

SCRIPTS UTILES

addWeaponCargo – B addWeaponCargo ["M16",5];

addMagazineCargo – B addMagazineCargo ["M16",12];

action – A action ["eject",B], A action ["GETINCARGO",heli1],A action ["sitDown",A], A action ["Salute",A]
A action ["heal", B]

assignTeam – A assignTeam "RED"

assignAsCargo – A assignAsCargo B.

***commandGetOut** – commandGetOut B ***NOTA : Hablar por radio.**

***commandTarget**

***commandMove** – A commandMove (position "Marker1")

***commandFire**

_variable = createVehicle ["tipo",posición,[markers],colocación,"especial"]

agent = createAgent [array] para personas,animales,etc.

doTarget – A doTarget B

doFire – A doFire B

doStop – A doStop A

doGetOut – doGetOut A → unit

doMove – A doMove (position "Marker1")

doFollow – A doFollow B

disableAI – A disableAI

fire – A fire [muzzle,mode,magazine], A fire ["throw","smokeshell","smokeshell"]

flyInHeight – A flyInHeight 200

getPos –A getPos B

hint – hint "text"

hintC – hintC "text"

if – if (alive player) and (enemycount= =0) then {"hint "you win"}

join- [s1,s2] join player;

joinSilent

land – A land "LAND" o "GET IN" o "GET OUT"

landAt – A landAt 0

leaveVehicule – A leaveVehicule B

playMove _ myDog playMove "Dog_sit3" u otros: Run, Sprint, Sitting

saveGame – Solamente un trigger con AI Act.= SaveGame.

salute- A action ["Salute",A]

sitdown- A action["sitDown",A]

move – A move pos B

moveInCargo – A moveInCargo B

moveInDriver - A moveInDriver B; idem : moveInCommander, moveInGunner.

this addweapon – this addweapon "M16"

CONTENIDO

1.- Básico de Edición.

Radio

1. Dar órdenes por radio. Pág. 3
2. Pedir refuerzos por radio. Pág. 3
3. Ataque aéreo. Pág. 3

Armamento.

1. Añadir armamento/quitar magazines/Añadir magazines. Pág. 3
2. Cambiar de armamento. Pág. 3
3. Hacer que un soldado dispare bengalas. Pág. 4
4. Crear humo o explosiones en un lugar preciso. Pág. 4

Varios.

1. Hacer que algo se mueva hacia un sitio. Pág. 4
2. Hacer patrullas de centinelas. Pág. 4
3. Activar un trigger selectivamente. Pág. 4
4. Sincronización de waypoint con triggers. Pág. 4
5. Animación de C-130. Pág. 4
6. Hablar con civiles. Pág. 5
7. Tener camuflaje en el rostro. Pág. 5
8. Condición que el final no se produzca antes de tiempo. Pág. 5
9. Copiar-Empastar en el editor. Pág. 5
10. Colocar barcos grandes vacíos en el agua. Pág. 5
11. A.I. entra a vehículo y lo conduce a un lugar determinado (waypoints). Pág. 5
12. Crear vehículo aéreo, asignarle piloto y dirigirlo a un lugar. Pág. 5
13. Lanzarse en paracaídas. Pág. 5

Objetivos.

1. Llegar vivos a un lugar. Pág. 5
2. Perder por muertos. Pág. 6
3. Ganar por matar a todos. Pág. 6

Helicóptero.

1. Aterrizar y cargar o descargar soldados. Pág. 6
2. Inserción en helicóptero con Helipuerto invisible. Pág. 6
3. Comenzar misión en helicóptero. Pág. 6
4. Llamar por radio al helicóptero y subirse a él. Pág. 6
5. Solicitar por radio refuerzos en helicóptero. Pág. 7
6. Entrar soldados A.I. y/o descargarlos automáticamente al aterrizar. Pág. 8

Módulos.

1. Artillería. Pág. 8
2. Vehículo aéreo no tripulado (L.A.V.). Pág. 9
3. Otros apoyos utilizando módulo SOM (sirve también para artillería). Pág. 10
4. High Command. Pág. 10

2.- Edición.

Briefing.

1. Hacer el Briefing y convertirlo en HTML (solo para Armed Assault). Pág. 10
2. Editar en ARMA 2. Pág. 10
3. Señal en pantalla sobre estatus del objetivo (por IMUTEP). Pág. 11
4. Task activado por accidente. Pág. 11
5. Contenido básico de "Briefing.sqs": Pág. 11
6. Crear o borrar tasks. Pág. 12
7. Información que debe contener el Briefing. Pág. 12

Overview.html.

1. Escogencia de la música disponible en ARMA 2. Pág. 12
2. Hacer el "Overview.html". Pág. 13

Intro.sqs/Outro.sqs.

