Assassination Mission (Random) Tutorial

This tutorial extends the original Assassination Mission Tutorial to have player insertion and extraction at random positions. It will also make a quick look at random loadouts for players.

Preparing to Create the Mission

Copy the MissionAssassinationRandom.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\
> MissionAssassinationRandom.Stratis\

Review the Map

No changes are required to the map. The location of the warlord, patrolling squad, ammo box, infiltration point, safe house and extraction point will be randomly generated with Framework code.

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.

Reviewing the Initialization Script

Open the *init.sqf* and *misc.sqf* files. This tutorial is not the ‘enter the code’ variety. It will just discuss the various sqf statements.

This tutorial is more complicated than prior ones but each step is a manageable handful of sqf statements.

The general order of the mission is this:

> Infiltrate player as civilian in automobile
> Locate ammo box and change to WEST assassin
> Assassinate the warlord
> Travel to safe house to meet local contact
> Extract by helicopter that evening from remote location

The mission starts with a declaration of a global variable. It will be used by the function that generates the position of the landing zone for the extracting helicopter. More on this variable later.

Infiltration

The code for infiltration is very similar to the other ‘randomizing’ tutorials and the original assassination tutorial.

One small change is that the function *f_getrandomcityAreaMarker* in *misc.sqf* has been changed to only select cities and the capital; small towns are excluded. Some of them don’t have roads or enough buildings to support this mission.

The major difference is the manner of filtering area markers. First, almost all water zones are removed. There’s too much clutter around the water: breakwaters, piers, shacks, etc.

Then all urban zones greater than 20% threshold are removed. But this isn’t what the tutorial needs. It needs all urban zones *greater than* 20%. So subtract the filtered array from the original array.

Ammo Box

Determining an indoor location for the ammo box is done in a while loop. This loop just insures that the mission finds a ‘placeable’ position; the designer doesn’t have to worry about the occasional situation where *Zen_FindBuildingPositions* returns empty array.

The *Zen_GiveLoadoutBlufor* function allows the specification of multiple loadouts in the second argument. One of these will be chosen at random.

Objective

Place the ‘eliminate’ warlord objective at a random position within the town. This area marker is an attribute of the town taken from the map config file.

Safe house

Generate the civilian contact that will harbor the assassin using familiar code.

After the player reaches the contact segue into a title text, change the time to evening and move both player and civilian to the landing zone.

Both are moved a few yards to one side so the helicopter doesn’t land on them.

Extraction

The generation of the landing zone position begins before the ammo box is reached. A custom function is ‘spawned’ just before the wait loop of the ammo box milestone:

```
0 = [_cityCenter] spawn f_createExtractionPosition;
```

Since the function is spawned it runs in the background and it doesn’t matter how long it takes to run. But it will certainly be complete before the warlord objective is completed.

The results of the function are stored in the global variable *extractionPosition*, which is specified as global because it doesn't begin with an underscore.

A line is put into the script as 'defensive' coding just to protect against unlikely possibly that *extractionPosition* is still an empty array:

```
waituntil {sleep 5;count extractionPosition > 0};
```

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

Interior Placements

The ARMA engine is apparently sensitive to instances of object collision. When the function *Zen_SpawnAmmoBox* is called the game will invariably put the ammo box just outside the perimeter of the building.

To force the ammo box back inside the building use the *setposATL* function. But it sometimes happens that ammo box ‘disappears’ from the game world. It is probably under the house or in some ‘fifth dimension’ of the geometry.

Extraction Grid

Note the technique used in the f_createExtractionPosition function to generate positions. It only creates positions that are on the outer edge of the grid.

This test:

```
    if (_i==0 or _j==0 or _i==_YDIM-1 or _j==_XDIM-1) then {
        [_positionarray,[_newX,_newY]] call Zen_ArrayAppend;
    }
```

Produces an array of positions that can be converted in this area marker grid: