

An aerial view of a military aircraft, possibly a P-51 Mustang, with a camouflage pattern of brown and green spots. The aircraft is flying over a game map that features a winding river, green fields, and a residential area with houses and roads. The aircraft is positioned in the upper left quadrant of the frame. The title 'SUDDEN3 STRIKE' is overlaid on the map, with the number '3' inside a green reticle. The text 'Arms for Victory' is in a blue, stylized font, and 'MAP DESIGN & MISSION DESIGN' is in a white, bold font. The author's name 'BY KANE PETERSON' is at the bottom in a smaller white font.

**SUDDEN3 STRIKE**

**Arms for Victory**

**MAP DESIGN &  
MISSION DESIGN**

**BY KANE PETERSON**

# CONTENTS

Sudden Strike 3 mapping manual.....	2
Map-making .....	2
Starting out .....	2
Geography .....	3
Roads and paths .....	10
Buildings and infrastructure .....	11
Pathfinding Grid .....	12
No pasaran layers .....	13
Different brushes for layers .....	14
Trenches .....	15
Mission Environment Settings .....	15
Finishing Touches .....	17
Avoiding data loss .....	17
Mission-making.....	18
Players and nations .....	18
Objectives .....	18
Victory and defeat conditions .....	34
Reinforcement spawn locations .....	38
Reinforcement packages .....	39
AI zones .....	41
AI zone to zone troop movements .....	42
AI guard groups and zones .....	46
AI defence scheme .....	50
Reinforcement conditions .....	51
Enemy strike conditions .....	57
Placing units .....	63
Player bonuses .....	64
Player airfields, battleships and aircraft carriers .....	68
Coloured messages .....	69
Cameras .....	71

# SUDDEN STRIKE 3 MAPPING MANUAL

By Kane Peterson

To get the most out of this manual, you should just follow all of it to completion when making the map. Take whatever notes you need. Go back to parts of it that you need to use again.

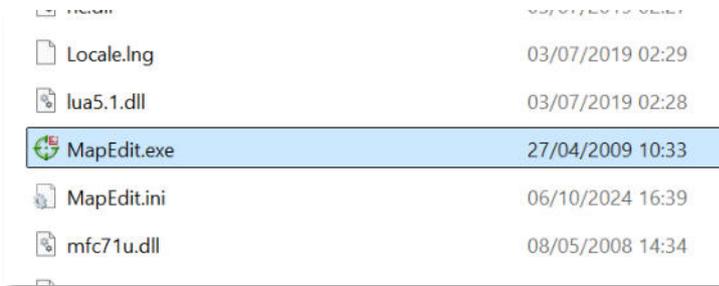
## MAP-MAKING

### Starting out

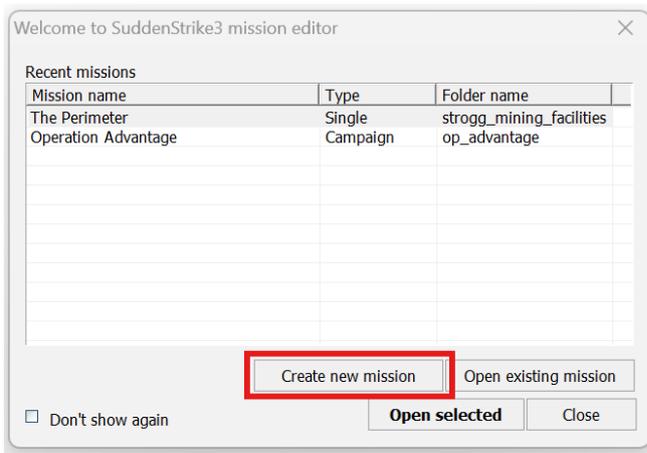
You must have the correct application: "MapEdit.exe"

On Steam, this may be located in C:\Program Files (x86)\Steam\steamapps\common\Sudden Strike 3

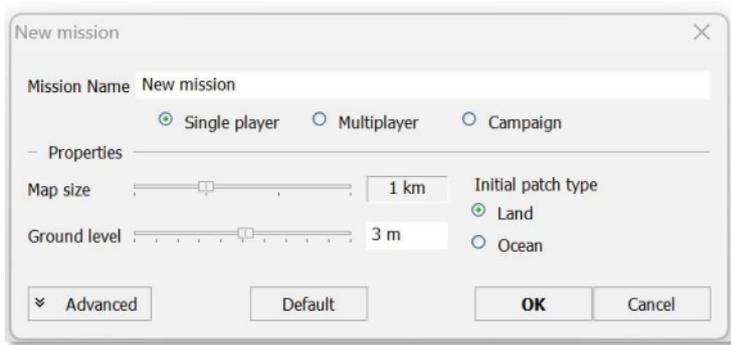
Double click the application to go on it.



After entering the application, select "Create New Mission".



Choose your preferred properties including map size. Also create a name for the mission. It is important to decide the map size now because this cannot be changed later. Other features like ground level can be useful if your scenario is set on predominantly high ground, because adjusting the terrain below 0 will result in being at sea level. When you are ready, select OK.

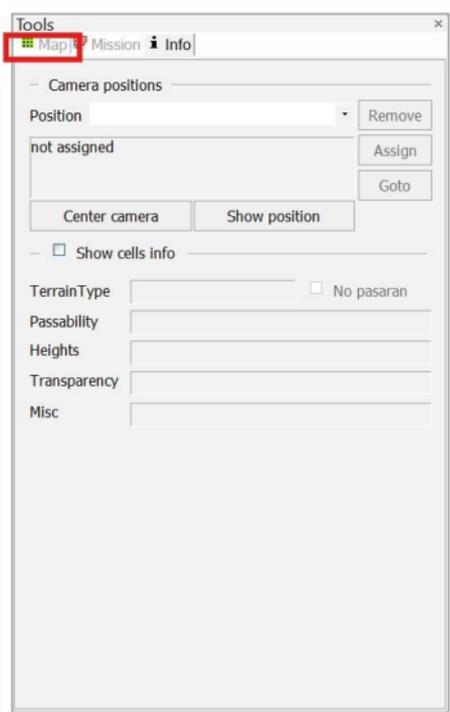


## Geography

### First layers e.g. grass, sand

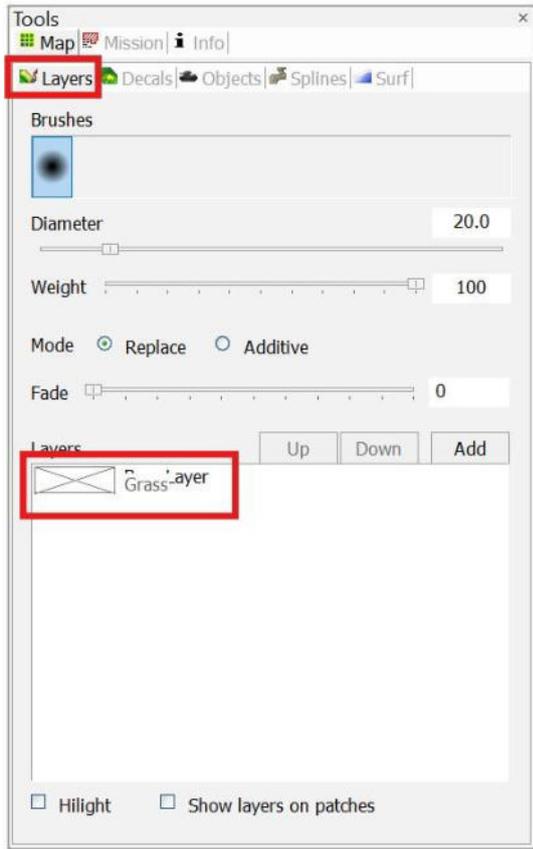
For now, the map will appear to be white and featureless. The first thing we want to do is add some layers that will give the terrain a texture.

See the "Tools" box (usually loading on the right) and click the "map" tab.

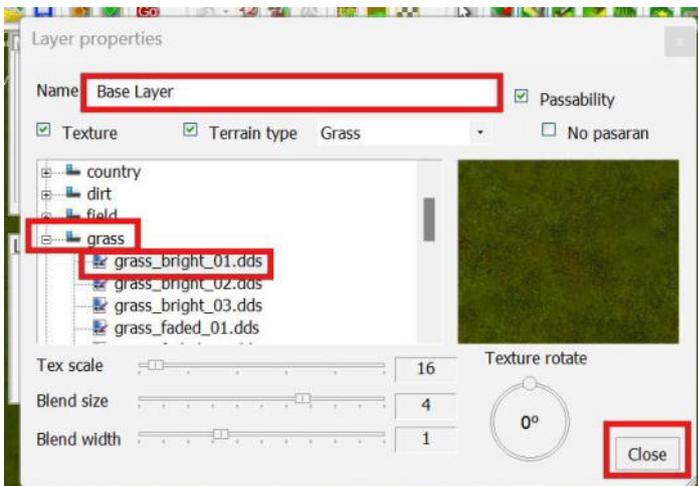


After this, you will see additional tabs. The one directly under "map" will say "Relief" by default, but that is not what we need first. Click the tab so that it changes to say "Layers". Clicking the tab will toggle it to a different option, so if it does not say "Layers", just click it again until it says this.

After reaching the "Layers" option, double click the box with a cross in it, where it says "Grass". This represents our first layer, which is blank and white by default.

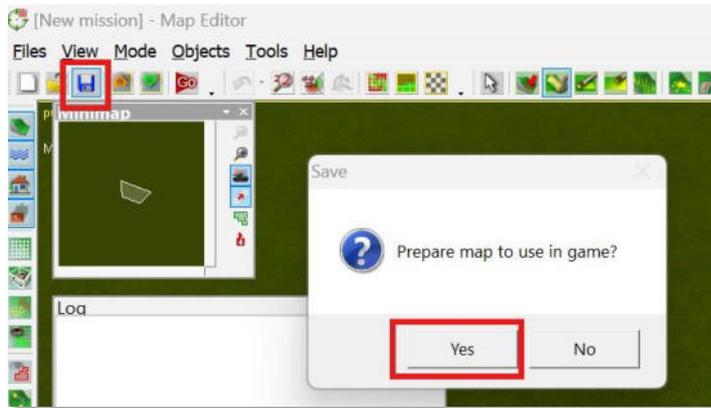


In the "Layer properties" box now open, you should see the terrain type is automatically set as grass for a new layer, which will cause this terrain to affect unit speed like a grass field and also generate muddy explosions when hit by shells. However, it has no texture. To change texture to grass, click the "+" at the left of where it reads "grass", then click one of the grass textures that appears below this. Select a different one if you prefer a different texture, for example, a brighter one. You can also edit the name of the layer where it says "Name", to "grass" if you prefer, allowing you to label the layers as you add them. When you are happy, just click "close".



### Updating minimap preview and saving

The minimap will not yet have updated to our new texture layer. For that, click the save button at the top left to save your map. Choosing "yes" to prepare map to be used in the game will allow everything to update including the texture, although clicking "no" may allow faster saving when you make more limited changes to your map and are not ready to use the map in the game.

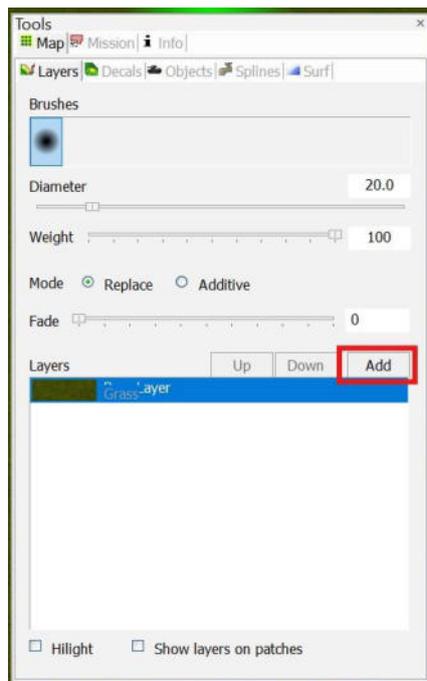


### Moving the editor camera

Now we might want to view the map. This is done by holding ctrl and left mouse button, while moving the mouse. If you want to alter the pitch or rotate your camera, hold ctrl and the right mouse button while still moving the mouse. In addition, hold ctrl while using the scroll wheel to zoom in and out significantly more than in the game. Try looking around the map for a while if you are not familiar with the controls.

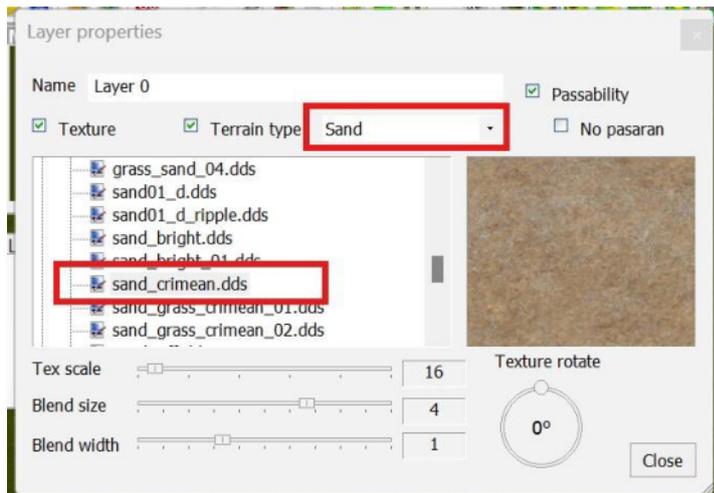
### Additional layers

When you are comfortable, create another layer. Just click "Add" on the Tools box while you are on the "Layers" tab there.

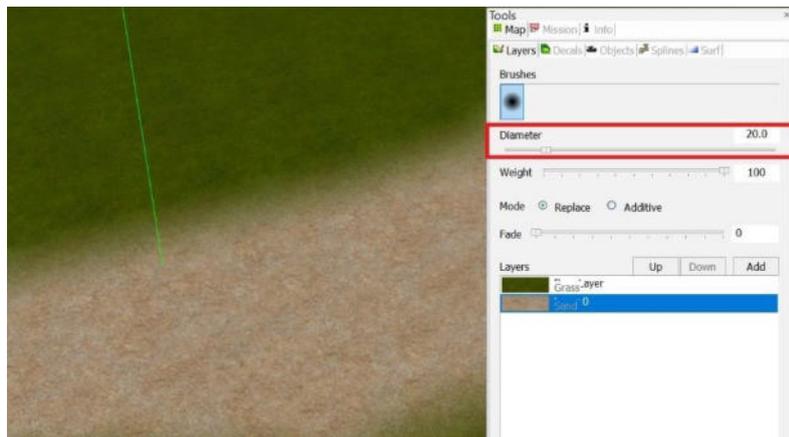


Creating areas of sand where you intend water bodies and rivers to go would be a good next step at this stage of making a map.

As before, select your preferred texture. You can scroll down here to find more options, and use the "+" to expand the options. Choose sand in this example, which will often be present in many maps. In addition, where it reads "Terrain type", this time click where it reads "Grass" to reveal more options. Choose "Sand". This will cause this layer to behave like sand. You can also name the layer "Sand" to help find it. As before, click Close when ready.



Now, at your map, hold down the left mouse button and move the mouse to paint the map where you want the sand to be. You can adjust the brush diameter with the slider at the Tools panel if you need it to be larger or smaller. You can try the other sliders to make the brush have a softer or less bold effect to blend with the grass, too.

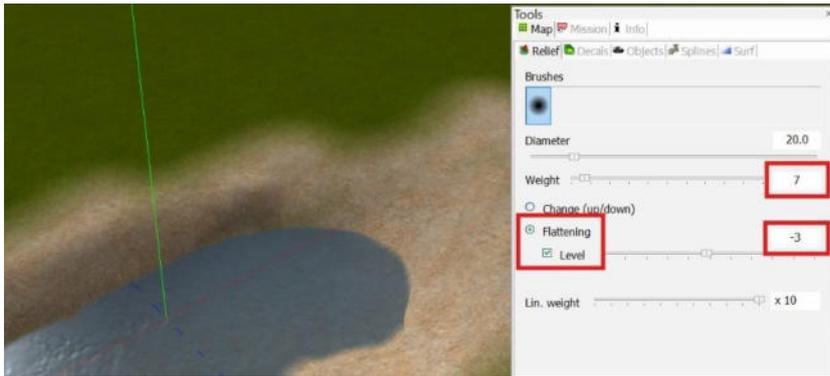


Try adding additional layers, but be mindful that more layers may be needed after you have added many more features to your map, so just three or four is okay to get started.

### Water bodies

Lakes and rivers should be added as the next step in the mapping, as they can be tricky to handle alongside buildings, trenches and other detail that will be placed in the map later.

Go to your tools panel again and keep toggling by left clicking the "Layers" etc. tab until it is back to saying "Relief". When making water bodies, we ideally want to be in control of the water depth. Click the option for "Flattening" and then the tick box for "Level". Now adjust the slider next to "Level" to how many meters deep you want the water to be. Try -3 for a basic lake depth. However, also adjust the "Weight" slider to something softer, like 7. Now paint the sandy part of your map gently to make a body of water there. Because the Weight setting is low, the bank of the river or water body will not be too steep, which makes it better for units to be able to reach it.



Try painting a river or lakes across your map if you want. Combining this effect with the sand will create a more realistic environment where there are sandy shores near the water bodies, and units will move slower on the sand, as well as the explosions and craters looking more appropriate here.

### **Elevation**

Be cautious with elevation early on, because too many hills and mountains may make it harder to decide where to place your roads and buildings. This may be different if your map is set in the mountains and there are very few roads or buildings here.

If you plan to add airfields to your map, be careful that hills are not placed in front of them, and ensure that the airfields have completely flat terrain to be placed on. Even small irregularities can cause the game's aircraft to explode on the runway because their collision box overlapped the terrain! Also, planes in Sudden Strike 3 tend to fly at 45 meters and may dive when performing attacks, and cannot detect the ground or avoid it, so hills that exceed 30 meters are not recommended as the planes cannot avoid them and will crash into them!

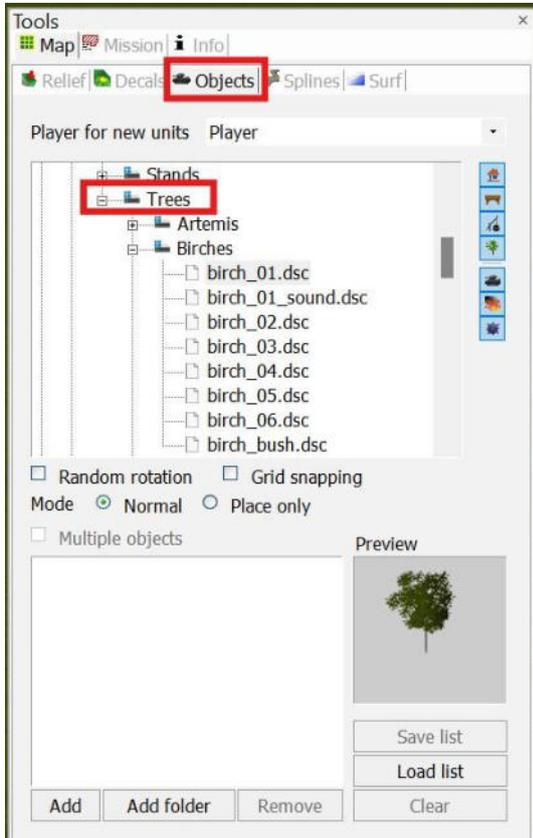
If you are sure where you want the hills to be, this early stage of the mapping is a good time to add them. To do this, simply repeat what you did with the water bodies but adjust the "Level" to a positive value at the Tools panel, e.g. 10. Then gently apply this to make hills that will not exceed 10 meters. To make the hills higher or begin adding taller peaks, adjust the slider for "Level" to the height desired.

It is good practice to make the natural features of a map first, but leave some flat areas for settlements and roads to connect. Add extra layers into your map for mud, swamps, stone at the sides of mountains and hills, and whatever else you think adds to the map.

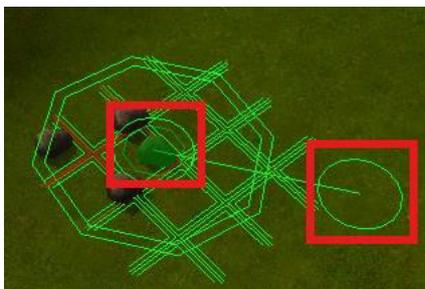
## Forest and rock

Any large areas of forest and foliage can be a good choice at the start of making a map.

Click "Objects" at the Tools panel to get started with this. Where it shows the file structure, click the "+" buttons to expand the category "Objects", then "Flora", then "Trees" and also expand the type of tree preferred.



Double-click (left mouse button) the name of the asset you want to use, and then left click the part of the map where you want the object to go. You can copy a selected object within the editor again by using ctrl+c while it is selected on the map, then move your cursor to where you want a copy to go, and use ctrl+v to place a copy. You can also move the object with the circle at the asset's position by dragging this with the left mouse button down, or rotate it by doing the same with the second circle extending from its position.



You can select entire groups of objects and drag or rotate all of them in the same way, as well as copying and pasting them. By placing and rotating several trees in clusters, you can copy and paste large groups of them or

drag them across the map. By this method, large wooded areas can be quickly present on the map.

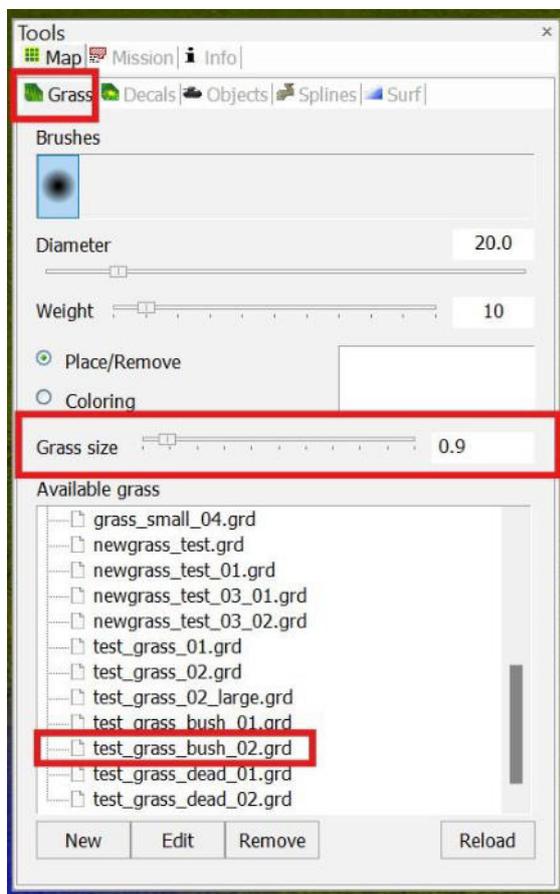


If you want undergrowth and extra foliage in the woods, try adding "bushes", also found under the flora category, placing large groups of these in the same way.

You can also place rocks and pieces of cliff from the objects library into the map just as you did with trees.

### 3D grass

For additional foliage effects and long grass in the natural landscape, you can toggle the left tab used for Relief and Layers until it says "Grass". Try altering the "Grass size" at the slider and choose available grass by double clicking the file name you see.

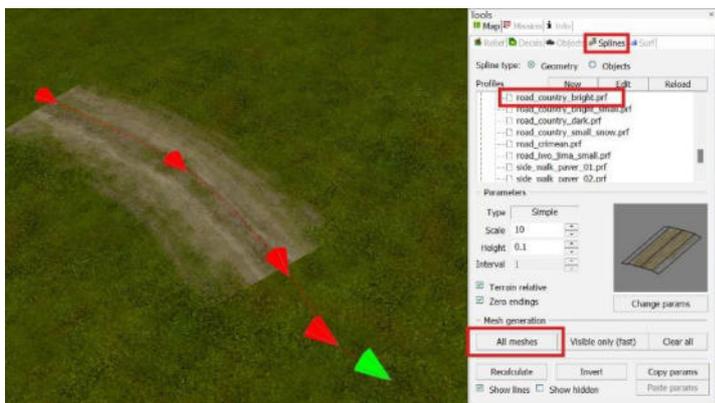


Paint the relevant area of the map with the grass, adjusting the sliders to make it denser or cover a larger area at your discretion. Some types of grass especially the "bush" grass set at 0.6 can be used with firs and wild areas of the map for convincing pine forest environments.

## Roads and paths

Some buildings can be added at this stage (later step), but you may prefer to add the road network on the map before placing most of the buildings, since this road network will decide how the buildings should be oriented. A more realistic city can be planned out with the roads, and they can be adjusted again later if change your mind about how the city is set up.

Go to the Tools panel again and click Splines, then double click your preferred spline, in this case one of the roads available.



Holding alt, use left click to place the nodes that the spline will follow. You can place them across the map. Deselect the spline or select it to start placing nodes for a new spline or add extra nodes for your selected spline, respectively. Drag the nodes as you would drag objects, by holding left mouse button over them and dragging the mouse, to change the shape of the spline. Double click nodes to delete them or hold alt over the spline and click once to add an extra node to a spline.

When you want to see the roads, click the "All meshes" button that will generate the roads or other features generated by splines across your entire map.

Note that splines for roads can cross each other to form crossroads, or end inside another road to connect to it, and it is okay for splines to extend across the entire map. Connecting the nodes of different roads is not recommended because it can cause tearing in the road meshes where they connect, so having them just pass into each other looks better.

Where roads reach the edge of the map, you should ideally not place the last node in the spline beyond the map edge because it can cause the road to bulge at the map edge when you save the game repeatedly during mapping. Instead, a "road end" decal or two should be placed there.

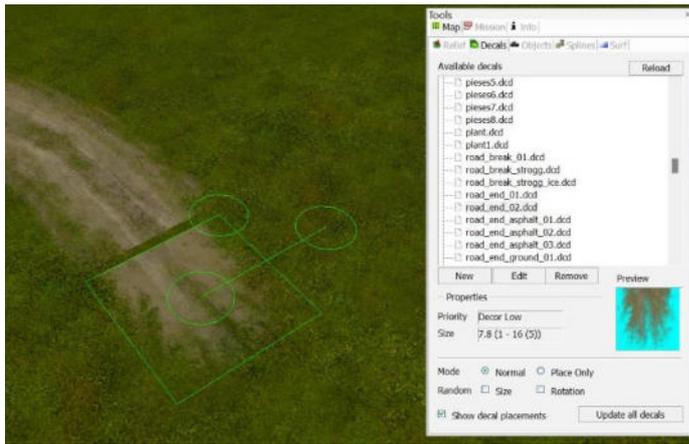
## Railways

Railways are placed the same way as roads. Trains automatically only move along these splines.

## Decals

Decals can add detail to the map terrain and also help patch up gaps between parts of the road mesh, preventing sharp edges being visible between road meshes or at the end of the road mesh.

To find and place decals, go to the second tab under the Map tab at the Tools panel where it reads Decals, and click that. You may now double click the decal you prefer, and then place it onto the map surface by clicking there. You can drag the circle at the corner of the decal to rescale it to a different size, or the centre circle to move it, or the circle at the end of the line to rotate the decal. "Road end" decals can be used at the end of a road so that the end is not abrupt or sharp but more natural.



You can use multiple "Road end" decals to create a fake section of road that is not affected by the shape of the terrain or the bulging effect at the map edge mentioned in the roads section. This is good practice as you will not have to worry about rebuilding the roads again.

## Buildings and infrastructure

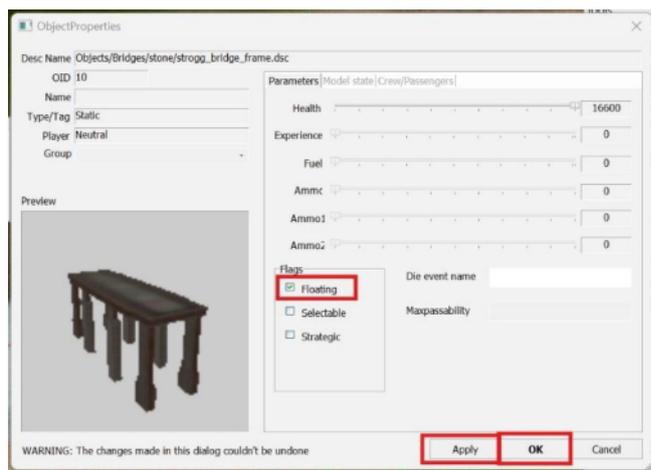
To place buildings, simply find them the same way as you did with the trees. You can also add lamp posts, telegraph posts and other infrastructure just as it would be placed alongside roads. You can also copy and paste these as desired.

You can pre-place infantry inside buildings, but that is covered in the mission building steps later in this manual.

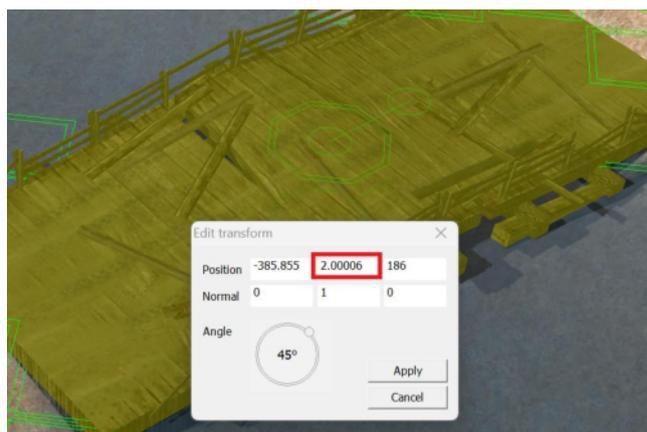
## Bridges

Bridges should be placed just as with other objects, but setting their height to connect properly to the land is more complicated.

Right click the object and select "Change parameters". After reaching the "ObjectProperties" window, tick "Floating" option and then click Apply, then OK.



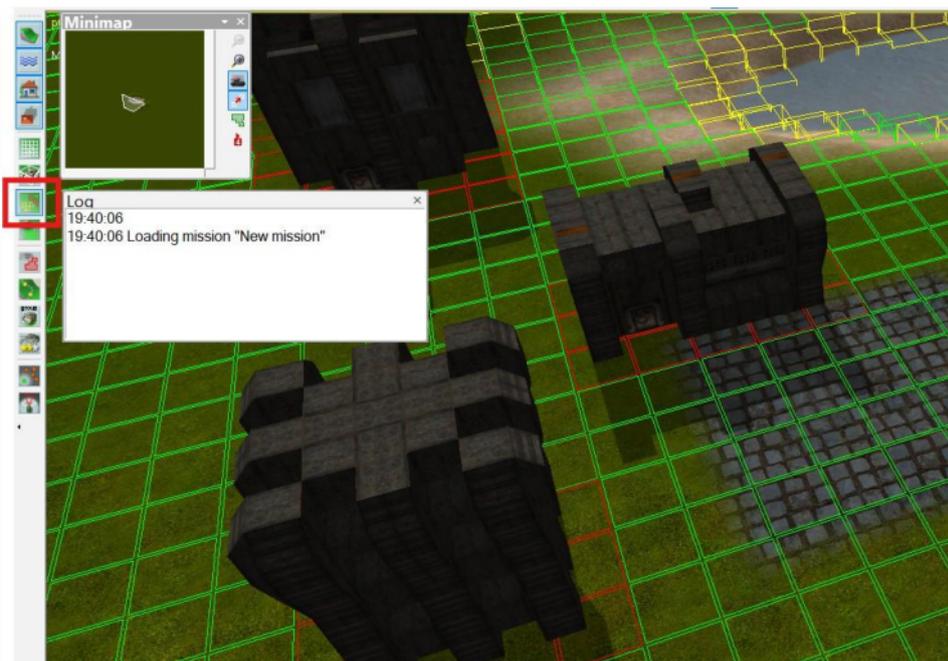
After this, right click the bridge again and select "Edit transformation". Edit the "position" box for the z axis (second one typically) and click apply until you are happy with the object height. Because this can be an unpredictable method, you may prefer to simply place the bridge in some deeper water where it is the correct height, then copy it and drag the copy to where you want. This is useful if there is more than one bridge piece being connected, as with a longer bridge.



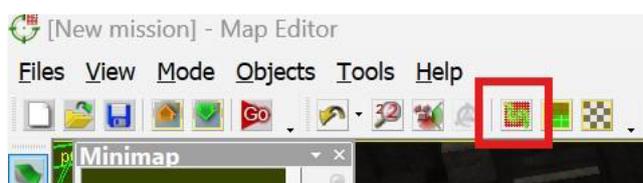
It is also useful to choose to change the parameters and tick the floating option for other objects that are partially inside the terrain, e.g. casemates and tunnel entrances.

## Pathfinding Grid

You may be concerned about units being able to pass objects or not. For this, you will want to make the pathfinding grid visible. Do this by clicking the Show/Hide Pathfinding Grid button on the left. Turn it off whenever you prefer with the same button.

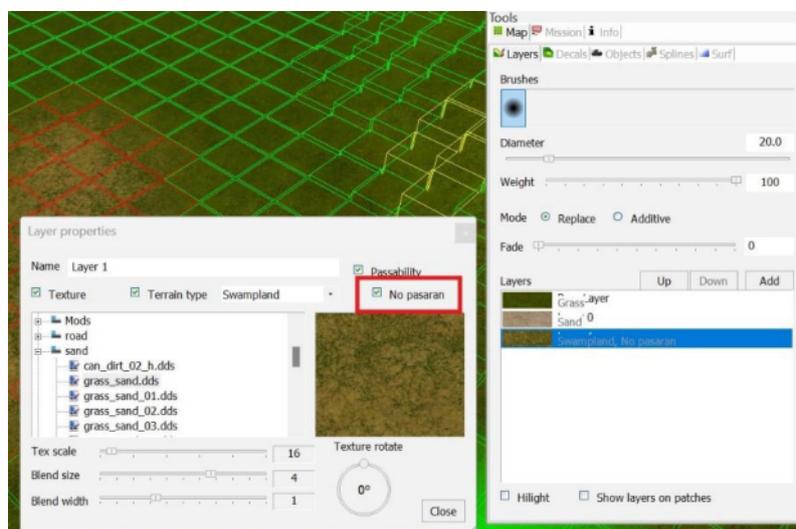


To generate the pathfinding grid after you have made changes, click the Generate Pathfinding Grid button on the top bar.



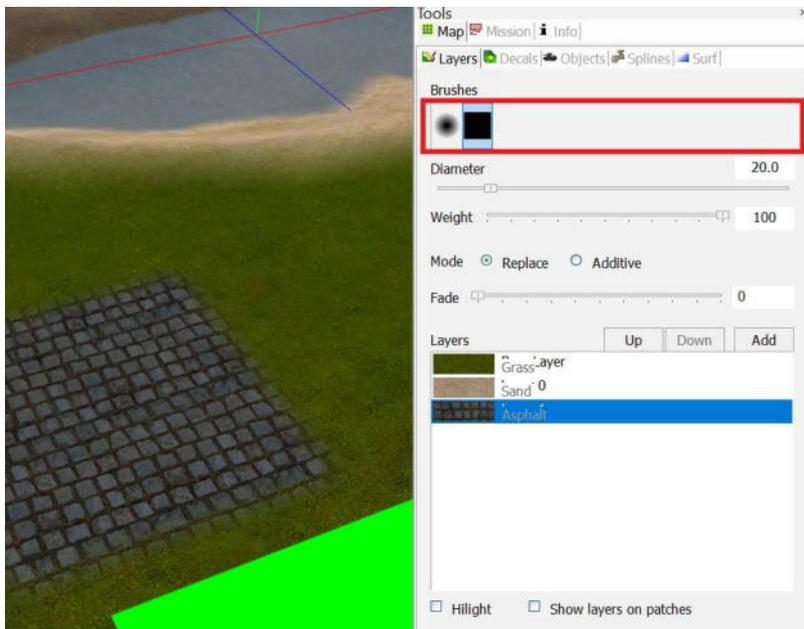
## No pasaran layers

You may sometimes want to make an area simply impassable to all units, e.g. quick sand, rocky slopes, and slippery edges that you imagine to be present on the map. For this, tick "no pasaran" when setting up the layer. Go back to the section about First Layers if you forgot how to set up layers. Generate the pathfinding grid when needed to see that the layer created impassable terrain.

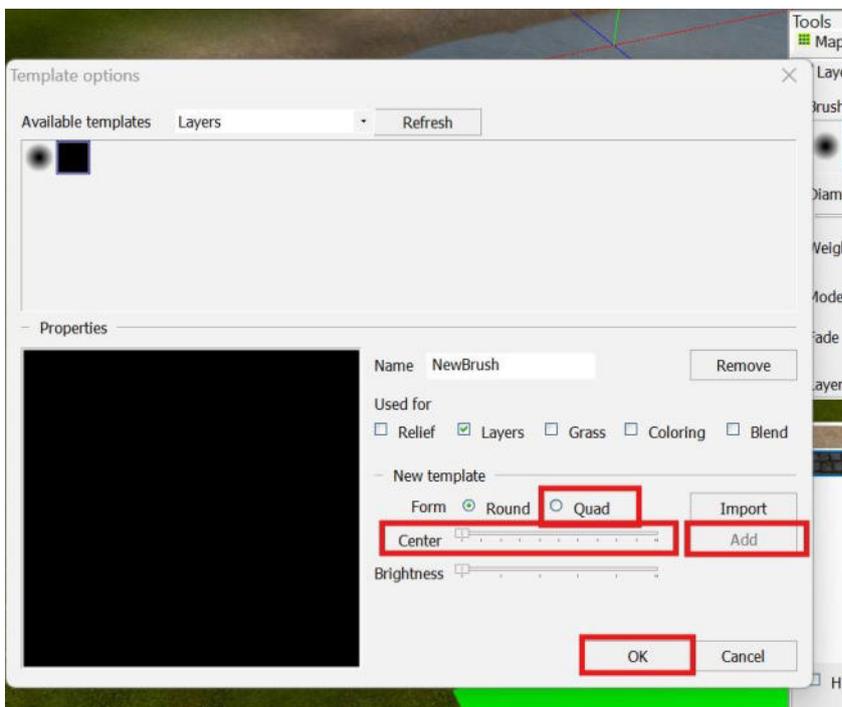


## Different brushes for layers

When painting layers with more finesse as the map gets more detailed, you can change brushes. You may want a sharp square brush for creating concrete areas of the map or fields with crops, for example. To do that, right click the box where it reads "Brushes" at the Layers tab and choose "Edit brushes".



Here, you can change the shape of the brush between circular ("round") and square ("quad") and adjust the "center" slider to get a sharper edge or a more blurred edge for the brush. When ready, click "Add" and then "OK".



You can then select this additional brush when painting the terrain with another layer.

## Trenches

Place trenches as you would place splines. Trenches are also selected as splines are, and placed the same. However, trenches can and should be connected up when appropriate by dragging nodes to other nodes, where they will become linked in the trench network.

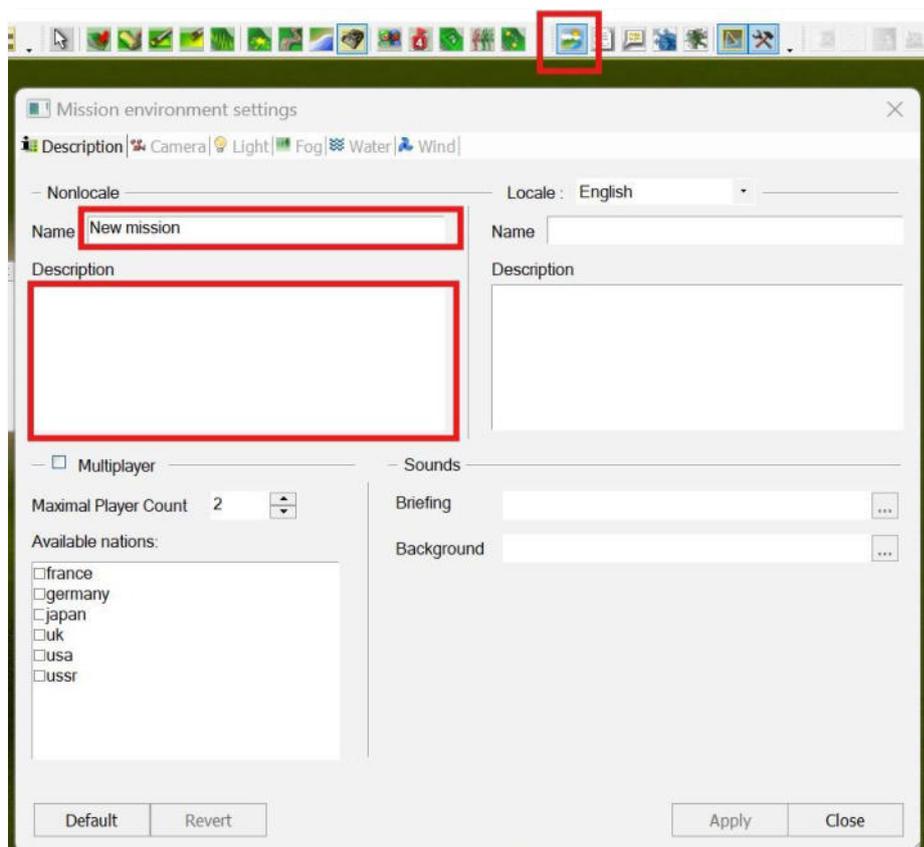
Trenches can be accompanied by placing fortifications and bunkers, as well as barbed wire, sandbags and other objects that can be found in the Objects library at the Tools panel, as explained in previous parts of this manual. Further terrain layers of dirt and sand can be applied with a fine brush along the trenches to make them look more like they were freshly dug by soldiers. All of this only uses steps explained already in this guidance.

## Mission Environment Settings

You may want to write the description of the mission, the official name of the mission, and also make some changes to lighting and weather for your map. All of this is stored at the same place.

### Mission Description

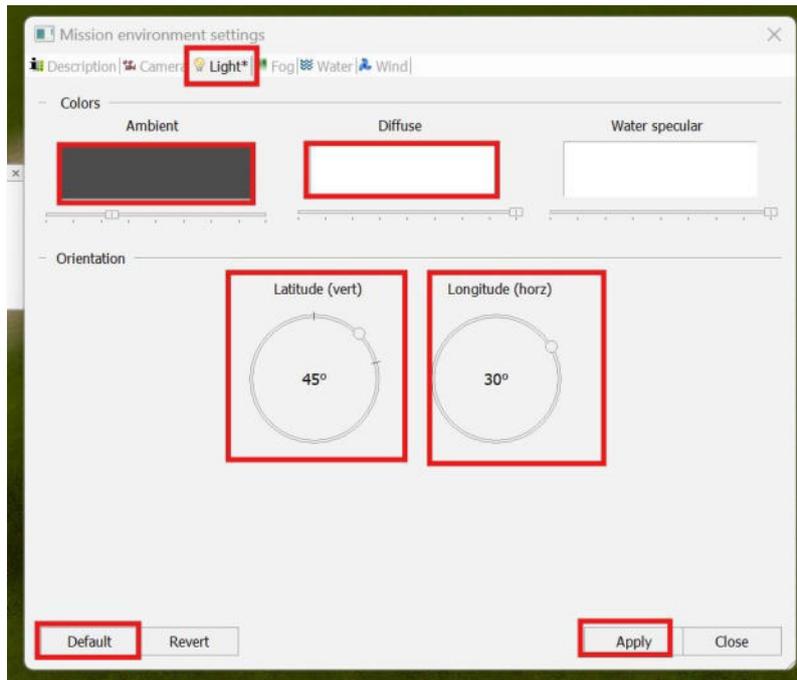
Click the Mission Environment Settings button. Here you can write the mission name and the mission description that will appear before people play the game.