1. "Intro.sqs". Pág. 13
2. "Outro.sqs" Pág. 17
3. Archivos .PBO. Pág. 17

Description.ext

1. Modelo para sonidos. Pág. 17

RADIO

1.- Dar órdenes por radio

Trigger : **Activación** = Radio Alpha (o Bravo, etc.)
 Tipo = Nada
 Texto = (escribir lo que se quiere decir)
 Condición = This
 AI Act = (Script de acción) Vg. : Alpha1 commandFire objetivo

Nota : Para activarlo se debe oprimir la tecla 0 (cero) . Los objetos “vacíos” no son atacados.

2.- Pedir refuerzos por radio.

Trigger 1 : **Activación** = Radio Alpha /Repetidas veces / Presente.
 Condición = This
 AI. Act. = vehi1 doMove (getPos B) → nombre de un objeto colocado o puede ser “player”.

Trigger 2 : **Activación** = Oeste
 Tipo = Nada
 Condición = This
 AI Act. = [S1,S2] joinSilent player

3.- Ataque aéreo.

Colocar la **ruta** y un trigger con **doFire**.
 Condicion = this
 AI Act. = heli1 doFire ene1

Nota : Solamente funciona con ArmA en los siguientes aviones : AV-88, AH60-FAPAL, HV-1, KA-50, MI-17, SU-34, A10.

ARMAMENTO

1.- Añadir armamento / Quitar Magazines / Añadir Magazines.

Existe una manera similar de hacerlo a la usada en (2), pero no se quita todo el armamento. Sirve para cosas pequeñas como marcador Láser, visión nocturna, etc.

Player addWeapon “Laserdesignator” ; o “NVGoggles”.
player removeMagazine “M16” ;
player addMagazine “Laserbatteries” ;

2.- Cambiar de armamento.

- (a) RemoveAllWeapons player ;
- (b) player addmagazine “ “ ; → nombre del magazine según el arma.
- (c) player addWeapon “ “ ;

3.- Hacer que un soldado dispare Bengalas.

- (a) Preparar el arma según se indica en 4, colocando bengalas y cargadores con munición en la inicialización del soldado.

removeAllWeapons A ; A addMagazine "FlareRed_GP25" ; A addMagazine "30Rnd_545x39_AK" ; A addWeapon "AK_74_GL" .

- (b) Colocar un trigger **activado por el Este y DETECTADO por el Oeste** .
(c) **AI Act.** = soldierE fire ["GP25Muzzle","GP25Muzzle","FlareRed_GP25"]

Nota : Ver tablas. **Flares** funcionan \geq 19:00 hrs. (19:15 hrs. OK)

4. Crear Humo o Explosiones en un lugar preciso.

Con la función **CreateVehicle** se puede crear una granada de humo o una explosión activadas mediante un Trigger.

- (a) Crear un objeto pequeño donde se quiere desarrollar el evento y darle nombre (ejemplo O1).
(b) Crear un Trigger con **AI Act.** : **Smoke1= "G_40mm_SmokeRed" createVehicle position O1;**

Para explosiones se reemplaza la Granada de humo por **"SH_125_HE"**. Colocar un auto encima para carro bomba.

(c) " VARIOS

1.- Hacer que algo se mueva hacia un sitio.

A doMove (getPos B)

2.- Hacer patrullas de centinelas.

Hacer mediante "waypoints" el trazado de la patrulla, y el ultimo corresponde a CICLO para que se devuelva.

Velocidad = Limitada

Comportamiento = Seguro

Modo = Avanzar

3.- Activar un Trigger selectivamente.

Sincronizar la unidad escogida con el trigger, seleccionado antes la tecla **F2**.

4.- Sincronización de Waypoint con Triggers.

Es posible lograr que una serie de waypoints se ejecuten en el momento requerido, al sincronizarlos con la activación de un trigger , ya sea por presencia, radio, etc., según se desee.

Importante recordar que la sincronización se hace sobre el waypoint origen de aquel que se quiere ejecutar, es decir en una serie, el inmediatamente anterior al deseado (radio), o cualquiera otro más abajo en la secuencia para otros casos.

5. Animación de C-130

B animate ["ramp_bottom",1]; "ramp_top","door_1", ect.

6.- Hablar con los civiles.

El jugador debe ser un soldado del CDF, o alguien que hable Chernarus.
Se puede tener un interprete como soldado CDF bajo control del jugador (presionar letra T)
O escoger la opción de unidades **Men (FR)**, algunos hablan Chernarus.

7.- Tener camuflaje en el rostro.

Colocar en **Iniciación** de la unidad = **this setFace "face04_camo4";** (6 opciones diferentes)

8. Condición para que el "Final" no se produzca antes de tiempo.

En cada **Trigger** que activa el cumplimiento del objetivo en particular, colocar en la **Activación**

Ai Act.1 = Target1 =True ;

Ai Act.2 = Target2 =True ; etc....

Crear un **Trigger Examinador** :

Ejes = 0 (o valores).

Tipo = Fin1

Condición = Target1 and Target2 ;

9.- Copy-Paste en el Editor.

Ctrl. + C , Ctrl. + V

10.-Colocar barcos grandes vacíos en el agua.

