altre lo sono state in parte oppure sono state riportate così come trovate, in attesa di una futura rielaborazione e validazione.

Poichè il mondo degli scripts e dei moduli di Arma3 è molto vasto e ci sono svariati modi per fare la stessa operazione (o simile), tale guida non vuole essere ne completa ne unica ed È DA PRENDERE COSÌ COM'È, lasciando al mission maker la facoltà di trovare modi diversi o apportare le dovute correzioni alle procedure qui trattate.

Si perdonino eventuali errori di traduzione fatte alcune volte in maniera sommaria.

Enigx

2 MODIFICARE FILE MISSIONE SCARICATI DAL WEB

Procedura per ricreare il file missione dagli scenari scaricata da Steam.

1. Download il programma PBO Manager [www.armaholic.com].
2. Scaricare il file missione voluto da Steam. Aprire un server host. Caricare la mappa e selezionare la missione. Farla partire per giocarla. Fare salva (viene creato il file missione)
3. Trovare lo scenario salvato in "C:\Users\(\yourname)\Documents\Arma 3 - Other Profiles\(\yourname)\Saved\steam (Questo può essere diverso ma dovrebbe trovarsi in tale cartella, controllare e trovare il file).
4. Aprire il programma PBO Manager. Cliccare su "Open file" e selezionare il file dello scenario salvato. Premere "Open".
5. Creare una nuova cartella all'interno della cartella di default "missions" che generalmente è in "C:\Users\(\yourname)\Documents\Arma 3 - Other Profiles\(\yourname)\missions". Nominarla come "(nameofscenario).(nameofmap) example: Fob.clafghan.
6. In PBO Manager si ha ora un file chiamato "mission.sqm". Premere il pulsante + (questo espanderà il file). Selezionare tutti i file e copiarli nella cartella create al passo 5.
7. Lanciare il gioco ed avviare l'Eden editor. Cliccare open file ed esso si troverà sotto "Missions".

In generale per scompattare i file PBO si usa la stessa procedura qui sopra tramite l'uso di PBO Manager.

3 SALVATAGGIO MISSIONE

Una volta creata la missione procederemo al suo salvataggio. Facendo salva essa verrà salvata in una cartella che si trova di default in C:\Users\(\yourname)\Documents\Arma 3 - Other Profiles\(\yourname)\missions. All'interno verrà creata una cartella che si chiamerà "\nomemissionedato.nomedellamappautilizzata". Il file risultante sarà mission.sqm.

Nota: nella schermata di salvataggio c'è una casella di spunta con la possibilità di salvarlo come file binarizzato. Se si sceglie questa opzione il file è illeggibile con un normale notepad.

È consigliabile salvarlo sempre NON BINARIZZATO. Così facendo è leggibile con un notepad e ciò può essere utile poichè contiene diverse informazioni, come ad esempio gli addon usati (come le mod).

Anche da Attributi/generali si può spuntare la binarizzazione come salvataggio di default.

È consigliabile, ma non obbligatorio, dare un nome alla missione (e quindi al file salvato) senza usare gli spazi tra una parola e l'altra.

Ricordarsi, nel caso si rinomini la cartella, di lasciare sempre il ".nomedellamappautilizzata" che identifica la mappa su cui è impostata. Come caso estremo se si cambia il nome della mappa si trasferisce l'intera missione su quest'ultima. Gli oggetti saranno nelle coordinate in cui risultano già definiti.



4 DISABILITARE IL SALVATAGGIO IN MULTIPLAYER

Per disabilitare la possibilità di salvataggio da parte dei clients basta creare un file `initServer.sqf` all'interno della cartella missione ed inserirvi la stringa

```
enableSaving [false,false];
```

5 DISABILITARE LA CHAT

Per disabilitare la chat mettere nel `description.ext` la stringa

```
//Disabilita chat  
// 0 = Globale, 1 = Side, 2 = Comando, 3 = Gruppo, 4 = Veicolo, 5 = Diretta, 6 = Sistema  
// {id canale, disabilita chat scritta, disabilita chat vocale}  
disableChannels[] =  
{0,false,true},{1,false,true},{2,false,true},{3,false,true},{4,false,true},{5,false,true},{6,false,true}};
```

6 VIRTUAL ARSENAL

Come Creare un Arsenale Virtuale.

Incollare questa stringa nell' "init" dell'oggetto cassa o qualsiasi altro oggetto che si sta usando:

Senza testo nella descrizione:

```
0 = ["AmmoboxInit",[this,true]] spawn BIS_fnc_arsenal;
```

Con Testo:

```
this addAction ["ARSENALE VIRTUALE",{["Open",true] call BIS_fnc_arsenal; }];
```

7 COME TROVARE LE CLASSNAMES PER GLI ADDON (MOD)

Supponendo che lo sviluppatore della mod non abbia fornito le classnames delle unità, puoi ritrovarli caricando la mod nel gioco e andando nell'editor. Una volta lì hai due opzioni:

1. Posizionare le unità, le armi o i veicoli per cui desideri i nomi delle classi e salva la missione. Apri il file chiamato `mission.sqm`. I nomi delle classi saranno qui.
Nota: per visualizzare l'`sqm` con un normale Notepad occorre NON salvarlo come file binarizzato, spuntando la casella relativa nella schermata di salvataggio
2. Aprire il visualizzatore di configurazione tramite la barra degli strumenti dell'editor o premendo ESC/pulsante configurazione. Sfoglia la sezione `cfgVehicles` e dovresti essere in grado di trovare i nomi delle classi per la maggior parte dei componenti aggiuntivi.

8 SELEZIONARE LOADOUT E FREQUENZE RADIO PREDEFINITI DALL'ARSENAL

8.1 SELEZIONARE LOADOUT PREDEFINITI

Per evitare di sprecare tempo nell'impostazione dei loadout in partita si può impostare l'arsenal in modo tale che:

1. il giocatore si avvicina alla cassa degli oggetti
2. viene visualizzato un menu dalla cassa, tramite rotella del mouse, con la lista dei loadout come:
"Autorifleman
Rifleman
Team leader
Medic"
3. Il giocatore seleziona qualsiasi ruolo
4. Il giocatore cambia automaticamente il suo loadout corrente con il loadout selezionato da menu.

La procedura per far ciò è la seguente:

***** PRIMO PASSO *****

1. Prima di tutto, posizionare una cassa (o anche un qualsiasi oggetto) sulla mappa.
2. Quindi nel campo init di quell'oggetto digitare la seguente stringa

```
this addAction ["Team Leader", "scripts\Team_Leader_Loadout.sqf"];
this addAction ["Rifleman", "scripts\Rifleman_Loadout.sqf"];
this addAction ["Grenadier", "scripts\Grenadier_Loadout.sqf"];
this addAction ["Medic", "scripts\Medic_Loadout.sqf"];
this addAction ["Rifleman AT/AA", "scripts\Rifleman_AT-AA_Loadout.sqf"];
this addAction ["Machine Gunner", "scripts\Machine_gunner_Loadout.sqf"];
this addAction ["Radio onde corte", "scripts\Radio_Onde_Corte.sqf"];
this addAction ["Radio onde lunghe", "scripts\Radio_Onde_Lunghe.sqf"];
```

Giusto come spiegazione del codice che ha come forma generale la :

```
this addAction ["Testo da far comparire azionando rotella mouse", "cartella_script\file_loadout.sqf"];
```

this: seleziona l'oggetto in questione

addAction: chiama il comando addAction per aggiungere il nome impostato in "" nel menu attivabile da rotella durante il gioco e lanciare lo script assegnato. Esempio

["Rifleman"] - il nome dell'opzione che apparirà nel menu quando ci si avvicina alla cassa con la rotellina del mouse

["scripts \ Rifleman_Loadout.sqf"] - il file di script che contiene il loadout / etc che si desidera.

***** SECONDO PASSO *****

Per creare i files .sqf dei loadout per ogni ruolo:

1. Crea il loadout desiderato nell'arsenal. Per esempio, si può creare un loadout di Rifleman usando vanilla.
2. Da quest'arsenal fare clic su ESPORTA. Questo copierà il loadout negli appunti
3. Creare un file di testo con il block notes nella cartella della missione e nominarlo come "Rifleman_Loadout.sqf". Assicurarsi che il tipo di file sia SQF o convertirlo da .txt a .sqf
Per impostazione predefinita, la cartella della missione è in (~ / Documents / Arma 3 - Altri profili / (il tuo nome) / missioni / (nome della missione) / script)
NOTA: se non è presente alcuna cartella di script, crearne una.
4. Incollare (CTRL + V) il loadout del Rifleman esportato nel file Rifleman_Loadout.sqf
5. Modificalo per farlo sembrare così

```
comment "Exported from Arsenal";

_unit = _this select 1;

comment "Remove existing items";
removeAllWeapons _unit;
removeAllItems _unit;
removeAllAssignedItems _unit;
removeUniform _unit;
removeVest _unit;
removeBackpack _unit;
removeHeadgear _unit;
removeGoggles _unit;

comment "Add containers";
_unit forceAddUniform "U_B_CombatUniform_mcam";
_unit addItemToUniform "FirstAidKit";
for "_i" from 1 to 2 do {_unit addItemToUniform "30Rnd_65x39_caseless_mag"};
_unit addItemToUniform "Chemlight_green";
_unit addVest "V_PlateCarrier1_rgr";
for "_i" from 1 to 7 do {_unit addItemToVest "30Rnd_65x39_caseless_mag"};
for "_i" from 1 to 2 do {_unit addItemToVest "16Rnd_9x21_Mag"};
_unit addItemToVest "SmokeShell";
_unit addItemToVest "SmokeShellGreen";
_unit addItemToVest "Chemlight_green";
for "_i" from 1 to 2 do {_unit addItemToVest "HandGrenade"};
_unit addHeadgear "H_HelmetB";

comment "Add weapons";
_unit addWeapon "arifle_MX_F";
_unit addPrimaryWeaponItem "acc_pointer_IR";
_unit addPrimaryWeaponItem "optic_Aco";
```



```
_unit addWeapon "hgun_P07_F";

comment "Add items";
_unit linkItem "ItemMap";
_unit linkItem "ItemCompass";
_unit linkItem "ItemWatch";
_unit linkItem "ItemRadio";
_unit linkItem "NVGoggles";
```

6. NOTA: eliminare nelle stringhe tutta la parte relative a "Set identity", altrimenti tutti avranno la stessa faccia!!

7. Ricordarsi di aggiungere

```
_unit = _this select 1;
```

nella prima parte dello script.

Giusto come spiegazione del codice:

```
_unit      nuovo nome variabile
= this select 1
```

IMPORTANTE:

Nel file sqf che si stà creando sostituire la variabile del codice

```
this
```

con la variabile

```
_unit
```

Infatti di default, esportando dall'arsenale, viene creato lo script avente dentro la variabile

`this`

che è usata per selezionare il giocatore / personaggio. Questa deve essere sostituita da

`_unit`

Alternativa

È possibile evitare di sostituire tutti i `this`, esportati da vanilla, in `_unit`, optando per un modo più "elegante". Basta inserire in ogni `sqf` dei ruoli esportati queste due righe.

```
this = player; // Va messa all'inizio del file del loadout
this = nil; // Va messa alla fine del file del loadout
```

La prima all'inizio del file del loadout e la seconda alla fine, es.:

```
this = player;
```

```
comment "Remove existing items";
removeAllWeapons this;
removeAllItems this;
removeAllAssignedItems this;
removeUniform this;
removeVest this;
removeBackpack this;
removeHeadgear this;
removeGoggles this;

comment "Add weapons";
this addWeapon "rhs_weap_m4a1";
this addPrimaryWeaponItem "rhsusf_acc_anpeq15";
this addPrimaryWeaponItem "rhsusf_acc_ACOG2";
this addPrimaryWeaponItem "rhs_mag_30Rnd_556x45_M855A1_Stanag";
this addWeapon "rhsusf_weap_glock17g4";
this addHandgunItem "rhsusf_mag_17Rnd_9x19_JHP";

comment "Add containers";
this forceAddUniform "rhs_uniform_acu_ucp";
this addVest "rhsusf_iotv_ucp_Teamleader";
this addBackpack "rhsusf_assault_eagleaiii_ucp";

comment "Add binoculars";
this addWeapon "Rangefinder";
```

```
comment "Add items to containers";
```



```
this addItemToUniform "FirstAidKit";
this addItemToUniform "rhsusf_ANPVS_14";
this addItemToUniform "rhsusf_patrolcap_ucp";
this addItemToUniform "ACE_EarPlugs";
for "_i" from 1 to 2 do {this addItemToUniform "ACE_CableTie";};
this addItemToUniform "ACE_microDAGR";
this addItemToUniform "ACE_MapTools";
for "_i" from 1 to 3 do {this addItemToUniform "ACE_morphine";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_epinephrine";};
this addItemToVest "rhs_mag_m18_purple";
this addItemToVest "rhs_mag_m18_yellow";
this addItemToVest "rhs_mag_m67";
this addItemToVest "rhsusf_mag_17Rnd_9x19_JHP";
for "_i" from 1 to 11 do {this addItemToVest "rhs_mag_30Rnd_556x45_M855A1_Stanag";};
for "_i" from 1 to 10 do {this addItemToBackpack "ACE_morphine";};
for "_i" from 1 to 8 do {this addItemToBackpack "ACE_epinephrine";};
for "_i" from 1 to 20 do {this addItemToBackpack "ACE_fieldDressing";};
this addItemToBackpack "ACE_Clacker";
for "_i" from 1 to 5 do {this addItemToBackpack "DemoCharge_Remote_Mag";};
this addHeadgear "rhsusf_ach_helmet_ESS_ucp";

comment "Add items";
this linkItem "ItemMap";
this linkItem "ItemCompass";
this linkItem "ItemWatch";
this linkItem "TFAR_anprc152";
this linkItem "ACE_NVG_Wide";

this = nil;
```

Molto semplicemente, assegna la variabile `this` temporaneamente a `player` riannullandola alla fine.

8.2 IMPOSTARE LA RADIO DIRETTAMENTE DALL'ARSENAL

Seguendo i passi precedenti riguardo ai loadout è possibile far impostare le frequenze dei canali radio (corte o lunghe) direttamente tramite la rotellina del mouse sull'arsenal in modo che compaiano come voci nel menu contestuale.

Per far ciò occorre:

1. Creare il file `“.sqf”` con i comandi per far impostare le frequenze da script. Per far ciò è sufficiente richiamare le apposite funzioni come di seguito

Contenuto file `sqf` per le Onde corte

```
comment "Frequenze radio onde corte";
[(call TFAR_fnc_activeSwRadio), 1, "184"] call TFAR_fnc_SetChannelFrequency;
```

Contenuto file `sqf` per le Onde lunghe

```
comment "Frequenze radio onde lunghe";
[(call TFAR_fnc_activeLrRadio), 1, "30"] call TFAR_fnc_SetChannelFrequency;
```

nella call `[(call TFAR_fnc_activeSwRadio), 1, "184"]`, 1 è il numero del canale radio e 184 la frequenza che sarà impostata su di esso.

2. Salvare il file `.sqf` nella cartella script (se non esiste crearla). Per esempio (`~/Documents/Arma 3 - Other Profiles/(your name)/missions/nomemissione/scripts/`)
3. Copiare la seguente stringa nel campo `init` dell'oggetto dell'arsenal

```
this addAction ["Setta radio onde corte", "scripts\nome_del_file_radio_set.sqf"];
```

NOTA: è possibile settare le frequenze anche tramite le opzioni degli addon senza “gestire” un file esterno.

8.3 SCRIPT PER ARSENAL (ESEMPIO)

Script per creare loadouts e impostare le frequenze dei canali radio tramite rotella nell'arsenal.

Nota: creare prima i file `.sqf` come descritto sopra

Menu senza colori (scritta nera)



```
this addAction ["Team Leader TL","scripts\teamleader_Loadout.sqf"];
this addAction ["Team Leader Ilbe","scripts\teamleader_ilbe_Loadout.sqf"];
this addAction ["Granatiere GL","scripts\granatiere_Loadout.sqf"];
this addAction ["Medico","scripts\medico_Loadout.sqf"];
this addAction ["Fuciliere FC","scripts\fuciliere_Loadout.sqf"];
this addAction ["Fuciliere AT/AA","scripts\fuciliereAT_Loadout.sqf"];
this addAction ["Marksman","scripts\marksman_Loadout.sqf"];
this addAction ["Mitragliere MG","scripts\mitragliere_Loadout.sqf"];
this addAction ["Specialista esplosivi","scripts\specialista esplosivi_Loadout.sqf"];

this addAction ["-----",""]; //serve per mettere uno spazio tra le due liste

this addAction ["Setta radio onde corte","scripts\Radio_Onde_Corte.sqf"];
this addAction ["Setta radio onde lunghe","scripts\Radio_Onde_Lunghe.sqf"];
```

Menu con colori e font grandi

```
this addAction ["<t color='#ff0000' size='1.3'>Team Leader TL</t>","scripts\teamleader_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Team Leader
Ilbe</t>","scripts\teamleader_ilbe_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Granatiere GL</t>","scripts\granatiere_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Medico</t>","scripts\medico_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Fuciliere FC</t>","scripts\fuciliere_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Fuciliere AT/AA</t>","scripts\fuciliereAT_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Marksman</t>","scripts\marksman_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Mitragliere MG</t>","scripts\mitragliere_Loadout.sqf"];
this addAction ["<t color='#ff0000' size='1.3'>Specialista esplosivi</t>","scripts\specialista
esplosivi_Loadout.sqf"];

this addAction ["-----",""]; //serve per mettere uno spazio tra le due liste

this addAction ["<t color='#00FF00' size='1.3'>Setta radio onde
corte</t>","scripts\Radio_Onde_Corte.sqf"];
this addAction ["<t color='#FFFF00' size='1.3'>Setta radio onde
lunghe</t>","scripts\Radio_Onde_Lunghe.sqf"];
```

8.3.1 ALTERNATIVA CON FILE ESTERNO

È possibile evitare di mettere le addAction nell'init della cassa optando per un modo più "elegante", cioè gestendole tramite un file esterno da richiamare nell'init cassa.

1. Creare la cartella \loadout all'interno della cartella missione
Nota: qui si è chiamata \loadout la cartella che prima era \scripts. Ovviamente basta stabilirne un nome ed essere congruenti nelle istruzioni delle stringhe
2. Creare un file esterno contenente le addAction. Creare il menu.sqf nella cartella \loadout. Mettere qui la stringa che sopra era messa nell'init della cassa

```
_cartello = _this select 0;
_cartello addAction ["<t color='#ff0000' size='1.3'>Selezione LOADOUT</t>",""];
_cartello addAction ["<t color='#ff0000' size='1.3'>-----</t>",""];
_cartello addAction ["<t color='#ff0000' size='1.3'>Team Leader
TL</t>","loadout\teamleader_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Team Leader
Ilbe</t>","loadout\teamleader_ilbe_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Granatiere
GL</t>","loadout\granatiere_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Medico</t>","loadout\medico_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Fuciliere FC</t>","loadout\fuciliere_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Fuciliere
AT/AA</t>","loadout\fuciliereAT_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Marksman</t>","loadout\marksman_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Mitragliere
MG</t>","loadout\mitragliere_Loadout.sqf"];
_cartello addAction ["<t color='#ff0000' size='1.3'>Specialista esplosivi</t>","loadout\specialista
esplosivi_Loadout.sqf"];

_cartello addAction ["-----",""]; //serve per mettere uno spazio tra le due liste

_cartello addAction ["<t color='#00FF00' size='1.3'>Selezione CANALI RADIO</t>",""];
_cartello addAction ["<t color='#00FF00' size='1.3'>-----</t>",""];
_cartello addAction ["<t color='#00FF00' size='1.3'>Imposta radio onde
corte</t>","loadout\Radio_Onde_Corte.sqf"];
```



```
_cartello addAction [ "<t color='#FFFF00' size='1.3'>Imposta radio onde  
lunghe</t>", "loadout\Radio_Onde_Lunghe.sqf" ];
```

3. All'interno della cartella \loadout mettere i file del loadout ruoli di cui sopra

4. Nell'init dell'oggetto cassa mettere la stringa che richiama il file menu.sqf

```
null = [this] execVM "loadout\menu.sqf";
```

8.4 SELEZIONARE DEI FILE SQF RANDOM PER UN DETERMINATO RUOLO

Se si vuole richiamare un file sqf random tra diversi files per la chiamata dell'addAction, si può impostare in tal modo la stringa addAction sulla cassa (o su file esterno come sopra):

```
this addAction [ "Team Leader TL",  
selectRandom[ "scripts\teamleader_Loadout_1.sqf", "scripts\teamleader_Loadout_2.sqf", "scripts\teamleader_  
Loadout_3.sqf" ] ];
```

questa dovrebbe scegliere tra i 3 diversi file di loadout per il ruolo TL chiamato nel menu dalla addAction.

Questa opzione può essere utile per impostare diversi loadout per il singolo ruolo, per esempio in un MP con contractors aventi uniformi tutte diverse.

Altro materiale trovato

Is there any way to make a script that will randomly pick between different map add on files each restart. (E.g select 1 at random from 1.sqf, 2.sqf, 3.sqf or 4.sqf)

yeah

```
_scriptArray = [ "1.sqf", "2.sqf", "3.sqf" ];
```

```
_randomScript = [_scriptArray] call BIS_fnc_selectRandom;
```

You can then execute the script like this:

```
[ ] execVM format[ "%1", _randomScript ];
```

Obviously you need to make sure that you include the path to the scripts in the execVM call.

```
_pathToScripts = "Scripts\Coolpath\";  
_scriptArray = [ "1.sqf", "2.sqf", "3.sqf" ];  
_randomScript = [_scriptArray] call BIS_fnc_selectRandom;  
[ ] execVM format[ "%1%2", _pathToScripts, _randomScript ];
```

Altro

if possible there would be a list of missions which it would choose randomly from a laptop.

you could add addAction to that laptop with either a list of missions or have a mission selected randomly. heres how to select from a list of missions. inside of mission1/mission2/mission3.sqf is where you would have your missions scripted out or to start your triggers and such.

```
laptop addAction [ "Mission 1", {execVM "mission1.sqf"}];  
laptop addAction [ "Mission 2", {execVM "mission2.sqf"}];  
laptop addAction [ "Mission 3", {execVM "mission3.sqf"}];
```

heres how you could have a random mission selected

```
laptop addAction [ "Random mission", {execVM "selectMission.sqf"}];
```

inside of selectMission.sqf this will select a random number and execute one of the mission.sqf files



```
_mission = random[1,2,3];  
  
switch (_mission) do  
{  
    case 1: {execVM "mission1.sqf"};  
    case 2: {execVM "mission2.sqf"};  
    case 3: {execVM "mission3.sqf"};  
};
```

8.5 ASSEGNARE L'ABILITÀ DI MEDICO O INGEGNERE (SPECIALISTA ESPLOSIVI) AD UN'UNITÀ DA SCRIPT

Se si vuole assegnare l'abilità di medico (o ingegnere) ad un'unità si può inserire una stringa dedicata alla fine del suo loadout sqf.

```
this setVariable ["ace_medical_medicClass", 0, true];  
sleep 1;  
this setVariable ["ace_medical_medicClass", 1, true];  
sleep 1;  
hint parsetext "<t font='TahomaB' t size='1.25' t color='#FF0000'>Sei abilitato come Medico</t>";
```

questa azzera l'abilitazione da medico, attende 1 secondo, poi riattribuisce l'abilitazione da medico e fa comparire un messaggio a video.

Se si sostituisce al numero 1 di ["ace_medical_medicClass", 1, true] il numero 2 si assegna l'abilità di dottore.

Analogamente per l'ingegnere si metterà nel suo loadout la stringa:

```
this setVariable ["ACE_IsEngineer", 0, true];  
sleep 1;  
this setVariable ["ACE_IsEngineer", 1, true];  
sleep 1;  
hint parsetext "<t font='TahomaB' t size='1.25' t color='#FF0000'>Sei abilitato come Ingegnere Avanzato</t>
```

Se si sostituisce al numero 2 di ["ACE_IsEngineer", 1, true] il numero 2 si assegna l'abilità di ingegnere avanzato

NOTA: verificare che sia abilitato il "save loadout" nelle impostazioni della ace. Il Salva equipaggiamento si trova in "Ace/Riapparizione"

NOTA: allo specialista esplosivi va assegnata l'abilità d'ingegnere!

Altra funzione trovata è questa:

```
player getUnitTrait "Medic";
```

questa però forse non funziona con la ace.

Funzione **getUnitTrait**

Description:

Gets the current value of a trait of the given unit.

For the default values please visit [setUnitTrait](#).

Syntax:

unit **getUnitTrait** skillName

Parameters:

Unit: [Object](#)

skillName: [String](#) - Name of the skill

Return Value:

[Bool](#) or [Scalar](#) - Value of the trait

[Nothing](#) if unit is empty vehicle

Example 1:

```
player getUnitTrait "Medic";
```

Funzione **setUnitTrait**

**Description:**

Enables or disables a trait or alters a trait of the given unit. Custom trait can only be a [bool](#).

Default traits are:

- [Number](#) **audibleCoef** - A lower value means the unit is harder to hear
- [Number](#) **camouflageCoef** - A lower value means the unit is harder to spot
- [Number](#) **loadCoef** - Equipment weight multiplier affecting fatigue and stamina
- [Boolean](#) **engineer** - Ability to partially repair vehicles with toolkit, equivalent to engineer = 1; in [CfgVehicles](#)
- [Boolean](#) **explosiveSpecialist** - Ability to defuse mines with toolkit, equivalent to canDeactivateMines = 1; in [CfgVehicles](#)
- [Boolean](#) **medic** - Ability to treat self and others with medikit, equivalent to attendant = 1; in [CfgVehicles](#)
- [Boolean](#) **UAVHacker** - Ability to hack drones, equivalent to uavHacker = 1; in [CfgVehicles](#)

Syntax:

unit **setUnitTrait** [skillName, value, isCustom]

Parameters:

unit: [Object](#) - The unit whose trait should be set

skillName: [String](#) - Name of the skill

value: [Bool](#) or [Number](#) - Value (see above)

isCustom (optional): [Bool](#) - Must be set to [true](#) in case of custom

Return Value:

Nothing

Example 1:

```
player setUnitTrait ["Medic",true];
```

9 RIPRODURRE UN SUONO ESTERNO IN UN OGGETTO O IN UN TRIGGER

Un 3D Positional Audio è il suono/musica che proviene da una posizione specifica nel gioco. È molto utile per aggiungere un'immersione più realistica alle missioni in quanto può essere utilizzato per creare un'atmosfera di sottofondo e per effetti sonori in generale.

Il suono può essere generato una volta sola oppure fatto ripetere più volte in un loop continuo.

Di seguito il modo più semplice per creare un audio posizionale 3D con loop in Arma 3.

Vedi anche §61.4.3.

Quello che serve:

- 1) Un file audio convertito in formato .ogg
- 2) Un file description.ext nella cartella della missione (nei seguenti passaggi come crearne uno)

Step 1 - Audio File

Innanzitutto, è necessario un file audio in formato .ogg per la riproduzione.

Ci sono molti convertitori online gratuiti. Uno freeware per esempio è [LameXP](#) che consente di convertire file audio in diversi formati anche mp3. Altro per esempio è Audacity

Una volta generato il file audio .ogg si deve metterlo nella cartella della missione che in generale è "... \Documents\Arma 3 - Other Profiles\Nome utente\missions\MissionName\"

Per mantenere le cose più pulite, è preferibile creare una cartella "Sound" nella cartella della missione per archiviare tutti i file audio della missione.

Creare quindi la cartella musicale "sound" (od anche "music" o qualunque altro nome) ed inserire il file audio all'interno di questa cartella.

Step 2 - Description File

Ora, è necessario creare un file "[description.ext](#)" nella cartella della missione.

- 1) Andare nella cartella della mission \Documents\Arma 3 - Other Profiles\Nome utente\missions\MissionName\"
- 2) Creare un nuovo documento di testo.
- 3) Incollare il seguente codice nel documento di testo:



```
class CfgSounds {
    sounds[] = {};
    class sound01 // This (sound01) is the classname you will use to refer to the sound in your
code.
    {
        name = "Sound01"; // This line is only needed if you want to be able to select the sound in
the Trigger effects drop-down menus for sound. Call it whatever you want otherwise remove the line.
        sound[] = {"sound\YOURSOUNDFILE.ogg",1,1,18,1,1,1,0}; // This is the path to the audio
file in your mission folder. The first 1 is volume. The second 1 is speed of playback.
        titles[] = {};
        empty[] = {"",0,0,0,0,0,0,0};
    };
};
```

- 4) Inserire il percorso esatto del file audio nel codice in modo che corrisponda al percorso del file audio ("sound\YOURSOUNDFILE.ogg" con "nome_cartella_audio_dato\YOURSOUNDFILE.ogg").
- 5) Salvare il documento di testo come description.ext nella cartella della missione (Verrà chiesto se sei sicuro di voler cambiare il formato del file. Fai clic su Sì.)

Step 3 – Uso del suono in un oggetto

Si può far riprodurre il suono inserendolo all'interno di un oggetto qualsiasi posizionato sulla mappa. Quando vicini a quell'oggetto il suono verrà sentito.

È possibile anche attivare un trigger che riproduca il suono come verrà spiegato più avanti.

Per riprodurre il suono in un oggetto occorrerà:

- 1) Torna all'editor "eden" e posizionare un oggetto sulla mappa.
- 2) Fare doppio clic sull'oggetto e inserire il seguente codice della "call" nel campo **init** dell'oggetto:

```
this say3D "sound01";
```

NOTA: "sound01" è il nome assegnato alla classe e non al "name"

Tra virgolette il nome della classname del suono "sound01" che si è definito nel file description.ext. Se si è chiamato il classname del suono in modo diverso, ovviamente va messo quest'altro nome, cioè `this say3D "nomedellaclasse dato";`.

Così il suono viene ripetuto una volta dall'oggetto desiderato (ad avvio scenario).

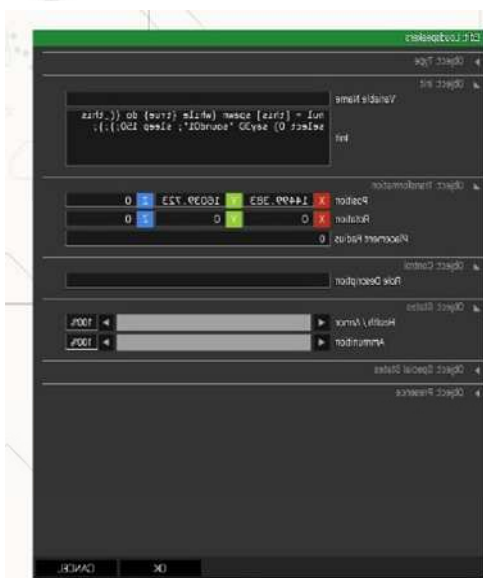
Se si vuole ripetere il suono all'infinito (ciclicamente) al posto della precedente istruzione nell'init dell'oggetto va messa quest'altra:

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "sound01"; sleep 150;}};
```

Nota: 150 alla fine della stringa è quanto è lungo in secondi il brano/suono che si sta riproducendo. Quando il conto alla rovescia arriva a 0 ripeterà l'intero script, mettendolo quindi in un ciclo.

Sleep è la funzione che fa attendere l'istruzione in gioco prima di riavviare la traccia audio, quindi se si imposta correttamente, si otterrà un loop audio continuo

- 3) 150 è la lunghezza in secondi del file audio che (ad esempio in questo caso) è lungo 2 minuti e 30 secondi. Quindi (2x60 + 30 = 150). **N.B.: Regolare questo numero a seconda della lunghezza del file audio utilizzato.**



Step 3 – Uso del suono in un trigger (interruttore)

Opzionale - Utilizzo di un trigger. Vedi anche §0.

Per utilizzare questo metodo tramite un trigger, attenersi alla seguente procedura che sostanzialmente è la precedente solo che la stringa della "call" è posta all'interno di un trigger:

- 1) Posizionare il trigger sulla mappa.
- 2) Fare doppio clic sul trigger ed inserire il seguente codice nel campo **onActivation** del trigger:

```
this say3D "sound01";
```

(Ancora una volta, il nome della classe audio "sound01", deve essere uguale a quello definito nel file description.ext)

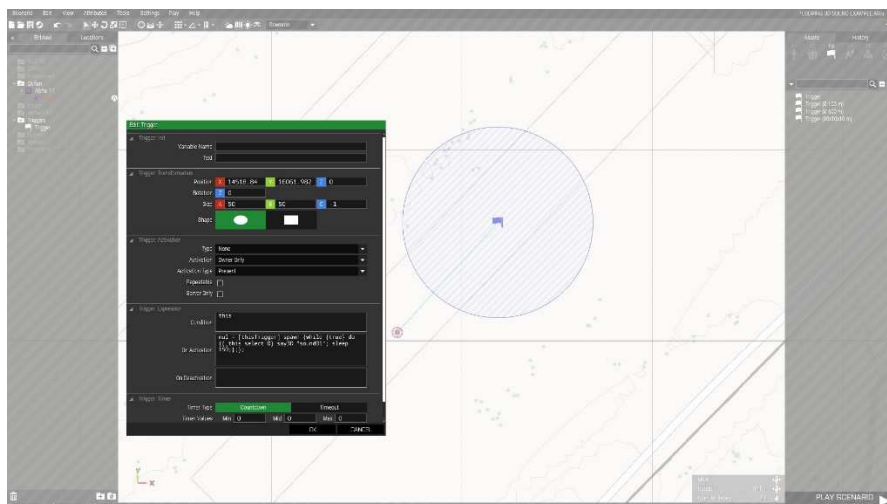
N.B.: qui il nome **NON** è quello della classe ma quello definito nella stringa `name = "sound01";`. È bene usare lo stesso nome della class

Come sopra se si vuole che il suono sia riprodotto in loop mettere questa stringa nel campo **onActivation**

```
nul = [thisTrigger] spawn {while {true} do {(_this select 0) say3D "sound01"; sleep 150;}};
```

Si noti che `nul = [this]` è stato cambiato in `nul = [thisTrigger]`

- 3) Regolare la durata del suono (150 nell'esempio) in modo che corrisponda alla lunghezza del file audio utilizzato in secondi.
- 4) Ora si possono impostare le condizioni del trigger. La loro realizzazione attiverà il loop audio con il suono proveniente dalla posizione del trigger.





9.1 CODICE DA INSERIRE (FUNZIONA)

Nell'init dell'oggetto:

```
this say3D "ISISpropoganda";
```

ISISpropoganda è il nome della class identificata del file description.

nel file Description.ext:

```
class CfgSounds {
    sounds[] = {};
    class ISISpropoganda
    {
        name = "ISISpropoganda"; // This line is only needed if you want to be able to select the
        sound in the Trigger effects drop-down menus for sound. Call it whatever you want otherwise remove the
        line.
        sound[] = {"sound\ISISpropoganda.ogg",1,1,18,1,1,1,0}; // The first 1 is volume. The second
        1 is speed of playback.The 18 indicates the distance at which the sound will be heard;
        titles[] = {};
        empty[] = {"",0,0,0,0,0,0,0};
    };
};
```

Così il suono viene ripetuto una sola volta sull'oggetto desiderato.

Se si vuole ripetere il suono all'infinito (ciclicamente) al posto della precedente istruzione nell'init dell'oggetto va messa quest'altra:

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "ISISpropoganda";
sleep 63;}};
```

Nota: 63 è la durata in secondi del file audio in oggetto (ISISpropoganda). Cambiare la durata a seconda del file che si sta inserendo

Nota: se si vuole che si attivi con un trigger, mettere la call say3D direttamente nell'init del trigger

Doppio click sul trigger e inserire la stringa seguente nel campo **onActivation**:

```
this say3D "ISISpropoganda";
```

Il suono classname "ISISpropoganda", deve essere lo stesso come definito nel description.ext file.

Se si vuole che io suono sia riprodotto in loop mettere questa stringa nel campo **onActivation**

```
nul = [thisTrigger] spawn {while {true} do {(_this select 0) say3D "ISISpropoganda"; sleep 63;}};
```

Notare come la condizione precedente nul = [this] è stata cambiata in nul = [thisTrigger]

Adesso si può aggiustare la condizione del trigger (blufor present per esempio). Soddisfacendola, il loop del suono si attiverà automaticamente.

Esempi di loop cicliche:

sfxsound3

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "sfxsound3"; sleep 31;}};
```

woman_crying

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "woman_crying"; sleep 22;}};
```

Sad_Male

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "Sad_Male"; sleep 24;}};
```

ALTRE FONTI

Vedi anche <https://community.bistudio.com/wiki/Description.ext#CfgSounds>

```
class CfgSounds
{
    sounds[] = {};
    class wolf1
    {
        // how the sound is referred to in the editor (e.g. trigger effects)
        name = "my_wolf_sound";
        // filename, volume, pitch, distance (optional)
```



```
sound[] = { "fx\wolf1.ogg", 1, 1, 100 };
// subtitle delay in seconds, subtitle text
titles[] = { 1, "*wolf growls*" };
};
class playerISHurt
{
    name = "Man, you are injured?! - by NPC";
    sound[] = { "Sound\playerIsHurt1.ogg", 1, 1 };
    titles[] = {
        0, "Dude! You are covered in blood!",
        6, "Are you sure you are OK?!"
    };
};
class AnotherSound
{
    name = "Sound with advanced subtitle options";
    sound[] = { "somesound.ogg", 1, 1 };
    forceTitles = true; // will display titles even if global show
titles option is off // titles font family
titlesFont = "LCD14"; // titles font size
titlesSize = 0.1; // treat titles as Structured Text (see below)
titlesStructured = true;
titles[] = {
    0, <t color="#ff0000">Red text</t>,
    1, <t color="#00ff00">Green text</t>
};
};
};
```

9.2 RIPRODURRE DELLA MUSICA ALL'INTERNO DI UN VEICOLO

<http://www.armaholic.com/forums.php?m=posts&q=33227>

Di seguito uno script che riproduce della musica in un veicolo. (auto civili e altri veicoli)

In questo script è possibile:

- Modificare lo script e aggiungere più brani.
- Riprodurre / arrestare qualsiasi brano.
- Riprodurre un brano casuale.

1) Posizionare un veicolo e nel relativo init inserire la stringa

```
PlaySongs = this addaction ["Start music
player", "radio_music\musicplayer.sqf", true, 1, false, true, "", "_this in _target"];
```

2) Nella cartella missione creare una cartella denominata "music". All'interno della cartella music, posizionare le canzoni in formato .ogg e chiamale come **track1**, **track2**, **track3** o qualsiasi altro nome coerente con lo script.

3) Creare il file description.ext nella cartella missione ed inserire la stringa

```
class CfgSounds
{
    sounds[] = { song1, song2, song3 };

    class song1
    {
        name = "song1";
        sound[] = { "music\track1.ogg", 0.5, 1 };
        titles[] = { };
    };
    class song2
    {
        name = "song2";
        sound[] = { "\music\track2.ogg", 0.5, 1 };
        titles[] = { };
    };
    class song3
    {
        name = "song3";
        sound[] = { "\music\track3.ogg", 0.5, 1 };
        titles[] = { };
    };
};
```



- 1) Creare la cartella "radio_music" nella cartella missione. Creare il file musicplayer.sqf nella cartella radio_music ed inserire la stringa

```
//Musicplayer.sqf
if (!isServer) exitWith {}; // To run this in MP
_vehicle = _this select 0;

hintSilent "Music player switched on!";

sleep 2;

hintsilent "";

sleep 2;
playtrack1 = _vehicle addaction ["Track 1","radio_music\music.sqf",1];
playtrack2 = _vehicle addaction ["Track 2","radio_music\music.sqf",2];
playtrack3 = _vehicle addaction ["Track 3","radio_music\music.sqf",3];
playrandomsong = _vehicle addaction ["Play Random song","radio_music\music.sqf",4];
closePlayer = _vehicle addaction ["Close Music player","radio_music\music.sqf",5];

sleep 0.1;
_vehicle removeaction PlaySongs;

exit;
```

- 2) Creare il file Music.sqf nella cartella radio_music ed inserire la stringa

```
//Music.sqf
if (!isServer) exitWith {}; // To run this in MP

_vehicle = _this select 0;
_playtracks = _this select 3;

sleep 2;
switch (_playtracks) do {

case 1: //playing song 1
{
soundsource1 = "Land_FMradio_F" createVehicle [0,0,0];
hideObject soundsource1;
sleep 1;
soundsource1 attachTo [_vehicle,[0,0,0]];
sleep 1;
hint "playing song!.....";
sleep 3;
soundsource1 say3d "song1";
closesong1 = _vehicle addaction ["stop song 1","radio_music\stopsong.sqf",1];
};

case 2: //playing song 2
{
soundsource2 = "Land_FMradio_F" createVehicle [0,0,0];
hideObject soundsource2;
sleep 1;
soundsource2 attachTo [_vehicle,[0,0,0]];
sleep 1;
hint "playing song!.....";
sleep 3;
soundsource2 say3d "song2";
closesong2 = _vehicle addaction ["stop song 2","radio_music\stopsong.sqf",2];
};

case 3: //playing song 3
{
soundsource3 = "Land_FMradio_F" createVehicle [0,0,0];
hideObject soundsource3;
sleep 1;
soundsource3 attachTo [_vehicle,[0,0,0]];
sleep 1;
hint "playing song!.....";
sleep 3;
soundsource3 say3d "song3";
closesong3 = _vehicle addaction ["stop song 3","radio_music\stopsong.sqf",3];
};
};
```



```
};
case 4://playing random song.
{
soundsource4 = "Land_FMradio_F" createVehicle [0,0,0];
hideObject soundsource4;
sleep 1;
soundsource4 attachTo [_vehicle,[0,0,0]];
sleep 1;
hint "playing any song!....";
sleep 3;
_songs = ["song1","song2","song3"];
soundsource4 say3d (_songs select floor(random(count _songs)));

closerandomsong = _vehicle addaction ["stop random song","radio_music\stopsong.sqf",4];
};
case 5: //closing music player
{
_vehicle removeaction playtrack1;
_vehicle removeaction playtrack2;
_vehicle removeaction playtrack3;
_vehicle removeaction playrandomsong;
_vehicle removeaction closeplayer;hint "Music player switched off!";
PlaySongs = _vehicle addaction ["Start music
player","radio_music\musicplayer.sqf",true,1,false,true,"","player in _target"];
};
};
```

3) Creare il file stopsong.sqf nella cartella radio_music ed inserire la stringa

```
//Stopsong.sqf
if (!isServer) exitWith {}; // To run this in MP

_vehicle = _this select 0;
_stopsongs = _this select 3;

sleep 2;
switch (_stopsongs) do {

case 1: //stopping song 1.
{
hint "select other song!";
deletevehicle soundsource1;

sleep 0.1;
_vehicle removeaction closesong1;
};
case 2://stopping song 2.
{
hint "select other song!";
deletevehicle soundsource2;

sleep 0.1;
_vehicle removeaction closesong2;
};
case 3: //stopping song 3.
{
hint "select other song!";
deletevehicle soundsource3;

sleep 0.1;
_vehicle removeaction closesong3;
};
case 4://closing random song.
{

hint "Random song stopped!....";
deletevehicle soundsource4;

sleep 0.1;
_vehicle removeaction closerandomsong;
```



```
};  
};
```

- 4) Per utilizzare questo script in gioco in un veicolo:
- Aprire il lettore musicale dal menu a rotella.
 - Selezionare e riprodurre/interrompere i brani con l'addAction.
 - Chiudere il lettore musicale con il menu a rotella

RIFORMULAZIONE DEI FILE SOPRA CON UN SOLO FILE MUSICA

1) Init del veicolo

```
PlaySongs = this addAction ["Start music  
player", "radio_music\musicplayer.sqf", true, 1, false, true, "", "_this in _target"];
```

- 2) Nella cartella missione creare una cartella denominata "music". All'interno della cartella music, posizionare il brano musicale in formato .ogg e chiamalo come **track1** o qualsiasi altro nome coerente con lo script.
- 3) Creare il file description.ext nella cartella missione ed inserire la stringa

```
class CfgSounds  
{  
    sounds[]={};  
  
    class song1  
    {  
        name = "song1";  
        sound[] = {"music\track1.ogg", 0.5, 1};  
        titles[] = { };  
    };  
};
```

- 4) Creare la cartella "radio_music" nella cartella missione. Creare il file musicplayer.sqf nella cartella radio_music ed inserire la stringa

```
//Musicplayer.sqf  
if (!isServer) exitWith {}; // To run this in MP  
_vehicle = _this select 0;  
  
hintSilent "Music player switched on!";  
  
sleep 2;  
  
hintSilent "";  
  
sleep 2;  
playtrack1 = _vehicle addAction ["Track 1", "radio_music\music.sqf", 1];  
closePlayer = _vehicle addAction ["Close Music player", "radio_music\music.sqf", 2];  
  
sleep 0.1;  
_vehicle removeAction PlaySongs;  
  
exit;
```

- 5) Creare il file Music.sqf nella cartella radio_music ed inserire la stringa

```
//Music.sqf  
if (!isServer) exitWith {}; // To run this in MP  
  
_vehicle = _this select 0;  
_playtracks = _this select 3;  
  
sleep 2;  
switch (_playtracks) do {  
  
case 1: //playing song 1  
{  
    soundsource1 = "Land_FMradio_F" createVehicle [0,0,0];  
    hideObject soundsource1;  
    sleep 1;  
    soundsource1 attachTo [_vehicle, [0,0,0]];  
    sleep 1;  
    hint "playing song!.....";  
}
```



```
sleep 3;
soundsource1 say3d "song1";
closesong1 = _vehicle addaction ["stop song 1","radio_music\stopsong.sqf",1];

};

case 2: //closing music player
{
_vehicle removeaction playtrack1;
_vehicle removeaction closeplayer;hint "Music player switched off!";
PlaySongs = _vehicle addaction ["Start music
player","radio_music\musicplayer.sqf",true,1,false,true,"","player in _target"];

};
};
```

6) Creare il file stopsong.sqf nella cartella radio_music ed inserire la stringa

```
//Stopsong.sqf
if (!isServer) exitWith {}; // To run this in MP

_vehicle = _this select 0;
_stopsongs = _this select 3;

sleep 2;
switch (_stopsongs) do {

case 1: //stopping song 1.
{
hint "select other song!";
deletevehicle soundsource1;

sleep 0.1;
_vehicle removeaction closesong1;
};
};
```

10 ANIMAZIONI CIVILI

Per impostare delle animazioni sulle unità AI si può usare la funzione **BIS_fnc_ambientAnim**. Essa permette di impostare una serie di animazioni su una determinata unità.

Nota: se tale funzione è usata su dei soldati nemici essi non reagiranno ad un contatto, ma continueranno a far "girare" l'animazione. Se l'unità deve reagire al combattimento, usare invece la funzione analoga **BIS_fnc_ambientAnimCombat**, la quale però ha una serie limitata di animazioni rispetto a quella sopra.

Le animazioni possono essere cancellate usando `_unit call BIS_fnc_ambientAnim__terminate`.

Sintassi:

[unit, animationSet, equipmentLevel, snapTo, interpolate, attachToLogic] call BIS_fnc_ambientAnim

Parametri:

unit: **Object** - unit that will play the animations

animationSet: **String** - the animation set to play. Possible values:

some values are missing due to duplicate results, e.g "STAND" does the same as "STAND1"

- | | | |
|-------------------|----------------|------------------------|
| • STAND1 | • SIT_AT_TABLE | • KNEEL |
| • STAND2 | • SIT1 | • REPAIR_VEH_PRONE |
| • STAND_U1 | • SIT | • REPAIR_VEH_KNEEL |
| • STAND_U2 | • SIT3 | • REPAIR_VEH_STAND |
| • STAND_U3 | • SIT_U1 | • PRONE_INJURED_U1 |
| • WATCH | • SIT_U2 | • PRONE_INJURED_U2 |
| • WATCH2 | • SIT_U3 | • PRONE_INJURED |
| • GUARD | • SIT_HIGH1 | • KNEEL_TREAT |
| • LISTEN_BRIEFING | • SIT_HIGH | • KNEEL_TREAT2 |
| • LEAN_ON_TABLE | • SIT_LOW | • BRIEFING |
| • LEAN | • SIT_LOW_U | • BRIEFING_POINT_LEFT |
| | • SIT_SAD1 | • BRIEFING_POINT_RIGHT |



- SIT_SAD2
- BRIEFING_POINT_TABLE

equipmentLevel: [String](#) - the equipment level of the unit. Possible values:

- NONE
- LIGHT
- MEDIUM
- FULL
- ASIS
- RANDOM

snapTo: [Object](#) - (Optional, default [objNull](#)) the object where the unit will be snapped to

interpolate: [Boolean](#) - (Optional, default [false](#)) should try to interpolate into the ambient animation. Works only for some default stances

attachToLogic: [Boolean](#) - (Optional, default [true](#)) [true](#) to [attach](#) the unit to the created logic object, forcing it in one position

Esempio 1:

```
[player, "STAND1", "ASIS"] call BIS_fnc_ambientAnim;
```

Esempio 2:

```
// will search for a chair within 2 meters around the unit and will place the unit on it  
[_unit, "SIT", "NONE"] call BIS_fnc_ambientAnim;
```

10.1 CODICE DA INSERIRE (FUNZIONA)

Selezionare il civile e mettere questa istruzione nel suo campo init

```
[this, "STAND1", "ASIS"] call BIS_fnc_ambientAnim;
```

vedi sopra per Stand1 e Asis

altro esempio

```
[this, "KNEEL_TREAT", "ASIS"] call BIS_fnc_ambientAnim;
```

in questo caso l'unità si inginocchia

11 MODULO ZEUS – COME IMPOSTARE ZEUS ED INTEREAGIRE CON OGGETTI GIÀ PRESENTI

Passi per impostare Zeus ed editare, tramite Zeus in partita, gli oggetti già presenti e creati con l'editor.

Nota: se si richiama Zeus in partita, esso di default non è in grado di intervenire sugli oggetti già creati con l'editor. Per far sì che egli possa "gestirli" occorre inserire una stringa specifica nei moduli come segue.

- 1) Aggiungere il personaggio che farà zeus e dargli un nome "nomedellozeus"
- 2) Mettere giù il modulo zeus ed assegnarlo al personaggio tramite il nome "nomedellozeus" da inserire nella scheda "Proprietario del modulo". In alternativa si può sincronizzare il modulo direttamente al personaggio senza dargli il nome.

Alternativa: si può impostare il proprietario invece che con il nome con l'UID di Steam. Mettere l'UID direttamente nel campo proprietario. Con tale metodo lo zeus può essere solo quell'utente steam

Alternativa: si può impostare il proprietario anche mettendo nel campo la stringa #adminlogged. Abilita a Zeus chiunque si loggy come admin (tramite la chat inserendo #login password_admin)

Così il personaggio sarà Zeus premendo la "Y" ma non potrà interagire con gli oggetti introdotti da Editor, ma solo crearne di nuovi. Se si vuole interagire con tali oggetti occorre compiere questi altri due step.

Per più giocatori Zeus in una missione:

- a. aggiungi i giocatori Zeus come sopra;
 - b. ogni coppia di "unità giocabile/modulo master di gioco" sincronizzata deve avere un nome univoco
- 3) Mettere giù il modulo "aggiungi oggetti modificabili" (è nella sezione Zeus) (non serve sincronizzarlo con zeus)
 - 4) Sincronizzare il modulo "aggiungi oggetti modificabili" con tutti gli elementi presenti che si vogliono modificare. In **alternativa**, per evitare di selezionare uno ad uno gli oggetti che possono essere tanti, copiare tale istruzione nell'init di tale modulo



```
{_x addCuratorEditableObjects [allMissionObjects "ALL", true]; _x addCuratorAddons activatedAddons;}  
forEach allCurators;
```

Questa permette di aggiungere tutti gli oggetti presenti nello scenario editor automaticamente al modulo.

Alternativa (funziona) – NOTA: non funziona con EOS in §62

Si può creare anche un file init.sqf nella cartella della missione e mettervi direttamente la stringa. Questo senza mettere il modulo “aggiungi oggetti modificabili”.

```
if (isServer) then { { _y = _x; { if (side _x != sideLogic) then { _y addCuratorEditableObjects  
[[_x],true]; }; } } forEach allMissionObjects ""; } forEach allCurators; };
```

Alternativa (funziona)

Sorry the solution wasn't as simple as ive written before this edit at this moment zeus cant place triggers or gamelogics (oppure all'init del modulo “aggiungi oggetti modificabili”) you need to edit the mission with a gamemodule with this content:

```
if (isServer) then {  
  {  
    _x addCuratorEditableObjects [allUnits,true];  
    _x addCuratorEditableObjects [vehicles,true];  
  } forEach allCurators;  
};
```

NOTA: non funziona con EOS in §62

For a multi zeus setup you need to add this inside the gamelogic (oppure all'init del modulo “aggiungi oggetti modificabili”):

```
if (isServer) then {  
  [] spawn {  
    while {true} do {  
      {  
        _x addCuratorEditableObjects [allUnits, true];  
        _x addCuratorEditableObjects [vehicles, true];  
        sleep 5;  
      } forEach allCurators;  
    };  
  };  
};
```

NOTA: questo funziona anche con EOS in §62

È possibile anche metterlo direttamente nell'init senza passare per il modulo “aggiungi oggetti modificabili”. Il modulo zeus “master di gioco” va assegnato con il nome proprietario

NOTA BENE → Alternativa senza passare per il modulo “aggiungi oggetti modificabili” (funziona)

Non mettere giù il modulo “aggiungi oggetti modificabili” ma impostare l'opzione direttamente nelle impostazioni della ACE. Andare in Impostazioni/opzioni per gli addon e scorrere il menu a “ACE Zeus”, qui spuntare “aggiungi oggetti al curatore”

11.1 ALTRO MATERIALE

ADDING A VIRTUAL ZEUS TO A MISSION

1. Select the map you want to use
2. Systems > Modules > Game Master {ModuleCurator_F}
 - Set Owner to zCurator
 - Check Forced Interface box
3. Systems > Logic Entities > Virtual Entities > Zeus {VirtualCurator_F}
 - Set Variable Name to zCurator
4. Sync the two Modules together



ADDING A PLAYER ZEUS TO A MISSION

1. Select the map you want to use
2. Systems > Modules > Zeus > Game Master {ModuleCurator_F}
 - Set Owner to zCurator
 - Uncheck Forced Interface box
3. Add a Playable unit for the Player Zeus
 - Set Variable Name to zCurator
 - Check Player and/or Playable box
4. Sync the Modules and the Player Zeus unit together

ADDING LOGGED ADMIN ZEUS TO A MISSION

1. Select the map you want to use
2. Systems > Modules > Game Master {ModuleCurator_F}
 - Set Owner to #adminLogged
 - Check Forced Interface

ADDING MULTIPLE ZEUS TO A MISSION

For Multiple Player Zeus' in a mission:

1. Add Player Zeus to mission as above; each synced Playable Unit / Game Master Module pair must have a unique name

For Multiple Virtual Zeus' in a mission:

2. Add Virtual Zeus to mission as above; each synced Virtual Entity / Game Master Module pair must have a unique name

Types of Zeus can be mixed as required.

ADDING EDITOR PLACED OBJECTS TO ZEUS

1. Add either a Virtual or Player Zeus to your mission as above
2. Systems > Modules > Zeus > Add Editable Objects {ModuleCuratorAddEditableObjects}
3. Set Assigned Zeus to zCurator
4. Set Add Crew to Add
5. Sync the Add Editable Objects {ModuleCuratorAddEditableObjects} Module to the Game Master {ModuleCurator_F} Module
6. Sync any units you want to be editable by Zeus during the mission to the Add Editable Objects {ModuleCuratorAddEditableObjects} Module

FACTION-SPECIFIC & MULTIPLE ZEUS'

Replace the global Zeus module with a faction-specific module of your choosing.
All other aspects of the faction-specific Zeus' are the same as for the global Zeus.

12 RICHIAMARE IL SODDISFACIMENTO DI UN TRIGGER

Capita di dover richiamare l'attivazione di un trigger, come per esempio un caso di trigger1 che ha come condizione il soddisfacimento di un altro trigger2 esterno (chiamiamolo "nomedeltriggerchiamato"). In questo caso nelle condizione di soddisfacimento occorre che sia TRUE il trigger2. Questo si fa per l'appunto richiamando il trigger2 con una call da inserire nel campo condizione del trigger1 e che sarà di questa forma:

```
Call{triggerActivated nomedeltriggerchiamato}
```

Per attivare un trigger solo quando sono stati attivati più di un trigger mettere la seguente stringa

```
triggerActivated nomedeltrigger1 AND triggerActivated nomedeltrigger2;
```

oppure

```
Call{triggerActivated nomedeltrigger1} && Call{triggerActivated nomedeltrigger2}
```

Se si vuole che almeno uno di più trigger risulti attivato si fa con la stringa



```
triggerActivated nomeDeltrigger1 OR triggerActivated nomeDeltrigger2;
```

13 COME ATTIVARE UN TRIGGER DA PARTE DI UNA SPECIFICA UNITÀ (THISLIST COMMAND)

Per impostare l'attivazione di un trigger da parte di un'unità specifica e non da tutte quelle di una fazione che vi entrano, è sufficiente aggiungere, oltre alla spunta di Blufor (od altro) presente, la seguente stringa nella condizione, la quale usa la funzione dedicata "thisList":

```
(nomeDellunità in thisList);
```

il trigger si attiverà quando l'unità elencata entra nella zona di trigger

La stessa condizione può essere utilizzata anche per impostare un trigger che si attiva sull'unità elencata lasciando anche l'area specificata usando il carattere "!" prima delle parentesi del comando di scripting nel campo della condizione di trigger.

```
!(nomeDellunità in thisList);
```

il trigger si attiverà quando l'unità elencata esce dalla zona di trigger

Se si vogliono includere più unità specifiche per l'attivazione (per esempio 3) la stringa diverrà:

```
nomeDellunità1 in thislist && nomeDellunità2 in thislist && nomeDellunità3 in thislist
```

Un modo alternativo più breve per fare la stessa cosa è

```
{_x in thislist}count [nomeDellunità1, nomeDellunità2, nomeDellunità3] == 3
```

Per ogni [nomeDellunità1, nomeDellunità2, nomeDellunità3] verrà eseguito il codice {_x in thislist}, dove _x è a turno ciascuna unità.

Se {_x in thislist} è vero, verrà incrementato un contatore nascosto, qui chiediamo se il conteggio totale è uguale a 3. L'intera espressione restituisce VERO se ciascuna unità è in questo elenco.

Altre forma dell'istruzione è

```
this && nomeDellunità in thisList
```

"this" sono i parametri che si sono già impostati sul trigger stesso.

13.1 TRIGGER RADIO ALPHA, BRAVO...

Con i **trigger radio** "La condizione" viene ignorata con la funzione thisList. Ha senso in quanto sarebbe strano avere Radio Alpha presente.

Il modo più semplice per rendere utilizzabile un trigger radio solo da alcuni giocatori è questo:

1. hai un trigger radio ma inserisci NULL nel suo campo di testo. (In questo modo sarà "nascosto" per tutti i giocatori). È attivato ad es. radio alfa.
2. Fai uno script (o un trigger) che inizia con

```
if (player == X) then  
{  
1 setRadioMsg "The text you need";  
You put the rest of the script or the trigger activation commands here if there are other things you  
need.  
};
```

dove X è il nome del giocatore che si desidera poter utilizzare il trigger radio.

In questo modo l'opzione radio diventa "nascosta" solo per un determinato giocatore e puoi avere diversi comandi radio disponibili per giocatori diversi.

Se si utilizza il comando if come sopra descritto, è anche possibile assicurarsi che un determinato script venga eseguito su un solo computer. È utile nel multiplayer.

Sintassi del comando *setRadioMsg*

```
<radio> setRadioMsg <menu text>
```



- <radio> – numero da 1 a 10
 - Alpha (key 1)
 - Bravo (key 2)
 - Charlie (key 3)
 - Delta (key 4)
 - Echo (key 5)
 - Foxtrot (key 6)
 - Golf (key 7)
 - Hotel (key 8)
 - India (key 9)
 - Juliet (key 0)
- <menu text> – Il titolo mostrato nel menu radio. "NULL" rimuoverà la voce di menu dal menu, "" ripristinerà il nome radio predefinito.

Procedura alternativa - Come rendere un trigger radio utilizzabile solo da un'unità giocabile

Dai il nome ai tuoi soldati spawnabili come p1, p2, p3 e così via (anche solo il soldato che dovrebbe attivarsi dovrebbe bastare, es. p1).

1. Crea un trigger con Activation Radio Alpha, Once, Detected by Bluefor.

Nel campo Condizione digitare:

```
(player == p1)
```

Nel campo all'Attivazione digitare:

```
hint "Action Radio Alpha";
```

Questo dovrebbe consentire al soldato p1 di usare la radio (1,0,0,1) e mostrare il suggerimento evidenziato.

2. Per disabilitare il pulsante Radio per altri soldati, a cui non è consentito utilizzare il pulsante radio, posizionare un secondo trigger. Fallo attivo impostando il cerchio o il rettangolo abbastanza grande attorno ai punti di respawn del/i soldato/i nominato/i.

Attivazione: Bluefor (o altra fazione)

Stato: presente

Nel campo Condizione digitare:

```
this
```

Nel campo all'Attivazione digitare:

```
if (player != p1) then //if player is NOT p1
{
    1 setRadioMsg "NULL" //deactivate the button in menu 1,0
};
hint "Radio Disabled";
```

il // testo è qui per spiegare, ma puoi ometterlo in Arma.

Si avrà che la radio è disabilitata, anche per il giocatore è uguale a p1. Ma il pulsante di opzione per lui è ancora utilizzabile con 1,0,0,1 e quando attivato vedrai il messaggio "Action Radio Alpha".

14 DEAD BODIES - POSTURE

Per collocare unità in pose di cadaveri in mappa si possono usare delle animazioni. Queste si possono trovare nel menu Animazione (nella parte inferiore della finestra di debug).

Se si vuole averli in una posa casuale, inseriscire nell'init dell'unità la seguente stringa:

```
this setDamage 1;
```

La fisica Ragdoll si occupa di tutto il resto. Un'altra opzione è posizionarli 50 m dal suolo in modo che cadano a terra morti (assicurarsi di rimuovere tutti i paracadute che potrebbero avere).

Se si sta cercando una posa particolare, si può la funzione "switchMove" con alcune delle impostazioni predefinite.

1. Posizionare l'unità nell'editor



2. Caricare l'arsenale virtuale se si desidera che abbiano loadout personalizzati e incollare il loadout nel loro campo di inizializzazione (opzionale).
3. Impostare il campo init dell'unità con la stringa

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "[THE ANIMATION POSE]"};
```

* cambiare "[THE ANIMATION POSE]" con l'animazione che si vuole che abbia quel corpo. Mantenere le "" attorno al nome della posa.

"KIA_passenger_boat_holdleft" (a faccia in giù come se fosse stato ucciso mentre scappava.)

"KIA_gunner_standup01" (sdraiato sulla schiena semi rannicchiato che sembra essere colpito dalla parte anteriore.)

"KIA_gunner_static_low01" (sdraiato sulla schiena quasi in una posizione crocifissa. Colpito da davanti.)

"KIA_driver_boat01" (sdraiato in posizione fetale. Morto)

"KIA_passenger_injured_medevac_truck01" (morto sdraiato sulla schiena guardando alla loro destra. ? Morto durante le cure mediche?)