### Light

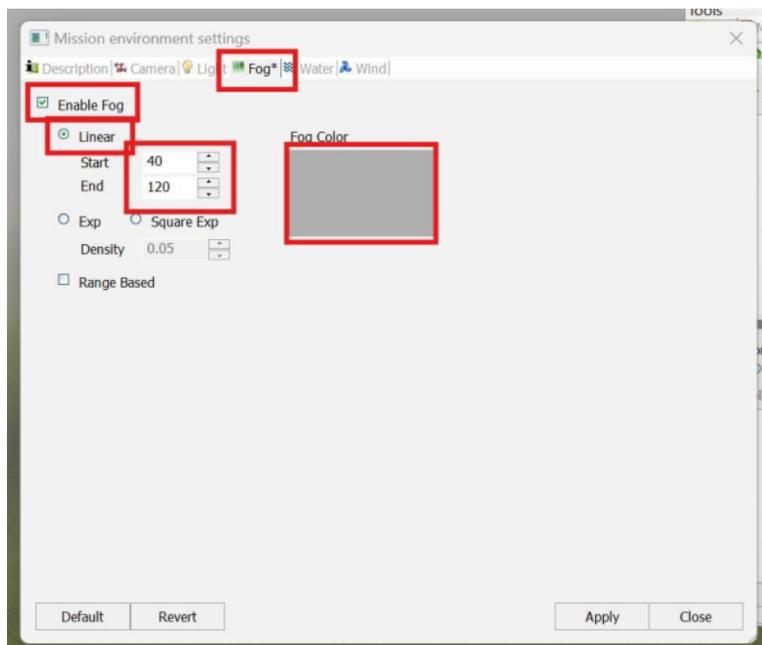
Next, try editing the lighting. Go to the "Light" tab. Adjust the ambient and diffuse light by double clicking the boxes there and choosing a different colour. Adjust the height of the sun in the sky (for example, make an evening map) by turning the Latitude dial and adjust the direction

of the sun by turning the Longitude dial). Maps can be set in a red sunset, for example by altering the lighting colours and latitude.



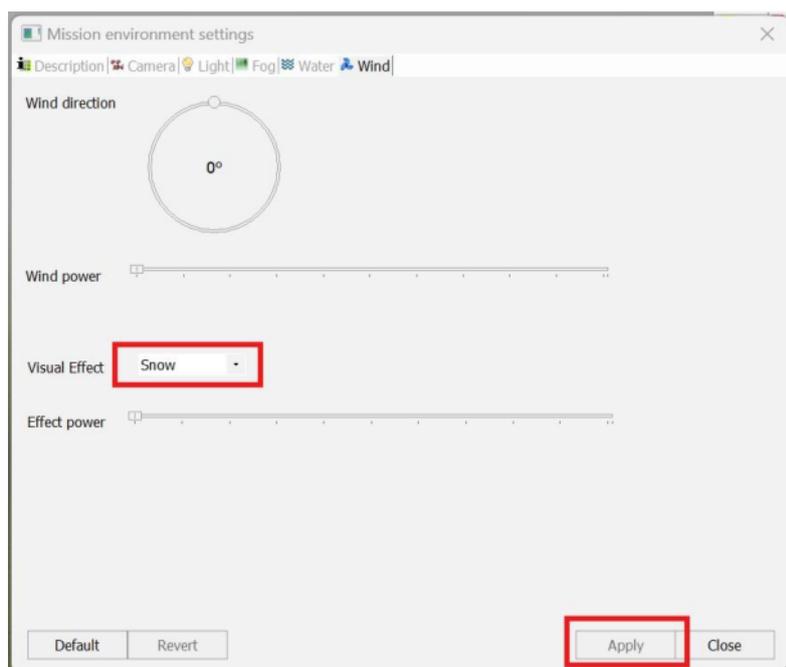
### Fog

Add fog by clicking the fog tab here too. Tick enable fog, choose Linear, set a starting distance for the fog e.g. 40 meters (where it starts), set an end distance e.g. 120 meters (where it become opaque) and double click fog colour to choose your preferred fog. If unsatisfied, revert to default by clicking the button for that. If satisfied, click Apply.



### Snow

For snow, move to the "Wind" tab. At "Visual Effect", choose Snow, and then apply.



## Finishing Touches

Look over the map and try to fill areas in with objects and some additional texture layers, for more diverse environments. Try to imagine yourself in the map, and place small objects like benches, even piles of rubbish, and other tiny details. You could place tree stumps to show that part of the map was deforested. Remember that video games are an art form and your goal should be to immerse the player and create a beautiful, detailed level.

When you are ready, you can start placing the opposing armies in your map and build the mission itself.

## Avoiding data loss

There are strange corruptions of files and errors that can result in a loss of progress. Save your files regularly.

Saved files are stored at "Missions" folder's "Single" folder. On Steam, this is normally C:\Program Files (x86)\Steam\steamapps\common\Sudden Strike 3\Media\Missions\Single

Each mission has its own folder based on your name for the mission when you initially saved the mission.

## Compressed backups

A good idea is to make a compressed backup of your mission each time you make significant progress making your map. Do this by creating a zipped folder containing your mission folder each time you make good progress. For example, you might save a version called "step 5 - trench network built.zip" so that if your progress is lost, you can continue from this step after you created trenches, by unzipping the files again and they will be intact.

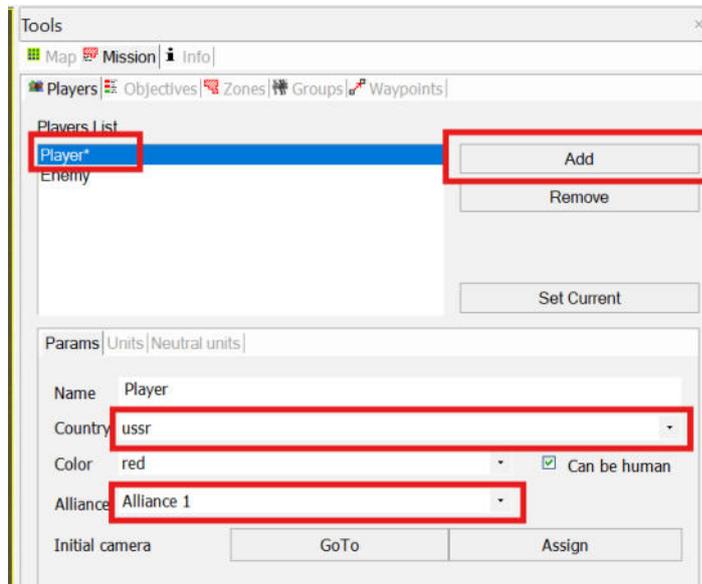
It is very frustrating if you were working on map in detail and then find that the files were corrupted or became inaccessible, so being able to go back to an earlier backup is very useful and will let you finish more maps and share them with the community.

## MISSION-MAKING

You should reserve the task of making the mission until all the mapping is complete in the map, otherwise you may have to change the scripts significantly due to mapping changes.

### Players and nations

At the Tools panel, click the player you want to set up.



#### Player

Additional AI players belonging to distinct nations can be added (e.g. adding British alongside Americans as their allies) with an extra player. Click "Add" to do this.

#### Country

Adjust the Country to choose the voice schemes you want to use for your units and the nation they will belong to. You can only choose four Allied nations and three Axis nations, of which Italy is not complete.

In the "Modern Arms" mod, Arab and Ukrainian voice schemes are used in their units by selecting USSR, whereas Israeli voice schemes are available by selecting USA. In the "Sudden Quake" mod, the alien Strogg replace Germany and Marines replace the USA in the files.

#### Alliance

There can only be two sides in a Sudden Strike 3 mission. These are Alliance 1 and Alliance 2, which must be chosen here.

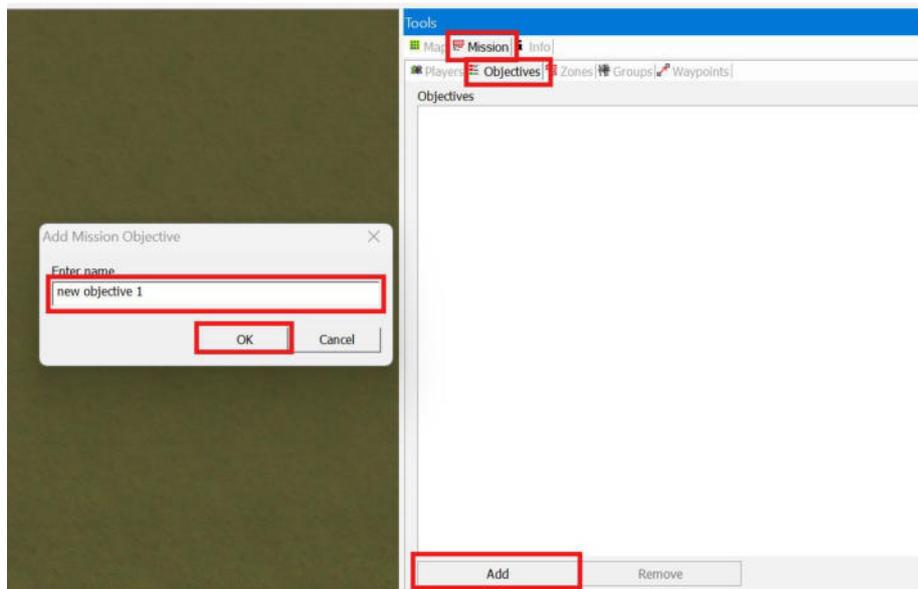
### Objectives

The next step should be to set up objectives in the important parts of the map, as the rest of the mission will be built around them.

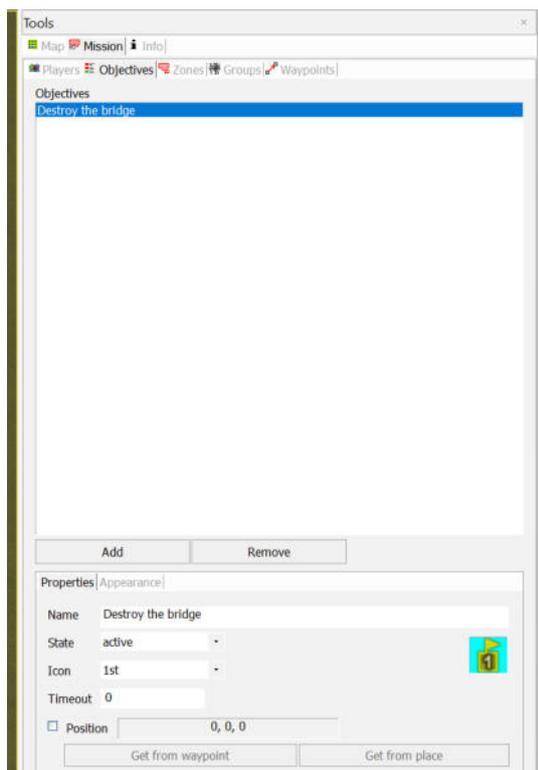
Objectives must be set up in the script editor, but are first defined at the Tools panel.

Choose the "Objectives" tab available while the "Mission" tab above it is selected.

Click "Add" to set up a new objective in your scenario. Type a name for the objective under "Enter name". You can use "Destroy the bridge" as an example that will be used next, and then choose "OK".



Now, you can set certain features for the objective.



You can change Icon, rename the objective again, or click "Appearance" for more options.

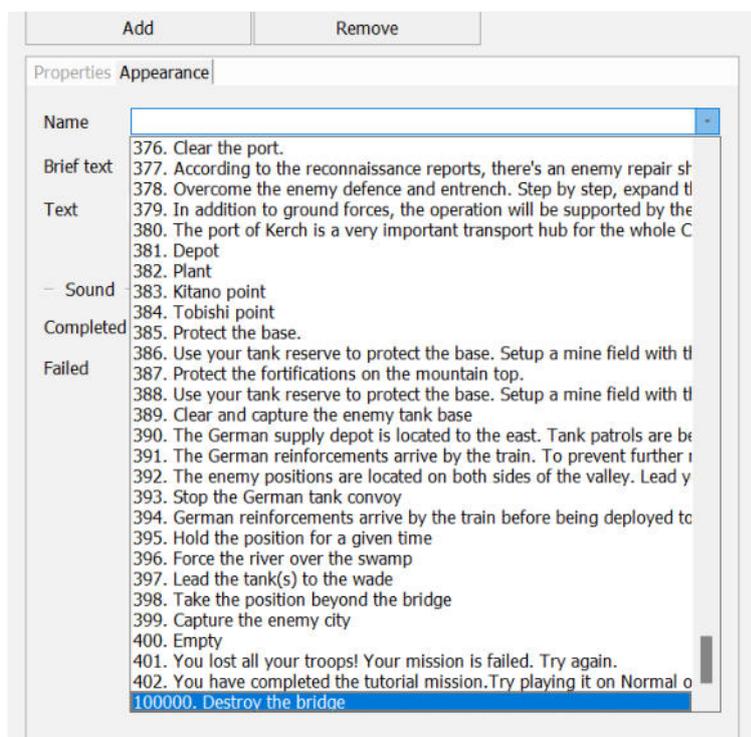
However, this only makes dropdown options available. To add new options, click "Message Editor". Here, you can



Now you can choose "Add", then edit the text of the message, and choose "Apply".



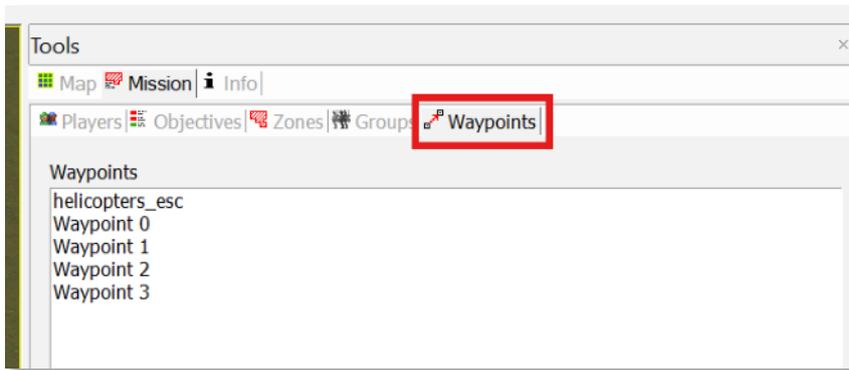
Close the Message Editor window, and your objective message should now become available at the Tools panel. Just scroll down until you find the message you made.



## Waypoint

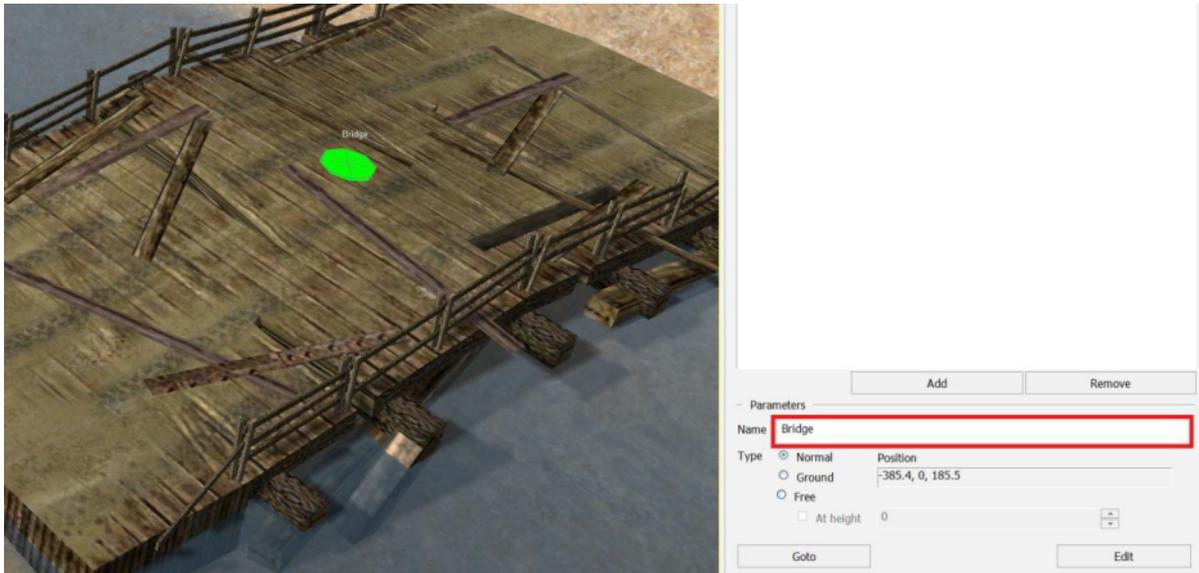
Next, you might want to create a location for the objective.

To do this, click the "Waypoints" tab still underneath the "Mission" tab at the Tools panel.

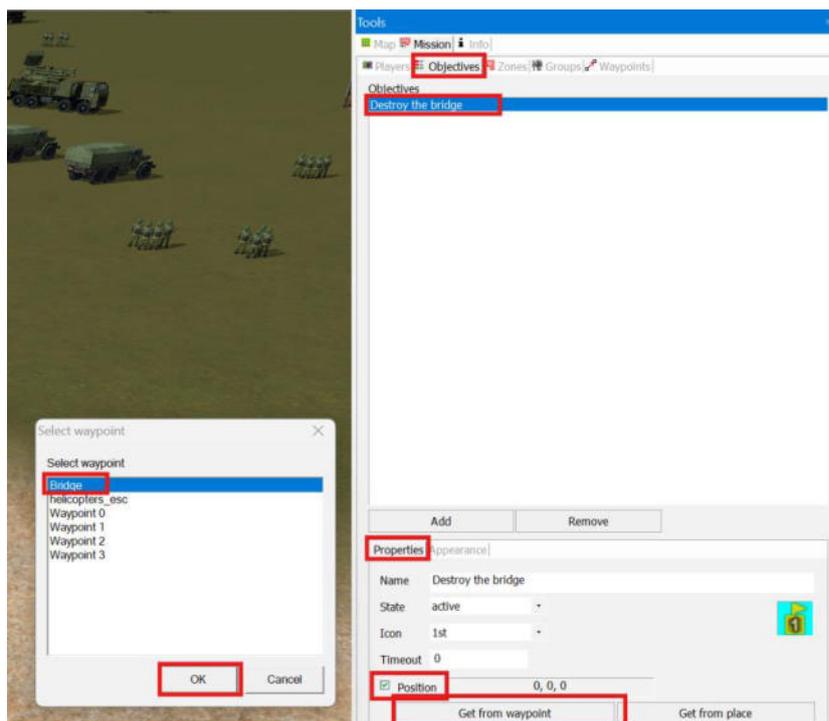


Although you can click "Add" and a new waypoint will appear in the center of the map that you can drag, the most effective way is to just hold the alt key and double click the part of the map where you want the new waypoint to be. You can drag the waypoint to a better location as you would with an object on the map in this mode.

You can also now rename the waypoint to something clearer, for example "Bridge".

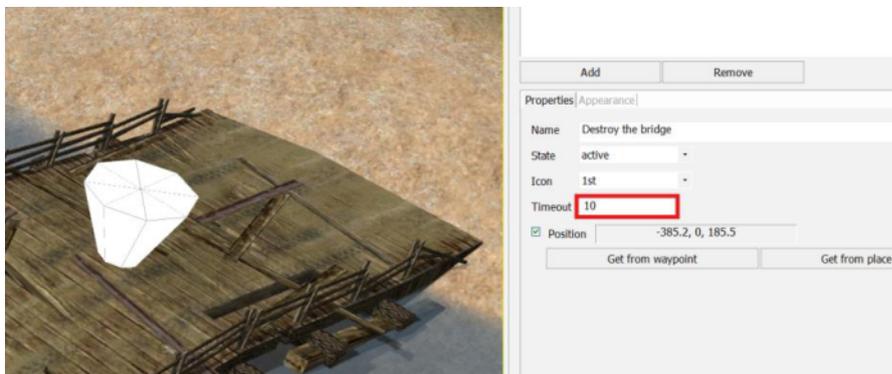


Now return to the Objectives tab, select your objective, and go to the objective's Properties tab.



Tick the Position box, click Get from waypoint, and then select the waypoint, in this example the Bridge, then click OK.

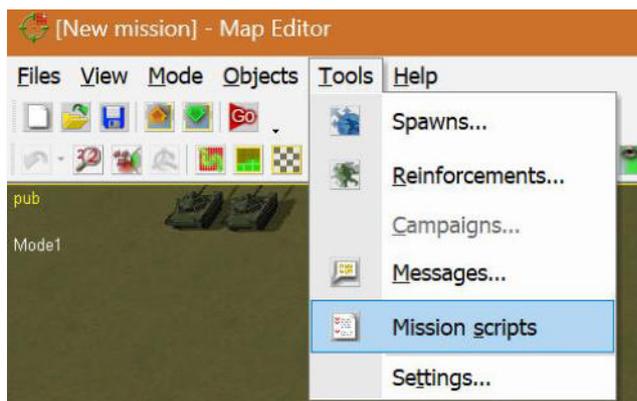
You can also edit the "Timeout" for the mission to create a delay in seconds (in this example 10 seconds) that will be shown to the player before the objective is completed.



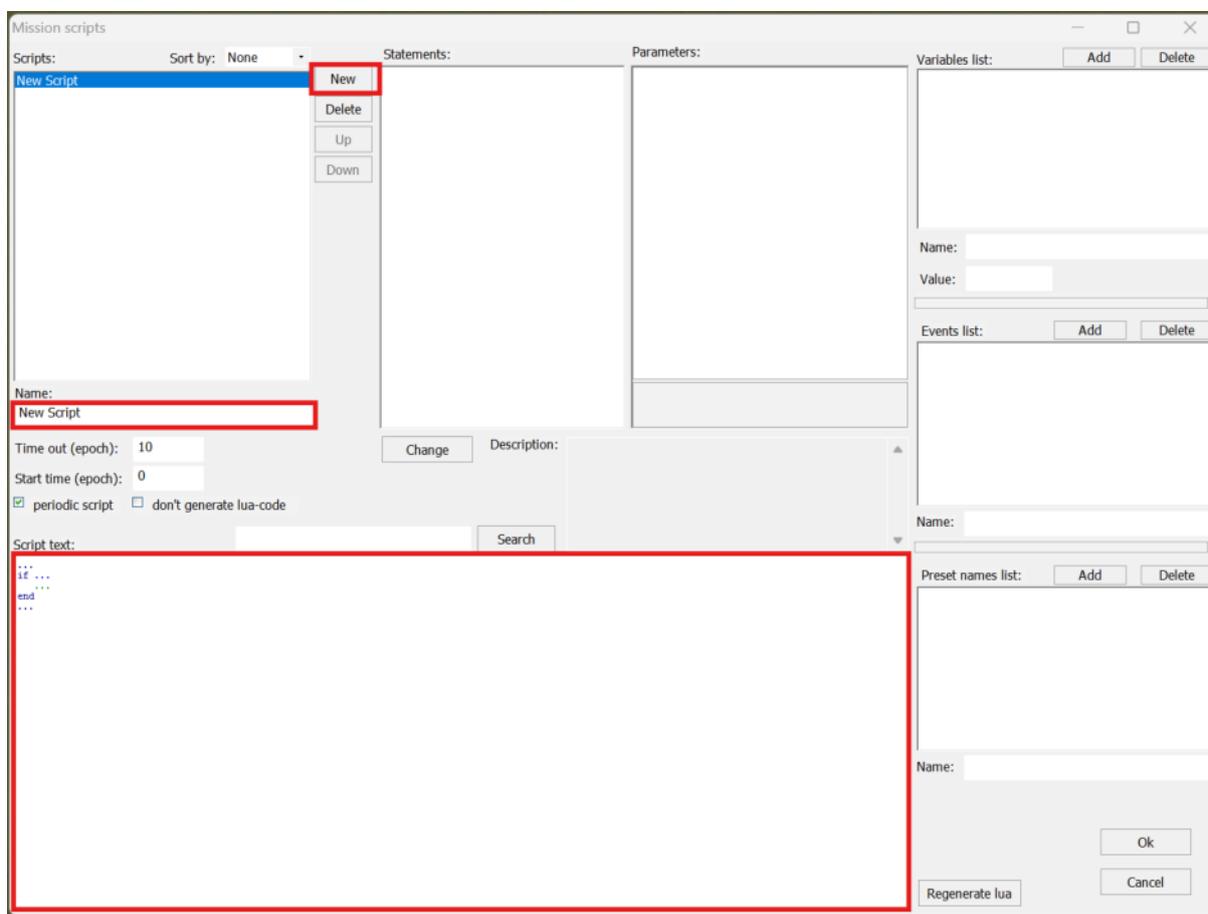
The game will now be able to tell the player details about the objective, and its location, and it will show a ticking timer at the location when the player is completing it. However, it would not be possible to complete it yet, because you have not created a script for this.