Colocar en **Iniciación** = This setPos [position this select 0, position this select 1,0];

11. A.I. entra a vehiculo y lo conduce a un lugar determinado utilizando waypoints.

Tres (3) waypoints muy cercanos entre si y junto al vehiculo :

MOVE (A moveInDriver B) → ENTRAR → MOVE etc...

12. Crear vehículo aéreo, asignarle piloto, y dirigirlo a un lugar.

v1= createVehicle ["C130J",position o1,[],0,"FLY"]; sc1 moveInDriver v1;sc1 doMove (getpos o2);

13. Lanzarse en paracaídas.

A action ["eject",v1]; A setVelocity [0,0,0];A exec "CA\air2\halo\data\scripts\halo_getout.sqs";

OBJETIVOS

1.- Objetivo llegar vivos a un lugar.

Trigger :

Activación = Oeste

Presente

Tipo = Fin # 1

Condición = This&& alive sc1 and this&& alive sc2 and this&& alive sc3; ***(this&& unidos)**

Efectos = Pista , tipo , efecto , texto

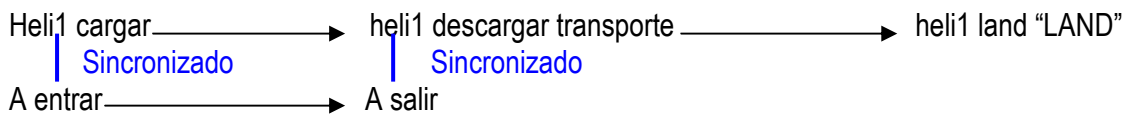
b) Utilizando waypoints.

(1) Con el helicóptero lleno.

Inicializar el helicóptero con = heli1 setFuel 0.
La tripulación abandona el helicóptero.

Trigger 1 : **Activación** = Radio Alpha /unavez/presente.
AI. Act.= heli1 setFuel 1

Colocar **waypoints** tal como se explica :



(2) Con el helicóptero vacío y su piloto.

Colocar un **piloto, waypoint** (1) de **avanzar**, helicóptero **vacío**, un **waypoint** (2) de **entrar** en el helicóptero .
El primer waypoint (1) esta sincronizado con un trigger de llamada de radio.
El segundo waypoint (2) tiene como **AI. Act.** = heli1 doMove (getPos eh1)
El ultimo waypoint es de **AI Act.** = heli1 land "LAND".
Player tiene un waypoint de avanzar **sincronizado** con el primero del helicóptero.

Debe llamarse por radio antes de avanzar al waypoint de player.
Los jugadores van como artilleros y para bajarse debe indicarse a F2 **desembarcar**.

5.- Solicitar por radio refuerzos en helicóptero.

Se puede escoger entre hacer que aterricen y apaguen motores, o descarguen y se dirijan a otro punto.

a) Aterrizar.

- Crear uno o más helicópteros dependientes con uno de ellos (líder) y en tierra.
- Ubicar los soldados en cada uno de ellos con **Iniciación** = **sc1 moveInCargo heli1**.
- Generar un objeto **H** invisible en el lugar deseado de aterrizaje del primer helicóptero y líder.
- Crear Waypoints, el primero **MOVE** en el sitio exacto en que se encuentra el helicóptero para que permanezca con el motor apagado al inicio del juego; el ultimo **TRANSPORT UNLOAD** sobre el Helipuerto virtual (**H**), con **Act.** = **heli1 land "LAND"**.
- Generar dos **Triggers**:
 - 1- **Radio Alpha**. "Llamar helicópteros" o cualquier texto. Sincronizarlo con el primer Waypoint.
 - 2- **Act.** = **heli2 land "LAND"**. Activado por BLUFOR.
Mejor ligeramente desviado del primero para lograr buen espacio entre los helicópteros.

b) Sin aterrizar.

- Crear uno o más helicópteros independientes entre si , y en tierra.
- Ubicar los soldados.
- Crear como mínimo tres Waypoints para cada helicóptero, el primero **MOVE** igual al anterior, y con unos seis segundos de retraso entre helicópteros para evitar el amontonamiento, el segundo **TRANSPORT UNLOAD** en el sitio deseado para desembarcar los refuerzos, y el tercero **MOVE** con **Act. = heli1 land "LAND"** hacia donde se deben dirigir y aterrizar.
- Trigger de **Radio Alpha**, sincronizado con el primer Waypoint.
- Sí se coloca en **Act.** del ultimo waypoint el script de **join** o **joinsilent**, los refuerzos automáticamente entran a ser controlados por el jugador, una vez desembarquen.

6. – Entrar soldados AI y/o Descargarlos automáticamente al aterrizar.

a) Entrar :

Utilizar un Trigger, Waypoint , o en la inicialización de la unidad según convenga. **A assignAsCargo B; [A] orderGetIn true;** Sí los ocupantes de un vehiculo deben bajarse y dirigirse al nuevo vehiculo para ingresar, el Trigger debe activarse antes de que el primer vehiculo llegue al waypoint de descarga (¿).

b) Salir :

Colocar Un Trigger en la zona de descarga con **AI Act. = A leaveVehicle B;**

MODULOS

1.- Artillería.