"KIA_passenger_injured_medevac_truck03" (morto sdraiato sul lato sinistro rivolto verso l'alto con il braccio sinistro sul petto)

"KIA_passenger_injured_medevac_truck02" (morto sdraiato sul lato sinistro rivolto verso l'alto con il braccio destro sul petto)

"Acts_SittingWounded_in" (james last moments leaned up against something minor head movements)

Esempi:

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "KIA_passenger_boat_holdleft"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove " KIA_gunner_standup01"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "KIA_gunner_static_low01"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "KIA_passenger_boat_holdleft"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "KIA_passenger_injured_medevac_truck03"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "KIA_passenger_injured_medevac_truck02"};
```

```
this setCaptive true;  
this setDamage 1;  
0 = this spawn {waitUntil {time > 0}; _this switchMove "Acts_SittingWounded_in"};
```

15 END MISSION

For example with Warload dead condition.

PreWork:

You need a "description" for full support of all features.

Create a description.ext file. Use your Windows Editor and open a new file.

copy and paste:



```
class CfgDebriefing { class End1 { title = "Mission Accomplished"; subtitle = "Well done"; description = "Warlord is dead "; pictureBackground = ""; picture = "b_inf"; pictureColor[] = {0.0,0.3,0.6,1}; };
```

1. Place your AI
name it eg: Warlord
2. Place Trigger1:
Activation: GameLogic
Condition: not alive Warlord
OnActivation:

```
task1done = true; publicVariable "task1done";
```

Explanation: After the warlord is dead task1done will be true or "fired".

3. Place Trigger2:
Activation: GameLogic
Condition: task1done
OnActivation:

```
task1 setTaskState "Succeeded"; ["TaskSucceeded",["","Warlord is dead"]] call BIS_fnc_showNotification;
```

Trigger: Timer: TimerType: Countdown

TimerValues: Min: 3 Mid: 3 Max: 3

Explanation: After 3 sec the task will be achieved and a hint "Warlord is dead" will popup.

4. Task: EndMission
Place Trigger3:
Activation: GameLogic
Condition: task1done
OnActivation:

```
["End1",true,true] call BIS_fnc_endMission;
```

Trigger: Timer: TimerType: Countdown

TimerValues: Min: 10 Mid: 3 Max: 10

Explanation: Same as Trigger2, only 10 sec later the BIS_fnc_endMission will be triggered. The Mission ends in a EndScreen

Alternativa:

Well, let's say your last task is to kill an officer, let's call him 'officer1'.

You obviously have a task set up for it and a trigger that lets the task be succesfull.

So the trigger condition would be: !alive officer1;

And here you could end the mission by typing in the triggers 'on Activation' field:

```
["Mission Complete",True] call BIS_fnc_endMission;
```

But, then you wouldn't get the last tasks state as success, so you make another trigger just like the previous and you put that code on its 'on Activation' field. And for this second trigger you put a delay for a few seconds, so that the tasks status has time to change to success.

If not succesfull use the code

```
["Mission Failed",false] call BIS_fnc_endMission;
```

15.1 END MISSION SCRIPT

È possibile definire finali diversi nel file description.ext

1. Creare il file nella cartella delle missioni e aggiungere qualcosa del genere:



Contenuto "description.ext"

```
class CfgDebriefing
{
class End1
{
    title = "The Hostages got rescued!";
    subtitle = "";
    description = "The Hostage-takers got annihilated!";
    pictureBackground = "";
    picture = "b_inf";
    pictureColor[] = {0.0,0.3,0.6,1};
};
class End2
{
    title = "The SWAT Team got wiped out!";
    subtitle = "";
    description = "No dinner at home tonight for these hostages.";
    pictureBackground = "";
    picture = "b_inf";
    pictureColor[] = {0.0,0.3,0.6,1};
};
class End3
{
    title = "Everyone died!";
    subtitle = "";
    description = "Suicide statistics going through the roof!";
    pictureBackground = "";
    picture = "b_inf";
    pictureColor[] = {0.0,0.3,0.6,1};
};
};
```

Qui ci sono definiti 3 possibili finali

Forma generale

config.cpp or description.ext:

```
class CfgDebriefing
{
    class End1
    {
        title = "Mission Completed";
        subtitle = "";
        description = "You successfully rescued our forces and neutralized the spotters,
despite being forced to retreat.";
        pictureBackground = "";
        picture = "b_inf";
        pictureColor[] = {0.0,0.3,0.6,1};
    };
};
```

Most of the params are shared both by the closing shot and by debriefing screen, see a table below:

- **title:** Ending main title. Default text (MISSION ACCOMPLISHED or MISSION FAILED, based on the ending type) is displayed when the param is missing.
- **subtitle:** Ending subtitle, further explaining the ending. Optional, no subtitle is displayed when the param is missing.
- **description:** Detailed description of the ending. Not available in closing shot. Optional, no description is displayed when the param is missing.
- **pictureBackground:** 2:1 picture displayed in the background. When undefined, [loading screen](#) is used instead (can be disabled by defining as empty string).
- **picture:** 1:1 icon, used mainly in multiplayer missions. Optional, no picture is displayed when the param is missing. Can be either link to a texture, or CfgMarkers class. In this case b_inf is the icon of a marker (see figure at §22)
- **pictureColor:** Picture color. Optional, white is used when the param is missing.

NOTA: per andare a capo di riga con il testo basta mettere il simbolo
 oppure

 per andare a capo e lasciare una riga vuota, es:



```
description = "testo riga 1.<br/>Testo riga 2.<br/><br/>Testo riga 2 dopo riga vuota";
```

Con questa stringa per esempio l'effetto è quello in figura

```
class CfgDebriefing
{
  class tutorial
  {
    title = "Tut - BIS_fnc_endMission";
    subtitle = "Tutorial by ScouseMouse";
    description = "--- Description off Mission ---";
    pictureBackground = "";
    picture = "b_inf";
    pictureColor[] = {0.0,0.3,0.6,1};
  };
};
```



Eventuale immagine in pictureBackground = "nome.jpg";

Logo in picture = "logo.jpg";

2. Nella missione si possono richiamare ogni finale individuale, con trigger o incarico completato, con questa semplice stringa:

```
"end1" call BIS_fnc_endMission;
```

Funzione "BIS_fnc_endMissionServer"

Description:

Ends the mission properly for all players in a multiplayer environment.

Syntax:

endType call BIS_fnc_endMissionServer

Parameters:

endType: [String](#) - class name of ending defined in [CfgDebriefing](#).

Can be **custom** or one of dynamically calculated ones:

- "EveryoneWon" - show "MISSION COMPLETED" to everyone
- "EveryoneLost" - show "MISSION FAILED" to everyone



- "SideScore" - show "YOUR SIDE WON" to side with largest score
- "GroupScore" - show "YOUR GROUP WON" to group with largest score
- "PlayerScore" - show "YOU WON" to player with largest score

Return Value:

[Boolean](#)

Example 1:

```
"SideScore" call BIS_fnc_endMissionServer;
```

15.2 ESEMPIO FUNZIONANTE

Nel file Description.ext inserire

```
// FINALI MISSIONE
// Per richiamare il finale in partita: fare ESC ed inserire la seguente stringa nella console:
// "Misscomp" call BIS_fnc_endMission; // se completata
// "Missfall" call BIS_fnc_endMission; // se fallita
class CfgDebriefing
{
class Misscomp
{
    title = "Missione completata!";
    subtitle = "La città è stata liberata con successo!";
    description = "Le forze dello stato islamico sono state ricacciate indietro!<br/>Un
ringraziamento a tutti per la partecipazione, spero vi siate divertiti.<br/>Enigx";
    pictureBackground = "logoEGXquadroNero.jpg";
//    picture = "b_inf"; // levare il commentato se serve
    pictureColor[] = {0.0,0.3,0.6,1};
};
class Missfall
{
    title = "Missione fallita!";
    subtitle = "La città non è stata liberata!";
    description = "Le forze dello stato islamico hanno preso il sopravvento!<br/>Un
ringraziamento a tutti per la partecipazione, spero vi siate divertiti.<br/>Enigx";
    pictureBackground = "logoEGXquadroNero.jpg";
//    picture = "b_inf"; // levare il commentato se serve
    pictureColor[] = {0.0,0.3,0.6,1};
};
};
```

Per richiamare il finale in partita: fare **ESC** in partita ed inserire la seguente stringa nella console debug:

```
"Misscomp" call BIS_fnc_endMission;
```

se completata, oppure

```
"Missfall" call BIS_fnc_endMission;
```

se fallita

Da provare per vedere se ha come effetto una schermata di chiusura con immagine

```
class RscPicture
{
    access=0;
    type=0;
    idc=-1;
    style=48;
    colorBackground[]={0,0,0,0};
    colorText[]={1,1,1,1};
    font="TahomaB";
    sizeEx=0;
    lineSpacing=0;
    text="";
    shadow = 0;
};
```



```
class Screen_chiusura
{
    idd=-1;
    movingEnable=1;
    duration=6;
    fadein=2;
    fadeout=0;
    name="Screen_chiusura";
    Controls[]={ "Logo" };
    class Logo: RscPicture
    {
        idc = 9999;
        text="image\file_image.paa"; //mettere l'immagine desiderata
        style = 0x30 + 0x800;
        x = safezoneX;
        y = safezoneY;
        w = safezoneW;
        h = safezoneH;
        colorBackground[]={1,1,1,1};
        colorText[]={1,1,1,1};
    };
};

// FINALI MISSIONE
// Per richiamare il finale in partita: fare ESC ed inserire la seguente stringa nella console:
// "Misscomp" call BIS_fnc_endMission; // se completata
// "Missfall" call BIS_fnc_endMission; // se fallita
class CfgDebriefing
{
    class Misscomp
    {
        2 cutRsc ["Screen_chiusura", "PLAIN"];
    };
};
```

16 VISUALIZZARE UN'IMMAGINE SU SCHERMO

16.1 IMMAGINE A SCHERMO

Un modo per far visualizzare un'immagine a schermo è usare la funzione `bis_fnc_dynamicText`.

Un esempio può essere il seguente:

```
["<img image='PathToImageHere' />",0,0.5,100,5,0,9459] spawn bis_fnc_dynamicText;
```

Questa viene visualizzata in posizione [0,0.5], dura 100 sec, viene rimossa con effetto fade in un tempo di 5 sec, l'immagine è ferma, posizionata su un layer 9459

Per disattivarla prima dei 100 sec si può dare l'sitruzione:

```
9459 cutText ["" ,"PLAIN",0]
```

Questo un esempio usato in una missione (funziona)

```
["<img size='15' image='images\doc_secret.jpg' />",0,0,12,0,0,1000] spawn bis_fnc_dynamicText;
```

siccome l'immagine riprodotta potrebbe venire piccola, qui si è ingrandita. La funzione "agisce" su un testo. Il testo può essere "trattato" anche come un'immagine tramite la `<img image=...` Qui si è ingrandita usando un font di size 15

Nota: occorre testarne la validità in MP. Dovrebbe funzionare. Ad ogni modo si può sostituire con questa:

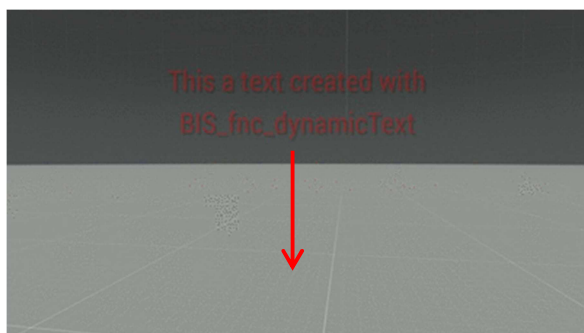
```
["<img size='15' image='images\doc_secret.jpg' />",0,0,12,0,0,1000] remoteExecCall  
['bis_fnc_dynamicText',0];
```

Questa usa la funzione `remoteExecCall` che fa sì che la stringa venga eseguita su server.

Funzione "BIS_fnc_dynamicText"

Description:

Shows dynamic opening credits or any type of text.

**Syntax:**

[text, x, y, duration, fadeInTime, deltaY, rscLayer] spawn BIS_fnc_dynamicText

Parameters:

text: [String](#) - text to display
x: [Number](#) - (Optional, default -1) X coordinates
y: [Number](#) - (Optional, default -1) Y coordinates
duration: [Number](#) - (Optional, default 4) display duration
fadeInTime: [Number](#) - (Optional, default 1) fade-in time
deltaY: [Number](#) - (Optional, default 0) Y position delta:

- = 0: Text will not move
- > 0: Text will move down
- < 0: Text will move up

duration and the absolute *deltaY* value will influence the movement speed.

rscLayer: [Number](#) - (Optional, default) resource layer

Return Value:

[Nothing](#)

Example 1:

```
["<t color='#ff0000' size = '.8'>Warning!<br />Stop doing what you are doing</t>", -1, -1, 4, 1, 0, 789]  
spawn BIS_fnc_dynamicText;
```

16.2 IMMAGINE A SCHERMO INTERNO

Di seguito le stringhe usate per far comparire un'immagine a schermo interno

1. Nel description.ext aggiungere la stringa

```
class RscTitles  
{  
#include "script\imm_schermo_interno.hpp"  
};
```

2. Creare la cartella \script all'interno della cartella missione
3. Creare il file "imm_schermo_interno.hpp" all'interno della \script e mettervi le seguenti stringhe

```
// Control types  
#define CT_STATIC 0  
  
// Static styles  
  
// #define ST_LEFT 0x00  
  
#define ST_LEFT 0  
#define ST_RIGHT 1  
#define ST_CENTER 2  
#define ST_UP 3  
#define ST_DOWN 4  
#define ST_VCENTER 5  
  
#define ST_SINGLE 0  
#define ST_MULTI 16
```



```
#define ST_PICTURE 48
#define ST_FRAME 64

#define ST_HUD_BACKGROUND 128
#define ST_TILE_PICTURE 144
#define ST_WITH_RECT 160
#define ST_LINE 176

#define ST_SHADOW 256
#define ST_NO_RECT 512

#define FontM "Bitstream"
#define ST_LEFT 0
#define ST_RIGHT 1
#define ST_CENTER 2
#define CT_STATIC 0

class RscText_Screen_Comp
{
    access = 0;
    idc = -1;
    type = CT_STATIC;
    style = ST_PICTURE; //style = ST_LEFT;
    colorBackground[] = {0,0,0,1};
    colorText[] = {1,1,1,1};
    //w = 0.1; h = 0.05;
    //x and y are not part of a global class since each rsctext will be positioned 'somewhere'
    font = "TahomaB"; //font = "Bitstream";
    sizeEx = 0; //sizeEx = 0.04;
    lineSpacing = 0;
    text = "";
    fixedWidth = 0;
    shadow = 0;
};

class schermo_intero
{
    idd=-1;
    movingEnable=0;
    duration= 7; //7
    fadein=0;
    name="schermo_intero";
    controlsBackground[] = {};
    controls[]={Bild};
        class Bild : RscText_Screen_Comp
        {
            idc = 1200; //1200
            text="Finale_missione\EGX_MissComp.jpg"; //or.paa
            //x = 0.1 * safezoneW + safezoneX;
            //y = 0.15 * safezoneH + safezoneY;
            //w = 0.8 * safezoneW;
            //h = 0.7 * safezoneH;

            x = 0 * safezoneW + safezoneX;
            y = 0 * safezoneH + safezoneY;
            w = 1 * safezoneW;
            h = 1 * safezoneH;
        };
};
```

4. Sempre all'interno della \script creare un file "imm_schermo_intero.sqf"

```
100012 cutRsc ["schermo_intero", "PLAIN",3]; // visualizzazione dell'immagine
```

Il primo numero 100012 è il livello di un layer. Usare quello che si vuole

5. Richiamare l'immagine, in un trigger od altro, tramite l'esecuzione del'sqf con la stringa

```
execVM "script\schermo_intero.sqf";
```



Funzione *cutRsc*

Display a resource defined in RscTitles of the mission's [Description.ext](#), the campaign's description.ext or the global config.

Syntax:

cutRsc [class, type, speed, showInMap]

Parameters:

[class, type, speed, showInMap]: [Array](#)

class: [String](#) - the class name of the resource. ("Default" will remove the current resource.)

type: [String](#) one of "PLAIN", "PLAIN DOWN", "BLACK", "BLACK FADED", "BLACK OUT", "BLACK IN", "WHITE OUT" and "WHITE IN". See [Title Effect Types](#) for more information about these values.

speed (Optional): [Number](#) - Time in seconds to fade in resource. Default: -1

showInMap (Optional): [Boolean](#) - [false](#) to hide the text when the map is opened. Default: [true](#)

Return Value:

[Nothing](#)

Altra fonte trovata (testate ma sembra non funzionare)

Question: I have a .jpg image I want to flash on the screen in a ARMA3 mission. What's the easiest way to do this?

Answer: In Description.ext for a FULL screen image I have done this

```
class RscPicture
{
    access=0;
    type=0;
    idc=-1;
    style=48;
    colorBackground[]={0,0,0,0};
    colorText[]={1,1,1,1};
    font="TahomaB";
    sizeEx=0;
    lineSpacing=0;
    text="";
    shadow = 0;
};

class GoA_Logo
{
    idd=-1;
    movingEnable=1;
    duration=6;
    fadein=2;
    fadeout=0;
    name="GoA_Logo";
    Controls[]={ "Logo" };
    class Logo: RscPicture
    {
        idc = 9999;
        text="image\file_image.paa";
        style = 0x30 + 0x800;
        x = safezoneX;
        y = safezoneY;
        w = safezoneW;
        h = safezoneH;
        colorBackground[]={1,1,1,1};
        colorText[]={1,1,1,1};
    };
};
```

and then a cut command to display it

```
2 cutRsc ["GoA_Logo", "PLAIN"];
```

The 2 can be any number, it's a layer, higher numbers stack.



16.3 CREARE UNA SCHERMATA FINE MISSIONE A TUTTO SCHERMO CON SUONO PERSONALIZZATO

Di seguito come far terminare una missione facendo comparire un'immagine a tutto schermo con un suono personalizzato.

1. Nel description.ext aggiungere

```
/* ----- SCHERMATA FINALE A TUTTO SCHERMO
Per richiamare il finale in partita si dovranno usare le seguenti stringhe:
    execVM "Finale_missione\EGX_MissComp.sqf"; // Per Missione Completata
    execVM "Finale_missione\EGX_MissFall.sqf"; // Per Missione Fallita
Si può fare in due modi:
- fare ESC in partita, digitare una delle due stringhe nella console e lanciarla su server
- creare dei trigger nell'editor e all'attivazione inserire la stringa appropriata
*/

class RscTitles
{
#include "Finale_missione\EGX_MissComp.hpp"
#include "Finale_missione\EGX_MissFall.hpp"
};

// ----- DEFINIZIONE DEL SUONO DI OUTRO DELLA SCHERMATA FINALE
// NOTA: se già è definita nel description.ext un'altra istruzione "class CfgSounds", aggiungere la
// seguente "class Finale" tra le altre già definite in questa istruzione CfgSounds

class CfgSounds {
// sounds[] = {};
// ... eventuali altre classes già definite
// ....
class Finale // This (Finale) is the classname you will use to refer to the sound in your code
{
name = "Finale";
sound[] = {\Finale_missione\Finale.ogg,5,1,18,1,1,1,0}; // This is the path to the audio file in your
mission folder.
titles[] = {};
empty[] = {"",0,0,0,0,0,0,0};
};
// ....
};

// ----- SCHERMATE DI DEBRIEFING FINALI -----
class CfgDebriefing
{
class Misscomp // definisci una classe di debriefing chiamata "Misscomp" con i parametri desiderati
(missione completata in questo caso)
{
title = "Missione completata!";
subtitle = "Avete svelato le manovre russe a Pecher";
description = "Inserire descrizione"; //descrizione completa nella finestra di debriefing
pictureBackground = "Finale_missione\logoEGXquadroNero.jpg"; // Immagine di sfondo della schermata di
debriefing
//picture = "b_inf"; // levare il commentato se serve far vedere l'icona selezionata
pictureColor[] = {0.0,0.3,0.6,1};
};
class Missfall // definisci una classe di debriefing chiamata "Missfall" con i parametri desiderati
(missione fallita in questo caso)
{
title = "Missione fallita!"; // titolo principale che compare in grande
subtitle = "Le manovre dei russi a Pecher restaranno oscure..."; // Inserire eventuale sottotitolo
description = "Inserire descrizione; //descrizione completa nella finestra di debriefing
pictureBackground = "Finale_missione\logoEGXquadroNero.jpg"; // Immagine di sfondo della schermata di
debriefing
// picture = "b_inf"; // levare il commentato se serve far vedere l'icona selezionata
pictureColor[] = {0.0,0.3,0.6,1};
};
};
```

2. Creare la cartella \Finale_missione all'interno della cartella missione
3. Creare l'immagini con nome "EGX_MissComp.jpg" (o .paa) e "EGX_MissFall.jpg" (o .paa) rispettivamente per missione completata e missione fallita. Metterli nella cartella \Finale_missione



4. Creare un file audio personalizzato per l'outro con nome "Finale.ogg". Metterlo nella cartella \Finale_missione
5. Creare all'interno della \Finale_missione due file .hpp, rispettivamente per missione completata ("EGX_MissComp.hpp") e missione fallita ("EGX_MissFall.hpp"). Ogni file sarà del tipo:

```
// Control types
#define CT_STATIC 0

// Static styles

// #define ST_LEFT          0x00

#define ST_LEFT      0
#define ST_RIGHT     1
#define ST_CENTER    2
#define ST_UP        3
#define ST_DOWN      4
#define ST_VCENTER   5

#define ST_SINGLE    0
#define ST_MULTI     16
#define ST_PICTURE   48
#define ST_FRAME     64

#define ST_HUD_BACKGROUND 128
#define ST_TILE_PICTURE 144
#define ST_WITH_RECT 160
#define ST_LINE 176

#define ST_SHADOW 256
#define ST_NO_RECT 512

#define FontM "Bitstream"
#define ST_LEFT      0
#define ST_RIGHT     1
#define ST_CENTER    2
#define CT_STATIC    0

class RscText_Screen_Comp
{
    access = 0;
    idc = -1;
    type = CT_STATIC;
    style = ST_PICTURE; //style = ST_LEFT;
    colorBackground[] = {0,0,0,1};
    colorText[] = {1,1,1,1};
    //w = 0.1; h = 0.05;
    //x and y are not part of a global class since each rsctext will be positioned 'somewhere'
    font = "TahomaB"; //font = "Bitstream";
    sizeEx = 0; //sizeEx = 0.04;
    lineSpacing = 0;
    text = "";
    fixedWidth = 0;
    shadow = 0;
};

class EGX_MissComp // N.B.: sostituire con EGX_MissFall per missione fallita
{
    idd=-1;
    movingEnable=0;
    duration= 7; //7
    fadein=0;
    name="EGX_MissComp"; // N.B.: sostituire con EGX_MissFall per missione fallita
    controlsBackground[] = {};
    controls[]={Bild};
        class Bild : RscText_Screen_Comp
        {
            {
                idc = 1200; //1200
                text="Finale_missione\EGX_MissComp.jpg"; //or.paa N.B.: sostituire con EGX_MissFall per
missione fallita
                //x = 0.1 * safezoneW + safezoneX;
                //y = 0.15 * safezoneH + safezoneY;
                //w = 0.8 * safezoneW;
                //h = 0.7 * safezoneH;
            }
        }
    };
};
```



```
x = 0 * safezoneW + safezoneX;  
y = 0 * safezoneH + safezoneY;  
w = 1 * safezoneW;  
h = 1 * safezoneH;  
  
};  
};
```

6. Sempre all'interno della \Finale_missione creare due file .sqf, rispettivamente per missione completata ("EGX_MissComp.sqf") e missione fallita ("EGX_MissFall.sqf"). Ogni file sarà del tipo:

```
playsound "Finale"; // chiamo il file audio  
sleep 5; // tempo in secondi prima che compaia l'immagine  
  
100012 cutRsc ["EGX_MissComp", "PLAIN",3]; // visualizzazione dell'immagine N.B.: sostituire con EGX_MissFall per missione fallita  
  
sleep 8; // tempo in secondi prima che compaia l'immagine  
100013 cutText [",", "BLACK OUT"]; // scomparsa dell'immagine  
  
["Misscomp",true,1,false,true] call BIS_fnc_endMission; // termina missione e chiama la schermata di debriefing pre-imposata N.B.: sostituire con Missfall per missione fallita  
//[ "Misscomp",true,1,false,true] remoteExecCall ['BIS_fnc_endMission',0]; // termina missione per tutti i clients e chiama la schermata di debriefing pre-imposata N.B.: sostituire con Missfall per missione fallita  
  
// Nota: i secondi di sleep vanno impostati in modo tale che la somma sia il tempo in secondi del file audio
```

All'esecuzione l'effetto è il seguente: riproduce il file audio, dopo 5 sec compare l'immagine a schermo intero, aspetta 8 sec e sovrappone all'immagine uno schermo nero (layer 100013>100012), contemporaneamente termina la partita e richiama la schermata di debriefing.

Nella ["Misscomp",true,1,false,true] call BIS_fnc_endMission; le istruzioni rimuovono l'effetto bianco e nero dell'end mission tipico di arma

7. Per richiamare il finale in partita si dovranno usare le seguenti stringhe:

```
execVM "Finale_missione\EGX_MissComp.sqf"; // Per Missione Completata  
execVM "Finale_missione\EGX_MissFall.sqf"; // Per Missione Fallita
```

Si può fare in due modi:

- fare ESC in partita, digitare una delle due stringhe nella console e lanciarla su server
- creare dei trigger nell'editor e all'attivazione inserire la stringa appropriata

17 HOW TO ADD AN IMAGE TO A HINT

images in .paa format in specific folder "image" into mission folder
at activation use code

```
hint parseText "<img image='image\test.paa' /> ";
```

Make sure your image's resolutions are a power of 2 (64x64, 1024x512 etc). Otherwise the .paa conversion will cause weird issues, like the one you're having.

Also you might want to use the size property to make it bigger.

```
<img size='2' image='image\test.paa' />
```

Esempio di testo compost con immagini:

```
title = "<t color='#ff0000' size='1.2' shadow='1' shadowColor='#000000' align='center'>SETTING RADIO CHANNEL</t>";  
_img = "<img size='12' image='9GU_gameSystem\img\lungoRaggio.jpg' align='center' />";  
_text = "<t color='#00ff00' size='1.2' shadow='1' shadowColor='#000000' align='center'><br/>LR CHANNEL</t>";
```




```
_text2 = "<br/>CH 1 = 31<br/>CH 2 = 32<br/>CH 3 = 33<br/>CH 4 = 44<br/>CH 5 = 35<br/>CH 6 = 36<br/>CH 7 = 37<br/>CH 8 = 38<br/>CH 9 = 39";  
hint parseText (_title + _img + _text + _text2);
```

18 AGGIUNGERE UN'INSEGNA PERSONALIZZATA SULL'UNIFORME DELL'UNITÀ

Per aggiungere un'insegna personalizzata all'uniforme, che poi può essere selezionata dal player, occorre aggiungere la stringa nel "description.ext" o nel "config.cpp"

```
class CfgUnitInsignia  
{  
    class Logo_uniforme  
    {  
        displayName = "Logo unità speciale"; // Name displayed in Arsenal  
        author = "Inserire l'autore"; // Author displayed in Arsenal  
        texture = "imagine_logo.paa"; // Image path  
        textureVehicle = ""; // Does nothing currently, reserved for future use  
    };  
};
```

La trama del logo dovrebbe idealmente essere un file paa di risoluzione 128x128 px.

Nota trovata (da testare): If it's possible to add a `cfgUnitInsignia` class in a `description.ext`, please note that the path of the texture differs for server and client in MP environment. `Config.cpp` (addon) is more reliable. `Description.ext`, in this case, should be used for SP mission only.

This function calls the `setObjectTextureGlobal`, which is a broken command in MP (see feedback tracker) August 2016.

Per assegnarlo direttamente all'unità, tramite per esempio i loadout preimpostati, è sufficiente inserire la seguente stringa nel file `.sqf` del loadout dell'unità (o nel suo `init`):

```
[this,"Logo_uniforme"] call BIS_fnc_setUnitInsignia;
```

NOTE: Accessing a Virtual Arsenal will erase the insignia from your character and it cannot be applied through the Virtual Arsenal.

Per ovviare a questo inconveniente si potrebbe creare una `addAction` sull'Arsenal con il comando di assegnazione della patch.

- Quindi mettere la seguente stringa nell'init della cassa

```
this addAction ["<t color='#FFFF00' size='1.3'>NOTA: se hai aperto l'Arsenal ricordati di attaccare la patch una volta uscito!!</t>",""];  
this addAction ["<t color='#FFFF00' size='1.3'>-----</t>",""];  
this addAction ["<t color='#00FF00' size='1.3'>ARSENALE VIRTUALE</t>",{["Open",true] call BIS_fnc_arsenal; }];  
this addAction ["<t color='#00FF00' size='1.3'>Attacca la patch sull'uniforme</t>","attacca_patch.sqf"];
```

- Creare un file "attacca_patch.sqf" nella cartella missione e mettervi le stringhe

```
fnc_removeUnitInsignia =  
{  
    _unit = _this select 0;  
    _index = -1;  
    {  
        if (_x == "insignia") exitwith {_index = _foreachindex;};  
    } foreach getarray (configfile >> "CfgVehicles" >> gettext ( configfile >> "CfgWeapons" >> uniform _unit >> "ItemInfo" >> "uniformClass") >> "hiddenSelections");  
    _unit setObjectTextureGlobal [_index, ""];  
};  
[this] call fnc_removeUnitInsignia;  
[this,"Logo_uniforme"] call BIS_fnc_setUnitInsignia;
```

In questo modo una volta che il player accede all'Arsenal e la patch viene rimossa, egli ha la possibilità di rimetterla tramite comando dedicato con rotella del mouse.



Funzione *BIS_fnc_setUnitInsignia*

Description:

Sets unit insignia (e.g., shoulder insignia on soldiers). See [Arma 3 Unit Insignia](#) for more details. This function is global and should work from any client. **Do not remote execute**

Syntax:

```
[target , className] call BIS_fnc_setUnitInsignia
```

Parameters:

target: [Object](#) - unit to which the insignia is going to be attached

class: [String](#) - CfgUnitInsignia class name to use. Use an empty string to remove the current insignia.

Return Value:

[Boolean](#) - Returns [true](#) if successful.

Example 1:

```
Place insignia: [player, "111thID"] call BIS_fnc_setUnitInsignia;
```

Example 2:

```
Remove insignia: [player, ""] call BIS_fnc_setUnitInsignia;
```

There are two ways of removing an insignia when placed.

1. The first way would be to run this function on the unit:

```
fnc_removeUnitInsignia =
{
    _unit = _this select 0;
    _index = -1;
    {
        if (_x == "insignia") exitwith {_index = _foreachindex;};
    } foreach getarray (configfile >> "CfgVehicles" >> gettext ( configfile >> "CfgWeapons" >>
uniform _unit >> "ItemInfo" >> "uniformClass") >> "hiddenSelections");
    _unit setObjectTextureGlobal [_index, ""];
};
```

E richiamare il remove

```
[player] call fnc_removeUnitInsignia;
```

2. The second way would be to create an empty `cfgUnitInsignia` class in the `description.ext`:

```
class CfgUnitInsignia
{
    class empty
    {
        author = "";
        displayName = "";
        texture = "";
    };
};
```

And then run it the normal way:

```
[player, "empty"] call BIS_fnc_setUnitInsignia
```

Per ottenere un'insegna di un'unità assegnata con la `BIS_fnc_setUnitInsignia` si può usare la `BIS_fnc_getUnitInsignia`.

```
_insignia = player call BIS_fnc_getUnitInsignia;
```

questa assegna alla variabile `_insignia` il logo assegnato al player con la `BIS_fnc_setUnitInsignia`

Altro metodo trovato (che dovrebbe rimediare anche alla cancellazione della patch in uscita dall'Arsenal) (DA TESTARE) (1)

1. Definire sempre la class `Logo_uniforme` nel `description.ext` come sopra
2. Creare un file `init.sqf` (o `initPlayerLocal.sqf`) e mettervi le seguenti stringhe



```
fnc_permanentInsignia = {
    private ["_unit","_class","_texture"];
    _unit = [_this,0,objnull,[objnull]] call bis_fnc_param;
    _class = [_this,1,"",[ ]] call bis_fnc_param;

    _texture = if (_class != "") then {gettext (configfile >> "CfgUnitInsignia" >> _class >>
"texture")} else {"#(argb,8,8,3)color(0,0,0,0)"};
    if (_texture == "") exitwith [{"'%1' not found in CfgUnitInsignia",_class] call
bis_fnc_error; false};

    private ["_index"];
    _index = -1;
    {
        if (_x == "insignia") exitwith {_index = _foreachindex;};
    } foreach getarray (configfile >> "cfgVehicles" >> typeof _unit >> "hiddenSelections");

    if (_index < 0) then {
        ["'insignia' hidden selection not found on object of type '%1'",typeof _unit] call
bis_fnc_error;
        false
    } else {
        _unit setObjectTextureGlobal [_index,_texture];
        _unit setVariable ["unitInsignia",_texture];
        _EH = _unit getVariable ["uI_res_EH", -1];
        _EH1 = _unit getVariable ["uI_inv_EH", -1];
        if (_EH < 0) then {
            _EH = _unit addEventHandler ["Respawn", {
                T_fnc_setUnitInsignia;
                [_unit,_unit getVariable "unitInsignia"] call
                T_fnc_setUnitInsignia;
            }];
            _unit setVariable ["uI_res_EH",_EH];
        };
        if (_EH1 < 0) then {
            _EH = _unit addEventHandler ["InventoryClosed", {
                T_fnc_setUnitInsignia;
                [_unit,_unit getVariable "unitInsignia"] call
                T_fnc_setUnitInsignia;
            }];
            _unit setVariable ["uI_inv_EH",_EH];
        };
    };
};
```

È una versione modificata della funzione BIS_fnc_setUnitInsignia.

3. Quindi applicare la patch all'unità inserendo quanto segue nel suo init

```
[[this, "Logo_uniforme"], "fnc_permanentInsignia", nil, true, true] call BIS_fnc_MP;
```

La patch verrà comunque rimossa quando si esce dall'arsenale ma viene riapplicata ogni volta che si chiude l'inventario. Pertanto rimane anche quando si cambia divisa e verrà riapplicata al respawn.

Altro metodo trovato (2)

Description.ext

```
class CfgUnitInsignia
{
    class Logo_uniforme
    {
        displayName = "Logo unità speciale"; // Name displayed in Arsenal
        author = "Inserire l'autore"; // Author displayed in Arsenal
        texture = "imagine_logo.paa"; // Image path
        textureVehicle = ""; // Does nothing currently, reserved for future use
    };
};
```

initPlayerLocal.sqf:

```
_playerUIDs = ["1234567","2345678"];
if ((getPlayerUID player) in _playerUIDs) then
{
    [player,"Logo_uniforme"] call bis_fnc_setUnitInsignia;
};
```

Altro metodo trovato (3)



Qui vengono assegnate patches diverse a seconda del nome definito per l'unità nello slot (S01, S02,...)

Description.ext

```
class CfgUnitInsignia
{
    class Logo_uniforme1
    {
        displayName = "Logo unità special 1"; // Name displayed in Arsenal
        author = "Inserire l'autore"; // Author displayed in Arsenal
        texture = "imagine_logol.paa"; // Image path
        textureVehicle = ""; // Does nothing currently, reserved for future use
    };
    class Logo_uniforme2
    {
        displayName = "Logo unità special 2"; // Name displayed in Arsenal
        author = "Inserire l'autore"; // Author displayed in Arsenal
        texture = "imagine_logo2.paa"; // Image path
        textureVehicle = ""; // Does nothing currently, reserved for future use
    };
};
```

initPlayerLocal.sqf

```
params ["_player", "_jip"];

_insignia = switch (vehicleVarName _player) do
{
    case "S01": {"Logo_uniforme1"};
    case "S02": {"Logo_uniforme2"};
};

[_player, _insignia] call bis_fnc_setUnitInsignia;
```

[Altro metodo trovato \(4\) https://forums.bohemia.net/forums/topic/176500-taking-a-uniform-from-virtual-arsenal-removes-insignia/?tab=comments#comment-2769901](https://forums.bohemia.net/forums/topic/176500-taking-a-uniform-from-virtual-arsenal-removes-insignia/?tab=comments#comment-2769901)

Description.ext

```
class CfgUnitInsignia
{
    class Logo_uniforme
    {
        displayName = "Logo unità speciale"; // Name displayed in Arsenal
        author = "Inserire l'autore"; // Author displayed in Arsenal
        texture = "imagine_logo.paa"; // Image path
        textureVehicle = ""; // Does nothing currently, reserved for future use
    };
};
```

initPlayerLocal.sqf:

```
player setVariable [ "VAcatch", [ "VAcatch", "onEachFrame", {
    if ( !( isNil { _this getVariable "VAcatch" } ) ) && { !( isNull ( uiNamespace getVariable [
"BIS_fnc_arsenal_cam", objNull ] ) ) ) } ) then {
        _this setVariable [ "VAcatch", nil ];
        _thread = _this spawn {
            _uniform = uniform _this;
            waitUntil { isNull ( uiNamespace getVariable [ "BIS_fnc_arsenal_cam", objNull ] ) };

            // Arsenal closed
            // Set unit insignia
            [player, "Test"] call BIS_fnc_setUnitInsignia; // Error line
            _this setVariable [ "VAcatch", "VAcatch" ];
        };
    }
}, player ] call BIS_fnc_addStackedEventHandler ];
```

Init dell'ammobox

```
[ "AmmoboxInit", [ _this, false, { true } ] ] call BIS_fnc_arsenal;
```



19 ACTIVATE A TRIGGER BY LANDING THE HELICOPTER IN A CERTAIN AREA/WITHIN THE TRIGGER ZONE

Informazioni varie dal web:

Trigger condition:

```
{_x isKindOf "Helicopter" && isTouchingGround _x} count thisList > 0
```

```
{"Helicopter" && isTouchingGround _heli} count thisList > 0
```

Dove in "helicopter" mettere il nome assegnato all'elicottero

If you don't want to have the heli named for the trigger to work (can be helpful for copy&paste, etc), you can also sync the trigger to the heli and use this condition:

```
{isTouchingGround _x} count thisList > 0;
```

maybe something like this:

Trigger Attributes:

Type: None

Activation: BLUFOR // or the side your heli is.

Activation Condition: Present

Condition:

```
this && isTouchingGround heli
```

On Activation: your code

should run the code when the heli is touching the ground.

if you want you can add a timer too so maybe wait like 5 sec after heli touched ground then exe code.

Maybe this helps you, here is my helicopter extraction script i use in my missions:

```
if (!isServer) exitwith {};
private
["_group", "_heli", "_side", "_spawnPos", "_goToPos", "_AirPos", "_safePos", "_range", "_maxGrad", "_landPos", "_m", "_vehicle", "_veh", "_vehgrp", "_x", "_wp1", "_smokePos", "_smoke", "_wp2", "_units", "_strobe", "_light"];
//_group = _this select 0;
_heli = _this select 0;
_side = _this select 1;
_spawnPos = _this select 2;
_goToPos = _this select 3;
waituntil {!isnil "bis_fnc_init"};
_AirPos = [(getmarkerpos _spawnPos select 0), (getmarkerpos _spawnPos select 1), 100];
//find safe landing position
_safePos = [];
_range = 35;
_maxGrad = 0.1;
while (((count _safePos) == 0)) do {
    _safePos = [
        ["position", getMarkerPos "lz"],
        ["number", 1],
        ["objDistance", 9],
        ["range", [0, _range]],
        ["maxGradient", _maxGrad]
    ] call RUBE_randomCirclePositions;
    _range = _range * 1.25;
    _maxGrad = _maxGrad + 0.01;
};
_landPos = createVehicle ["Land_HelipadEmpty_F", (_safePos select 0), [], 0, "NONE"];
/*_m = createMarker ["exma", position _landPos];
_m setMarkerShape "Icon";
_m setMarkerType "mil_pickup";
_m setmarkerColor "ColorWEST";
_m setMarkerText "EXTRACTION";*/
//hint format ["%1", _safePos];
[["RadioAmbient5"], "sh_sound_fnc", nil, false] call BIS_fnc_MP;
_vehicle = [_AirPos, random 360, _heli, _side] call bis_fnc_spawnvehicle;
_veh = _vehicle select 0;
```



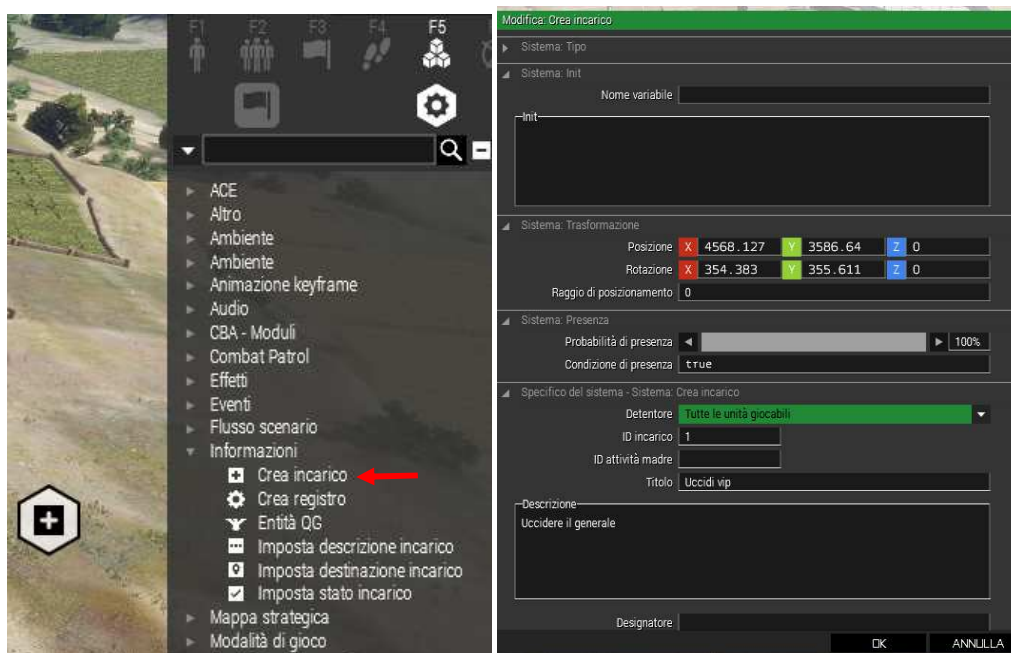
```
_vehgrp = _vehicle select 2;
_veh setVehicleVarName "she_heli";
she_heli = _veh;
_veh setposATL [(getPosATL _veh) select 0, (getPosATL _veh) select 1, 50];
_veh flyInHeight 50;
{_x setBehaviour "STEALTH"; _x setCombatMode "BLUE"; _x allowFleeing 0; _x disableAI "AUTOTARGET"; _x
disableAI "TARGET"; _x disableAI "FSM";} foreach units _vehgrp;
[[_veh], "sh_fnc", nil, true] call BIS_fnc_MP;
clearWeaponCargoGlobal _veh;
clearMagazineCargoGlobal _veh;
_wp1 = _vehgrp addWaypoint [position _landPos, 0];
_wp1 setWaypointType "MOVE";
_wp1 setWaypointSpeed "NORMAL";
_wp1 setWaypointCompletionRadius 5;
_wp1 setWaypointStatements ["true", "she_heli land 'GET OUT';{she_heli animateDoor [_x, 1]} forEach
['door_back_L', 'door_back_R', 'door_L', 'door_R']"];
sleep 1;
//Smoke or IR Strobe
waitUntil { (_veh distance _landPos) < 350 };
_smokePos = [getPos _landPos, 10, 25, 1, 0, 60 * (pi / 180), 0] call BIS_fnc_findSafePos;
_smoke = createVehicle ["SmokeShellYellow", _smokePos, [], 0, "NONE"];
// _smoke = createVehicle ["SmokeShell", _smokePos, [], 0, "NONE"];
// _light = createVehicle ["Chemlight_yellow", _smokePos, [], 0, "NONE"];
// _strobe = createVehicle ["I_IRStrobe", position _smokePos, [], 0, "NONE"];
_units = [] call BIS_fnc_listPlayers;
sleep 0.1;
//Wait until everyone is onboard
waitUntil { {alive _x && _x in she_heli} count _units == count ([] call BIS_fnc_listPlayers) };
{_veh animateDoor [_x, 0]} forEach ["door_back_L", "door_back_R", "door_L", "door_R"];
_veh lock true;
_wp2 = _vehgrp addWaypoint [getmarkerpos _goToPos, 0];
_wp2 setWaypointType "MOVE";
_wp2 setWaypointSpeed "NORMAL";
_wp2 setWaypointCompletionRadius 5;
/*deleteMarker "exma";*/
//Outro Variable
waitUntil { ([position _veh select 0, position _veh select 1, 0] distance _landPos) > 100 };
sh_Outro = true;
publicvariable "sh_Outro";
```

20 CREARE UN OBIETTIVO

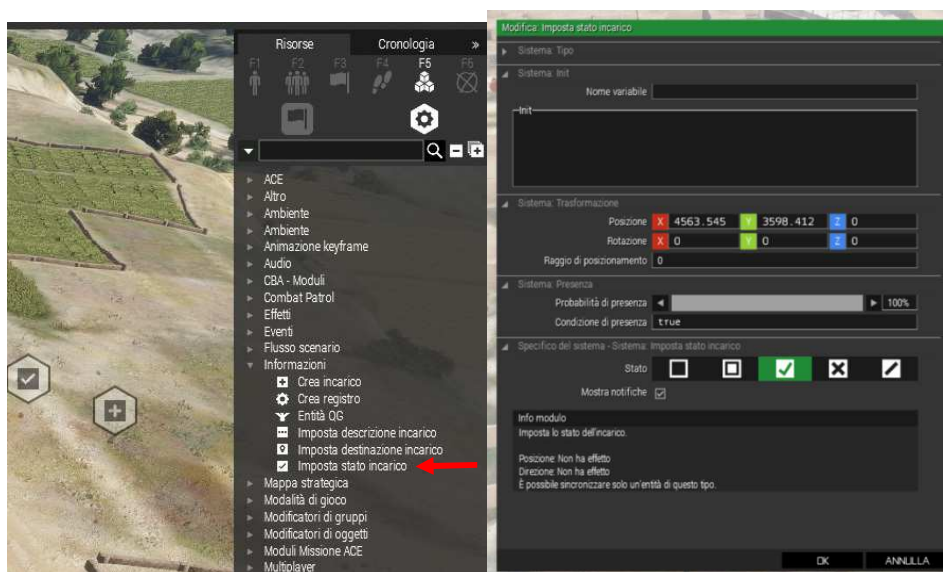
20.1 UCCIDI IL VIP (O DISTRUGGERE UN OGGETTO)

Creare un obiettivo come per esempio uccidi il Vip o qualsiasi oggetto. Passi da fare:

1. Mettere in mappa il personaggio vip e assegnargli un nome nel campo "nome variabile", per esempio diamogli il nome "uff". Nota: può essere anche un qualsiasi oggetto, basta assegnargli un nome specifico
2. Nei moduli Zeus mettere giù il modulo "Crea incarico" dal menu "informazioni". Assegnare quali giocatori, per esempio "per tutti i giocatori", il numero dell'incarico e la descrizione. Metterne lo stato ad "assegnato", cosicché è già operativo



3. Mettere giù il modulo "imposta stato incarico" ed assegniamogli il valore "compiuto"



4. Mettere giù il trigger ("interruttore") e mettere quest'istruzione nella condizione

```
!alive uff
```

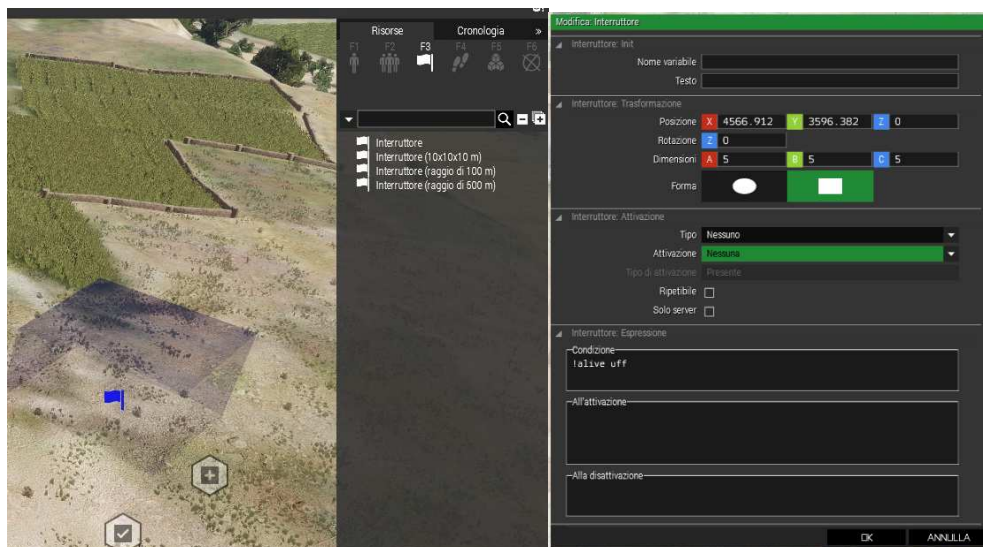
Questa controlla se il vip è morto o vivo.

Nota: nel caso fosse un oggetto la stessa controlla se esso è distrutto o meno.

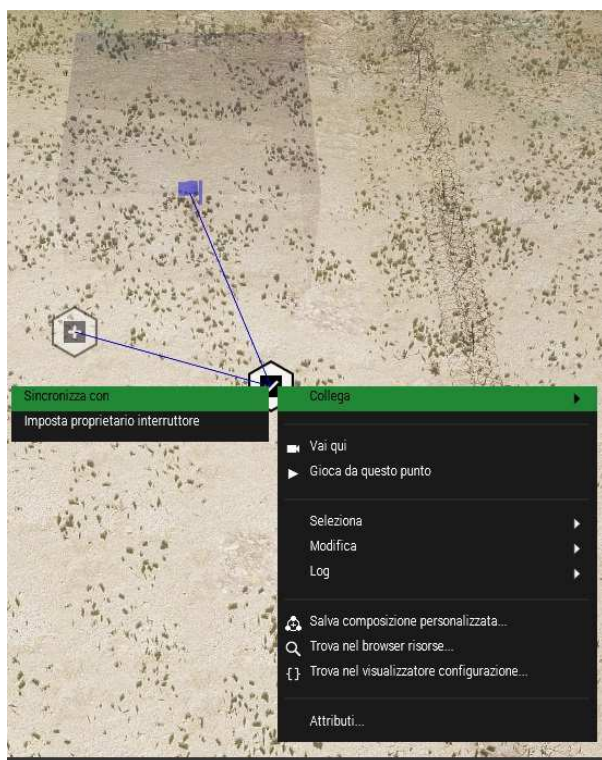
Il carattere "!" significa già NON. Quindi è come scrivere NONalive.

Se sono più obiettivi si scriverà

```
(!alive uff1)&&!alive uff2)
```



5. Sincronizzare i moduli con il trigger (tasto destro/collega/sincronizza con). Il modulo stato va sincronizzato con modulo incarico e quest'ultimo con l'interruttore



20.2 LIBERA L'OSTAGGIO

20.2.1 PRIMO MODO:

1. Mettere giù il civile ostaggio
2. Copiare questo script in un file testo e salvarlo come hostageScript.sqf. mettere il file nella cartella della missione

```
// Set AI Hostage Script  
// By Galactic Twinkles  
// Define Captive  
_captive = _this select 0;
```




```
// Select random animation
_anim = selectRandom
["Acts_AidlPsitMstpSsurWnonDnon01", "Acts_AidlPsitMstpSsurWnonDnon02", "Acts_AidlPsitMstpSsurWnonDnon03",
"Acts_AidlPsitMstpSsurWnonDnon04", "Acts_AidlPsitMstpSsurWnonDnon05", "Acts_ExecutionVictim_Loop"];

// Set Captive Settings
_captive setCaptive true;
// Remove Items
removeAllWeapons _captive;
removeBackpack _captive;
removeVest _captive;
removeAllAssignedItems _captive;
_captive switchMove _anim; // SwitchMove to random animation
// Set unit as hurt if it's the Execution animation
if (_anim == "Acts_ExecutionVictim_Loop") then {
    _captive setDamage .5;
};
_captive disableAI "MOVE"; // Disable AI Movement
_captive disableAI "AUTOTARGET"; // Disable AI Autotarget
_captive disableAI "ANIM"; // Disable AI Behavioural Scripts
_captive allowFleeing 0; // Disable AI Fleeing
_captive setBehaviour "Careless"; // Set Behaviour to Careless because, you know, ARMA AI.

// Add Hold Action to Free Hostage
[
/* 0 object */ _captive,
/* 1 action title */ "Free Hostage",
/* 2 idle icon */ "\a3\ui_f\data\IGUI\Cfg\HoldActions\holdAction_unbind_ca.paa",
/* 3 progress icon */ "\a3\ui_f\data\IGUI\Cfg\HoldActions\holdAction_unbind_ca.paa",
/* 4 condition to show */ "true",
/* 5 condition for action */ "true",
/* 6 code executed on start */ {},
/* 7 code executed per tick */ {},
/* 8 code executed on completion */ {
    if (_this select 3 select 0 ==
        "Acts_ExecutionVictim_Loop") then {
            _this select 0 playMove
        } else {
            _this select 0 switchMove
        };
        _complMessage = selectRandom ["I
        thought I was gonna die in here!", "Thank you so much man.", "I think I shit my pants...", "Can I hug
        you?", "I'M ALIVE.", "Where the hell am I?"];
        ["Hostage", _complMessage]
        remoteExec ["BIS_fnc_showSubtitle"];
        sleep 5.5;
        (_this select 0) enableAI "MOVE";
        (_this select 0) enableAI
        "AUTOTARGET";
        (_this select 0) enableAI "ANIM";
        (_this select 0) setBehaviour
        "SAFE";
        [(_this select 0)] joinSilent
        player;
        [(_this select 0), (_this select 2)]
        remoteExec ["bis_fnc_holdActionRemove", [0, -2] select isDedicated, true];
    },
/* 9 code executed on interruption */ {
        _intrMessage = selectRandom ["Hey! I don't
        wanna die here!", "Don't leave me here man! Please!", "THEY'RE EATING PEOPLE. GET ME OUT OF
        HERE.", "*Mumbles* Shit shit shit..."];
        ["Hostage", _intrMessage] remoteExec
        ["BIS_fnc_showSubtitle"];
    },
/* 10 arguments */ [_anim],
/* 11 action duration */ 3,
/* 12 priority */ 0,
/* 13 remove on completion */ true,
/* 14 show unconscious */ false
] remoteExec ["BIS_fnc_holdActionAdd", [0, -2] select isDedicated, true];
```



3. Copiare la seguente stringa nell'init del civile ostaggio

```
null = [this] execVM "hostageScript.sqf";
```

4. Una volta liberato l'ostaggio si unirà automaticamente alla squadra

Lo script funziona creando un comando attivabile quando ci si avvicina all'ostaggio.

20.2.2 SECONDO MODO CON SODDISFACIMENTO OBIETTIVO

1. Mettere giù il civile ostaggio assegnargli un nome nella variabile nome, per esempio "ostaggio"
Mettere questa stringa nell suo init

```
ostaggio switchMove "Acts_ExecutionVictim_Loop"; ostaggio disableAI "anim";
```

Descrizione della funzione "disableAI":

Disables parts of the AI behaviour to get a better control over the actions of a unit. Must be executed where AI unit is local. If unit changes locality, it might need to be executed again at the new locality to maintain effect. Possible values are:

- "TARGET" - stop the unit to watch the assigned target / group commander may not assign targets
- "AUTOTARGET" - prevent the unit from assigning a target independently and watching unknown objects / no automatic target selection
- "MOVE" - disable the AI's movement / do not move
- "ANIM" - disable ability of AI to change animation. **Available only since Arma: Cold War Assault (OFP 1.99).**
- "TEAMSWITCH" - AI disabled because of Team Switch
- "FSM" - disable the execution of AI behavior scripts. **Available only since Operation Arrowhead v1.60.**
- "WEAPONAIM" - no weapon aiming
- "AIMINGERROR" - prevents AI's aiming from being distracted by its shooting, moving, turning, reloading, hit, injury, fatigue, suppression or concealed/lost target **Available only since Arma 3 v1.42.**
- "SUPPRESSION" - prevents AI from being suppressed **Available only since Arma 3 v1.42.**
- "CHECKVISIBLE" - disables visibility raycasts **Available only since Arma 3 v1.54.**
- "COVER" - disables usage of cover positions by the AI **Available only since Arma 3 v1.56.**
- "AUTOCOMBAT" - disables autonomous switching to COMBAT when in danger **Available only since Arma 3 v1.56.**
- "PATH" - stops the AI's movement but not the target alignment **Available only since Arma 3 v1.61.**
- "MINEDETECTION" - disable Ai mine detection.
- "NVG" - Stops AI from putting NVGs on but not taking them off **Available since Arma 3 v1.92**
- "LIGHTS" - Stops AI from operating vehicle headlights as well as collision lights **Available since Arma 3 v1.92**
- "RADIOPROTOCOL" - Stops AI from talking and texting while still being able to issue orders **Available since Arma 3 v1.95**

NOTA: nel multiplayer la funzione disableAI potrebbe avere qualche problema

In alternativa si può usare anche questa

```
0 = [] spawn {while {alive ostaggio} do {ostaggio playMove "Acts_ExecutionVictim_Loop"; sleep 1;}};
```

NOTA: testarla perché una volta liberato il personaggio si rimette ostaggio

Nota: se sono più ostaggi definire più nomi e mettere la stringa per ognuno in ognuno di essi (non nel gruppo)

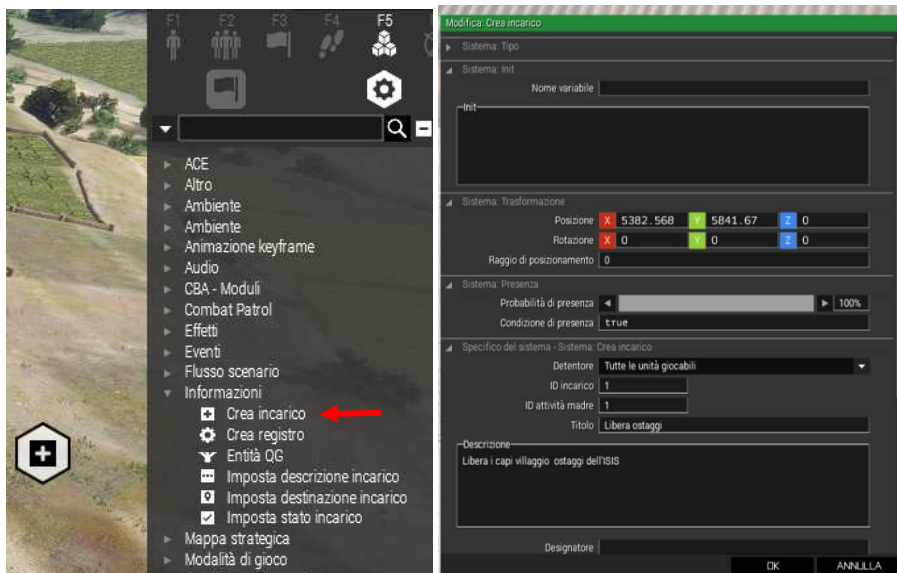
```
Ostaggio1 switchMove "Acts_ExecutionVictim_Loop"; Ostaggio1 disableAI "anim";
```

```
Ostaggio2 switchMove "Acts_ExecutionVictim_Loop"; Ostaggio2 disableAI "anim";
```

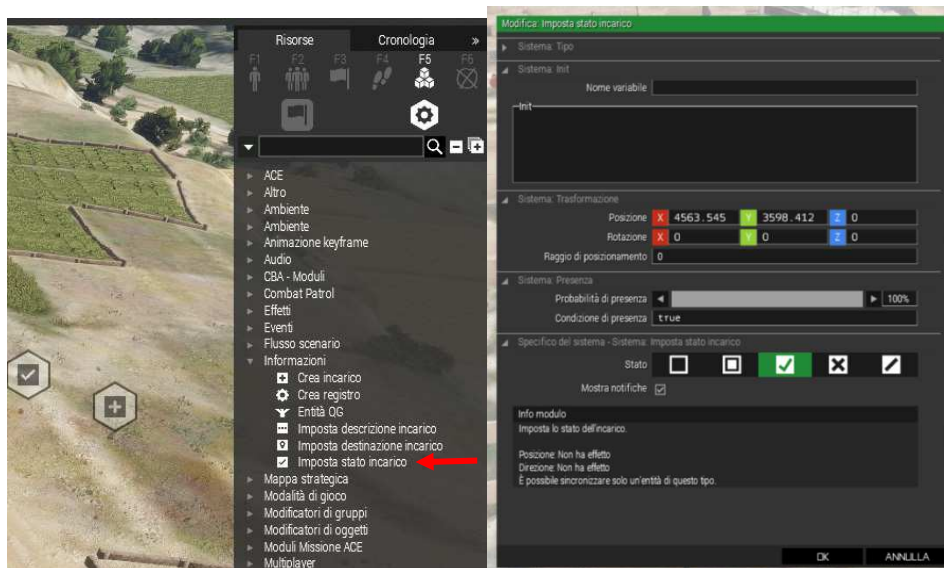
.....



1. Nei moduli Zeus mettere giù il modulo “Crea incarico” dal menu “informazioni”. Assegnare quali giocatori, per esempio “per tutti i giocatori”, il numero dell’incarico e la descrizione. Metterne lo stato ad “assegnato”, cosicché è già operativo



2. Mettere giù il modulo “imposta stato incarico” ed assegniamogli il valore “compiuto”

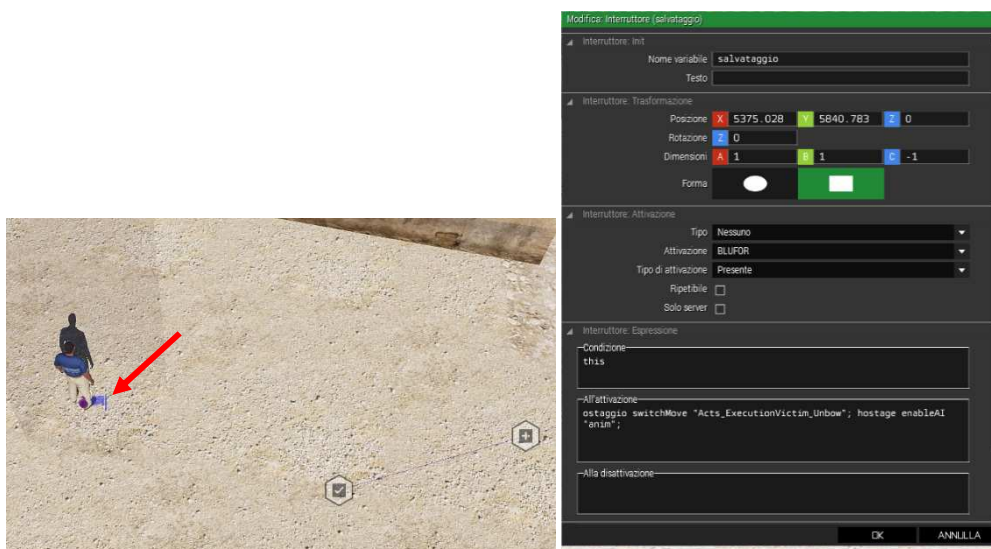


- 3. Mettere giù il trigger (“interruttore”) e posizionarlo intorno all’ostaggio con una dimensione per esempio di 2m. Assegnare un nome alla variabile, per esempio “salvataggio” e impostare l’attivazione con Blufor. All’attivazione del trigger inserire la stringa seguente, che libera l’ostaggio e attiva l’AI:

```
ostaggio switchMove "Acts_ExecutionVictim_Unbow"; ostaggio enableAI "anim";
```

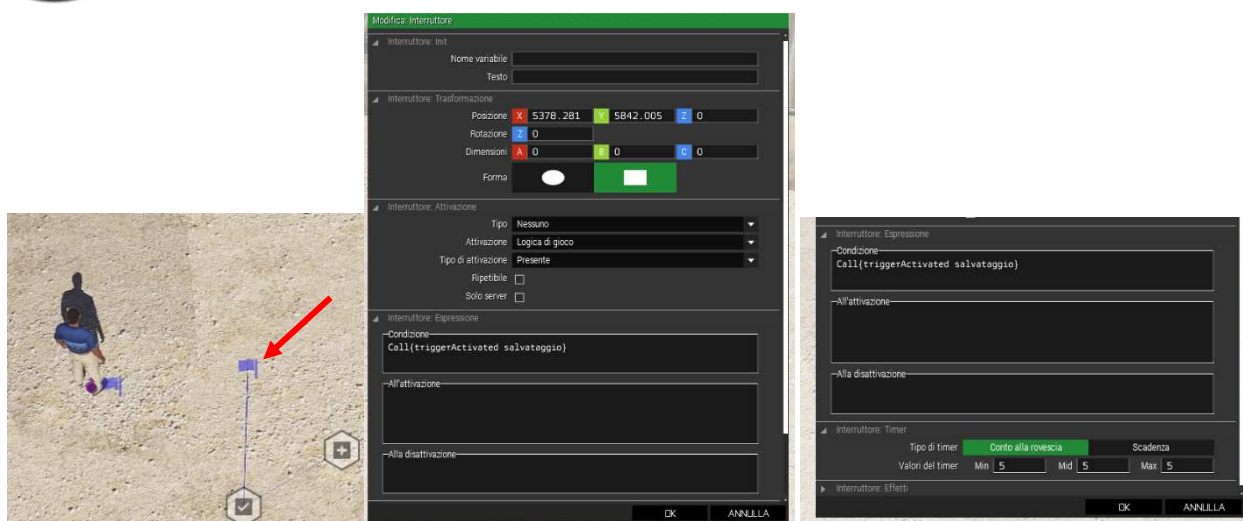
Nota: se sono più ostaggi ripetere la stringa per ognuno

```
Ostaggio1 switchMove "Acts_ExecutionVictim_Unbow"; Ostaggio1 enableAI "anim";  
Ostaggio2 switchMove "Acts_ExecutionVictim_Unbow"; Ostaggio2 enableAI "anim";  
Ostaggio3 switchMove "Acts_ExecutionVictim_Unbow"; Ostaggio3 enableAI "anim";
```



- 4. Mettere giù un altro trigger che attiverà il completamento dell’obbiettivo con un certo ritardo rispetto alla liberazione. Impostare i valori del timer per esempio a 5 sec. Attivazione “logica di gioco” e come condizione di soddisfacimento l’attivazione del trigger precedente “salvataggio”. Per far ciò basta mettere quest’istruzione nella condizione

```
Call{triggerActivated salvataggio}
```



5. Sincronizzare i moduli con quest'ultimo trigger (tasto destro/collega/sincronizza con). Il modulo stato va sincronizzato con modulo incarico e quest'ultimo con il trigger



20.2.3 TERZO MODO SEMPRE CON SODDISFACIMENTO OBIETTIVO (DA VERIFICARE)

In analogia con il secondo modo:

1. Mettere giù il civile ostaggio assegnargli un nome nella variabile nome, per esempio "ostaggio"
Mettere questa stringa nell suo init

```
ostaggio switchMove "Acts_ExecutionVictim_Loop"; ostaggio disableAI "anim";
```

```
[this,true] call ace_captives_fnc_setHandcuffed;
```

```
ostaggio1 getVariable ["ace_captive_isHandcuffed", false]
```

```
ostaggio1 getVariable ["ace_captives_fnc_setHandcuffed ", false]
```

Completing a task upon freeing a unit in ACE?

Title. I've set the unit to be handcuffed through the ACE settings in the editor and in the Init.sqf file i want to set it up so it waits until the unit has been free'd to complete a task, how would i do this ?

if the unit is named captive1, you can check if he is currently handcuffed with
 captive1 getVariable ["ace_captives_isHandcuffed", false]

You can put that for example in the condition of a trigger and link that with a task module, if you are using those.

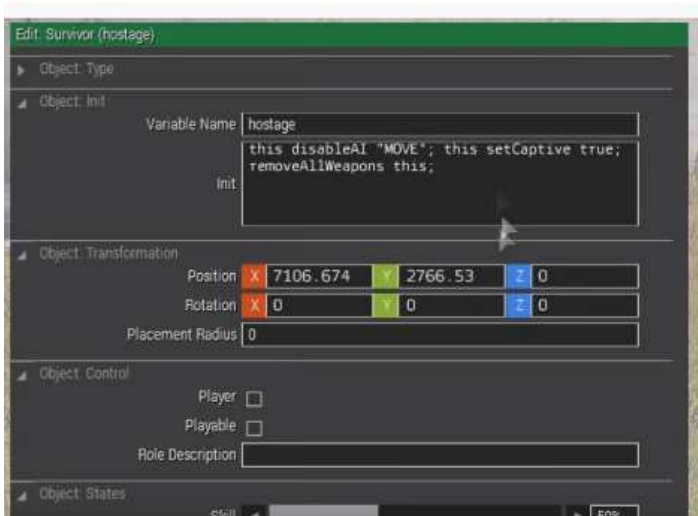
It also fires an event whenever the handcuff state changes, to you could try working with the
 "ace_captiveStatusChanged" CBA event handler too.

Altre info trovate sul web da testare:

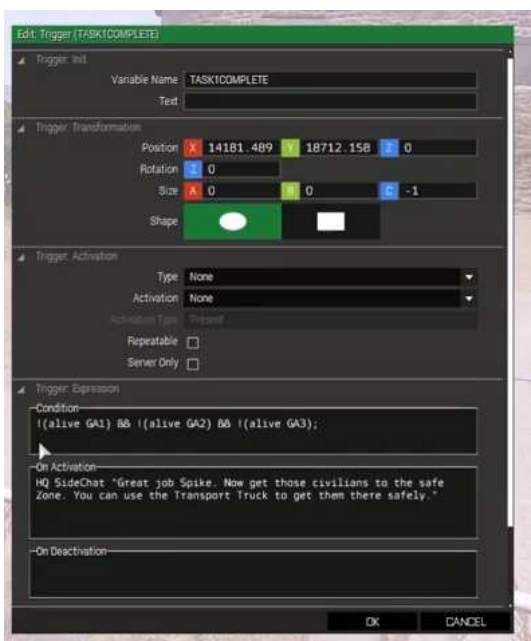
```
_hos setVariable ["Enh_isHostage",false]; // is NOT a hostage
```

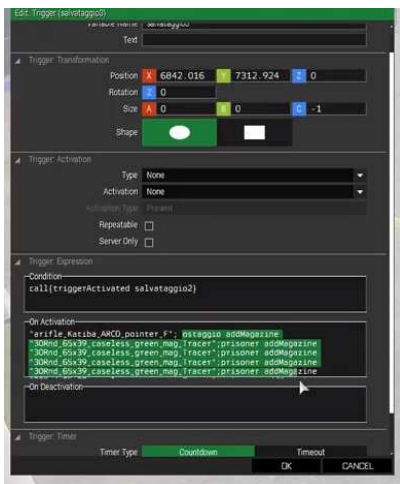