### Script editor

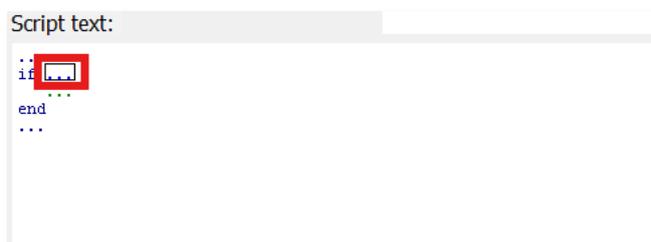
Open "Mission Scripts" at the Tools option on the top bar.



Click "New" to begin. With your new script created, change the Name of the script to "Objectives" by editing the indicated box.



At the Script text box, click the area "..." to begin writing the "if" statement for your first objective.

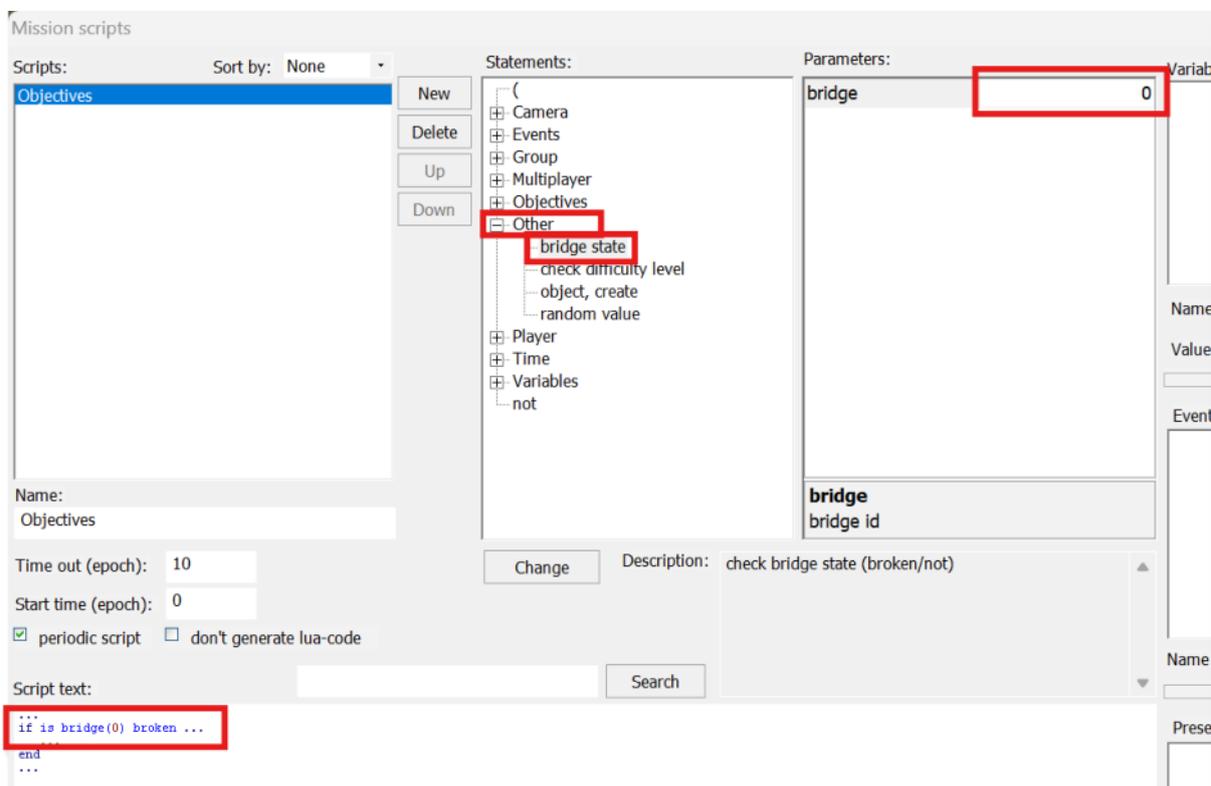


As an objective, you can set up destroying the bridge as an example.

### Destroy the bridge

This is an example objective that can be set up by knowing the ID of the bridge in the scenario.

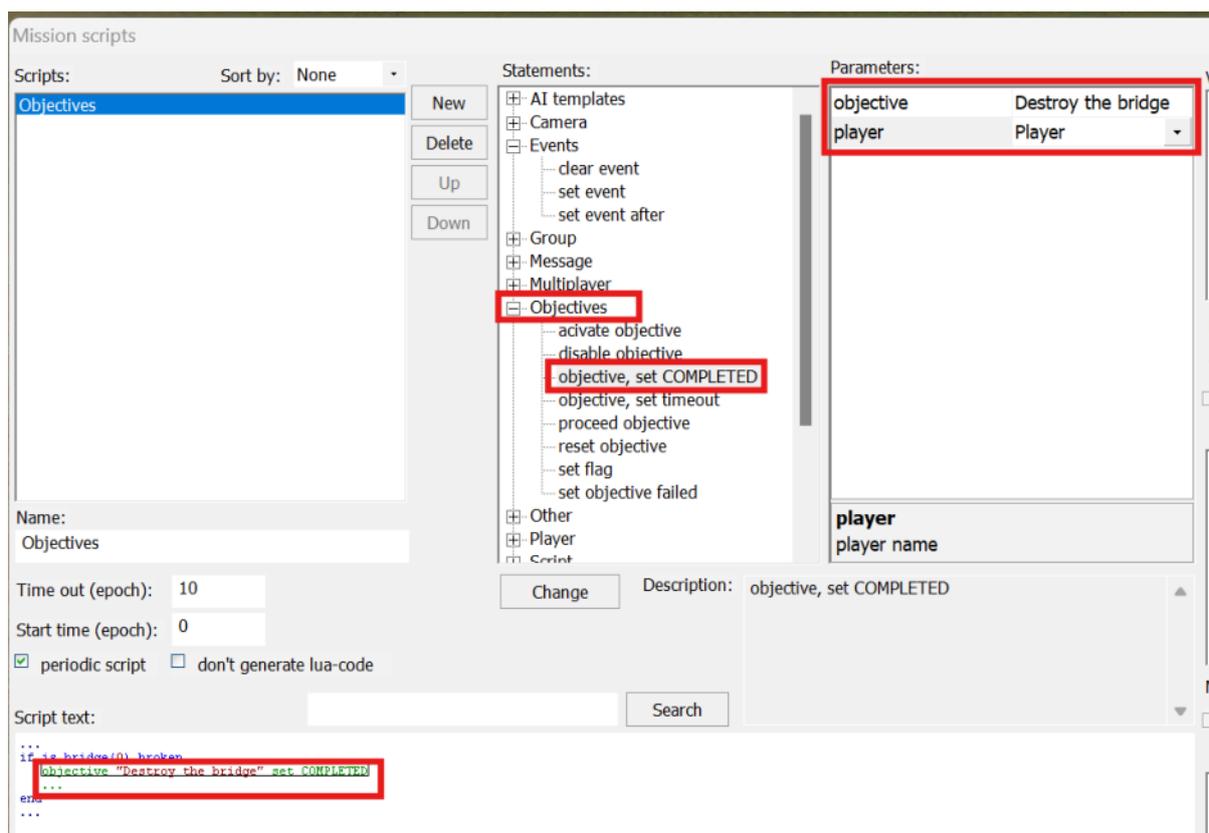
Before you get the ID, you can set up the statement at Mission scripts.



Having previously clicked the "..." after "if", expand the option "Other" in the list of available statements. Double-click "bridge state".

By default, the bridge ID will now be 0 (see Parameters box) and the script will now be checking if it is broken.

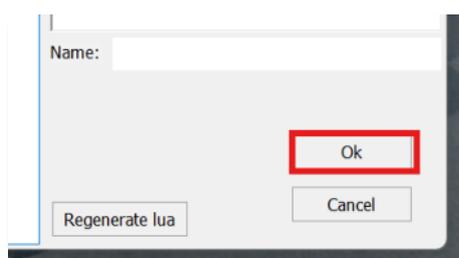
Click the three green dots "..." under the "if" statement at the Script text. Then expand the "Objectives" option at the Statements options and choose "objective, set COMPLETED". At the Parameters box, choose Destroy the bridge as the objective and Player as player.



The script will now have written itself to complete an objective when a bridge is destroyed, but the bridge ID is not present yet, so the objective will still not work.

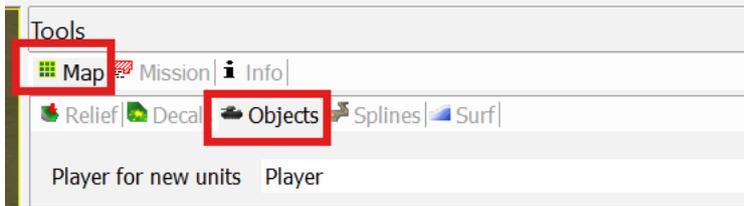
To get the bridge ID, you need to exit Mission Scripts after saving what you have written.

Click "OK" in the lower right corner to keep the changes and exit.

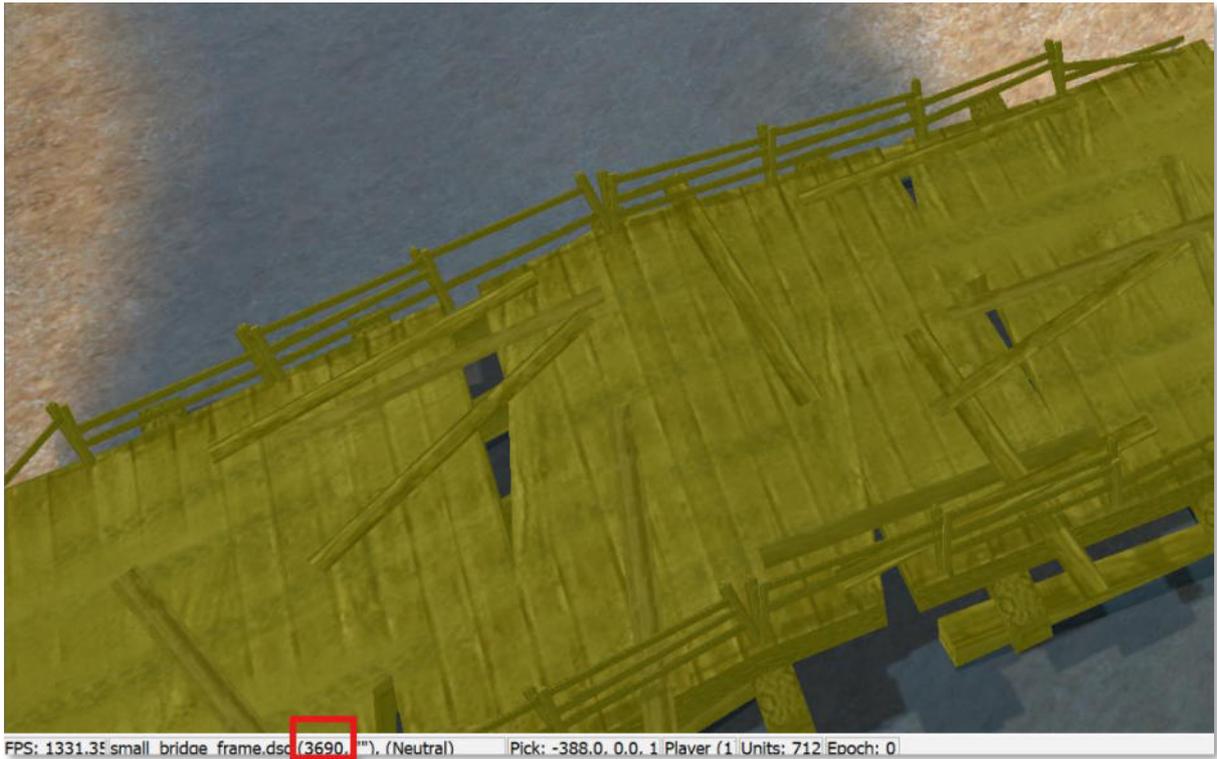


Scripts are not saved unless you save the entire scenario. If you exit the scenario after you have only clicked OK, the changes to the scripts will be lost.

Now move to the bridge and highlight the object in the editor. You would need to return to Objects mode under the Map tab at the Tools panel to select or highlight objects like this.

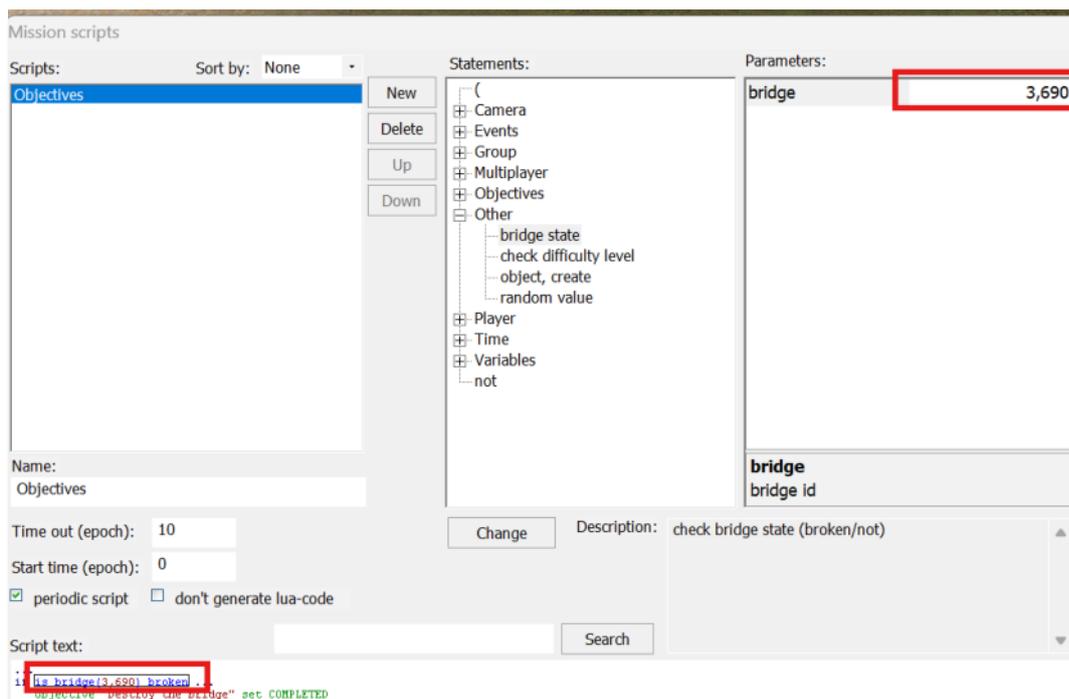


With the object highlighted, read the ID of the object. It is the first numerical value that appears in brackets after the dsc file name on the bottom bar. Take note of this ID.



Now we need to use this ID. Go back to Mission scripts (top bar, Tools, Mission scripts) as before.

You can set the ID of the bridge by selecting the "if" statement part reading "is bridge(0..." and changing the number 0 to the correct bridge ID at the Parameters. The script will adjust to use this ID.



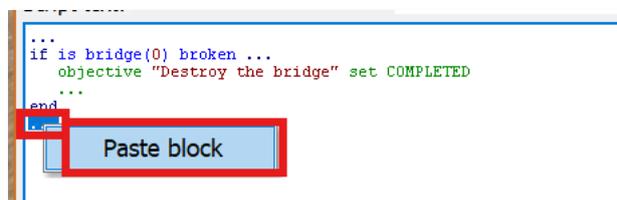
### Resetting objectives

To reset an objective such as the bridge if it has been rebuilt, you would need another "if" statement asking if the bridge is intact.

To do this, first copy what you already wrote for this objective.

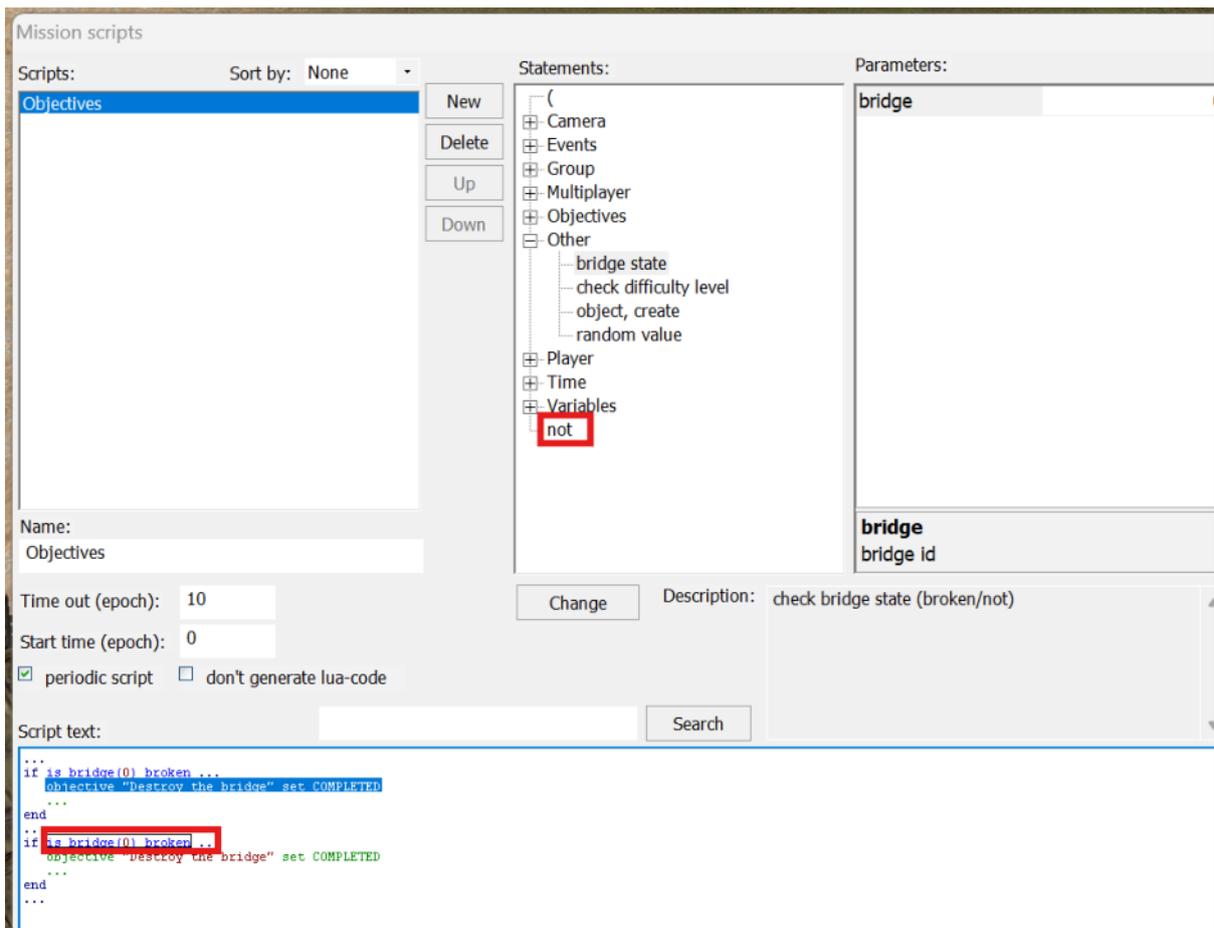


Then, at the three dots following where it says "end", click and then right click. Following this, select the paste block option to make another block that you can change.



The ability to paste scripts like this is especially useful because you can copy the scripts from one scenario to another scenario and then change them to fit your scenario.

We want to negate the objective so that it will reset if the bridge is no longer destroyed. To do this, click the duplicated statement reading "is bridge(0) broken". Then double-click "not" from the Statements options.