(1) Utilizando SECOP Manager – (SOM)

- 1 - Crear jugador y nombrarlo **Player**.
- 2 – Abrir la función de Módulos y seleccionar **Director de Operaciones Secundarias (SOM)**.
- 3 – Para eliminar las misiones secundarias automáticas, (IMUTEPE), colocar en la **INIC** en el módulo **SOM** :
this setVariable ["settings", [[],true, nil, nil, false]];
Sincronizarlo con **Player**.
- 4 – Colocar un Trigger cerca al jugador con :
Activación = Lógica de juego
Condición = this.
(*) **AI Act.** = [["artillery_barrage"], player, [[RIPPER,[1,7,9]]]] call BIS_SOM_addSupportRequestFunc ;
- 5 – Colocar otro Trigger cercano :
Eje a = 0
Eje b = 0
Activación = Radio Alpha.
Texto = + 1 Artillery Barrage SecOp (o cualquier otro).
(*) **AI Act.** = [["artillery_barrage"], player, [[RIPPER,[1,7,9]]]] call BIS_SOM_addSupportRequestFunc ;
- 6 – Crear las **unidades de artillería**, teniendo en cuenta que para apoyo cercano no se pueden utilizar cañones, solamente morteros, tales como el M 252 (81mm.) o el Podnos 2B14 (82 mm.). Cuidar la orientación para que estén enfilados al posible blanco.
- 7 – Abrir la función de módulos y seleccionar **Artillería**.
- 8 – Nombrarlo **RIPPER**, y **Sincronizarlo** con la pieza líder de **artillería**.

(*) **NOTA** : Los números corresponden a tres (3) de las diversas posibilidades que existen de apoyo :

1. **Supresión inmediata, HE** - Dispara 10 salvas tan rápido como la batería puede hacerlo.
2. **Humo inmediato**.- Dispara 6 granadas de humo.
3. **Supresión inmediata, Willie Pete**.- Ídem a 1 pero con fósforo blanco.
4. **Misión de iluminación**.- Una bengala cada 10 segundos por 3 minutos.
5. **Láser** – Dispara 2 granadas guiadas por láser.
6. **SADARM** - Tres SADARM diez segundos aparte. (Sense And Destroy ARMor 155 mm.)
7. **HE Fuego de Barrera**.- Bombardeo continuo por un minuto.
8. **WP Fuego de Barrera**. – Bombardeo continuo por un minuto con fósforo blanco.
9. **Ajustar el fuego** – Dispara dos granadas HE al blanco.

(2) Disparar Mortero.

- 1 – Crear pieza de artillería (**mortero**).
- 2 – Crear un Trigger que abarque la pieza de artillería :
Activación = Lógica de juego.
- 3 – Abrir módulo y escoger **Artillería (ARTY)**.
- 4 – Sincronizar el **mortero** con **ARTY**.
- 5 – Se opera escogiendo las acciones que aparecen al mover la rueda del **Mouse**.

(3) Artillería virtual (modo simple).

- 1 – Crear **Player**.
- 2 – Crear Módulo **SOM**. No es necesario nombrarlo.
- 3 – Crear Trigger con :
Ejes a/b = 0
Activación = Radio Alpha.
Texto = Solicitar apoyo de H.Q. (o cualquier otro)
Condición = this
AI Act. = [“artillery_barrage”, player, []] call BIS_SOM_addSupportRequestFunc ;
- 4 – Sincronizar **Player** con el Módulo **SOM**.
- 5 – Crear Módulo **ARTY** en donde se quiere la batería virtual (puede omitirse).

2.- Vehículo aéreo no tripulado L.A.V.

Para crear un vehículo no tripulado **MQ-9** :

- 1 – Crear **Player**.
- 2 – Crear en AIR un **L.A.V. volando** , y unlocked.
- 3 – Crear OBJETO :
Side = Empty
Class = Objects (fortifications)
Unit = Bunker (tower).
- 4 – Se puede colocar para ayudar en el realismo una Terminal de UAV (no opera).
- 5 – Crear **Módulo UAV**.
- 6 – Sincronizar : **Player** con **MQ-9**, y con **Bunker** ; **Bunker** con **Módulo UAV** : **MQ-9** con **Módulo UAV**.
- 7 – Acercarse al **Bunker** y operar rueda de Mouse para que aparezca la opción de UAV. Aceptar.
- 8 – Controlar con el Mouse.

3.- Otros apoyos utilizando módulo SOM.(sirve también para artillería)

Reemplazando una pequeña parte del texto del Trigger usado en (20/3) para artillería virtual, y sin el módulo ARTY, pero conservando el módulo SOM, es posible solicitar fácilmente apoyo CAS, SUPPLY- DROP, y TRANSPORTE (en varios tipos de vehículos aéreos).

El script de AI Act. cambia ligeramente :

```
AI Act. = [ [ "tactical_airstrike"], player] call BIS_SOM_addSupportRequestFunc ;  
    "supply_drop"  
    "transport"  
    "artillery_barrage"
```

NOTA: No olvidar que se deben sincronizar el módulo SOM y el jugador.