```
_hos setVariable ["Enh_isHostage",true]; // IS a hostage
```

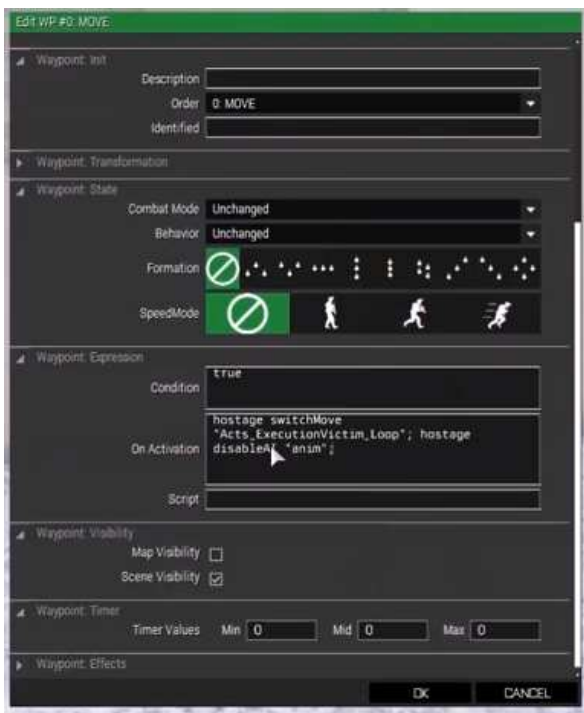


```
this disableAI "MOVE"; this setCaptive true;
```





```
Call{triggerActivated salvataggio}
```



```
hostage switchMove "Acts_ExecutionVictim_Loop"; hostage disableAI "anim";
```

21 RECUPERO INFORMAZIONI (INTEL)

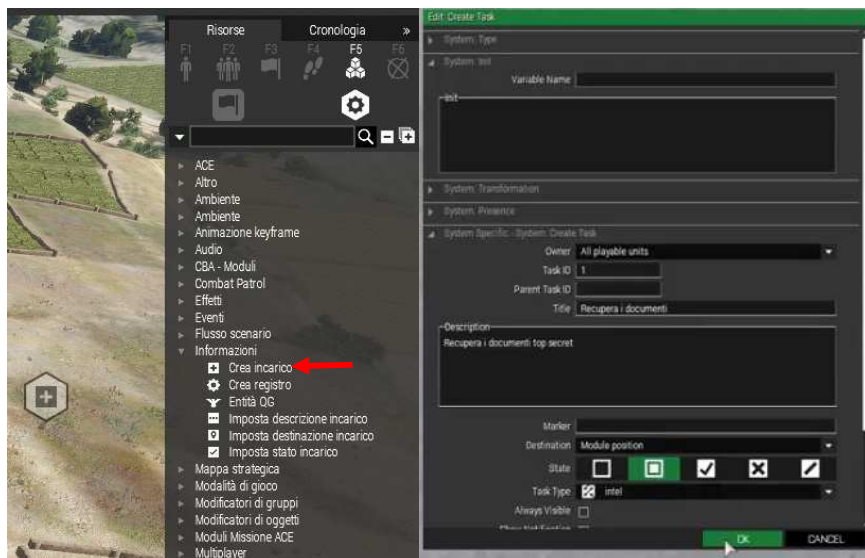
Procedura per creare un documento o similare da recuperare contenente informazioni.

1. Inserire un oggetto da recuperare come per esempio un documento e dargli un nome, es. "nomeintel"

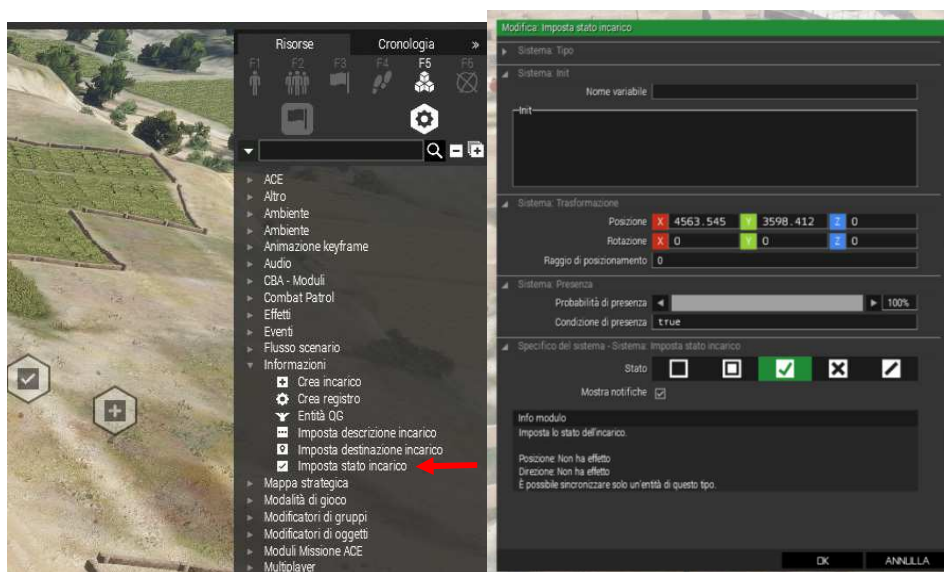




2. Mettere giù il modulo “Crea incarico”. Proprietario: tutti i giocatori, task: 1, Stato: assegnato



3. Mettere giù il modulo “imposta stato incarico” ed assegniamogli il valore “compiuto”



4. Mettere giù un trigger e sincronizzare i due moduli sopra con esso.



5. Aprire l’oggetto intel creato ed inserire la stringa nel suo campo init:

```
call{this AddAction [ "Prendi Documento", "collect.sqf" ]};
```




6. Creare un file .sqf nella cartella della missione ("... \Documents\Arma 3 - Other Profiles\Nome utente\missions\MissionName\") e chiamarlo collect.sqf. Mettere la seguente stringa all'interno del file

```
DeleteVehicle nomeintel;
```

7. Aprire il trigger e mettere la stringa seguente nel campo Condizioni

```
call{!alive nomeintel;}
```

Questa cancella il documento una volta che è stato preso e l'incarico è passato a successo



8. Volendo un messaggio aggiuntivo una volta presa l'intel si può creare un secondo trigger e mettere nella Condizione:

```
call{!alive nomeintel;}
```

Nel campo Attivazione la stringa:

```
hint "Testo del suggerimento che comparirà in alto a destra"
```

Utilizzare **hintC** per richiedere all'utente di fare clic su "Continua" prima di procedere.

Utilizzare **hintSilent** per disabilitare il segnale acustico.

Per visualizzare nuovamente l'ultimo suggerimento nel gioco, premere H.

Si potrebbe impostare anche un ritardo di qualche secondo in modo che la scritta appaia qualche secondo dopo il messaggio di obiettivo completato

Un modo alternativo più veloce per assegnare intel con eventuali messaggi (senza passare per il modulo incarico) è il seguente:

1. Creare l'oggetto e nominarlo, es. "nomeintel"
2. Inserire la stringa nel suo campo init:

```
call{this addAction ["Prendi Documento", {DeleteVehicle nomeintel}];}
```

3. Creare un trigger e mettere nella Condizione:

```
call{!alive nomeintel;}
```

Nel campo Attivazione inserire la stringa:

```
hint "Testo del suggerimento che comparirà in alto a destra"
```

// short and sweet

```
player addAction ["a useless action that does nothing", {}];  
player addAction ["<t color=#FF0000>This Useless Action Is RED</t>", {hint "RED"}];  
player addAction ["Hint Hello!", { hint format ["Hello %1!", name player] }];  
player addAction ["String Exec", "hint 'this is also compiled'"];
```



22 FAR COMPARIRE UN MARKER IN MAPPA CON TRIGGER

Per posizionare un marker in mappa basta andare nella sezione indicatori e piazzarli direttamente in mappa.

Qui però si vuole far apparire un marker (per esempio di obiettivo) in mappa una volta soddisfatta una condizione di trigger qualsiasi.

1. Mettere giù un marker vuoto e dargli un nome, es "marker01". Questo serve per identificare una posizione precisa dove far comparire il marker obiettivo che vogliamo creare. Andare sulla sezione degli indicatori e selezionare un marker vuoto dal menu system.
2. Creare un trigger con una condizione d'attivazione qualsiasi e mettere nel campo Attivazione la stringa

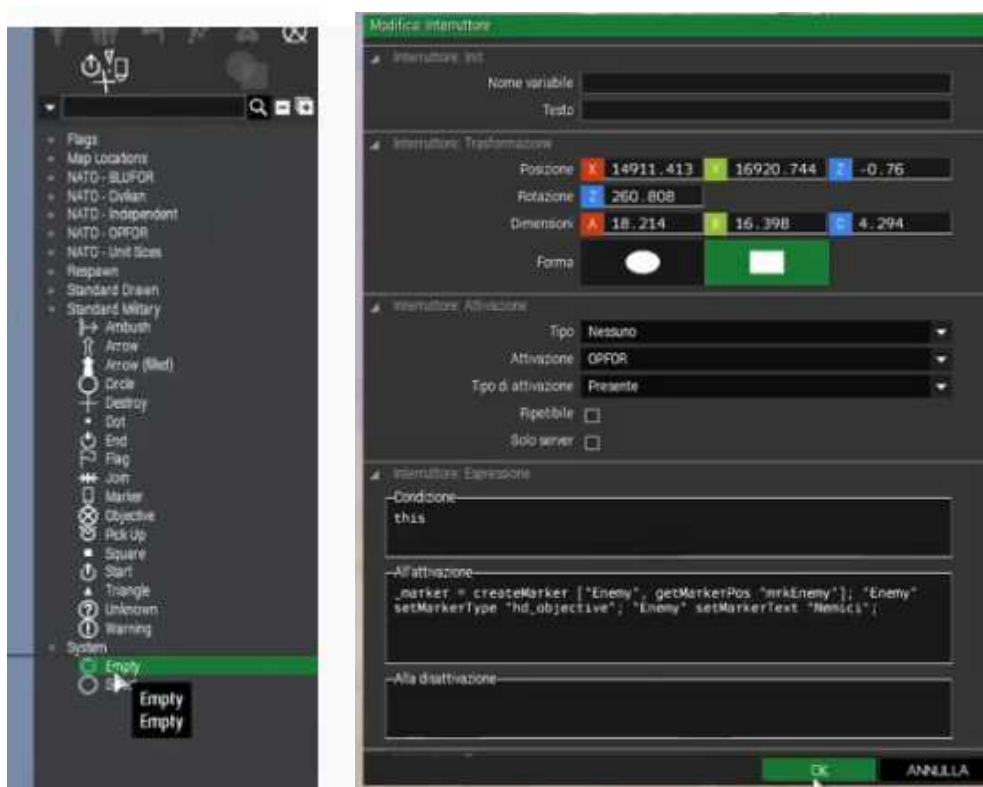
```
_marker = createMarker ["Enemy", getMarkerPos "marker01"];  
"Enemy" setMarkerType "hd_objective";  
"Enemy" setMarkerText "Nemici!!";
```

Questa crea una variabile locale `_marker` del tipo marker di nome Enemy nella posizione identificata dal marker precedente marker01.

A questo nuovo marker assegna un marker del tipo obiettivo "hd_objective" (vedi figura qui sotto).

NOTA: per aggiungere un tipo diverso con relativa icona, nella figura si possono trovare i nomi delle loro variabili da sostituire a "hd_objective".

Infine viene visualizzato un testo nel marker





0. Empty / "Empty"	<!-- All the above markers have a _noShadow variant (eg "mil_end_noShadow") with no shadow. -->	56. Naval (Blue) / "b_naval"	86. Plane (Civilian) / "c_plane"
1. Objective / "hd_objective"		57. Naval (Red) / "o_naval"	87. Fire Team / "group0"
2. Marker / "hd_marker"		58. Naval (Green) / "n_naval"	88. Squad / "group1"
3. Flag / "hd_flag"		59. Medical (Blue) / "b_med"	89. Section / "group2"
4. Arrow / "hd_arrow"		60. Medical (Red) / "o_med"	90. Platoon / "group3"
5. Arrow (filled) / "mil_arrow2"	29. Unknown (Blue) / "b_unknown"	61. Medical (Green) / "n_med"	91. Company / "group4"
6. Ambush / "hd_ambush"	30. Unknown (Red) / "o_unknown"	62. Artillery (Blue) / "b_art"	92. Battalion / "group5"
7. Destroy / "hd_destroy"	31. Unknown (Green) / "n_unknown"	63. Artillery (Red) / "o_art"	93. Regiment / "group6"
8. Start / "hd_start"	32. Infantry (Blue) / "b_inf"	64. Artillery (Green) / "n_art"	94. Brigade / "group7"
9. End / "hd_end"	33. Infantry (Red) / "o_inf"	65. HQ (Blue) / "b_hq"	95. Division / "group8"
10. Pickup / "hd_pickup"	34. Infantry (Green) / "n_inf"	66. HQ (Red) / "o_hq"	96. Corps / "group9"
11. Join / "hd_join"	35. Motorized (Blue) / "b_motor_inf"	67. HQ (Green) / "n_hq"	97. Army / "group10"
12. Warning / "hd_warning"	36. Motorized (Red) / "o_motor_inf"	68. Support (Blue) / "b_support"	98. Army Group / "group11"
13. Unknown / "hd_unknown"	37. Motorized (Green) / "n_motor_inf"	69. Support (Red) / "o_support"	99. Respawn / "respawn_unknown"
14. Dot / "hd_dot"	38. Mechanized (Blue) / "b_mech_inf"	70. Support (Green) / "n_support"	100. Infantry Respawn / "respawn_inf"
15. Square / "mil_box"	39. Mechanized (Red) / "o_mech_inf"	71. Maintenance (Blue) / "b_maint"	101. Car Respawn / "respawn_motor"
16. Triangle / "mil_triangle"	40. Mechanized (Green) / "n_mech_inf"	72. Maintenance (Red) / "o_maint"	102. Armor Respawn / "respawn_armor"
<!-- Non HD Marker versions below! -->	41. Armor (Blue) / "b_armor"	73. Maintenance (Green) / "n_maint"	103. Helicopter Respawn / "respawn_air"
17. Dot / "mil_dot"	42. Armor (Red) / "o_armor"	75. Service (Blue) / "b_service"	104. Plane Respawn / "respawn_plane"
18. Objective / "mil_objective"	43. Armor (Green) / "n_armor"	76. Service (Red) / "o_service"	105. Naval Respawn / "respawn_naval"
19. Flag / "mil_flag"	44. Recon (Blue) / "b_recon"	77. Service (Green) / "n_service"	106. Airborne Respawn / "respawn_para"
20. Arrow / "mil_arrow"	45. Recon (Red) / "o_recon"	78. Installation (Blue) / "b_installation"	107. Tree / "loc_Tree"
21. Ambush / "mil_ambush"	46. Recon (Green) / "n_recon"	79. Installation (Red) / "o_installation"	108. Small Tree / "loc_smallTree"
22. Destroy / "mil_destroy"	47. Helicopter (Blue) / "b_air"	80. Installation (Green) / "n_installation"	109. Bush / "loc_Bush"
23. Start / "mil_start"	48. Helicopter (Red) / "o_air"	81. Installation (Green/Round) / "u_installation"	
24. End / "mil_end"	49. Helicopter (Green) / "n_air"	82. Unknown (Civilian) / "c_unknown"	
25. Pick up / "mil_pickup"	50. Plane (Blue) / "b_plane"	83. Car (Civilian) / "c_car"	
26. Join / "mil_join"	51. Plane (Red) / "o_plane"	84. Ship (Civilian) / "c_ship"	
27. Warning / "mil_warning"	52. Plane (Green) / "n_plane"	85. Helicopter (Civilian) / "c_air"	
28. Unknown / "mil_unknown"	53. UAV (Blue) / "b_uav"		
	54. UAV (Red) / "o_uav"		
	55. UAV (Green) / "n_uav"		



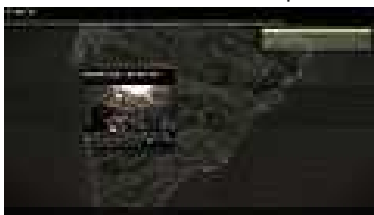
110. Church / "loc_Church"	124. Ruin / "loc_Ruin"	138. United Nations / "flag_UN"	154. Poland / "flag_Poland"
111. Chapel / "loc_Chapel"	125. Hotel / "loc_Tourism"	139. Belgium / "flag_Belgium"	155. Slovakia / "flag_Slovakia"
112. Cross / "loc_Cross"	126. Water Tower / "loc_WaterTower"	140. Canada / "flag_Canada"	156. Slovenia / "flag_Slovenia"
113. Rock / "loc_Rock"	127. Power Plant / "loc_Power"	141. Croatia / "flag_Croatia"	157. Spain / "flag_Spain"
114. Bunker / "loc_Bunker"	128. Solar Power Plant / "loc_PowerSolar"	142. Czech Republic / "flag_CzechRepublic"	158. UK / "flag_UK"
115. Fountain / "loc_Fountain"	129. Wave Power Plant / "loc_PowerWave"	143. Denmark / "flag_Denmark"	159. USA / "flag_USA"
116. View Tower / "loc_ViewTower"	130. Wind Power Plant / "loc_PowerWind"	144. France / "flag_France"	160. Select / "Select"
117. Lighthouse / "loc_Lighthouse"	131. NATO / "flag_NATO"	145. Georgia / "flag_Georgia"	161. Waypoint / "waypoint"
118. Pier / "loc_Quay"	132. CSAT / "flag_CSAT"	146. Germany / "flag_Germany"	162. Selector - Selectable / "selector_selectable"
119. Fuel Station / "loc_Fuelstation"	133. Altis Armed Forces / "flag_AAF"	147. Greece / "flag_Greece"	163. Selector - Selected Enemy / "selector_selectedEnemy"
120. Hospital / "loc_Hospital"	134. Altis / "flag_Altis"	148. Hungary / "flag_Hungary"	164. Selector - Selected Friendly / "selector_selectedFriendly"
121. Bus Stop / "loc_BusStop"	135. Altis (Colonial) / "flag_AltisColonial"	149. Iceland / "flag_Iceland"	165. Selector - Selected Mission / "selector_selectedMission"
122. Radio Tower / "loc_Transmitter"	136. FIA / "flag_FIA"	150. Italy / "flag_Italy"	166. KIA / "KIA"
123. Chimney Stack / "loc_Stack"	137. European Union / "flag_EU"	151. Luxembourg / "flag_Luxembourg"	167. Minefield / "Minefield"
		152. Netherlands / "flag_Netherlands"	168. Minefield (AP) / "MinefieldAP"
		153. Norway / "flag_Norway"	

23 CREARE UNA SCHERMATA DI CARICAMENTO PERSONALIZZATA DELLA MISSIONE

Le schermate di caricamento di Arma 3 sono composte da due sezioni principali:

Mappa - basata sul mondo attualmente caricato (Altis, Stratis...); a schermo intero. Configurato nella configurazione del mondo, non può essere modificato in missione.

Missione: visualizza l'anteprima della missione attualmente caricata; taglia unica. Configurato in missione.



Singleplayer mission



Multiplayer mission with missing configuration

23.1 MISSION CONFIGURATION

Description.ext

Per far apparire l'anteprima della missione con un'immagine, inserire la stringa seguente nel file description.ext (crearlo con notepad e salvarlo come .ext nella cartella della missione)

```
author = "Nome dell'autore"; // Author
onLoadName = "Nome della missione"; // Mission name (temporary solution; actual name set in editor will be used later in the alpha)
onLoadMission = "Descrizione della missione o quel che si vuole"; // Description
loadScreen = "nomefileimmagine.paa"; // Preview picture (nome del file immagine)
class Header
{
    gameType = Coop; // MP mission type
    minPlayers = 5; // numero minimo di players necessario per giocare la missione. Non inizia finchè non viene soddisfatto
    maxPlayers = 30; // numero massimo di players. Non viene avviata se superiori. Questo parametro viene by-passato se è impostata l'istruzione disabledAI=1; (vedi §24)
};
```

Nota1: in alcune fonti si trova anche
overviewPicture = "nomefileimmagine.paa";
al posto di
loadScreen = "nomefileimmagine.paa";



Alcuni lamentano che overviewPicture dia problemi in MP, mentre loadscreen funziona bene sia in SP che MP. Da test sembrano funzionare entrambi in maniera equivalente.

Nota2: minPlayers e maxPlayers si possono anche omettere (non si da limitazione di numero)

Creazione dell'immagine dell'anteprima:

- Il file deve essere in formato **PAA**
- Deve avere un aspect ratio di 2:1 (differenti rapporti vengono visualizzati correttamente, ma non coprono l'intera area o non riproducono correttamente i colori).
- L'ideale è 1024x512 pixels.
- Il file .paa va messo nella cartella della missione

Per creare il file in formato .paa si può usare TexView 2 scaricabile dal sito di Armaholic. Questo accetta formati png in ingresso.

Nota: il formato immagine potrebbe essere anche in jpg. Ad ogni modo sul sito Bohemia si richiede espressamente il paa, forse per un semplice motivo di ottimizzazione delle risorse.

23.2 WORLD CONFIGURATION

Giusto come nota l'immagine della mappa principale è configurata nel modo seguente.

Config.cpp

Contiene la stringa seguente

```
class CfgWorlds
{
    class Stratis: CAWorld
    {
        author = "Bohemia Interactive"; // Author
        description = "Stratis"; // World name
        pictureMap = "A3\map_Stratis\data\pictureMap_ca.paa"; // Background map
        pictureShot = "A3\map_Stratis\data\ui_Stratis_ca.paa"; // Default overview picture
        // Loading texts displayed on the bottom of the loading screen (picked randomly)
        loadingTexts[] =
        {
            "Area of Stratis is 19.3 square kilometers.",
            "Stratis is of volcanic origin, with many rocks and crevices.",
            "Kamino Firing Range uses remnants of the nearby old Kamino village as a
target area."
        };
    };
};
```

Creazione dell'immagine:

- Il file deve essere in formato **PAA**
- Grandezza di 2048x2048 della mappa satellitare effettiva utilizzata sul terreno facente parte del gioco o mod specifica

23.3 FULL SCREEN MISSION LOADING SCREEN

Esiste una mod in grado di applicare a schermo intero l'immagine di preview al §23.1. Questa è sottoscrivibile al <https://steamcommunity.com/sharedfiles/filedetails/?id=1121950907>

Questa mod rende lo schermo di caricamento della missione a schermo intero. La mappa non verrà più visualizzata. C'è una leggera sovrapposizione, che maschera alcuni dettagli su immagini di bassa qualità!

Da test sembra funzionare anche se non perfettamente.



24 DISABILITAZIONE DELLE AI AD INIZIO PARTITA (SOLO POSTI PER PLAYERS) E ABILITAZIONE DELLA CONSOLE PER L'ADMIN

Quando si lancia la missione in MP, di default gli slot non occupati dai players sono assegnati alle AI a meno di disabilitarli con il pulsante apposito che compare nella schermata di selezione.

Per disabilitare questa funzione, ossia far sì che tutti gli slot siano di default assegnati solo ai players, si possono inserire queste istruzioni nel description.ext

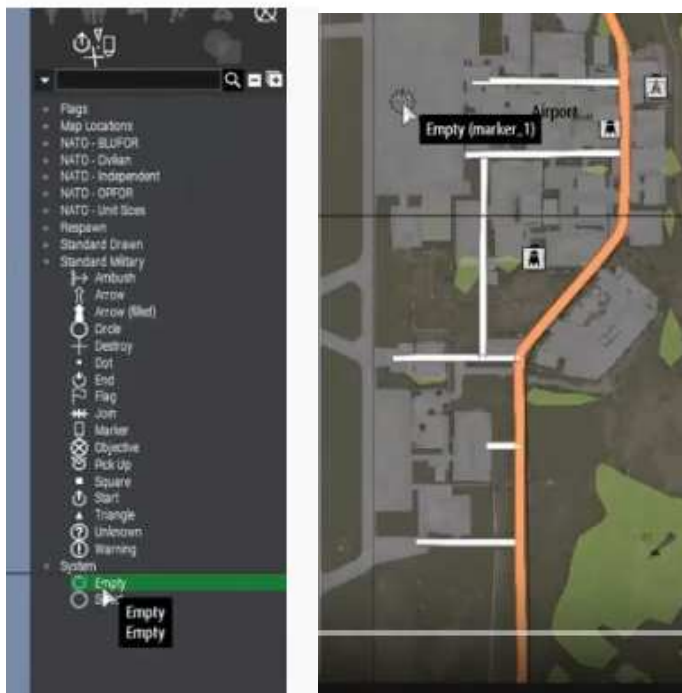
```
//DISABILITO LE AI AD INIZIO PARTITA (SOLO POSTI PER PLAYERS) E ABILITO LA CONSOLE PER L'ADMIN
disabledAI=1;
debriefing=1;
enableDebugConsole = 1;
```

L'ultima riga è importante perchè consente all'admin di accedere alla console dei comandi quando in partita altrimenti non accessibile. Questa può essere usata per scrivere eventuali istruzioni di codice.

25 IMPOSTARE LE POSIZIONI DI RESPAWN

Impostare la/le posizione/i del respawn per le unità.

1. Piazzare un marker in ogni posizione in mappa dove si vuole il respawn. Andare sui marker e selezionare un marker vuoto dal menu system.
2. Nominarlo come "respawn_west_nomedelrespawn" ("west" – indica che il respawn è della fazione Blufor, per la OPFOR usare _est)
Nota: "_est" è OPFOR, "_west" è BLUFOR, il civile è _civilian e la guerriglia è _independents. NOTA: rigenererà solo unità "giocabili".



3. Inserire questa stringa del file "description.ext" all'interno della cartella della missione. Crearlo nel caso non sia già stato definito

```
//Respawn Settings
respawn = 3; //Type of respawn (0=NONE, 1=BIRD, 2=INSTANT, 3=BASE, 4=GROUP, 5=SIDE)
respawnDelay = 10; //imposta il tempo di ritardo del respawn
respawnDialog = 0;
respawnOnStart = -1; //fa sì che all'inizio missione il respawn sia solo in una posizione predefinita,
-1 - Dont respawn on start. Don't run respawn script on start, 0 - Dont respawn on start. Run respawn
script on start. 1 - Respawn on start. Run respawn script on start.
```



```
respawnTemplates[] = { "MenuPosition","Tickets" }; //fa apparire il menu con le posizioni dove  
respawnare
```

alternativa di istruzioni per lo script:

```
respawn = 3; //Type of respawn (0=NONE, 1=BIRD, 2=INSTANT, 3=BASE, 4=GROUP, 5=SIDE)  
respawnDelay = 10; //Time before respawning player  
respawnVehicleDelay = 30; //Time before respawning vehicle  
respawnTemplatesWest[] = {"MenuPosition","MenuInventory"}; //Allows BLUFOR player to pick respawn  
points and loadout  
respawnTemplatesEast[] = {"MenuPosition","MenuInventory"}; //Allows OPFOR player to pick respawn  
points and loadout  
respawnTemplatesGuer[] = {"MenuPosition","MenuInventory"}; //Allows INDEPENDANT player to pick  
respawn points and loadout  
respawnOnStart = 0; //Respawns player on mission startup if set to "1"  
disabledAI = 0; //Disables AI taking over non-filled player slots if set to "1"  
joinUnassigned = 0; //Player will be auto-assigned to side with fewest players if set to "0"  
aiKills = 0; //Enables score list for AI players if set to "1"
```

Conviene sempre utilizzare la procedura di respawn con il file description.ext. NON da mai problemi al contrario di altre come queste di seguito.

IMPORTANTE: abilitare il salvataggio del loadout nelle impostazioni di respawn della ACE. Andare in Impostazioni/opzioni per gli addon e scorrere il menu a "ACE Riapparizioni", qui spuntare "Salvare equipaggiamento"

Alternativa

Impostare il respawn tramite gli attributi in multiplayer. Open the attributes > multiplayer and configure the "Respawn on Custom Position"

NOTA: questa procedura però è poco pratica e può dare dei problemi.

Alternativa

Si può usare in alternativa il metodo aggiornato con il modulo/multiplayer/respawn

Place a "Respawn Position" module on the location you want your units to respawn at. (Systems (F5) > Modules > Multiplayer0

Define the parameters to your liking in the module, as well as in the "Respawn" section in the multiplayer settings.

NOTA: questa procedura però è poco pratica e può dare dei problemi.

25.1 IMPOSTARE MANUALMENTE L'ALTEZZA DEL RESPAWN (O UNA COORDINATA)

Può capitare di dover impostare diversamente una delle coordinate di un marker per diverse ragioni. Per esempio quando si fa il respawn su una nave, ciò evita che i players cadino in mare al respawn. Si procederà:

1. Piazzare un marker vuoto nella posizione di respawn (sulla nave per esempio).
2. Nominarlo come "respawn_west".
3. Creare il file init.sqf all'interno della cartella missione e mettervi la stringa

```
_mrk = getMarkerPos "respawn_west";  
"respawn_west" setMarkerPoLocal[_mrk select 0, _mrk select 1, 35]; //variare l'altezza 35m con quella  
voluta
```

Questa definisce una variabile locale `_mrk` sulla posizione del marker `respawn_west`, e poi ridefinisce il marker `respawn_west` in una nuova posizione che ha le prime due coordinate X,Y in quelle originarie di `_mrk` e la coordinata dell'altezza Z ad una quota di 35m.

Variando la quota si potrà respawnare per esempio sul ponte della nave.

Il metodo vale anche se si volesse cambiare una delle quote X o Y.

26 ESEMPIO DI FORMA BASE DI UN FILE DESCRIPTION.EXT

È consigliabile usare sempre un file description.ext che di base potrebbe essere della forma

```
//DICHIARO IL NOME, L'IMMAGINE, IL SOTTOTITOLO DELLA MISSIONE E L'AUTORE  
onLoadName = "";  
overviewPicture = "";
```



```
onLoadMission = "";
author = "";

//DEFINISCO IL TIPO DI MISSIONE
class Header
{
    gameType = COOP;
    minPlayers = 1;
    maxPlayers = 30;
};

//DISABILITO LE AI AD INIZIO PARTITA (SOLO POSTI PER PLAYERS) E ABILITO LA CONSOLE PER L'ADMIN
disabledAI=1;
debriefing=1;
enableDebugConsole = 1;

//DEFINISCO DOVE FARE RESPAWN, IL TEMPO IN SECONDI PRIMA DEL RESPAWN ED IL TIPO DI INFORMAZIONE DA DARE
DURANTE IL RESPAWN
respawn = BASE;
respawnDelay = 10;
respawnDialog = 0;
```

Tenendo conto per le varie sezioni quanto scritto anche ai paragrafi §23.2, §24 e §25, la forma può essere:

```
// DICHIARO IL NOME, L'IMMAGINE, IL SOTTOTITOLO DELLA MISSIONE E L'AUTORE
author = "Nome dell'autore"; // Author
onLoadName = "Nome della missione"; // Mission name (temporary solution; actual name set in editor will
be used later in the alpha)
onLoadMission = "Descrizione della missione o quel che si vuole"; // Description
loadScreen = "nomefileimmagine.paa"; // Preview picture (nome del file immagine)

//DEFINISCO IL TIPO DI MISSIONE
class Header
{
    gameType = Coop; // MP mission type
    minPlayers = 1; // numero minimo di players necessario per giocare la missione. Non inizia
fincè non viene soddisfatto
    maxPlayers = 30; // numero massimo di players. Non viene avviata se superiori. Questo
parametro viene by-passato se è impostata l'istruzione disabledAI=1
};

//DISABILITO LE AI AD INIZIO PARTITA (SOLO POSTI PER PLAYERS) E ABILITO LA CONSOLE PER L'ADMIN
disabledAI=1;
debriefing=1;
enableDebugConsole = 1;

//DEFINISCO DOVE FARE RESPAWN, IL TEMPO IN SECONDI PRIMA DEL RESPAWN ED IL TIPO DI INFORMAZIONE DA DARE
DURANTE IL RESPAWN
respawn = 3; //Type of respawn (0=NONE, 1=BIRD, 2=INSTANT, 3=BASE, 4=GROUP, 5=SIDE)
respawnDelay = 10; //imposta il tempo di ritardo del respawn
respawnDialog = 0;
respawnOnStart = -1; //fa si che all'inizio missione il respawn sia solo in una posizione predefinita,
-1 - Dont respawn on start. Don't run respawn script on start, 0 - Dont respawn on start. Run respawn
script on start. 1 - Respawn on start. Run respawn script on start.
respawnTemplates[] = { "MenuPosition","Tickets" }; //fa apparire il menu con le posizioni dove
respawnare
```

Vedere §23.2, §24 e §25 per i dettagli.

27 INSERIRE UN TESTO/FRASE ALL'INIZIO DELLA MISSIONE

Questo semplice script serve per far apparire una scritta in basso a destra all'inizio della missione.

1. Creare il file init.sqf
2. inserire nel file init.sqf

```
execVM "intro.sqf";
```

3. Creare un file chiamato intro.sqf ed incollarci questo:

```
sleep 5;
```




```
[str("Testo") , str("Testo") , str("Testo")] spawn BIS_fnc_infoText;
```

Dove sleep 5 è il tempo in secondi, dopo il quale apparirà il messaggio. "Testo" la frase o la parola che apparirà, ognuna su una riga diversa.

esempio:

Testo

Testo

Testo

Esempio tipo macchina da scrivere militare

```
sleep 5;  
disableUserInput false;  
sleep 10;  
["Dark Thunder",str(date select 1) + "." + str(date select 2) + "." + str(date select 0),"75.5km from  
Kandahar"] spawn BIS_fnc_infoText;
```

Questo di seguito è un esempio di briefing di missione che può essere inserito con tale metodo (fonte 9GU).

intro.sqf

```
waitUntil { !isNil {player} };  
waitUntil { player == player };  
  
//Terzo "paragrafo"  
player createDiaryRecord ["Diary", ["OBIETTIVI DI MISSIONE", "  
  
- Riprendere il controllo dell'aeroporto<br/>  
- Abbattere l'anti aerea<br/>  
- Proteggere i civili<br/>  
- Recuperare eventuali diplomatici ancora vivi  
  
"]];  
  
//secondo "paragrafo"  
player createDiaryRecord ["Diary", ["SITUAZIONE ATTUALE", "  
  
Il gruppo ha già messo a ferro e fuoco l'intera isola, prendendo il controllo dell'aeroporto e  
dell'unica batteria anti aerea presente sull'isola,  
residuo della guerra fredda ma perfettamente funzionante, con il quale hanno abbattuto un elicottero  
con all'interno dei diplomatici e dei negoziatori,  
inviati dagli americani, per cercare di trovare un accordo pacifico.  
  
"]];  
  
//primo "paragrafo"  
player createDiaryRecord ["Diary", ["CONTESTO", "  
  
Utes, piccola isola nel Mare di Bering, ex colonia russa poi diventata 'Repubblica di Utes' a seguito  
della caduta dell'unione sovietica, da molti anni  
è ormai un tranquillo angolo di paradiso, dove la popolazione vive serena.<br/><br/>  
Annessa ideologicamente agli USA e popolata principalmente da americani, è da alcune settimane sotto  
assedio da parte  
di un gruppo non meglio definito di ribelli che hanno deciso di renderla la loro patria  
  
"]];
```



28 CREARE UN VIDEO INTRO DELLA MISSIONE

28.1 PRIMO METODO

Questo modo sfrutta dei file sqf già preimpostati che si possono trovare all'indirizzo

<http://www.armaholic.com/page.php?id=30731>

I file che servono sono i seguenti:

- File da inserire in una cartella **/AL_intro** dentro la cartella missione (~\Documents\Arma3-Altri profili/(il tuo nome)\missioni/(nome della missione)\AL_intro)
 - **camera_work.sqf**
 - **intro.sqf**
 - **time_srv.sqf**
- File da inserire dentro la cartella della missione (~\Documents\Arma3-Altri profili/(il tuo nome)\missioni/(nome della missione))
 - **description.ext** (contenente l'eventuale musica dell'intro)
 - **init.sqf**
- File da inserire in una cartella **/sound** dentro la cartella missione (~\Documents\Arma3-Altri profili/(il tuo nome)\missioni/(nome della missione)\sound)
 - **intro_music.ogg** (file musica dell'intro)

Creare le cartelle **/AL_intro** e **/sound** se non già presenti.

Un sito dove trovare musiche intro per missioni, oltre a youtube ovviamente, è <https://www.fesliyanstudios.com/>

Questo metodo si basa sul posizionamento di elipad invisibili per indicare la posizione delle camere e dei target inquadrati. Rispetto al primo consente di avere dei riferimenti visibili nell'editor per identificare camere ed inquadrature. Occorre però fare qualche piccola modifica in alcune righe degli script dei file che sono richiamati. Come procedere:

1. Posizionare un elipad invisibile dove si vuole la camera e nominarlo "cam1"
 2. Posizionare un elipad invisibile dove si vuole che la camera inquadrati e nominarlo "target1"
 3. Creare altre cam a piacimento (cam2,... cam6) ed altri target da inquadrare (target2,... target4), sempre tramite gli elipad invisibili. Nota: non necessariamente le cam devono essere in numero uguale ai target, ma dipende dai punti e le inquadrature da impostare nella propria intro.
 4. Aprire il file intro.sqf
- Esso è del tipo

```
// by ALIAS
// nul = [JIP] execVM "AL_intro\intro.sqf";

waitUntil {time > 0};

_jip_enable          = _this select 0;

[[_jip_enable], "AL_intro\time_srv.sqf"] remoteExec ["execVM"];
waitUntil {!isNil "curr_time"};

if (!hasInterface) exitWith {};

if ((!curr_time) or (_jip_enable<0)) then {
cutText ["INSERIRE IL NOME MISSIONE QUI", "BLACK IN", 10];
playsound "intro_music";

/* ----- how to use camera script -----
--

_camera_shot = [position_1_name, position_2_name, target_name, duration, zoom_level1, zoom_level_2,
attached, x_rel_coord, y_rel_coord, z_rel_coord, last_shot] execVM "camera_work.sqf";

Where
_camera_shot          - string, is the name/number of the camera shot, you can have as many as you want
see examples from down bellow
position_1_name       - string, where camera is created, replace it with the name of the object you want
camera to start from
position_2_name       - string, the object where you want camera to move, if you don't want to move from
initial position put the same name as for position_1_name
target_name           - string, name of the object you want the camera to look at
duration              - seconds, how long you want the camera to function on current shot
```



```
zoom_level_1      - takes values from 0.01 to 2, FOV (field of view) value for initial position
zoom_level_2      - takes values from 0.01 to 2, FOV value for second position, if you don't to change
you can give the same value as you did for zoom_level_1
attached          - boolean, if you want to attach camera to an moving object its value has to be
true, otherwise must be false

                                        in this case position_1_name must have the same value
as position_2_name
x_rel_coord       - meters, relative position to the attached object on x coordinate
y_rel_coord       - meters, relative position to the attached object on y coordinate
z_rel_coord       - meters, relative position to the attached object on z coordinate
last_shot         - boolean, true if is the last shot in your movie, false otherwise

-----*/

// - do not edit or delete the lines downbelow, this line must be before first camera shot
loopdone = false;
while {!loopdone} do {
//^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ DO NOT EDIT OR DELETE
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
}

// EXAMPLES----- insert your lines for camera shots starting from here -----
-----

_firstshot = [cam1, cam2, target1, 9, 0.3, 0.1, false, 0, 0, 0,FALSE] execVM
"AL_intro\camera_work.sqf";
waitUntil {scriptdone _firstshot};

_secondshot = [cam3, cam3, target2, 10, 1, 1, false, 0, 0, 0,FALSE] execVM "AL_intro\camera_work.sqf";
waitUntil {scriptdone _secondshot};

_thirdshot = [cam4, cam4, target3, 15, 1, 1, true, 0, -6, 3,TRUE] execVM "AL_intro\camera_work.sqf";
waitUntil {scriptdone _thirdshot};

/*
if you want to add a forth or a fifth camera shot use a code like:
_forthshot = [cam5, cam6, target4, 7, 1, 1, false] execVM "camera_work.sqf";
waitUntil {scriptDone _forthshot};

** Note that last boolean parameter will tell the script if the camera shot is the last one or not,
make sure that last camera has that parameter true and the intermediar cameras has it false as in my
examples above

>>!! don't forget to name the objects cam5, cam6, target4 in editor

You can add as many camera shots as you want
but you have to name the script differently
and don't forget to add the wait line after each shot
waitUntil {scriptDone _xxxshot};
*/

// ----->>> end of camera shots <<-----
};

cutText [" ", "BLACK IN", 3];
_camera = "camera" camCreate (getpos player);
_camera cameraeffect ["terminate", "back"];
camDestroy _camera;
"dynamicBlur" ppEffectEnable true;
"dynamicBlur" ppEffectAdjust [100];
"dynamicBlur" ppEffectCommit 0;
"dynamicBlur" ppEffectAdjust [0.0];
"dynamicBlur" ppEffectCommit 4;
};
```

In esso vengono definite la sequenza delle inquadrature.

5. Mettere il nome della propria missione sostituendo il testo in “INSERIRE IL NOME MISSIONE QUI”
6. Nell’esempio sono messe 6 camere (cam1, ... cam6) e 4 inquadrature (target1,... target4) identificate dalle funzioni _firstshot, ... _forthshot.

Modificare la stringa dell’inquadratura a seconda dell’esigenze. La stringa da modificare sono quelle del tipo



```
_firstshot = [cam1, cam2, target1, 9, 0.3, 0.1, false, 0, 0, 0,FALSE] execVM  
"AL_intro\camera_work.sqf";  
waitUntil {scriptdone _firstshot};
```

la spiegazione dei parametri è nello script.

In questa stringa la cam1 è il nome della camera; cam2 l'oggetto verso cui si sposta la cam1 (se si vuole mantenerla fissa basta mettere di nuovo il nome della cam1 invece di cam2); target1 è l'oggetto inquadrato; 9 la durata in secondi della ripresa; 0.3 e 0.1 i valori di zoom nella prima e seconda posizione; gli altri lasciarli così come sono.

7. Creare la cartella sound e mettervi il file audio dell'intro. Chiamarlo come "intro_music.ogg"

8. Nel file description.ext inserire il richiamo al file audio dell'intro

```
class CfgSounds  
{  
    class intro_music  
    {  
        name = "intro_music"; // Name for mission editor  
        sound[] = {\sound\intro_music.ogg, .5, 1.0};  
        titles[] = {0, ""};  
    };  
};
```

9. Nel file init.sqf impostare la modalità di riproduzione per i giocatori che entrano (vedi istruzioni nello script)

```
if (!(isServer) && (player != player)) then {waitUntil {player == player}};  
  
/*  
nul = [JIP] execVM "intro.sqf";  
  
JIP - number, time in seconds  
        - if negative the intro will be played for all JIP players regardless the  
time they join  
        - if is bigger than 0, players joining after the amount of seconds  
specified will not see the intro  
  
Examples  
> INTRO will be played for all JIP players regardless of joining time  
nul = [-1] execVM "AL_intro\intro.sqf";  
  
> INTRO will be played for all JIP players if they join in the first 10 seconds after mission start  
nul = [10] execVM "AL_intro\intro.sqf";  
*/  
  
nul = [-1] execVM "AL_intro\intro.sqf";  
//nul = [60] execVM "AL_intro\intro.sqf";
```

10. File camera_work.sqf. Esso NON va modificato

```
// by ALIAS  
// _camera_shot= [position_1_name, position_2_name, target_name, duration, zoom_level1, zoom_level_2,  
attached, x_rel_coord, y_rel_coord, z_rel_coord,last_shot] execVM "camera_work.sqf";  
_campos1= _this select 0;  
_campos2= _this select 1;  
_targetcam = _this select 2;  
_camera_duration = _this select 3;  
_zoom_level1 = _this select 4;  
_zoom_level2 = _this select 5;  
_cam_attached = _this select 6;  
_x_coord = _this select 7;  
_y_coord = _this select 8;  
_z_coord = _this select 9;  
_last_shot = _this select 10;  
  
// -----  
  
if (!loopdone) then  
{  
    if (_cam_attached) then {  
        _camera = "camera" camCreate (getpos _campos1);  
        showCinemaBorder true;  
        _camera attachTo [_campos2, [_x_coord,_y_coord,_z_coord]];
```



```
_camera cameraEffect ["internal", "BACK"];
_camera camCommand "inertia on";
_camera camPrepareTarget _targetcam;
_camera camPrepareFOV _zoom_level1;
_camera camCommitPrepared _camera_duration;
sleep _camera_duration;

} else
{
// initial/start position where camera is created
_camera = "camera" camCreate (getpos _campos1);
    showCinemaBorder true;
_camera cameraEffect ["internal", "BACK"];
_camera camCommand "inertia on";
_camera camPrepareTarget _targetcam;
_camera camPrepareFOV _zoom_level1;
_camera camCommitPrepared 0;

// poz 2 - where camera is moving next - target2
_camera camPreparePos (getpos _campos2);
_camera camPrepareTarget _targetcam;
_camera camPrepareFOV _zoom_level2;
_camera camCommitPrepared _camera_duration;
sleep _camera_duration;
};
if (_last_shot) then {loopdone = true};
};
```

11. File time_srv.sqf. Esso NON va modificato

```
// by ALIAS
if (!isServer) exitWith {};

_check_time = _this select 0;

curr_time = false;
publicVariable "curr_time";

if (_check_time>0) then
{
    sleep _check_time;
    curr_time = true;
    publicVariable "curr_time";
};
```

28.2 SECONDO METODO

Di seguito come creare un'intro per la missione sfruttando il posizionamento di più camere che riprendono una scena preimpostata in mappa.

1. Posizionare un'unità in un punto qualsiasi
2. Mettere tale stringa nel suo init:

```
this exec "camera.sqs";
```

questa attiva la funzione che permette di salvare la posizione di una camera

3. Avviare il gioco con l'unità
4. Spostare la visuale (vedi i comandi in fondo) fino alla posizione voluta per piazzare la prima camera che riprende l'inizio della intro. Quando in posizione left click per copiare la posizione
5. Aprire un file testo e salvarlo come "Intro.sqs" nella cartella della missione. Incollarvi quanto copiato. Sarà qualcosa di simile

```
;comment "23:17:07"; // Posizione della camera
_camera camPrepareTarget [-71785.54,57616.60,-37720.29]; //quello che la camera stà guardando
_camera camPreparePos [3221.69,3317.72,17.44]; //posizione della camera nel mondo
_camera camPrepareFOV 0.700; //FOV camera (campo della vista)
_camera camCommitPrepared 0 //quanto tempo si impiegherà per passare alla camera successiva
```

6. Creare la camera aggiungendo le seguenti 4 righe all'inizio del testo e rinominare la posizione della camera. Sarà qualcosa del tipo

```
=== Create Camera Start
```



```
_camera = "camera" camcreate [3695.91,3610.04,2.00]
_camera cameraeffect ["internal","back"]
;=== Create Camera End

;=== Intro Start

;comment "Camera 1st Position"; // Rinominare la posizione della 1st camera
_camera camPrepareTarget [-71785.54,57616.60,-37720.29]; //quello che la camera stà guardando
_camera camPreparePos [3221.69,3317.72,17.44]; //posizione della camera nel mondo
_camera camPrepareFOV 0.700; //FOV camera (campo della vista)
_camera camCommitPrepared 0 //quanto tempo si impiegherà per passare alla camera successiva
@camCommitted _camera // N.B. METTERE TALE ISTRUZIONE ALLA FINE DI OGNI CAMERA CREATA

~5 // Secondi di ritardo prima del termine dell'intro. Quanto tempo rimane fissa la camera. Se non si
vuole un ritardo eliminare tale riga
```

7. Ritornare in gioco e spostarsi dove si vuole inserire la seconda camera. Click tasto sinistro e copiare il contenuto nel file Intro.sqs

```
;=== Create Camera Start
_camera = "camera" camcreate [3695.91,3610.04,2.00]
_camera cameraeffect ["internal","back"]
;=== Create Camera End

;=== Intro Start

;comment "Camera 1st Position"; // Rinominare la posizione della 1st camera
_camera camPrepareTarget [-71785.54,57616.60,-37720.29]; //quello che la camera stà guardando
_camera camPreparePos [3221.69,3317.72,17.44]; //posizione della camera nel mondo
_camera camPrepareFOV 0.700; //FOV camera (campo della vista)
_camera camCommitPrepared 0 //quanto tempo si impiegherà per passare alla camera successiva
@camCommitted _camera // N.B. METTERE TALE ISTRUZIONE ALLA FINE DI OGNI CAMERA CREATA

~5 // Secondi di ritardo prima del termine dell'intro. Quanto tempo rimane fissa la camera. Se non si
vuole un ritardo eliminare tale riga

;comment "Camera 2nd Position"; // Rinominare la posizione della 2nd camera
_camera camPrepareTarget [-43733.02,-79063.80,-31500.20];
_camera camPreparePos [3191.84,3418.68,20.23];
_camera camPrepareFOV 0.700;
_camera camCommitPrepared 10
@camCommitted _camera // N.B. METTERE TALE ISTRUZIONE ALLA FINE DI OGNI CAMERA CREATA

~5 // Secondi di ritardo prima del termine dell'intro. Quanto tempo rimane fissa la camera. Se non si
vuole un ritardo eliminare tale riga

// ---- INSERIRE ALTRE CAMERE CON LO STESSO PROCEDIMENTO
```

8. Inserire eventualmente altre camere con lo stesso procedimento.

9. Per finire la Intro occorre eliminare la camera. Questo si fa mettendo le ultime due righe allo script precedente

```
;=== Create Camera Start
_camera = "camera" camcreate [3695.91,3610.04,2.00]
_camera cameraeffect ["internal","back"]
;=== Create Camera End

;=== Intro Start

;comment "Camera 1st Position"; // Rinominare la posizione della 1st camera
_camera camPrepareTarget [-71785.54,57616.60,-37720.29]; //quello che la camera stà guardando
_camera camPreparePos [3221.69,3317.72,17.44]; //posizione della camera nel mondo
_camera camPrepareFOV 0.700; //FOV camera (campo della vista)
_camera camCommitPrepared 0 //quanto tempo si impiegherà per passare alla camera successiva
@camCommitted _camera // N.B. METTERE TALE ISTRUZIONE ALLA FINE DI OGNI CAMERA CREATA

~5 // Secondi di ritardo prima del termine dell'intro. Se non si vuole un ritardo eliminare tale riga

;comment "Camera 2nd Position"; // Rinominare la posizione della 2nd camera
_camera camPrepareTarget [-43733.02,-79063.80,-31500.20];
_camera camPreparePos [3191.84,3418.68,20.23];
_camera camPrepareFOV 0.700;
```



```
_camera camCommitPrepared 10 //N.B. Se si imposta 0 si passa da una camera all'altra in modo fisso. Se si imposta un tempo si passerà alla 2nd camera in modo continuo con quel tempo. Qui per esempio sono 10 sec
@camCommitted _camera // N.B. METTERE TALE ISTRUZIONE ALLA FINE DI OGNI CAMERA CREATA

~5 // Secondi di ritardo prima del termine dell'intro. Quanto tempo rimane fissa la camera. Se non si vuole un ritardo eliminare tale riga

// ---- INSERIRE ALTRE CAMERE CON LO STESSO PROCEDIMENTO

player cameraEffect ["terminate","back"] //Termina l'effetto camera e la elimina
camDestroy _camera
```

Nota: nell'istruzione "camera camCommitPrepared 10" della seconda camera mettere un tempo diverso da 0 se si vuole passare dalla prima alla successiva in modo continuo con quel tempo. Qui per esempio sono 10 sec. Così vale anche per le camere successive

10. Riaprire l'init dell'unità e cambiare la stringa in:

```
this exec "Intro.sqs";
```

11. Salvare tutto

--ARMA 3 CAMERA CONTROLS! --

L: Removes Crosshair

Numpad *: Allows You To Move Camera Around Using Mouse

Q: Camera Up

Z: Camera Down

28.3 TERZO METODO

Questo metodo più semplice fa partire un video youtube all'inizio della missione.

NOTA: non funziona per i client con Steam Overlay disabilitato. Inoltre, il comando non può essere chiamato al di fuori di uno script init in MP.

1. Creare il file InitPlayerLocal.sqf e inserire la stringa

```
execVM "YT.sqf";
```

2. Creare il file YT.sqf e inserire la stringa

```
openYoutubeVideo "inserire qui l'indirizzo http del video Youtube";
```

All'avvio della missione si aprirà la schermata Youtube con il video richiamato.

28.4 QUARTO METODO (ATTIVABILE ANCHE CON TRIGGER IN GIOCO)

Istruzione per riprodurre un video d'intro con un effetto fade all'inizio e alla fine.

Mettere questa nel file "init.sqf" oppure nel "InitPlayerLocal.sqf"

```
2 cutText [ "", "BLACK OUT" ];
sleep 1;
_video = ["media\intro.ogv"] spawn bis_fnc_playVideo; // path and function to play your video
waitUntil {scriptDone _video};
sleep 2;
2 cutText [ "", "BLACK IN" ];
```

Dove intro.ogv è il file video posto nella cartella cartella-missione\media\

*.ogv : Ogg Vorbis Video File.

I file OGV vengono spesso utilizzati per riprodurre contenuti video di pagine Web utilizzando HTML5 < video > etichetta. Tuttavia, vengono generalmente indicati nel codice sorgente HTML usando l'estensione ".ogg", anche se i file contengono contenuti video.

Una lista di video già compilati del gioco (con specifiche cartelle) è riportata di seguito:

```
\a3\missions_f\video\helicopters.ogv
```

```
\a3\missions_f_epa\video\b_in2_12_hours_later.ogv
```



<pre>\a3\missions_f\video\infantry.ogv \a3\missions_f\video\scuba.ogv \a3\missions_f\video\supports.ogv \a3\missions_f\video\vehicles.ogv \a3\missions_f_beta\video\combined_arms.ogv \a3\missions_f_beta\video\commanding.ogv \a3\missions_f_beta\video\defend.ogv \a3\missions_f_beta\video\night.ogv \a3\missions_f_beta\video\supports.ogv \a3\missions_f_bootcamp\video\boot_m05_sometime_later.ogv \a3\missions_f_bootcamp\video\vr_boot.ogv \a3\missions_f_bootcamp\video\vr_generictransition_1.ogv \a3\missions_f_bootcamp\video\vr_generictransition_2.ogv \a3\missions_f_bootcamp\video\vr_generictransition_3.ogv \a3\missions_f_epa\video\A_hub_quotation.ogv \a3\missions_f_epa\video\A_in_intro.ogv \a3\missions_f_epa\video\A_in_quotation.ogv \a3\missions_f_epa\video\A_in2_two_hours_later.ogv \a3\missions_f_epa\video\A_m01_quotation.ogv \a3\missions_f_epa\video\A_m02_quotation.ogv \a3\missions_f_epa\video\A_m03_quotation.ogv \a3\missions_f_epa\video\A_m04_quotation.ogv \a3\missions_f_epa\video\A_m05_quotation.ogv \a3\missions_f_epa\video\A_out_quotation.ogv \a3\missions_f_epa\video\A_out_some_time_later.ogv \a3\missions_f_epa\video\A_out_to_be_continued.ogv \a3\missions_f_epa\video\B_hub01_10_days_later.ogv \a3\missions_f_epa\video\B_hub01_few_hours_later.ogv \a3\missions_f_epa\video\B_m05_quotation.ogv \a3\missions_f_epa\video\B_m05_quotation.ogv</pre>	<pre>\a3\missions_f_epa\video\B_m01_quotation.ogv \a3\missions_f_epa\video\B_m02_quotation.ogv \a3\missions_f_epa\video\B_m03_quotation.ogv \a3\missions_f_epa\video\B_m06_quotation.ogv \a3\missions_f_epa\video\B_out2_quotation.ogv \a3\missions_f_epa\video\B_out2_sometime_later.ogv \a3\missions_f_epa\video\C_ea_quotation.ogv \a3\missions_f_epa\video\C_eb_quotation.ogv \a3\missions_f_epa\video\C_in1_quotation.ogv \a3\missions_f_epa\video\C_in2_quotation.ogv \a3\missions_f_epa\video\C_m01_quotation.ogv \a3\missions_f_epa\video\C_m02_quotation.ogv \a3\missions_f_epa\video\C_out1_quotation.ogv \a3\missions_f_epa\video\C_out2_sometime_later.ogv \a3\missions_f_epa\video\fixed_wings.ogv \a3\missions_f_gamma\video\arma.ogv \a3\missions_f_gamma\video\drones.ogv \a3\missions_f_gamma\video\faction_blufor.ogv \a3\missions_f_gamma\video\faction_independent.ogv \a3\missions_f_gamma\video\faction_opfor.ogv \a3\missions_f_gamma\video\gunships.ogv \a3\missions_f_gamma\video\tanks.ogv \a3\missions_f_heli\video\slingloading.ogv \a3\missions_f_mark\video\ffv.ogv \a3\missions_f_mark\video\marksman.ogv \a3\missions_f_epa\video\B_hub01_quotation.ogv \a3\missions_f_epa\video\B_in_quotation.ogv \a3\missions_f_epa\video\A_in2_quotation.ogv \a3\missions_f_epa\video\B_m02_1_quotation.ogv \a3\missions_f_epa\video\A_in2_quotation.ogv</pre>
---	--

L'istruzione per vederli è sempre la:

```
_video = ["\a3\missions_f_mark\video\marksman.ogv"] spawn bis_fnc_playVideo;
waitUntil {scriptDone _video};
```

quest'istruzione dovrebbe funzionare anche in gioco attivabile tramite trigger. Basta metterla all'attivazione.

Alternativa

Istruzione per far partire un video in MP.

Mettere tale stringa in un sqf oppure un trigger ecc..

```
bigScreen setObjectTextureGlobal [0, "scripts \ video \ action1.ogv"];
["cartella_video\file_video.ogv", [10,10]] remoteExec ["BIS_fnc_playVideo", ([0, -2] select
isDedicated), true];
```

Oppure

```
["cartella_video\file_video.ogv", [10,10]] remoteExec ["BIS_fnc_playVideo", 0, true];
```

Esempio di intro trovato:

Here is what I do and it works well with our group. This function will include the random video clip (I only put four in), intro text, place for custom music (or Arma3 stock music), and an animated welcome message once the player is in game. create a file called **initPlayerLocal.sqf**. This file will go into the same place as your init.sqf and description.ext files. Note the playMusic command has a selected Arma track; you can search the track name on the interwebz to get the rest of them as well.

You can substitute your own custom intro music (properly formatted ogg file, sound folder, description etc.) and use playSound "trackname" instead.

```
_Video = ["\A3\Missions_F_EPA\video\A_hub_quotation.ogv",
"\A3\Missions_F_EPA\video\A_m02_quotation.ogv", "\A3\Missions_F_EPA\video\A_m03_quotation.ogv",
"\A3\Missions_F_EPA\video\A_m04_quotation.ogv"] call bis_fnc_selectRandom;
[_Video, "STR_A3_Campaign_Quote_10"] call BIS_fnc_quotations;

waitUntil {BIS_fnc_quotations_playing || !(isNil "BIS_fnc_quotations_skip")};

sleep 10;

//wait for player to exist
```




```
waitUntil { !isNull player };
waitUntil { player == player };

playMusic "LeadTrack05_F_EPC";

MissionIntro = [] spawn {
    ["BIS_blackStart", false] call BIS_fnc_blackOut;
    [[["OPERATION INTRO", "<t align = 'center' shadow = '1' size = '2.0'
font='PuristaBold'>%1</t><br/>",
    ["By Intro Text Noob", "<t align = 'center' shadow = '1' size = '0.7'>%1</t><br/>"]], 0, 0, "<t
color='#FFFFFF' align='center'>%1</t>"] spawn BIS_fnc_typeText;
    sleep 10;
    "dynamicBlur" ppEffectEnable true;
    "dynamicBlur" ppEffectAdjust [7];
    "dynamicBlur" ppEffectCommit 0;
    "dynamicBlur" ppEffectAdjust [0.0];
    "dynamicBlur" ppEffectCommit 7;
    ["BIS_blackStart", true] call BIS_fnc_blackIn;
};

waitUntil {scriptDone MissionIntro};

[] spawn {
    waitUntil{!(isNil "BIS_fnc_init")};
    // Info text
    [str("Welcome to") , str("This Intro Map."), str("Good Luck.")] spawn BIS_fnc_infoText;
};
```

28.5 QUINTO METODO

description.ext

```
class RscPicture
{
    access=0;
    type=0;
    idc=30;
    style=48;
    colorBackground[]={0,0,0,0};
    colorText[]={1,1,1,1};
    font="TahomaB";
    sizeEx=0;
    lineSpacing=0;
    text="";
};

//Macro defining a simple and standard scene picture.
//Parameters: index of the scene, path to the picture, x-coord, y-coord, width, height.
#define SCENE(INDEX, PIC, X, Y, W, H) \
class Scene##INDEX \
{ \
    idd = -1; \
    movingEnable = 0; \
    fadein = 0; \
    fadeout = 0; \
    duration = 300; \
    name = __EVAL("Scene" + #INDEX); \
    \
    class controls \
    { \
        class Overlay \
        { \
            idc = -1; \
            x = SafeZoneX; y = SafeZoneY; \
            w = SafeZoneH; h = SafeZoneH; \
            type = 0; \
            style = 48; \
            colorBackground[] = {0, 0, 0, 0}; \
            colorText[] = {1, 1, 1, 1}; \
            font = BitStream; \
            sizeEx = 0; \
            lineSpacing = 0; \
            text = PIC; \
        } \
    } \
};
```



```
    } \
};

class RscTitles
{
//      SCENE(1, "t\effect_ca.paa", -0.379, -0.135, 1.77, 1.181)
SCENE(1, "t\effect_ca.paa", 0,0,1,1)
};
```

picture.sqf

```
titleRsc ["scenel","PLAIN"];
sleep 0.3;
```

starting from mission trigger

```
bla = execVM "picture.sqf"; _video = ["filename.ogv",[-0.6,-0.5,2.2,2]] spawn bis_fnc_playVideo;
```

Never ventured deeper into this as 12mb missions are a huge nono in MP.

You'll have to play around with the description.ext to fill the screen, above code was made long ago on my old 19" screen for a small fun mission.

28.6 ALTRO

```
/* *****
Custom video tutorial for Arma 3 - additional code examples by Feuerex. Enjoy!
***** */

//In-game cutscene

    //Camera creating
    _camera = "camera" camCreate [0,0,0];
    _camera cameraEffect ["internal","back"];
    <import from clipboard.txt>

    //Cinematic borders - easier, simple
    ShowCinemaBorder true;
    ShowCinemaBorder false;

    //Animated, new method used in Arma 3
    [0] call BIS_fnc_CinemaBorder;
    //Use 0 or 1 for moving the border in and out of the screen

//Custom video
//You need: a videofile, video to OGV converter
//Convert the video to .ogv and put it into your mission folder

    play = ["videoName.ogv"] spawn BIS_fnc_PlayVideo;
    //You could eventually use other parameters, that are optional.
    //Video size - array in format [X (upper left corner, horizontal), Y (upper left corner,
vertical), W, H]
    play = ["videoName.ogv",[0,0,1,0.5]] spawn BIS_fnc_PlayVideo;
    //Video color - array in format [R,G,B,A]
    play = ["videoName.ogv",[1,1,1,1]] spawn BIS_fnc_PlayVideo;

    //If you play the video and then want to continue with more commands,
    // you can make the script wait for the end of the video
    waituntil {scriptDone play};

//Steam interface link

    //variable = OpenYoutubeVideo "videoLink";
    //The link needs to be the part of the link right after http://youtube.com/. It should always
start with watch?v=

    open = OpenYoutubeVideo "watch?v=CERTE-XaqaI";
```



```
//If you have trouble making this work,  
//try launching the code from init.sqf and make sure your Steam interface is set to  
"Enabled";
```

29 MOVIE MAKER SCRIPT

Si basa sullo script in §28.1. Scaricabile da <http://www.armaholic.com/page.php?id=31327>

30 COME RIPRODURRE UN VIDEO DURANTE IL GIOCO

Vedi § 28.4.

31 CREARE UNA BANDIERA CON LOGO PERSONALIZZATO

Per creare una bandiera personalizzata:

12. Creare la texture della bandiera personalizzata:
 - Le dimensioni devono essere 200x200 pixel
 - Il formato deve essere *.jpg
13. Mettere il file creato, es BandieraEGX.jpg, dentro la cartella della missione
14. Posizionare un'asta con bandiera in mappa
15. Inserire nella sua casella Init la stringa:

```
this setflagtexture "BandieraEGX.jpg";
```

32 CHIAMATE IN GLOBALE (REMOTEEEXEC E BIS_FNC_MP)

Esempi dell'uso della funzione RemoteExec, RemoteExecCall e la "vecchia versione" BIS_fnc_MP.