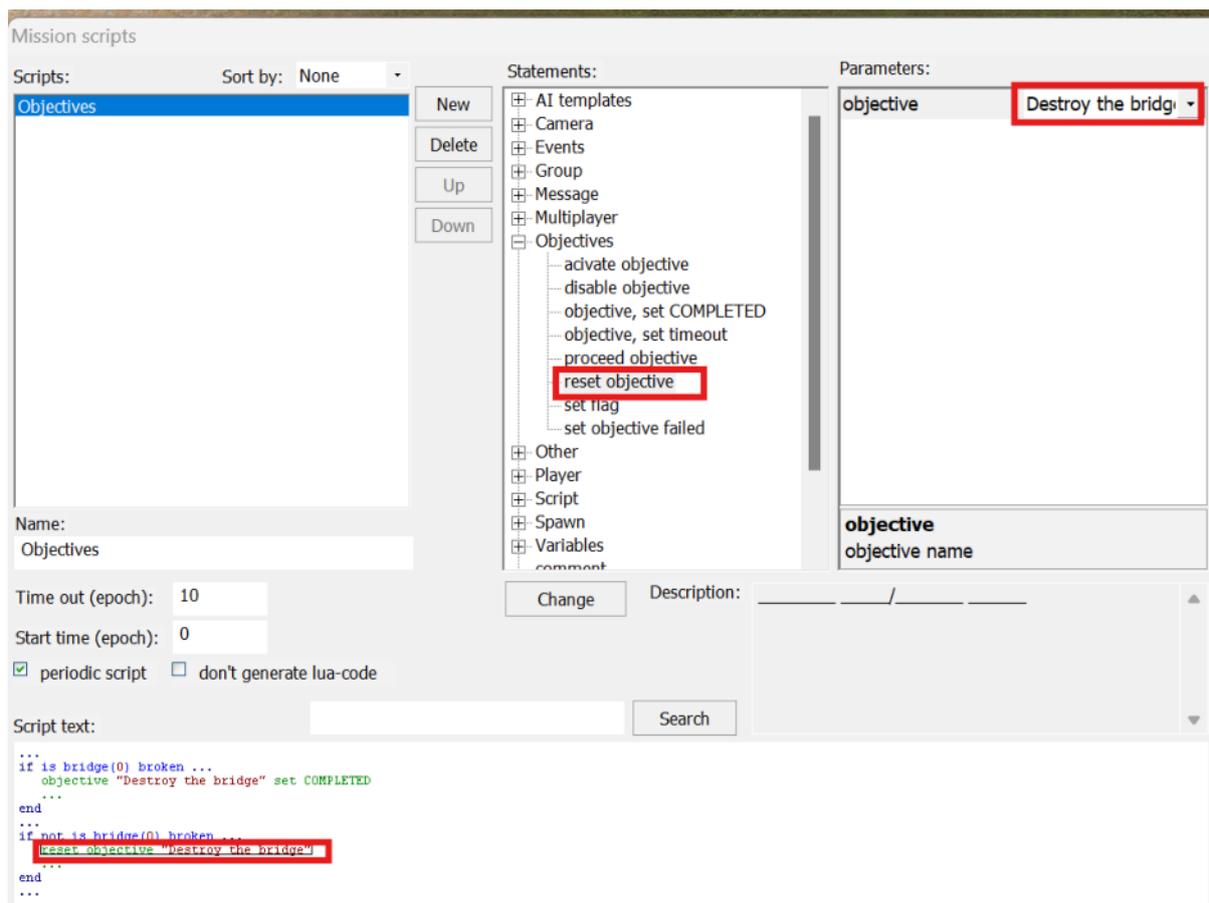
The second block will now have changed to ask if "not" the bridge is broken, in other words, now asking if the bridge is intact.

```

...
...
if is bridge(0) broken ...
  objective "Destroy the bridge" set COMPLETED
...
end
...
if not is bridge(0) broken ...
  objective "Destroy the bridge" set COMPLETED
...
end
...

```

Now click the line under it to select that, and double-click the Statement "reset objective" to change your line to this. Choose "Destroy the bridge" to reset this objective.

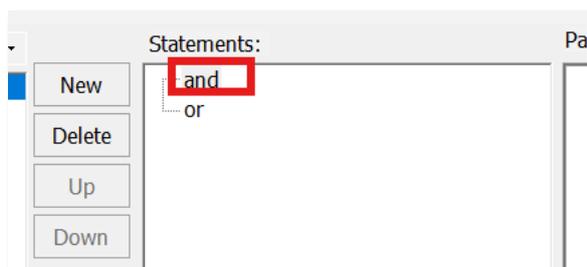


However, the bridge is already intact by default, so the objective would keep resetting every second. Instead, more conditions should be added.

To add an extra condition, click the three dots after the statement about the bridge not being broken.

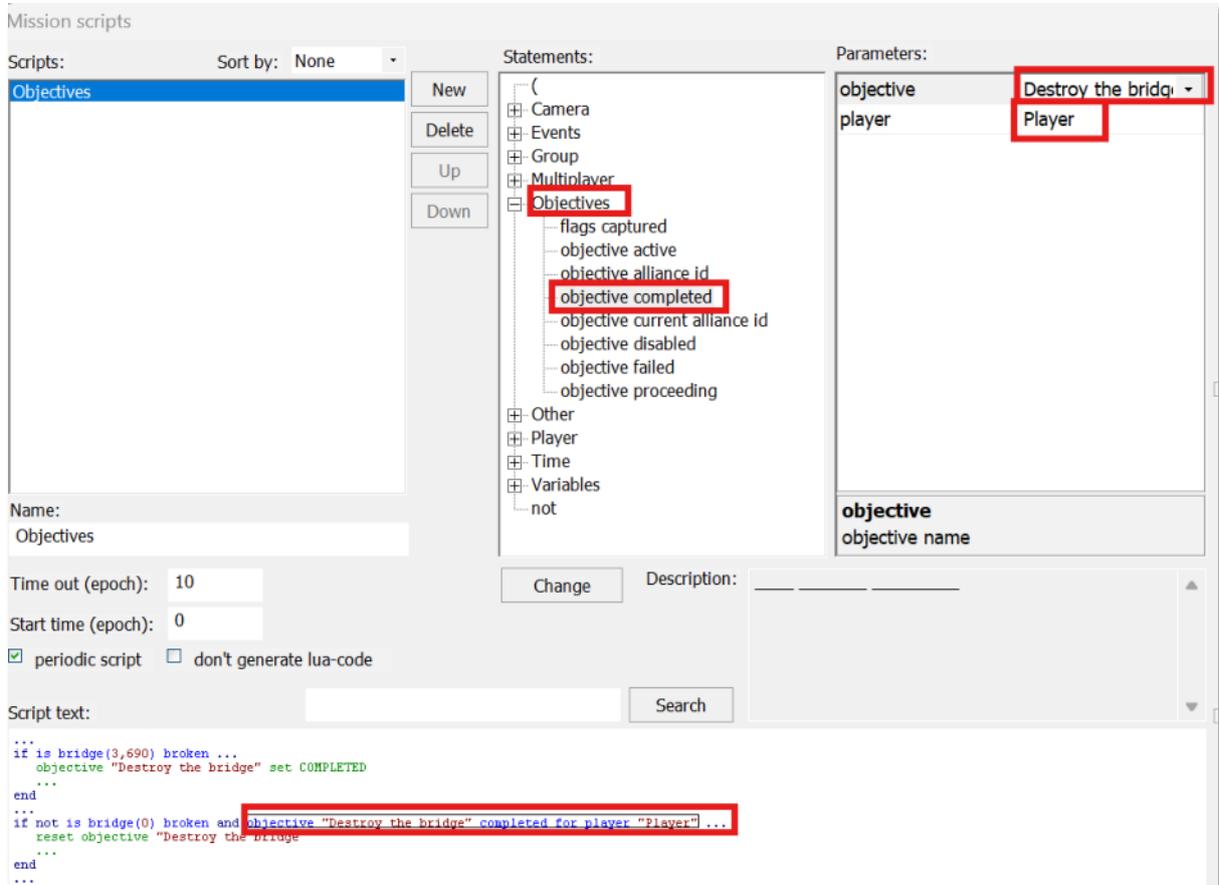


Then choose "and" at the Statements available by double-clicking it.



At the other three dots following the "and" at the Script text, click.

Now choose Objective completed under the Objectives option at the Statements box, and choose the bridge objective and the player again as parameters.



This part of the script now reads that if the bridge is not broken, and the player has completed the objective to destroy it, the objective will reset so that the player has no longer completed it.

Resetting objectives like this can be useful for scenarios where the player must maintain an objective throughout the mission until they have completed all their objectives.

### Enemy objectives and flags

What is done above might be satisfactory for some mappers, but there is a better way, where objectives are more like flags that change to the player or the enemy.

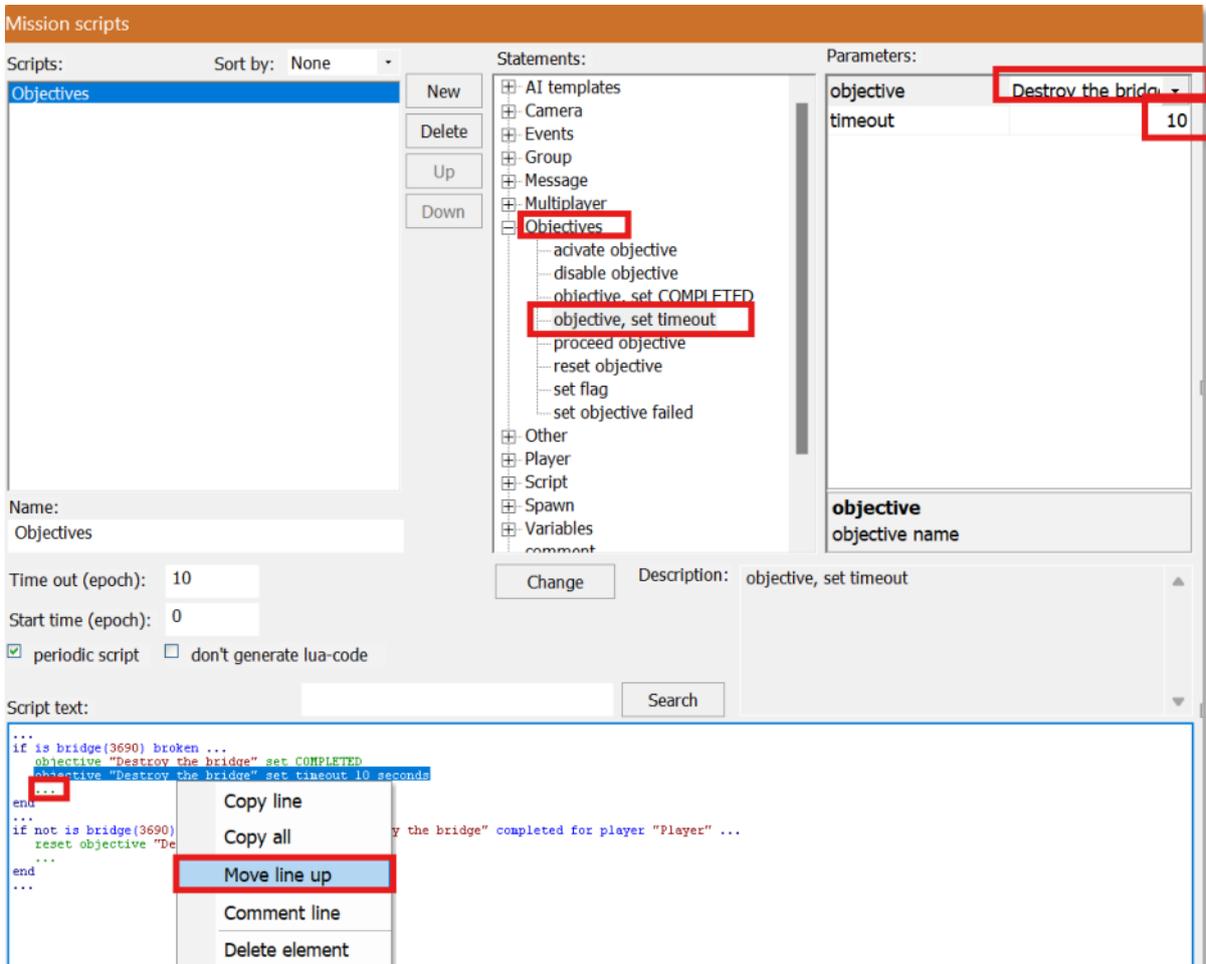
We want to set the objectives to work like flags in multiplayer. This way, they are flashing in red, showing enemy control, until the player completes the objective and they change to green.

To do this, the objective must be set as completed also for the enemy when the player has not achieved them, and the timeout should ideally be set to a significantly long time so the flashing will last a long time, reminding the player to secure the objective.

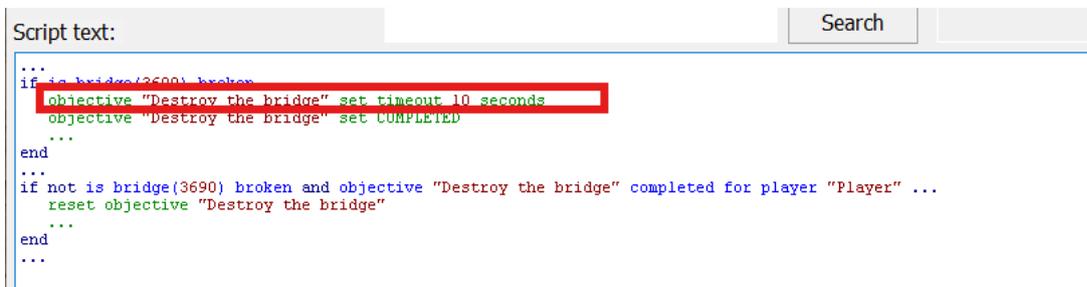
Choose OK to exit the Mission scripts and go back to your Objective on the Tools panel as explained already in the manual. Set the Timeout to 0 there.

Return to Mission scripts as before, and click the three green dots at the end of the first block of script where you set the objective to be Completed for the player.

Here, choose Objectives and then double-click "objective, set timeout" to add this to the block. Set the objective parameter to your objective, and the timeout to 10. Now move the timeout line up by right-clicking it in the Script text and choosing Move line up.



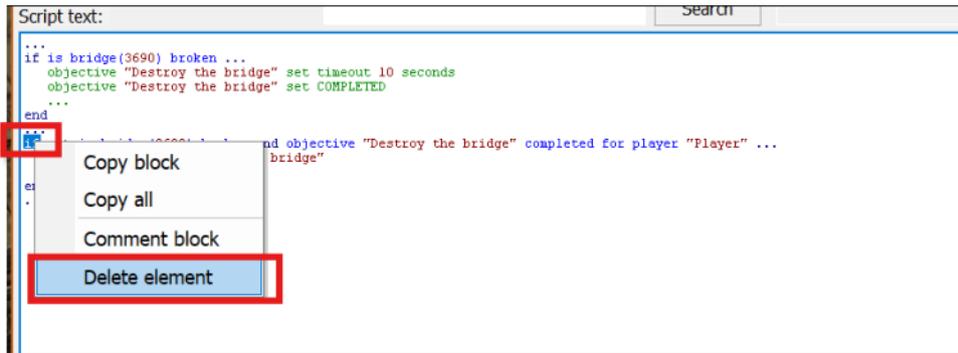
This will set the timeout to 10 seconds before the player completes the objective, and the timer will appear to the player on the map, too.



Now, we need to set the objective so that the enemy is completing it as long as the player is not completing it.

For this, we no longer need the objective to reset.

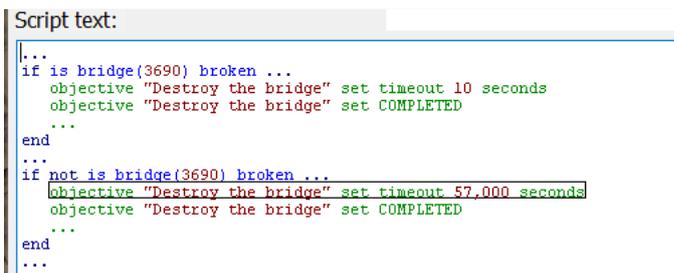
Delete the entire second block of script. Do this by right clicking the "if" statement at the start and then choosing "Delete element"



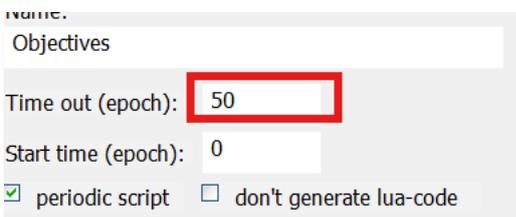
Now copy and paste the first block again, as you did before (at the three dots after "end").

As before, at the pasted block, negate the part of the statement asking if the bridge is broken by selecting it and then double-clicking "not" from the list of available statements.

However, after doing this, click the line underneath the "if" statement and edit the timeout value to 57000 seconds via the Parameters box, and also change player to Enemy via the Parameters box after selecting the "objective "x" set to COMPLETED" line.



Also, change the Timeout (epoch) to 50. This will make the game check the objectives every 5 seconds instead of every second, possibly lessening the burden on the computer as it checks the objectives less frequently.



Now, as soon as the mission begins, the timer of the objective will be set to a very long time so that it is flashing when the player is trying to complete it. It is also captured by the enemy at the start of the mission by default.

The enemy-held objective will eventually stop flashing after 57000 seconds, but this still means it will flash for many hours, and the player will either complete the mission or be well aware of the objective by the time it stops flashing.

Click OK, save your mission, place some strong units that can destroy the bridge if you want, and then test the mission to see what happens!

Missions can always be accessed and played even when they are incomplete, via Custom missions inside the game.

### Player captures the area

See the altered script below as an example of how you might set up capturing the Bridge (or any area of the map), rather than destroying the Bridge.

Here, the objective name has been changed at the Tools panel. Then, instead of going to "Other" at the list of available statements to find bridge state, we check "is player's unit present", then we use an "and" statement added and then negated with "not" to check enemy units are not also present within 70 meters of the point. The same is then done for the enemy to capture the bridge by changing "player" to "enemy" and "enemy" to "player" in the parameters at each statement in the second block, and again adjusting the timeout to 57000 seconds for the enemy to make the flag flash.

The screenshot shows the 'Mission scripts' editor. The 'Statements' list on the left has 'Player' selected, and 'is player's unit present' is highlighted. The 'Parameters' table on the right shows the following values:

Parameter	Value
player	Player
point	Bridge
radius	70

The 'Script text' area contains the following Lua code:

```

...
if [is units of "Player" near point "Bridge"] and not is units of "Enemy" near point "Bridge" ...
    objective "Capture the bridge" set timeout 10 seconds
    objective "Capture the bridge" set COMPLETED
...
end
...
if is units of "Enemy" near point "Bridge" and not is units of "Player" near point "Bridge" ...
    objective "Capture the bridge" set timeout 57000 seconds
    objective "Capture the bridge" set COMPLETED
...
end
...

```

You can use the above information to set up flags for any location in the map. It can also be used if the player is supposed to hold a location against the enemy and retake it if it has fallen.

Be aware that enemies hidden in buildings inside an area will count as still being in the area of the objective, possibly preventing the objective from being met when you expected.

Also, enemy airfields cannot be destroyed and still count as a unit in the game. This can prevent capturing an objective if the objective required all enemy units to be destroyed in the area.

Alternatively, you can check "zones" and define if you consider the objective completed if a certain number of your units are in the zone and the zone is clear of enemies. This can allow more specific areas to be checked for capture.

Setting up AI zones and checking numbers of units present inside them, is explained later on in the manual.

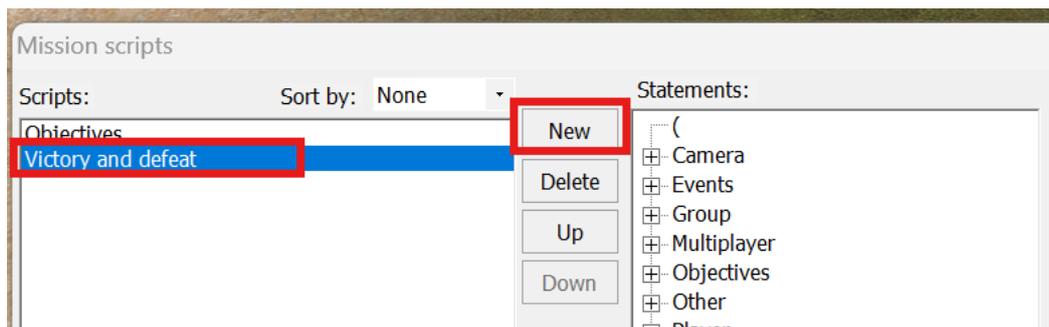
## Victory and defeat conditions

### All objectives met means victory

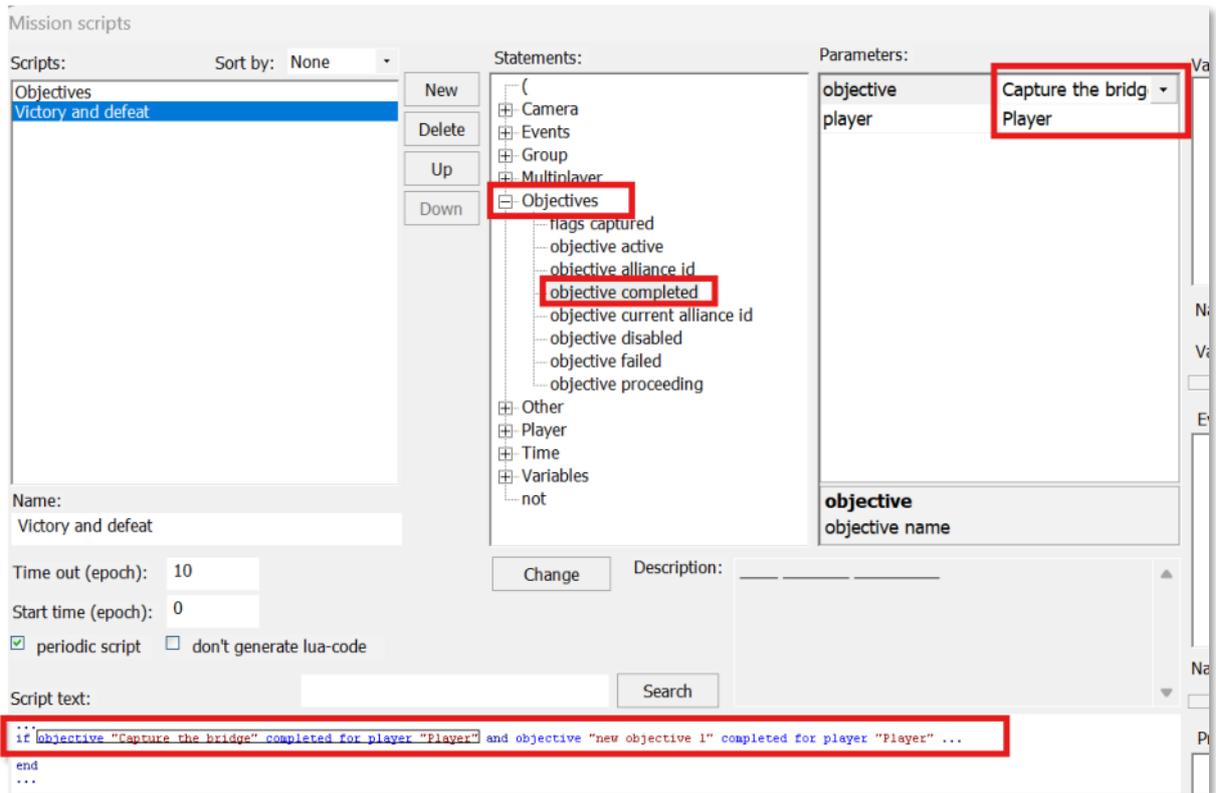
To test out victory conditions, make a second objective. It can be exactly like the first example, or a variation on it if you are confident.