En todos los casos se señala en el mapa cuando es solicitado.

*** NOTA MUY IMPORTANTE :** Los compañeros deben entrar primero al transporte, para que no se pierda la función de coordenadas.

Para variar de transporte aéreo se añade en el Trigger el siguiente script :

```
SOM setVariable ["TSS_vehicle_custom" OSPREY];
```

4.- High Command

Colocar el modulo y sincronizarlo con el comandante y los grupos subordinados.

Utilizar **Ctrl. + Space** para abrir la interfase y dar instrucciones.

1) EDICION : Briefing

1.- Hacer el Briefing y convertirlo en HTML (solamente ARMED ASSAULT)

Se guarda como "briefing.html" (ojo, entre comillas) el texto previamente creado con bloq de notas.

Tipo = Todos los archivos. **Código** = ANSI.

Se puede COPIAR un texto Vg. Frankeke o en Word y modificarlo en Bloq de notas.

Se coloca en el fólder de la Misión.

2.- Editar en ARMA 2.

- Abrir el EDITOR.
- Escoger el lugar : Chernarus o la isla de Utes.
- Colocar unidades, waypoints, triggers, sincronizar y marcadores. ¡No olvidar poner el nombre a todo!
- Crear :

init.sqf (código= [] execVM "briefing.sqf").Para hacerlo se utiliza el BLOC DE NOTAS , y se guarda como "init.sqf"/**Todos los archivos** / UTF-8

briefing.html (en realidad se trata del **Debriefing**; se puede utilizar el ejemplo de Armaholic , y editarlo en un texto en Office .doc, y luego abrir el BLOC DE NOTAS que esta en Herramientas del sistema, copiar el texto y guardarlo como se indica en el numero 1)

Briefing.sqf (código = ver ejemplos varios en web o **Nº19** ; contiene las notas, tasks etc..).Se guarda como documento.sqf.

- Colocar todos los archivos anteriores manualmente en la carpeta **ARMA 2 otros perfiles (DOCUMENTOS)**, subcarpeta **MISSIONS**, junto al archivo **missions.sqm** de la misión correspondiente.
- Abrir el **briefing** tecleando SHIFT + VISTA PREVIA.

3.- Señal en pantalla sobre status del objetivo.(por IMUTEP)

Se pueden escoger varias opciones : "SUCCEEDED", "FAILED", "CREATED", "CANCELED".

Para hacerlo se utiliza un Trigger con la información del **estado del Task**, y un **Script** para indicar que debe aparecer en pantalla si lo anterior se da :

```
Trigger      tsk1 setTaskState "SUCCEEDED" ; nul = [ objNull,objNull,tsk1,"SUCCEEDED" ] execVM
             "CA\Modules\MP\data\scriptCommands\taskHint.sqf";
```

Según se escoja ,se cambian ambos (ejemplo: "SUCCEEDED" por "FAILED").

(*)**Notas** = " \ " se genera como carácter ASCII, utilizando la combinación **NUM PAD + ALT + 92**.

Soltar **ALT** para que aparezca una vez teclada la combinación en el teclado del Pad. **No funciona en el teclado principal!!** La palabra "**scriptCommands**" es una sola palabra, no debe separarse!!

Aparentemente no funciona con ejes a , b = 0.

4.- Task activado por accidente.

Es posible crear objetivos utilizando probabilidades de aparición. Para esto se crea en el ejemplo un vehiculo **tank1** con una condición de presencia de vgr. 60% ; entonces se añade un **task** para destruirlo , ya sea en el **Briefing** o con un **trigger** (?) :

```
if (alive tank1) then {
tskShilka = player createSimpleTask["Destruir Shilka"];
tskShilka setSimpleTaskDescription["Encuentre y destruya el Shilka que se encuentra en algún lugar cerca del bosque.", "Destruir Shilka", "Destruir Shilka"];
};
```

5.- Contenido básico de "briefing.sqf".

Para que aparezcan las **notas** básicas, así como los diferentes **objetivos** de la misión, puede utilizarse la siguiente plantilla :

```
player createDiaryRecord ["Diary", ["Informe", "Tenemos un objetivo <marker name='obj1'>aquí </marker> y otro<marker name='obj2'> allá </marker>"];
player createDiaryRecord ["Diary", ["Situacion", " Estamos ensayando el briefing<br/> Estamos en la isla de Utes. Debemos avanzar <br/> Avanza hasta la entrada del hangar."]];
player createDiaryRecord ["Diary", ["Mision", " El nombre de la mision es La reunion"]];
player createDiaryRecord ["Diary", ["Ejecucion", " Caminar según instrucciones"]];
player createDiaryRecord ["Diary", ["Apoyo", " Ninguno.Estamos solos"]];
```

```
tsk1 = player createSimpleTask ["Llegar al frente del hangar"];
tsk1 setSimpleTaskDescription ["El lugar esta muy cerca,esta en frente del hangar", "Llegar al frente del hangar", "Punto de reunion"];
tsk1 setSimpleTaskDestination (getMarkerPos "obj1");
player setCurrentTask tsk1;
```

```
tsk2 = player createSimpleTask ["Llegar al punto de reunion"];
tsk2 setSimpleTaskDescription ["Desviarse a la derecha hasta la intersección de caminos", "Llegar al punto de reunion", " Punto de reunion"];
tsk2 setSimpleTaskDestination (getMarkerPos "obj2");
```

```
tsk3 = player createSimpleTask ["Salir del área"];  
tsk3 setSimpleTaskDescription ["Subir por el camino", "Salir del área", "Punto de reunion"];  
tsk3 setSimpleTaskDestination (getMarkerPos "obj3");
```