```
['Script\script_1.sqf', 'BIS_fnc_execVM', true, true]call BIS_fnc_MP
```

```
["Misscomp",true,1,false,true] remoteExecCall ["BIS_fnc_endMission",0];
```

```
[objNull, player] remoteExecCall ['ace_medical_treatment_fnc_fullHeal',0];
```

```
/* Note per far udire il suono in globale. Nota: say3D agisce in locale  
[_bomber,"shout"] remoteExecCall ["say3D",0];  
[_bomber, "shout"] remoteExec ["say3D", 0];  
playSound3D ["shout", _bomber]; // playSound3D è una funzione che di default agisce in globale
```

```
_text1 = format["<t font='TahomaB' t size='1.25' t color='#00FF00'>%1<br/></t>", name player];  
_text2 = "<t font='TahomaB' t size='1.25' t color='#00FF00'> ha preso l'abilitazione di Ingegnere</t>";  
[parseText (_text1 + _text2)] remoteExec ["hint",0,false];
```

```
[boxVA,"GEN_fnc_addVASAction",nil,true] spawn BIS_fnc_MP;  
E nella funzione addVASAction la chiamata:  
_this addaction ["ARSENALE VIRTUALE",{["Open",true] call BIS_fnc_arsenal; }];
```

```
message = format ["E' STATO APERTO L'ARSENAL SENZA PERMESSO DA %1",name player];  
//hint _message;  
["Allarme"] remoteExec ["playsound",0,false];  
[_message] remoteExec ["hint",0,false];
```

```
[[getMarkerPos "StrikePosMark",_gradiDir], "EGX_system\functions\CAS.sqf"] remoteExec  
["BIS_fnc_execVM",0,false];
```



33 CREARE UN IED

33.1 PRIMO METODO (NON DISINNECABILE)

This is what I do for IEDs, and it works for any item in game, static or dynamic. Boxes, trash, cars, people, etc. For clarity, I'm going to explain it like we're setting up a car bomb (but it works with any editor item):

1. Put down your desired IED/car or every object, and give it a name, we'll say we name it "IED".
2. Create a trigger to set off the bomb, I usually use a 10m radius. You can use radio calls also, to set an ambush.
3. In the "onAct" of the trigger:

```
bomb="M_Mo_82mm_AT_LG" createVehicle (getPos IED); deletevehicle IED;
```

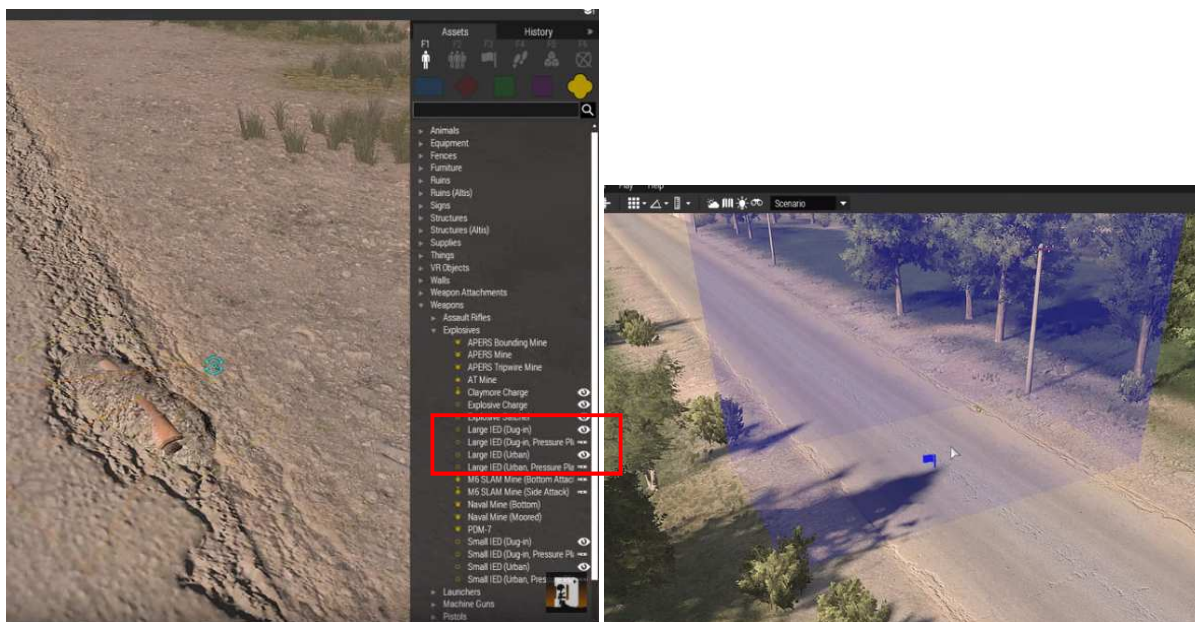
od anche

```
bomb="Bo_GBU12_LGB" createVehicle (getPos IED); deletevehicle IED;
```

Nota: l'IED creato con questo metodo NON è disinnescabile e NON può essere individuato con un rilevatore di mine.

Si può usare direttamente una mina IED tra quelle disponibili tra gli oggetti. Nominarla sempre come "IED". In tal caso bisogna farla saltare in aria mettendo nell'attivazione del trigger la stringa

```
IED setDamage 100;
```



33.2 SECONDO METODO

Metodo con uso di uno script dedicato scaricabile al <http://www.armaholic.com/page.php?id=32348>

Description:

'Simple IED Script' will spawn IEDs on / alongside random road segments within predetermined IED area markers setup by the mission designer. The position, direction, appearance, and blast magnitude of the IEDs are all random. Additionally, the script will also spawn trash randomly along the roads; trash can sometimes spawn on the position of an IED as well. The script is tiny, optimized, and compatible in all environments (SP, MP, Dedi). It doesn't require any addons or additional scripts and is compatible with any map that has roads.

Features:

- Tiny filesize! 2KB, fast performance and Easy to implement!
- Compatible with SP, MP, and Dedicated
- Works on any map with roads
- Doesn't require any mods or additional scripts
- Randomized Results:



- IED type spawned
- IED blast magnitude is random (damage / radius)
- IED position & direction
- Chance for trash to spawn with IED
- Trash appearance variety
- IEDs are proximity based:
 - If player is within 10 meters of IED AND is moving faster than a slow crawl, the IED will detonate
- IEDs can be disarmed:
 - If player is an Explosive Specialist and has a ToolKit, he can slow crawl up to the IED safely and disarm it
 - IEDs can be detected with a Mine Detector

Uso dello script:

1. Mettere il file "ied.sqf" scaricato nella cartella della missione. Nel caso crearne uno con il seguente contenuto:

```
iedMkr=["iedMkr0","iedMkr1","iedMkr2"]; //List of markers to spawn IEDs in
iedNum=5; //Number of IEDs per marker, defined in iedMkr [Default: 5]
iedDmg=true; //Can the IED be killed with weapons? [Default: false] TRUE = Yes | FALSE = Can only be
disarmed
Dbug=false; //Show IED markers on map? [Default: false]

//!!DO NOT EDIT BELOW!!
iedBlast=["Bo_Mk82","Rocket_03_HE_F","M_Mo_82mm_AT_LG","Bo_GBU12_LGB","Bo_GBU12_LGB_MI10","HelicopterExploSmall"];
iedList=["IEDLandBig_F","IEDLandSmall_F","IEDUrbanBig_F","IEDUrbanSmall_F"];
iedAmmo=["IEDUrbanSmall_Remote_Ammo","IEDLandSmall_Remote_Ammo","IEDUrbanBig_Remote_Ammo"];
iedJunk=["Land_Garbage_square3_F","Land_Garbage_square5_F","Land_Garbage_line_F"];
if(!Dbug)then{[_x setMarkerAlpha 0;}forEach iedMkr;

if(!isServer)exitWith{};
iedAct={_iedObj=this select 0;
if(mineActive _iedObj)then{
_iedBlast=selectRandom iedBlast;
createVehicle[_iedBlast,(getPosATL _iedObj),[],0,""];
createVehicle["Crater",(getPosATL _iedObj),[],0,""];
{deleteVehicle _x}forEach nearestObjects[getPosATL _iedObj,iedJunk,4];
deleteVehicle _iedObj;};

(private["_ieds","_trig"];_ieds=[];_iedArea=getMarkerSize _x select 0;_iedRoad=(getMarkerPos _x)nearRoads _iedArea;
for "i" from 1 to iedNum do{
if(count _ieds==iedNum*4)exitWith{};
_iedR=selectRandom _iedRoad;
_ied=selectRandom iedList;_junk=selectRandom iedJunk;
_ied=createMine[_ied,getPosATL _iedR,[],8];_ied setPosATL(getPosATL _ied select 2+1);_ied setDir(random 359);
if(!iedDmg)then{_ied allowDamage false;};
if(round(random 2)==1)then{_iedJunk=createVehicle[_junk,getPosATL _ied,[],0,""];_iedJunk setPosATL(getPosATL _iedJunk select
2+1);_iedJunk enableSimulationGlobal false;};
_jnkR=selectRandom _iedRoad;_junk=createVehicle[_junk,getPosATL _jnkR,[],8,""];_junk setPosATL(getPosATL _junk select 2+1);
_junk enableSimulationGlobal false;
_trig=createTrigger["EmptyDetector",getPosATL _ied];
_trig setTriggerArea[10,10,0,FALSE,10];
_trig setTriggerActivation["ANY","PRESENT",false];
_trig setTriggerTimeout[1,1,1,true];
if(isMultiplayer)then{
_trig setTriggerStatements[
"({vehicle _x in thisList && speed vehicle _x>4}count playableUnits>0",
"({if((typeof _x)in iedAmmo)then{[_x]call iedAct;}}forEach nearestObjects[thisTrigger,[],10];",
"deleteVehicle thisTrigger"];}else{
_trig setTriggerStatements[
"({vehicle _x in thisList && isPlayer vehicle _x && speed vehicle _x>4}count allUnits>0",
"({if((typeof _x)in iedAmmo)then{[_x]call iedAct;}}forEach nearestObjects[thisTrigger,[],10];",
"deleteVehicle thisTrigger"];};

_ieds pushBack _ied;
if(Dbug)then{
iedMkrs=[];
_iedPos=getPosWorld _ied;
_mkrID=format["m %1",_iedPos];
_mkr=createMarker[_mkrID,getPosWorld _ied];
_mkr setMarkerShape"ICON";_mkr setMarkerType"mil_dot";_mkr setMarkerBrush"Solid";_mkr setMarkerAlpha 1;_mkr
setMarkerColor"ColorEast";
iedMkrs pushBack _mkr;};
};
}forEach iedMkr;
sleep 5;
(CIVILIAN revealMine _x;EAST revealMine _x;}forEach allMines;
```

2. Creare il file "init.sqf" nella cartella missione e copiarci dentro la stringa:

```
execVM"ied.sqf";
```

3. Creare un marker d'area quadrato in mappa e nominarlo (es: "myMkr0")

N.B.: Assicurarsi che sia un quadrato quindi con dimensioni dei lati A e B uguali! Esempio: 500x500

4. Nel file "ied.sqf", aggiungere i nomi dei marker delle aree IED nella stringa alla linea 1. Per esempio:

```
iedMkr=["myMkr0","myMkr1","myMkr2"];
```



qui sono aggiunte tre aree

5. Sempre nel file "ied.sqf", si puo' cambiare il numero di IED da spawnare per ogni marker sulla stringa alla linea 2. Per esempio:

```
iedNum=8; //Will spawn 8 IEDs per IED marker
```

6. Cambiare le stringe iedDmg=false; o Dbug=true; a seconda delle necessità (vedi i commenti relativi per la spiegazione)

33.3 TERZO METODO

Scaricabile al <http://www.armaholic.com/page.php?id=26865>

Description:

IED/Bomb disarming script which gameplay is same as popular and cool minesweeper.

The script provides unique functionality on disarming bombs in Arma. It's not random number generator which decided whether to blow up bombs in front of you or rescue you for god's sake. Now everything depends on your skills! To disarm bomb you need to complete minesweeper game. It has the same rules as normal minesweeper with little exceptions:

- Firstly, it always have 9x9 grid. There can be 8, 10 or 12 bombs. It depends on difficulty.
- Secondly, if you are Explosive Specialist you are able to determine difficulty of the bomb before disarming (green - easiest, red - hardest) and you have some fields revealed from the start!

Installation / Usage:

1. Put things from IED folder inside mission folder (IED folder, init.sqf, description.ext). If you already have init.sqf or description.ext then do not overwrite them.
2. If you already had init.sqf put

```
call compile preprocessFile "IED\init.sqf";
```

inside the script

3. If you already had description.ext put below code inside it:

```
#include "IED\defines.hpp"  
#include "IED\minesweeper.hpp"
```

4. Put desired IED/Bomb object on map and put this code in initialization field:

```
nic = [this, "easy", west] execVM "IED\init_ied.sqf";
```

This will create easy bomb that will react only on west's units.

Possible difficulties:

- "easy"
- "medium"
- "hard"

Possible sides:

- west
- east
- resistance
- civilian
- "west"
- "east"
- "civilian"
- "everyone"

NOTE: If you want to have explosive specialist who are better with disarming bomb put this code in their initialization field

```
this setvariable ["ShoterIed_Specialist", true, true];
```



33.4 QUARTO METODO

Scaricabile al <http://www.armaholic.com/page.php?id=23995>

Adding the script to your mission file is pretty simple, all you need to do is put the folder into your mission folder and add the following to your init.sqf

```
[ ] spawn {call compile preprocessFileLineNumbers "EPD\Ied_Init.sqf";};
```

And copy the following into your description.ext

```
class CfgFunctions {  
    #include "EPD\EPDcfgFunctions.hpp"  
};
```

If you want to change where or how many IEDs are spawned when the mission starts, modify the variable called `iedInitialArray` in `Ied_Settings.sqf`.

There are several ways to define where you want the IEDs placed.

```
//These are the actual IEDs that will spawn. Add them using one of the following formats.  
//mapLocations must have their type defined as one of "NameCityCapital","NameCity","NameVillage",  
"NameLocal"  
//[ "All", side]  
//[ "AllCities", side]  
//[ "AllVillages", side]  
//[ "AllLocals", side]  
//[ "mapLocation", side]  
//[ "mapLocation", amountToPlace, side];  
//[ "mapLocation", iedsToPlace, fakesToPlace, side]  
//[ "mapLocation", amountToPlace, [fakeChance, smallIedChance, mediumIedChance, largeIedChance], side]  
//[ "predefinedLocation", side]  
//[ "predefinedLocation", amountToPlace, side];  
//[ "predefinedLocation", iedsToPlace, fakesToPlace, side]  
//[ "predefinedLocation", amountToPlace, [fakeChance, smallIedChance, mediumIedChance, largeIedChance],  
side]  
/*****Marker size > 1*****/  
//[ "marker", amountToPlace, [fakeChance, smallIedChance, mediumIedChance, largeIedChance], side]  
//[ "marker", iedsToPlace, fakesToPlace, side]  
//[ "marker", amountToPlace, side]  
//[ "marker", side]  
/*****Marker size = 1*****/  
//[ "marker", side]  
//[ "marker", chanceToBeReal, side]  
//[ "marker", [fakeChance, smallIedChance, mediumIedChance, largeIedChance] , side]  
  
//The side can be a single side, or an array of sides  
//Ex. "West" or ["West","East"]  
//http://community.bistudio.com/wiki/side
```

The way it works is that it will use the `markerName` as the center and find all the roads within the radius of the marker (make sure you set it!), then it will randomly place real and fake IEDs somewhere within it.

There are several predefined locations for Altis in `EPD\Ied_Settings.sqf` that you can use if you don't feel like making your own markers. As of version 1.3, you can also use the names of locations in the game. If you want to share your predefined locations for other maps for others to use, feel free to, and I will post them here on the front post.

Here is a map of where each `AltisRandom` corresponds to.

If you do not specify how many to place, it will calculate an amount based on the size of the marker you gave it. If the marker has a size of 1, it will place exactly 1 IED in that exact spot, allowing you to pick where the IED is at, rather than a random road near the marker. There is an example of each type in `EPD\Ied_Settings.sqf`.

There are currently 4 types of IEDs

- Secondary IED - Designed to kill first responders after the other IEDs goes off.
- Infantry IED - Will kill the person who sets it off and injure most other squad members near him
- Light Armor IED - Will destroy hunter type vehicles and usually disable marshall type vehicles.



- Heavy Armor IED - Will destroy all vehicles and cause lots of mayhem

The type of IED also determines what size the object it is hiding in is. Heavy Armor IEDs will be hidden in things like trash piles and wrecked vehicles. Infantry IEDs will be hidden in things like buckets and tires.

IEDs are set off based on speed and proximity. If your horizontal velocity squared is higher than 2.8, and you are within 11 meters of it, you will set it off.

The IEDs can be disarmed if you have the appropriate items and are within 3 meters. If you fail while trying to disarm an IED, you will set it off.

Settings that can be changes as of version 2.0

- EPD_IED_debug - If this is set to true, it will create map markers indicating where the IEDs and fakes are. It will create a message showing if a player is near an IED and give their distance and speed (see picture 2). It will also create a message if there are secondaries incoming. These will only show up to the person who is the server. This is mostly useful for testing while developing a map.
- hideIedSectionMarkers - Set this to true if you want it to hide the marker you used to position the IED circle.
- hideSafeZoneMarkers - Set this to true if you want it to hide the marker you used to position safezones.
- itemsRequiredToDisarm - A player must have all of these items in order to have a disarm option.
- betterDisarmers - Players of these classes have an increased chance of disarming IEDs
- baseDisarmChance - Default chance of successfully disarming an IED
- bonusDisarmChance - Players who are in the betterDisarmers array will have this amount added to their baseDisarmChance
- secondaryChance - Chance that secondary explosions will spawn
- smallChance - Chance that an IED will be small sized
- mediumChance - Chance that an IED will be medium sized
- largeChance - Chance that an IED will be large sized
- iedSecondaryItems - Items secondary IEDs will hide in
- iedSmallItems - Items Infantry IEDs will hide in
- iedMediumItems - Items Light Armor IEDs will hide in
- iedLargeItems - Items Heavy Armor IEDs will hide in
- predefinedLocations - If you want to use the same locations over lots of missions without having to create markers over and over, define them here using the following format ["Name",[LocationX,LocationY,LocationZ],size]. Most of Altis has been provided as an example.
- allowExplosiveToTriggerIEDs - If this is set to true explosions can set off the IEDs.
- iedInitialArray - This is where you actually pick where the IEDs are spawned and how many when the mission starts. See the code above for the format
- iedSafeZones - Place the mapLocations, predefinedLocations, and markerNames of places you don't want any random IEDs spawning

Explosives Requirements

- Planted explosives and Bombs have a 100% chance of setting off IEDs if they are within 6 meters
- Hand grenades have a 35% chance of setting off IEDs if they are within 6 meters
- Rockets, Missiles, Shells, and Submunitions have a 100% chance of setting off IEDS if they are a direct hit
- Launched grenades have a 50% chance of setting off IEDs if they are a direct hit
- Explosive bullets have a 40% chance of setting off IEDs if they are a direct hit

Creating IEDs after the mission has started

Version 2.0 introduced the ability to create IEDs in the middle of a mission. Create an array similar to how you did for the iedInitialArray then call the CREATE_IED_SECTION function. For example,

```
[["OreoKastro", "West"]] call CREATE_IED_SECTION;
```

This will return a randomly generated string of the new section created, hold onto this if you plan on doing any other functions with this section.

OR



```
["sectionName", ["OreoKastro", "West"]] call CREATE_IED_SECTION;
```

This will return the sectionName string you passed in, hold onto this if you plan on doing any other functions with this section.

Checking the status of a section of IEDs

If you want to know how many IEDs have been blown up or disarmed in a section, use the GET_IED_SECTION_INFORMATION function.

```
"sectionName" call GET_IED_SECTION_INFORMATION;  
//Returns [#iedsExploded, #iedsDisarmed]
```

If you want to know how many IEDs remain in a section, use the GET_REMAINING_IED_COUNT function.

```
"sectionName" call GET_REMAINING_IED_COUNT;
```

Removing sections of IEDs

If you want to remove a section of IEDs, use the REMOVE_IED_SECTION function.

```
"sectionName" call REMOVE_IED_SECTION;
```

Getting the last IED explosion location

If you would like to set up ambushes and the like, you can check the value of lastIedExplosion to get the location of the last IED to go off. Then have your units react appropriately.

34 CREARE UN EFFETTO TORMENTA

34.1 TORMENTA DI NEVE

Per creare l'effetto di una tempesta di neve si può utilizzare uno script scaricabile al link

<http://www.armaholic.com/page.php?id=31841>

1. Mettere le cartelle scaricate "AL_snowstorm" e "sound" nella cartella missione
2. Creare o aprire il file description.ext all'interno della cartella missione ed aggiungere le classnames dei suoni

```
class CfgSounds  
{  
    class bcg_wind  
    {  
        name = "bcg_wind";  
        sound[] = {"\sound\bcg_wind.ogg", 0.3, 1};  
        titles[] = {1, ""};  
    };  
    class bcg_wind_mild  
    {  
        name = "bcg_wind_mild";  
        sound[] = {"\sound\bcg_wind_mild.ogg", 0.05, 1};  
        titles[] = {1, ""};  
    };  
    class rafala_1  
    {  
        name = "rafala_1";  
        sound[] = {"\sound\rafala_1.ogg", 1, 1};  
        titles[] = {1, ""};  
    };  
    class rafala_2  
    {  
        name = "rafala_2";  
        sound[] = {"\sound\rafala_2.ogg", 1, 1};  
        titles[] = {1, ""};  
    };  
    class rafala_3  
    {  
        name = "rafala_3";  
        sound[] = {"\sound\rafala_3.ogg", 1, 1};  
        titles[] = {1, ""};  
    };  
};
```



```
class rafala_6
{
    name = "rafala_6";
    sound[] = {"\sound\rafala_6.ogg",1,1};
    titles[] = {1, ""};
};
class rafala_7
{
    name = "rafala_7";
    sound[] = {"\sound\rafala_7.ogg",1,1};
    titles[] = {1, ""};
};
class rafala_8
{
    name = "rafala_8";
    sound[] = {"\sound\rafala_8.ogg",1,1};
    titles[] = {1, ""};
};
class rafala_5
{
    name = "rafala_9";
    sound[] = {"\sound\rafala_9.ogg",1,1};
    titles[] = {1, ""};
};
class tree_crack_1
{
    name = "tree_crack_1";
    sound[] = {"\sound\tree_crack_1.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_2
{
    name = "tree_crack_2";
    sound[] = {"\sound\tree_crack_2.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_3
{
    name = "tree_crack_3";
    sound[] = {"\sound\tree_crack_3.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_4
{
    name = "tree_crack_4";
    sound[] = {"\sound\tree_crack_4.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_5
{
    name = "tree_crack_5";
    sound[] = {"\sound\tree_crack_5.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_6
{
    name = "tree_crack_6";
    sound[] = {"\sound\tree_crack_6.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_7
{
    name = "tree_crack_7";
    sound[] = {"\sound\tree_crack_7.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_8
{
    name = "tree_crack_8";
    sound[] = {"\sound\tree_crack_8.ogg", db+5, 1};
    titles[] = {1, ""};
};
class tree_crack_9
{
    name = "tree_crack_9";
```



```
        sound[] = {"\sound\tree_crack_9.ogg", db+5, 1};
        titles[] = {1, ""};
};
class lup_01
{
    name = "lup_01";
    sound[] = {"\sound\lup_01.ogg", 1, 1};
    titles[] = {};
};
class lup_02
{
    name = "lup_02";
    sound[] = {"\sound\lup_02.ogg", 1, 1};
    titles[] = {};
};
class lup_03
{
    name = "lup_03";
    sound[] = {"\sound\lup_03.ogg", 1, 1};
    titles[] = {};
};
class tremurat_1
{
    name = "tremurat_1";
    sound[] = {"\sound\tremurat_1.ogg", 1, 1};
    titles[] = {1, ""};
};
class tremurat_2
{
    name = "tremurat_2";
    sound[] = {"\sound\tremurat_2.ogg", 1, 1};
    titles[] = {1, ""};
};
class tremurat_3
{
    name = "tremurat_3";
    sound[] = {"\sound\tremurat_3.ogg", 1, 1};
    titles[] = {1, ""};
};
class tremurat_4
{
    name = "tremurat_4";
    sound[] = {"\sound\tremurat_4.ogg", 1, 1};
    titles[] = {1, ""};
};
class NoSound
{
    name = "NoSound";
    sound[] = {"", 0, 1};
    titles[] = {0, ""};
};
class tuse_1
{
    name = "tuse_1";
    sound[] = {"\sound\tuse_1.ogg", 1, 1};
    titles[] = {1, ""};
};
class tuse_2
{
    name = "tuse_2";
    sound[] = {"\sound\tuse_2.ogg", 1, 1};
    titles[] = {1, ""};
};
class tuse_3
{
    name = "tuse_3";
    sound[] = {"\sound\tuse_3.ogg", 1, 1};
    titles[] = {1, ""};
};
class tuse_4
{
    name = "tuse_4";
    sound[] = {"\sound\tuse_4.ogg", 1, 1};
    titles[] = {1, ""};
};
};
```



```

class tuse_5
{
    name = "tuse_5";
    sound[] = {"\sound\tuse_5.ogg", 1, 1};
    titles[] = {1, ""};
};
class tuse_6
{
    name = "tuse_6";
    sound[] = {"\sound\tuse_6.ogg", 1, 1};
    titles[] = {1, ""};
};
};

```

3. Creare un trigger o attivazione qualsiasi ed inserire la stringa nel suo campo attivazione

```

null =
["_snowfall", "_duration_storm", "_ambient_sounds_al", "_breath_vapors", "_snow_burst", "_effect_on_objects",
 "_vanilla_fog", "_no_snow_indoor", "_local_fog", "_intensifywind", "_unitsneeze"] execvm
"AL_snowstorm\al_snow.sqf";

```

Questa creerà l'effetto della tempesta desiderato.

I campi delle variabili “_xxx” da cambiare sono descritti come di seguito a seconda delle esigenze:

```

=====
>>>> SNOW STORM Parameters =====
=====
null =
["_snowfall", "_duration_storm", "_ambient_sounds_al", "_breath_vapors", "_snow_burst", "_effect_on_objects", "_vanilla_fog", "_no_snow_indoor",
 "_local_fog", "_intensifywind", "_unitsneeze"] execvm "AL_snowstorm\al_snow.sqf";

snowfall           - boolean, if true snowflakes made out of particles will be created
duration_storm     - number, life time of the SNOW STORM expressed in seconds
ambient_sounds_al - seconds/number, a random number will be generated based on your input value and used to set the frequency for played
ambient sounds
                    - if is negative NO custom ambient sounds will be played
breath_vapors      - boolean, if true you will see breath vapors for all units, however if you have many units in your mission you should
set this false to diminish the impact on frames
snow_burst         - seconds/number, if higher than 0 burst of snow will be generated at intervals based on your value
effect_on_objects - boolean, if is true occasionally a random object will be pushed by the wind during the snow burst if the later is
enabled
vanilla_fog        - boolean, vanilla fog will be managed by the script if true, otherwise the values you set in editor will be used
local_fog          - boolean, if true particles will be used to create sort of waves of fog and snow
intensifywind      - boolean, if is true the wind will blow with force otherwise default value from Eden or other script will be used
unitsneeze         - boolean, if is true the at random units will sneeze/caugh and will shiver when snow burst occurs
*/
>>>>>>>> EXAMPLE
null = [true,3000,15,true,5,true,true,false,false,true] execvm "AL_snowstorm\al_snow.sqf";

```

od anche attivandolo da console con la stringa:

```

null = [true,3000,15,true,5,true,true,false,false,true] execvm "AL_snowstorm\al_snow.sqf";

```

Per terminarlo (da testare)
impostarlo nel trigger come

```

_script = [true,3000,15,true,5,false,true,false,false,true] execvm "AL_snowstorm\al_snow.sqf";

```

poi da console

```

terminate _script;

```

```

_script = [] execVM "script.sqf";
sleep 5;
terminate _script;

```

34.2 TORMENTA DI POLVERE O SABBIA

Per creare l'effetto di una tempesta di polvere si può utilizzare uno script scaricabile al link
<http://www.armaholic.com/page.php?id=31939>

1. Mettere le cartelle scaricate “AL_dust_storm” e “sound” nella cartella missione



```
null = [340,600,true,true,false,0.3] execvm "AL_dust_storm\al_duststorm.sqf";

// wind effect on objects surrounding players ON, wall of dust ON, wall of dust destructive
null = [340,600,true,true,true,0.3] execvm "AL_dust_storm\al_duststorm.sqf";

// Example for storm coming and going randomly
// place lines below at the end of init.sqf
// edit the value of my_dust_storm_duration as desired

while {true} do {
if (isServer) then {
my_dust_storm_duration = 240 + random 600;
publicVariable "my_dust_storm_duration";
pause_between_dust_storm = 240 + random 600;
publicVariable "my_dust_storm_duration";
};
waitUntil (!(isNil "my_dust_storm_duration") and !(isNil "pause_between_dust_storm"));
null = [340,my_dust_storm_duration,false,false,false,0.3] execvm "AL_dust_storm\al_duststorm.sqf";
sleep (my_dust_storm_duration + pause_between_dust_storm);
};
```

34.3 TEMPESTA DI MONSONI

Per creare l'effetto di una tempesta di monsoni si può utilizzare uno script scaricabile al link

<http://www.armaholic.com/page.php?id=31906>

4. Mettere le cartelle scaricate "AL_monsoon" e "sound" nella cartella missione
5. Creare o aprire il file description.ext all'interno della cartella missione ed aggiungere le classnames dei suoni

```
class CfgSounds
{
class bcg_wind
{
name = "bcg_wind";
sound[] = {"\sound\bcg_wind.ogg", db+5, 1};
titles[] = {1, ""};
};
class rafala_1
{
name = "rafala_1";
sound[] = {"\sound\rafala_1.ogg", db+10, 1};
titles[] = {1, ""};
};
class rafala_2
{
name = "rafala_2";
sound[] = {"\sound\rafala_2.ogg", db+5, 1};
titles[] = {1, ""};
};
class rafala_4_dr
{
name = "rafala_4_dr";
sound[] = {"\sound\rafala_4_dr.ogg", db+10, 1};
titles[] = {1, ""};
};
class rafala_5_st
{
name = "rafala_5_st";
sound[] = {"\sound\rafala_5_st.ogg", db+10, 1};
titles[] = {1, ""};
};
class rafala_6
{
name = "rafala_6";
sound[] = {"\sound\rafala_6.ogg", db+15, 1};
titles[] = {1, ""};
};
class rafala_7
{
name = "rafala_7";
sound[] = {"\sound\rafala_7.ogg", db+10, 1};
titles[] = {1, ""};
};
class rafala_8
{
name = "rafala_8";
sound[] = {"\sound\rafala_8.ogg", db+15, 1};
titles[] = {1, ""};
};
class rafala_9
{
name = "rafala_9";
sound[] = {"\sound\rafala_9.ogg", db+10, 1};
titles[] = {1, ""};
};
class 01_far
{
name = "01_far"; // Name for mission editor
sound[] = {\Sound\01_far.ogg, 1, 1.0};
titles[] = {0, ""};
};
class 02_far
{
name = "02_far"; // Name for mission editor
sound[] = {\Sound\02_far.ogg, 1, 1.0};
titles[] = {0, ""};
};
class 03_far
{
name = "03_far"; // Name for mission editor
sound[] = {\Sound\03_far.ogg, 1, 1.0};
titles[] = {0, ""};
};
};
```



```
};
class 04_far
{
    name = "04_far"; // Name for mission editor
    sound[] = {\Sound\04_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 05_far
{
    name = "05_far"; // Name for mission editor
    sound[] = {\Sound\05_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 06_far
{
    name = "06_far"; // Name for mission editor
    sound[] = {\Sound\06_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 07_far
{
    name = "07_far"; // Name for mission editor
    sound[] = {\Sound\07_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 08_far
{
    name = "08_far"; // Name for mission editor
    sound[] = {\Sound\08_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 09_far
{
    name = "09_far"; // Name for mission editor
    sound[] = {\Sound\09_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 10_far
{
    name = "10_far"; // Name for mission editor
    sound[] = {\Sound\10_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 11_far
{
    name = "11_far"; // Name for mission editor
    sound[] = {\Sound\11_far.ogg,1, 1.0};
    titles[] = {0, ""};
};
class 12_far
{
    name = "12_far"; // Name for mission editor
    sound[] = {\Sound\12_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 13_far
{
    name = "13_far"; // Name for mission editor
    sound[] = {\Sound\13_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 14_far
{
    name = "14_far"; // Name for mission editor
    sound[] = {\Sound\14_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 16_far
{
    name = "16_far"; // Name for mission editor
    sound[] = {\Sound\16_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 17_far
{
    name = "17_far"; // Name for mission editor
    sound[] = {\Sound\17_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 18_far
{
    name = "18_far"; // Name for mission editor
    sound[] = {\Sound\18_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 19_far
{
    name = "19_far"; // Name for mission editor
    sound[] = {\Sound\19_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 20_far
{
    name = "20_far"; // Name for mission editor
    sound[] = {\Sound\20_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 21_far
{
    name = "21_far"; // Name for mission editor
    sound[] = {\Sound\21_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 22_far
{
    name = "22_far"; // Name for mission editor
    sound[] = {\Sound\22_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 23_far
{
    name = "23_far"; // Name for mission editor
    sound[] = {\Sound\23_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 24_far
{
    name = "24_far"; // Name for mission editor
    sound[] = {\Sound\24_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
```



```
};
class 25_far
{
    name = "25_far"; // Name for mission editor
    sound[] = {\Sound\25_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 26_far
{
    name = "26_far"; // Name for mission editor
    sound[] = {\Sound\26_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
class 27_far
{
    name = "27_far"; // Name for mission editor
    sound[] = {\Sound\27_far.ogg, 1, 1.0};
    titles[] = {0, ""};
};
};
```

6. Creare un trigger o attivazione qualsiasi ed inserire la stringa nel suo campo attivazione

```
null = [direction_monsoon, duration_monsoon,
effect_on_objects,debris,fog_rain,rain_drops,thunder,delay_thunder] execvm "AL_monsoon\al_monsoon.sqf";
```

Questa creerà l'effetto della tempesta desiderato.

I campi delle variabili da cambiare sono descritti come di seguito a seconda delle esigenze:

```
=====
>>>> MONSOON Parameters =====
=====

null = [direction_monsoon, duration_monsoon, effect_on_objects,debris,fog_rain,rain_drops,thunder,delay_thunder]
execvm "AL_monsoon\al_monsoon.sqf";

direction_monsoon - integer, from 0 to 360, direction towards the wind blows expressed in compass degrees
duration_monsoon - integer, life time of the monsoon expressed in seconds
effect_on_objects - boolean, if is true occasionally a random object will be thrown in the air

// >>>> new parameters
debris - boolean, make it false if you dont want branches and stuff flying around
fog_rain - boolean, if is true fog around players will be generated out of particles
rain_drops - boolean, if is false not rain drops will be generated
thunder - boolean, if is true you will hear thunders and see lights otherwise only vanilla thunder will
be present... if ever
delay_thunder - number, based on this number a delay will be generated between thunders
```

35 CREARE (SPAWNARE) UN VEICOLO IN UNA DATA POSIZIONE

Per creare un veicolo in una data posizione:

1. Inserire un marker vuoto nella posizione voluta e dargli un nome, es. "markheli". Può essere anche un helipad invisibile
2. Mettere un trigger con una condizione qualsiasi e alla sua attivazione mettere la stringa

```
_veh = createVehicle ["B_Heli_Light_01_F", getMarkerPos "mrkheli"]; // crea un little bird nella
posizione del marker
Sleep 3; // da un tempo di 3 sec all'avvio dell'istruzione successiva
_veh engineOn true; // accende automaticamente il veicolo una volta creato (opzionale)
```

La classname del veicolo va cercata nell'editor una volta selezionato compare in sovrainpressione.

Se il trigger è viene ripetuto più volte l'oggetto verrà respawnato vicino ai precedenti.

Funzione *createVehicle*

Description:

Creates an empty object of given classname type.

For objects of type "Logic" use [createUnit](#) instead.

To avoid vehicle randomisation in Arma 3, set the *BIS_enableRandomization* variable immediately after creating the vehicle:

```
private _vehicle = "C_Offroad_01_F" createVehicle position player;
_vehicle setVariable ["BIS_enableRandomization", false];
```

Check [Vehicle Customization \(VhC\)](#) page for more information and settings.

Syntax:

```
type createVehicle position
```


**Parameters:**

type: [String](#) - vehicle/object className

position: [Position](#) - Desired placement position. If the exact position is occupied, nearest empty position is used.

Return Value:

[Object](#)

Alternative Syntax**Syntax:**

createVehicle [type, position, markers, placement, special]

Parameters:

[type, position, markers, placement, special]: [Array](#)

type: [String](#) - vehicle/object className

position: [PositionATL](#) ([PositionAGL](#) if boat or amphibious), [Position2D](#), or [Object](#) - desired placement position

markers: [Array](#) - (Optional, default []) if the markers array contains any markers, the position is randomly picked from array of given markers plus desired placement position. If any of the markers were given z coordinate with [setMarkerPos](#), the vehicle will also be created at given z coordinate.

placement: [Number](#) - (Optional, default 0) the vehicle is placed inside a circle with given position as center and placement as its radius.

special): [String](#) - (Optional, default "NONE") can be one of the following:

- "NONE" - will look for suitable empty position near given position (subject to other placement params) before placing vehicle there.
- "CAN_COLLIDE" - places vehicle at given position (subject to other placement params), without checking if others objects can cross its 3D model.
- "FLY" - if vehicle is capable of flying and has crew, it will be made airborne at default height.

If *special* is "" or not specified, default "NONE" is used.

Return Value:

[Object](#) - created vehicle or [objNull](#) if failed

Examples**Example 1:**

```
_jeep = "Jeep" createVehicle position player;
```

Example 2:

```
_heli = "AH1Z" createVehicle getMarkerPos "hspawn";
```

Example 3:

```
_veh = createVehicle ["ahlw", position player, [], 0, "FLY"];
```

Example 4:

```
_veh = createVehicle ["2S6M_Tunguska", getMarkerPos "marker1", ["marker2", "marker3"], 0, "NONE"];
```

Example 5:

Objects such as

- "test_EmptyObjectForBubbles"
- "test_EmptyObjectForFireBig"
- "test_EmptyObjectForSmoke"

create additional emitters, which are stored in "effects" variable on the object. Since Arma 3 v1.71 these emitters are automatically deleted when object is deleted

```
[ ] spawn  
{  
    private _fire = "test_EmptyObjectForFireBig" createVehicle position player;  
    sleep 5;  
    deleteVehicle _fire;  
};
```

Example 6:

The following explosives (ending with *_Scripted*) can be set off by applying [setDamage](#) 1 to them for ease of scripting:

- "DemoCharge_Remote_Ammo_Scripted"
- "SatchelCharge_Remote_Ammo_Scripted"
- "ClaymoreDirectionalMine_Remote_Ammo_Scripted"



```
_claymore = "ClaymoreDirectionalMine_Remote_Ammo_Scripted" createVehicle position player;  
_claymore spawn  
{  
    sleep 5;  
    _this setDamage 1;  
};
```

Example 7:

Add inventory to objects without inventory:

```
_boxes = "Land_Pallet_MilBoxes_F" createVehicle position player;  
_cargo = "Supply500" createVehicle [0,0,0];  
_cargo attachTo [_boxes, [0,0,0.85]];  
  
// optional for objects that can take damage  
_boxes addEventHandler ["Killed",  
{  
    {  
        detach _x,  
        deleteVehicle _x;  
    }  
    forEach attachedObjects (_this select 0);  
}];
```

Example 8:

Drop player's weapon:

```
_wh = "GroundWeaponHolder_Scripted" createVehicle position player;  
player action ["DropWeapon", _wh, currentWeapon player];
```

Example 9:

The following weapon holders (ending with *_Scripted*) do **not** auto-delete when empty. It is up to the mission maker to take care of these:

- "GroundWeaponHolder_Scripted"
- "WeaponHolderSimulated_Scripted"
- "Weapon_Empty" (a special weaponholder that displays only a single weapon, even if it contains magazines for this weapon)

```
private _weaponHolder = createVehicle ["Weapon_Empty", getPosATL player, [], 0, "CAN_COLLIDE"];  
_weaponHolder addWeaponCargo "arifle_Katiba_F";  
hint "You have 5 seconds to grab this weapon";  
sleep 5;  
deleteVehicle _weaponHolder;
```

35.1 CREARE UN CARTELLO PER SPAWNARE VEICOLI

Si può applicare il comando precedente **createVehicle** per creare dei veicoli in una data posizione tramite il comando assegnato ad un oggetto, quale per esempio un cartello. Avvicinandosi al cartello si selezionerà il comando tramite la rotella del mouse.

Per rendere la cosa più pulita usiamo dei file .sqf esterni.

Per creare un veicolo in una data posizione tramite cartello:

1. Inserire un marker vuoto nella posizione voluta e dargli un nome, es. "markheli". Può essere anche un helipad invisibile
2. Mettere un cartello (o qualsiasi cosa) ed inserire la stringa nel suo init:

```
_this = [this] execVM "menuVeicoli.sqf";
```

Il this attacca l'azione al cartello per eseguire (execVM) il file sqf

3. Creare il file sqf richiamato "menuVeicoli.sqf" e metterlo nella cartella missione. In esso inserire la stringa

```
// menu veicoli  
_cartello = _this select 0; // crea una variabile locale e specifica che chi esegue il comando sul  
cartello è colui stesso che lo esegue  
_cartello addAction ["Elicotteri",""]; // serve per creare una riga d'intestazione senza dargli però  
nessuna azione  
_cartello addAction ["Little bird non armato","veicoli\littleBird.sqf"]; // richiama l'sqf del veicolo
```

La prima riga serve per specificare che chi sta eseguendo il comando sul cartello è colui che lo esegue.

4. Creare un file "littleBird.sqf" nella cartella "missione\veicoli\" (crearla se assente). Inserire la stringa seguente nel file littleBird.sqf



```
_veh = createVehicle ["B_Heli_Light_01_F", getMarkerPos "mrkheli"]; // crea un little bird nella
posizione del marker
Sleep 3; // da un tempo di 3 sec all'avvio dell'istruzione successiva (opzionale)
_veh engineOn true; // accende automaticamente il veicolo una volta creato (opzionale)
```

Questa crea l'eli selezionato.

In questo modo avvicinandosi al cartello si selezionerà il comando di creazione tramite la rotella del mouse.

Se si volessero più veicoli diversi basta aggiungere markers e nuove classi di veicoli.

Per esempio:

1. Creare e nominare un marker per gli elicotteri ("mrkheli") e uno per i veicoli di terra ("mkrterra")
2. Nell'init del cartello (o oggetto)

```
_this = [this] execVM "menuVeicoli.sqf";
```

3. File menuVeicoli.sqf nella cartella missione\

```
// menu veicoli
_cartello = _this select 0;
//
_cartello addAction ["ELICOTTERI",""];
_cartello addAction ["Little bird non armato","veicoli\littleBird.sqf"];
_cartello addAction ["BlackHooke armato","veicoli\BlackHooke.sqf"];
//
_cartello addAction [","", ""]; // aggiunge una riga vuota
//
_cartello addAction ["VEICOLI DI TERRA",""];
_cartello addAction ["Carro armato Abram","veicoli\Abram.sqf"];
_cartello addAction ["Trasporto truppe MRap","veicoli\MRap.sqf"];
//
```

4. Creare i file nella cartella "missione\veicoli"

littleBird.sqf

```
_veh = createVehicle ["B_Heli_Light_01_F", getMarkerPos "mrkheli"]; // crea un little bird nella
posizione del marker mrkheli
```

BlackHooke.sqf

```
_veh = createVehicle ["inserire la classname specifica del BlackHooke", getMarkerPos "mrkheli"]; //
crea un BlackHooke nella posizione del marker mrkheli
```

Abram.sqf

```
_veh = createVehicle ["inserire la classname specifica del carro Abram", getMarkerPos "mrkheli"]; //
crea un Abram nella posizione del marker mkrterra
```

MRap.sqf

```
_veh = createVehicle ["inserire la classname specifica dell'MRap", getMarkerPos "mrkheli"]; // crea un
MRap nella posizione del marker mkrterra
```

Le classnames dei veicoli si trovano nell'editor una volta selezionato il veicolo interessato.

E così via per inserire altri veicoli.

Di seguito una possibile forma colorata del menu del menuVeicoli.sqf

```
// menu veicoli
_cartello = _this select 0;
//
_cartello addAction ["<t color=#00ffff>-----SPAWN VEICOLI-----</t>",""];
_cartello addAction ["<t color=#ffff66>ELICOTTERI</t>",""];
_cartello addAction ["Little bird non armato","veicoli\littleBird.sqf"];
_cartello addAction ["BlackHooke armato","veicoli\BlackHooke.sqf"];
//
//_cartello addAction [","", ""]; // aggiunge una riga vuota
//
_cartello addAction ["<t color=#ff0000>VEICOLI DI TERRA</t>",""];
_cartello addAction ["Carro armato Abram","veicoli\Abram.sqf"];
_cartello addAction ["Trasporto truppe MRap","veicoli\MRap.sqf"];
//
```



36 AGGIUNGERE UN'IMMAGINE AD UN OGGETTO

Per aggiungere una texture ad un oggetto:

1. Piazzare un oggetto, es un cartello
2. Incollare nel suo init la stringa

```
this setObjectTextureGlobal [0,"myPicture.paa"];
```

Funziona anche in multiplayer.

File myPicture.paa nella cartella missione o in una cartella dedicata, nel qual caso va messo "cartellaimmagini\myPicture.paa"

Dimensioni valide per l'immagine

Tutte le textures devono avere una risoluzione di $2^x / 2^y$ (ad es. 16x16, 16x32, 64x256, 512x32, ecc.). La dimensione della texture più grande supportata dal motore RV è 4096x4096.

Esempi di risoluzioni:

64x64

256x256

512x512

1024x1024

2048 x 2048

512x256

Formati supportati: .paa od anche .jpg

Funzione **setObjectTextureGlobal**

Description:

Set the texture of the given selection on all computers in a network session.

Not all objects can be textured this way. See [getObjectTextures](#) for supported texture selections.

Multiplayer:

The effect is [JIP](#) compatible.

Syntax:

```
obj setObjectTextureGlobal [selection, texture]
```

Parameters:

obj: [Object](#)

[selection,texture]: [Array](#)

selection: [Number](#)

texture: [String](#)

Return Value:

[Nothing](#)

Example 1

```
_block setObjectTextureGlobal [0, "#(rgb,8,8,3)color(0,0,1,1)"];
```

Example 2

```
_block setObjectTextureGlobal [0, "images\myPicture.paa"];
```

Esempio di un cartello con una texture contenente le regole del server. Nell'init del cartello:

```
this enableSimulationGlobal false;  
this allowDamage false;  
this setObjectTextureGlobal [0,"ServerRules.paa"];
```



37 CAMBIARE IL VOLTO AI PLAYERS (CAMO FACE) (DA TESTARE)

Di seguito un modo per assegnare un volto con una texture mimetica ai players. Utile per esempio se si fa un missione in una foresta e si vuole che i giocatori abbiano die segni mimetici sul viso.

Questo metodo rimuove la faccia di Arma impostata dal player sostituendola con una assegnata in missione.

Il comando per far ciò è tramite la funzione "setFace".

Queste le facce assegnabili:

AfricanHead_01	CamoHead_White_09_F	Kerry_A_F	TanoanHead_A3_07
AfricanHead_02	CamoHead_White_10_F	kerry_B1_F	TanoanHead_A3_08
AfricanHead_03	CamoHead_White_11_F	kerry_B2_F	WhiteHead_01 // Default
AsianHead_A3_01	CamoHead_White_12_F	Kerry_C_F	WhiteHead_02
AsianHead_A3_02	CamoHead_White_13_F	LivonianHead_1	WhiteHead_03
AsianHead_A3_03	CamoHead_White_14_F	LivonianHead_2	WhiteHead_04
AsianHead_A3_04	CamoHead_White_15_F	LivonianHead_3	WhiteHead_05
AsianHead_A3_05	CamoHead_White_16_F	LivonianHead_4	WhiteHead_06
AsianHead_A3_06	CamoHead_White_17_F	LivonianHead_5	WhiteHead_07
AsianHead_A3_07	CamoHead_White_18_F	LivonianHead_6	WhiteHead_08
Barklem	CamoHead_White_19_F	LivonianHead_7	WhiteHead_09
CamoHead_African_01_F	CamoHead_White_20_F	LivonianHead_8	WhiteHead_10
CamoHead_African_02_F	CamoHead_White_21_F	LivonianHead_9	WhiteHead_11
CamoHead_African_03_F	Custom	LivonianHead_10	WhiteHead_12
CamoHead_Asian_01_F	Default	Mavros	WhiteHead_13
CamoHead_Asian_02_F	Dwarden	Miller	WhiteHead_14
CamoHead_Asian_03_F	GreekHead_A3_01	Nikos	WhiteHead_15
CamoHead_Greek_01_F	GreekHead_A3_02	O_Colonel	WhiteHead_16
CamoHead_Greek_02_F	GreekHead_A3_03	PersianHead_A3_01	WhiteHead_17
CamoHead_Greek_03_F	GreekHead_A3_04	PersianHead_A3_02	WhiteHead_18
CamoHead_Greek_04_F	GreekHead_A3_05	PersianHead_A3_03	WhiteHead_19
CamoHead_Greek_05_F	GreekHead_A3_06	PersianHead_A3_04_a	WhiteHead_20
CamoHead_Greek_06_F	GreekHead_A3_07	PersianHead_A3_04_l	WhiteHead_21
CamoHead_Greek_07_F	GreekHead_A3_08	PersianHead_A3_04_sa	WhiteHead_22_a
CamoHead_Greek_08_F	GreekHead_A3_09	Pettka	WhiteHead_22_l
CamoHead_Greek_09_F	GreekHead_A3_10_a	RussianHead_1	WhiteHead_22_sa
CamoHead_Persian_01_F	GreekHead_A3_10_l	RussianHead_2	WhiteHead_23
CamoHead_Persian_02_F	GreekHead_A3_10_sa	RussianHead_3	WhiteHead_24
CamoHead_Persian_03_F	GreekHead_A3_11	RussianHead_4	WhiteHead_25
CamoHead_White_01_F	GreekHead_A3_12	RussianHead_5	WhiteHead_26
CamoHead_White_02_F	GreekHead_A3_13	Sturrock	WhiteHead_27
CamoHead_White_03_F	GreekHead_A3_14	TanoanBossHead	WhiteHead_28
CamoHead_White_04_F	Hladas	TanoanHead_A3_01	WhiteHead_29
CamoHead_White_05_F	IG_Leader	TanoanHead_A3_02	WhiteHead_30
CamoHead_White_06_F	Ioannou	TanoanHead_A3_03	WhiteHead_31
CamoHead_White_07_F	Ivan	TanoanHead_A3_04	WhiteHead_32
CamoHead_White_08_F	Jay	TanoanHead_A3_05	
	Kerry	TanoanHead_A3_06	

Nel campo init dell'unità giocabile o nel loadout

Set persistent face for a unit in MP

```
if (isServer) then {[_unit, "AsianHead_A3_02"] remoteExec ["setFace", 0, _unit]};
```

Nel campo init dell'unità giocabile.

```
if (isServer) then {[this, "WhiteHead_04_cfaces_lrrpcamo02", ""] call BIS_fnc_setIdentity};
```

Per avere una faccia random

Creare un file "camoface.sqf" nella cartella script

```
//This script will apply a random camo face
//Check if script is running on server, if not, then exit

if ( isServer ) then {

//Set Camoface!
_arrayFace =
[ "CamoHead_White_01_F", "CamoHead_White_02_F", "CamoHead_White_03_F", "CamoHead_White_04_F", "CamoHead_Whit
e_05_F", "CamoHead_White_06_F", "CamoHead_White_07_F", "CamoHead_White_08_F", "CamoHead_White_09_F", "CamoHe
ad_White_10_F", "CamoHead_White_11_F", "CamoHead_White_12_F", "CamoHead_White_13_F", "CamoHead_White_14_F",
```



```
"CamoHead_White_15_F", "CamoHead_White_16_F", "CamoHead_White_17_F", "CamoHead_White_18_F", "CamoHead_White_19_F", "CamoHead_White_20_F", "CamoHead_White_21_F"];  
_randomFace = selectRandom _arrayFace;  
  
[_unit, _randomFace] remoteExec ["setFace", 0, _unit];  
};
```

Oppure più sintetica nel caso della sola tipologia CamoHead_White_xx_F

```
//This script will apply a random camo face  
//Check if script is running on server, if not, then exit  
  
if ( isServer ) then {  
  
//Set Camoface!  
_camo_ix = floor(random 21) + 1; // floor(random 21) seleziona un numero da 0 a 20  
if ( _camo_ix < 10 ) then {  
[_unit, format["CamoHead_White_0%1_F", _camo_ix]] remoteExec ["setFace", 0, _unit]; }  
else {  
[_unit, format["CamoHead_White_%1_F", _camo_ix]] remoteExec ["setFace", 0, _unit]; }  
};
```

Nel campo init dell'unità giocabile.

```
null = [this] execVM "script\camoface.sqf";
```

Nota: Provare a mettere quest'istruzione nel loadout preimpostato che verrà "letto" alla selezione

38 FUOCO DI SOPPRESSIONE

38.1 CREARE UN FUOCO DI SOPPRESSIONE

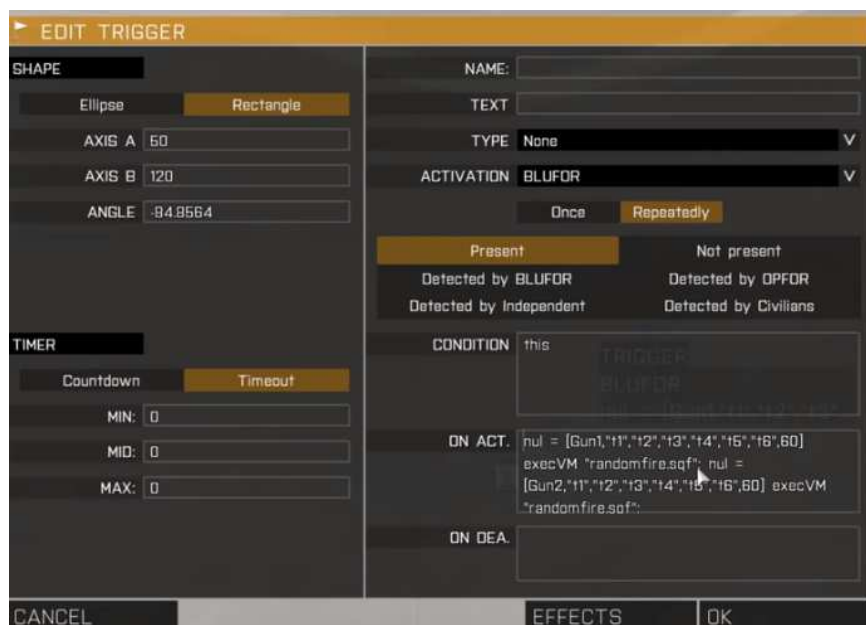
Script per effettuare un fuoco di soppressione da parte di un'unità. Di seguito come crearlo per esempio tramite l'attivazione di un trigger

1. Copiare il file "SupressiveFire.sqf" (vedi più avanti per il contenuto) all'interno della cartella della missione (vedi più avanti per il file sqf)
2. Piazzare l'unità e assegnargli un nome nel nome variabile, per esempio "gun1". Si può anche mettere un'altra unità ("gun2"), o un veicolo ("vec1") (in tal caso farà fuoco il gunner del veicolo), od anche un gruppo di unità (in questo caso assegnare il nome al gruppo per esempio "group1")
3. Piazzare un trigger ed all'attivazione della condizione, sia essa con blufior presente o quant'altro, assegnare la stringa seguente nel campo attivazione

```
0 = [ [gun1, gun2, vec1, group1], [[100,102,0], "marker1", Car1] ] execVM "SupressiveFire.sqf";
```

Nota: le [] vanno mantenute, cioè deve essere della forma 0 = [[qualsiasi unità], [qualsiasi bersaglio]]
execVM "SupressiveFire.sqf";

Il risultato è il seguente: gun1, gun2, vec1 e tutte le unità nel gruppo chiamato group1 spariranno alla posizione in coordinate [100,102,0], al marker in mappa indicato come marker1 e all'unità car1 per 60 secondi (default).



Altro esempio di trigger

4. Specificare i bersagli. Si possono dare coordinate ([100,102,0]), marker in mappa (identificate con un nome nel campo variabile, in questo caso marker1) ed altre unità (identificate con un nome nel campo variabile, in questo caso Car1)



Markers e trigger

5. All'attivazione del trigger verrà chiamato l'esecuzione del file SupressiveFire.sqf e le unità inizieranno a fare fuoco sui bersagli designati.

Varianti della stringa:

Example 2:

```
0 = [ [myVehicle,car1,man1], ["marker1","marker2","marker3","marker4",car2], 120, .1, true ] execVM "SupressiveFire.sqf";
```

Result: myVehicle, car1 e man1 spareranno sui "marker1" a "marker3" e car2 per 120 secondi, ognuno non ricaricherà l'arma. Spareranno di continuo

Example 3:

```
0 = [ [veh1,veh2,veh3,u1,u2,u3,u4,u5,u6,g1,g2,group1], ["marker_0","marker_1","marker_2","marker_3","marker_4","marker_5"], 60, .1, false ] execVM "SupressiveFire.sqf";
```

Result: I vehs, I gruppi e le unità nell'elenco spareranno sui marker 0-5 per 60 secondi, ciascuno verrà riarmato prima di sparare. Se esauriscono le munizioni non sparano.

Example 4:

```
0 = [ [veh1], ["marker_0"], 60, .1, false ] execVM "SupressiveFire.sqf";
```



In generale lo script è della forma:

```
0 = [[0],[1],2,3,4] execVM "SupressiveFire.sqf";
```

Dove:

0: ARRAY: of Object; Può essere uno dei seguenti:

Vehicle – solo fuoco con mitragliere

Units - Variable containing a unit.

Group – tutte le unità del gruppo di fuoco

1: ARRAY: of Position; Può essere uno dei seguenti:

ARRAY – posizione nel formato [x,y] o [x,y,z]

OBJECT - oggetto

GROUP - group leader

LOCATION - location

STRING - marker

2: NUMBER: Il tempo di fuoco delle unità in secondi. Default 60 secondi.

3: NUMBER: Tempo massimo tra le unità che sparano. Default 0.1.

4: BOOL: Give vehicles full ammo before firing and top up units magazines? (true / false) If they run out of ammo they will not fire. Default true (units will not reload of true).

Da ai veicoli munizioni piene prima di sparare e ricarica le unità di caricatori. (vero / falso) Se esauriscono le munizioni non sparano. Valore predefinito true (le unità non ricaricano il valore true).

6. Contenuto del file SupressiveFire.sqf

```
/*  
  
    Supressive Fire Example  
  
    Author: Big_Wilk  
  
    Description: Forces units or vehicles including static weapons to fire randomly at passed  
world positions.  
  
    Parmas:  
    0: ARRAY: of Object; Can be any of the following:  
        Vehicle - "gunner" fires only  
        Units - Variable containing a unit.  
        Group - All units in the group fire  
    1: ARRAY: of Potion; Can be any of the following:  
        ARRAY - position in format [x,y] or [x,y,z]  
        OBJECT - object  
        GROUP - group leader  
        LOCATION - location  
        STRING - marker  
    2: NUMBER: The time the units will fire for in seconds. Default 60 seconds.  
    3: NUMBER: Max time between units fireing. Default 0.1.  
    4: BOOL: Give vehicles full ammo before firing and top up units magazines? (true / false) If  
they run out of ammo they will not fire. Default true (units will not reload of true).  
  
    Example 1:  
    0 = [ [gun1, gun2, group1], [[100,102,0],"marker1",Car1] ] execVM "SupressiveFire.sqf";  
    Result: gun1, gun2 and all units in a group named group1 will fire at "marker1" and car1 for  
60 seconds.  
  
    Example 2:  
    0 = [ [myVehicle,car1,man1], ["marker1","marker2","marker3","marker4",car2], 120, .1, true ]  
execVM "SupressiveFire.sqf";  
    Result: myVehicle, car1 and man1 will fire at "marker1" to "marker3" and car2 for 120  
seconds, each will be rearmed before firing.  
  