After making the objective at the Tools panel, repeat the steps done at the Mission scripts, building additional blocks of script about the other objective just as you did for the first objective.

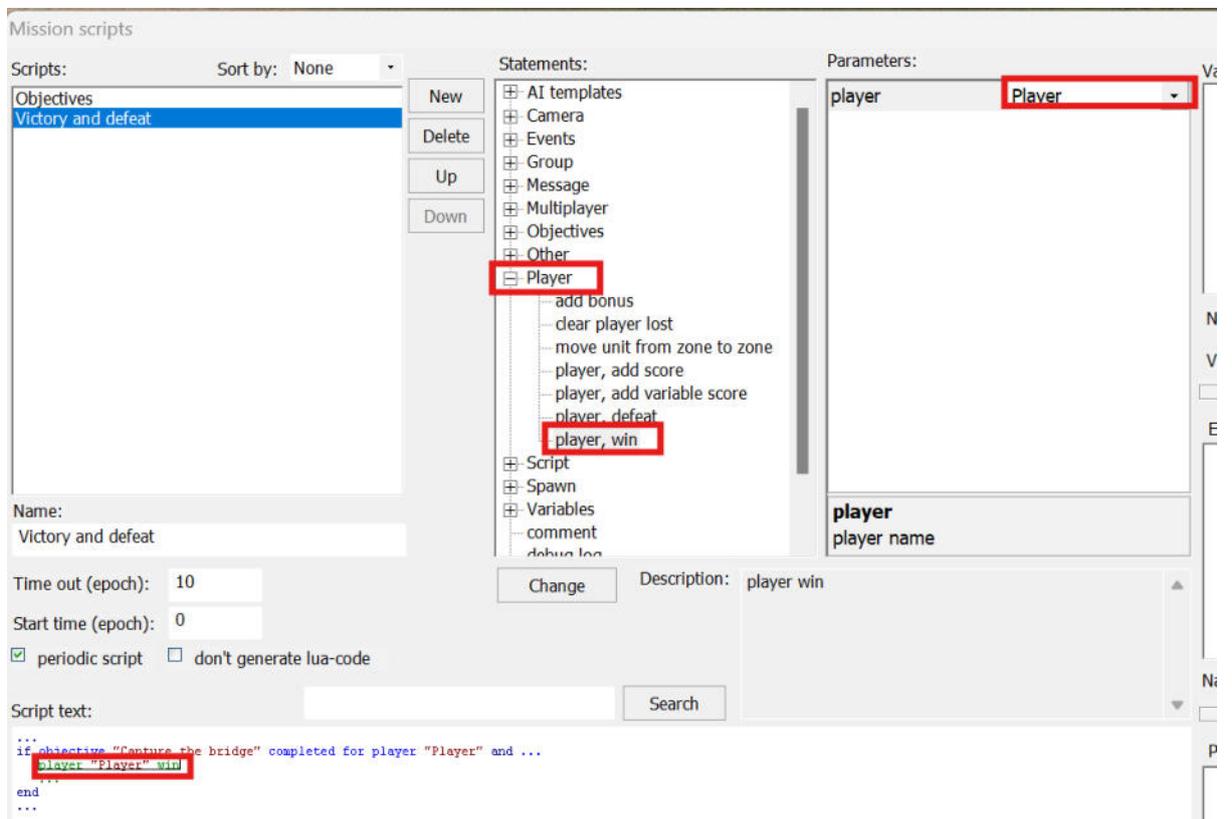
When you have set up two objectives that can be completed separately like the first example, now choose New at the Mission scripts to create another script. Call it "Victory and defeat".



Here, using the same options explained earlier for when writing objectives, write the script to check each objective, with an "And" element placed between them by clicking the three dots and adding this.



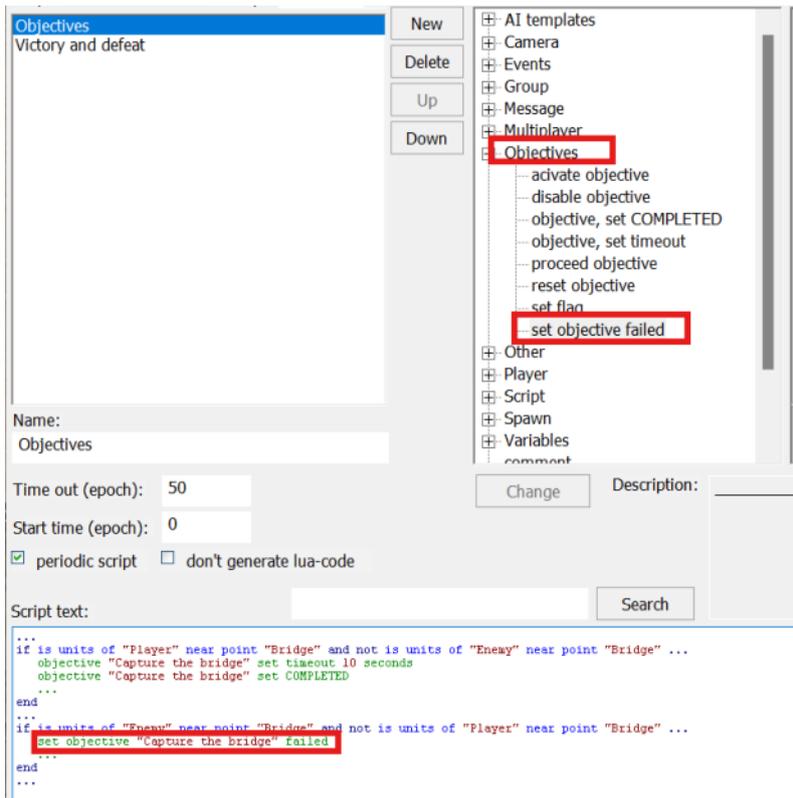
This sets up the condition for multiple objectives to be achieved or controlled, and in this condition you want the player to win. To do that, click the three green dots under the "if" statement, then pick Player, player win, and choose Player at Parameters.



## Objective failed means defeat

To set up defeat due to a critical objective failing, you can edit the objectives script again to delete elements letting the enemy achieve the critical objective and instead replace it with a line for "Objective failed" (applies only to the player) in the block where the enemy objective was to be completed. Set objective accordingly

That critical objective does not need to be a flag like the objectives that are captured and recaptured, because it can only be lost once by the player and the scenario will end.



Once the objective can be set to failed, you can now use this to trigger a defeat at the Victory and defeat script.

An "if" statement would check if the objective has failed in a new block at the Victory and defeat script.

Mission scripts

Scripts: Sort by: None

Objectives  
 Victory and defeat

New  
 Delete  
 Up  
 Down

Statements:

- (
- Camera
- Events
- Group
- Multiplayer
- Objectives
  - flags captured
  - objective active
  - objective alliance id
  - objective completed
  - objective current alliance id
  - objective disabled
  - objective failed
  - objective proceeding
- Other
- Player
- Time
- Variables
- not

Parameters:

objective Capture the bridg  
 player Player

objective  
 objective name

Name: Victory and defeat

Time out (epoch): 10

Start time (epoch): 0

periodic script  don't generate lua-code

Script text: Search

```

...
if objective "Capture the bridge" completed for player "Player" and objective "new objective 1" completed for player "Player" ...
  player "Player" win
...
end
...
if objective "Capture the bridge" failed for player "Player" ...
  player "Player" defeat
...
end
...

```

Then the outcome in the second line in this block of script would be player defeat.

Mission scripts

Scripts: Sort by: None

Objectives  
 Victory and defeat

New  
 Delete  
 Up  
 Down

Statements:

- AI templates
- Camera
- Events
- Group
- Message
- Multiplayer
- Objectives
- Other
- Player
  - add bonus
  - clear player lost
  - move unit from zone to zone
  - player, add score
  - player, add variable score
  - player, defeat
  - player, win
- Script
- Spawn
- Variables
- comment
- debug log

Parameters:

player Player

player  
 player name

Name: Victory and defeat

Time out (epoch): 10

Start time (epoch): 0

periodic script  don't generate lua-code

Script text: Search

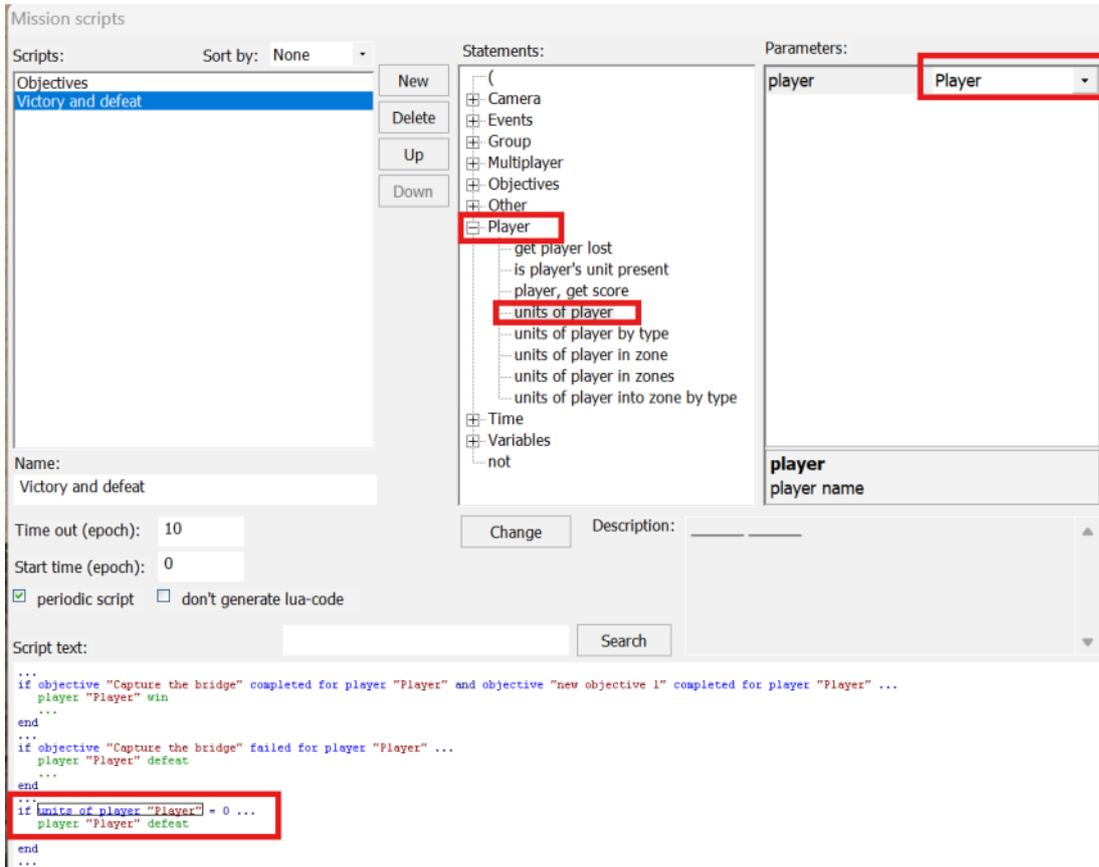
```

...
if objective "Capture the bridge" completed for player "Player" and objective "new objective 1" completed for player "Player" ...
  player "Player" win
...
end
...
if objective "Capture the bridge" failed for player "Player" ...
  player "Player" defeat
...
end
...

```

## Player units destroyed means defeat

For this, you would just check units of player equal zero, and then set player defeat just as with the other example.



Finally, change Time out (epoch) to something different than the Objectives, maybe something longer, like 120 (12 seconds). If you want the player to linger for a while after achieving their objectives rather than ending the mission, make it longer.

Always check your objectives and victory triggers are working before doing anything else to make the mission. Place units in the map (just as you placed objects but choose units) and try to destroy the objectives or capture the flags before you make the next parts of the mission. See if you get the victory screen. Do the same to check if you can be defeated if the enemy destroys you or occupies the objectives.

Otherwise, you risk creating a complex scenario that takes many hours but cannot be completed anyway (this even happened with a scenario in "The Last Stand" and possibly other missions released by the creators of the game!)

## Reinforcement spawn locations

When you are ready, click the "Edit spawns" button on the top bar.



Select which player you want to give a spawn location for units, then add a spawn (these must be noted because you will need them for writing the scripts later).

Vectors show the individual paths that units will try to follow when they spawn into the map. They do not have to be very long if the unit will just deploy at the map edge and wait for orders.

Adjust the arrows for the vectors to indicate where units spawn.

You may need to return to edit the spawn vectors again, if the units are not spawning properly in the map because they are stuck off the map or their spawn position was taken by a tree or other object. It is worth testing reinforcement spawn vectors in the game, after making the scripts that will be explained later.



Choose "Apply" when you are satisfied.

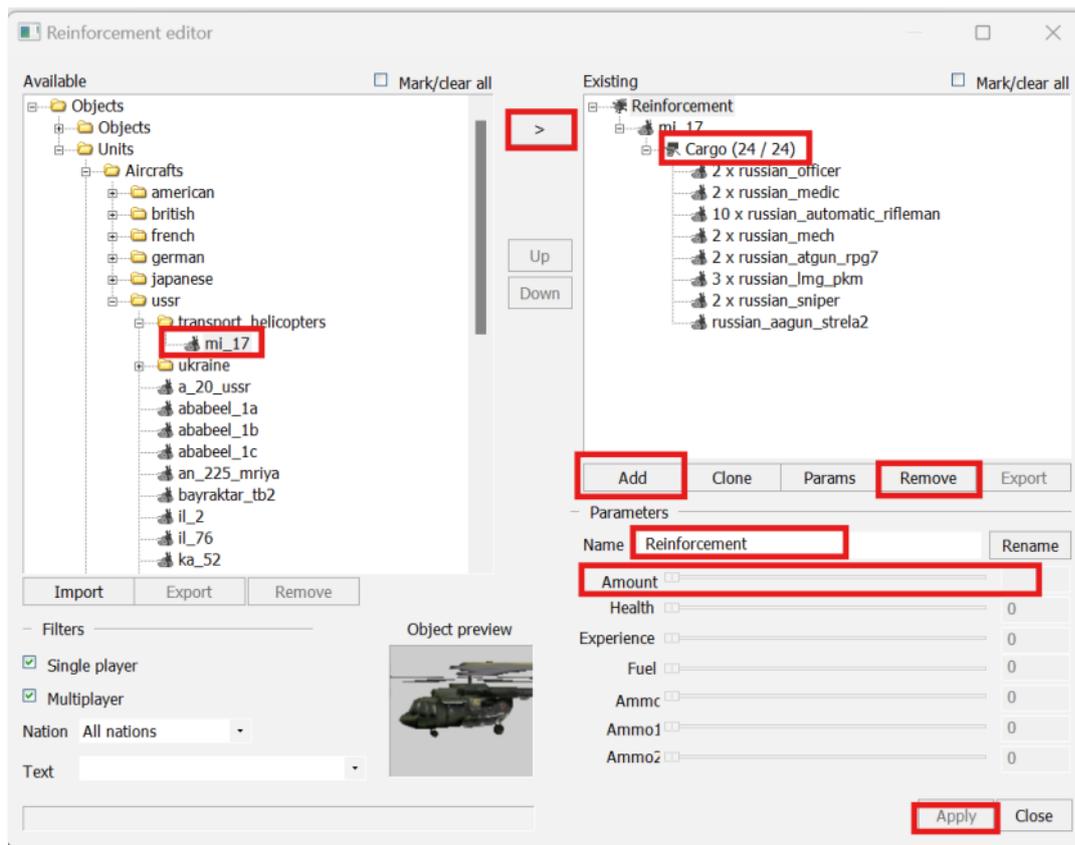
## Reinforcement packages

Now click the Reinforcements button also on the top bar.

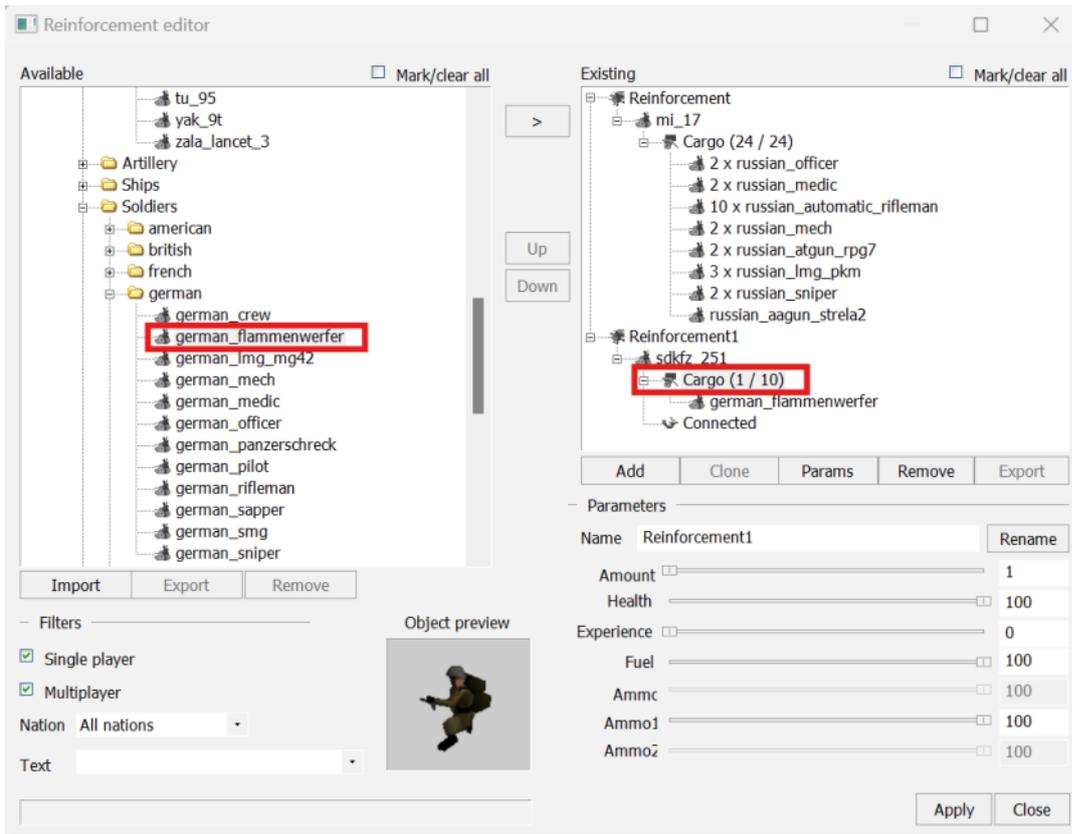


On the Reinforcement editor, you can make packages of reinforcements, grouping them together and naming them.

You can for example click "Add" under the "Existing" list at the right, and then name the reinforcement "heavy tanks", navigate the "Available" folders and files on the left to find different types of Tiger tank, and double click a unit name to add one of them to your reinforcement package. The ">" button will also add them to the reinforcement package if they are highlighted.



With a reinforcement unit chosen and then selected at the right, you can adjust the "amount" slider to increase the number of units of one type included in the reinforcement package. You can also add passengers to transports or APCs by clicking "Cargo" where it is indicated how many are present in the unit, then navigate your way to soldiers in the Available folders, and double click your preferred soldier type. You can also increase the Amount slider while having the passenger selected at the Existing list, to put more soldiers of the same kind inside the vehicle.



Select "Apply" when you have created sufficient varieties of reinforcement packages for both sides in the scenario, not forgetting that any army needs some fuel tankers, ammunition trucks, and plenty of infantry.

Clearly Name the reinforcements to show their nation and the quantity, as this can help you decide what should spawn later when writing the scripts for reinforcements. For example, "RUSSIAN Squad x12 men"

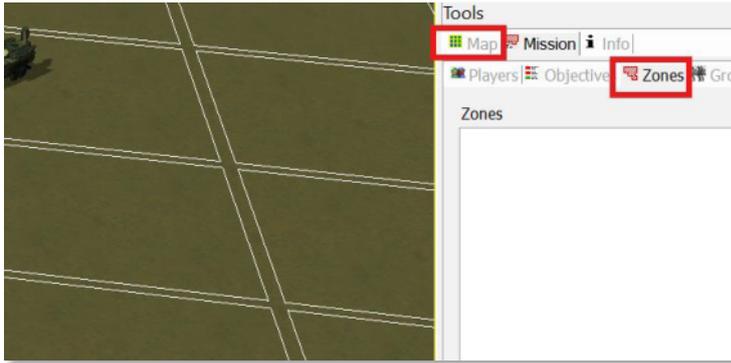
You do not want to call massive reinforcements in or tiny reinforcements when it is not appropriate to the situation, so you need to see what you are calling in.

You also do not want to call in some reinforcements for the wrong player and nation, which the developers did by mistake in one of the original missions! By labelling a squad clearly as Russian, you will not accidentally set it to spawn as a German reinforcement in the scripts.

## AI zones

AI zones can assist with triggering events in the scripts. They are good for troop movements and for telling if objectives are captured or overrun by enemy forces entering them.

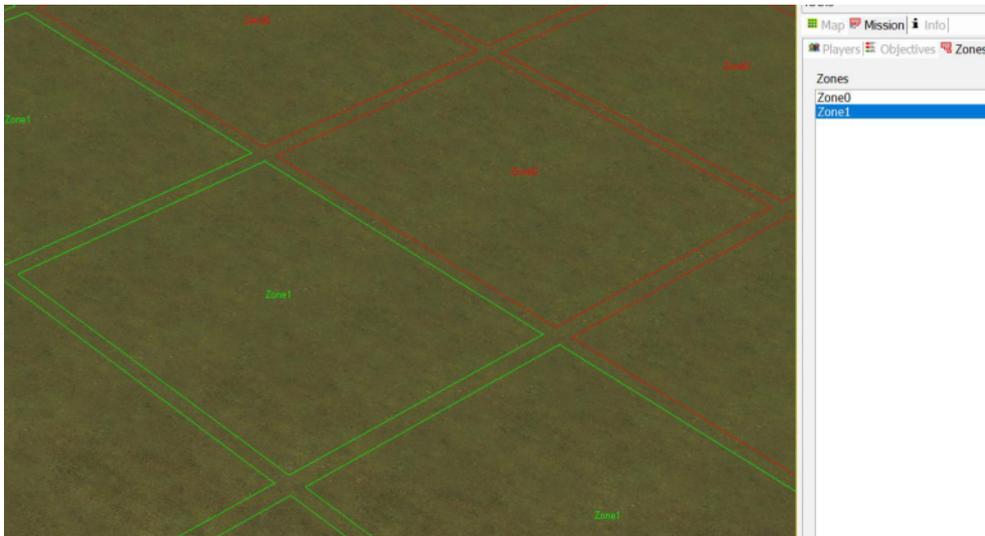
Select Map tab and Zones at the Tools window for this option. This will make them visible and let you place new zones.