6.- Crear o borrar task's.

También se pueden crear mediante un **trigger** con el **script** de briefig.sqf para cada nuevo task. Igualmente, pueden borrarse de la misma manera, pero utilizando :

AI Act.= Player (o cualquier unidad) **removeSimpleTask** tsk4.

7.- Información que debe contener el Briefing.

SITUACION

- Premisa de la misión. Porque está la unidad en ese lugar, y que está pasando alrededor.
- Que clase de fuerzas tiene su unidad.
- Que clase de fuerzas lo soportan.
- Que clase de fuerzas y soporte tiene el enemigo.
- Que se espera que haga el enemigo.

MISION

- Que debe hacer su unidad. Quien más está involucrado en la misión.
- Cuando y donde toma lugar la misión. Cual es el tiempo permitido.
- Porque ha sido dada a su unidad.
- Cual es el final deseado. Cuales son los objetivos.

EJECUCION

- Intención del comandante.
- ¿Cómo será conducida?.Esquema de maniobra, objetivos etc.
- Como la unidad llegará al estado final.

ADMINISTRACION / LOGISTICA.

- ¿Existe reaprovisionamiento de armamento?.
- ¿Existen medios de evacuación como helicópteros o ambulancias?.
- ¿Existe soporte de fuego?, tal como artillería,naval,etc.
- Disponibilidad de soporte aéreo.

COMANDOS Y SEÑALES

- Reglas especiales o consideraciones que se deben hacer de comunicación sí no son SOP.: humo, bengalas, colores y significado.

2) EDICION : Overview.html.

1.- Escogencia de la música disponible en ARMA -2.

New Wave

*Bosque muerto – Sombras indefinidas – Espacio de combate abandonado – Viento helado – *Cobalt – Fallout - Wasteland – Sky net – MAD.

Sinfónica

Desierto – *Resurgimiento de los caídos – Últimos supervivientes – Asalto en gran escala – El movimiento *Canción del merodeador.

Rock

The Keh-Shan Riff – Cuchillos afilados – Valentin.

Jazz

Parte de la misión.- Debriefings.

Folklórica

Verano indio.

Otros

Logistics.- Defcon Three.- Refuerzos. – Reforger.

2.- Hacer el "Overview.html".

Utiliza el formato de Imutep modificado en las dimensiones del cuadro ("200" x "121"). Partir de una foto en relación 1:1.163 aproximadamente, y con Photoshop redimensionarla a 512 x 512 pixeles. Se deforma, pero el juego la normaliza luego. También puede usarse el formato.paa. con Photoshop CS2 (Kegety's). Se guarda como documento **html** en la carpeta de la misión, junto con la fotografía en formato .jpg /.paa

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<meta name="GENERATOR" content="VB">
<title>Overview</title>
</head>
<body bgcolor="#FFFFFF">
<p align="center"></p>
<p>
Einer unserer Nachschub Konvois wurde von Guerillas in Myshkino angegriffen. Wir haben seither jeden Funkkontakt zu der Einheit verloren. Steckt der Warlord Tristian dahinter?
<br>
</p>
</body>
</html>
```

Superimportante! : El archivo que contiene la foto y se coloca en el folder de la misión, debe tener como nombre en este caso **saw** solamente, sin la partícula .jpg! El **Overview** **si debe** indicar el tipo de archivo.

Igual que el Briefing.sqf, para que aparezca, debe existir el Briefing.html en la carpeta de la misión!

3) EDICION : Intro.sqs./ Outro.sqs.

1.- "Intro.sqs"

A) - Básico.

Se puede comenzar antes de hacer la misión con el intro (no probado), o una vez realizada hacerlo sobre la misma. El comando **Alt +Tab** permite salir del juego hasta la pantalla de Windows.

Con **AI Act = this exec "camera.sqs"**; colocada en objeto o jugador, se activa el modo de CAMARA.

También con trigger accionado por radio con **AI Act.= player exec "camera.sqs"**;

Ensayar la posición adecuada , y una vez escogida, guardarla en RAM con el clip izquierdo del Mouse.

Ir a pantalla Windows y en el Notepad colocar las coordenadas anteriores usando **Ctrl. + V.**

Completar el script en el Notepad, teniendo en cuenta que el carácter ASCII para ~ es **Alt + Numpad 126.**

Muy importante utilizar comillas rectas en el script!

Ejemplo de script:

```
Playmusic "Ambient05_Cobalt";
titlecut ["" ,"black in",2]
_camera= "camera" camcreate [0,0,0]
_camera cameraEffect ["internal","back"]
;=== 8:40:11
_camera camPrepareTarget [-53135.14,-70677.89,-32496.00]
_camera camPreparePos [6826.91,2452.50,2.00]
_camera camPrepareFOV 1.380
_camera camCommitPrepared 0
@camCommitted _camera
~1
cutText ["Chedakys rule with brutality!","Plain Down"]
~4
;=== 8:49:44
_camera camPrepareTarget [-42594.18,-79761.73,-28303.90]
_camera camPreparePos [6844.86,2422.53,1.79]
_camera camPrepareFOV 0.956
_camera camCommitPrepared 7
@camCommitted _camera
~1
cutText ["And this hated officer is their inspiration!!!","Plain Down"]
~3
cutText ["" ,"Plain down"]
cutText ["Our final objective!!","Plain Down"]
~5
cutText ["" ,"Plain down"]
~3
cuttext ["" ,"black out",2]
~2
cuttext ["" ,"black in",2]
player cameraEffect ["terminate","back"]
camdestroy_camera
playmusic"";
0 fademusic 1;
~2
exit
```

Generado con Ctrl. + v

Tiempo de "paneo".

Guardar la información de NotePad como archivo **"intro.sqs" /todos los archivos/ANSI.**

Colocar en la carpeta de la misión. En la misión colocar en **player AI Act = this exec "intro.sqs";**, y borrar los scripts que se hicieron : **this exec "camera.sqs";**

B)- Cámara en movimiento.

Se designa un objeto con la tecla **F**. Sí puede enfocarlo, aparece un recuadro de color rojo. Es muy importante que el objeto tenga nombre para que permanezca fijada en el mismo. La cámara seguirá al objeto en movimiento. Pueden reemplazarse directamente las coordenadas de `_camera camPrepareTarget` por el nombre escogido, y se obtiene el mismo efecto.

Al utilizar esta función (**F**), los desplazamientos de la cámara cambian, y permiten dar vueltas alrededor del objeto.

Sí se toman las coordenadas de otro punto diferente al primero mediante el clic izquierdo del Mouse y se guardan en el script de Intro, Outro o la escena, pueden lograrse efectos de "paneo" interesantes.

Al agregar valores a la línea : **_camera CamCommittedPrepared 0** (0 segundos en este caso) del último punto, se establece la velocidad del mismo.

Conviene separar los puntos por espacio de mínimo ~3 segundos y colocar un título.

C- Títulos y mensajes.

Mediante el comando **TitleText** pueden insertarse textos en el centro o en la parte de debajo de la imagen. Para centrarlo se utiliza **Plain**, y **Plain Down** para bajarlo. Por default dura unos diez segundos, pero puede eliminarse antes de tiempo con el comando **Title Text** [" ", "Plain Down"] o **Plain** según el caso. Con **TitleFadeOut** se borra progresivamente.

Ejemplo : **TitleText** ["4:45 horas", "Plain Down"]
~5
TitleText [" ", "Plain Down"]; **TitleFadeOut** 4

Con (\ n) se realiza el interlineado si se quiere.

Los textos creados con **TitleText** no se borran al acabarse la escena, pero al utilizar en su lugar **CutText** , lo hacen automáticamente.

D) – Musica.

Pueden insertarse sonidos musicales mediante el comando **playMusic** . Es necesario seguir la denominación abajo relacionada de las distintas pistas musicales del juego para que funcione efectivamente.

Ejemplo : **playMusic** "Track03_First_to_Fight";

Con el comando **FadeMusic** se logra disminuir su volumen progresivamente. Para evitar que aplique durante toda la misión, es necesario revertir el valor posteriormente.

Ejemplo : **playMusic** "Ambient05_Cobalt" (Al inicio)....
camDestroy_camera (al final)
playMusic " " ;
0 fadeMusic 1 ; **0** = segundos ; **1** Volumen Máximo , **0** mínimo.
Exit

12 FadeMusic 0 : En **12** segundos disminuir el volumen a **0**.

Este tipo de comando se utiliza para lograr efectos agradables durante el Intro, Outro ect., y debe colocarse de manera tal que los tiempos coincidan con las acciones que se desarrollan en seguida, para que no se corte bruscamente o quede muy corto ,con un gran espacio de silencio.

Tabla :

- Ambient01_Cold_Wind
- Ambient02_Vague_Shapes
- Ambient03_Indian_Summer
- Ambient04_Electronic_Warfare
- Ambient05_Cobalt
- Ambient06_Khe_Sanh_Riff
- Ambient07_Manhattan
- Ambient08_Reforger
- Short01_Defcon_Three
- Track01_Dead_Forest
- Track02_Insertion
- Track03_First_to_Fight
- Track04_Reinforcements
- Track05_Warpath
- Track06_Abandoned_Battlespace
- Track07_Last_Men_Standing
- Track08_Harvest_Red
- Track09_Movement_To_Contact
- Track10_Logistics
- Track11_Large_Scale_Assault