    Example 3:  
    0 = [ [veh1,veh2,veh3,u1,u2,u3,u4,u5,u6,g1,g2,group1],  
["marker_0","marker_1","marker_2","marker_3","marker_4","marker_5"], 60, .1, false ] execVM  
"SupressiveFire.sqf";  
    Result: The vehs groups and units list will fire at markers 0-5 for 60 seconds and wont be  
armed, if they run out of ammo they will stop firing.
```




```
*/  
  
private ["_arrayOfUnits", "_arrayOfVehicles"];  
  
// Parmas  
_gunnerArray = _this param [0, [], [[]]];  
_posArray = _this param [1, [], [[]]];  
_forSeconds = _this param [2, 60, [60]];  
_sleepTime = _this param [3, .1, [.1]];  
_rearm = _this param [4, true, [true]];  
  
if ( count _gunnerArray == 0 or count _posArray == 0 ) exitWith { systemChat "No input for  
SuppressiveFire.sqf"; };  
  
// Covert Pos array into real pos  
_allPos = [];  
{  
    if !(isNil "_x") then {  
        _posX = _x call BIS_fnc_position;  
        if (str _posX != "[0,0,0]") then {  
            _allPos pushBack _posX;  
        };  
    };  
} forEach _posArray;  
  
// Split Units and vehicles;  
_arrayOfUnits = [];  
_arrayOfVehicles = [];  
{  
    private "_unit";  
    _unit = _x;  
    switch (typeName _unit) do {  
        case "GROUP": {  
            {  
                if (_x isKindOf "man") then {  
                    _arrayOfUnits pushBack _x;  
                } else {  
                    _arrayOfVehicles pushBack _x;  
                };  
            } forEach units _unit;  
        };  
        case "OBJECT": {  
            if (_unit isKindOf "man") then {  
                _arrayOfUnits pushBack _unit;  
            } else {  
                _arrayOfVehicles pushBack _unit;  
            };  
        };  
    };  
} forEach _gunnerArray;  
  
// Set combat mode of units:  
{  
    _x setBehaviour "combat";  
    _unit = _x;  
    {_unit reveal _x;} forEach allUnits;  
    //_x suppressFor _forSeconds;  
} forEach _arrayOfUnits;  
  
{  
    _x setVehicleAmmo 1;  
} forEach _arrayOfVehicles;  
  
sleep 0.5;  
  
// Fire the weapons for time passed:  
_endTime = time + _forSeconds;  
While {time < _endTime} do {  
  
    {  
  
        {  
            if (alive _x) then {  
                if _rearm then {  
                    _x setAmmo [primaryWeapon _x, 1000];  
                };  
            };  
        };  
    };  
};
```



```
};
_x doWatch (_allPos call BIS_fnc_selectRandom);
sleep random _sleepTime;
_x forceWeaponFire [(primaryWeapon _x), "FullAuto"];
    } else {
        _arrayOfUnits = _arrayOfUnits - [_x];
    };
} forEach _arrayOfUnits;
{
    if (canFire _x) then {
        if _rearm then {
            _x setVehicleAmmo 1;
        };
        gunner _x doWatch (_allPos call BIS_fnc_selectRandom);
        sleep random _sleepTime;
        _x action ["useWeapon", _x, gunner _x,1];
    } else {
        _arrayOfVehicles = _arrayOfVehicles - [_x];
    };
} forEach _arrayOfVehicles;
} forEach _arrayOfUnits;
};
true;
```

38.2 TERMINARE (ABORTIRE) UNO SPAWNED OR EXECVMED SCRIPT

Per terminare uno script si può usare il comando **terminate**

Syntax:

terminate scriptHandle

Parameters:

scriptHandle: Script

Return Value:

Nothing

Lo script specificato non terminerà immediatamente al termine dell'esecuzione del comando, lo farà la prossima volta che lo script viene elaborato dallo scheduler

Example 1:

```
_script = [] execVM "script.sqf";
sleep 5;
terminate _script;
```

nel nostro caso si può impostare un trigger ad alla sua attivazione mettere la stringa che termina il fuoco di soppressione dell'unità

```
terminate execVM "SupressiveFire.sqf";
```

Create a trigger (enemy faction present & detected by friendly faction), add in the on-activation of the trigger the following code:

```
_nil = [] call resetFunction;
terminate _scripthandle;
```

You'll have to make the reset function global to be able to do this.

Nel nostro caso

```
_nil = [] call resetFunction;
```



```
terminate execVM "SupressiveFire.sqf";
```

```
_script = [] execVM "SupressiveFire.sqf";  
_nil = [] call resetFunction;  
terminate _script;
```

Terminating a script on a trigger.

Terminate the script unconditionally by using terminate command, for that you will need to assign a handle to your script, like so:

```
myHandle = [] spawn myScript;  
//or  
myHandle = [] execVM "filename.sqf";  
Then, in your trigger you use terminate myHandle;
```

This option is not optimal, as it terminates script unconditionally, without letting clean up whatever it was doing, the second option, introducing a boolean flag into the script, is much better:

```
myScriptIsRunning = true;  
myHandle = [] execVM "filename.sqf"; //still a good idea to assign a handle to all scripts you run on scheduler
```

```
//inside of your script, if you have a while {true} loop, replace it with  
while { myScriptIsRunning } do { /* rest of the code */ };  
//some cleanup after loop exits
```

And in the trigger you just set the value of myScriptIsRunning to false:

```
myScriptIsRunning = false;
```

38.3 ALTRA PROCEDURA PER FUOCO DI SOPPRESSIONE

Questo usa un oggetto target invece di un marker per indirizzare il fuoco di soppressione.

1. Piazzare un oggetto su cui fare fuoco di soppressione e assegnargli un nome nel campo nome variabile, per esempio "nomeoggetto"
2. Inserire la stringa per il fuoco di soppressione sul campo init dell'unità o (se si vuole) su un suo waypoint

```
{_x doSuppressiveFire nomeoggetto;}  
forEach units group this;
```

l'ultima riga applicherà il fuoco di soppressione a tutti i membri se l'unità è un TL di un gruppo.

Descrizione funzione **doSuppressiveFire**

Syntax:

```
unit doSuppressiveFire target
```

Parameters:

unit: Object or Array - a single unit or array of units

target: Object or Array - actual target or position of the target in format PositionASL

Return Value:

Nothing

Example 1:

```
_soldier doSuppressiveFire [1869.508,5760.962,0.000]
```

Example 2:

```
_soldier doSuppressiveFire cursorTarget
```



Un modo alternativo è usare quest'altra stringa

```
{_x suppressFor 10;}  
forEach units group this;
```

essa costringerà l'unità (e tutte quelle del suo gruppo se TL) a sparare in modo repressivo ai nemici noti per 10 secondi

Descrizione funzione **suppressFor**

Syntax:

```
unit suppressFor duration
```

Parameters:

unit: Object

duration: Number

Return Value:

Nothing

Example 1

```
_soldier1 suppressFor 10;
```

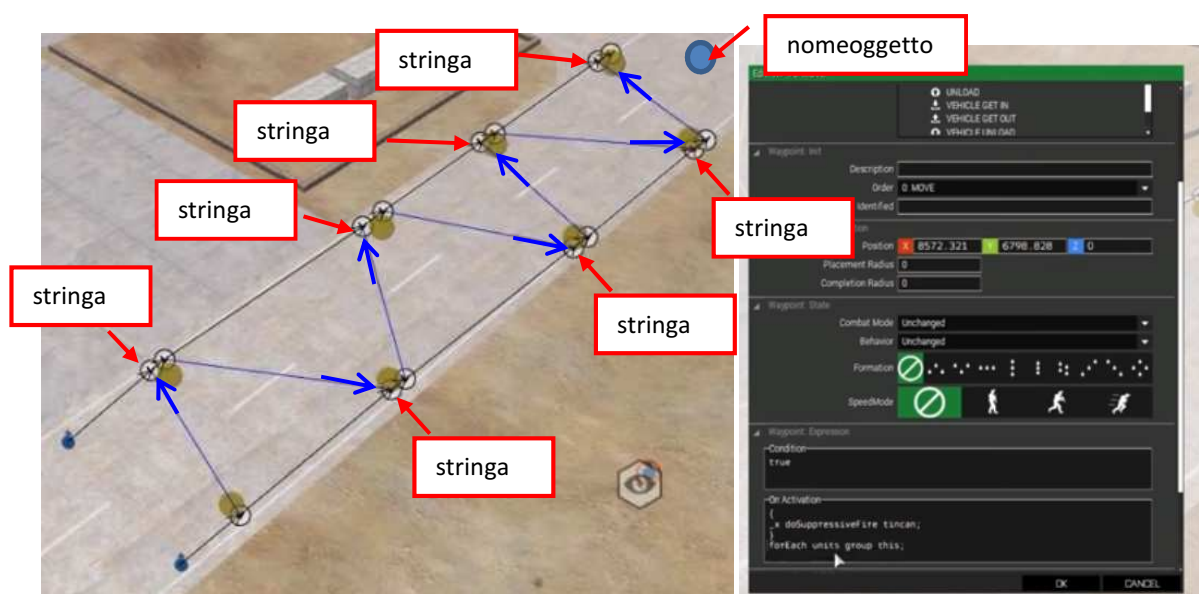
" Costringerà il soldato1 a sparare in modo repressivo ai nemici noti per 10 secondi "

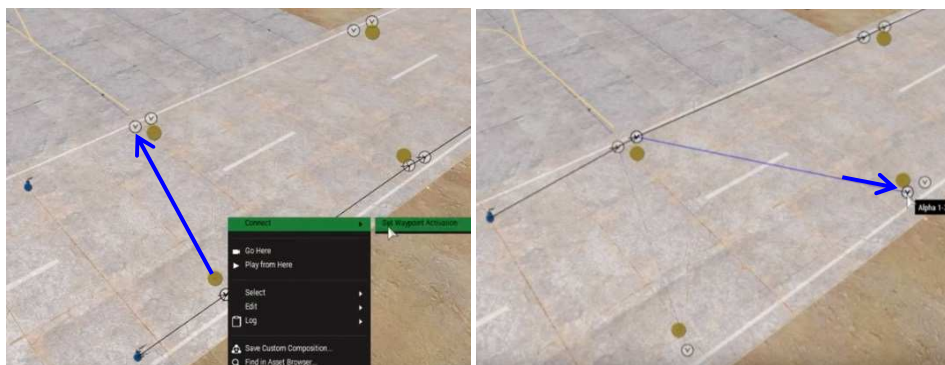
38.4 ESEMPIO DI MOVIMENTO A SBALZI CON FUOCO DI SOPPRESSIONE

1. Piazzare un oggetto su cui fare fuoco di soppressione e assegnargli un nome nel campo nome variabile, per esempio "nomeoggetto"
2. Impostare i waypoint di movimento. Nota: metterne due vicini come nell'esempio
3. Inserire la stringa per il fuoco di soppressione su ognuno dei primi waypoint come in figura

```
{  
_x doSuppressiveFire nomeoggetto;  
}  
forEach units group this;
```

4. Collegare l'attivazione di ogni waypoint agli altri come nelle figure (tasto destro sul waypoint Collega/imposta attivazione waypoint). Così impostati il movimento dell'unità A si attiva quando la B farà soppressione. Una volta che A ha raggiunto la posizione farà a sua volta soppressione e la B smetterà di sparare ed avanzerà.





39 SUPPORT (ARTILLERY, TRANSPORT ETC.)

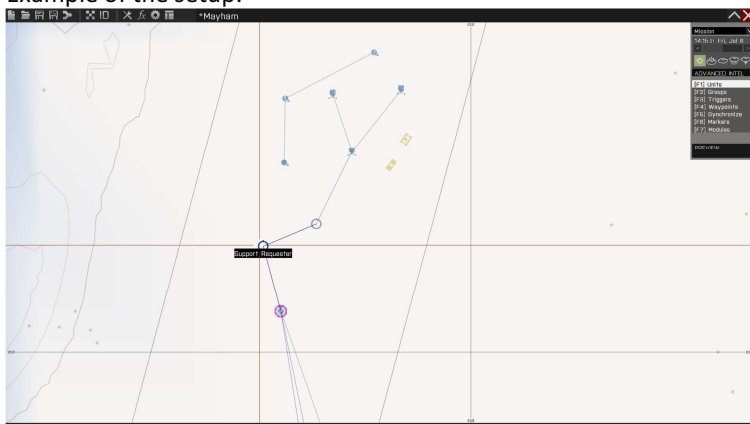
Many of the current support modules do not work, seeing as how there are no planes to do bombing runs or air strikes and the only artillery that you can use are mortars, but there is still a little bit you can do. Here are a few easy ways to get it all setup.

39.1 ARTILLERY

Vedi anche §39.3.

1. Create mortar units wherever, somewhere flat and safe (unless you want virtual ones)
2. Create a support requester module (the empty one that doesn't apply to anything specific)
3. Create a support provider: artillery module
4. With the group tool ungroup modules and mortars if they automatically did it
5. Using the sync tool link the support requester module to the player
6. Using the sync tool link the support requester module to the support provider: artillery module
7. Using the sync tool link the support provider: artillery module to the mortar squad (they should stay grouped if more than one mortar)
8. Profit

Example of the setup:



39.2 HELICOPTER TRANSPORT TAXI

Vedi anche §44.1.4.

1. Create a friendly AI controlled helicopter somewhere safe and flat.
2. Create a support requester module (the empty one again)
3. Create a Support Provider: Helicopter Transport module.
4. Once again, ungroup everything if it has become automatically grouped.
5. Using the sync tool link the support requester module to the player



6. Using the sync tool link the support requester module to the support provider: Helicopter Transport module.
7. Using the sync tool link the support provider: Helicopter Transport module to the actual AI controlled helicopter.
8. Free helicopter taxi at your beck and call.

With both of these you can sync to multiple units, not just one, incase you want them to work for multiple players, but make sure two players are not trying to use the same support at the same time (could cause issues).

To access support hit 10 and then 8 (communications). You can then select an area on your map or wherever you are pointing, but be careful with trying to call in support just from looking somewhere with your gun, it is very inprecise. I recommend using your map or binoculars, especially for artillery.

39.3 ADDING AN ARTILLERY PROVIDER

ADDING A VIRTUAL ARTILLERY PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: Artillery (Virtual) {SupportProvider_Virtual_Artillery}
4. Sync the two Modules together
5. Sync each unit that will be able to request artillery support to the Support Requester {SupportRequester} Module

ADDING A PHYSICAL ARTILLERY PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: Artillery {SupportProvider_Artillery}
4. Sync the two Modules together
5. Place Artillery units on map
6. Sync each Artillery unit to the Support Provider: Artillery {SupportProvider_Artillery} Module
7. Sync each unit that will be able to request artillery support to the Support Requester {SupportRequester} Module

39.4 ADDING AN AIRCRAFT CLOSE AIR SUPPORT PROVIDER

ADDING A VIRTUAL AIRCRAFT CLOSE AIR SUPPORT PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: CAS (Bombing Run, Virtual) {SupportProvider_Virtual_CAS_Bombing}
4. Sync the two Modules together
5. Sync each unit that will be able to request transport to the Support Requester {SupportRequester} Module

ADDING A PHYSICAL AIRCRAFT CLOSE AIR SUPPORT PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: CAS (Bombing Run) {SupportProvider_CAS_Bombing}
4. Sync the two Modules together
5. Place transport Helicopters on map
6. Sync each helicopter to the Support Provider: Helicopter Transport {SupportProvider_CAS_Bombing} Module
7. Sync each unit that will be able to request transport to the Support Requester {SupportRequester} Module

39.5 ADDING A HELICOPTER CLOSE AIR SUPPORT PROVIDER

ADDING A VIRTUAL HELICOPTER CLOSE AIR SUPPORT PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: CAS (Helicopter Attack, Virtual) {SupportProvider_Virtual_CAS_Heli}



4. Sync the two Modules together
5. Sync each unit that will be able to request transport to the Support Requester {SupportRequester} Module

ADDING A PHYSICAL HELICOPTER CLOSE AIR SUPPORT PROVIDER

1. Systems > Modules > Supports > Support Requester {SupportRequester}
2. Set any limits you wish to enforce
3. Systems > Modules > Supports > Support Provider: CAS (Helicopter Attack) {SupportProvider_CAS_Heli}
4. Sync the two Modules together
5. Place Attack Helicopters on map
6. Sync each helicopter to the Support Provider: Helicopter Transport {SupportProvider_CAS_Heli} Module
7. Sync each unit that will be able to request transport to the Support Requester {SupportRequester} Module

40 AGGIUNGERE CIVILI IN MAPPA

CIVILIAN PRESENCE

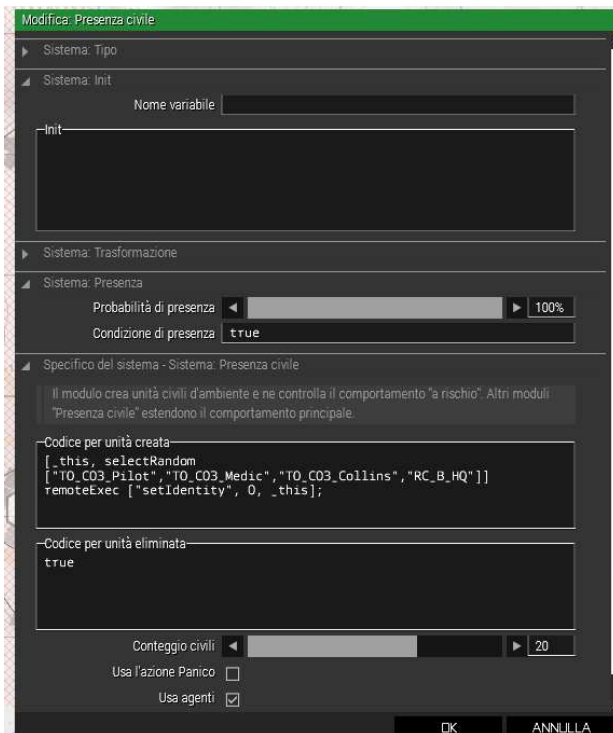
1. Systems > Modules > Ambient> Civilian Presence {ModuleCivilianPresence_F} module
2. Set position, shape, size and rotation as needed.
3. Set Civilian Count to maximum value required.
4. Sytems > Modules > Civilian Presence Spawnpoint {ModuleCivilianPresenceUnit_F} at required civ spawnpoint.
5. Sytems > Modules > Civilian Presence Position {ModuleCivilianPresenceSafeSpot_F} at required journey end points.

*Spawnpoint and Position modules MUST be placed within the boundaries set by the Presence module.

Di default vengono spawnati civili della vanilla. Per selezionare die civile da una mod in particolare mettere la seguente stringa in Codice per unità creata del modulo presenza civile:

```
[_this, selectRandom ["TO_C03_Pilot", "TO_C03_Medic", "TO_C03_Collins", "RC_B_HQ"]] remoteExec ["setIdentity", 0, _this];
```

Le classi incluse vanno scelte nelle classname delle unità della mod.



Per civili Takistan:

```
[_this, selectRandom ["LOP_Tak_Civ_Man_01", "LOP_Tak_Civ_Man_02", "LOP_Tak_Civ_Man_04", "LOP_Tak_Civ_Man_05", "LOP_Tak_Civ_Man_06", "LOP_Tak_Civ_Man_07", "LOP_Tak_Civ_Man_08", "LOP_Tak_Civ_Man_09", "LOP_Tak_Civ_Man_10", "LOP_Tak_Civ_Man_11"]]
```



```
_11", "LOP_Tak_Civ_Man_12", "LOP_Tak_Civ_Man_13", "LOP_Tak_Civ_Man_14", "LOP_Tak_Civ_Man_15", "LOP_Tak_Civ_Man_16"] remoteExec ["setIdentity", 0, _this];
```

41 PLAY RADIO MESSAGE

1. Add Audio > Play Radio Message {ModuleRadio_F} module
2. Select the Side for which radio message will play
3. Select the Sentence that will play

You can, of course, set this up to be activated via a trigger, otherwise it will always be played.

42 EFFETTI AMBIENTE

<https://www.youtube.com/watch?v=hHFT8NNIsy8>

43 WAYPOINTS

Queste funzioni, perché si possano sfruttare, hanno bisogno d'essere abbinare con un'unità o gruppo, sia essa di terra, di mare o di aria.

Le tipologie di waypoints sono:

- A BORDO
- A BORDO PIU' VICINO
- CARICARE
- CERCARE E DISTRUGGERE
- CICLICO
- CON SCRIPT
- CONGIUNGERSI
- CONGIUNGERSI E COMANDARE
- DISTRUGGERE
- ENTRA NEL VEICOLO
- ESCI DAL VEICOLO
- GUARDIA
- IN ATTESA
- LIBERO
- MANTENERE
- MUOVERSI
- PARLARE
- SCARICARE VEICOLO
- SCARICO DEL VEICOLO
- SCENDERE
- SCENDERE
- SEGUI
- SENTINELLA
- SGANCIA CARICO
- SOLLEVA CARICO
- SUPPORTO

E' buona regola, in qualsiasi situazione, utilizzare la funzione "IN ATTESA". Questa serve a dire all'unità o al gruppo di attendere prima di fare qualsiasi altra cosa. Ci sono diverse soluzioni per usufruire dell'attesa, una può essere legata al tempo con minimo-medio-massimo l'altra tramite l'attivazione di un trigger collegandoli tra loro.

Questa pratica evita comportamenti anomali non voluti prima di qualsiasi evento prestabilito nella missione. Subito dopo questa precauzione potete aggiungere qualsiasi altra funzione in elenco per portare a termine l'azione delle unità di vostro interesse.



Spiegazione dei waypoint:

Cercare e distruggere

Questo comando sollecita l'unità, gruppo o mezzo a cercare unità nemiche e distruggerle, molto utile. Da segnalare una differenza di comportamento tra mezzi e unità, per i primi in raggio d'azione, nella ricerca, non supera i 300m mentre per le unità o gruppi 50m.

Ciclico

Con questo comando si obbliga le unità collegate a compiere all'infinito un certo comportamento pre impostato. Meglio non abusare molto di questo comando con in mezzi onde evitare che si esaurisca il carburante prima del tempo!

Congiungersi

L'unità od il gruppo si muoverà verso un'altra unità o gruppo e si unirà a quest'ultima.

Congiungersi e comandare

L'unità od il gruppo si muoverà verso un'altra unità o gruppo e si unirà a quest'ultima comandandola.

Distruggere

Questo waypoint risulta tra i più ambigui, sembra non funzionare e molti nella community si lamentano del suo utilizzo. Questo tipo di waypoint funziona meglio quando è collegato a un oggetto. Il gruppo tenterà di distruggere qualsiasi oggetto su cui è posizionato il waypoint, indipendentemente dal lato dell'oggetto bersaglio. Se il gruppo non è in grado di distruggere l'oggetto attaccato, si sposterà nel raggio di poter identificare l'oggetto, quindi aspetta fino a quando non viene distrutto (anche se il bersaglio non è distruttibile!). Se il waypoint viene posizionato spazialmente, il comportamento del waypoint è meno prevedibile. Se il waypoint è lontano dagli oggetti posizionati dall'editor della missione, il waypoint verrà generalmente considerato istantaneamente completo senza che il gruppo si sposti verso la posizione del waypoint. Se c'è un oggetto vicino al waypoint spaziale, il gruppo può trattare quell'oggetto come un normale bersaglio di distruzione. Tuttavia, a volte il gruppo può iniziare ad attaccare l'oggetto, quindi passare immediatamente al waypoint successivo senza distruggere nulla. L'oggetto esatto che il gruppo attaccherà non è facilmente prevedibile e se il gruppo stesso si trova vicino al waypoint Distruggi spaziale quando diventa attivo, il gruppo può persino scegliere di attaccare, ma non necessariamente distruggere, uno dei suoi membri.

Guardia

Raggiunta la zona da sorvegliare l'unità farà la guardia dell'area di interesse. Se verrà attaccata cercherà riparo per difendersi ma una volta cessato l'allarme i sopravvissuti si riposizioneranno nei loro punti precedenti NON abbandonano mai l'area di pertinenza.

Libero

Con questo comanda si libera la squadra dalla formazione predefinita e le unità si muoveranno in diverse direzioni con comportamenti diversi.

Mantenere

Questo tipo di waypoint farà spostare il gruppo e rimarrà in questa posizione indefinitamente. Solo un trigger di tipo Switch o un comando di script sposta il gruppo dal waypoint. Durante l'attesa, vengono controllati tutti i bersagli sconosciuti (i gruppi si muovono nella direzione del bersaglio) e tutti i nemici riconosciuti vengono ingaggiati.

Muoversi

Tra i più usati per permettere il movimento delle unità. Se utilizzato per un mezzo di terra e posto sulla strada anche se il movimento in linea d'aria non segue l'andamento della strada, in mezzo si muoverà seguendo correttamente la direzione stradale come se fosse su "binari".

Scaricare veicolo

Il gruppo si sposterà sul waypoint (spaziale o oggetto), dove sbarcheranno tutte le unità di altri gruppi che si trovano in spazi di carico dei veicoli del gruppo originale. Su un server dedicato, questo waypoint non funziona se il comandante del gruppo trasportato nella parte posteriore del veicolo è umano. Funzionerà solo con AI. Su un server dedicato quando lo si utilizza con elicotteri controllati dall'IA. Questo comando funziona solo se nel carico del veicolo è presente un



comandante del team AI. Se ci sono solo giocatori umani, l'elicottero non atterrerà. Si libererà troppo in alto per sbarcare in sicurezza.

Scarico del veicolo

Il gruppo si sposterà sul waypoint (spaziale o oggetto), quindi una qualsiasi delle sue unità che si trovano nello spazio di carico di qualsiasi veicolo sbarcherà. Le unità nelle posizioni dell'equipaggio non sbarcheranno. Una nota, su un server dedicato quando lo si utilizza con elicotteri controllati dall'IA. Questo comando funziona solo se nel carico del veicolo è presente un comandante del team AI. Se ci sono solo giocatori umani, l'elicottero non atterrerà. Si libererà troppo in alto per sbarcare in sicurezza.

In particolare per l'atterraggio di un elicottero AI a terra e lo scarico di passeggeri AI, la destinazione del waypoint per Transport Unload deve essere un oggetto eliporto ed essere assegnata ai piloti e il gruppo separato come passeggeri deve avere il waypoint Get Out nella stessa posizione. Sia il waypoint di scarico del trasporto che il waypoint di uscita devono essere sincronizzati tra loro. Nota: createVehicleCrew interrompe questa funzionalità.

Sentinella

Sembra simile a guardia ma palesemente diverso. Da abbinare ad altri waypoint decisionali per avere comportamenti secondari. In questo caso se attaccati ingaggeranno il nemico e alla fine di tutto si attiverà il waypoint successivo. Il gruppo si sposterà sul waypoint e manterrà la posizione fino a quando non saprà abbastanza su un'unità nemica per identificare da quale parte appartiene e che sono in grado di attaccare. A quel punto, il gruppo procederà al prossimo waypoint. Normalmente attaccheranno anche il nemico individuato lungo la strada.

Supporto

Un gruppo con un waypoint corrente di questo tipo si sposterà nella posizione del waypoint, quindi attenderà fino a quando non sarà in grado di fornire supporto per un altro gruppo che richiede supporto pertinente utilizzando il menu di comando "Chiama supporto". A questo punto, il gruppo in attesa procederà verso la posizione in cui è stato richiesto il supporto, sbarcare da qualsiasi veicolo sia in possesso e attendere che il gruppo supportato dichiari "Supporto fatto" prima di continuare ad attendere per rispondere a qualsiasi ulteriore richiesta di supporto dal proprio nuova sede. Solo i gruppi con un medico, un'ambulanza, carburante, munizioni o veicoli di riparazione possono rispondere alle richieste di supporto. Nel caso di più gruppi di supporto, il primo gruppo disponibile posizionato risponderà sempre per primo, indipendentemente dalla distanza. Sebbene questo sia un tipo di waypoint potente, richiede diligenza da parte di tutti i leader umani, in quanto devono annunciare "Support Done" per consentire al gruppo di supporto AI di continuare a essere in grado di fornire supporto. I leader dell'IA tendono anche a richiedere che ogni singolo gruppo di supporto appropriato li assista. L'intelligenza artificiale non prenderà in considerazione la vulnerabilità di un gruppo di supporto quando richiede assistenza.

44 HELICOPTER INSERTION/EXTRACTION

<https://www.youtube.com/watch?v=w4IF3yESNI>

44.1 INSERTION/EXTRACTION

<https://www.youtube.com/watch?v=2QBUQ8JKfuM>

44.1.1 INSERZIONE

Elitrasporto con scarico truppe da un punto ad un altro.

EFFETTO:

Una volta entrati in elicottero alla base, tramite rotellina è disponibile il comando "Helicopter Go!" che lo farà decollare ed atterrare nel punto designato.

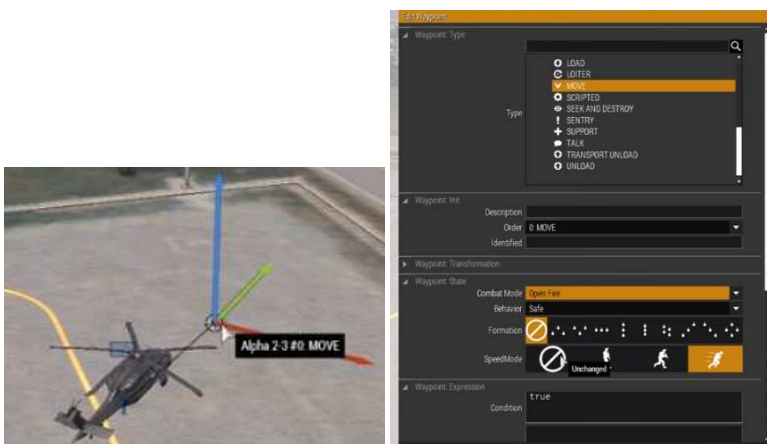
Una volta che il carico trasportato è sceso dall'elicottero esso ritorna automaticamente in base.

Procedura:

1. Creare l'unità elicottero in una posizione A (per esempio una base).
2. Cliccare sul simbolo dell'elicottero e creare un waypoint ("muovere") accanto ad esso. Questo serve ad attivare l'elitrasporto tramite trigger.



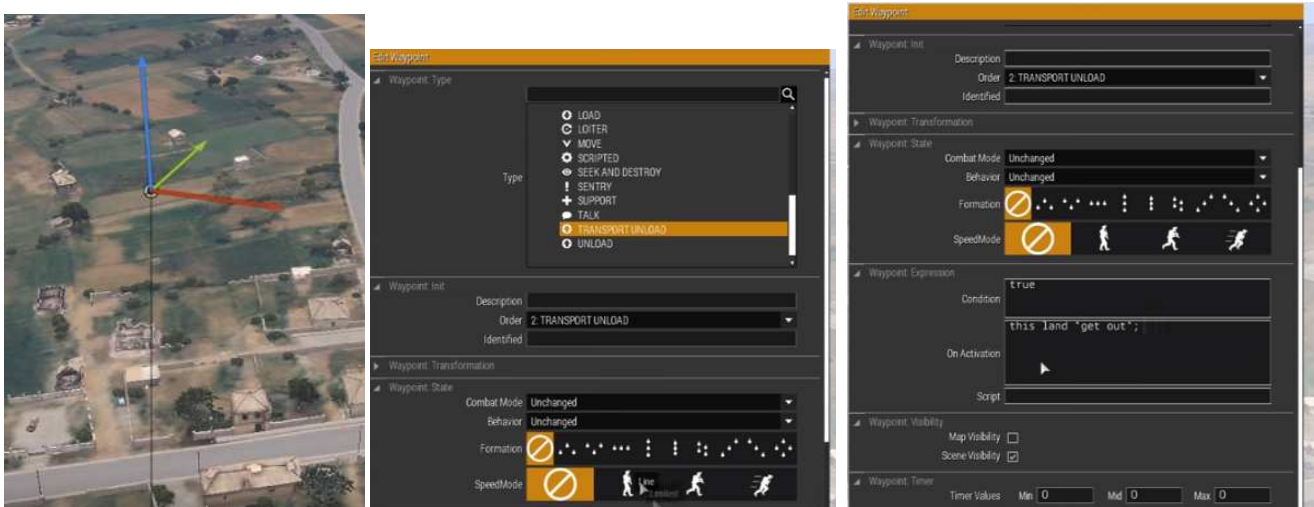
Impostarne il modo di combattimento: combattimento (aprirà il fuoco se serve)
Comportamento: safe
Velocità massima



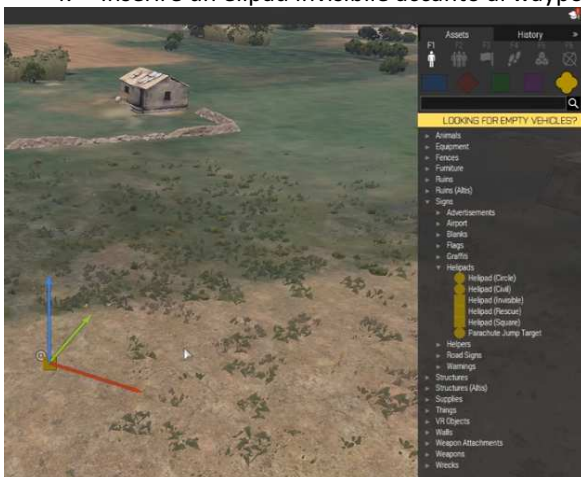
- 3. Creare la rotta di waypoint fino alla posizione desiderata. Mettere il waypoint finale come “scarica trasporto”. Nel suo campo Attivazione mettere la seguente stringa

```
this land 'get out';
```

Una volta nel waypoint aspetterà lo sbarco delle unità trasportate dietro prima di ripartire



- 4. Inserire un elipad invisibile accanto al waypoint per far atterrare l'elicottero esattamente in quella posizione





5. Aggiungere un waypoint nuovamente in base del tipo “muovere”



Nel suo campo Attivazione mettere la seguente stringa

```
Vehicle this land 'land'; if (isServer) then {BW_HelicopterGo = false; publicVariable 'BW_HelicopterGo' }
```

6. Aggiungere un waypoint vicino del tipo “ciclico”

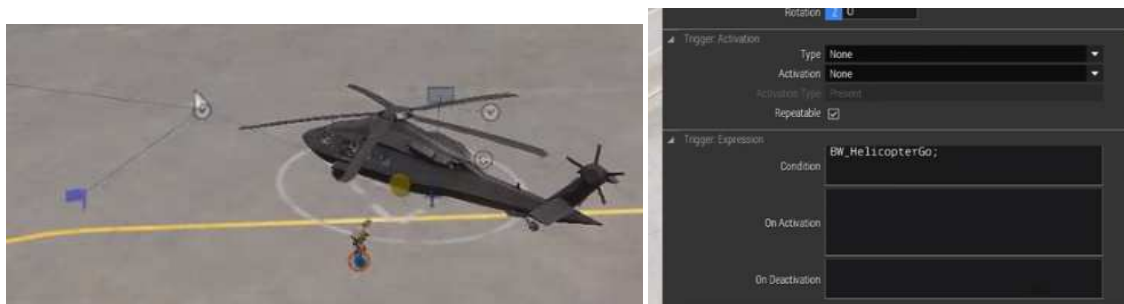
Nota: testare se occorre mettere la seguente stringa anche in questo waypoint ciclico

```
Vehicle this land 'land';
```

7. Aggiungere anche un elipad, questa volta anche visibile, accanto al waypoint per far atterrare l'elicottero esattamente in quella posizione
8. Aggiungere un trigger e sincronizzarlo con il primo waypoint (“set attivazione waypoint”). Esso servirà a chiamare l'eli da parte dei giocatori. spuntare “ripetibile” per permetterne il richiamo più volte, ed inserire la stringa seguente nella Condizione d'attivazione.

```
BW_HelicopterGo;
```

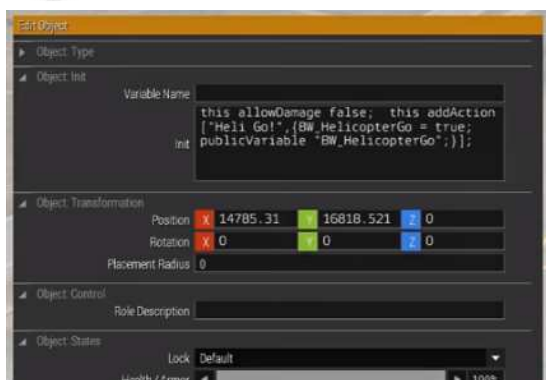
il trigger si attiverà quando la funzione in questione sarà richiamata permettendo il decollo dell'elicottero.



9. Nel campo init dell'oggetto elicottero inserire la stringa

```
this allowDamage false; // serve a rendere l'elicottero invincibile ai danni. Togliere tale riga se non necessaria  
this addAction ["Helicopter Go!", {BW_HelicopterGo = true; publicVariable "BW_HelicopterGo"}];
```

la prima istruzione lo rende invincibile. Rimuoverla se necessario. La seconda attiva il decollo



44.1.2 ESTRAZIONE

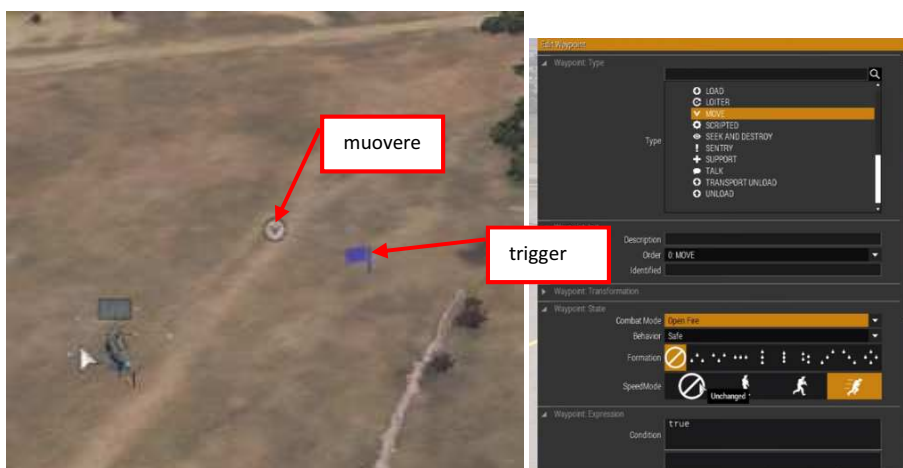
Estrazione quando tutti i player sono sull'elicottero.

EFFETTO:

L'elicottero verrà chiamato in radio ed atterrerà nella punto stabilito. Una volta che tutti saranno a bordo decollerà automaticamente ed atterrerà al punto di ritorno prefissato.

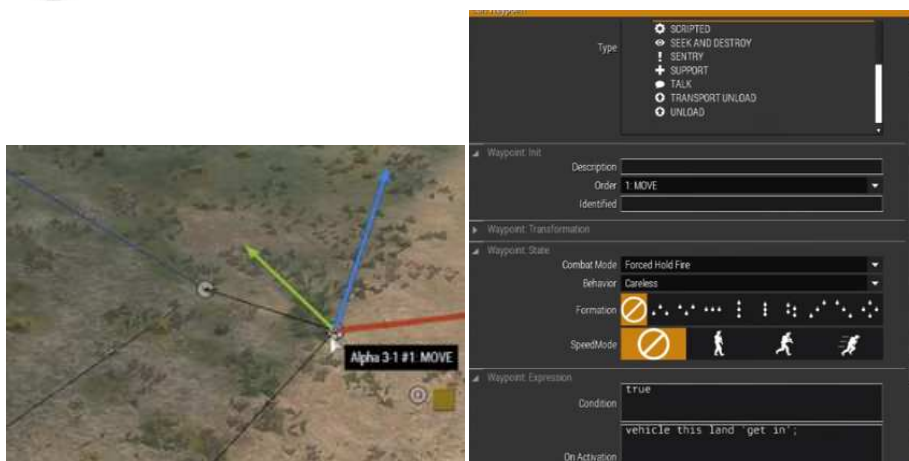
Procedura:

1. Creare l'unità elicottero in una posizione e assegnargli un nome, es "heli1".
2. Cliccare sul simbolo dell'elicottero e creare un waypoint ("muovere") accanto ad esso. Questo serve ad attivare l'elitransporto tramite trigger.
Impostarne il modo di combattimento: combattimento (aprirà il fuoco se serve)
Comportamento: safe
Velocità massima



3. Aggiungere un trigger e sincronizzarlo con il primo waypoint ("set attivazione waypoint"). Esso servirà a chiamare l'eli da parte dei giocatori.
Spuntare "ripetibile" per permetterne il richiamo più volte. Settare la condizione this ed l'attivazione tramite "Radio Alpha". In tal modo l'elicottero decollerà tramite radio dal menu attivabile con "\
4. Definire la rotta e mettere un waypoint nella posizione di estrazione del tipo "muovere"
Nel suo campo Attivazione mettere la seguente stringa

```
Vehicle this land 'get in';
```



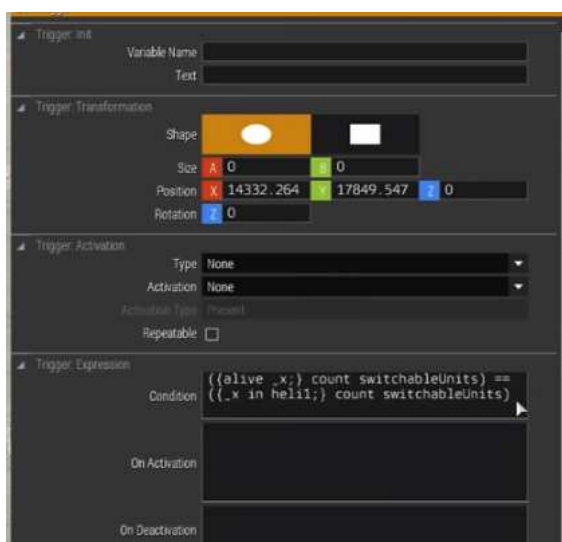
5. Mettere un altro waypoint vicino del tipo “muovere” ed un trigger.



6. Sincronizzare l’attivazione del waypoint con il trigger. Nella condizione del trigger mettere la seguente stringa

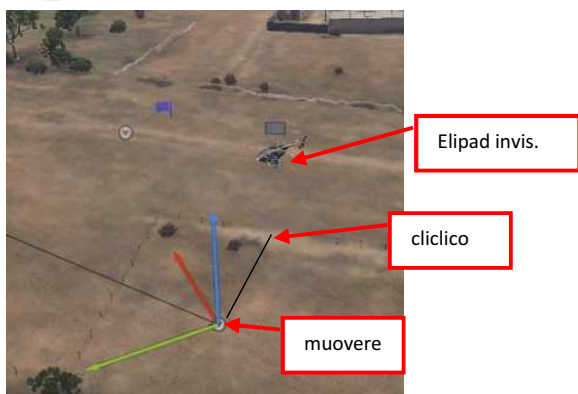
```
{{alive _x;} count playableUnits) == ({{_x in heli1;} count switchableUnits)
```

La prima conta i player vivi in gioco (se multiplayer nel primo count deve essere playableUnits altrimenti se single player switchableUnits) e li confronta con gli occupanti in quel momento sull’elicottero. Se sono uguali allora decolla



7. Nella posizione di ritorno creare un nuovo waypoint “muovere” e mettere nel suo campo init la stringa

```
Vehicle this land 'land';
```



8. Aggiungere un waypoint vicino del tipo "cliclico". NOTA: questo waypoint forse non è necessario, testarlo
9. Aggiungere anche un elipad accanto al waypoint per far atterrare l'elicottero esattamente in quella posizione

44.1.3 ESTRAZIONE 2 (DA TESTARE)

Estrazione con comando di ritorno quando i players (anche parziali) sono a bordo dell'elicottero.

EFFETTO:

L'elicottero verrà chiamato in radio alpha (attivabile tramite "\") ed atterrerà nella punto stabilito. Una volta a bordo si potrà ordinare il ritorno a casa tramite il comando a comparsa "Helicopter GO Home!", attivabile dal menu con rotella del mouse. Non è necessario che tutti siano a bordo, proprio perché attivabile tramite comando.

Procedura:

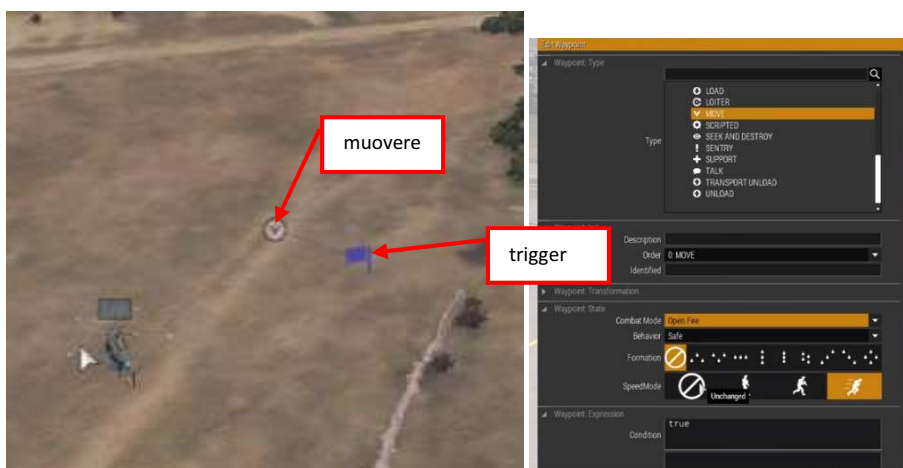
1. Creare l'unità elicottero in una posizione e assegnargli un nome, es "heli1". Nel campo init dell'oggetto elicottero inserire la stringa

```
this allowDamage false; // serve a rendere l'elicottero invincibile ai danni. Togliere tale riga se non necessaria
if (Call{triggerActivated zonaestraz}) then {this AddAction ["Helicopter GO Home!", {BW_HelicopterGo = true; publicVariable "BW_HelicopterGo";}];}
```

la prima istruzione lo rende invincibile. Rimuoverla se necessario.

La seconda attiva la funzione di decollo tramite menu se il trigger "zonaestraz" è attivato. Il trigger identifica la posizione dell'elicottero nella posizione dell'estrazione

2. Cliccare sul simbolo dell'elicottero e creare un waypoint ("muovere") accanto ad esso. Questo serve ad attivare l'elitransporto tramite trigger di chiamata via radio.
Impostarne il modo di combattimento: combattimento (aprirà il fuoco se serve)
Comportamento: safe
Velocità massima





3. Aggiungere un trigger e sincronizzarlo con il primo waypoint ("set attivazione waypoint"). Esso servirà a chiamare l'eli da parte dei giocatori.
Spuntare "ripetibile" per permetterne il richiamo più volte. Settare la condizione this ed l'attivazione tramite "Radio Alpha". In tal modo l'elicottero decollerà tramite radio dal menu attivabile con "\"
4. Definire la rotta voluta e mettere un waypoint nella posizione di estrazione del tipo "muovere"
Nel suo campo Attivazione mettere la seguente stringa

```
Vehicle this land 'get in';
```



5. Mettere un helipad invisibile dove atterra l'elicottero proprio a ridosso del waypoint al punto precedente
6. Mettere un trigger con dimensione che contiene l'elipad. Chiamare questo trigger "zonaestraz".
Impostare presenza Blufor e mettere la stringa seguente nella condizione

```
(helil in thisList);
```

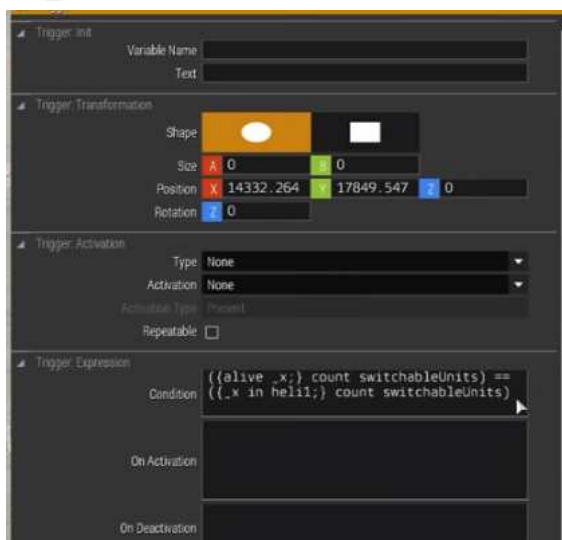
si attiva solo se l'elicottero heli1 è presente. Il comando "Helicopter GO Home!" è attivabile solo se tale trigger è soddisfatto

7. Mettere un altro waypoint vicino del tipo "muovere" ed un trigger.



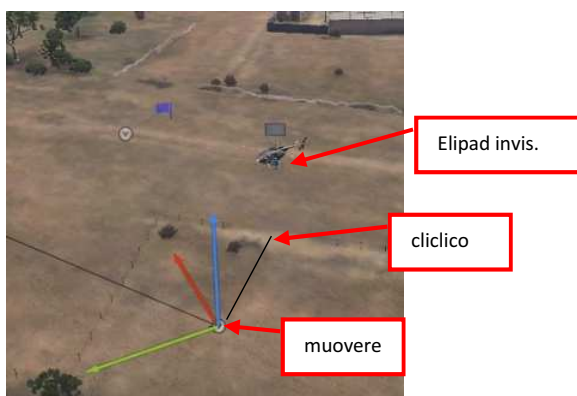
8. Sincronizzare l'attivazione del waypoint con il trigger. Nella sua condizione mettere la seguente stringa

```
BW_HelicopterGo;
```

9. Nella posizione di ritorno creare un nuovo waypoint “muovere” e mettere nel suo campo init la stringa

```
Vehicle this land 'land'; if (isServer) then {BW_HelicopterGo = false; publicVariable 'BW_HelicopterGo'}
```



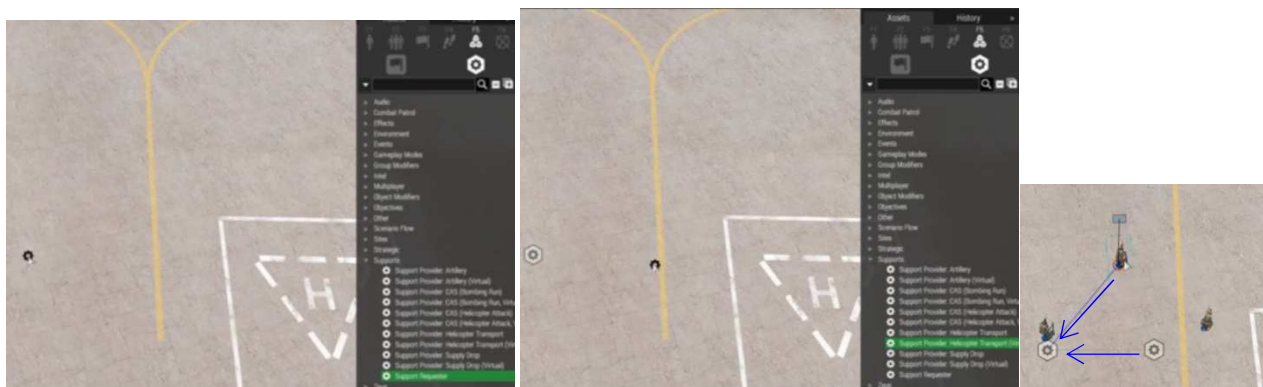
10. Aggiungere un waypoint vicino del tipo “cliclico”. NOTA: questo waypoint forse non è necessario, testarlo

11. Aggiungere anche un elipad accanto al waypoint per far atterrare l’elicottero esattamente in quella posizione

44.1.4 ESTRAZIONE 3 (VIA MODULI) - AGGIUNGERE UN TRASPORTO CON ELICOTTERO VIRTUALE O FISICO

AGGIUNGERE UN TRASPORTO CON ELICOTTERO VIRTUALE

1. Mettere giù il modulo Systems > Modules > Supports > Support Requester {SupportRequester}
2. Imposta i limiti che desideri applicare
3. Mettere giù il modulo Systems > Modules > Supports > Support Provider: Helicopter Transport (Virtual) {SupportProvider_Virtual_Transport}
4. Sincronizzare i due Moduli insieme
5. Sincronizzare ogni unità che sarà in grado di richiedere il trasporto al Support Requester {SupportRequester} Module



L'unità sincronizzata sarà in grado di chiamare il trasporto in qualsiasi punto in mappa e farsi trasportare dove vuole.

AGGIUNGERE UN TRASPORTO CON ELICOTTERI FISICO

1. Mettere giù il modulo Systems > Modules > Supports > Support Requester {SupportRequester}
2. Imposta i limiti che desideri applicare
3. Mettere giù il modulo Systems > Modules > Supports > Support Provider: Helicopter Transport {SupportProvider_Transport}
4. Sincronizzare i due Moduli insieme
5. Posizionare gli elicotteri da trasporto sulla mappa
6. Sincronizza ciascun elicottero con il Support Provider: Helicopter Transport {SupportProvider_Transport} Module
7. Sincronizzare ogni unità che sarà in grado di richiedere il trasporto al modulo Support Requester {SupportRequester} Module

44.2 ATTIVAZIONE VIA RADIO

Esempio di come attivare un elitransporto in posizione prefissata usando solamente i comandi radio.

1. Creare l'unità elicottero (o più unità) in una posizione A (per esempio una base).
2. Dare all'unità (o a ciascuna di esse) il proprio waypoint di "attesa" proprio accanto ad essa.
3. Creare un nuovo waypoint da lì come "Land" ("atterra") dove si vuole (posizione B). Si deve scegliere una posizione e aggiungere un eli pad invisibile. Questo li farà atterrare direttamente su di esso.
4. Quindi un altro waypoint di "attesa" accanto a B.
5. Quindi si dovrebbe avere una catena di "attesa", "atterrare" e "attesa".
6. Impostare un nuovo waypoint con "Scarica trasporto" all'indietro alla posizione A.
7. Fatto ciò, creare due trigger. Ognuno di essi sarà Tipo: Salta waypoint.
8. Dare a ciascuno l'opzione di attivazione Radio Alpha / Bravo / ecc ...
9. Sincronizzali con ciascuno dei waypoint di attesa precedenti.
10. Denominarli rispettivamente "Richiedi medevac" (o "richiedi elitransporto") e "Prendi vittima" (o "elitransporto in base").

Una volta in gioco, premere "\ " per avere il menu classico con i comandi via radio da selezionare. Ora si può chiamare un medevac (o richiedere un elitransporto) in qualsiasi punto si voglia, farlo aspettare fino a quando si carica quante persone vuole, quindi rimandarlo alla base. Nessuna codifica, nessuno script, ma solo tramite comandi alla radio.

Nota: la posizione B è prefissata in mappa, nel senso che l'elicottero andrà solo dove indicato e non in qualsiasi punto della mappa dove può essere il giocatore.



45 CREARE UN EFFETTO DI UNO SCHIANTO D'ELICOTTERO

Di seguito come creare uno schianto a sorpresa di un elicottero con a bordo dei players, senza che ovviamente muoiano tutti.

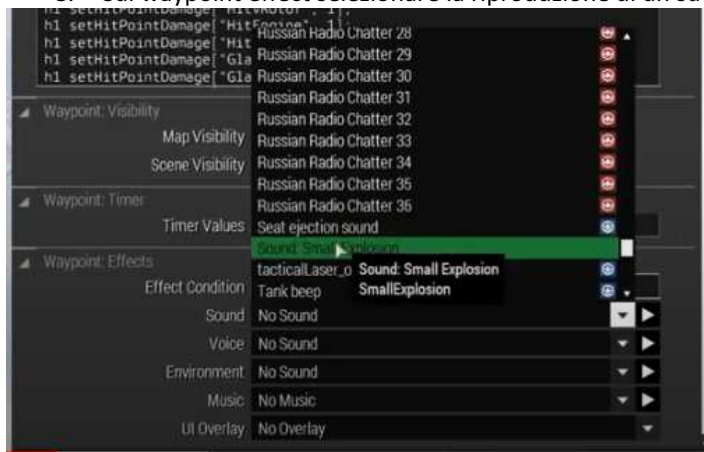
1. Piazzare un elicottero in volo e chiamarlo per esempio "h1"
2. Assegnargli un pilota e copilota e demninarli per esempio "h1d" e "h1d_1"
3. Mettervi a bordo i players (vedi anche §58)
4. Impostare i waypoints della rotta
5. L'ultimo waypoint va impostato come "scaricare veicolo". Questo è quello relativo allo schianto. Selezionarlo su un'area libera e leggermente distante dal penultimo
6. Impostare la velocità "full" e comportamento accorto



7. Selezionare il waypoint dove si vuole inizio i problemi all'elicottero e nel suo campo attivazione mettere le stringhe

```
h1 setHitPointDamage["HitVRotor", 1];  
h1 setHitPointDamage["HitEngine", 1];  
h1 setHitPointDamage["HitHull", 0.5];  
h1 setHitPointDamage["Glass1", 0.5];  
h1 setHitPointDamage["Glass2", 0.5];  
h1 setHitPointDamage["Glass3", 0.5];  
h1 setHitPointDamage["Glass4", 0.5];  
h1 setHitPointDamage["Glass5", 0.5];  
h1 allowDamage false;  
h1d sideChat "Merda, siamo stati colpiti!";  
h1d sideChat "Proverò ad atterrare.";
```

8. Sul waypoint effect selezionare la riproduzione di un suono d'esplosione





Si potrebbe mettere anche un'istruzione che evita il danno all'equipaggio dell'eli tramite la stringa

```
{  
x_ allowDamage false;  
} forEach(crew h1);
```

Mettendo questa però l'AI potrebbe non atterrare lì dove stabilito, soprattutto se è un'area stretta e con alberi. Evitandola invece l'AI tenterà un atterraggio d'emergenza anche se si rischia un ferimento dell'equipaggio.

9. Mettere giù un trigger e mettere nella condizione

```
isTouchingGround h1;
```

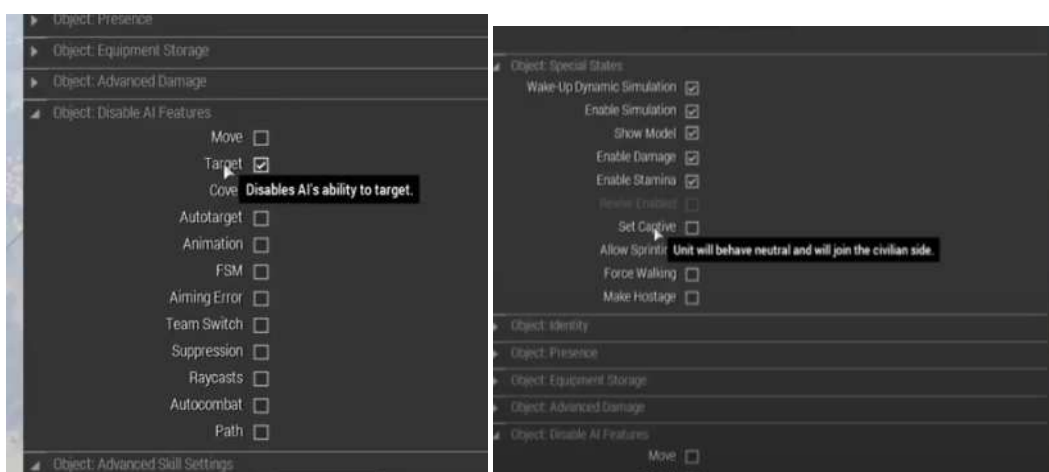
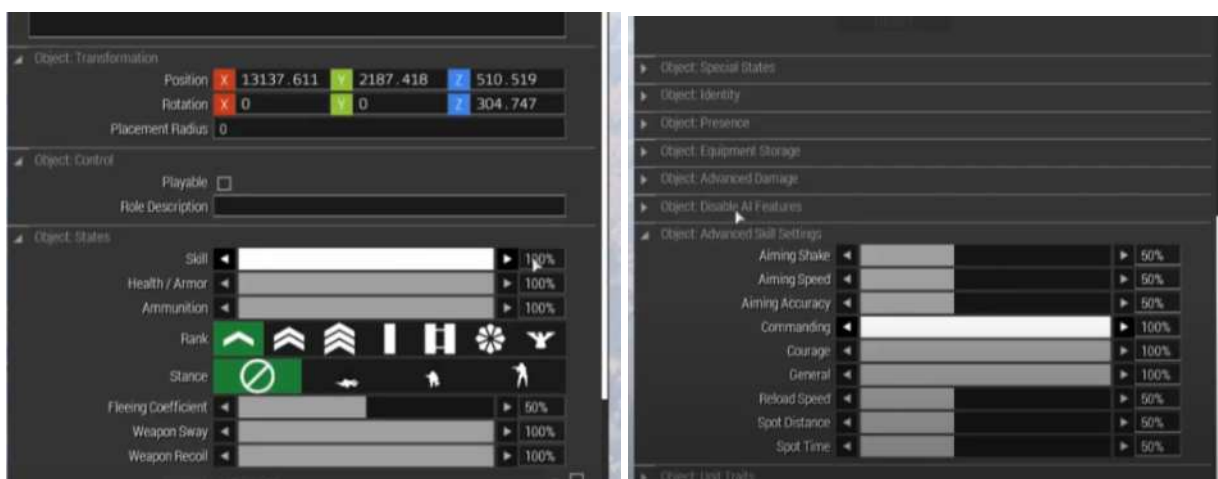
ed all'attivazione

```
// uccido i piloti  
h1D_1 setDamage 1;  
h1D setDamage 1;
```

all'attivazione, se impostato il "allowDamage false" di cui sopra, si deve aggiungere la sua riattivazione con la stringa seguente

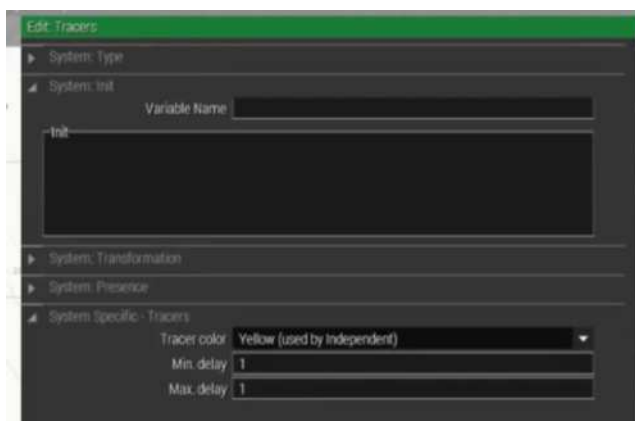
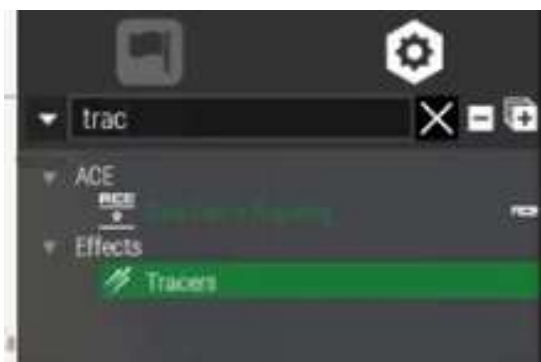
```
{  
x_ allowDamage true;  
} forEach(crew h1);
```

10. Negli attributi dei piloti aumentarne al massimo le loro skill ed il coraggio. Inoltre disabilitare il target nelle features dell'AI in modo tale da far sì che non tentino di ingaggiare nemici a terra. In aggiunta levare la spunta a set captive in modo tale che non ricevino fuoco di ritorno dai nemici





11. Per creare un effetto di fuoco nemico si può usare il modulo dei flare e sincronizzarlo a vari trigger con blufor presente, in modo tale da attivarli al passaggio dell'elicottero.





46 ATTACCO AD ONDATE SEQUENZIALI

Di seguito come creare un attacco ad ondate in sequenza.

1. Creare i gruppi di ondate e nominarli.
Es: prima ondata composta dai gruppi "ondata1gruppo1", "ondata1gruppo2" ...
Seconda ondata composta dai gruppi "ondata2gruppo1", "ondata2gruppo2" ...



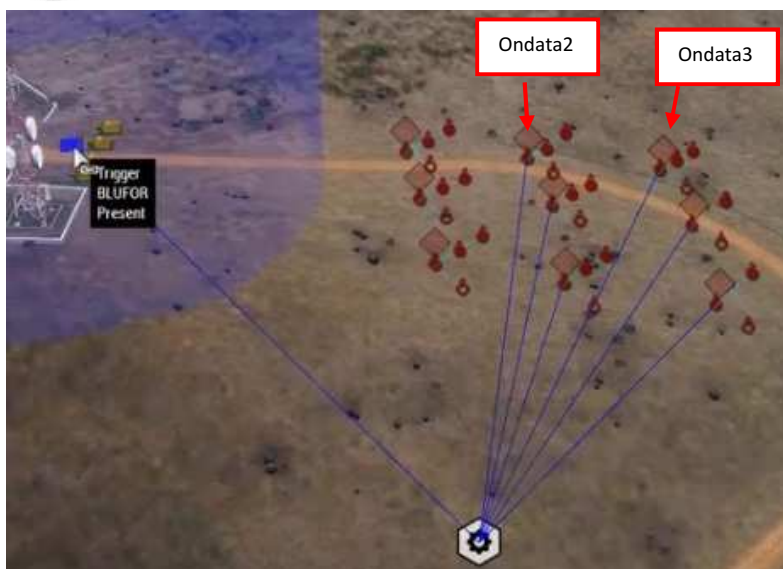
2. Mettere giù il modulo "modificatore di oggetti/mostra-nascondi" che è settato su "nascondi unità".



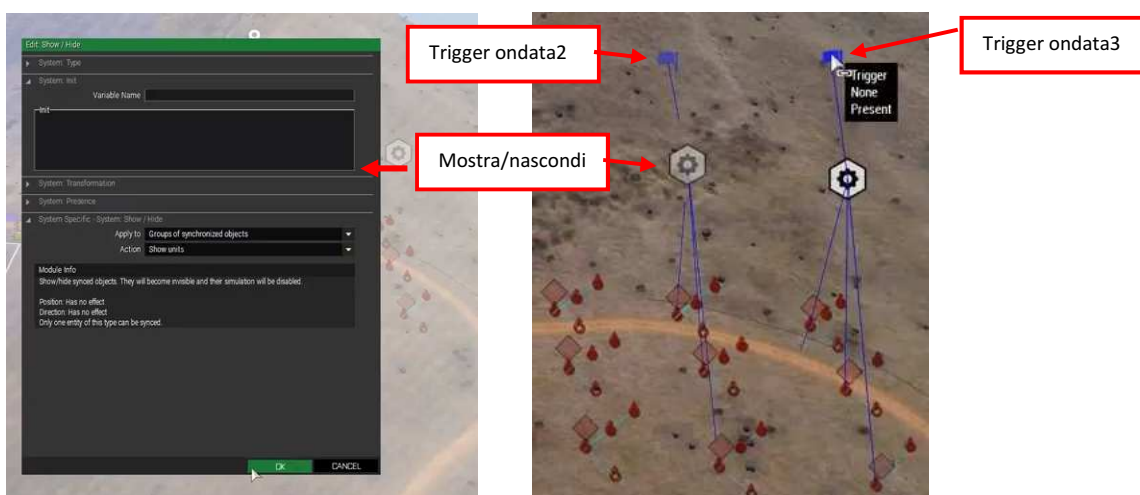
3. Mettere giù un trigger con blufor presenti che andrà ad attivare il modulo nascondi unità seguente. Nota: questo in realtà potrebbe anche essere omissso in quanto basta il modulo precedente da solo per nascondere le unità



4. Sincronizziamo il modulo con le ondate successive alla prima (con i capisquadra di ogni unità) e con il trigger.



5. Sincronizziamo il modulo con le ondate successive alla prima (con i capisquadra di ogni unità) e con il trigger di attivazione
6. Mettere giù altri due moduli mostra/nascondi ed impostarli questa volta su “mostra unità”. Questi serviranno a far mostrare le ondate successive all’attivazione di altri due trigger che cremerò di seguito



1. Creare due trigger i quali attiveranno i due moduli alla condizione di eliminazione delle ondate precedenti
2. Sincronizzare i due due moduli con i capisquadra delle ondate e con uno dei trigger (vedi figura punto 6)
3. Sul trigger dell’ondata 2 metteremo la stringa seguente

```
{ { alive _x } count units Ondata1Gruppo1 == 0 } && ( { alive _x } count units Ondata1Gruppo2 == 0 ) && ( { alive _x } count units Ondata1Gruppo3 == 0 )
```

Questa conta le morti dei gruppi dell’ondata1 e attiverà quindi il mostra dell’ondata2 una volta che la 1 è tutta giù

Sul trigger dell’ondata 3 metteremo la stringa seguente

```
{ { alive _x } count units Ondata2Gruppo1 == 0 } && ( { alive _x } count units Ondata2Gruppo2 == 0 ) && ( { alive _x } count units Ondata2Gruppo3 == 0 )
```

Questa conta le morti dei gruppi dell’ondata2 e attiverà quindi il mostra dell’ondata3 una volta che la 2 è tutta giù

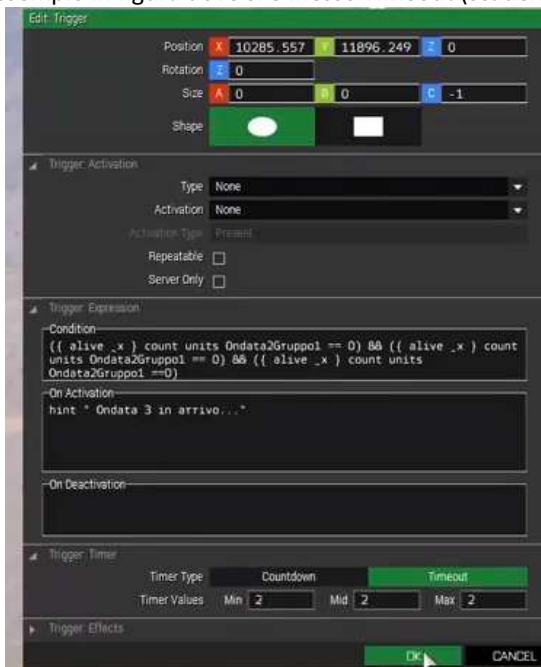
Si possono aggiungere più ondate con lo stesso metodo e più gruppi all’interno di ogni ondata.



Nota: se si vuole mettere un messaggio di avviso che compare ad ogni ingresso dell'ondata si può mettere all'attivazione del trigger dell'ondata la stringa

Hint "Ondata 2 in arrivo o qualsiasi altro testo..."

Si può inoltre ritardare la comparsa dell'ondata inserendo un tempo di ritardo dell'attivazione del trigger, come per esempio in figura dove si è messo Timeout (scadenza) un tempo di 2 o più secondi



47 NASCONDERE UNITÀ E FARLE APPARIRE CON TRIGGER

Per nascondere delle unità in gioco e farle apparire quando serve, per esempio tramite attivazione di un trigger, si può far riferimento al §0 precedente. In sostanza:

1. Creare un'unità e assegnargli un nome nel campo variabile.
2. Mettere giù il modulo "modificatore di oggetti/mostra-nascondi" e impostarlo su "nascondi unità"
3. Sincronizzare il modulo con l'unità. Sincronizzare con il caposquadra se si vuole nascondere tutto il gruppo



In questo modo l'unità è nascosta in gioco. Per farla apparire quando serve operare come segue:

4. Mettere giù un trigger per esempio con blufor presenti (oppure attivabile tramite radio alfa...) che andrà ad attivare il modulo mostra unità successivo.
5. Mettere giù il modulo "modificatore di oggetti/mostra-nascondi" e impostarlo su "mostra unità"
6. Sincronizzare questo modulo con l'unità (o il caposquadra) e con trigger





48 CREARE UN ATTACCO SUICIDA CON ESPLOSIVO

48.1 PRIMO METODO

Per creare un'unità civile che si fa esplodere operare come segue:

1. Inserire un'unità civile e nominarla per esempio "bomber01"
2. Inserire la stringa nel suo init

```
this addvest "V_HarnessOGL_gry"; bomber01 attachTo [bomber01,[0,0,0]]
```

questo assegna all'unità un corpetto e gli "attacca" una bomba



V_HarnessOGL_gry

3. Creare un trigger qualsiasi d'attivazione (blufor presente per esempio) che faccia esplodere il civile. Nel campo attivazione inserire:

```
bomb="R_60mm_HE" createVehicle[(getPos bomber01 select 0),(getPos bomber01 select 1),0];
```

questa fa esplodere una granata 60mm HE

48.2 SECONDO METODO

Altro metodo per creare un'unità civile che si fa esplodere.

File scaricabili da <http://www.armaholic.com/page.php?id=20562>. Usare tali file nella posizione della cartella missione.

Procedura:

1. Inserire un'unità civile e nominarla per esempio "bomber01"
2. Creare un trigger qualsiasi d'attivazione (blufor presente per esempio) che faccia esplodere il civile. Nel campo attivazione inserire:

```
[bomber01,[CIVILIAN,WEST,EAST,RESISTANCE],"grenadeHand"] execVM "suicideBomber.sqf"; \\The unit you want to be the bomber, the sides you want the bomber to attack, classname of explosive you want to use
```

Dove bomber01 è l'unità, [...] la fazione che attaccherà e "grenadeHand" il tipo di esplosivo. Richiama poi il file suicideBomber.sqf

Contenuto file suicideBomber.sqf scaricato dal link

```
_bomber = _this select 0;
_targetSide = _this select 1;
_explosiveClass = _this select 2;
_info = [_bomber,_targetSide,_explosiveClass];
_runCode = 1;

while {alive _bomber && _runCode == 1} do
{
    _nearUnits = nearestObjects [_bomber,["CAManBase"],70];
    _nearUnits = _nearUnits - [_bomber];
    {
        if(!(side _x in _targetSide)) then {_nearUnits = _nearUnits - [_x]};
    } forEach _nearUnits;
    if(count _nearUnits != 0) then
    {
        _pos = position (_nearUnits select 0);
        _bomber doMove _pos;
        waitUntil {_bomber distance _pos < 15};
        if(_bomber distance (_nearUnits select 0) < 15)
        exitWith
        {
            _runCode = 0;
            _explosive = _explosiveClass createVehicle (position _bomber);
        }
    }
}
```



```
[_bomber,_explosive] spawn {_bomber = _this select 0; _explosive = _this select 1;
sleep 2; _bomber say3D "shout"; _explosive setDamage 1; _bomber addRating -1000000;};
[_explosive,_bomber] spawn {_explosive = _this select 0; _bomber = _this select 1;
waitUntil {!alive _bomber}; deleteVehicle _explosive;};
    if(round(random(1)) == 0) then
    {
        _explosive attachTo [_bomber,[-0.02,-0.07,0.042],"rightHand"];
    }
    else
    {
        _explosive attachTo [_bomber,[-0.02,-0.07,0.042],"leftHand"];
    }
};
};
sleep 1;
};
```

L'attentatore suicida correrà verso l'unità più vicina del lato specificato e si farà esplodere usando il nome l'esplosivo assegnatogli.

Se il kamikaze viene ucciso prima che l'esplosivo esploda, l'esplosivo viene eliminato.

3. Se è già definito un file description.ext della missione inserire la stringa aggiuntiva. Se non già definito usare il description.ext scaricato dal link.

