

Assassination Mission Tutorial

This tutorial demonstrates the insertion and the extraction of playable and AI controlled characters. It will also demonstrate how to use Framework loadout functions to outfit characters.

To complete this tutorial you will modify a mission script. The map elements needed for the tutorial are supplied.

Preparing to Create the Mission

Copy the MissionAssassination.Stratis subdirectory to:

```
<MyDocuments>\Arma 3 - Other Profiles\<MyProfile>\missions\
```

Copy the Zen_FrameworkFunctions directory from the Shell.Stratis directory to:

```
<MyDocuments>\Arma 3 - Other Profiles\<MyProfile>\missions\  
MissionAssassination.Stratis\
```

Review the Map

Open Arma and then the Editor. Select Stratis Island and Continue.

Choose the load function and select MissionAssassination.

At Camp Rogain is a single BLUFOR player unit. Like other Framework tutorial missions, X11 is 'parked' on the map and will be moved to a starting position.

No changes are required to the map. Review the area markers in Agia Marina for some ideas on 'decorating' markers; these markers use different colors and icons.

Updating the Initialization Script

Open the *init.sqf* file in

```
<MyDocuments>\Arma 3 - Other Profiles\<MyProfile>\missions\  
MissionAssassination.Stratis\
```

Immediately above the lines

```
// All clients stop executing here, do not delete this line  
if (!isServer) exitWith {};
```

Add a mission briefing statement:

```
Player creatediaryRecord["Diary", ["Assassination Tutorial", "Travel  
West from your drop off point and find the ammo box. Kill the warlord  
on the beach. Go North East to the extraction point.<br/>"]];
```

At the point labeled 'Enter the mission code here' enter these statements:

Clothe your character as a civilian with a Framework load out function and assign your character a protected status:

```
0 = [X11, "Civilian"] call Zen_GiveLoadoutBlufor;  
X11 SetCaptive true;
```

Insert your character in a chauffeur driven car:

```
_CivilianCarPosition = ["Destination", [200, 250], [], 0, [1,200], [0,  
90]] call Zen_FindGroundPosition;  
_CivilianCar = [_CivilianCarPosition, "c_offroad_01_f", 270] call  
Zen_SpawnGroundVehicle;  
0 = [X11, _CivilianCar] call Zen_MoveInVehicle;  
0 = [_CivilianCar, ["Destination", _CivilianCarPosition], X11,  
"limited"] spawn Zen_OrderInsertion;
```

Create the objective for this mission using a Framework function. As usual, first generate a random point within an area marker and then place the warlord at that position:

```
_ObjectivePos = ["WarlordMissionOPFOR"] call Zen_FindGroundPosition;  
_warlordReference = [_ObjectivePos, X11, east, "Officer","eliminate"]  
call Zen_CreateObjective;  
_warlord = (_warlordReference select 0) select 0;
```

Place a guard on patrol NorthEast of the Warlord:

```
_InnerPos = [_warlord, [40,100]] call Zen_FindGroundPosition;  
_InnerGuard = [_InnerPos, EAST, "SOF", [2,3]] call Zen_SpawnInfantry;  
0 = [[_InnerGuard], _InnerPos, [70,150]] spawn  
Zen_OrderInfantryPatrol;
```

Create a squad to patrol further out:

```
_OuterPos = [_warlord, [170,220]] call Zen_FindGroundPosition;  
_OuterGuard = [_OuterPos, EAST, "infantry", [2,3]] call  
Zen_SpawnInfantry;  
0 = [[_OuterGuard], _OuterPos, [120,170]] spawn  
Zen_OrderInfantryPatrol;
```

Wait (loop) until your player character is one meter from the ammo box inside the building:

```
waituntil { sleep 2; X11 Distance BLUFORammo < 2 };
```

Then call a Framework load out function to outfit him as sniper:

```
0 = [X11, "Sniper"] call Zen_GiveLoadoutBlufor;  
X11 SetCaptive false;
```

Wait (loop) until the warlord is dead:

```
waituntil { sleep 2; !(alive _warlord) };
```

The helicopter is called as soon as the warlord is dead. It will wait at the landing zone until you board. Then it will fly away and the mission will end. Of course, it can be destroyed by enemy gunfire, either before or after you board.

```
_ExtractionPos = ["BLUFOR_Extraction"] call Zen_FindGroundPosition;  
_heliSpawnPos = [_ExtractionPos, [1000,1500], [], 0, [0,0], [45,90]]  
call Zen_FindGroundPosition;  
_heliEndPos = [_ExtractionPos, [1500,2000], [], 0, [0,0], [45,90]]  
call Zen_FindGroundPosition;  
_X11Helicopter = [_heliSpawnPos, "b_heli_light_01_f", 60] call  
Zen_SpawnHelicopter;  
0 = [_X11Helicopter, [_ExtractionPos, _heliEndPos], X11, "normal", 60]  
spawn Zen_OrderExtraction;
```

Wait (loop) until you are in the helicopter and 1200 meters from the landing zone then end mission.

```
waituntil {  
    sleep 2;  
    ((X11 distance _ExtractionPos) > 1200)  
};  
endMission "endl"
```

Play the Mission.