- Track12_The_Movement
- Track13_Sharping_Knives
- Track14_Close_Quarter_Combat
- Track15_Morning_Sortie
- Track16_Valentine
- Track17_Marauder_Song
- Track18_Ghost_Waltz
- Track19_Debriefing
- Track20_Badlands
- Track21_Rise_Of_The_Fallen
- Track22_Chernarussian_Anthem
- Track26_Organ_Works
- Track27_Eastern_Eggs_-_Killing_Machines

E)- Seguir helicóptero o vehículo.

Utilizar una posición inicial de la cámara, preferiblemente con el jugador (pivota).

Modificar el script de la posición inicial, generado con el clic izquierdo del Mouse, cambiando las coordenadas de **_camera camPrepareTarget**, reemplazándolas por el nombre del avión o vehículo que se quiere seguir..

Igualmente puede modificarse la altura de la cámara en el tercer elemento (**Z**) del array [x,y,z] del segundo script **_camera camPreparePos**.

F)- Colocar visión nocturna.

En el correspondiente script de "intro.sqs", añadir las líneas **CamUseNVG true** (después de la iniciación del efecto de cámara, pero antes de **_camera camPrepareTarget**) , y **CamUseNVG false** ,antes de **Exit**.

2.-"Outro.sqs"

Script igual a "intro.sqs", colocando solamente **TrueEnd = true** al final, antes de **Exit**.

Es necesario crear dos (2) Triggers:

- Nº1 : **Activación** = Blufor, Presente .
AI Act = [] exec "outro.sqs";
- Nº2 : **Activación** = Ninguna, Fin # 1 .
Condición = TrueEnd.
AI Act. = ForceEnd.

***NOTAS:** Se pueden escoger otras activaciones.
Muy importante [] !!

Se puede crear un trigger dentro de otro, o poner el segundo (Nº2) con ejes a,b igual a 0.

En el trigger Nº2 ,escogiendo el final (#1,#2, Perder, etc.) se define si se activa para **GANAR** o **PERDER**. La escogencia debe corresponder exactamente a las terminaciones definidas en **Briefing.html**.

Sí se quiere permitir la aparición en pantalla del estado del objetivo (ver **Nº 17**), se hacen coincidir los trigger de "outro.sqs" e **Imutep**, dando un tiempo de activación al trigger Nº1 de los segundos que se quieran (por ejemplo 5 segundos), para que "outro.sqs" no enmascare la información del estado del objetivo.

3.-Archivos .PBO

Los archivos de las misiones guardadas en el editor como Misiones de un solo jugador , se encuentran en el disco duro en la ruta : **Bohemia Interactive/ Arma 2/ Missions**.

Para ver estos archivos .PBO, es necesario utilizar una herramienta como **EliteNess v2.32**.

Para usarla, hacer clic en Elitness. Automáticamente se abre la interfaz y se puede seleccionar la ruta deseada. La herramienta permite borrarlos selectivamente además de otras cosas.

Para leer los archivos directamente en Windows 7, basta con abrir la carpeta y seleccionar en la barra de tareas **-Archivos de Compatibilidad-**. Sí esta opción aparentemente no aparece, buscar ">>".

4.-"Description.ext"

En este archivo se consignan los elementos adicionales al contenido normal del editor y que serán utilizados en la misión. Los errores cometidos acá son de consecuencias graves, por eso se debe ser muy cuidadoso!

Muy importante utilizar comillas " " rectas para que el script sea reconocido, así como incluir todos los archivos descritos, en la declaración inicial. Si aparece "sound xyz not found", verificar que Arma2 haya cargado bien , sí no hay defecto de script!

Los archivos de sonido pueden crearse con la herramienta Audacity, y guardarse como archivos Ogg Vorbis, para que el juego los reconozca.

Una vez creado en Notepad, se guarda como "description.exe"/ **todos los archivos/ ANSI.**, y se coloca en el archivo de la misión.

Para utilizarlos, se hace uso de triggers, intros (playSound) etc. de la manera acostumbrada, con el mismo nombre declarado.

Para voces y sonidos, se puede seguir el siguiente ejemplo:

```
class CfgSounds
{
    Sounds[] = {muestra2,muestra3,muestra4,muestra5};
    class muestra2
    {
        name = "muestra2";
        sound[] = {"\sounds\muestra2.ogg",db+10,1.0};
        titles[] = {};
    };
    class muestra3
    {
        name = "muestra3";
        sound[] = {"\sounds\muestra3.ogg",db+10,1.0};
        titles[] = {};
    };
    class muestra4
    {
        name = "muestra4";
        sound[] = {"\sounds\muestra4.ogg",db+10,1.0};
        titles[] = {};
    };
    class muestra5
    {
        name = "muestra5";
        sound[] = {"\sounds\muestra5.ogg",db+10,1.0};
        titles[] = {};
    };
};
```