```
class CfgSounds
{
    #include "Sounds.hpp"
};
```

Contenuto file sounds.hpp scaricato dal link (non modificarlo)

```
class shout
{
    name = "";
    sound[] = {"\shout.ogg", 1, 1, 100};
    titles[] = {};
};
```

Richiama il file audio shout.ogg (scaricato dal link) da mettere nella cartella missione

Il file init.sqf contenuto dal download dal link in realtà non serve metterlo nella cartella missione. Dopo averlo testato si puo' non copiarlo nella cartella. Contenuto file init.sqf

```
/*
    Suicide Bomber Script - by Stealthstick/Zooloo75
*/
[bomber,[CIVILIAN,WEST,EAST,RESISTANCE],"grenadeHand"] execVM "suicideBomber.sqf"; //The unit you want
to be the bomber, the sides you want the bomber to attack, classname of explosive you want to use
```

48.3 TERZO METODO

Altro metodo per creare un kamikaze o spawnarne un gruppo in un'area.

File scaricabili da <http://www.armaholic.com/page.php?id=33096>

Procedura:

1. Nel file description.ext all'interno della cartella missione (crearlo nel caso), inserire le stringhe delle classi dei suoni

```
class CfgSounds
{
    // List of sounds (.ogg files without the .ogg extension)
    sounds[] = {gun1, gun2, gun4, gun4, gun5, ringtone, urale, allahu_akbar ,geiger, sandstorm, prayer_1, prayer_2, expozie,
    alarma_militar, intro_music, outro_music, NoSound};

    // Definition for each sound
    class 01
    {
        name = "01";
        sound[] = {"\sound\01.ogg", 0.8, 1};
        titles[] = {1, ""};
    };
    class 02
    {
        name = "02";
        sound[] = {"\sound\02.ogg", 0.8, 1};
    };
};
```



```
};
titles[] = {1, ""};
class 03
{
    name = "03";
    sound[] = {"\sound\03.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 04
{
    name = "04";
    sound[] = {"\sound\04.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 05
{
    name = "05";
    sound[] = {"\sound\05.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 06
{
    name = "06";
    sound[] = {"\sound\06.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 07
{
    name = "07";
    sound[] = {"\sound\07.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 08
{
    name = "08";
    sound[] = {"\sound\08.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 09
{
    name = "09";
    sound[] = {"\sound\09.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class 099
{
    name = "099";
    sound[] = {"\sound\099.ogg", 0.8, 1};
    titles[] = {1, ""};
};
class NoSound
{
    name = "NoSound";
    sound[] = {"", 0, 1};
    titles[] = {0, ""};
};
};
```

2. Copiare la cartella scaricata \Scripts e \sound nella cartella missione

Contenuto cartella Scripts:

- kam.sqf
- kam_group.sqf

Contenuto cartella sounds:

- files: 01.ogg, 02.ogg, 03.ogg, 04.ogg, 05.ogg, 06.ogg, 07.ogg, 08.ogg, 09.ogg, 099.ogg

• Creare un kamikaze singolo

3. Aggiungere un civile e mettere una stringa di questo tipo nel suo init:

```
nul = [unit_name,trigger_distance,sound,chase_players] execvm "Scripts\kam.sqf";
```

where:

unit - string, is the name of the kamikaze unit
trigger_distance - number, distance from targeted player at which the kamikaze will blow himself up
sound - boolean, if is true a random sound from a given array is played, if is false kamikaze will not shout
chase_players - boolean, if true the kamikaze unit will chase players, otherwise will be pasive and blow himself up when a random player is in its proximity

Esempio 1:

```
nul = [this,50,true,true] execvm "Scripts\kam.sqf";
```

- kamikaze will blow up at 50 meters from target
- kamikaze will yell
- Kamikaze will chase players

Esempio 2:



```
nul = [this,50,false,false] execvm "Scripts\kam.sqf";
```

- kamikaze will blow up at 50 meters from target
- kamikaze will NOT yell
- Kamikaze will NOT chase players

File kam.sqf in \Scripts richiamato (NON va modificato):

```
// by ALIAS
// nul = [this,distance_trigger] execvm "Scripts\kam.sqf";
private ["_soundk","_chase_players","_dist","_kamm","_expl1","_expl2","_expl3","_bombk","_list_p","_grp","_strigat","_check_p"];
if (!isServer) exitWith {};
[] spawn
{
    if (isNil "unitus") then
    {
        while {true} do
        {
            _allunits = [];
            if (alive _x) then
            {
                [_x, "NoVoice"], "setSpeaker", true] call BIS_fnc_MP;
                _allunits pushBack _x;
            }
            } foreach (if (isMultiplayer) then {playableUnits} else {switchableUnits});
            unitus = _allunits call BIS_fnc_selectRandom;
            publicVariable "unitus";
            sleep 60;
        };
    };
};
waitUntil {!isNil "unitus"};
_kamm = _this select 0;
_dist = _this select 1;
_soundk = _this select 2;
_chase_players = _this select 3;
_kamm enableFatigue false;
_grp = createGroup civilian;
[_kamm] joinSilent _grp;
_grp setBehaviour "AWARE";
_grp setCombatMode "RED";
_grp setSpeedMode "FULL";
_expl1 = "DemoCharge_Remote_Ammo" createVehicle position _kamm;
_expl1 attachTo [_kamm, [-0.1, 0.1, 0.15], "Pelvis"];
_expl1 setVectorDirAndUp [ [0.5, 0.5, 0], [-0.5, 0.5, 0] ];
_expl2 = "DemoCharge_Remote_Ammo" createVehicle position _kamm;
_expl2 attachTo [_kamm, [0, 0.15, 0.15], "Pelvis"];
_expl2 setVectorDirAndUp [ [1, 0, 0], [0, 1, 0] ];
_expl3 = "DemoCharge_Remote_Ammo" createVehicle position _kamm;
_expl3 attachTo [_kamm, [0.1, 0.1, 0.15], "Pelvis"];
_expl3 setVectorDirAndUp [ [0.5, -0.5, 0], [0.5, 0.5, 0] ];
while {alive _kamm} do
{
    if (_soundk) then
    {
        _strigat = ["NoSound","01","NoSound","02","NoSound","04","NoSound","05","NoSound","06","NoSound","07","NoSound","08","NoSound","09","NoSound","099","NoSound"] call BIS_fnc_selectRandom;
        [_kamm,[_strigat,300]] remoteExec ["say3d"];
    };
    if (_chase_players) then {{_x doMove getPos unitus;} forEach units _grp};
    _check_p = [];
    _check_p = allPlayers inAreaArray [getPosATL _kamm,_dist,_dist,0,false,-1];
    if (count _check_p > 0) then
    {
        _strigat = ["01","02","04","05","06","07","08","09","099"] call BIS_fnc_selectRandom;
        [_kamm,[_strigat,300]] remoteExec ["say3d"];
        sleep 1 + random 2;
        _kamm setVelocity [random 3, random 3, 5+random 15];
        sleep 0.01;
        _bombk = "M_Mo_82mm_AT_LG" createVehicle (getPos _kamm);
        sleep 0.01;
        _kamm setDamage 1;
        deleteVehicle _expl1;
        deleteVehicle _expl2;
        deleteVehicle _expl3;
    };
    sleep 2+random 3;
};
if (!isNull _expl1) then
{
    _strigat = ["01","02","04","05","06","07","08","09","099"] call BIS_fnc_selectRandom;
    [_kamm,[_strigat,300]] remoteExec ["say3d"];
    _kamm setVelocity [random 3, random 3, 5+random 15];
    sleep 0.1;
    _bombk = "M_Mo_82mm_AT_LG" createVehicle (getPos _kamm);
    _kamm setDamage 1;
    deleteVehicle _expl1;
    deleteVehicle _expl2;
    deleteVehicle _expl3;
};
```



- Creare un gruppo di kamikaze su un'area
 3. Creare un marker area e nominarlo, es "area-kamikaze"
 4. Inserire una stringa di questo tipo nel file init.sqf all'interno della cartella della missione (crearlo nel caso):

```
// SPAWNING A CUSTOM NUMBER OF KAMIKAZE AT A GIVEN MARKER POSITION
null = [MARKER_name,kamikaze_number,sound,chase_players,trigger_distance] execvm
"Scripts\kam_group.sqf";

where:
MARKER_name - string, is the name of the marker area
Kamikaze_number - number og kamikazes to be spawned in the area
sound - boolean, if is true a random sound from a given array is played, if is false
kamikaze will not shout
chase_players - boolean, if true the kamikaze unit will chase players, otherwise will be pasive
and blow himself up when a random player is in its proximity
trigger_distance - number, distance from targeted player at which the kamikaze will blow himself up
```

Esempio:

```
null = ["area-kamikaze",10,true,true,10] execvm "Scripts\kam_group.sqf";
```

Il tipo di civili spawnati dallo script è contenuto nel file kam_group.sqf che di default è il seguente:

```
// by ALIAS
private ["_k_unit","_grp","_ksound","_chasep","_distep"];

if (!isServer) exitWith {};

_origin = _this select 0;
_no = _this select 1;
_ksound = _this select 2;
_chasep = _this select 3;
_distep = _this select 4;

if (_no>0) then
{
    _nr_k = 0;
    while {_nr_k<_no} do
    {
        _grp = createGroup civilian;
        _type_k =
["C_Man_casual_6_F_afro","C_man_1_1_F","C_Man_casual_1_F_asia","C_man_1_3_F","C_man_polo_1_F","C_man_polo_1_F_afro","C_Man_casual_3_F_asi
a","C_man_polo_1_F_asia","C_man_polo_2_F","C_man_polo_2_F_afro","C_man_polo_2_F_euro","C_Man_casual_4_F_euro","C_man_polo_3_F","C_man_pol
o_3_F_afro","C_Man_casual_5_F_euro","C_man_polo_3_F_asia","C_man_polo_4_F","C_man_polo_4_F_afro","C_man_polo_4_F_euro","C_man_polo_4_F_as
ia","C_man_polo_5_F","C_man_polo_5_F_afro","C_man_polo_5_F_euro","C_man_polo_5_F_asia","C_man_polo_6_F","C_man_polo_6_F_afro","C_man_polo
_6_F_euro","C_man_polo_6_F_asia","C_man_p_fugitive_F","C_man_p_fugitive_F_afro","C_man_p_fugitive_F_euro","C_man_p_fugitive_F_asia","C_ma
n_hunter_1_F","C_Orestes","C_Nikos","C_Nikos","C_man_sport_2_F_afro","C_man_sport_3_F_asia"] call BIS_fnc_selectRandom;
        _type_k createUnit [getMarkerPos _origin, _grp, "[this,_distep,_ksound,_chasep] execvm 'Scripts\kam.sqf'",1];
        _nr_k = _nr_k +1;
        sleep 1;
    };
};
```

in grassetto-rosso i civili che lo script userà selezionandoli in modo random.

Se si vuole per esempio usare civili diversi di una mod in particolare, sostituire le classnames dei civili a quelle in rosso. Per civili Takistan per esempio si può usare:

```
// by ALIAS
private ["_k_unit","_grp","_ksound","_chasep","_distep"];

if (!isServer) exitWith {};

_origin = _this select 0;
_no = _this select 1;
_ksound = _this select 2;
_chasep = _this select 3;
_distep = _this select 4;

if (_no>0) then
{
    _nr_k = 0;
    while {_nr_k<_no} do
    {
        _grp = createGroup civilian;
        _type_k =
["LOP_Tak_Civ_Man_01","LOP_Tak_Civ_Man_02","LOP_Tak_Civ_Man_04","LOP_Tak_Civ_Man_05","LOP_Tak_Civ_Man_06","LOP_Tak_Civ_Man_07","LOP_Tak_C
iv_Man_08","LOP_Tak_Civ_Man_09","LOP_Tak_Civ_Man_10","LOP_Tak_Civ_Man_11","LOP_Tak_Civ_Man_12","LOP_Tak_Civ_Man_13","LOP_Tak_Civ_Man_14",
"LOP_Tak_Civ_Man_15","LOP_Tak_Civ_Man_16"] call BIS_fnc_selectRandom;
        _type_k createUnit [getMarkerPos _origin, _grp, "[this,_distep,_ksound,_chasep] execvm 'Scripts\kam.sqf'",1];
        _nr_k = _nr_k +1;
        sleep 1;
    };
};
```



49 CREARE UN ANIMALE CHE ATTACCA (DA TESTARE)

49.1 PRIMO METODO (ATTACCO E PATTUGLIAMENTO CON CANE)

Questo metodo sfrutta lo script che può essere scaricato all'indirizzo <http://www.armaholic.com/page.php?id=35026>

Description:

vDog Scripts Main Features:

- Easily add patrolling ai Dogs to allied, enemy or even playable Arma 3 units.
- Easily add ai Feral Dogs who will roam a trigger area and viciously attack any unit who enters their territory.
- Lightweight Scripts. (Your FPSs will thank you!).
- Multiplayer Compatible!
- Fully featured script also comes with sound effects. (Dog barks, growls, cries etc).
- Since version 1.3 the scripts also support the ACE 3 damage model!
- Even thou the scripts were made with Multiplayer and Performance in mind they work in SINGLE PLAYER just as well.

Due modalità:

- vDogFeral: I cani in un'area attaccano chiunque vi entri
- vDog Patrol: i cani seguono un'unità

vDogFeral

- Feral dogs do not require a unit to work like the patrol Dogs, just a trigger created via Editor or via Script.
- The Area of the trigger will be the Feral Dogs's Territory.
- You can set the trigger activation to any condition. Set condition to True to spawn the feral dogs on mission start.
- The dogs initially bark and point (reveal) their closest target. If the target unit gets closer the Dogs will then start running towards their prey and then sprint. A vicious attack on the victim follows.
- Unlike the patrol Dogs (that are trained to disarm and attack the legs of their targets) these FERAL Dogs do full damage to their victims and will kill anyone with 6 to 8 bites.

vDogFeral.sqf Usage:

1. Download the files
2. Move the scripts and sounds folders found inside the downloaded files to your mission folder.
3. Update your mission's description.ext file to include the contents of the DEMO MISSION's description.ext file or just move the file if you don't have it.
4. In the Eden Editor, on a trigger's on ACTIVATION field add:

```
null = [thisTrigger,3] execVM "scripts\vDogFeral.sqf";
```

So in this example:

3 is the number of Feral Dogs to be added inside the trigger area.

vDog Patrol

vDog Patrol Script Usage:

1. Download the vDog Patrol Demo Mission and open it with 7zip.
2. Move the scripts and sounds folders found inside the vDogPatrolv13.VR folder to your mission folder.
3. Update your mission's description.ext file to include the contents of the DEMO MISSION's description.ext file or just move the file if you don't have it.
4. In the Eden Editor, on a unit's Init field add:

```
null = [this,"Alsatian_Random_F",1] execVM "scripts\vDogPatrol.sqf";
```

So in this example:

this (is the name of the Unit who will get the patrol Dog companion).

Alsatian_Random_F is the kind of Dog that will be used. The script accepts two inputs right now: Alsatian_Random_F and Fin_random_F.

1 is the number of Dogs to be added under that unit's command. (Can be more than one but it is not suggested).

vDog Patrol Notes:



- The patrol dogs can be added to both Player Units or Ai Units but since this is a MP compatible light weight script there are no commands for the dogs. They still behave like well trained dogs even without manual commands.
- The dogs cannot kill enemy units but will disable them for a few seconds and will damage their legs. They are more like a detection tool rather than killing machines.
- The dogs initially bark and point (reveal) the closest enemy in a 100 meters radius. If the enemy unit gets closer than 75 meters the Dog will then start running towards its prey and then sprint. A vicious attack on the victim follows.
- The dog will only stop chasing selected targets if the enemy unit gets 120 mtrs away from the dog/dog owner or if its owner is dead. So if you want your dog to return to you or stop attacking get 120 mtrs away from the enemy.
- Patrol Dogs are not attacked by the Arma 3 Ai (Arma 3 engine limitation and the work arounds are not compatible with the lightweight nature of this script).
- Still I have set up the script in a way that if the Dog Owner dies then the dog stops being aggressive to compensate the Ai not attacking them. (The Ai will attack the owner and if they kill him then it is like they have killed the Dog).
- If a dog owner gets very far away from the Dog the Dog will teleport back to its owner only if he is not inside a vehicle. This also simulates the dog getting in the vehicle with its owner.

49.2 SECONDO METODO

There is no standard way of making animals attack. You can force it though.

Here's how:

1. put down a marker, name it *goatSpawn*.
2. put down your unit, in its init line put:

```
goat = createAgent ['Goat_random_f', getMarkerPos 'goatSpawn', [], 0, 'NULL']; goat addRating -10000;
```

this spawns a goat at the marker and makes it renegade, so units will attack it.

3. put down one trigger with a short timer, set the condition to

```
!isNil 'goat'
```

and the on activation to

```
goat moveTo (position player);
```

this makes the goat charge the player in ALL ITS FURY!

4. put down the second trigger with the condition

```
(player distance goat) < 5;
```

5. and activation

```
player setDamage 1;
```

this kills the player at a certain distance from the goat (change 1 to <1 value to cause only damages). You can use the `setDamage` command, I personally find the results of

```
nul = "M_PG_AT" createVehicle (getPos goat)
```

far more entertaining (molto più divertente.)

That is the idea behind it, it's clunky and the `moveTo` command sometimes fails so you have to refresh the goats mind in the debug. The full solution to it should of course use scripts.

Altri script trovati

Script per far causare danni ad un'unità da parte di un animale sfruttando la sua distanza dall'unità:

```
if (player distance dog <2) then {player setDamage (getDammage palyer) 0.1};
```

```
myDog = group myPlayer createUnit ["Pastor", position myPlayer, [], 0, "FORM"];  
enableRadio false;
```

This creates an AI working dog that follows your commands like `move` and `stuff`.

The `doMove` command works to in a trigger, so should be OK in a local script even in MP.

The `enableRadio false` is useful if the dog is in your group, otherwise he replies "Roger" all the time, which increases the fun factor a bit too much in serious missions



50 ESEMPI DI TITOLI DI TESTO A SCOMPARSA - TITLE TEXT EXAMPLES

La funzione associata è titleText

Esempi:

```
titleText ["This is written in white letters in the middle of your screen, fading in over the course of 3 seconds and fading away in 10 seconds or so","PLAIN", 3];
```

```
titleText ["This is written in white letters at the bottom of your screenfading away in 4 seconds because of the titleFadeOut command.,"PLAIN DOWN"]; titleFadeOut 4;
```

```
titleText ["This will turn your screen black and display this text slowly fading away after 8 seconds.,"BLACK IN",8];
```

```
titleText ["This will fade your screen to black while displaying this text and never come back, usually. Adding the titleFadeOut will make it fade back in, kind of, 20 seconds or so. It's wierd.,"BLACK OUT",8];titleFadeOut 4;
```

Quest'ultimo è molto suggestivo: lo schermo sfuma sul nero, compare la scritta e poi sfuma di nuovo a normale

51 TELEPORT

51.1 TELEPORT CON CLICK SU MAPPA

Questo script serve a teletrasportare il player su un punto qualsiasi della mappa. Scorrendo con la rotellina si fa teleport, vi si apre la mappa e cliccando sulla posizione prescelta si viene immediatamente teletrasportati al punto selezionato.

1. Create un file e chiamatelo teleport.sqf inseritelo nella cartella principale della mappa.
2. Inserire le seguenti stringhe ne file teleport.sqf

```
openMap true;
sleep 1;
hint "Click on map to teleport.";
onMapSingleClick "player setPos _pos; onMapSingleClick ' '; openMap false; true;";
waitUntil{!(visibleMap)};
```

3. Dall'editor nell'init dell'unità inserire, per ogni unità a cui volete dare questa opzione, la stringa

```
this addAction ["Teleport", "teleport.sqf", [], 1, false, true];
```

Nota: nella versione originale dello script c'era player al posto di this. Dopo averlo testato funziona solo con this e non con player

51.2 TELEPORT SU UN PUNTO CON ATTIVAZIONE DI UN TRIGGER

1. Aggiungere un trigger e nell'attivazione inserire la stringa

```
{_x setPos (getMarkerPos "nomedelmarker")} forEach thislist;
```

2. Aggiungere un marker in mappa e nominarlo nomedelmarker
3. Definire una condizione qualsiasi d'attivazione del trigger

51.3 TELEPORT SU UN PUNTO IDENTIFICATO DA UN OGGETTO O MARKER

Semplice script di teletrasporto tra due oggetti:

1. Inserire un oggetto B e definirne un nome, es "nomeoggettoB"
2. Definire un secondo oggetto A ed inserire la seguente stringa nel suo campo init

```
this addAction ["Teleport to B", {player setPos (getPos nomeoggettoB)}]
```

dove nomeoggettoB è il nome dell'oggetto punto finale.

Avvicinandosi all'oggetto A verrà data l'opzione di teletrasporto al punto identificato dall'oggetto B

Lo stesso si può fare identificando il punto B con un marker. In tal caso nel campo init dell'oggetto A bisognerà mettere

```
this addAction ["Teleport to B", {player setPos (getMarkerPos "nomedelmarkerB")}]
```

dove in nomedelmarkerB va sostituito il nome assegnato al marker

**Altra procedura:**

1. Creare un file sqf nella cartella della missione e nominarlo per esempio tele1.sqf
2. Aggiungere tale stringa nel file tele1.sqf

```
_tele = _this select 0;  
_caller = _this select 1;  
_caller setPos (getpos (Name of Invisible Heli Pad 1 or object1));
```

3. Creare un altro file sqf nella cartella e nominarlo per esempio tele2.sqf
4. Aggiungere tale stringa nel tele2.sqf

```
tele = _this select 0;  
_caller = _this select 1;  
_caller setPos (getpos (Name of Invisible Heli Pad 2 or object2));
```

5. Inserire un palo con bandiera (1) ed inserire la stringa nel suo init

```
this addaction ["Name of Location Here es Tele1", "tele1.sqf"];
```

6. Inserire un helipad invisibile vicino al palo 1 dove si vuole che compaiano le unità e chiamarlo "tele2"
7. Inserire un altro palo con bandiera (2) in una nuova posizione ed inserire la stringa nel suo init

```
this addaction ["Name of Location Here es Tele2", "tele2.sqf"];
```

8. Inserire un altro helipad invisibile vicino al palo 2 dove si vuole che compaiano le unità e chiamarlo "tele1"

In tale modo andando vicino al palo 1 si verrà portati all'elipad del palo 2 e viceversa.

Si possono inserire più destinazioni mettendo altri elipad e quindi più addaction all'interno della stringa ossia

```
this addaction ["Name of Location Here es Tele2", "tele2.sqf"]; this addaction ["Name of Location Here  
es Tele3", "tele3.sqf"];
```

La procedura precedente si può realizzare anche senza l'uso di file sqf esterni, in tal caso:

1. Inserire un palo con bandiera (1) ed inserire la stringa nel suo init

```
this addaction ["Name of Location 1 Here", {player setPos (getPos nomehelipad2)}];
```

2. Inserire un helipad invisibile vicino al palo 1 dove si vuole che compaiano le unità e chiamarlo "nomehelipad1"
3. Inserire un altro palo con bandiera (2) in una nuova posizione ed inserire la stringa nel suo init

```
this addaction ["Name of Location 2 Here", {player setPos (getPos nomehelipad1)}];
```

4. Inserire un altro helipad invisibile vicino al palo 2 dove si vuole che compaiano le unità e chiamarlo "nomehelipad2"

52 TENERE FERMA UNA IA

Per tenere ferma una IA inserire nel suo init:

```
this enablesimulation false;
```

funziona anche su server.

53 SPAWN AND RESPAWN INTO A PARACHUTE

Here's a method to spawn, and respawn, into a parachute. There's a demo mission available.

```
// Function that creates a parachute over the dropzone marker and moves the player into it.  
fnc_letsFly = {  
    _para = createVehicle ["Parachute_US_EP1", getMarkerPos "mrk_dropZone", [], 0, "FLY"];  
    _para setPosATL [getPosATL _para select 0, getPosATL _para select 1, 200];  
    player moveInDriver _para;  
};  
  
// This starts the player in the air at gamestart  
call fnc_letsFly;  
  
// This adds the MP event to the player  
player addMPEventHandler ["MPRespawn", {_this spawn fnc_letsFly;}];
```



54 MASS AI PARADROP

54.1 PRIMO METODO

1. Aggiungere delle unità e assegnargli un nome ad ognuna, es "c1", "c2", "c3" ... "c6"
Nota: assicurarsi che non siano tutti nello stesso gruppo
2. Aggiungere un mezzo di trasporto (elicottero o C130 che sia) e assegnargli un nome, es "heli1". Eliminare il co-pilota



3. Mettere le unità create a bordo dell'elicottero
4. Muovere l'eli da qualche parte in cielo e assegnargli una rotta con un waypoint dove si vuole paracadutare le unità.



5. Mettere un trigger proprio su quest'ultimo waypoint, grande ed alto abbastanza per "prendere" l'heli. Sincronizzarlo con l'heli con "assegna proprietario" (tasto destro sul trigger/connetti/assegna propr)
6. Nel trigger impostare
Tipo: presenza
On activation:

```
c1 action ["EJECT", heli1];  
c2 action ["EJECT", heli1];  
c3 action ["EJECT", heli1];  
c4 action ["EJECT", heli1];  
c5 action ["EJECT", heli1];  
c6 action ["EJECT", heli1];
```

"EJECT" o l'analogo "GET OUT"

La stringa in generale è del tipo

```
unitName action ["EJECT", vehicleName];
```

Oppure

```
unitName action ["GET OUT", vehicleName];
```

L'effetto è che all'attivazione del trigger le unità verranno paracadutate.

7. A lancio avvenuto si provvederà a cancellare eventualmente l'heli con un altro trigger su un punto lontano che avrà all'attivazione:

```
deleteVehicle HeliName;  
deleteVehicle PilotName;
```

Nota: ricordarsi di assegnare un nome anche a pilota in modo che venga cancellato assieme al velivolo

Alternative (da testare)

Invece di usare un trigger che attivi il lancio si potrebbe inserire le stringhe `unitName action ["EJECT", vehicleName];` direttamente nell'init del waypoint del punto di lancio



54.2 SECONDO METODO

“Mass paradrop” di AI da un elicottero o C-130 per esempio.

Questo codice può essere utilizzato per espellere qualsiasi gruppo intero di unità dal veicolo in cui si trovano con una semplice chiamata di script. Questo codice può anche essere utilizzato per espellere qualsiasi gruppo dal veicolo in cui si trovano, basta inserire il nome del gruppo.

1. Creare il file "groupEject.sqf" nella cartella della missione

Contenuto del file "groupEject.sqf"

```
// Text of the "groupEject.sqf" file
if (isServer) then
{
    _grp = _this select 0;

    sleep (random 3);

    {
        unassignVehicle (_x);
        (_x) action ["EJECT", vehicle _x];
        sleep 0.4;
    } foreach units _grp;
};
```

La funzione **unassignVehicle** annulla l'assegnazione di una singola unità da un veicolo, ovvero rimuove l'assignedVehicleRole dell'unità. Se l'unità è attualmente in quel veicolo, il capogruppo darà l'ordine di evacuazione.

Esempio 1:

```
unassignVehicle player;
```

Esempio 2:

```
{ unassignVehicle _x } forEach crew _vehiclename;
```

2. Assegnare un nome al gruppo dell'unità "groupName"
3. Impostare un trigger ed all'attivazione (oppure nell'init del waypoint) mettere

```
nul = [groupName] execVM "groupEject.sqf";
```

Alternativa

Sulla falsa riga di questo, senza creare un sqf esterno, si può operare anche così:

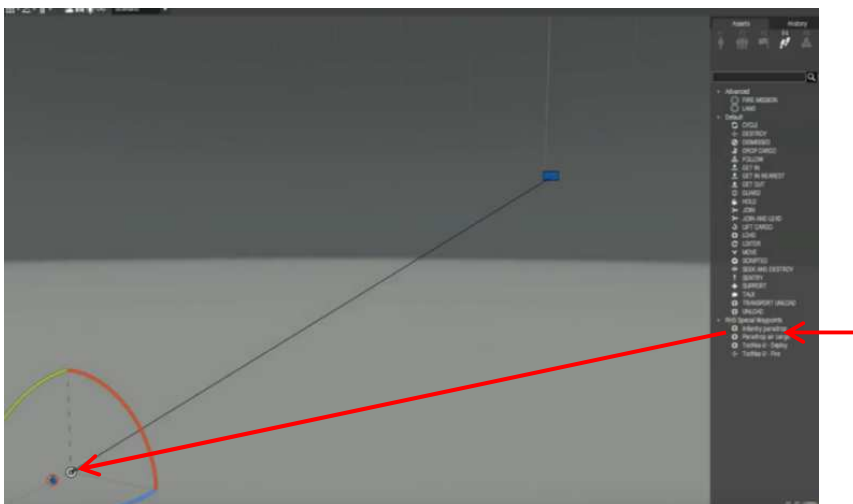
1. Assegnare un nome al gruppo dell'unità "groupName"
2. Impostare la rotta
3. Nel campo init del waypoint di lancio mettere

```
{unassignVehicle _x; _x action ["Eject", vehicle _x]} forEach units group groupName
```

54.3 TERZO METODO

Un metodo più immediato è l'uso dei moduli della mod dell'RHS, che fornisce un waypoint dedicato per il paradrop delle unità.

1. Piazzare le unità nel velivolo
2. Selezionare la rotta
3. Impostare il waypoint di lancio selezionandolo tra quelli RHS come "Paradrop air cargo"
4. Le unità a bordo verranno di conseguenza paracadutate nel waypoint selezionato



55 SUPPLY DROP

55.1 PRIMO METODO

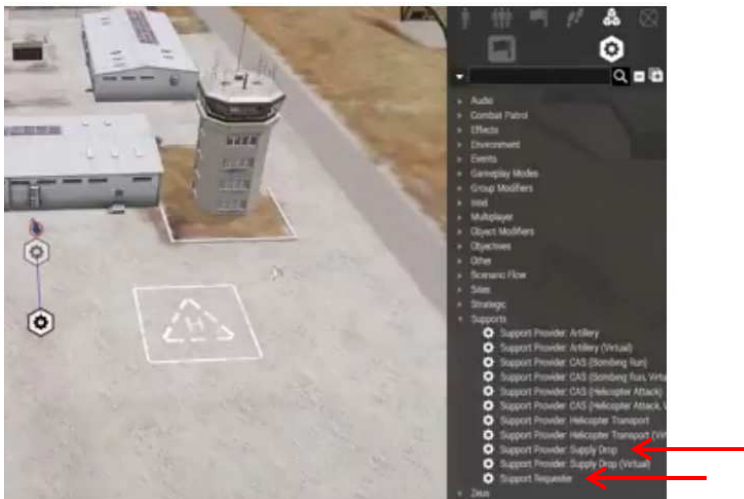
<https://www.youtube.com/watch?v=8teRi2srqQA>

<https://steamcommunity.com/sharedfiles/filedetails/?id=866990159>

1. Mettere un'unità e darle un nome
2. Mettere giù il modulo Supports > Support Requester e sincronizzarlo con l'unità

Se si usa un elicottero fisico:

- a. Mettere giù il modulo Supports > Supply Drop e sincronizzarlo con il Support Requestor.



- b. Mettere un elicottero e sincronizzarlo con il Supply Drop



Qui se ne sono messi addirittura 2 di elicotteri

Se si usa un elicottero virtuale:



- a. Mettere giù il modulo Supports > Supply Drop Virtual e sincronizzarlo con il Support Requestor

Il supply drop sarà attivabile dal player con il menu “\”

Nota: da testare se valido in MP.

Leggere anche di seguito:

Requires:

1. Player / Playable Unit(s) {<Various>}
2. Support Requester {SupportRequester}
 - a. Support Provider: Supply Drop {SupportProvider_Drop}
 - i. Physical Transport Helicopter placed on map
 - or
 - b. Support Provider: Supply Drop (Virtual) {SupportProvider_Virtual_Drop}

Methodology

1. Place Player / Playable Unit(s) that you want to be able to call supports
2. Place Systems > Modules > Supports > Support Requester {SupportRequester} module
 - Set any support limits you require
 - Sync Player / Playable Unit(s) to Support Requester module
3. If you are using physical helicopters:
 - a. Place Systems > Modules > Supports > Support Provider: Supply Drop module
 - b. Sync the Support Provider: Supply Drop module to the Support Requester module
 - c. Place Physical Transport Helicopter(s)
 - d. Sync Physical Transport Helicopter(s) to the Support Provider: Supply Drop module
4. If you are using virtual helicopters:
 - a. Place Systems > Modules > Supports > Support Provider: Supply Drop (Virtual) module
 - b. Sync the Support Provider: Supply Drop (Virtual) module to the Support Requester module

Variables:

System Specific - Support Provider: Supply Drop

1. Crate Init: Code executed upon crate spawn (Affected crate: `_this`)

System Specific - Support Provider: Supply Drop (Virtual)

1. System: Transformation > Position sets where virtual aircraft will spawn
2. Vehicle Types: Types of vehicles available (Empty array = defaults loaded)
 - UH-80 Ghost Hawk {B_Heli_Transport_01_F}
 - CH-67 Huron {B_Heli_Transport_03_F}
 - CH-67 Huron (Unarmed) {B_Heli_Transport_03_unarmed_F}
 - UH-80 Ghost Hawk (Sand) {B_Heli_Transport_01_sand_F}
 - UH-80 Ghost Hawk (Tropic) {B_Heli_Transport_01_tropic_F}
3. Vehicle Init: Code executed upon vehicle spawn (Affected vehicle: `_this`)
4. Crate Init: Code executed upon crate spawn (Affected crate: `_this`)
5. Filtered By: Default pool of vehicles to use
 - Side
 - Faction
6. Cooldown: Support availability delay in seconds

55.2 SECONDO METODO

<http://www.armaholic.com/page.php?id=31915>

Description:

This script/FSM gives the functionality for resupply drop in MP/SP scenario.

Created because the bis support resupply drop is not working in MP currently.

Installation / Usage:

For usage instructions and information of how to use the Supply drop script please refer to the included example mission.



This is similar to support virtual resupply drop module, gives the possibility to select aircraft types, box types and its init by editing the support\supplydrop.sqf

```
//***** editable *****//
private _heliclass = selectRandom [

    "B_Heli_Transport_03_F",
    "B_Heli_Transport_01_F",
    "B_Heli_Transport_01_camo_F",
    "B_Heli_Transport_03_unarmed_F",
    "B_T_VTOL_01_vehicle_F"

];
private _boxclass = selectRandom [

    "B_supplyCrate_F",
    "Box_NATO_AmmoVeh_F",
    "B_CargoNet_01_ammo_F"

];

private _chuteType = "B_Parachute_02_F";
private _boxCode = compile "null = [_this] spawn fnc_dropboxinit;";
private _helistart = "helistartpos"; // marker where the heli spawns or position array [posX,posY,posZ]

//***** editable *****//
```

1. Edit the above to fit your needs.
2. For helicopter spawn position create a marker somewhere on the map in editor or use position array.
3. Set it for _helistart variable value.
4. Copy paste all files (besides mission.sqm) from example mission to your mission or if your mission contains those files merge the files content.
5. Check init.sqf for this part.

```
//if you want to limit support to certain units add its classnames to the array below
//last entry without comma at the end
//uncomment "All" if you want it for all players
private _givesupport = [

    //"All",
    "B_Soldier_SL_F",
    "B_Soldier_TL_F"

];
```

6. Edit/add your desired class names which players will have the drop menu available or Un-comment "All" for all players.
7. Edit support\dropbox.sqf as you want. Currently only arsenal,smoke,and IR added.

```
// init supplydrop box
// by DaVdoSS

params ["_oldbox"];
private ["_supplyLight", "_supplySmoke"];

waitUntil {sleep 1;(getPos _oldbox select 2) < 2};

clearWeaponCargoGlobal _oldbox;
clearMagazineCargoGlobal _oldbox;
clearItemCargoGlobal _oldbox;
clearBackpackCargoGlobal _oldbox;

_supplyLight = "NVG_TargetW" createVehicle (position _oldbox);
_supplyLight attachTo [_oldbox, [0,0,0.75]];

_supplySmoke = "SmokeShellGreen" createVehicle (position _oldbox);
_supplySmoke attachTo [_oldbox, [0,0,0.6]];

null = ["AmmoboxInit",[_oldbox,true,({_target distance _this} < 5)]] remoteExec
["BIS_fnc_arsenal", [0,-2] select isDedicated,_oldbox];
//put code below for any box inventory.
```



```
//_oldbox is your box
```

- Put an Headquarters entity module (found in systems > Intel) with:
VARIABLE NAME - hq_suppcomm
SIDE - select player side
CALLISGN - NATO
CUSTOM CALLISGN - your custom callisgn (will be displayed with radio sentences)
SPEAKER - not important

That guy will handle the radio communications for you.

Addon

- Thanks for this. Got it working for my missions, but only locally. In MP on dedicated server the supply drop icon does not even appear (works for same mission locally). Anything i miss?

The radio comm's are added locally by this part:

```
if (hasInterface) then {  
  
    fnc_supplydrop = compileFinal preProcessFileLineNumbers "support\supplydrop.sqf";  
  
    //if you want to limit support to certain units add its classnames to the array below  
    //last entry without comma at the end  
    //uncomment "All" if you want it for all players  
    private _givesupport = [  
  
        //"All",  
        "B_Soldier_SL_F",  
        "B_Soldier_TL_F"  
  
    ];  
  
    if ((typeof player) in _givesupport || {"All" in _givesupport}) then {  
  
        null = [player, "supplydrop"] call BIS_fnc_addCommMenuItem;//add radio link  
  
    };  
  
};
```

Try move this part to the initPlayerLocal.sqf, maybe your init.sqf are busy in some loop or wait's above this code

55.3 TERZO METODO

This is what I use.

```
/*  
    Author: HallyG  
    Spawns a supply drop of desired object.  
  
    Arguments(s):  
    0: Supply Drop Object (classname) to spawn - <STRING>  
    1: Supply Drop Centre - <MARKER, OBJECT, LOCATION, GROUP, TASK, POSITION>  
    2: Supply Drop Height - <SCALAR>  
    3: Supply Drop attachTo Position - <ARRAY>  
  
    Return Value:  
    <OBJECT>  
  
    Example:  
    [  
        "CargoNet_01_box_F",  
        position player,  
        100,  
        [0,0,-1.2],  
        WEST  
    ] call FUNCTIONNAME  
*/  
params [  
    ["_object", "CargoNet_01_box_F", [""]],  
    ["_centre", [0, 0, 0], ["", objNull, taskNull, locationNull, [], grpNull], [3]],
```



```
        ["_height", 100, [0]],
        ["_attachTo", [0, 0, -1.2], [[]], [3]]
];

_centre = _centre call {
    if (_this isEqualType objNull) exitWith {getPosASL _this};
    if (_this isEqualType grpNull) exitWith {getPosASL (leader _this)};
    if (_this isEqualType "") exitWith {getMarkerPos _this};
    if (_this isEqualType locationNull) exitWith {locationPosition _this};
    if (_this isEqualType taskNull) exitWith {taskDestination _this};
    if (_this isEqualType []) exitWith {_this};
    [0, 0, 0]
};

if (!(isClass (configfile >> "cfgVehicles" >> _object)) || _centre isEqualTo [0, 0, 0]) exitWith {
    objNull
};

private _obj = createVehicle [_object, _centre vectorAdd [0, 0, _height], [], 0, "NONE"];
private _para = createVehicle ["B_parachute_02_F", [0,0,0], [], 0, "FLY"];

_obj setDir getDir _obj;
_obj setPos getPos _obj;
_obj lock false;
_obj attachTo [_para, _attachTo];

[_obj, _para] spawn {
    params ["_obj", "_para"];

    waitUntil {
        sleep 0.01;
        ((position _obj) select 2) < 2
        ||
        isNull _para
        ||
        (count (lineIntersects [getPosASL _obj, (getPosASL _obj) vectorAdd [0, 0, -0.5],
_obj, _para])) > 0
    };

    _para disableCollisionWith _obj;
    _obj setVectorUp [0,0,1];
    _obj setVelocity [0,0,0];
    detach _obj;

    if (!isNull _para) then {deleteVehicle _para};

    (format ["A supply drop has touched down, grid %1.", mapGridPosition getPosATL _obj])
remoteExec ["systemChat", 0, false];
};

_obj
```

56 SPAWNING UNITS

Short version: In the "On Act" field of a trigger named trigger_1, put

```
if (isServer) then {_myGroup = createGroup side; _myUnit = _myGroup createUnit ["Soldier_classname", getPos
trigger_1,[], 0, "NONE"];};
```

Long version:

There are a couple of commands you need to familiarize yourself with: [createUnit](#) and [createUnit Array](#) for [Persons](#), and [createVehicle](#) and [createVehicle Array](#) for empty vehicles and objects. If what you're spawning has AI, you want to use a createUnit command.

The array-variants are much more efficient in terms of performance (up to 500 times faster, some say), so we'll stick to using those.

Spawn a single unit:

```
_myUnit = _myGroup createUnit ["Soldier_classname", position, [markers], placement radius, "SPECIAL"];
```

- *_myGroup* is the group the spawned unit will be part of. More on this in the example below.
- *_myUnit* is the name you've given to your unit.



- "*Soldier_classname*" is the type of soldier to spawn, e.g. "*B_Soldier_F*", "*I_Soldier_lite_F*" etc. Remember to use "" when dealing with [strings](#).
- [position](#) is an array of coordinates, [x,y,z], or a function that returns a position: `getMarkerPos "myMarker"`; `getPos myObject`; and so on .
- *[markers]* is an array of markers. If this is left empty, the unit will be created at *position*. If you put several markers here `["myMarker1","myMarker2"]`, *myUnit* will spawn at a position chosen at random, including the *position* argument.
- *placement radius* is a number. Spawned units will be placed within a circle with *position* as center and *placement radius* as radius. It's a fast and easy way to spawn a unit at a random location, at the cost of the fine control you get by using *[markers]* with a low *placement radius*.
- "*SPECIAL*". Valid choices for `createUnit` are: "NONE", "FORM" and "CAN_COLLIDE". "FORM" spawns the unit in formation. "CAN_COLLIDE" is used to place the unit exactly on the *position*. This can lead to units spawning in walls and objects, so be careful. It's very useful for placing furniture with the `createVehicle` command.

Example: spawn a single BluFor rifleman

```
//creating a group, as spawned units MUST be member of a group.  
_myGroup = createGroup west;  
_myUnit = _myGroup createUnit ["B_Soldier_F", [14000,17000,0],["myMarker1","myMarker2"], 50, "NONE"];
```

This spawns a BluFor Rifleman belonging to your newly created group. *west* tells us which [side](#) the unit belongs to. The unit will spawn within 50 meters of coordinates [14000,17000,0] (just north of the large runway on Altis) or *myMarker1* or *myMarker2*, selected at random. It will spawn at a safe location (meaning, not inside a wall).

Spawn a group:

```
//syntax: [arguments] call BIS_fnc_spawnGroup;  
_myGroup = [position, side, character details, relative positions, list of ranks, skill range, ammo count range,  
randomization controls, azimuth] call BIS_fnc_spawnGroup;
```

```
//some arguments are optional, this is the bare minimum:  
_myGroup = [position, side, character details] call BIS_fnc_spawnGroup;
```

- **Required arguments**
- *position* see above
- [side](#). A group placed by `BIS_fnc_spawnGroup` cannot be the first group spawned belonging to a *side*. Placing a unit of the same side in the editor with probability of presence = 0 takes care of that problem, or you can use [createCenter](#).
- *character details* can be one of three types: *number*, *array* or *config entry*. *Number* is straightforward, the function will spawn this many units from the correct *side*, classes are chosen at random. I've never got this to work with number, for some reason. *Array* is a list of classnames, e.g. ["B_Soldier_F", "B_engineer_F"]. This will spawn two soldiers, one rifleman and one engineer. *Config entry* will spawn a group like the ones you get when you press F2 in the editor and place a whole group. E.g. (`configFile >> "CfgGroups" >> "West" >> "BLU_F" >> "Infantry" >> "BUS_InfSquad"`). Use the config viewer in the editor, open `CfgGroups` and you'll find all the valid groups. Note that you must use the classname (BUS_InfSquad), not the name ("Rifle Squad"). You can copy the string from the bottom of the configviewer. This also has less impact on performance compared to spawning an identical group "by hand", since it's predefined.
- **Optional arguments**
- *relative positions* is how far from *position* the individual units will spawn, or offset if you like. E.g. `[[0,0,0],[0,30,0]]` will spawn the first unit on *position* and the second unit 30 meters north of *position*. Be careful with this, if the number of offsets doesn't match the exact number of units spawned the whole function will fail.
- *list of ranks* is an array containing the unit ranks. E.g. ["Captain", "Private"]. Again, the number of ranks needs to match the exact number of units spawned. The squad will automatically be organized with the highest ranking unit as number 1, second highest rank as number 2, etc.
- *skill range* is an array with format [min,max] or [max,min], both work. It assigns a random skill-level within this range to all spawned units. Default value is 0.5, valid values are 0 to 1.
- *ammo count range* is an array with format [min, max]. Valid values are 0 to 1. This sets the ammo for the units to min + a random value, where the result can not be above max. The actual number of magazines and ammo is dependent on the units class.



- *randomization controls* is an array with format [minimum number of units to spawn, propability of the rest of the units to spawn]. Valid values are [less than max number of units to spawn, 0 to 1]
- *azimuth* is the direction the spawned group is facing, 0 to 359.

Examples:**Spawn a group with 3 riflemen from OpFor at myMarker:**

```
_myGroup = [getMarkerPos "myMarker", east, ["O_Soldier_F","O_Soldier_F","O_Soldier_F"]] call BIS_fnc_spawnGroup;
```

Spawn a group with a BluFor rifleman (captain) at myMarker, and an engineer (private) 100 meters east of the rifleman:

```
_myGroup = [getMarkerPos "myMarker", west, ["B_Soldier_F","B_engineer_F"], [[0,0,0],[100,0,0]],["Captain", "Private"]] call BIS_fnc_spawnGroup;
```

Spawn a predefined group, an Independent Weapons Squad. No offset, default ranks, skill between 0.3 and 0.6, ammo count between 20 and 50% of full loadout, minimum 2 units spawned with 80% chance of the rest spawning, facing southeast (135 degrees):

```
_myGroup = [getMarkerPos "myMarker", resistance,(configfile >> "CfgGroups" >> "Indep" >> "IND_F" >> "Infantry" >> "HAF_InfSquad_Weapons"),[],[],[0.3,0.6],[0.2,0.5],[2,0.8],135] call BIS_fnc_spawnGroup;
```

Implementation

Getting back to how to spawn units based on a condition (can be a trigger that's fired, a variable that's been set or any other possible condition):

If using triggers, put your code in the onAct field of the trigger, and wrap it in a server-check.

```
if (isServer) then { _myGroup = createGroup side; _myUnit = _myGroup createUnit ["Soldier_classname", getPos trigger_1,[], 0, "NONE"];};
```

If you want to call a script version:

Call:

```
if (isServer) then { myScriptHandle = [] execVM "myScript.sqf"; };
```

myScript.sqf:

```
waitUntil {triggerActivated trigger_1};  
_myGroup = createGroup side;  
_myUnit = _myGroup createUnit ["Soldier_classname", getPos trigger_1,[], 0, "NONE"];};
```

The waitUntil line is the key here. You can check for lots of different conditions here, and even put in a sleep to lessen impact on performance. Here's an example that waits until a unit is dead, and it checks once every second:

```
waitUntil {sleep 1;!alive myUnit};  
//code to do after condition is met goes here
```

'!alive myUnit' is the statement being evaluated, and waitUntil will loop as long as myUnit is alive, sleeping for 1 second between every check.

References:

[createUnit Array](#)

[createVehicle Array](#)

[position](#)

[side](#)

[waitUntil](#)

[BIS_fnc_spawnGroup](#)

57 COPY POS TO CLIPBOARD

```
copyToClipboard str (getPos player)
```

58 INIZIARE LA PARTITA A BORDO DI UN ELICOTTERO

Per iniziare la partita a bordo di un mezzo, per esempio un elicottero che si sta dirigendo verso la zona di sbarco, operare in questo modo:

1. Mettere giù l'elicottero con equipaggio, nominarlo per esempio "heli1" e posizionarlo alla quota di volo



2. Impostarne la rotta con i waypoints e all'atterraggio mettere il waypoint "scaricare veicolo"
3. Mettere giù le unità giocabili in un punto qualsiasi in mappa e ad ognuna mettere nell'init la stringa

```
this moveincargo heli1;
```

questa le mette nell'elicottero una volta avviata la partita
oppure mettere la stringa nell'init del caposquadra

```
{_x moveincargo heli1} foreach units group this;
```

se non sono raggruppate anche con la seguente, nominando le unità per esempio con e1,e2,e3...

```
if their not grouped they should be named fe e1 e2 e3
```

```
{_x moveincargo NAMEOFCHOPPER} foreach [e1,e2,e3.....];
```

4. (opzionale) Impostare la rotta di ritorno dell'elicottero e nel suo waypoint finale un trigger con blufor presente ed all'attivazione la stringa che lo elimina

```
deletevehicle heli1;
```

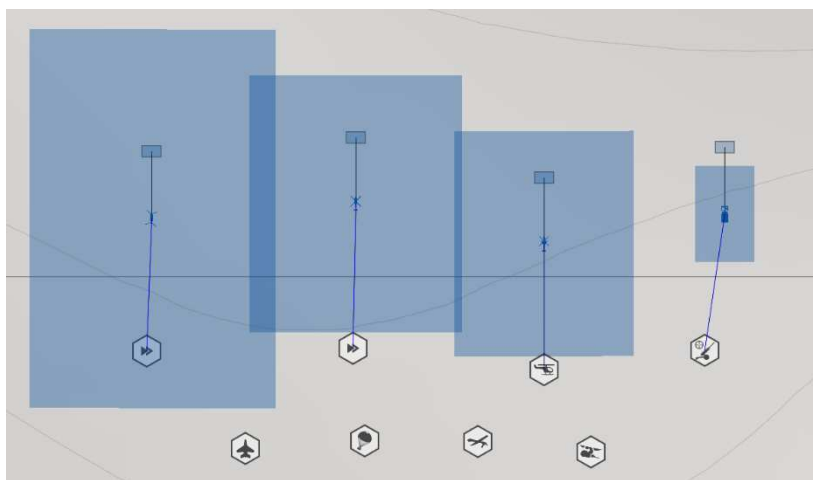
59 SIMPLEX SUPPORT SERVICE MOD – MODULI DA POSIZIONARE

Per usare la mod "Simplex support service" è conveniente preparare in editor I moduli che serviranno, attivabili dal menu ace. La figura seguente mostra come impostare i vari moduli in editor.

Alcuni moduli vanno sincronizzati con dei veicoli specifici (con equipaggio) preventivamente posizionati (trasporto heli, CAS heli, artiglieria).

Altri moduli si mettono giù e basta senza sincronizzarli con un veicolo in particolare.

I moduli si possono mettere giù in partita anche da zeus. Per quelli che necessitano un veicolo occorre metterli sopra al veicolo voluto, in tal caso da metter giù con zeus.



60 ADDACTION

60.1 ADDACTION COMMAND

Adds an entry to the action menu of an object (scroll wheel menu). The action can only be activated when in proximity to the object (default 50m). Adding an action to the player obviously makes that action available to the player at all times. The appearance of onscreen text could be further tweaked with setUserActionText. For event handling of user interaction see inGameUISetEventHandler.

(i) In Arma 3 addAction does not work on animals, this is intended behavior.

Syntax:



object **addAction** [title, script, arguments, priority, showWindow, hideOnUse, shortcut, condition, radius, unconscious, selection, memoryPoint]

Parameters:

object:

[Object](#) - unit, vehicle or static object

title:

[String](#) - The action name which is displayed in the action menu, may contain [XML like syntax](#). Because of that < and > symbols will be interpreted as opening and closing XML tags. To avoid this use **<** for < and **>** for >. The title text can be changed with [setUserActionText](#)

script:

[String](#) or [Code](#) - Either path to the script file, relative to the mission folder or string with code or the actual script code. If the string is a path to script file, the script file *must* have extension [.SQS](#) or [.SQF](#). The script, whether it is a file or a code, will run in [scheduled environment](#), i.e. it is ok to use [sleep](#).

- Parameters array passed to the script upon activation in [_this](#) variable is:

```
params ["_target", "_caller", "_actionId", "_arguments"];
```

- target* (_this select 0): [Object](#) - the object which the action is assigned to
- caller* (_this select 1): [Object](#) - the unit that activated the action
- ID* (_this select 2): [Number](#) - ID of the activated action (same as ID returned by addAction)
- arguments* (_this select 3): [Anything](#) - arguments given to the script if you are using the extended syntax

arguments:

[Anything](#) - (Optional, default [nil](#)) arguments to pass to the script (will be *_this select 3* inside the script). If [Array](#) is used as an argument for example, its 1st element reference would be *_this select 3 select 0*

priority:

[Number](#) - (Optional, default 1.5) priority value of the action. Actions will be arranged in descending order according to this value. Every game action has a preset priority value. Value can be negative or decimal fraction. Actions with same values will be arranged in order which they were made, newest at the bottom. The bigger the number the higher the action will be positioned on the menu. Typical range is 0 (low priority. Eg: 'Get out') to 6 (high priority. Eg: 'Auto-hover on')

showWindow:

[Boolean](#) - (Optional, default [true](#)) if set to [true](#), players see "Titletext" at mid-lower screen, as they approach the object. Only the title text for the action with highest priority and 'showWindow true' will be shown. Setting it to [false](#) disables the feature

hideOnUse:

[Boolean](#) - (Optional, default [true](#)) if set to [true](#), it will hide the action menu after selecting that action. If set to [false](#), it will leave the action menu open and visible after selecting that action, leaving the same action highlighted, for the purpose of allowing you to reselect that same action quickly, or to select another action

shortcut:

[String](#) - (Optional, default "") one of the [key names](#) defined in bin.pbo (e.g. "moveForward"). Adding available shortcut will bind corresponding keyboard key to this action. Shortcut availability can be tested with [inputAction](#) command

condition:

[String](#) - (Optional, default "true") script code that must return [true](#) for the action to be shown. Special variables passed to the script code are:

- _target*: [Object](#) - the object to which action is attached or, if the object is a unit inside of vehicle, the vehicle
- _this*: [Object](#) - caller person to whom the action is shown (or not shown if condition returns [false](#))
- _originalTarget*: [Object](#) - the original object to which the action is attached, regardless if the object/unit is in a vehicle or not
- condition* is evaluated on each frame in [non-scheduled environment](#).



- *condition* is **not** evaluated if a dialog is open.
- If action is added to an object (and not to [player](#)) *condition* will only get evaluated IF player is closer than ~50m to the object surface AND is looking at the object.
- If action is added to [player](#), *condition* is evaluated all the time.

radius:

[Number](#) - (Optional, default 50) 3D [distance](#) in meters the unit activating the action must be within to activate it. -1 disables this radius

unconscious:

[Boolean](#) - (Optional, default [false](#)) if [true](#) will be shown to incapacitated player (see [setUnconscious](#), [lifeState](#))

selection:

[String](#) - (Optional, default "") named selection in Geometry LOD to which the action is attached

memoryPoint:

[String](#) - (Optional, default "") memory point on the object to which the action is attached. If *selection* is supplied, *memoryPoint* is not used

Return Value:

[Number](#) - The ID of the action is returned. Action can be removed with [removeAction](#) (see also [removeAllActions](#)). IDs are incrementing, the first given action to each unit has the ID 0, the second the ID 1, etc. IDs are also passed to the called script (see *script* parameter)

Example 1:

```
// short and sweet
player addAction ["a useless action that does nothing", {}];
player addAction ["<t color='#FF0000'>This Useless Action Is RED</t>", {hint "RED"}];
player addAction ["Hint Hello!", { hint format ["Hello %1!", name player] }];
player addAction ["String Exec", "hint 'this is also compiled'"];
```

Example 2:

```
_actionID = player addAction ["Exec the file", "scriptFile.sqf"]
```

scriptFile.sqf:

```
hint str _this;
```

Example 3:

```
// create object on the server and add action to the object on every client
if (isServer) then
{
    _object = "some_obj_class" createVehicle [1234, 1234, 0];
    [_object, ["Greetings!", {hint "Hello!"}]] remoteExec ["addAction"];
};
```

Example 4:

```
Default parameters:
this addAction
[
    "<title>",
    {
        params ["_target", "_caller", "_actionId", "_arguments"];
    },
    [],
    1.5,
    true,
    true,
    "",
    "true", // _target, _this, _originalTarget
    50,
    false,
    "",
    ""
];
```

60.2 REMOVE ADDACTION

Here's how to remove a single addAction from an object after it's called:



```
_target = _this select 0; // Object that had the Action (also _target in the addAction command)
_caller = _this select 1; // Unit that used the Action (also _this in the addAction command)
_action = _this select 2; // ID of the Action

// Remove the rescue option
_target removeAction _action;
```

Here's how to remove all addAction from an object:

```
_tmpaction = _object addAction["foo", "foo.sqf"];
while {_tmpaction >= 0} do {
    _target removeAction _tmpaction;
    _tmpaction = _tmpaction - 1;
};
```



61 OTHER SCRIPTS

61.1 COMMON SCRIPTS

61.1.1 TELEPORT TO EMPTY MARKER

```
setPos (getMarkerPos "markername")
```

Or , use set position module. Vedi anche §51.

61.1.2 TRIGGER TO CREATE A MARKER - ALSO CREATES A HINT

```
hint "An S.O.S Signal Was Detected"; _marker1 = createMarker ["markerSOS", [2188, 5362]]; "markerSOS"  
setMarkerShape "Ellipse"; "markerSOS" setMarkerSize [50, 50]; "markerSOS" setMarkerBrush "DIAGGRID";  
"markerSOS" setMarkerColor "ColorYellow";
```

61.1.3 ONLY FIRE A TRIGGER WHEN MULTIPLE TRIGGERS HAVE FIRED

```
triggerActivated TR_KILL_Parent_SU AND triggerActivated TR_Destroy_Supplies_Parent_SU;
```

61.1.4 FIRE A TRIGGER WHEN EITHER TRIGGERS HAVE FIRED

```
triggerActivated TR_KILL_Parent_SU OR triggerActivated TR_Destroy_Supplies_Parent_SU;
```

61.1.5 CREATE A VIRTUAL ARSENAL OBJECT

- In the Init field of the object place

```
["AmmoboxInit",[this,true]] call BIS_fnc_arsenal;
```

In the Role Description, place : Arsenal

Guarda anche §0.

61.1.6 ADD BI AND ACE ARSENAL TO THE SAME BOX

```
["AmmoboxInit",[this,true]] call BIS_fnc_arsenal; [this, true] call ace_arsenal_fnc_initBox;
```

61.1.7 GROUP PATROL

Have a group patrol a vicinity randomly (ie 20 meters)

edit the Init of the group leader =

```
[group this, getPos this, 20] call bis_fnc_taskPatrol
```

Have a group man nearby static defenses within a 100 metre radius of the defense position and guard the position. Some units will man weapons, others will patrol and the remainder will sit on the ground.

edit the Init of the group leader =

```
[group this, getPos this] call BIS_fnc_taskDefend;
```

61.1.8 DELETE AN OBJECT

IE: Deletes the Helo and the Pilot - Units have to have variable name

```
deleteVehicle Insertion_Heli; deleteVehicle Insertion_Heli_CP;
```

61.1.9 SET FLYING HEIGHT FOR CHOPPER

```
Insertion_Heli flyInHeight 40;
```



61.2 MISSION STORY TELLING

61.2.1 AI CHAT

Note the AI object must exist

```
Insertion_Heli_P sideChat "Chat Message"
```

61.2.2 SET A HINT – INSERIRE UN TESTO A VIDEO

La funzione hint introduce un suggerimento che compare a video. Digita questo nel campo "All'attivazione":

```
hint "Testo del suggerimento che comparirà in alto a destra"
```

Utilizzare **hintC** per richiedere all'utente di fare clic su "Continua" prima di procedere.

Utilizzare **hintSilent** per disabilitare il segnale acustico.

Per visualizzare nuovamente l'ultimo suggerimento nel gioco, premi H.

61.3 CONDIZIONI

61.3.1 USE SPEED AS A CONDITION

To use the speed of a unit as a condition, use this code in the 'Condition' field of a trigger:

```
speed car1 > 55;
```

You could add a hint on activation to read "You are speeding!"

61.3.2 HOLD AT WAYPOINT UNTIL TRIGGER IS ACTIVATED

To make a unit or group hold at a waypoint, click 'Synchronize' and drag from the unit's waypoint to the trigger.

61.3.3 THIS GROUP ACTIVATES TRIGGER

To make a specific group be the one to activate a trigger, click 'Groups' and drag a line from the activating group to the trigger. Go back to the trigger menu and change to 'Entire Group', 'Group Member', or 'Group Leader' to declare who from the group activates it.

61.3.4 HOW MUCH DOES UNIT KNOW ABOUT THIS?

Every unit knows between 0% and 100% about every other unit. To utilize this, you can check how much a unit knows about another. This is useful for creating intruder alarms and such. Used as a 'Condition', this code will determine if Nick knows at least 50% of the information about Bob.

```
Nick knowsabout Bob >2; // use number between 0 and 4 for knowledge level
```

61.3.5 IS UNIT DEAD?

Per verificare se un'unità o un'unità è morta, inserisci questo codice nel campo "Condizione":

```
!alive Bob;
```

Il "!" praticamente dice NO, quindi NOTalive Bob. Questo può essere omesso per chiedere se Bob è vivo.

61.3.6 CHECK IF ENGINE IS ON

Used as a condition, this code is typed into waypoints or triggers:

```
isEngineOn car1;
```

To check if engine is off, type

```
!isEngineOn car1
```

61.3.7 IS UNIT PRESENT INSIDE TRIGGER RADIUS?

To see if a specific unit is inside a trigger, type this code into the triggers 'Condition' field:

```
Bob in thislist;
```




For multiple units use:

```
(Bob in thislist) && (Nick in thislist);
```

or use

```
(Bob in thislist) or (Nick in thislist);
```

61.3.8 IS UNIT A HUMAN PLAYER?

To see if a unit is a human player type this as a 'Condition':

```
isplayer Bob;
```

61.3.9 WHAT IS THE DAMAGE LEVEL OF A UNIT?

To check the damage level of a unit, type this into any 'Condition' field:

```
getdammage Bob >0.7;
```

This will trip the trigger if Bob's damage is greater than 70%. It has a value from 0 to 1.

61.3.10 IS UNIT IN THIS VEHICLE?

To check to see if a unit is in a specific vehicle, use this code as a 'Condition':

```
Bob in car1;
```

61.3.11 CHECK NUMBER OF UNITS IN VEHICLE

To make a trigger fire when a certain number of units enter car1, use this code as a trigger 'Condition':

```
count crew car1 == 10;
```

or use

```
count crew car1 >2;
```

61.3.12 DISTANCE BETWEEN TWO OBJECTS/UNITS

This code should be used as a condition. It will compare the distance between two objects and trip the trigger if it equals true.