Click "Add" to make a new zone. Rename it if you want, although for the next step it is not important. Click parts of the map to add to the zone.

We just want a series of zones for enemy forces to capture in order, as they try to capture the map.

Zone0 can just be where the enemy reinforcements spawn. Zone1 can be right next to it, where you want their troops to pool when they arrive.



Now, put Zone2 at the next area you would like the enemy forces to stop as they try to reach their objective. This might be the kind of distance that you would order your troops to advance before you reorganise them and try to regroup them. It is up to you how far apart you want the zones to be.

The zones should cover the areas of the objectives, so that the enemy can capture them, starting with the areas more important or near the enemy. A total of 6 zones would be normal, with the last one intended to overrun the player's reinforcements or base and defeat the player.

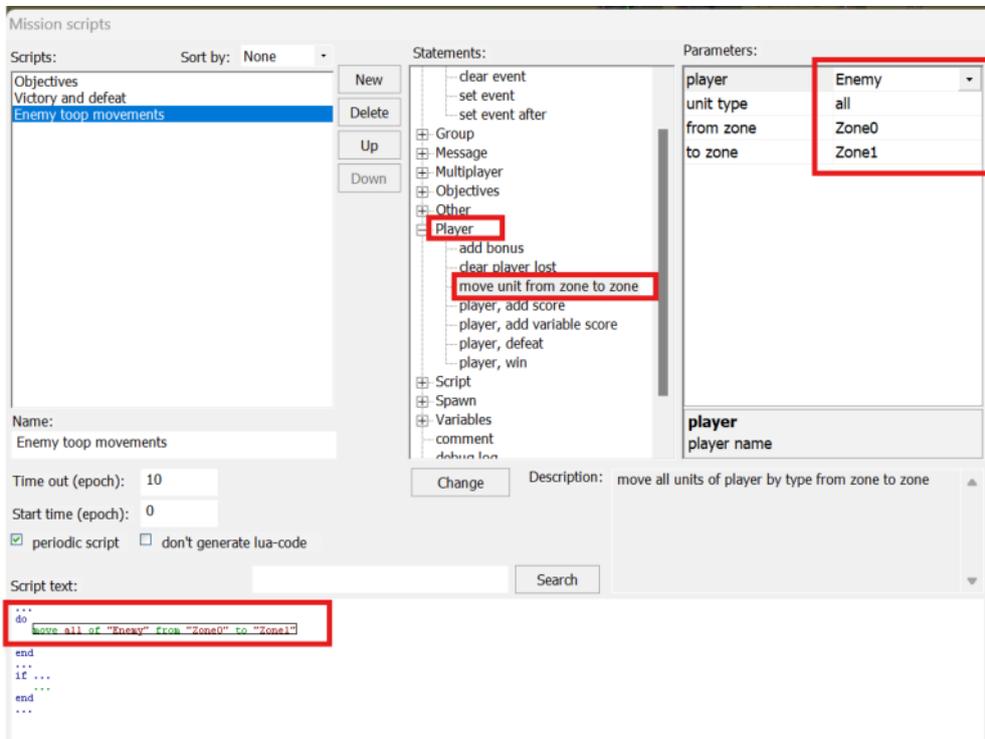
### **AI zone to zone troop movements**

This is the best method when the enemy is carrying out an offensive against the player. Enemy forces will wait until they are numerous enough to attack, then try to optimally scout and swarm the objectives with infantry, followed up by tanks, and then supply units and escorts will follow cautiously behind with the hope to supply them.

Go to Mission scripts again. Make a new script.

Clicking the top three dots at the script text, start a "do" statement by just double-clicking "do" from the available statements box.

At the next line, click the three green dots to make a statement of what to do. Choose Player, "move unit from zone to zone" and adjust the parameters to move enemy units of all types from zone 0 to zone 1.



This is not a proper troop movement for an offensive, but just to make the enemy reinforcements not pool in the same locations, instead spreading out a little as they arrive by moving to zone 1.

Now, at the first "if" statement, click the three blue dots and make a real troop movement.

Choose the condition for if "units of player in zone" and choose Enemy as player and zone 1 as the zone. Click the "=" symbol and change that to ">" (greater than), and click and change "0" to the parameter for how many units the enemy should accumulate in the zone before moving to the next zone (30 for example). You might want this to increase as the enemy gets nearer to the objective, or keep it the same if you want rapid, small attacks by forces of similar size and concentration at all stages of the offensive. Keep in mind that even a group of fewer than 100 troops supported by vehicles is a lot for one attack.

Mission scripts

Scripts: Sort by: None

Objectives  
Victory and defeat  
Enemy troop movements

New  
Delete  
Up  
Down

Statements:

- (
- Camera
- Events
- Group
- Multiplayer
- Objectives
- Other
- Player
  - get player lost
  - is player's unit present
  - player, get score
  - units of player
  - units of player by type
  - units of player in zone
  - units of player in zones
  - units of player into zone by type
- Time
- Variables
- not

Parameters:

player	Enemy
zone	Zone1

Name: Enemy troop movements

Time out (epoch): 10

Start time (epoch): 0

periodic script  don't generate lua-code

Script text:

```

...
do
  move all of "Enemy" from "Zone0" to "Zone1"
  ...
end
...
if units of player "Enemy" are in zone "Zone1" > 30 ...
...
end
...

```

When this condition is set up, click the three green dots below to set up what will happen. Copy and paste or select same option as the previous block, but change to move enemy from zone 1 to zone 2 instead this time. In addition, change "all" to "soldier" so only soldiers are ordered at first.

Parameters:

player	Enemy
unit type	soldier

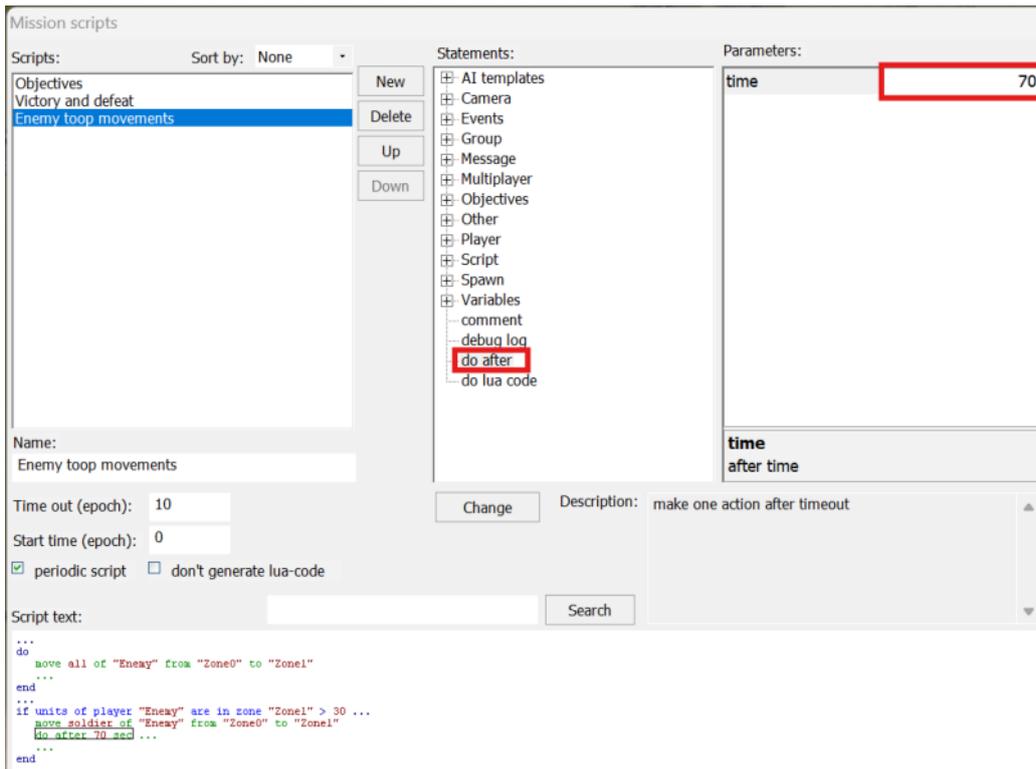
Script text:

```

...
do
  move all of "Enemy" from "Zone0" to "Zone1"
  ...
end
...
if units of player "Enemy" are in zone "Zone1" > 30 ...
  move soldier of "Enemy" from "Zone1" to "Zone2"
  ...
end
...

```

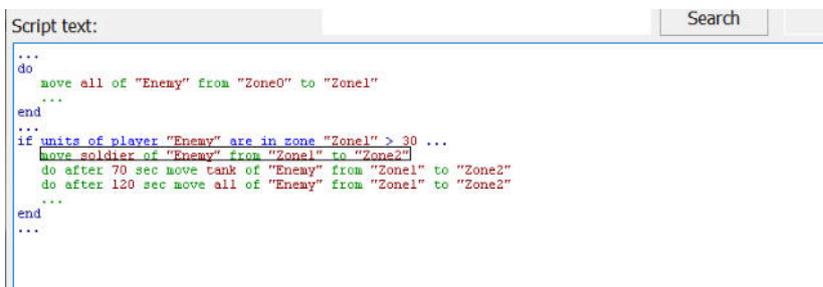
Then, at the next three green dots below this line, click and choose a "do after" statement to introduce a delay at this line. Type 70 as the parameter for time to add a 70 second delay while the infantry fan out and observe ahead with their better vision ranges in the game, before the tanks move.



Copy the line for “move soldier...” and then paste it to the three green dots on the same line but after “do after 70 sec”, so the order would be repeated after the 70 seconds. However, click this element you have pasted, and change the unit type to tank.

This means tanks will start moving from the zone after the infantry have fanned out and moved to scout the area ahead in this advance.

Now right click on this latest line, select “copy line”, and paste the line again at the three green dots below it so it occurs again. However, click the “do after” element on it and change the parameter to 120 seconds. Then change it to move “all” units as you did in the first example. Avoid using “all” and specify exact types if the zone might contain trains, as they will become confused if they are made to advance too!



This means all remaining units at zone 1 such as supply units, APCs, etc, will follow last, after 120 seconds have passed.

Troop movements should occur every 45 seconds or 450 epochs, and may not need to begin even hundreds of seconds/thousands of epochs from the start of a scenario (edit Start time for this) depending on the enemy AI’s plans.

Time out (epoch):	450
Start time (epoch):	650
<input checked="" type="checkbox"/> periodic script	<input type="checkbox"/> don't generate lua-code

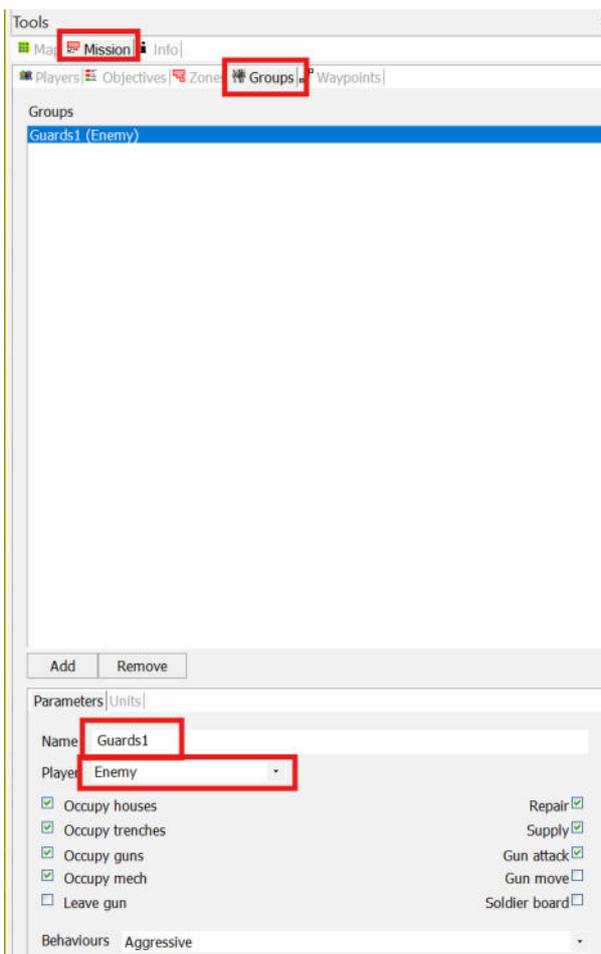
You can also change the delays and timers and add additional lines to be more specific about different unit types, e.g. APCs, AA and others, if you prefer. You could try adding more to these scripts to delay attacks or require more troops before attacking under other circumstances, like the player's own numbers of units or completed objectives.

All the troop movements from one zone to the next will be like this. Repeat this with 'zone 2 to zone 3', 'zone 3 to zone 4', etc. until the enemy can reach all the target zones. Just copy the block and change the zone names if you prefer.

## AI guard groups and zones

You can set up enemy infantry to fan out even further and try to take some other good locations on the map, like high ground, buildings and woods not completely essential to the objectives. Vehicles can be included with them if you want, as you can choose what units are donated to these groups.

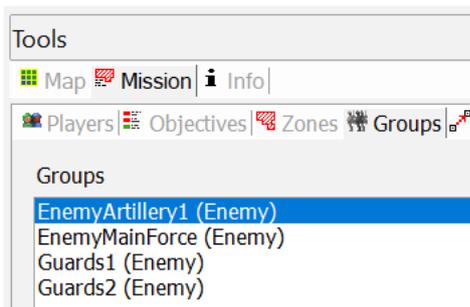
For this, we must create groups. Exit the Mission scripts. Go to Groups after choosing the Mission tab at the Tools panel.



Set the name as Guards1 and set the desired default behaviours for the units assigned to guard secondary areas of interest to the enemy AI. Aggressive will support them taking more independent action, chasing the player's units and standing up to be able to observe further. This can be effective for some ambushes you want the enemy to stage and create a sense of further depth in the mission, as if the enemy soldiers are getting special orders in other parts of a map.

Make a second group called Guards2, for the enemy to assign to more areas of interest to their plans. If you want, give it different behaviour and default settings.

Make a third group called EnemyMainForce. This will be the group all the enemy reinforcements belong to when arriving into the map, which can donate units to the guard groups.



You can also make an artillery group or two here for the enemy, if you intend the enemy to use artillery units to blindly shell areas of the map later when the player crosses them.

Before going to scripts, you need to set the areas where the guards should go on the map. Change tab to "Zones" at the "Mission" mode on the Tools panel and place two zones (see "AI zones" for instructions again), but name the zones GuardZone1 and GuardZone2. Place the GuardZone1 squares in whatever supplemental areas you think are worth the enemy securing, with GuardZone1 being closer to the core enemy defences and GuardZone2 being further spread out to guard the enemy flanks and ambush the player. It is okay for the same zone to have squares in completely different parts of the map, as enemy forces will split up to guard them.

Now go into Mission scripts. Go into the Enemy troop movements script.

Build an additional block with an "if" statement as follows (click the three dots after "End" to get that option as a statement). We want the "if" statement to check if "units in group" exceed 250 (see "Group" and "units in group" to find this type of statement). After double clicking the statement to add this type of statement to the script, change the "=" to ">" and change the default "0" to 250 by clicking these and altering the values at the parameters box. Set the "from group" parameter to the enemy main force.

Mission scripts

Scripts: Sort by: None

Objectives  
Victory and defeat  
Enemy toop movements

New  
Delete  
Up  
Down

Statements:

- (
- Camera
- Events
- Group
  - group assault now
  - group move now
  - group see enemy now
  - group units near point
  - units in group
  - units of group in trench
  - units of group in zone
- Multiplayer
- Objectives
- Other
- Player
- Time
- Variables
- not

Parameters:

group	EnemyMainForce
unit type	all
cargo	<input type="checkbox"/> false
crew	<input type="checkbox"/> false
percent	<input type="checkbox"/> false

Name: Enemy toop movements

Time out (epoch): 10  
Start time (epoch): 0  
 periodic script  don't generate lua-code

Script text:

```

...
do
  move all of "Enemy" from "Zone0" to "Zone1"
  ...
end
if units of player "Enemy" are in zone "Zone1" > 30 ...
  move soldier of "Enemy" from "Zone1" to "Zone2"
  do after 70 sec move tank of "Enemy" from "Zone1" to "Zone2"
  do after 120 sec move all of "Enemy" from "Zone1" to "Zone2"
  ...
end
...
if all in group "EnemyMainForce" > 250 ...
end
...

```

We also want to check if the first guard force is smaller than a certain reasonable size, too, and we will have that size be 50 men. To do that, include an "and" statement in the line, and check if units in Guards1 are less than (" $<$ ") 50.

```

if all in group "EnemyMainForce" > 250 and all in group "Guards1" < 50 ...
...
end
...

```

Under these conditions, let us say 12 soldiers should be assigned to Guards1. Click the three green dots under that line, and choose the "Group" option, then double-click "transfer units", and choose the parameters so that the main force will transfer a count of 12 soldiers to the first guards force.

Mission scripts

Scripts: Sort by: None

Objectives  
Victory and defeat  
Enemy toop movements

New  
Delete  
Up  
Down

Statements:

- AI templates
- Camera
- Events
  - clear event
  - set event
  - set event after
- Group**
  - group, attack place
  - group, disappear units
  - group, unload transports
  - move group to
  - move group to point
  - set group behaviour
  - set group microAI behaviour
  - set group units selectable
  - transfer units**
  - update group counters
- Message
- Multiplayer
- Objectives
- Other

Parameters:

from group	EnemyMainForce
unit type	soldier
to group	Guards1
count	12

Name: Enemy toop movements

Time out (epoch): 10

Start time (epoch): 0

periodic script  don't generate lua-code

Script text:

```

...
do
  move all of "Enemy" from "Zone0" to "Zone1"
  ...
end
...
if units of player "Enemy" are in zone "Zone1" > 30 ...
  move soldier of "Enemy" from "Zone1" to "Zone2"
  do after 70 sec move tank of "Enemy" from "Zone1" to "Zone2"
  do after 120 sec move all of "Enemy" from "Zone1" to "Zone2"
  ...
end
...
if all in group "EnemyMainForce" > 250 and all in group "Guards1" < 50 ...
  transfer soldier from "EnemyMainForce" to "Guards1"
  ...
end
...

```

Change Description: transfer units

Copy this block to make another block, and then edit the newly pasted block, so that if guards1 group exceeded 35 men and guards2 are under 50 men, guards1 will transfer 7 men to guards2.

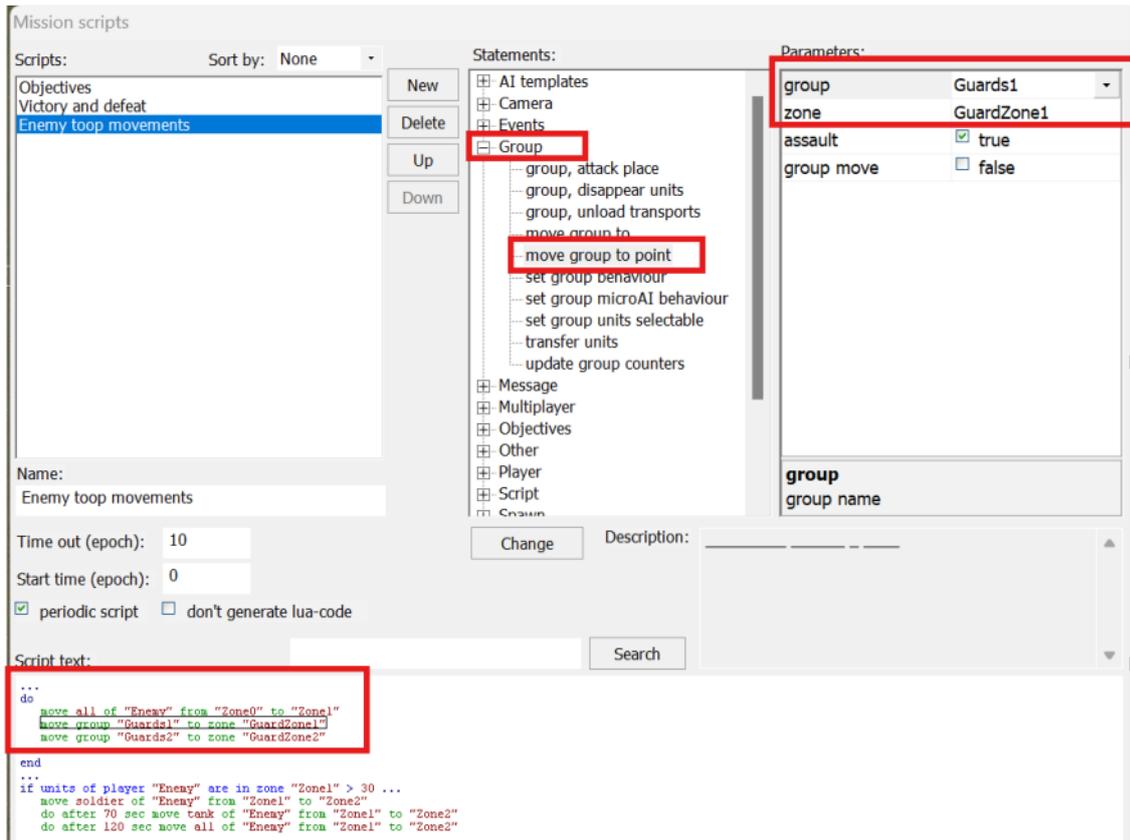
Script text:

```

...
do
  move all of "Enemy" from "Zone0" to "Zone1"
  ...
end
...
if units of player "Enemy" are in zone "Zone1" > 30 ...
  move soldier of "Enemy" from "Zone1" to "Zone2"
  do after 70 sec move tank of "Enemy" from "Zone1" to "Zone2"
  do after 120 sec move all of "Enemy" from "Zone1" to "Zone2"
  ...
end
...
if all in group "EnemyMainForce" > 250 and all in group "Guards1" < 50 ...
  transfer soldier from "EnemyMainForce" to "Guards1"
  ...
end
...
if all in group "Guards1" > 35 and all in group "Guards2" < 50 ...
  transfer soldier from "EnemyMainForce" to "Guards2"
  ...
end
...