To launch this mission from inside the editor select 'Preview'.

Post-Mortem

If you played the mission here's what you should have seen:

- A briefing
- A task for the kill the warlord/officer objective.
- After killing the warlord the task completion notification should display.
- After flying 1200 meters the mission should end.

Technical Corner

This mission highlights these features of the Co-op Framework:

- Custom loadouts
- Insertion of the player using a car
- Player extraction using a helicopter.

Loadouts

The Loadout functions are standardized 'kits' for all types of units, both playable and non-playable. The three functions are *Zen_GiveLoadoutBlufor*, *Zen_GiveLoadoutOpfor* and

Zen_GiveLoadoutIndfor. Within each there about 20 kits; they include Rifleman, Sniper, Grenadier, Spotter and Diver. These functions will obviously always be works-in-progress as BIS adds new weapons and uniforms to Arma III.

The kits can enhance a unit's loadout whether the unit is placed within the Arma III editor or spawned by a script.

For example, if you weren't satisfied with the random makeup generated by *SpawnInfantry* you could have all the units in *InnerGuard* possess machine guns. Add this line after the call to *OrderInfantryPatrol*:

```
0 = [_InnerGuard, "AutoRifleman"] call Zen_GiveLoadoutOpfor;
```

NOTE: The ability of a unit is determined by the skill level as defined in *Zen_SpawnInfantry*; the loadout is the uniform and carried equipment.

It might be tough going to escape with just a sniper rifle, especially if an OPFOR squad is waiting for you near the landing zone. After the wait loop that tests if the warlord is dead you could spawn an inline trigger and change the player's loadout. This inline trigger is not a feature of the Framework but is useful language structure for a script designer to know.

```
0 = [] Spawn{
    waituntil { sleep 2; X11 Distance BLUFORammo < 2 };
    0 = [X11, "AutoRifleman"] call Zen_GiveLoadoutBlufor;
};
```

Insertion and Extraction

There are one insertion and one extraction function in the Co-op Framework. As expected, they have the same order in their parameters:

- Reference to the vehicle
- One or two locations (formed in an array)
- Passengers – An array of units or a reference to a group
- Speed
- Height (to fly at, if appropriate to vehicle type)

Both insertion and extraction give no consideration as to where the unit is on the map when the function is called.

If a single location is specified in the second parameter, the unit moves to that location.

If insertion is specified and two locations are indicated in the second parameter, the function moves the unit to the first location, waits for all the passengers to exit, and then travels to the second location. Calling the extraction function with two locations specified causes the unit to move to the first location, wait for the passengers to board and then travel to the second location.

Insertion and extraction also work with non-playable units opposing the player.

The next code example inserts an OPFOR squad into the mission after the warlord is killed and transports them to the original drop-off point of the assassin. The squad is spawned with a [0,0,0] starting position since they will be moved into the vehicle with a Framework function after the vehicle is spawned.

```
_VehiclePosition = ["Destination", [400, 550], [], 0, [1,200], [0,
90]] call Zen_FindGroundPosition;
_Zamak = [_VehiclePosition, "O_Truck_02_covered_F", 270] call
Zen_SpawnGroundVehicle;
_ReactionGroup = [[0,0,0], EAST, "infantry", [3,5]] call
Zen_SpawnInfantry;
0 = [_ReactionGroup, _Zamak] call Zen_MoveInVehicle;
0 = [_Zamak, ["Destination"], _ReactionGroup, "limited"] spawn
Zen_OrderInsertion;
```

(Place this code after the call for helicopter extraction.)

Framework Return Values

Throughout these tutorials the script code shows return values from Framework function calls. Often these references aren't used by subsequent statements. It is recommended that return values be coded to allow convenient future enhancements of the script.

This script contains the statement:

```
_warlord = (_warlordReference select 0) select 0;
```

The variable *_warlordReference* is an array with two elements. '*_warlordReference select 0*' means to select the first element in the array. This element is itself an array. So select the first element of this array with the command '*select 0*'.

Objective Completion

Note the test for the death of the warlord that indicates that the init script should pass control to the next statement. This is an alternative technique to calling the Framework function *Zen_AreTasksComplete*, which has been demonstrated in earlier tutorials.