```
Bob distance Nick < 10;
```

So, if Bob is within 10 meters of Nick, the trigger will trip. An = may be substituted for the < to trip when Bob's distance to Nick is equal to 10 meters.

61.3.13 CHECK THE DAMAGE LEVEL OF UNIT/OBJECT

To compare the damage level of anything, use this code in a 'Condition' field:

```
getdammage Bob >0.7;
```

In this code, 0.7 is equal to 70% damage. If Bob is over 70% injured, the trigger will trip.

61.3.14 CHECK TIME OF DAY

To use the time of day as a condition, use this code in a 'Condition' field:

```
daytime >=12;
```

12 is 12 hours past 0000 hrs or noon. 16 would be 1600hrs or 4pm.

61.4 MISSION MANIPULATION

61.4.1 DISABLE USER INPUT

To disable all user input including all keypad keys except for Ctrl, Alt, Del use this code in the 'On Activation' field:

```
disableUserInput true;
```

WARNING: You must have a secondary trigger set to automatically fire that says

```
disableUserInput false;
```



Without this, the game will not allow you to escape the program or enable user input. You will need to manually restart the program by hitting Ctrl, Alt, Del and shutting down the program thus losing any unsaved data.

61.4.2 RAIN/FOG ON COMMAND OR RANDOM

To trigger rainfall at any time, first be sure the weather forecast is at least 70% overcast by clicking on the Weather/Date/Time button in the upper right-hand corner of the screen and sliding the current weather bar down to the left a bit. Next, set a triggers 'On Activation' to

```
60 setRain 1;
```

where 60 is transition time in seconds and 1 is true.

For fog you may also use

```
60 setFog 0.6;
```

To make the weather random at the start of the mission use:

```
0 setOvercast (Random 0.8);
```

Put this into a trigger in the 'On Activation' field and change the 'Condition' from

```
this;
```

to

```
true;
```

```
// This will remove the fog from Arma completely, visually nice but will effect performance.  
0 setFog 0; forceWeatherChange; 999999 setFog 0;
```

61.4.3 ADD SOUNDS

1. Record sound on Audacity, export as .wav and convert to .ogg on FormatFactory.
2. Put in mission folder under folder 'sound' or 'music' respectively.
3. Create description wizard file on ARMA edit and check the Custom sounds box. In the lower right corner, click 'Add sound' and select the .ogg sound file. Do this for each sound.
4. Click 'Create Description File' and then hit 'OK'.
5. Now save to the desired mission folder.
6. In the text box, you can adjust the sound volume by changing the number in +0 dB.
7. To play music in-game, set the trigger 'Effect' to 'Track' and select the music file. To play sounds, type

```
playsound "gunshot";
```

or

```
Bob say ["gunshot", 0.1];
```

in the triggers 'On Activation' field. By having Bob "say" the sound, it will be local to him and if he dies the sound will stop.

61.4.4 SLOW OR SPEED UP TIME

To create a high speed camera effect or fast forward everything in single player missions, just change the acceleration time:

```
SetAccTime 0.5;
```

This should be inserted into the 'On Activation' field of a waypoint or trigger. This only works on single player missions!

61.4.5 SKIP TIME

To immediatly skip ahead or behind by a decided amount of hours use this code in any 'On Activation' field:

```
skiptime 5;
```

61.4.6 PLACE RANDOM CIVILIAN VEHICLES AROUND MAP

To have civilian vehicles spawn around the map at the start of the mission, simply place a module called 'Ambient Civilian Vehicles'.

61.4.7 RESPAWNING EMPTY VEHICLES

To make any empty vehicle respawn after it is completely destroyed, it takes a few lines to complete. First, in the vehicle's initialization field, type:



this respawnVehicle [10,0]; this respawnVehicle [10];

The 10 in each line declares how many seconds until respawn.

Next make a marker "Respawn_Vehicle_West". East is OPFOR, west is BLUFOR, civilian is CIVILIAN and guerilla is INDEPENDENTS.

Add

```
respawnVehicle = 3;  
respawnvehicledelay = 10;
```

to the mission's 'Description.ext' text file on the line after the unit respawn code. The 3 declares what type of respawn it will be with 3 being 'base' respawn, or rather marker respawn. It will spawn the vehicle at the marker after 10 seconds. You can also download Tophe's Simple Vehicle Respawn Script.

61.4.8 UNIT RESPAWNS

Vedi anche §40.

There are several types of respawn. The first is INSTANT. To make this, go to ARMA Edit on the desktop.

Click 'File', 'Wizard created files', and 'Description Wizard'. Be sure to select 'Show Compass, Show GPS, Show Watch'.

Click Instant and set the delay time.

Simply save this file to your specific mission folder. (ARMA Edit will save it as 'Description.ext'. Do not change!)

To make BASE respawns, Do the same thing but select 'BASE'. Go to your mission and add a marker named "respawn_east" or "respawn_west" at the location you wish to respawn.

East is OPFOR, west is BLUFOR, civilian is CIVILIAN and guerilla is INDEPENDENTS. NOTE: It will only respawn 'playable' units.

61.4.9 ADD ENVIRONMENTAL EFFECTS

To add dust particle effects and lighting effects to the environment add the modules Environmental Effects and Particle Effect.

61.4.10 ADDING MEDIC AND DRAG CAPABILITY

To allow units to become incapacitated and allow healing and dragging, you need three modules. Click 'Modules' and place "First Aid: Battlefield Clearance", "First Aid: Action", and "First Aid: Simulation". Synchronize these with each group leader.

61.4.11 END A MISSION AS WIN/LOSE

Per far terminare la missione a causa di una vittoria o di una perdita, puoi far scattare un trigger in entrambi i casi. Crea un trigger, imposta "Tipo" su "Fine 1" o "Perdita".

61.4.12 SEND RADIO MESSAGE

Usa questo codice nel campo 'On Activation' di un trigger o waypoint per inviare un messaggio radio dal quartier generale ai giocatori:

```
[Side, "HQ"] sideChat "Spostati nella tua posizione e attendi ulteriori ordini!"
```

"Side" deve essere cambiato in blufor, opfor, civilian or guerilla.

```
Bob sideChat "Test 1-2"; // Bob talks to his side only
```

```
Bob groupChat "Test 1-2"; // Bob talks to his group only
```

```
Bob globalChat "Test 1-2"; // Bob talks to all parties
```

```
Bob vehicleChat "Test 1-2"; // Bob talks to passengers inside vehicle
```



61.4.13 CREATE ADDACTION

To add an action to the player or a vehicle place

```
Firesmoke = Bob addAction [{"<t color="ffffff">">"+("Fire a smokeshell!") + "</t>"}, "smoke.sqs",  
["Bob"], 98, false, true, ""]; itrue, "";
```

in a 'On Activation' field or 'Initialization field. 'Firesmoke' is the name of the action, Bob is the unit that gets the action, 'ffffff' is the color code for the menu phrase, and 'Fire a smokeshell!' is the phrase that will appear as an option to select on the action menu (mouse scroller). 'Smoke.sqs' is the name of the script you wish to start when selecting this action.

61.4.14 SET DATE AND TIME OF DAY

To skip to a particular day or time use

```
setdate [1969, 8, 9, 5, 20];
```

1969 is the date, 8 is the month August, 9 is the day of the month, 5 is the hour and 20 is the minute using a 24hr clock (0000-2400hrs).

61.4.15 ADD SCORE TO PLAYERS

To add 10 points to a players score, type this into 'On Activation' field:

```
Bob addscore 10;
```

61.4.16 DISABLE/ ENABLE RADIO ALPHA, BRAVO, ETC

To disable the ability to use radio ALPHA for instance, type

```
1 setRadioMsg "null";
```

into any 'On Activation' field. The number 1 in the beginning describes ALPHA. The number 2 would describe BRAVO.

To enable and describe the message, type

```
1 setRadioMsg "Detonate Bomb";
```

into any 'On Activation' field.

61.4.17 ADD ON SCREEN HINT

To add a hint on screen, type this into the 'On Activation' field:

```
hint "Objective completed";
```

Use hintC to require the user to click "Continue" before proceeding.

Use hintSilent to disable the beep.

To re-display the last hint in-game, hit H.

61.4.18 MAKE HELICOPTER LAND AND DROP OFF TROOPS

To get a helicopter to land and unload troops, start with a manned helicopter or have a unit walk into an empty helo. The unit acting as pilot may NOT be scripted into the driver's seat using moveindriver command.

Give the chopper a landing pad by using the invisible helipad found in Empty/Objects/Invisible helipad. Let's name it "pad1".

Now place the pilot's waypoint on this pad and select TRANSPORT UNLOAD as the type. Type

```
this land "pad1";
```

in the waypoint's 'On Activation' field.

Place another waypoint away from it so he will fly off after unloading. That's it.

If you have trouble unloading all of the units in cargo, give them a GET OUT waypoint near the helipad and synchronize it with the TRANSPORT UNLOAD waypoint. This will cause the helo to wait until the squad exits the helo before taking off.

61.4.19 CREATE MISSION NAME AND IMAGE IN MENU

To make a nice polished look for your mission when found in the MP mission list, create a file called description.ext in your mission's folder. Enter the following lines at the very top:

```
author="Ganst949";  
onLoadName = "Jurassic Convoy";  
OnLoadMission = "Take a mixed convoy to the evac point without being eaten alive!";  
overviewPicture = "JurCon.jpg";
```



Save the file. Now go into the editor and save your mission as a user mission and an MP mission. Done!

61.4.20 ZEUS PLAYER RESPAWNS IN USER MADE MISSION

When playing as Zeus in an ARMA 3 user made mission, you may have found that dropping a respawn module doesn't work for human players. They still die indefinitely. To fix this, open or create a description.ext file in your mission's folder. After the intro lines (see above) paste this:

```
respawn = 3; //Type of respawn (0=NONE, 1=BIRD, 2=INSTANT, 3=BASE, 4=GROUP, 5=SIDE)
respawnDelay = 10; //Time before respawning player
respawnVehicleDelay = 30; //Time before respawning vehicle
respawnTemplatesWest[] = {"MenuPosition","MenuInventory"}; //Allows BLUFOR player to pick respawn
points and loadout
respawnTemplatesEast[] = {"MenuPosition","MenuInventory"}; //Allows OPFOR player to pick respawn
points and loadout
respawnTemplatesGuer[] = {"MenuPosition","MenuInventory"}; //Allows INDEPENDANT player to pick
respawn points and loadout
respawnOnStart = 0; //Respawns player on missson startup if set to "1"
disabledAI = 0; //Disables AI taking over non-filled player slots if set to "1"
joinUnassigned = 0; //Player will be auto-assigned to side with fewest players if set to "0"
aiKills = 0; //Enables score list for AI players if set to "1"
```

61.4.21 CREATE FLOATING CAMERA

Place a unit (named Bob for example) in the map to use as the camera host.

Place a trigger of no size on the map.

Set the 'Activation' of the trigger to Radio Alpha. Type

```
Bob exec "camera.sqs"; showcinemaborder false;
```

Enter the game and fire off the Radio Alpha trigger by hitting 0, 0, 1.

Use the 'Bulldozer' controls found in the Controls menu to control the camera.

Hit 'L' to hide the crosshair and use 'N' to change to Thermal vision or Night vision.

61.5 OBJECT MANIPULATION

61.5.1 SET UNIT/OBJECT HEIGHT

This can be used as a Condition, Initialization or Activation. It will immediately place the unit at the desired height. In any of the fields described above, type:

```
this setPos [(getpos this select 0),(getpos this select 1),20];
```

The height is declared by the last number, in meters. If this is typed into a units 'Initialization' field, 'Bob' can be substituted with 'this'.

61.5.2 SET FLYING HEIGHT

To set the flying height of an aircraft at each waypoint, just type the code

```
p1anel flyinheight 60;
```

into the 'On Activation' field of the waypoint. 60 is in meters.

61.5.3 CHANGE AN OBJECTS ANGLE

To angle an object on any axis, use this code:

```
car1 setVectorUp[0,0,0];
```

or

```
car1 setVectorDir[0,0,0];
```

The numbers represent x, y, z axis.

61.5.4 SET FUEL CAPACITY

"setfuel" has a value from 0 to 1 with 1 being 100% fuel. Upon waypoint activation: in waypoint's 'On activation' field type

```
car1 setfuel 0.5;
```



Upon trigger activation: in trigger's 'On activation' field type

```
car1 setfuel 0.5;
```

61.5.5 SET UNIT DAMAGE

"setdamage" is purposely spelled wrong. It has a value from 0 to 1 with 1 being 100% damage. Upon waypoint activation: in waypoint's 'On activation' field type

```
Bob setdamage 0.5;
```

Upon trigger activation: in trigger's 'On activation' field type

```
Bob setdamage 0.5;
```

Se si vuole distruggere un oggetto e dare un 100% di danno quindi dobbiamo mettere

```
nomeoggettounità setdamage 1;
```

61.5.6 LIGHT/PUT OUT CAMPFIRE ON COMMAND

Use this code in the 'On Activation' field to light a campfire:

```
fire1 inFlame true;
```

Switch to false to put it out.

61.5.7 UNITS TELEPORT INTO VEHICLE

For each unit use

```
Bob moveincargo car1;
```

This can be used in the 'Initialization' field of a unit from the beginning or used in the 'On Activation' field of a trigger or waypoint.

The word 'cargo' can be swapped with 'gunner', 'driver', 'commander', and 'turret' to force these positions. If trying to place more than one unit as gunner, use

```
Bob moveinturret [heli, [0]];
```

where #0 is gun position 1, #1 is position 2, etc.

61.5.8 DO NOT ALLOW FLEEING OF A UNIT

When a unit feels it is at a disadvantage it may flee. To disable this, the code may be used as an 'Activation' or 'Initialization'. Type:

```
Bob allowfleeing 0;
```

To allow change 0 to 1.

61.5.9 CREATE A SMOKE GRENADE XX METERS ABOVE UNIT

This code, if put into the 'On Activation' field of a trigger will create a purple smoke grenade 50 meters above Bob's head:

```
Bobsmoke = "SMOKESHELLPURPLE" createVehicle [(getPos Bob select 0),(getPos Bob select 1),50];
```

You must use the correct object ID "SMOKESHELLPURPLE" or the trigger won't work. Change to RED, YELLOW, BLUE, ORANGE for different colors or use classnames for other objects like grenades, vehicles, etc.

61.5.10 ACCENDERE IL MOTORE DI UN VEICOLO

Accendere il motore di un veicolo di nome veh

```
_veh engineOn true;
```

Funzione **engineOn**

Syntax:

vehicle **engineOn** state

Parameters:

vehicle: [Object](#)

state: [Boolean](#) - true to turn the engine on, false to turn it off

Return Value:

[Nothing](#)

**Example 1:**

```
if (!isEngineOn _jeep) then { _jeep engineOn true; };
```

Example 2:

```
waitUntil { player == driver _jeep; };  
vehicle player engineOn true;
```

61.5.11 CHANGE SPEED OF A UNIT

To change the speed of a unit or limit it there are several codes:

```
Bob setSpeedMode "Full";
```

is available in Auto, Limited, Normal, Full.

```
Bob forceSpeed 120;
```

forces the unit to move at a certain speed up to the max speed for the unit.

```
Bob limitSpeed 60;
```

limits the units speed to 60kph; Bob setSpeed getSpeed Nick; Bob matches speed of Nick;

61.5.12 REMOVE ALL WEAPONS AND ADD WEAPONS TO UNIT

Use this to remove all of the units weapons, not counting accessories. It can be used as an 'Activation' or 'Initialization'.

Type:

```
removeAllWeapons Bob;
```

Use this to add specific weapons to a unit. Used the same way as the removal code, type

```
Bob addWeapon "NVGoggles";
```

It only works with weapon codes that are present in the version of the game that is installed. For instance, any weapon code starting with ACE will only be available if ace is installed.

First use addweapon and addmagazine to add the desired weapons.

Now it is necessary to remove the old unwanted items.

AT THE END of the add codes type Bob removeweapon 'ACE_M9'; or Bob removemagazine "12rnd_9mm_m9"; once for each unwanted gun or mag that the item normally spawns with.

EXAMPLE: If a unit normally respawns with an M4A1, six mags and two grenades, and you just want to add an M16A2, and have a total of eight mags, add two mags and the M16A2 and only remove the M4A1 at the end of the code. Refer to the weapon code list for specific weapon codes.

To preset custom loadouts for a unit, create an ARMAEdit Custom Unit script and save it as 'player1.sqs'. for example. Use

```
[player1] exec "player1.sqs";
```

in a triggers 'On Activation' field to activate it. Be sure the 'Condition' says

```
alive player1;
```

61.5.13 ADDING OR REMOVING WEAPONS AND AMMO FROM AN AMMO BOX

To remove weapons and ammo from an ammo box, this code can be used in the 'Initialization' field of the ammo box or the 'On Activation' fields of a trigger or waypoint. Type:

```
clearweaponcargo box1; and clearmagazinecargo box1;
```

To add weapons and ammo to a box use this code instead:

```
box1 addweaponcargo ["AKS_GOLD", 5]; box1 addmagazinecargo ["30rnd_762x39_ak47", 120];
```

The last number in each set of brackets refers to the quantity of items added to the box. Refer to the weapon code list for specific weapons and magazines. Remember, some codes don't work.

61.5.14 NOT ALLOW UNIT TO ENTER VEHICLE

To block a unit from entering a certain vehicle use this code in the 'Initialization' field of the vehicle:

```
[Bob, Nick] allowGetIn false;
```

Change false to true to allow access. Add as many units as needed to the brackets.



61.5.15 ATTACH/DETACH AN OBJECT TO ANOTHER

To attach an object to another object/unit use this code in either the 'Initialization' field or 'On Activation' field:

```
box1 attachTo [Bob,[ 0, 0, 0]];
```

The first 0 is fore-aft, the second is right-left, the third is up-down. Use a negative sign to declare aft, left or down.

Now to detach the object on command use the detach command by typing this into the 'On Activation' field of a trigger:

```
detach box1;
```

Be aware, ammo boxes will detach and remain exactly where they are detached even if they are in mid air.

61.5.16 DISPLAY THE SPEED OF A UNIT

To display a unit's speed as a hint, use this code:

```
hint format ["Your Speed: %1" , speed car1];
```

where %1 is where the speed will input itself. You may change "Your Speed" to any text.

61.5.17 MAKE A UNIT PRISONER

To make a unit a prisoner of the enemy and not get shot by the enemy type this in an 'On Activation' field:

```
Bob setCaptive true;
```

Change true to false to disable.

61.5.18 NAME A GROUP

To name a group to be used later in the mission, it is necessary to declare this in the group leaders 'Initialization' field:

```
police= group this;
```

61.5.19 UNIT JOINS GROUP

Used in an 'On Activation' field, this will make a unit join a pre-declared group as explained above:

```
[Bob] join police;
```

61.5.20 TELEPORT A UNIT OR OBJECT TO ANY POSITION

Posizionare innanzitutto un marker nella posizione in cui si desidera spostare l'unità / oggetto. Assicurati di dargli un nome come "putBobHere".

Quindi digitare questo in qualsiasi campo "On Activation":

```
Bob setPos[(getMarkerPos "putBobHere") select 0, (getMarkerPos "putBobHere") select 1, 0];
```

All'attivazione del trigger o del waypoint, Bob si teletrasporta sul marker "putBobHere". Tieni presente che, se Bob ha già raggiunto dei waypoint, continuerà a camminare fino al prossimo waypoint.

61.5.21 IMPOSTA LA POSTURA DI UN'UNITÀ

Per far stare in piedi un'unità, accovacciarsi o sdraiarsi a comando, usa questo codice in qualsiasi campo " On Activation ".

```
Bob setUnitPos "Up";
```

Questo lo farà stare in piedi. Puoi anche usare "Middle", "Down" e "Auto" per farlo accovacciare, sdraiare e resettarlo su auto.

61.5.22 CREATE A NAME FOR NEAREST BUILDING (FOR DESTROYING PARTICULAR BUILDINGS)

To name a building nearest an object or unit, use this code:

```
House_1=nearestbuilding Bob;
```

House_1 is the name the user can give to the building and can be anything. This is useful for destroying particular buildings on command. One needs only to type

```
House_1 setdammage 1;
```

into an 'On activation' field to destroy the selected building.

61.5.23 IMPOSTARE IL BLOCCO DI UN VEICOLO

Per abilitare / disabilitare il blocco sui veicoli, utilizzare questo semplice codice in un campo " On Activation ":



```
car1 lock true;
```

per bloccare il veicolo. Cambia in “falso” per sbloccare un veicolo.

61.5.24 RANDOMIZE UNITS/OBJECT START LOCATION

To do this, you need to utilize multiple markers.

Place the unit and multiple markers to act as starting points.

Group these markers to the unit/object.

When the game starts, the unit/object will randomly pick a marker to start at.

You can also double click the unit and set its 'Placement Radius' to further randomize its location.

61.5.25 MOVE MARKER TO ANOTHER MARKER

To teleport a marker to another one, name both markers and use this code in any 'On Activation' field:

```
"mark1" setMarkerPos getMarkerPos "mark2";
```

This moves mark1 from its current position to mark2.

61.5.26 MAKE A UNIT MOVE TO ANY OBJECT/MARKER

There are several codes to use. These should be inserted into 'On Activation' fields:

Using objects :

```
Bob doMove getPos Nick;
```

Using ID's :

```
Bob doMove getPos (Object ID);
```

Using coordinates:

```
Bob doMove [X,Y,Z];
```

Using markers :

```
Bob doMove getMarkerPos "MarkerName";
```

61.5.27 METTERE UN'UNITÀ ALL'INTERNO DI UN VEICOLO

Aggiungere un veicolo e nominarlo, es “boat_1”

Nell'azione di un trigger o sqf usare la stringa seguente per mettere l'unità all'interno del veicolo selezionato

```
player moveInCargo boat_1;
```

Player può essere sostituito anche da this o dal nome dell'unità

Funzione ***moveInCargo***

Description:

Move soldier into vehicle cargo position (Immediate, no animation).

Syntax:

unitName moveInCargo vehicle

Parameters:

unitName: [Object](#)

vehicle: [Object](#)

Return Value:

[Nothing](#)

Example 1:

```
_soldierOne moveInCargo _jeepOne;
```

Alternative Syntax

Syntax:

unitName moveInCargo [vehicle, CargoIndex]

Parameters:

unitName: [Object](#)

vehicle: [Object](#)

CargoIndex: [Number](#)

**Return Value:**

[Nothing](#)

Example 2:

```
_soldierOne moveInCargo [_jeepOne, 1];
```

61.5.28 MOVING OBJECTS, TRIGGERS OR MARKERS

Use these codes in the 'On Activation' field:

Using an object:

```
Bob setPos getPos Nick;
```

In front, behind, beside an Object:

```
Bob setpos Nick modelToWorld [0,3,0];
```

Using an ID:

```
Bob setPos getPos (Object ID);
```

Using coordinates:

```
Bob setPos [X,Y,Z];
```

Using markers:

```
Bob setPos getMarkerPos "Marker1";
```

From marker to marker:

```
"Marker1" setMarkerPos getMarkerPos "Marker2";
```

From marker to object:

```
"Marker1" setMarkerPos getPos Bob;
```

Using a vehicle:

```
Bob setPos getPos vehicle car1;
```

Using a vehicle II:

```
"Marker1" setMarkerPos getPos vehicle car1;
```

Moving a trigger to object

```
vehicle nameoftrigger setPos getPos Bob
```

61.5.29 MAKE OBJECT INVISIBLE

To make an existing object or unit invisible on command for singleplayer, type

```
car1 hideobject TRUE;
```

into any 'On Activation' or 'Initialization' field. To reverse it, use FALSE.

To make an object invisible to all players use

```
_nic = [nil, _plane, "per", rHideobject, false] call RE;
```

in the script. '_plane' is the name of the object being made invisible. 'false' is off while 'true' is on.

61.5.30 AFFECT ALL UNITS IN A GIVEN PARAMETER

To cause something to happen to all units near an object use:

```
{_x setdammage 1} forEach nearestObjects [GBUTarget, ["man"],20];
```

'Man' is the type of unit and 20 is the radius of effect.

The format is: {_x DosomeAction} forEach UnderWhatCondition;

Use

```
{_x setdammage 1} forEach units group Bob;
```

to kill every member of Bob's group.

Use

```
{_x setcaptive TRUE; RemoveAllWeapons _x;} forEach units group Bob;
```

for multiple commands.

61.5.31 SHOW CLASSNAME OF UNIT/OBJECT

To display a hint showing the classname of a unit or object type:

```
hint format ["%1", typeof this];
```



in the unit's Initialization field.

61.5.32 SET A WAYPOINT TO A AI UNIT TO GET A WEAPON FROM AN AMMO BOXES

One thing to know about AI is that if the unit is a soldier and doesn't have ammo or a weapon, they are going to run up to a ammo box anyway. Now to give them a specific weapon:

Create a waypoint for your unit to move to the box.

Name the ammo box something like **ammo1**.

Name the AI unit **Bob**.

Now either type the following in the waypoint or create a trigger that has 0/0 size and set it to 'Once'. In the 'Condition' field type:

```
Bob distance ammo1 < 1.5;
```

In the 'On Activation' field type:

```
Bob addmagazine "30Rnd_556_STANAG"; Bob addweapon "M4a1"; Bob selectweapon "M4a1";
```

61.5.33 CREATING VEHICLES

```
// Will spawn vechicle at player, or player(s) location. "C_Offroad" = cfg name.  
_veh = "C_Offroad_01_F" createVehicle position player;
```

61.5.34 CREATE DOGS

```
// Create group of Dogs at player or players locations.  
_dog = createAgent ["Fin_random_F", getPos player, [], 5, "CAN_COLLIDE"];
```

61.5.35 MAP TELEPORT

```
// This will teleport you around the map freely by alt then left clicking.  
player onMapSingleClick "if (_alt) then {player setPosATL _pos}";
```

61.5.36 GOD MODE

```
// You will take damage, however you will not die.  
player allowdamage false;
```

61.5.37 KILLS PLAYER

```
// Kills player, or player(s)  
player setdamage 1
```

61.5.38 HEAL PLAYER, OR PLAYER(S)

```
// Heals yourself, or all players.  
player setDamage 0;
```

61.5.39 DESTROY TARGETS

```
// Will Destory whatever object at crosshair.  
cursortarget setdamage 1
```

61.5.40 DISABLE FATIGUE (WITHOUT ACE LOADED)

```
// Disable now  
player enableFatigue false;  
// Disable on next spawn  
player addEventHandler ["Respawn", {player enableFatigue false}];
```

61.5.41 REPAIR VECHICLE

```
// Will repair any Vechicle you're currently in.  
_timeForRepair = 0; _vehicle = vehicle player; hint format ["Please wait %1 seconds for  
repair/flip",_timeForRepair]; sleep _timeForRepair; if (_vehicle == player) then {_vehicle =
```



```
cursorTarget;}; _vehicle setfuel 1; _vehicle setdamage 0; _vehicle = nil; vehicle = this select 0;
_vehicle setvectorup [0,0,1];
```

61.5.42 REMOVE FUEL FROM PLAYER, OR PLAYER(S) VEHICLES

```
// Rremove all fuel from all active vehicles, or your own.
vehicle player setfuel 0;
```

61.5.43 ADD WEAPON TO PLAYER OR PLAYER(S)

```
// Add weapon via ID to player, or players.
player addweaponglobal "arifle_MX_GL_F";
```

61.5.44 SET PLAYER AMMO

```
// Set current player, or player(s) ammo by adjusting the number.
player setAmmo [currentWeapon player, 1];
```

61.5.45 SEE ALL PEOPLE ON THE MAP

```
// Display ALL players and spawned units on the map.
if(stealthMarkerToggle == 1) exitWith {stealthMarkerToggle = 0; onEachFrame {; {deleteMarkerLocal _x;}
forEach markerList; hint "Markers disabled";}; stealthMarkerToggle = 1; markerList = []; markerUnits =
[]; hint "Markers enabled - Check map!"; while {true} do { if(stealthMarkerToggle == 0) exitWith {}; {
_unit = _x; markerUnits = markerUnits + [_x]; _markerName = str(format ["%1",name _x]); _mName = "m" +
_mName; //player sidechat format ["%1",_markerName]; if(side _x == side player) then { _mName =
createMarkerLocal [_markerName, position _x]; _mName setMarkerSizeLocal [0.6, 0.9]; _mName
setMarkerShapeLocal "ICON"; _mName setMarkerTypeLocal "mil_triangle"; _mName setMarkerColorLocal
"ColorBlue"; _mName setMarkerTextLocal _markerName; _mName setMarkerDirLocal (direction _x); markerList
= markerList + [_mName]; } else { _unit = _x; markerUnits = markerUnits + [_x]; _mName
setMarkerSizeLocal [0.6, 0.9]; _mName = createMarkerLocal [_markerName, position _x]; _mName
setMarkerShapeLocal "ICON"; _mName setMarkerTypeLocal "mil_triangle"; _mName setMarkerColorLocal
"ColorRed"; _mName setMarkerTextLocal _markerName; _mName setMarkerDirLocal (direction _x); markerList
= markerList + [_mName]; }; //hint format ["%1",_mName]; } forEach allUnits; sleep 1;
if(stealthMarkerToggle == 0) exitWith {}; {_x setMarkerPosLocal getPos (markerUnits select (markerList
find _mName)); _x setMarkerDirLocal getDir(markerUnits select (markerList find _mName));} forEach
markerList; sleep 1; if(stealthMarkerToggle == 0) exitWith {}; {_x setMarkerPosLocal getPos
(markerUnits select (markerList find _mName)); _x setMarkerDirLocal getDir(markerUnits select
(markerList find _mName));} forEach markerList; sleep 1; if(stealthMarkerToggle == 0) exitWith {}; {_x
setMarkerPosLocal getPos (markerUnits select (markerList find _mName)); _x setMarkerDirLocal
getDir(markerUnits select (markerList find _mName));} forEach markerList; sleep 1;
if(stealthMarkerToggle == 0) exitWith {}; {_x setMarkerPosLocal getPos (markerUnits select (markerList
find _mName)); _x setMarkerDirLocal getDir(markerUnits select (markerList find _mName));} forEach
markerList; sleep 1; if(stealthMarkerToggle == 0) exitWith {}; {deleteMarkerLocal _x;} forEach
markerList; markerUnits = []; markerList = []; };
```

61.5.46 SHOW ALL VEHICLES ON THE MAP WITH MARKERS

```
// Show ALL spawned Vechicles on the map. (Need testing if it updates in real time)
if (isnil "gggggggggggggggggggg") then {gggggggggggggggggg=0};

if (gggggggggggggggggg==0) then
{
hint "Adding Vehicle Markers";
gggggggggggggggggg=1;
VL = vehicles;
j = count VL;
i = 0;
MV = true;

while {MV} do
{
VL = vehicles;
j = count VL;
i = 0;

for "i" from 0 to j do
{
veh = VL select i;
deleteMarkerLocal ("VM"+ (str i));
mk2 = "VM" + (str i);
```



```
mk2 = createMarkerLocal [mk2,getPos veh];
mk2 setMarkerTypeLocal "waypoint";
mk2 setMarkerPosLocal (getPos veh);
mk2 setMarkerColorLocal("ColorGreen");
mk2 setMarkerTextLocal format ["%1",typeOf veh];
};
sleep 0.5;
};
}
else
{
hint "VM Stopping";
i = 0;
MV = false;
gggggggggggggggggggggg=0;

for "i" from 0 to j do
{
veh = VL select i;
deleteMarkerLocal ("VM"+ (str i));
};
```

61.5.47 ATTACHING A MARKER TO A UNIT

Short and sweet, thanks to shk for this code you could run from your init.sqf file:

```
[ ] spawn {
    while {not isnull Unit1} do { "mkr_unit1" setmarkerpos getpos Unit1; sleep 0.5; };
};
```

61.6 COMPLEX FUNCTIONS

61.6.1 UNIT SETS PIPEBOMB

First, the unit must possess a pipebomb. Russian Spetznaz Operators carry these. Create a waypoint and in the waypoint's 'On activation' field type:

```
Bob fire "PipeBombMuzzle"; Bob playMove "AmovPercMstpSnonWnonDnon_AinvPknLMstpSnonWnonDonon";
```

Now to detonate it make another waypoint or trigger, and in the 'On activation' field type:

```
Bob action ["TOUCHOFF", Bob];
```

61.6.2 IMPROVISED EXPLOSIVE DEVICE

To place a randomly located, random fail rate IED click 'Units', 'Class', and set as 'Objects'.

Select any item from 'Unit' list as your IED and name it "IED". To randomize, also set multiple named markers nearby set as 'Empty'.

Group them to the IED by clicking 'Groups' and dragging lines between them. Go to directory in documents/arma2/missions/'mission name'. Create new text file and name 'IED.sqf'. In it type:

```
_shellArray =
[
"B-30mm_HE",
"ARTY_Sh_81_HE"
"ARTY_Sh_120_HE"
];
_shell = _shellArray call BIS_fnc_selectRandom;
_explosion = _shell createvehicle position IED;
```

Create activating trigger circle and type

```
_nil = [ ] execVM "IED.sqf"
```

in the 'On Activation' Field.



61.6.3 MASS PARADROP

Make infantry 'Group' and name group leader something like "Bob". Place an aircraft and name it something like "plane1". In group leader Bob's 'Initialization' field write

```
SWAT=group this;
```

with SWAT being the desired group name. Also write

```
{Bob moveincargo plane1}foreach units group this;
```

in Bob's 'Initialization' field. Create a trigger to eject them and 'Synchronize' it to the paradrop group leader Bob. Type

```
{bob action ["eject", plane1]} foreach units SWAT;
```

in the triggers 'On Activation' field.

61.6.4 CALL AIR SUPPORT

Create a support unit like a BLUFOR Ah64 Apache, and name it Apache1. Make sure it is not set to 'Flying'. Slide the 'Fuel' bar to 0% fuel.

Give the aircraft a waypoint in the direction of the target area.

Make another waypoint at the target. On this waypoint set 'type' to 'Seek and destroy'.

Place 'Trigger' near vehicle with no size.

Set 'Activation type' to 'Radio alpha', bravo etc. In 'Text' type the name of the support action ie. 'Airstrike'.

Now type

```
Apache1 setfuel 1;
```

in the 'On Activation' field of the trigger. When Radio Alpha or "Airstrike" is selected in-game, the pilot will start engines and fly to the target to seek and destroy.

61.6.5 HALO JUMP

To make a unit spawn at any altitude and skydive followed by the option to open a steerable parachute simply write this in the 'On Activation' field of a trigger or waypoint:

```
[Bob,2000] exec "ca\air2\halo\data\Scripts\HALO_init.sqs";
```

where Bob will spawn at his current position, but at an altitude of 2000 meters.

You may use **getpos heli1** in place of **2000** to HALO out of an aircraft.

HINT: You could precede this code with the teleportation code to make him jump elsewhere.

61.6.6 CREATE SUPPORT OPTIONS

This will give the player the option to call in vehicle airdrop, UAV, ammo supply drop, tactical airstrike, and artillery barrage.

Create player unit and place 'SecOps Manager' module nearby and 'Synchronize' them.

Create 'Radio Alpha' trigger and name it something like "Support" and set to trigger to 'Repeatedly'. The 'On Activation' field should read

```
[["transport", "aerial_reconnaissance", "supply_drop", "tactical_airstrike", "artillery_barrage"],  
player] call BIS_SOM_addSupportRequestFunc;
```

You may omit any of the items in quotes to limit or expand your support options. You may also change "player" to a unit name.

As is, the module will send you random objectives every couple of minutes and once completed will give you free support.

To disable the objectives go to the SecOps Manager module and type this into the "Initialization" field:

```
this setVariable ["settings",[[], true, nil, nil, false]];
```

To utilize this in-game, hit "0", "0", "Radio Alpha". It will show a window saying "Support Available". Hit "0", "8" and then "Request Support". The support option will show up and you select what you want and follow the prompts. Airstrikes can be directed with a SOFLAM Laser Designator as well.

To disable various support options, type this into the "Initialization" field of the module:

```
BIS_SSM_Ammodrop_available_west = false;
```

```
BIS_SSM_unitsdrop_available_west = false;
```



```
BIS_SSM_mortar_available_west = false;
```

```
BIS_SSM_artillery_available_west = false;
```

```
BIS_SSM_casfire_available_west = false;
```

You may also create player(forward air controller) and locate 'Simple support' module nearby and synchronize them.

61.6.7 CREATE PLAYABLE UAV

Create unit MQ-9 etc. & set to flying.

Create 'UAV' module and create empty vehicle or object such as a TV or line keeper's box to act as terminal.

Synchronize player, MQ9 & terminal to module.

To move UAV's flight pattern on the map, just click on the map while you are in the UAV. Also, while using the UAV camera just point the crosshair at any location below to cause the aircraft to circle that point.

Hit tab to lock onto vehicular targets.

61.6.8 HELICOPTER AIRLIFTS OBJECT

To make an object attach itself to a vehicle like a helicopter start by naming both vehicle and object.

Now create 3 triggers with no size. The first should have

```
car1 distance heli1 < 8;
```

written in the 'Condition' field. 8 meters seems to work well.

In the 'On activation field type

```
car1 attachTo [heli1,[ 0, 1, -10]];
```

where 0 is left to right, 1 is front to back and -10 is up and down. This code will attach car1 10 meters below and 1 meter ahead of heli1's centerpoint.

This is the best configuration for most helicopters.

Set this trigger's 'TIMEOUT' to about 2 seconds to simulate the time required to attach the hook.

Now in the second trigger's 'Condition' field write

```
(getpos car1 select 2) <0.5;
```

In the 'On Activation' field write

```
detach car1;
```

This will detach car1 from heli1 when car1 discovers that it is again under 0.5 meters off of the ground.

You could also set the 'Condition' to **this**; and set the trigger 'Activation' to 'Radio Alpha'. This will detach car1 on command.

BE SURE TO SET ALL TRIGGERS TO 'Repeatedly'.

If you want another helicopter to be able to pick up car1, just copy the first trigger and change the name of the helicopter.

61.6.9 MC-130 REFUELER

To build a refueler/rearming capable C-130 start by creating a C-130(empty or piloted) and naming it 'c130'.

Now create the boom by placing a 'flagpole' (Objects) and a '10cm Sphere' ('Objects -Helpers').

Name the flagpole 'boom' and the sphere 'nozzle'.

Type this into the boom's 'Initialization' field:

```
boom setVectordirandup [[0,.5,-.6],[0,.5,0]];
```

Now type

```
nozzle attachTo [boom,[ -.1, -.35, -4.2]];
```

into the nozzle's 'Initialization' field.

Ok, now we need 5 triggers, all with no size and set to 'Repeatedly'. The first needs to be set to 'Radio Alpha' and called 'Extend boom'. Type this into the 'On activation' field:

```
boom attachTo [c130,[ 0, -11, -4]]; boom setVectordirandup [[0,.5,-.6],[0,.5,0]]; nozzle attachTo [boom,[ -.1, -.35, -4.2]];
```

Now create a trigger set to 'Radio Bravo', called 'Retract boom' write

```
detach boom; boom setPos[(getMarkerPos "marker1") select 0, (getMarkerPos "marker1") select 1, 0];
```



Place a marker called 'marker1' anywhere on the map.

Ok, now we need to allow any receiving aircraft to attach itself when near. Make a trigger and type this into the 'Condition' field:

```
j1 distance nozzle < 3;
```

'J1' is the receiving aircraft's name so change accordingly. In the 'On Activation' field write this for an:

F-35:

```
j1 attachTo [nozzle,[ 0, -1, -2.6]]; j1 setVectorup [0,-1,1];
```

A-10:

```
j1 attachTo [nozzle,[ 0, -3.6, -6]]; j1 setVectorup [0,-1,1];
```

AV-8B:

```
j1 attachTo [nozzle,[ 0, -2, -1.5]]; j1 setVectorup [0,-1,1];
```

The next trigger will refuel/rearm 'J1' after 20 seconds of being attached. Be sure to set the 'TIMEOUT' to 20 seconds.

In the 'Condition' field write:

```
j1 distance nozzle < 5;
```

Write

```
j1 setfuel 1; j1 setvehicleammo 1;
```

in the 'On activation' field for *fuel*; *ammo*.

Now for the last trigger. Name it 'Detach aircraft' and set to 'Radio Charlie' or set it to go off 20 seconds after j1 attaches itself to the nozzle. In the 'On activation' field write

```
detach j1; That's it!
```

Now when the F-35 called 'J1' flies up very close to the red sphere at the end of the boom, the F-35 will "stick" to the boom and after 20 seconds will take on full fuel and ammo.

Either operator may then hit 'Detach Aircraft' (Radio Charlie) and the F-35 will fall away from the refueler.

61.6.10 ADD LOADOUTS TO AIRCRAFT

To add to the loadout of your aircraft just add this to the 'Initialization' field of the aircraft or to the 'On activation' field of any trigger:

```
this addweapon "Mk82BombLauncher_6"; this addmagazine "6rnd_Mk82";
```

If added to a trigger, be sure to change this to the aircraft's name.

Refer to ARMAII_library.pdf or the config_viewer for weapon codes.

61.6.11 RECORD PATH FOR AI TO FOLLOW

To make the AI follow an exact path or flight pattern, you can use a recording function that records your movement in the editor and then makes the AI do the exact same thing in the finished mission.

Start by creating a trigger set to RADIO ALPHA. In the 'On Activation' field write

```
rec = [heli1,200] spawn BIS_fnc_UnitCapture;
```

where heli1 is the name of the vehicle being used and 200 is the maximum amount of time you need. (NOTE: You will be able to stop recording if you use less time, so a little extra time never hurts.)

Be sure to now place the 'Functions' module(f7) on the map (not required in ARMA 3!).

Create a notepad document and call it init.sqf. Place this inside of the document: MyPath = compile preprocessFile "path.sqf";

Now create another notepad document and name it path.sqf. Be sure it is not saved as path.sqf.txt or path.sqf.sqs but as path.sqf. In this one type:

```
Path = ; [heli1, Path] spawn BIS_fnc_UnitPlay;
```

Be sure both of these documents are saved in the mission's folder.

Now run the mission in the editor and get into the vehicle. Do not start the engine yet!

Hit RADIO ALPHA and then start the engine.

Proceed to drive the intended path from start to finish.

When complete, hit ESC once to bring up the menu, and hit ESC again to stop recording.

View the info that appears on the screen and then press F1 to save the data.

Go immediately to path.sqf and click your cursor between the = sign and the semi-colon after **Path** and hit Ctrl V to paste.



Save the document.(NOTE: For additional paths, save them on seperate documents and name them path2.sqf, path3.sqf etc. For the init.sqf document be sure to copy the first line in init.sqf and change to path2 as shown below):

```
MyPath = compile preprocessFile "path.sqf";
```

```
MyPath2 = compile preprocessFile "path2.sqf";
```

```
MyPath3 = compile preprocessFile "path3.sqf";
```

Now to play back the recorded path.

First set the vehicle to be driven by AI and be sure the name is heli1 or whatever you decided on for the recording.

Give the AI a nearby waypoint so that they start the engine prior to the script going off, otherwise there will be no vehicle sounds.

Now simply activate the script by typing this into any 'On Activation' field:

```
rec = [] spawn Mypath;
```

The vehicle will now do everything you did when you were driving it. You can even loop the trigger for things like a loitering MC-130 that continuously orbits the map.



62 NOZIONI DI BASE ED AVANZATE DI SCRIPTING ED EDITOR

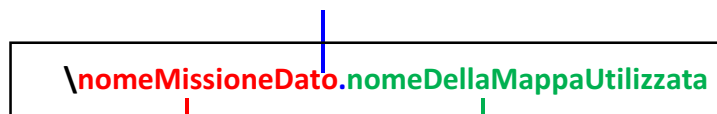
62.1 GESTIONE FILE MISSIONE

Una volta creata la missione procederemo al suo salvataggio.
Facendo salva essa verrà salvata in una cartella che si trova di default in

C:\Documents\Arma 3 - Other Profiles\(\tuoNome)\missions.

All'interno verrà creata una cartella che si chiamerà

N.B: c'è un PUNTO !!



Nome che si assegna alla missione quando si salva.
Si può cambiare direttamente da gestione risorse

Nome della mappa usata.
NON CAMBIARLO MAI !!!!!

All'interno della cartella sarà creato un file missione che si chiama **mission.sqm**

**** NON CAMBIARE MAI NOME AL FILE MISSION.SQM !! *****

Nota: nella schermata di salvataggio dell'Editor c'è una casella di spunta con la possibilità di salvarlo come file binarizzato. Se si sceglie questa opzione il file è illeggibile con un normale Notepad. È consigliabile salvarlo sempre NON BINARIZZATO. Così facendo è leggibile con un notepad e ciò può essere utile poiché contiene diverse informazioni, come ad esempio gli addon usati (come le mod). Anche da Attributi/generali si può spuntare la binarizzazione come salvataggio di default.

È **ALTAMENTE** consigliabile, ma non obbligatorio, dare un nome alla missione (e quindi alla cartella di missione salvata) senza usare gli spazi tra una parola e l'altra.

Es:

\Operazione_Horizon_Fase5.pulau

Nel caso si rinomini la cartella, lasciare sempre il **".nomeDellaMappaUtilizzata"** che identifica la mappa su cui è impostata

Es:

\Op_Horizon_Capitolo_5.pulau

In questo caso aprendo l'editor lo troverete con questo nome

Curiosità: come caso estremo se si cambia il nome della mappa, si trasferisce l'intera missione su quest'ultima. Gli oggetti, però, saranno nelle coordinate in cui risultano già definiti !



FILE PBO

Il file con estensione PBO è un file compresso della cartella missione e di tutto il suo contenuto.

È il formato che serve far caricare al server per far partire la missione.

Può essere usato per “distribuire” la missione all’esterno
2 modi per distribuire una missione

L’intera cartella

`\nomeMissioneDato.nomeDellaMappaUtilizzata`

Il file compresso PBO

Forma del file pbo:

`nomeMissioneDato.nomeDellaMappaUtilizzata.pbo`

esempio: `Operazione_Horizon_Fase5.pulau.pbo`

**** NON CAMBIARE MAI NOME AL FILE PBO !! ****

Creare un file pbo

Direttamente dall’editor (SCOMODISSIMO)

Usando il programma “PBOManager v.1.4 beta”

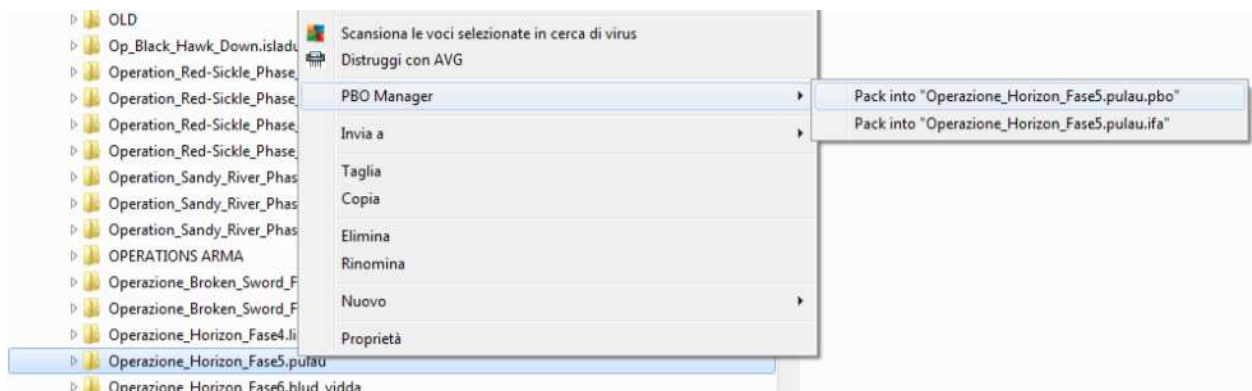
A cosa serve il programma (freware da Armaholic)

Aprire un file pbo

Creare un file pbo

Per creare il file pbo (PBOManager già istallato sul pc):

1. Tasto destro sulla cartella missione



2. Selezionare "Pack into ...". Il file verrà creato a livello della cartella superiore \mission

FILES MISSIONE AGGIUNTIVI

Description.ext ← *Va messo nella cartella missione!*

Viene utilizzato per impostare gli attributi generali della missione o per definire entità globali che saranno disponibili per altri script (suoni, formati..). Si trova nella cartella radice della missione e utilizza la stessa sintassi del file config.cpp, ma supporta solo un numero limitato di classi di configurazione.

Definizione della schermata d'intro iniziale

```
// ***** SCHERMATA DI CARICAMENTO (NOTA: MODIFICARE CON I TESTI VOLUTI)
author = "ENIGX"; // nome autore
onLoadName = "Corso EGX"; // Nome missione
onLoadMission = "Descrizione corso"; // Descrizione breve
loadScreen = "images\intro_screen.jpg"; // Immagine della schermata di caricamento
class Header
{
gameType = Coop; // Tipo della missione MP
};
```

**Il formato dell'immagine di preview (loadScreen) deve essere 2:1 nel formato .jpg oppure .paa.
In formati diversi c'è distorsione dei colori !!!**

Per il formato .paa si usa il programma freeware TexView 2 con il quale si può convertire un .png in .paa

Definizione del respawn

```
// ***** RESPAWN SETTINGS
respawn = 3; //tipo di respawn (0=NONE, 1=BIRD, 2=INSTANT, 3=BASE, 4=GROUP, 5=SIDE)
respawnDelay = 3; //imposta il tempo di ritardo del respawn
respawnDialog = 0; // Mostra tabella e conto alla rovescia per la rigenerazione
respawnOnStart = -1; //fa si che all'inizio missione il respawn sia solo in una posizione predefinita (non mostra la scelta di respawn)
respawnTemplates[] = { "MenuPosition", "Tickets" }; //fa apparire il menu con le posizioni dove respawnare
```



Sarà necessario mettere in editor solo il marker nominato "respawn_west" nella posizione della Base

Disabilitazione delle AI negli slot liberi

```
// *** DISABILITAZIONE AI AD INIZIO PARTITA (SOLO SLOTS PER PLAYERS) E
ABILITAZIONE DELLA CONSOLE PER L'ADMIN

disabledAI=1; // Rimuovo le unità giocabili non controllate da player umani
(1=si/0=no)
debriefing=1; // abilita il debriefing
aiKills = 0; // Enable scorelist for AI players (1=yes/0=no)

// **Abilitazione console
//enableDebugConsole = 1; // 0 - Default behavior, available only in editor; 1 -
Available in SP and for hosts / logged in admins in MP; 2 - Available for everyone
enableDebugConsole[] = {"xxxxxx", "xxxxxxxx"}; // abilito console per gli UIDs
elencati
```

Definizione delle schermate di fine missione (missione fallita o compiuta)

```
class cfgDebriefing
{
    // Used when all players die in MP
    class endDeath
    {
        title = "Both teams have died.";
        description = "Everyone was killed by the bomb.";
        picture = "KIA";
    };
};
```

← Classe "madre" che definisce le schermate di debriefing

← Classe specifica per il finale qui chiamato "endDeath"

Classi debriefing EGX:

```
// ----- SCHERMATE DI DEBRIEFING FINALI (NOTA: MODIFICARE CON I TESTI VOLUTI) -----
class CfgDebriefing
{
    // == MISSIONE COMPLETATA
    class Misscomp // definisce una classe di debriefing chiamata "Misscomp" con parametri
    {
        title = "Missione completata!"; // titolo
        subtitle = "Avete sconfitto le forze nemiche"; // Inserire eventuale sottotitolo
        description = "Un ringraziamento a tutti per la partecipazione"; //descrizione completa nella
finestra di debriefing
        pictureBackground = "images\logoEGXquadroNero.jpg"; // Immagine di sfondo della schermata di
debriefing
        //picture = "b_inf"; // eventuale icona
        pictureColor[] = {0.0,0.3,0.6,1};
    };

    // == MISSIONE FALLITA
    class Missfall // definisce una classe di debriefing chiamata "Missfall" con parametri
    {
        title = "Missione fallita!"; // titolo
        subtitle = "Peccato siamo stati sopraffatti"; // Inserire eventuale sottotitolo
        description = "Un ringraziamento per la partecipazione"; //descrizione completa nella finestra di
debriefing
        pictureBackground = "images\logoEGXquadroNero.jpg"; // Immagine di sfondo della schermata di
debriefing
        //picture = "b_inf"; // eventuale icona
        pictureColor[] = {0.0,0.3,0.6,1};
    };
};
```

← Attenzione al percorso



```

};

/* -----
Nota: si può far terminare la partita eseguendo le "class CfgDebriefing".
In tal caso la partita terminerà nel modo classico di Arma e verrà mostrata la finestra di debriefing
definita sopra.
Per fare questo: fare ESC ed inserire la seguente stringa nella debug console cliccandone l'esecuzione
in GLOBALE

"Misscomp" call BIS_fnc_endMission; // Per Missione Completata
"Missfall" call BIS_fnc_endMission; // Per Missione Fallita

Oppure inserire una delle due stringhe in un trigger da editor. In tal caso occorre sostituirle con le
seguenti per l'esecuzione in globale

["Misscomp",true,1,false,true] remoteExecCall ['BIS_fnc_endMission',0]; // Per Missione Completata
["Missfall",true,1,false,true] remoteExecCall ['BIS_fnc_endMission',0]; // Per Missione Fallita
-----*/

```

- **pictureBackground:** formato 2:1
- **picture:** icona nel formato 1:1

La classe dei finali dei trigger sono End1, End2, ...

```

// Singleplayer / client-side
"end1" call BIS_fnc_endMission;
// Multiplayer / server-side
"end1" call BIS_fnc_endMissionServer;

```

Esempio



Eventuale immagine in
pictureBackground = "nome.jpg";

Logo in picture = "logo.jpg";

icone



Empty	Flag	mil_objective	mil_marker	mil_flag	mil_arrow
mil_arrow2	mil_ambush	mil_destroy	mil_start	mil_end	mil_pickup
mil_join	mil_warning	mil_unknown	mil_circle	mil_dot	mil_box
mil_triangle	hd_objective	hd_flag	hd_arrow	hd_ambush	hd_destroy
hd_start	hd_end	hd_pickup	hd_join	hd_warning	hd_unknown
hd_dot	b_unknown	o_unknown	n_unknown	b_inf	o_inf
n_inf	b_motor_inf	o_motor_inf	n_motor_inf	b_mech_inf	o_mech_inf
n_mech_inf	b_armor	o_armor	n_armor	b_recon	o_recon
n_recon	b_air	o_air	n_air	b_plane	o_plane
n_plane	b_uav	o_uav	n_uav	b_med	o_med
n_med	b_art	o_art	n_art	b_mortar	o_mortar
n_mortar	b_hq	o_hq	n_hq	b_support	o_support
n_support	b_maint	o_maint	n_maint	b_service	o_service
n_service	c_unknown	c_car	c_ship	c_air	c_plane
group_0	group_1	group_2	group_3	group_4	group_5
group_6	group_7	group_8	group_9	group_10	group_11
loc_Tree	loc_SmallTree	loc_Bush	loc_Church	loc_Chapel	loc_Cross
loc_Rock	loc_Bunker	loc_Fortress	loc_Fountain	loc_ViewTower	loc_Lighthouse
loc_Quay	loc_Fuelstation	loc_Hospital	loc_BusStop	loc_Transmitter	loc_Stack
loc_Ruin	loc_Tourism	loc_WaterTower	loc_Power	loc_PowerSolar	loc_PowerWave
loc_PowerWind	flag_NATO	flag_EU	flag_UN	flag_Belgium	flag_Canada
flag_Catalonia	flag_Croatia	flag_CzechRepublic	flag_Denmark	flag_France	flag_Georgia
flag_Germany	flag_Greece	flag_Hungary	flag_Iceland	flag_Italy	flag_Luxembourg
flag_Netherlands	flag_Norway	flag_Poland	flag_Portugal	flag_Slovakia	flag_Slovenia
flag_Spain	flag_UK	flag_USA	flag_Iran	flag_IranArmy	Select
waypoint	selector_selectable	selector_selectedEnemy	selector_selectedFriendly	selector_selectedMission	



Definizione di eventuali suoni personali aggiuntivi

```
// ***** DEFINIZIONE DELLE CLASSI DEI SUONI UTENTE
class CfgSounds
{
    class NomeClasse // This is the classname you will use to refer to the sound in your code
    {
        name = "Nome"; // This line is only needed if you want to be able to select the sound in the Trigger
        effects drop-down menus for sound. Call it whatever you want otherwise remove the line.
        sound[] = {Percorso\FileDelSuono.ogg,3,1,18,1,1,1,0}; // This is the path to the audio file in your
        mission folder. The first number is volume. The second 1 is speed of playback. The 18 indicates the
        distance at which the sound will be heard
        titles[] = {};
        empty[] = {"",0,0,0,0,0,0,0};
    };

    // ... altre classi
};
```

Esempio delle sirene della missione della partenza del razzo

```
class CfgSounds
{
    class alarm01 ← Il nome da richiamare per la riproduzione del suono
    {
        name = "alarm01";
        sound[] = {"sound\alarm01.ogg",3,1,50,1,1,1,0};
        titles[] = {};
        empty[] = {"",0,0,0,0,0,0,0};
    };

    class alarm02
    {
        name = "alarm02";
        sound[] = {"sound\alarm02.ogg",3,1,50,1,1,1,0};
        titles[] = {};
        empty[] = {"",0,0,0,0,0,0,0};
    };
};
```

Il format dei suoni deve essere .ogg

Esistono diversi freeware per convertire in formato .ogg (es. Audacity). Ci sono anche diversi siti online dove si possono convertire mp3 in ogg

Richiamare un file audio all'interno della missione.

Metodo 1 - Trigger

1. mettere un trigger
2. selezionare il suono dal menu a tendina. Comparirà quello che avete scritto in "name".
3. il suono è riprodotto al soddisfacimento del trigger

Metodo 2 - Trigger

1. mettere un trigger
2. nel campo attivazione mettere la stringa



```
this say3D "sound01"; // nome della classe del suono
```

*** NOTA: VIENE RIPRODOTTO SOLO UNA VOLTA ****

Se si vuole che il suono sia riprodotto in loop mettere questa stringa

```
nul = [thisTrigger] spawn {while {true} do {(_this select 0) say3D  
"sound01"; sleep 150;};};
```

Dove 150 (in questo esempio) è la durata del file audio utilizzato in secondi. Cambiarlo all'occorrenza!

Metodo 3 – In un oggetto

1. mettere un oggetto
2. nel suo campo init mettere la stringa

```
this say3D "sound01"; // nome della classe del suono
```

oppure per la durata in loop

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "sound01"; sleep  
150;};};
```

init.sqf ← Va messo nella cartella missione!

File eseguito allo start della missione (prima della finestra di briefing). È eseguito per tutti i player e server.

Usato per l'inizializzazione di molti script che si scaricano da Armaholic.

initServer.sqf ← Va messo nella cartella missione!

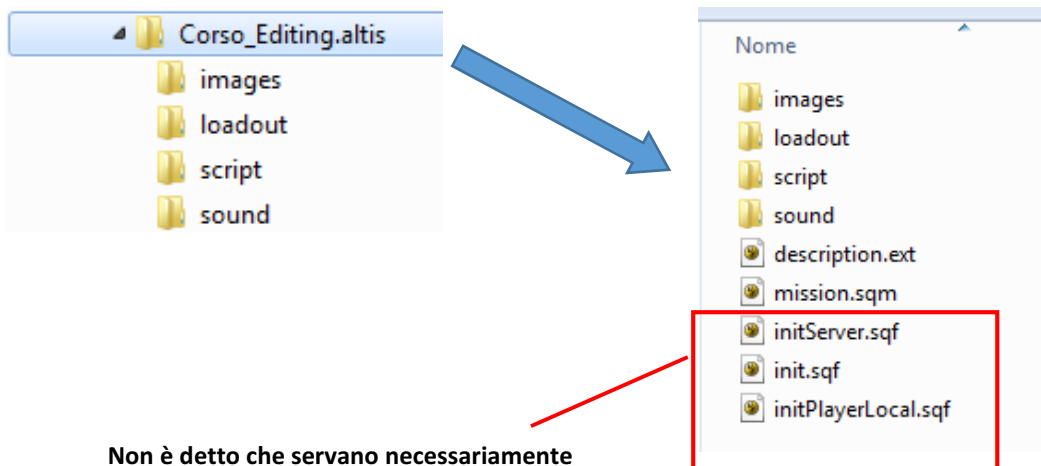
File d'inizializzazione eseguito allo start della missione (prima della finestra di briefing). Analogo dell'init con la differenza che è eseguito solo sul server.

initPlayerLocal.sqf ← Va messo nella cartella missione!

File d'inizializzazione del singolo player. È eseguito solo sul pc del player quando si connette.

Usato per esempio per definire specifiche funzioni al player (es. assegnazione del Tablet)

STANDARDIZZIAMO! STRUTTURA PRINCIPALE CARTELLE





62.2 ALCUNE FUNZIONI BASE DI SCRIPT

Tutte le funzioni di Arma le trovate nella Community Wiki della Bohemia
https://community.bistudio.com/wiki/Main_Page

Messaggio di testo a video

Funzione: **hint**

Consente di far comparire messaggi in alto a destra

Messo per esempio dentro l'attivazione di un trigger.

I testi in Arma vanno messi tra doppio apice

Esempio

```
hint "testo messaggio";
```

Esempio con formattazione del testo

```
hint parsetext "<t font='TahomaB' t size='1.25' t color='#FF0000'>Testo  
messaggio</t>";  
// oppure:  
hint parsetext format["<t font='TahomaB' t size='1.25' t color='#00FF00'> Testo  
messaggio </t>"];
```

Tutto bene finché eseguito dal server (esempio all'interno di trigger)!

Problemi se "eseguito" dal singolo client poiché "hint" è una funzione con effetto locale! Va sostituita con:

```
["Testo del messaggio"] remoteExec ["hint",0,false];
```

Questa esegue la funzione hint in globale

Distruzione oggetto

Funzione: **deleteVehicle**

Consente di cancellare qualsiasi oggetto messo in editor (unità/veicoli/oggetti).

Es.

```
deleteVehicle ufficiale;
```

Elimina dalla mappa l'unità nominata come "ufficiale" nel suo campo init

Es.

```
deleteVehicle oggetto_qualsiasi;
```

Elimina dalla mappa l'oggetto denominato come "oggetto_qualsiasi" nel suo campo init

La funzione opposta, ossia il creare oggetti è la seguente

createVehicle → per gli oggetti e veicoli

createUnit → per le unità

Necessitano di una forma della stringa diversa



Controllo di uccisione/distruzione unità nemica

Funzione: **!alive**

Consente di verificare se un'unità è viva o morta oppure se un veicolo/oggetto è distrutto o integro.

ATTENZIONE: Il simbolo ! sta per "not" → significa notAlive (nonVivo)

Il contrario è ovviamente **alive**

Principalmente usato nel campo "Condizione" di un trigger dove in genere verifichiamo che l'unità/oggetto sia morta/ distrutto

Es.

```
!alive ufficiale_nemico;
```

Controlla se l'unità, nominata come "ufficiale_nemico" nel suo campo init, [è morta](#)

Es.

```
(!alive uff1)&&!alive uff2)
```

Esempio con 2 unità nominate uff1 e uff2. Verifica se entrambe [sono morte](#)

Es

```
!alive veicolo_nemico;
```

Controlla se il veicolo, nominato come "veicolo_nemico" nel suo campo init, [è distrutto](#)

Es

```
!alive Oggetto_qualsiasi;
```

Controlla se l'oggetto, denominato qui "oggetto _qualsiasi" nel suo campo init, [è cancellato](#)

Eseguire uno script esterno

Funzione: **execVM**

Consente di avviare uno script esterno con estensione .sqf

Tutti (o quasi) i files script di Arma hanno estensione **.sqf**

Utile per avviare, per esempio all'interno di triggers, files contenenti istruzioni più o meno complesse o lunghe.

Consente di "gestire" istruzioni separatamente dall'editor tramite richiamo di files appositamente scritti.

ATTENZIONE:

la cartella principale (root) della missione è per l'appunto la cartella missione!!

Non occorre includerla nel percorso del file!!

Es

```
execVM "nomefile.sqf";
```

esegue il file che è contenuto dentro la cartella missione (dove vi è anche il file missione mission.sqm)

Es.

```
execVM "cartella_script\nomefile.sqf";
```



esegue il file contenuto nella sottocartella "cartella_script"

L'ORDINE DELLE CARTELLE PRIMA DI TUTTO:
Mettere tutti gli script .sqf personali in una cartella apposita!!
Consigliato chiamarla "script"

[Creare un Marker in mappa](#)

Funzione: **createMarker**

Consente di creare un marker in mappa (per esempio un obiettivo)

Si può per esempio mettere all'attivazione di un trigger.

La sintassi semplificata è della forma

```
createMarker [nome, posizione]
```

Il nome è del tipo stringa di testo (string)

Le variabili "string" vanno messe tra doppio apice

Es.

```
createMarker ["Enemy", [50,50,0]];
```

Crea un marker "Enemy" nella posizione in coordinate [50,50,0].

Nota: il createMarker crea SOLO il marker!! Esso "esiste" ma non è visualizzato in mappa!!

"Enemy" identifica solo il nome della "variabile marker"! Non sarà quello che comparirà in mappa

Il tipo di marker da visualizzare va definito dalle seguenti istruzioni opzionali da aggiungere a quella sopra:

```
"Enemy" setMarkerType "hd_objective"; // definisce il tipo di icona  
"Enemy" setMarkerText "Nemici!!"; // aggiunge una scritta a fianco  
"Enemy" setMarkerColor "colorRed"; // lo fa per esempio rosso  
"Enemy" setMarkerDir 180; // lo fa ruotare di un angolo (utile per le frecce)
```

Lista delle classi di icone con



La posizione in coordinate è scomoda e poco pratica alla sua individuazione esatta in mappa!

Meglio usare un oggetto/entità già piazzata in mappa.

Es.

```
createMarker ["Enemy", getMarkerPos "MarkerVuotoAggiuntivo"];
"Enemy" setMarkerType "hd_objective";
"Enemy" setMarkerText "Nemici!!!";
"Enemy" setMarkerColor "ColorRed";
```

È una posizione

Questa “usa” la posizione di un Marker Vuoto messo preventivamente nell’editor e chiamato "MarkerVuotoAggiuntivo"

La funzione **getMarkerPos** (applicabile ai marker) “prende” la posizione di questo marker vuoto.

Es.

```
createMarker ["Enemy", getpos oggettoqualsiasi];
```

Questa “usa” la posizione di un oggetto messo preventivamente nell’editor e chiamato "oggettoqualsiasi"

La funzione **getpos** “prende” la posizione di un oggetto/unità.

Una volta creato in una posizione, posso successivamente anche spostare il marker in una nuova posizione in mappa tramite la funzione **setMarkerPos**.

Es.

```
"Enemy" setMarkerPos (getMarkerPos "MarkerVuotoAggiuntivo_2");
```

La funzione opposta al createMarker, ossia cancellare un marker, è la **deleteMarker**

Es

```
deleteMarker "Enemy";
```



[Menù d'interazione su oggetto con azione](#)

Funzione: **addAction**

Consente di creare su un oggetto in mappa il menù interazione con rotella con azione.

Applicazione: le INTEL !!

Forma semplificata della sintassi:

```
Oggetto_target addAction ["titolo dell'azione", script_azione]
```

Es

```
this addAction ["Prendi Documento", {DeleteVehicle nomeintel}];
```

messo nell'init dell'oggetto consente, quando ci si è vicini, di azionare il comando "Prendi Documento" dal menu a tendina e cancella l'oggetto denominato "nomeintel"

Es

```
this addAction ["Team Leader", "scripts\Team_Leader_Loadout.sqf"];
```

o analogo

```
this addAction ["Team Leader ", {execVM "scripts\Team_Leader_Loadout.sqf"}];
```

come sopra ma all'azione del comando "Team Leader" esegue lo script del file .sqf indicato

[Mettere unità a bordo di un veicolo](#)

Funzione: **moveInCargo**

Consente di mettere un'unità a bordo di un mezzo.

Es

```
Unita_qualsiasi moveInCargo heliAmico;
```

mette l'unità denominata "unita_qualsiasi" nel suo campo init all'interno del veicolo nominato "heliAmico"

ATTENZIONE: evitare come la peste le vocali accentate nelle istruzioni degli script!!

Es

```
this moveincargo mrapAlleato;
```

messo direttamente nell'init dell'oggetto lo mette nel veicolo denominato "mrapAlleato"

[Attivare un trigger tramite un altro trigger](#)

Funzione: **triggerActivated**

Consente di attivare un trigger che ha come condizione il soddisfacimento di un altro trigger esterno (chiamiamolo "nomedeltriggerchiamato").

Va messo nel campo "**condizione**" del trigger.

Es.

```
Call{triggerActivated nomedeltriggerchiamato}
```

Es.

```
Call{triggerActivated nomedeltrigger1} && Call{triggerActivated nomedeltrigger2}
```



o l'equivalente

```
triggerActivated nomeDelTrigger1 AND triggerActivated nomeDelTrigger2;
```

Questo attiva il trigger solo quando sono stati attivati i due trigger 1 e 2 contemporaneamente

Es.

```
triggerActivated nomeDelTrigger1 OR triggerActivated nomeDelTrigger2;
```

Questo attiva il trigger solo quando almeno uno dei due trigger 1 e 2 è stato attivato (o l'uno o l'altro)

[Attivare un trigger tramite una sola unità \(uso della thisList\)](#)

Funzione: **thisList**

Consente di attivare un trigger da parte di una sola (o più) unità.

Oltre alla spunta di Blufor presente (od altro), aggiungere nella condizione l'istruzione

```
nomeDellUnita in thisList;
```

il trigger si attiverà quando l'unità, nominata "nomeDellUnita" entra nel trigger

Es.

```
nomeDellUnita1 in thisList && nomeDellUnita2 in thisList && nomeDellUnita3 in thisList;
```

Questa include più unità specifiche per l'attivazione (3)

Un modo alternativo più breve per fare la stessa cosa è

```
{_x in thisList}count [nomeDellUnita1, nomeDellUnita2, nomeDellUnita3] == 3
```

Es.

```
!(nomeDellUnita in thisList);
```

In questa il trigger si attiverà quando l'unità elencata esce dalla zona del trigger

[Riproduzione di un suono](#)

Esistono due tipologie di suono:

1. Suono "direzionale" → scompare allontanandosi dalla fonte
2. Suono "globale" → si sente sempre in cuffia con la stessa intensità

Il suono va sempre definito preventivamente nel description.ext tramite la **class CfgSounds**

Esistono diverse funzioni con sintassi diversa

- Funzioni del tipo "direzionale": **say3D, playsound3D**
- Funzioni del tipo "globale": **say, playsound**

Richiamare un file audio all'interno della missione.

Metodo 1 - Trigger

1. mettere un trigger
2. selezionare il suono dal menu a tendina. Comparirà quello che avete scritto in "name"
3. il suono è riprodotto al soddisfacimento del trigger

In tal caso il suono va definito nel description.ext nella classe dei suoni del tipo **CfgSFX**



Metodo 2 - Trigger

1. mettere un trigger
2. nel campo **attivazione** mettere la stringa

```
this say3D "sound01"; // nome della classe del suono
```

*** NOTA: VIENE RIPRODOTTO SOLO UNA VOLTA ****

Se si vuole che il suono sia riprodotto in loop mettere questa stringa

```
nul = [thisTrigger] spawn {while {true} do {(_this select 0) say3D  
"sound01"; sleep 150;}};
```

Dove 150 (in questo esempio) è la durata del file audio utilizzato in secondi. Cambiarlo all'occorrenza!

Metodo 3 – In un oggetto

1. mettere un oggetto
2. nel suo campo init mettere la stringa

```
this say3D "sound01"; // nome della classe del suono
```

oppure per la durata in loop

```
nul = [this] spawn {while {true} do {(_this select 0) say3D "sound01"; sleep  
150;}};
```

Nota: **say3D** e **playsound** hanno effetto locale! Tutto bene se messi in oggetti/trigger da editor
Esempio con esecuzione in globale del playsound

```
["Allarme"] remoteExec ["playsound",0,false];
```




62.3 FILES DI LOADOUT

Scopo:

Impostare correttamente i files dei loadout

I loadout sono contenuti in files pre-impostati che sono "letti" dal sistema del Tablet.
I files sono in formato script **.sqf**

File di loadouts da creare:

- ✓ squadleader.sqf
- ✓ teamleader.sqf
- ✓ mitragliere.sqf
- ✓ granatiere.sqf
- ✓ fuciliere.sqf
- ✓ fuciliereAT.sqf
- ✓ marksman.sqf
- ✓ spec_esplosivi.sqf
- ✓ medico.sqf



**SEGUIRE QUESTI NOMI POICHÉ
ALTRIMENTI NON VENGONO LETTI
DAL TABLET!!**

[Procedura operativa per la creazione dell'.sqf PER OGNI RUOLO](#)

Il modo più semplice è usare l'editor e copiare il loadout dalla schermata dell'equipaggiamento arsenal.

1. Mettere giù un'unità blufor dall'editor con le mod istallate
2. Tasto destro sull'unità e selezionare "modifica equipaggiamento"
3. Si apre l'arsenal da dove quindi creare il loadout desiderato (per esempio il fuciliere)
4. Da quest'arsenal fare clic su **ESPORTA**. Questo copierà il loadout negli appunti
5. Creare un file di testo in una sottocartella (es loadout/) nella cartella della missione e nominarlo come "fuciliere.sqf". L'estensione deve essere SQF. Se necessario convertirlo da .txt a .sqf
6. **Incollare** (CTRL + V) il loadout del fuciliere esportato nel file fuciliere.sqf
7. Modificarlo cancellando le istruzioni nella parte iniziale e finale per farlo sembrare così

Es: fuciliere.sqf

```
comment "Remove existing items";
removeAllWeapons this;
removeAllItems this;
removeAllAssignedItems this;
removeUniform this;
removeVest this;
removeBackpack this;
removeHeadgear this;
removeGoggles this;

comment "Add weapons";
this addWeapon "rhs_weap_m4a1";
this addPrimaryWeaponItem "rhsusf_acc_nt4_black";
this addPrimaryWeaponItem "rhsusf_acc_ACOG_RMR";
this addPrimaryWeaponItem "rhs_mag_30Rnd_556x45_M855A1_Stanag";
```



```
comment "Add containers";
this forceAddUniform "rhs_uniform_acu_ucp";
this addVest "rhsusf_spcs_ucp_rifleman";
this addBackpack "rhsusf_assault_eagleaiii_ucp";

comment "Add binoculars";
this addWeapon "ACE_VectorDay";

comment "Add items to containers";
for "_i" from 1 to 2 do {this addItemToUniform "ACE_epinephrine";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_morphine";};
for "_i" from 1 to 20 do {this addItemToUniform "ACE_packingBandage";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_CableTie";};
for "_i" from 1 to 4 do {this addItemToUniform "ACE_tourniquet";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_salineIV_250";};
this addItemToUniform "ACE_EarPlugs";
this addItemToUniform "ACE_MapTools";
for "_i" from 1 to 12 do {this addItemToVest
"rhs_mag_30Rnd_556x45_M855A1_Stanag";};
this addItemToBackpack "ACE_EntrenchingTool";
for "_i" from 1 to 4 do {this addItemToBackpack "rhs_mag_an_m8hc";};
for "_i" from 1 to 4 do {this addItemToBackpack "rhs_mag_m67";};
this addItemToBackpack "SmokeShellGreen";
this addItemToBackpack "SmokeShellPurple";
this addHeadgear "rhsusf_ach_helmet_ESS_ucp";
this addItemToBackpack "ACE_SpraypaintGreen";
this addItemToBackpack "ACE_SpraypaintRed";
this addItemToBackpack "ACE_SpraypaintBlue";
this addItemToBackpack "ACE_SpraypaintBlack";

comment "Add items";
this linkItem "ItemMap";
this linkItem "ItemCompass";
this linkItem "ItemWatch";
this linkItem "TFAR_anprc152";
this linkItem "rhsusf_ANPVS_14";
```

8. Inserire queste due righe nello script

```
this = player; // Va messa all'inizio del file del loadout
this = nil; // Va messa alla fine del file del loadout
```

La prima all'inizio e la seconda alla fine

Quindi

fuciliere.sqf

```
this = player;

comment "Remove existing items";
removeAllWeapons this;
removeAllItems this;
removeAllAssignedItems this;
removeUniform this;
removeVest this;
removeBackpack this;
removeHeadgear this;
removeGoggles this;
```



```
comment "Add weapons";
this addWeapon "rhs_weap_m4a1";
this addPrimaryWeaponItem "rhsusf_acc_nt4_black";
this addPrimaryWeaponItem "rhsusf_acc_ACOG_RMR";
this addPrimaryWeaponItem "rhs_mag_30Rnd_556x45_M855A1_Stanag";

comment "Add containers";
this forceAddUniform "rhs_uniform_acu_ucp";
this addVest "rhsusf_spcs_ucp_rifleman";
this addBackpack "rhsusf_assault_eagleaiii_ucp";

comment "Add binoculars";
this addWeapon "ACE_VectorDay";

comment "Add items to containers";
for "_i" from 1 to 2 do {this addItemToUniform "ACE_epinephrine";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_morphine";};
for "_i" from 1 to 20 do {this addItemToUniform "ACE_packingBandage";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_CableTie";};
for "_i" from 1 to 4 do {this addItemToUniform "ACE_tourniquet";};
for "_i" from 1 to 2 do {this addItemToUniform "ACE_salineIV_250";};
this addItemToUniform "ACE_EarPlugs";
this addItemToUniform "ACE_MapTools";
for "_i" from 1 to 12 do {this addItemToVest
"rhs_mag_30Rnd_556x45_M855A1_Stanag";};
this addItemToBackpack "ACE_EntrenchingTool";
for "_i" from 1 to 4 do {this addItemToBackpack "rhs_mag_an_m8hc";};
for "_i" from 1 to 4 do {this addItemToBackpack "rhs_mag_m67";};
this addItemToBackpack "SmokeShellGreen";
this addItemToBackpack "SmokeShellPurple";
this addHeadgear "rhsusf_ach_helmet_ESS_ucp";
this addItemToBackpack "ACE_SpraypaintGreen";
this addItemToBackpack "ACE_SpraypaintRed";
this addItemToBackpack "ACE_SpraypaintBlue";
this addItemToBackpack "ACE_SpraypaintBlack";

comment "Add items";
this linkItem "ItemMap";
this linkItem "ItemCompass";
this linkItem "ItemWatch";
this linkItem "TFAR_anprc152";
this linkItem "rhsusf_ANPVS_14";

this = nil;
```

L'operazione va ripetuta per ogni ruolo

Il file così creato può essere letto dal Tablet

[Assegnazione specializzazione da medico, specialista esplosivi/ingegnere](#)

Per assegnare queste abilità occorre aggiungere delle stringhe ai loadout dei files:

- ✓ medico.sqf
- ✓ spec_esplosivi.sqf



medico.sqf

```
this = player;

comment "Remove existing items";
removeAllWeapons this;
removeAllItems this;
removeAllAssignedItems this;
removeUniform this;

... ..

comment "Add items";
this linkItem "ItemMap";
this linkItem "ItemCompass";
this linkItem "ItemWatch";
this linkItem "TFAR_anprc152";
this linkItem "rhsusf_ANPVS_14";

this setVariable ["ACE_IsEngineer", 0, true];
this setVariable ["ace_medical_medicClass", 0, true];
sleep 1;
this setVariable ["ace_medical_medicClass", 1, true];

this = nil;
```

spec_esplosivi.sqf

```
this = player;

comment "Remove existing items";
removeAllWeapons this;
removeAllItems this;
removeAllAssignedItems this;
removeUniform this;

... ..

comment "Add items";
this linkItem "ItemMap";
this linkItem "ItemCompass";
this linkItem "ItemWatch";
this linkItem "TFAR_anprc152";
this linkItem "rhsusf_ANPVS_14";

this setVariable ["ace_medical_medicClass", 0, true];
this setVariable ["ACE_IsEngineer", 0, true];
sleep 1;
this setVariable ["ACE_IsEngineer", 1, true];

this = nil;
```

NOTA: allo specialista esplosivi va assegnata l'abilità d'ingegnere!
Le istruzioni d'assegnazione sono le stesse



Cassa con selezione del loadout

Comunque sia i files dei loadout li metteremo nel gaming system

Ad ogni modo, di seguito come creare una cassa con selezione dei loadout tramite rotellina

1. Creare un file *Menu.sqf* nella cartella \loadout.
In questa cartella devono essere salvati i file .sqf di loadout sopra!

Menu.sqf

```
comment "Menu del LOADOUT";

//*****
// Stringa da inserire nell'init dell'oggetto cassa dell'arsenal
// null = [this] execVM "\loadout\menu.sqf";
//*****

// Menu di selezione
_cartello = _this select 0;

_cartello addAction ["<t color='#ff0000' size='1.3'>Selezione LOADOUT</t>",""];
_cartello addAction ["<t color='#FFFF00' size='1.3'>-----</t>",""];
_cartello addAction ["<t color='#00FF00' size='1.3'>Squad Leader SL</t>","\loadout\squadleader.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Team Leader TL</t>","\loadout\teamleader.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Mitragliere MG</t>","\loadout\mitragliere.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Granatiere GL</t>","\loadout\granatiere.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Fuciliere RM</t>","\loadout\fuciliere.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Fuciliere AT</t>","\loadout\fuciliereAT.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Marksman MK</t>","\loadout\marksman.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Specialista esplosivi SE</t>","\loadout\spec_esplosivi.sqf"];
_cartello addAction ["<t color='#00FF00' size='1.3'>Medico MD</t>","\loadout\medico.sqf"];
```

2. Mettere giù da editor una cassa o qualsiasi oggetto si voglia e copiare tale stringa nel suo campo init

```
null = [this] execVM "\loadout\menu.sqf";
```

NOTA: per inserire un ARSENAL VIRTUALE in un oggetto mettere nel suo campo init la stringa

```
this addAction ["<t size='1.3'>ARSENAL VIRTUALE</t>",{["Open",true] call BIS_fnc_arsenal; }];
```



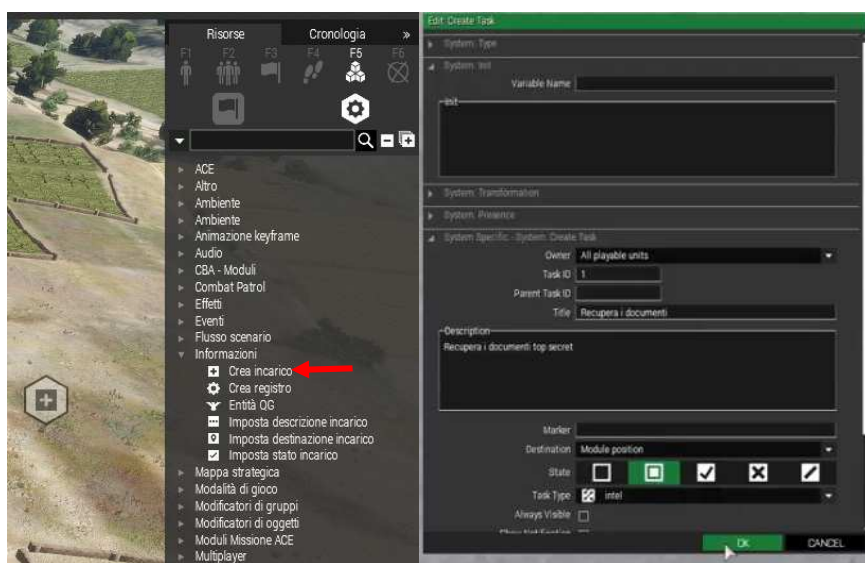
62.4 RECUPERO INFORMAZIONI (INTEL)

Procedura per creare un documento o similare da recuperare contenente informazioni.

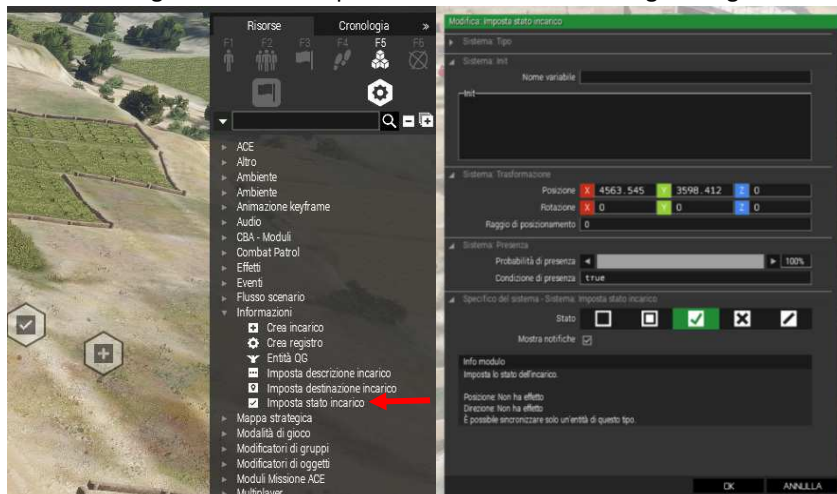
1. Inserire un oggetto da recuperare come per esempio un documento e dargli un nome, es. "nomeintel"



2. Mettere giù il modulo "Crea incarico". Proprietario: tutti i giocatori, task: 1, Stato: assegnato



3. Mettere giù il modulo "imposta stato incarico" ed assegniamogli il valore "compiuto"



4. Mettere giù un trigger e sincronizzare i due moduli sopra con esso.



5. Aprire l'oggetto intel creato ed inserire la stringa nel suo campo init:

```
call{this AddAction ["Prendi Documento", {DeleteVehicle nomeintel}];}
```

6. Aprire il trigger e mettere la stringa seguente nel campo Condizioni

```
call{!alive nomeintel;}
```

Questa controlla se il documento è stato cancellato ed "autorizza" il passaggio dell'incarico a "successo"



7. Volendo un messaggio aggiuntivo una volta presa l'intel si può creare un secondo trigger e mettere nella Condizione:

```
call{!alive nomeintel;}
```

Nel campo Attivazione la stringa:

```
hint "Testo del suggerimento che comparirà in alto a destra"
```

Utilizzare **hintC** per richiedere all'utente di fare clic su "Continua" prima di procedere.

Utilizzare **hintSilent** per disabilitare il segnale acustico.

Si può impostare anche un ritardo di qualche secondo in modo che la scritta appaia qualche secondo dopo il messaggio di obiettivo completato

METODO ALTERNATIVO

Un modo alternativo più veloce per assegnare intel con eventuali messaggi (senza passare per il modulo incarico) è il seguente:

4. Creare l'oggetto e nominarlo, es. "nomeintel"
5. Inserire la stringa nel suo campo init:

```
call{this AddAction ["Prendi Documento", {DeleteVehicle nomeintel}];}
```

6. Creare un trigger e mettere nella Condizione:

```
call{!alive nomeintel;}
```

Nel campo Attivazione inserire la stringa:

```
hint "Testo del suggerimento che comparirà in alto a destra";  
playsound "nomeSuono"; // facoltativo. "nomeSuono" da definire del description.ext
```



62.5 ENEMY SPAWINING SYSTEM (ESS)

Procedura per impostare l'ESS system

1. Copiare tutta la cartella "ESS_system\" nella cartella missione



2. Creare il file "*initServer.sqf*" nella cartella missione e copiarvi dentro questa istruzione:

```
[ ]execVM "ESS_system\ESS_OpenMe.sqf";
```



3. Mettere in mappa da editor uno o più marker area in cui generare lo spawning dei nemici



4. Nominare ogni marker (es: MarkerAreaESS1, MarkerAreaESS2,)



5. Compilare il file "ESS_OpenMe.sqf" per la generazione della singola zona ESS di spawn tramite la seguente stringa

```
null =  
[FACTION, SIDE, "MARKER", [NUM_TEAM_ROAD, MAX_UNITS_PER_TEAM_ROAD], [NUM_TEAM_HOUSE, MAX_UNITS_PER_TEAM_HOUSE], [STATIC], [LIGHT_VEHICLE, COMPORTAMENTO_LGH], [HEAVY_VEHICLE, COMPORTAMENTO_HVY], DISTANCE] execVM "ESS_system\core\ESS_Launch.sqf";
```

N.B. * UNA ISTRUZIONE PER OGNI ESS MARKER AREA MESSA NELL'EDITOR !! *****

Gli argomenti contenuti nella stringa sono i seguenti:

- 0 - **FACTION**: Fazione usata per i nemici - <NUMBER>. Questa è definita nel file "ESS_Units.sqf".
- 1 - **SIDE**: Tipo di fazione da creare - < east / west / civilian / independent >
- 2 - **"MARKER"**: Nome del marker area definito nell'editor in cui creare la zona ESS - <STRING>
- 3 - **NUM_TEAM_ROAD**: Numero di fire teams in strada - <NUMBER>
- 4 - **MAX_UNITS_PER_TEAM_ROAD**: Numero massimo di unità per fire team in strada - <NUMBER>
- 5 - **NUM_TEAM_HOUSE**: Numero di fire teams negli edifici - <NUMBER>
- 6 - **MAX_UNITS_PER_TEAM_HOUSE**: Numero massimo di unità per fire team negli edifici - <NUMBER>
- 7 - **STATIC**: Numero di statiche fisse - <NUMBER>
- 8 - **LIGHT_VEHICLE**: Numero di veicoli leggeri - <NUMBER>



9 - **COMPORAMENTO_LGH**: Definisce se il veicolo leggero deve pattugliare o meno - <NUMBER> (0 o 1): 1 per pattugliamento, 0 è semplicemente spawnato nella posizione

10 - **HEAVY_VEHICLE**: Numero di veicoli pesanti - <NUMBER>

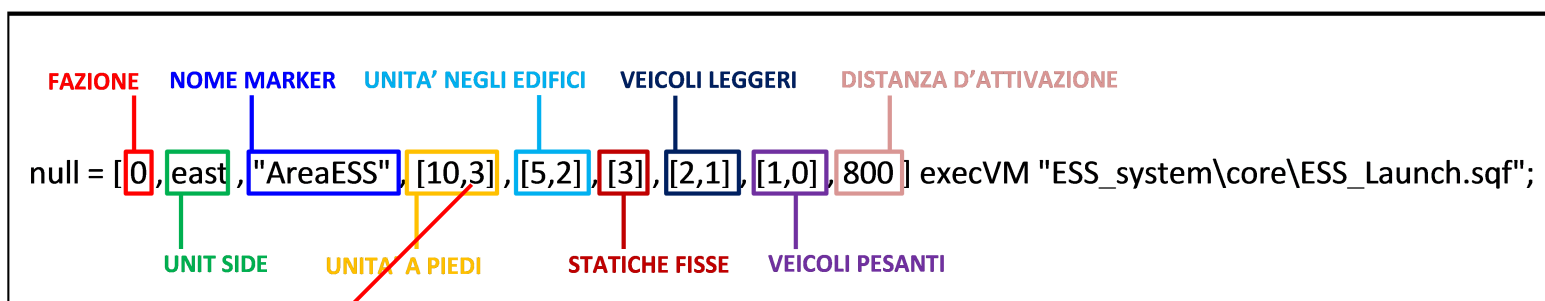
11 - **COMPORAMENTO_HVY**: Definisce se il veicolo pesante deve pattugliare o meno - <NUMBER> (0 o 1): 1 per pattugliamento, 0 è semplicemente spawnato nella posizione

12 - **DISTANCE**: Distanza del player dal contorno del marker all'interno della quale inizia lo spawn delle unità - <NUMBER> (metri)

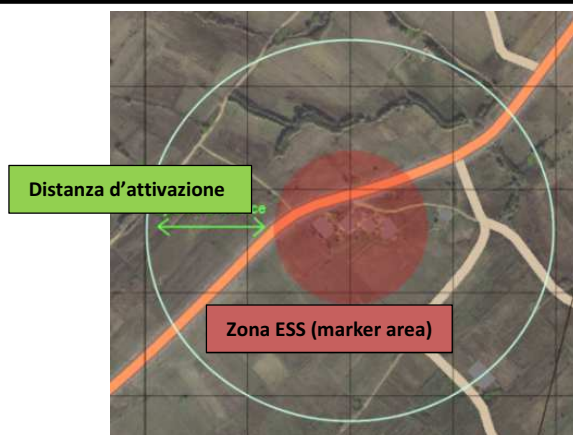


6. Aprire il file "ESS_Units.sqf" per verificare la fazione da usare ed eventualmente inserirvene una specifica (se non inclusa nel file)

Schema generale della chiamata della SINGOLA zona ESS



Il numero di unità sarà random da 1 a 3 !!
Stesso discorso per le unità negli edifici



Esempio

- Creazione di una zona ESS sull'area " AreaESS " definita in editor, impiegando:
 -) La fazione numero 0 del file ESS_Units.sqf
 -) Fazione definita come EAST
 -) 10 fire teams di pattuglia in strada con massimo 3 unità per team
 -) 5 fire teams negli edifici con massimo 2 unità per team
 -) Tre statiche fisse
 -) Due veicoli leggeri in pattuglia
 -) Un veicolo pesante fermo
 -) Lo spawn inizia quando il player è a 800m dal contorno dell'area

```
null = [0, east, "AreaESS", [10,3], [5,2], [3], [2,1], [1,0], 800] execVM "ESS_system\core\ESS_Launch.sqf";
```



FILE "ESS Units.sqf"

```

14
15 private ["_UnitListFac", "_StaticListFac", "_LgtVehListFac", "_HvyVehListFac"];
16 _faction = (_this select 0);
17
18 // ** CSAT Vanilla
19 if ( _faction==0) then {
20   _UnitListFac =
21   ["O SoldierU_SL_F", "O_soldierU_repair_F", "O_soldierU_medic_F", "O_sniper_F", "O Soldi
22   "O Soldier_AAT_F", "O Soldier_AR_F", "O Soldier_AT_F", "O_soldier_exp_F", "O Soldier_F
23   "O_recon_JTAC_F", "O_recon_LAT_F", "O_recon_M_F", "O_recon_medic_F", "O_recon_TL_F"];
24   _StaticListFac = ["O_HMG_01_F", "O_HMG_01_high_F", "O_Mortar_01_F", "O_static_AT_F", "O
25   _LgtVehListFac = ["O_Truck_02_covered_F", "O_Truck_02_transport_F", "O_MRAP_02_F", "O
26   _HvyVehListFac = ["O_APC_Tracked_02_AA_F", "O_APC_Tracked_02_cannon_F", "O_APC_Wheele
27 };
28
29 // ** Islamic State Classnames Project OPFOR
30 if ( _faction==1) then {
31   _UnitListFac =
32   ["LOP_ISTS_OPF_Infantry_Engineer", "LOP_ISTS_OPF_Infantry_Corpsman", "LOP_ISTS_OPF_Ir
33   "LOP_ISTS_OPF_Infantry_Rifleman_6", "LOP_ISTS_OPF_Infantry_Rifleman_4", "LOP_ISTS_OPF
34   "LOP_ISTS_OPF_Infantry_AR_Asst_2", "LOP_ISTS_OPF_Infantry_AR_2", "LOP_ISTS_OPF_Infant
35   "LOP_ISTS_OPF_Infantry_Marksman", "LOP_ISTS_OPF_Infantry_Rifleman_8", "LOP_ISTS_OPF_I
36   _StaticListFac =
37   ["LOP_ISTS_OPF_Static_AT4", "LOP_ISTS_OPF_AGS30_Tripod", "LOP_ISTS_OPF_Static_DSHKM",
38   "LOP_ISTS_OPF_Static_M2_Minipod", "LOP_ISTS_OPF_Static_Mk19_Tripod", "LOP_ISTS_OPF
39   _LgtVehListFac = ["LOP_ISTS_OPF_Landrover_M2", "LOP_ISTS_OPF_M1025_W_M2", "LOP_ISTS_C
40   _HvyVehListFac =
41   ["LOP_ISTS_OPF_T34", "LOP_ISTS_OPF_T55", "LOP_ISTS_OPF_T72BA", "LOP_ISTS_OPF_BMP1", "LO
42   "LOP_ISTS_OPF_M113_W"];
43 };
44
45 // ** Middle Eastern Militia Classnames Project OPFOR
46 if ( _faction==2) then {

```

RIFERIMENTO DELLA FAZIONE

... ..

```

98 // ***** INSERIRE DI SEGUITO ALTRE CLASSNAMES DI FAZIONI PER LA MISSIONE CHE SI STA EDITANDO
99
100 // ** USER DEFINED 1
101 // CLASSE ATTUALMENTE INSERITA - NAPA Independent RHS-GREF
102 if ( _faction==10) then {
103   _UnitListFac =
104   ["rhsgref_nat_rifleman", "rhsgref_nat_machinegunner_mg42", "rhsgref_nat_rifleman_mp44", "rhsgref_nat_commander", "rhsgref_nat_machinegunner",
105   "rhsgref_nat_rifleman_akms"];
106   _StaticListFac = [];
107   _LgtVehListFac = ["rhsgref_nat_uaz", "rhsgref_nat_uaz_dshkm"];
108   _HvyVehListFac = [];
109 };
110
111 // ** USER DEFINED 2
112 // CLASSE ATTUALMENTE INSERITA - FIA Independent Vanilla
113 if ( _faction==11) then {
114   _UnitListFac =
115   ["I_G_Soldier_F", "I_G_Soldier_exp_F", "I_G_Soldier_lite_F", "I_G_Soldier_AR_F", "I_G_engineer_F", "I_G_Soldier_SL_F", "I_G_medic_F", "I_G_Soldier_M_F"];
116   _StaticListFac = ["B_G_Mortar_01_F"];
117   _LgtVehListFac = ["B_G_Offroad_01_F", "B_G_Offroad_01_armed_F", "B_G_Van_01_transport_F", "B_G_Van_01_fuel_F"];
118   _HvyVehListFac = [];
119 };
120
121 // ** USER DEFINED 3
122 // CLASSE ATTUALMENTE INSERITA -
123 if ( _faction==12) then {
124   _UnitListFac = [];
125   _StaticListFac = [];
126   _LgtVehListFac = [];
127   _HvyVehListFac = [];
128 };
129
130 // ** USER DEFINED 4
131 // CLASSE ATTUALMENTE INSERITA -
132 if ( _faction==13) then {
133   _UnitListFac = [];
134   _StaticListFac = [];
135   _LgtVehListFac = [];
136   _HvyVehListFac = [];

```

FAZIONI UTENTE AGGIUNTIVE

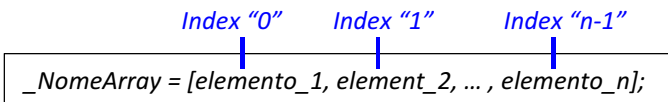
Composizione degli array della fazione

```

118 // ** USER DEFINED 3
119 // CLASSE ATTUALMENTE INSERITA -
120 if ( _faction==12) then {
121   _UnitListFac = [];
122   _StaticListFac = [];
123   _LgtVehListFac = [];
124   _HvyVehListFac = [];
125 };

```

- Classnames delle unità
- Classnames delle statiche
- Classnames dei veicoli leggeri
- Classnames dei veicoli pesanti





Forma compilazione dell'array:

*Definizione di "array": variabile contenente una lista di elementi "indicizzati"

```
_UnitListFac = ["rhsoref_nat_rifleman", "rhsoref_nat_machinegunner_mg42", ....., "rhsoref_nat_rifleman_akms"];
```

Classnames tra doppio apice "" separate da virgole

N.B. L'ultimo elemento va senza virgola finale!!

COME TROVARE LE CLASSNAMES DELLE UNITÀ PER GLI ADDON (MOD)

Il modo più semplice per trovare le Classnames degli elementi della MOD specifica (unità, veicoli, statiche, oggetti...) è usare direttamente l'EDITOR.

Una volta aperto l'editor con la MOD caricata:

1. Mettere giù le unità, i veicoli, etc... di cui si vogliono trovare i nomi delle classi



2. Selezionare tutte le unità, o veicoli, o statiche. Nota: copiarli separatamente!!



3. Tasto destro del Mouse e selezionare nella finestra il sottomenu "Log"



4. Selezionare "copia classe negli appunti"



5. Aprire il Notepad (o l'editore di testo usato) ed incollare.

Si avrà qualcosa del genere (esempio):

```
O_SoldierU_SL_F  
O_soldierU_repair_F  
O_soldierU_medic_F  
O_sniper_F  
O_Soldier_A_F
```

La classname identifica un'unità del gioco, completa del loadout di default e tutte le sue "caratteristiche".
La stessa cosa vale per veicoli, statiche, oggetti, ... etc





6. Modificare aggiungendo il doppio apice, le virgole, mettendoli su un'unica riga

"O_SoldierU_SL_F","O_soldierU_repair_F","O_soldierU_medic_F","O_sniper_F","O_Soldier_A_F"



7. Copia ed incolla negli array del file "ESS_Units.sqf"

```
117  
118 // ** USER DEFINED 3  
119 // CLASSE ATTUALMENTE INSERITA -  
120 if (_faction==12) then {  
121     _UnitListFac = { };  
122     _StaticListFac = [ ];  
123     _LgtVehListFac = [ ];  
124     _HvyVehListFac = [ ];  
125 };  
126
```

8. Stesso discorso per statiche e veicoli (NOTA: devono essere SENZA equipaggio!)



63 ENEMY OCCUPATION SYSTEM (EOS)

Enemy Occupation System (EOS) è uno script altamente personalizzabile e potente che consente ai progettisti di missioni di popolare facilmente i loro scenari con unità AI. Le unità generate nell'ambiente EOS vengono rimosse automaticamente quando non sono necessarie per risparmiare risorse e alla fine consentono missioni multiplayer più grandi senza ritardi e de-sincronizzazioni. Basta posizionare semplici markers che coprono le aree della mappa e EOS farà il resto.

Caratteristiche:

- Genera fanteria difensiva, sommozzatori, barche, veicoli, elicotteri e statiche
- Risparmia risorse del server
- Memorizza in modo affidabile le unità
 - a) Le unità si generano quando il giocatore è vicino
 - b) Le unità uccise in un precedente ingaggio non verranno più generate
 - c) Le unità vengono eliminate quando il giocatore non è vicino
- Funzionalità avanzate
 - a) Imposta le abilità dell'unità
 - b) Imposta il danno personalizzato per le unità
 - c) Contatore uccisioni
 - d) Aggiunta di unità da mods in EOS
 - e) Nascondere/mostrare i marker EOS in missione
 - f) Possibilità di visualizzare le unità

EOS è scaricabile all'indirizzo <http://www.armaholic.com/page.php?id=20262>

Le istruzioni su come usare lo script sono contenute nel file di descrizioni scaricati. Ad ogni modo sono qui di seguito riportate per completezza:

1. Mettere i file scaricati init.sqf e la cartella \eos nella cartella della propria missione
2. Inserire il modulo "Logica di gioco" (Gamelogic) e nominarlo SERVER
3. Inserire un marker (Rettangolo o Ellissi) sopra l'area dove si vuole spawnare le unità. Chiamarla **EOSzone1**.
4. Aprire il file OpenMe.sqf all'interno della cartella 'eos'. Missions\Missionname\eos\OpenMe.sqf
5. Andare fino alla fine e copiare questa stringa nel file OpenMe.sqf

```
null = [{" EOSzone1"}, [2,1], [2,2], [1,3], [1], [2], [1,2], [1,0,250, WEST, FALSE, FALSE]] call EOS_Spawn;
```

Per impostare più zone basta aggiungere marker, **EOSzone2**, **EOSzone3**... Ogni marker diventerà una zona EOS completa. Occorrerà quindi aggiungere più chiamate delle zone nel file OpenMe.sqf, cioè:

```
null = [{" EOSzone2"}, [2,1], [2,2], [1,3], [1], [2], [1,2], [1,0,250, WEST, FALSE, FALSE]] call EOS_Spawn;  
null = [{" EOSzone3"}, [2,1], [2,2], [1,3], [1], [2], [1,2], [1,0,250, WEST, FALSE, FALSE]] call EOS_Spawn;  
.....
```

6. Salvare il file OpenMe.sqf
7. Fare una preview della missione per test

NOTA1: dopo avere fatto qualche test sembra che le unità sono spawnate solo alla presenza dei players.

NOTA2: è possibile usare anche altri nomi per i marker basta richiamarli in maniera coerente nell'OpeMe.sqf.

NOTA3: Per far "vedere" le unità spawnate in mappa da Zeus, occorre usare la striga seguente tra quelle esposte al §11.

Si può mettere sia nell'init, che nel modulo "Logica di gioco" impostato come server, che nel modulo "aggiungi oggetti modificabili"

```
if (isServer) then {  
    [] spawn {  
        while {true} do {  
            {  
                _x addCuratorEditableObjects [allUnits, true];  
                _x addCuratorEditableObjects [vehicles, true];  
                sleep 5;  
            } forEach allCurators;  
        };  
    };  
};
```

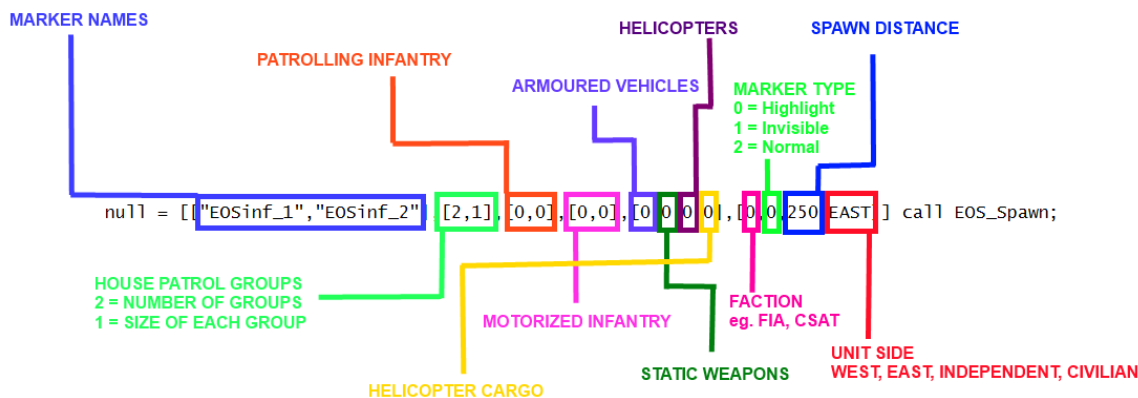
Descrizione della stringa inserita nel OpenMe.sqf. Essa ha la forma generale:



```

null=[["M1","M2","M3"],[HOUSE GROUPS,SIZE OF GROUPS,PROBABILITY],[PATROL GROUPS,SIZE OF
GROUPS,PROBABILITY],[LIGHT VEHICLES,SIZE OF CARGO,PROBABILITY],[ARMoured VEHICLES,PROBABILITY],[STATIC
VEHICLES,PROBABILITY],[HELICOPTERS,SIZE OF HELICOPTER
CARGO,PROBABILITY],[FACTION,MARKERTYPE,DISTANCE,SIDE,HEIGHTLIMIT,DEBUG]] call EOS_Spawn;

```



What is an EOS zone?

After a marker becomes an EOS zone, units will spawn when the player is within the **spawn distance** and units will be deleted when the player is outside the **spawn distance**.

By default an EOS zone will; appear red on the map, glow a bright red when you get within the spawn distance and turn green if the zone is cleared.

GROUP SIZES KEY (definisce il numero di unità presenti in quel gruppo ossia le unità che compongono il fire team)

- 0 = 1
- 1 = 2,4
- 2 = 4,8
- 3 = 8,12
- 4 = 12,16
- 5 = 16,20

Es. [2,1]: crea 2 fire teams (gruppi) ognuno con unità da 2 a 4. Il sistema "deciderà" quante metterne.

NOTE: FOR 100% PROBABILITY OF UNITS SPAWNING SET 3rd PARAMETER TO 100 OR LEAVE EMPTY.

Nell'esempio [2,1] è sicuro a 100% che creerà 2 fire teams. Se fosse [2,1,60] la probabilità che crea i gruppi è del 60%.

House Patrol groups spawn in buildings within the marker. After spawning each group will remain inside buildings.

To spawn 1 house patrol groups - [1,2]

Spawn 3 house patrol groups with 50% probability of spawning - [3,2,50]

To spawn a group with between 2 and 4 units - [1,1]

Spawn three groups with between 16 and 20 units and a 70% probability of spawning - [3,5,70]

Patrolling Infantry spawn anywhere within the marker. After spawning each group will be given waypoints and patrol the marker.

To spawn 1 group of infantry - [1,2]

Spawn 3 patrol groups with 50% probability of spawning - [3,2,50]

To spawn a group with between 2 and 4 units - [1,1]

Spawn three groups with between 16 and 20 units and a 70% probability of spawning - [3,5,70]

Motorised Infantry spawn anywhere within the marker. After spawning each group will be given waypoints and patrol the marker. Motorised infantry are light vehicles and carry units in the free seats.

To spawn 1 motorised infantry - [1,2]

Spawn 3 motorised infantry with 50% probability of spawning - [3,2,50]

Cargo with between 2 and 4 units - [1,1]

Cargo with between 16 and 20 units - [3,5]



Armoured Vehicles spawn within the marker. Similar to Motorised infantry except they do not carry cargo.

Spawn 3 armoured vehicles with 50% probability of spawning - [3,50]

Spawn 3 armoured vehicles with 100% probability of spawning - [3]

Helicopters spawn outside the marker and fly towards the zone. If the **helicopter cargo** is 0 then an attack helicopter will spawn. If the **helicopter cargo** is above 0 then a transport helicopter will carry units into the marker and land.

Spawn 1 transport helicopter with cargo and 75% probability of spawning - [1,3,75]

Spawn 1 Attack helicopter and 15% probability of spawning - [1,0,15]

Faction classes

```
null = [{"EOSinf_1", "EOSinf_2"}, [2,1], [0,0], [0,0], [0], [0], [0,0], [0,0], [0,0,250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

Imposta la classe della fazione che verrà spawnata.

0 = EAST CSAT FACTION

1 = WEST NATO FACTION

2 = INDEPENDENT AAF FACTION

3 = CIVILIAN

4 = WEST FIA FACTION

5,6,7 = Custom classes. Add mods factions ect

Marker setup

```
null = [{"EOSinf_1", "EOSinf_2"}, [2,1], [0,0], [0,0], [0], [0], [0,0], [0,0], [0,0,250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

0 = Default. Markers will show as red and glow red when active and turn green after cleared

1 = Markers will be invisible

2 = Markers will show as red and turn green after cleared

Spawn Distance

```
null = [{"EOSinf_1", "EOSinf_2"}, [2,1], [0,0], [0,0], [0], [0], [0,0], [0,0,250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

When a player is within 250m of the marker all the units will spawn. If the player leaves the 250m radius then the units will be deleted.



Faction classes

```
null = [{"EOSinf_1", "EOSinf_2"}, [2,1], [0,0], [0,0], [0], [0], [0,0], [0,0,250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

Imposta il tipo di unità che verranno spawnata. Creerà le unità selezionandole dalla classe specificata.

0 = EAST CSAT FACTION

1 = WEST NATO FACTION

2 = INDEPENDENT AAF FACTION

3 = CIVILIAN

4 = WEST FIA FACTION



5,6,7 = Custom classes. Add mods factions ect

Unit side

```
null = [{"EOSinf_1", "EOSinf_2"}, [2, 1], [0, 0], [0, 0], [0], [0], [0, 0], [0, 0, 250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

Imposta di quale schieramento "politico" è il tipo di unità che verranno create.

WEST = WEST NATO SIDE

EAST = EAST CSAT SIDE

INDEPENDENT = INDEPENDENT AAF SIDE

CIVILIAN = CIVILIAN SIDE

Per esempio se viene specificato

```
null = [{"EOSinf_1", "EOSinf_2"}, [2, 1], [0, 0], [0, 0], [0], [0], [0, 0], [1, 0, 250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

Verranno create delle unità NATO che però fanno parte dello schieramento CSAT.

Height limit

A height limit will prevent flying units from activating EOS zones. To use this feature (Boolean).

```
null = [{"EOSinf_1", "EOSinf_2"}, [2, 1], [0, 0], [0, 0], [0], [0], [0, 0], [0, 0, 250, EAST, TRUE, FALSE]] call EOS_Spawn;
```

Debug mode

Debug mode will show EOS units on the map and show information about EOS. (Boolean).

```
null = [{"EOSinf_1", "EOSinf_2"}, [2, 1], [0, 0], [0, 0], [0], [0], [0, 0], [0, 0, 250, EAST, false, TRUE]] call EOS_Spawn;
```

Additional functions (Advanced users)

Delete EOS zone

- Manually deactivating a zone will delete all the units active the in the area and units will no longer spawn in the future. The marker will turn invisible but still exist in the mission.
- Place the following code in a trigger or script to manually deactivate a zone.

```
[["BAS_zone_1", "EOSinf_1", "EOSinf_2", "EOSmot_1", "EOSmot_2"]] call EOS_deactivate;
```

NOTE: This function is for advanced users only.

EOS kill counter

EOS kill counter will count the number of EOS units killed. By default the killcounter will simply display a hint showing the amount of units killed every time a unit is killed. You can customise the kill counter with your own code by editing *eos\functions\EOS_KillCounter.sqf*

To enable the kill counter open *openMe.sqf*. Find *EOS_KILLCOUNTER=false* and replace with *EOS_KILLCOUNTER=true*

Damage Amplifier

The damage amplifier will make units spawned by EOS easier (or harder) to kill.

To change the damage amplifier open *openMe.sqf*. Find *EOS_DAMAGE_MULTIPLIER=1*.

By default the damage amp is set to 1. To make units easier to kill increase the number to 2 (This will make units take 2x damage. Alternative set to a decimal to make units harder to kill. For example 0.5 (This will make units take only 50% of the damage).

AI skill

You can customise the [skill levels](#) of each type of unit spawned.

To change the skill pre-sets open *eos\AI_Skill.sqf*

Inside you will find details explaining each skill such as accuracy, spot distance, courage, endurance and aiming speed.

Adding mods

To use units from mods open *eos\unitpools.sqf*. Browse to line 67. Add unit classnames of into the arrays relevant arrays. Factions 5,6,7 are reserved for custom factions.



```
_InfPool = Infantry classnames  
_ArmPool = Armoured vehicle classnames  
_MotPool = Light vehicle classnames  
_ACHPool = Attack helicopter classnames  
_CHPool = Transport helicopter classnames  
_stPool = Static weapon classnames  
_shipPool = Boat classnames  
_diverPool = Diver classnames  
_crewPool = Vehicle crew classnames  
_heliCrew = Helicopter crew classnames
```

```
// ADD CLASSNAMES  
if (_faction==6) then {  
  _InfPool=  [];  
  _ArmPool=  [];  
  _MotPool=  [];  
  _ACHPool=  [];  
  _CHPool=   [];  
  _uavPool=  [];  
  _stPool=   [];  
  _shipPool= [];  
  _diverPool= [];  
  _crewPool= [];  
  _heliCrew= []};
```

Per trovare le classnames delle unità della mod che si sta usando vedi §7

63.1 MODALITÀ BASTION

Bastion places players in a defensive scenario. After activating an BASTION zone, waves of units will attack until they are defeated or the zone is overrun.

- Spawn attacking Infantry, divers, boats, vehicles and helicopters
- Saves server resources
- Set wave delay
- Set Initial pause period
- Set number of waves
- Integrated with EOS for lost zones

Procedura:

1. Mettere i file scaricati description.ext, init.sqf e la cartella \eos nella cartella della propria missione
2. Inserire il modulo "Logica di gioco" (Gamelogic) e nominarlo SERVER
3. Inserire un marker (Rettangolo o Ellissi) sopra l'area da difendere. Chiamarla **BASS_zone_1**.
4. Aprire il file OpenMe.sqf all'interno della cartella 'eos'. Missions\Missionname\eos\OpenMe.sqf
5. Andare fino alla fine e copiare questa stringa nel file OpenMe.sqf

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[10,2,120,FALSE,true]] call Bastion_Spawn;
```

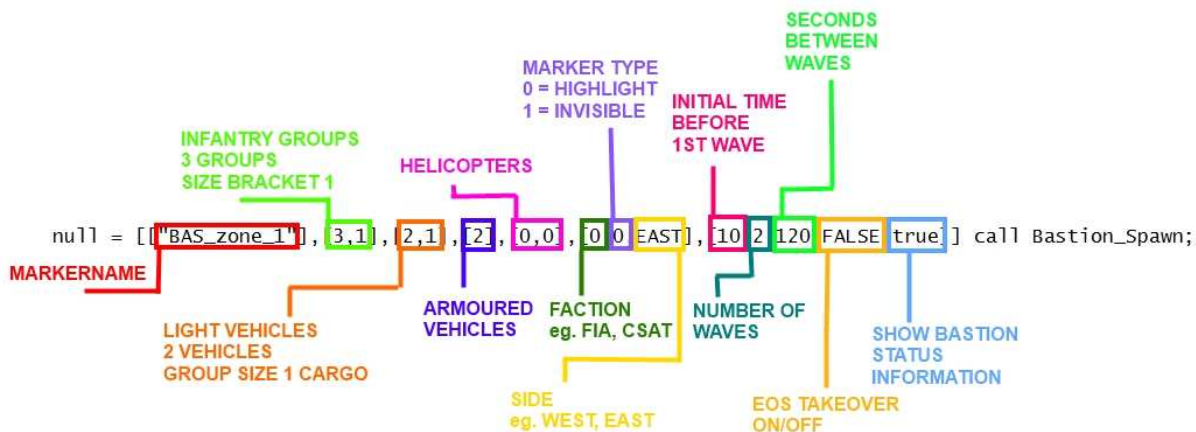
Per impostare più zone basta aggiungere marker, **EOSzone2**, **EOSzone3**... Ogni marker diventerà una zona EOS completa. Occorrerà quindi aggiungere più chiamate delle zone nel file OpenMe.sqf, cioè:

```
null = [{"BAS_zone_2"},[1,1],[3,4],[2],[0,0],[0,0,EAST],[10,2,120,FALSE,true]] call Bastion_Spawn;  
null = [{"BAS_zone_3"},[2,2],[2,1],[1],[1,1],[0,0,EAST],[10,2,120,FALSE,true]] call Bastion_Spawn;  
.....
```

6. Salvare il file OpenMe.sqf
7. Fare una preview della missione per test

Descrizione della stringa inserita nel OpenMe.sqf. Essa ha la forma generale:

```
null=[["M1","M2","M3"],[PATROL GROUPS,SIZE OF GROUPS],[LIGHT VEHICLES,SIZE OF CARGO],[ARMoured  
VEHICLES],[HELICOPTERS,SIZE OF HELICOPTER CARGO],[FACTION,MARKERTYPE,SIDE,HEIGHTLIMIT,DEBUG],[INITIAL  
PAUSE,NUMBER OF WAVES,DELAY BETWEEN WAVES,INTEGRATE EOS,SHOW HINTS]] call Bastion_Spawn;
```



Calling bastion is very similar to calling EOS. Bastion can be called using scripts & triggers.

What is a Bastion zone?

Upon entering a marker that has been called by bastion waves of units will spawn. Units spawned will proceed to attack the marker area until players leave the marker area or all waves are defeated.

By default a BASTION zone will; appear orange on the map, glow bright orange when activated and turn green once completed.

GROUP SIZES KEY

- 0 = 1
- 1 = 2,4
- 2 = 4,8
- 3 = 8,12
- 4 = 12,16
- 5 = 16,20

Assault Infantry spawn 500m away from the edge of the marker. After spawning each group will be given waypoints and attack the marker.

To spawn 1 group of infantry - [1,2]

To spawn 3 groups of infantry - [3,2]

To spawn a group with between 2 and 4 units - [1,1]

To spawn three groups with between 16 and 20 units - [3,5]

Motorised Infantry spawn 700m away from the edge of the marker. After spawning each group will be given waypoints and attack the marker. Motorised infantry are light vehicles and carry units in the free seats.

To spawn 1 motorised infantry - [1,2]

To spawn 3 motorised infantry - [3,2]

Cargo with between 2 and 4 units - [1,1]

Cargo with between 16 and 20 units - [3,5]

Armoured Vehicles spawn 500m away from the edge of the marker. After spawning each vehicle will be given waypoints and attack the marker.

Helicopters spawn outside the marker and fly towards the zone. If the **helicopter cargo** is 0 then an attack helicopter will spawn. If the **helicopter cargo** is above 0 then a transport helicopter will carry units into the marker and land.

Faction classes

```
null = [["BAS_zone_1"],[3,1],[2,1],[2],[0,0],[*0,0,EAST],[10,2,120,FALSE,true]] call Bastion_Spawn;
```

*0 = EAST CSAT FACTION

1 = WEST NATO FACTION

2 = INDEPENDENT AAF FACTION

3 = CIVILIAN



4 = WEST FIA FACTION

5,6,7 = Custom classes. Add mods factions ect

Marker setup

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,*0,EAST],[10,2,120,FALSE,true]] call Bastion_Spawn;
```

*0 = Default. Markers will show as red and glow red when active and turn green after cleared

1 = Markers will be invisible

BASTION specific parameters

Initial pause time

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[*10,2,120,FALSE,true]] call Bastion_Spawn;
```

When the bastion zone is triggered by a player entering the marker zone. EOS will wait *10 seconds before spawning units.

Number of waves

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[10,*2,120,FALSE,true]] call Bastion_Spawn;
```

Every wave EOS will spawn the units defined in the call. Set this number to the number of waves that will attack before the zone is successfully defended.

Wave delay

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[10,2,*120,FALSE,true]] call Bastion_Spawn;
```

EOS will wait *120 seconds before spawning the next wave of units.

Enable EOS integration

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[10,2,20,*FALSE,TRUE]] call Bastion_Spawn;
```

Simulates capturing of areas.

If TRUE the marker will act as a normal EOS zone if the player leaves the bastion zone. If *FALSE then bastion zone will turn red on the map but will not contain any units.

Enable hints

```
null = [{"BAS_zone_1"},[3,1],[2,1],[2],[0,0],[0,0,EAST],[10,2,20,FALSE,TRUE]] call Bastion_Spawn;
```

EOS will display hints with the remaining time before the next wave and alert the player if they have lost the bastion zone.



64 EAGLE WING EFFECTS

it's a very cool post apocalyptic feel with wind blowing ash and dust clouds all around you, closed in darker corners of the screen and a very desolate atmosphere.



description.ext:

```
#include "ca\missions_EW\campaign\description_mission.hpp"
```

init.sqf:

```
execVM "screen.sqf";

"colorCorrections" ppEffectAdjust [2, 30, 0, [0.0, 0.0, 0.0, 0.0], [0.8*2, 0.5*2, 0.0, 0.7], [0.9, 0.9, 0.9, 0.0]];
"colorCorrections" ppEffectCommit 0;
// "colorCorrections" ppEffectAdjust [1, 1, 0, [0.0, 0.0, 0.0, 0.0], [0.8*2, 0.5*2, 0.0, 0.7], [0.9, 0.9, 0.9, 0.0]];
"colorCorrections" ppEffectAdjust [1, 0.8, -0.001, [0.0, 0.0, 0.0, 0.0], [0.8*2, 0.5*2, 0.0, 0.7], [0.9, 0.9, 0.9, 0.0]];
"colorCorrections" ppEffectCommit 3;
"colorCorrections" ppEffectEnable true;
"filmGrain" ppEffectEnable true;
"filmGrain" ppEffectAdjust [0.02, 1, 1, 0.1, 1, false];
"filmGrain" ppEffectCommit 5;

//--- Wind & Dust
[] spawn {
    waituntil {isplayer player};
    setwind [0.201112,0.204166,true];
    while {true} do {
        _ran = ceil random 5;
        playsound format ["wind_%1",_ran];
        _obj = vehicle player;
        _pos = position _obj;

        //--- Dust
        setwind [0.201112*2,0.204166*2,false];
        _velocity = [random 10,random 10,-1];
        _color = [1.0, 0.9, 0.8];
        _alpha = 0.02 + random 0.02;
        _ps = "#particlesource" createVehicleLocal _pos;
        _ps setParticleParams [["Ca\Data\ParticleEffects\Universal\universal.p3d", 16, 12, 8], "", "Billboard", 1, 3, [0, 0, -6], _velocity, 1, 1.275, 1, 0, [9], [_color + [0], _color + [_alpha], _color + [0]], [1000], 1, 0, "", "", _obj];
        _ps setParticleRandom [3, [30, 30, 0], [0, 0, 0], 1, 0, [0, 0, 0, 0.01], 0, 0];
        _ps setParticleCircle [0.1, [0, 0, 0]];
        _ps setDropInterval 0.01;

        sleep (random 1);
        deletevehicle _ps;
        _delay = 10 + random 20;
    }
}
```



```
sleep _delay;

};

//--- Ash
[] spawn {
    waituntil {isplayer player};
    _pos = position player;
    _parray = [
/* 00 */ ["\Ca\Data\ParticleEffects\Universal\Universal", 16, 12, 8,
1], /* 01 */ "",
/* 02 */ "Billboard",
/* 03 */ 1,
/* 04 */ 4,
/* 05 */ [0,0,0],
/* 06 */ [0,0,0],
/* 07 */ 1,
/* 08 */ 0.000001,
/* 09 */ 0,
/* 10 */ 1.4,
/* 11 */ [0.05,0.05],
/* 12 */ [[0.1,0.1,0.1,1]],
/* 13 */ [0,1],
/* 14 */ 0.2,
/* 15 */ 1.2,
/* 16 */ "",
/* 17 */ "",
/* 18 */ vehicle player
];
    _snow = "#particlesource" createVehicleLocal _pos;
    _snow setParticleParams _parray;
    _snow setParticleRandom [0, [10, 10, 7], [0, 0, 0], 0, 0.01, [0, 0, 0, 0.1], 0, 0];
    _snow setParticleCircle [0.0, [0, 0, 0]];
    _snow setDropInterval 0.01;

    _oldPlayer = vehicle player;
    while {true} do {
        waituntil {vehicle player != _oldPlayer};
        _parray set [18,vehicle player];
        _snow setParticleParams _parray;
        _oldPlayer = vehicle player;
    };
};
```

screen.sqf:

```
screen = {

sqf = [true] spawn bis_ew_fnc_screen;

};

while {true} do {
    player spawn screen;
    sleep 5;
};
```



65 ATTRIBUTI DEL TRIGGER

Info			Development	
Name	Category	Description	Property	Type
Variable Name	Init	Unique system name. Can contain only letters, numbers and underscore. The name is not case sensitive, so 'someName' and 'SOMENAME' are treated as the same variables.	name	String
Text	Init	Trigger description. Players can see it in the radio menu when its activation is set to 'Radio'. Also visible in tooltip when hovering over the trigger in the editor.	text	String
Shape	Transformation	Area shape. Available options: <ul style="list-style-type: none"> • Ellipse • Rectangle 	IsRectangle	Bool
Size	Transformation	Area size in meters. [A,B,C]	size3	Array
Position	Transformation	World coordinates in meters. X goes from West to East, Y from South to North and Z is height above terrain.	position	Position3D
Rotation	Transformation	Local rotation in degrees. X is pitch, Y is roll and Z is yaw.	rotation	Number
Type	Activation	Trigger type, determines special behavior upon activation. Available options: <ul style="list-style-type: none"> • None - No other effects except of those defined by 'On Activation' expression. • Guarded by BLUFOR, Guarded by OPFOR, Guarded by Independent - The trigger position becomes a point to be guarded by the given side. Groups with a 'Guard' waypoint will protect all guard points in the scenario, prioritizing them by the distance and order in which they were placed (the first placed is the most important). • Skip Waypoint - Meant to work with a waypoint linked to the trigger using the 'Set Waypoint Activation' connection. Once activated, the trigger will force the waypoint to skip. Particularly useful for 'Hold' or 'Guard' waypoint types, which do not complete automatically. • End #1, End #2, End #3, End #4, End #5, End #6 - Complete the scenario successfully. • Lose - Fail the scenario. 	TriggerType	String
Activation	Activation	What or who can activate the trigger. Some options further depend on 'Activation Condition'. Available options: <ul style="list-style-type: none"> • None - No default activation, only a custom condition expression can activate the trigger. • Any Player - Activated when any player 'Activation Condition'. • Anybody - Activated when any object satisfies the 'Activation Condition'. • BLUFOR, OPFOR, Independent, Civilian, Game Logic - Activated when any object of the given side satisfies the 'Activation Condition'. • Seized by BLUFOR, Seized by OPFOR, Seized by Independent - Activated when the given side is in control of the area. Strength matters, which means 	ActivationBy	String



		Info	Development	
Name	Category	Description	Property	Type
		<p>one tank can be in control of an area while ten enemy infantrymen are still present.</p> <ul style="list-style-type: none"> • Radio Alpha, Radio Bravo, Radio Charlie, Radio Delta, Radio Echo, Radio Foxtrot, Radio Golf, Radio Hotel, Radio India, Radio Juliet - Activated by a player using a radio command (accessible for the player by pressing 0-0 on a standard keyboard). The trigger 'Text' will be used as the command title. 		
Activation	Activation	<p>What or who can activate the trigger. Some options further depend on 'Activation Type'. The available options are specific to the connected trigger owner. Available options:</p> <ul style="list-style-type: none"> • Owner Only - Activated when the connected trigger owner (or its vehicle, if the owner is a crew member) satisfies the 'Activation Condition'. • Whole Group - Activated when all the owner's group members satisfy the 'Activation Condition'. • Group Leader - Activated when leader of the owner's group satisfies the 'Activation Condition'. • Any Group Member - Activated when anyone from the owner's group satisfies the 'Activation Condition'. 	activationByOwner	String
Activation Condition	Activation	<p>Condition of the 'Activation' attribute. Available options:</p> <ul style="list-style-type: none"> • Present - Activated when objects are in the area. • Not Present - Activated when objects are not in the area. • Detected by BLUFOR, Detected by OPFOR, Detected by Independent, Detected by Civilian - Activated when objects are in the area and are discovered by the given side. 	activationType	String
Repeatable	Activation	<p>Repetition rules. When enabled, the trigger can be activated again once deactivated.</p>	repeatable	Bool
Condition	Expression	<p>Repeatedly calculated condition, must return boolean expression. When true, the trigger will be activated. Passed variables are:</p> <ul style="list-style-type: none"> • this - boolean value of activation condition • thisList - list of all objects in trigger area, based on 'Activation' • thisTrigger - trigger object 	condition	String
On Activation	Expression	<p>Expression executed once the trigger is activated. Passed variables are:</p> <ul style="list-style-type: none"> • thisList - list of all objects in trigger area, based on 'Activation' • thisTrigger - trigger object 	onActivation	String
On Deactivation	Expression	<p>Expression executed once the trigger is deactivated. Passed variables are:</p> <ul style="list-style-type: none"> • thisTrigger - trigger object 	onDeactivation	String
Timer Type	Timer	<p>Type of activation timer. Available options:</p> <ul style="list-style-type: none"> • Countdown - Once the conditions are met, the trigger will activate after the specified amount of 	interruptable	Bool



Info			Development	
Name	Category	Description	Property	Type
		time has elapsed. <ul style="list-style-type: none">• Timeout - The trigger's conditions must be satisfied for the specified duration for the trigger to be activated.		
Timer Values	Timer	Timer values in seconds, selected randomly in a range from Min to Max, gravitating towards Mid.	timeout	Array in format [min, mid, max]
Effect Condition	Effects	Condition for effects to be played, must return boolean expression.	effectCondition	String
Sound	Effects	Sound played upon activation.	sound	String
Voice	Effects	Sound spoken by the first unit which activated the trigger.	voice	String
Environment	Effects	Environment sounds played upon activation.	soundEnvironment	String
SFX	Effects	Sound effect played by the trigger upon activation. Repeats as long as the trigger is active.	soundTrigger	String
Music	Effects	Music played upon activation. Replaces previously playing music track.	music	String
UI Overlay	Effects	User interface overlay shown upon activation.	title	String



66 EQUIPAGGIAMENTO STANDARD PER GLI OPERATORI

Di seguito le linee guida per l'equipaggiamento standard per i diversi ruoli.

SQ / TL	GL	FC	FC/AT	MG	MD
<i>UNIFORME</i>					
Compressivi n°20	Compressivi n°20	Compressivi n°20	Compressivi n°20	Compressivi n°20	Compressivi n°20
Lacci n°3	Lacci n°3	Lacci n°3	Lacci n°3	Lacci n°3	Lacci n°3
Morfina n°5	Morfina n°5	Morfina n°5	Morfina n°5	Morfina n°5	Morfina n°5
Adrenalina n°5	Adrenalina n°5	Adrenalina n°5	Adrenalina n°5	Adrenalina n°5	Adrenalina n°5
Salina 250ml n°1-2	Salina 250ml n°1-2	Salina 250ml n°1-2	Salina 250ml n°1-2	Salina 250ml n°1-2	Salina 250ml n°1-2
Sam splint n°5 (*)	Sam splint n°5 (*)	Sam splint n°5 (*)	Sam splint n°5 (*)	Sam splint n°5 (*)	Sam splint n°5 (*)
Strum. cartogr. n°1	Strum. cartogr. n°1	Strum. cartogr. n°1	Strum. cartogr. n°1	Strum. cartogr. n°1	Strum. cartogr. n°1
Fascette n°2	Fascette n°2	Fascette n°2	Fascette n°2	Fascette n°2	Fascette n°2
Tappi auric. n°2	Tappi auric. n°2	Tappi auric. n°2	Tappi auric. n°2	Tappi auric. n°2	Tappi auric. n°2
<i>TATTICO</i>					
Caricatori n°12	Caricatori n°12	Caricatori n°12	Caricatori n°12	Caricatori n°4	Caricatori n°10
Granate HEDP n°4	Granate HEDP n°4				Salina 250ml n°2
Granate HE n°4	Granate HE n°4				Salina 500ml n°6
Fumog white M714 n°2	Fumog white M714 n°2				
<i>ZAINO</i>					
Fumogena bianca n°4	Fumogena bianca n°4	Fumogena bianca n°4	Fumogena bianca n°4	Fumogena bianca n°4	Autoext defribill. n°1
Fumogena verde n°1	Fumogena verde n°1	Fumogena verde n°1	Fumogena verde n°1	Fumogena verde n°1	Adenosina n°10
Fumogena viola n°1	Fumogena viola n°1	Fumogena viola n°1	Fumogena viola n°1	Fumogena viola n°1	Adrenalina n°20
Granata frag M67 n°4	Granata frag M67 n°4	Granata frag M67 n°4	Granata frag M67 n°4	Granata frag M67 n°4	Morfina n°10
Gran. rnd HEP n°5	Gran. rnd HEP n°5	Pala trincea n°1	Pala trincea n°1	Pala trincea n°1	Bend. Elastici n°40
Gran. M397 HET n°5	Gran. M397 HET n°5				Compressivi n°5
Granate HE n°5	Granate HE n°5				Kit chirurgico n°1
GPS Microdagr n°1					Salina 250ml n°4
					Salina 1000ml n°4
					Fumogena bianca n°4
					Fumogena verde n°1
					Fumogena viola n°1
					Granata frag M67 n°4
<i>EXTRA</i>					
Radio AN/PRC-152 n°1	Radio AN/PRC-152 n°1	Radio AN/PRC-152 n°1	Radio AN/PRC-152 n°1	Radio AN/PRC-152 n°1	Radio AN/PRC-152 n°1
Radio onde lun. ILBE n°1	Bussola n°1	Bussola n°1	Bussola n°1	Bussola n°1	Bussola n°1
Bussola n°1	Mappa n°1	Mappa n°1	Mappa n°1	Mappa n°1	Mappa n°1
Mappa n°1					

(*) solo se la partita utilizza l'opzione



SPEC ESPLOSIVI					
<i>UNIFORME</i>					
Compressivi n°20					
Lacci n°3					
Morfina n°5					
Adrenalina n°5					
Salina 250ml n°1-2					
Sam splint n°5 (*)					
Strum. cartogr. n°1					
Fascette n°2					
Tappi auric. n°2					
<i>TATTICO</i>					
Caricatori n°10					
<i>ZAINO</i>					
Fumogena bianca n°4					
Fumogena verde n°1					
Fumogena viola n°1					
Granata frag M67 n°4					
Pala trincea n°1					
Detonatore M57n°1					
Kit EOD n°1					
Rilevatore di mine n°1					
Cariche da demolizione M112 n°3					
<i>EXTRA</i>					
Radio AN/PRC-152 n°1					
Bussola n°1					
Mappa n°1					

(*) solo se la partita utilizza l'opzione

***** FINE DOCUMENTO *****