```

Now, we want the guard groups to always move to their relevant areas. For this, you would go to the "do" statement at the top, click the three green dots, and choose the statement under "Group" category to "move group to point" (which appears to be mislabelled here because in fact it lets you move the group to a zone rather than a point). You would pick Guards1 and make it always move to GuardZone1, and likewise add a line to always move Guards2 to GuardZone2.



## AI defence scheme

This is an alternative where the enemy is meant to be defending, or "turtling" as some call it in RTS games. The enemy will be more cautious and not throw units at the player in waves. They will instead concentrate units in a core defence area or else expand, depending on how many units they possess. This can be much more dangerous to the player because the AI will act smarter and will be willing to retreat units, even with a preference for withdrawing heavy vehicles to a better position if needed.

In a complex scenario, some groups can carry out a defence scheme while others launch offensives from zone to zone as explained before.

For a defence scheme, you would make three or more groups belonging to the enemy (see "AI guard groups and zones" section again for how to do that as it is the same).

Go into Mission scripts. Create a new script called "Defence scheme". You may use this instead of the "Enemy troop movements" one as there will be no enemy offensive in this option, just three groups changing position. You only need to use the same methods as before to write the script for this. The example is below.

## Script text:

```

...
do
  move group "EnemyMainForce" to zone "Zone0"
  ...
end
...
if all in group "Guards1" <= 40 ...
  move group "Guards1" to zone "Zone0"
  ...
end
...
if all in group "Guards2" <= 40 ...
  move group "Guards2" to zone "Zone1"
  ...
end
...
if all in group "Guards1" > 40 ...
  move group "Guards1" to zone "Zone1"
  ...
end
...
if all in group "Guards2" > 40 ...
  move group "Guards2" to zone "Zone2"
  ...
end
...
if all in group "EnemyMainForce" > 250 ...
  transfer all from "EnemyMainForce" to "Guards1"
  ...
end
...
if all in group "Guards1" > 250 ...
  transfer all from "Guards1" to "Guards2"
  ...
end
...
if all in group "EnemyMainForce" < 150 ...
  transfer all from "Guards1" to "EnemyMainForce"
  ...
end
...
if all in group "Guards1" < 70 ...
  transfer all from "Guards2" to "Guards1"
  ...
end
...

```

Here, the enemy main force thinks zone 0 is essential and always moves to it, whereas the guards groups think their zones are essential and move to them. The main force will make massive transfers of forces to the guards groups (these are set about 50 in parameters), which will be strong, but if the main force is weakened, all the guards (put 0 to transfer all) will transfer back to the main force and fight their way through the player's siege to reinforce or relieve the main force. Likewise, the second guards group will dissolve and will transfer all troops to the positions of the first guards group if they become too weak. Additionally, the guard groups here always withdraw to the core defence area at zone 0 if their troops are reduced to fewer than 40 men.

## Reinforcement conditions

### Friendly reinforcement conditions

Go into Mission scripts.

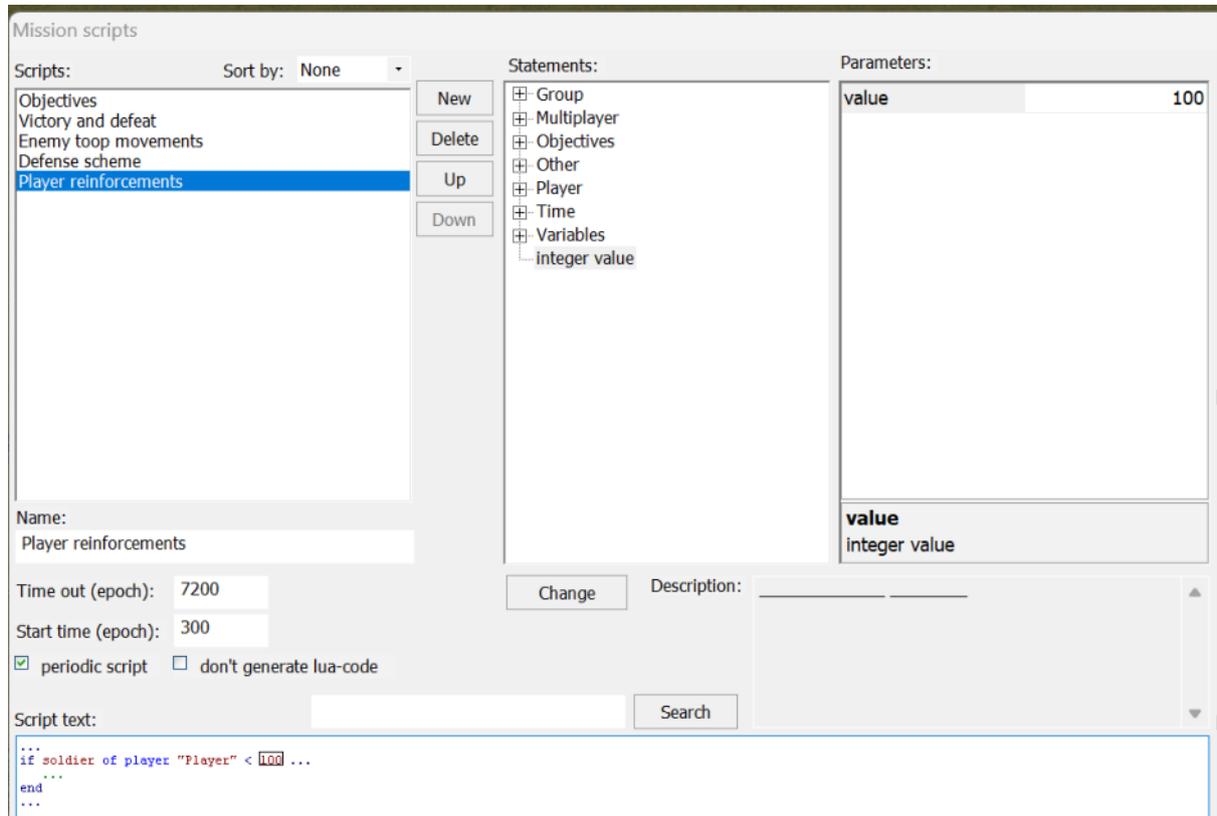
Make a new script and call it Player reinforcements, or the name of the nation if you prefer.

First, you want the arrival times to be something reasonable. 7200 epochs would equate to 12 minutes, but we might want some to spawn in during the first seconds, too. Go for 7200 epochs for "Time out" and 300 epochs (30 seconds) for "Start time". That way, you'll have some starting reinforcements at the beginning.

Now write the "if" statement for the condition, starting by clicking as always the three blue dots after "if".

Choose player, units of player by type, and then choose the Player as player, and pick the soldier parameter in the dropdown options for unit type.

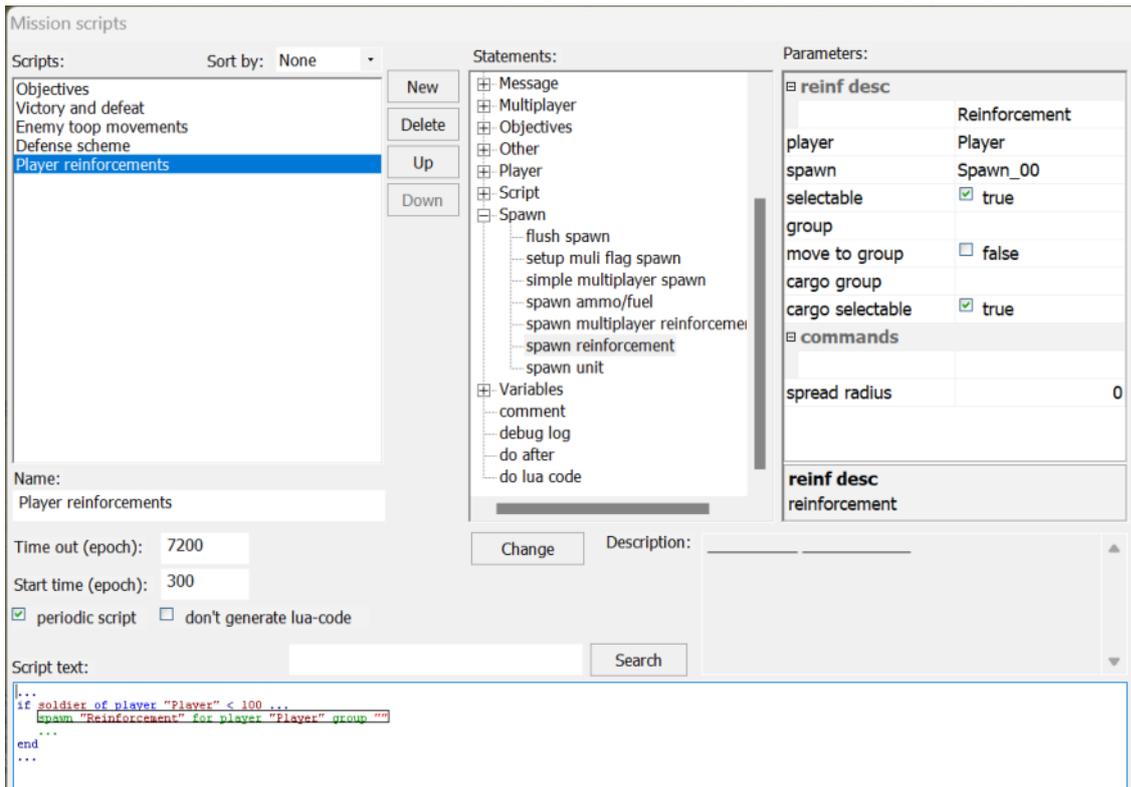
Click the "=" at the script text and change the statement to "<" (less than) by double clicking the statement you want to replace that with at the list of statements. Click and change the "0" to a value of 100 at the parameter box.



You are now checking if you have fewer than 100 soldiers.

Click the three green dots under this statement, to choose what to do about this.

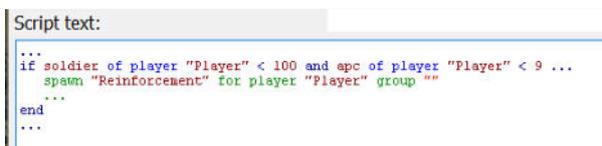
Enlarge the Spawn option at Statements, then choose "spawn reinforcement" by double-clicking that option. With this statement about spawning a reinforcement now set up as the line in the block for what to do, choose an infantry reinforcement type that you set up in the "Reinforcements" screen earlier. These can be chosen in Parameters. Set Player as the player, and choose the correct Spawn position that you set up in the "Spawns" screen earlier for the player.



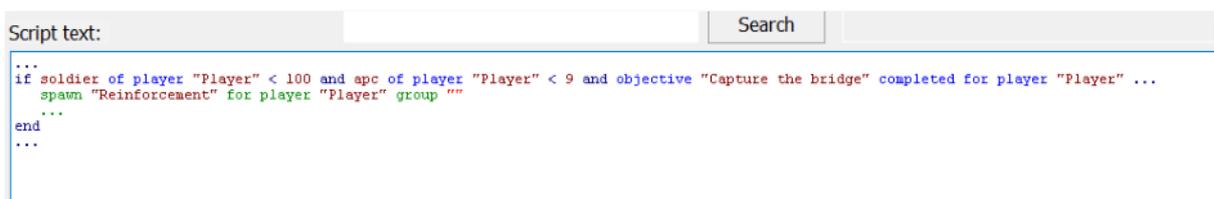
Now the mission will periodically provide you with some infantry if you have fewer than 100 soldiers.

However, your soldiers may be arriving inside an APC. This means that if your soldiers are still inside the APCs, the game will not know about them. Nor will the game be able to tell if your soldiers are dead, and you have hundreds of APCs because you did not lose them.

Using an "and" statement to add further conditions required for the reinforcement to arrive, change the script so that it will also check if the player has fewer than 9 APCs before spawning the reinforcement for the player.



You can add additional blocks of scripts in the same way. You can check for tanks. You can check if the player has more than (" $>$ ") 100 soldiers before beginning to add a heavier artillery reinforcement package into their reinforcements to support them. You can check if a flag is captured, too, like below.



Checking "service" and "tanker" unit quantities for the player can decide if the player requires more supply units (service will be supply trucks and tanker will be fuel trucks).

```

...
if service of player "Player" < 2 and tanker of player "Player" < 2 ...
  spawn "Reinforcement1" for player "Player" group ""
  ...
end
...

```

### Enemy reinforcement conditions

Go into Mission scripts.

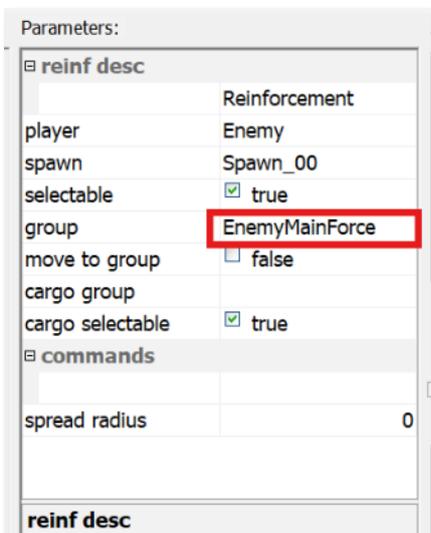
For this example, you can give the enemy powerful offensive forces. Because they will spawn into the AI zones set up earlier for "AI zone to zone troop movements", they will automatically receive orders to reach their objectives when they are numerous enough.

All you need to worry about is making the enemy spawn into the map.

Make a new script called "Enemy reinforcements" and make it more frequent, such as 4800 epochs (8 minutes).

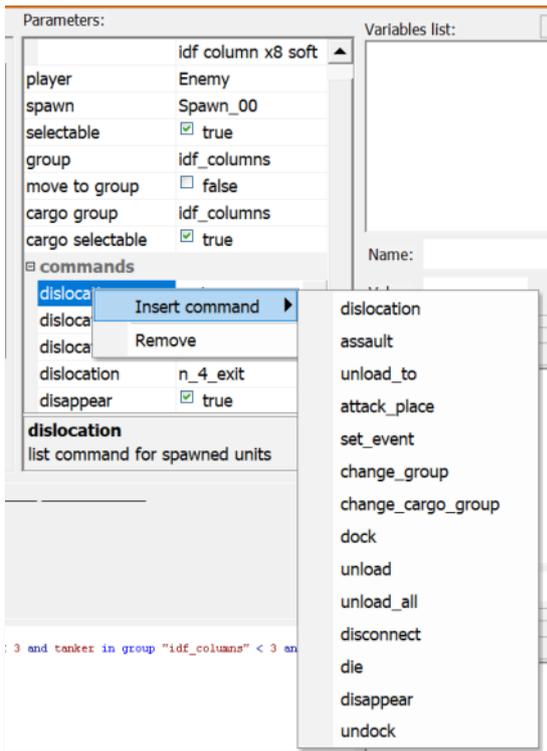
Follow the same example as with the player reinforcements in the last section, but in the parameters, change "Player" to "Enemy" or the name of the enemy player.

An extra parameter needs to be added, namely the default enemy reinforcement group EnemyMainForce as we called it earlier. Troops belonging to this group can be transferred to other groups when managing enemy forces (see "AI guard groups and zones").



### Enemy patrols

Sometimes, you may just want an enemy group to enter the map, patrol across it, and then disappear. This is useful for making covert missions where the enemy has no specific plans and you are behind enemy lines.



When you are making reinforcements, you can add commands to the incoming reinforcements that they will follow in sequence. They will complete each command before moving to the next one.

These can be "dislocation" (go to a waypoint, which can be added here), "disappear" (unit vanishes). A patrol can spawn with these commands, follow several "dislocation" commands to patrol a route, and then "disappear" at the edge of the map at the last waypoint.

The only precaution you might want to take in the scripts, is to make sure the patrol units are part of a group, and then add a condition to the script to only spawn them when this group is under a certain number (" $<$ "). Otherwise, the number of patrol units may become ridiculously high as a result of their patrol being interrupted by combat and additional patrols arriving behind them. Very few units should be allowed to be part of such patrolling groups, to keep the mission realistic.

### Deploy guns or troops and exit the map

You can add extra instructions so that enemy transports will deploy equipment or troops to oppose the player, and then continue travelling and leave the map. This can be a nice detail instead of spawning things directly into the map. Players will like to destroy enemy guns while they are being moved to their deployment point.

To do this, use the option to "Insert command" explained in the previous section, but try "disconnect" for dropping off an artillery gun, then another "dislocation" command after it for it to continue to a waypoint, and finish with a "disappear" command. Make the unit disappear after going to a waypoint you have made at the edge of the map. To deploy troops and then leave the map, a unit can use "unload".

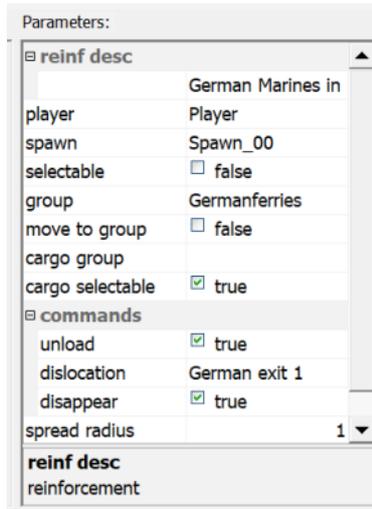
You can see other commands that can be given to the incoming units, too, and these are only suggestions. Be as creative as you want.

## Amphibious reinforcements

First, you will need at least one waypoint for the amphibious transport to vanish when its task is done. You want it to disappear from the map after completing its mission. The waypoint to reach before disappearing should be at the map edge.

After setting up a landing ship with troops inside it as a reinforcement type, go into Mission scripts and set it to spawn (in the sea, where you should have made its spawn and vector and taken note of the spawn number).

Create an example reinforcement script like this one, using the same option of adding commands explained in the "Enemy patrols" section of the manual.



Here, the reinforcement has the command to unload first (the reinforcement spawn vector goes directly to the beach where the unload order will be completed). Then, there is a dislocation command at one waypoint, the exit for German transport at the edge of the map. Finally, it disappears, just as in the patrol example. This script can be used exactly the same for any transport, whether amphibious or not.

Notice, however, that "selectable" is set to false. This will be necessary for the player's reinforcements, otherwise they can accidentally control the landing ship and cancel its order to exit the map. Additional safeguards should be scripted like checking there are no ships in the group "Germanferries" and also disappearing all ships in the group "Germanferries" 60 seconds before spawning any new ships, in case any got stuck.

```
...
if soldier of player "Germany" < 300 and all in group "Germanferries" = 0 ...
  spawn "German Marines in boat x20" for player "Player" group "Germanferries"
  do after 540 sec disappear all units of group "Germanferries"
...
```

This would prevent endless hordes of ships becoming stuck, and keeps it to two ferries transporting all the infantry in this scenario at any time. Players won't care if the script cleans up uncontrolled ferries by making them vanish before new ones arrive.

## Train deliveries

A train and its contained reinforcements would need to be set up as a reinforcement with attached carriages containing units as passengers, at the Reinforcements window in the editor.

Be warned that there may be difficulty with doing this for tanks being transported if you are trying to set them up inside the editor. Edits to the game files outside the editor may be needed to do tank-deliveries via trains, but infantry arriving in train carriages seem to be a safe choice.

You will need a continuous railway spline that is accessed at the edge of the map, for a train to deliver reinforcements and exit the map. You will also need a reinforcement spawn and a vector unique to this train, created at the Spawns window. The vector should lead to the train station where the train will drop off its cargo and troops.

In addition, you will need at least one waypoint where the train will disappear from the map after completing this task. It should be at the end of the railway line, at the edge of the map.

Add another group to the scenario such as "PlayerTrains".

When you have prepared as above, go into Mission scripts.

Create a script for the train to arrive, just as you would with the other examples. It will not be selectable and will have three commands.

Parameters:	
reinf desc	train_superheavies_2
player	Player
spawn	Spawn_00
selectable	<input type="checkbox"/> false
group	Trains
move to group	<input type="checkbox"/> false
cargo group	
cargo selectable	<input checked="" type="checkbox"/> true
commands	
unload	<input checked="" type="checkbox"/> true
dislocation	German trains return
disappear	<input checked="" type="checkbox"/> true

## Enemy strike conditions

Place waypoints throughout the map (see "Waypoint" under "Objectives" section for how to place these.) Names aren't important, you can just call them Waypoint1, Waypoint2, etc. About 30 different waypoints covering various fortifications, heights and other positions of advantage should be made.

Now go into Mission scripts again.

Make a new script called Enemy strikes.

Using the previous information in this manual, set it to a timer you find appropriate, for example, every 6 minutes.

Now choose conditions at the default "if" statement. You might add two conditions with an "and" to use both of them. This should check if the player is within 30 meters of the target position, for example Waypoint 1 "and" check if the enemy is within 110 meters of the target position (this is about engagement range and close enough to decide to bombard the area as a precaution, whether they can see the location or not).

You would choose Other, and "bombardment" as the statement type for what to do, following the "if" statement.

You can then set spread radius, number of bombs, period and start offset at the parameters.

The screenshot shows the 'Mission scripts' editor with the following details:

- Scripts:** A list on the left with 'Enemy strikes' selected. Buttons for 'New', 'Delete', 'Up', and 'Down' are to its right.
- Statements:** A central list of actions including 'move group to', 'set group behaviour', and 'bombardment' (under the 'Other' category).
- Parameters:** A table on the right with the following values:
 

Parameter	Value
position	Waypoint 1
spread radius	30
num bombs	8
period	6
start offset (e)	30
- Name:** 'Enemy strikes' is entered in the text field.
- Time out (epoch):** 3600
- Start time (epoch):** 2340
- Options:**  periodic script,  don't generate lua-code
- Script text:** A text area containing the following Lua code:
 

```
...
if is units of "Player" near point "Waypoint 1" and is units of "Enemy" near point "Waypoint 1" ...
  bombardment("Waypoint 1", 30, 8, 6, 30)
...
end
...
```

For an enemy plane, you would first need to set desc for the enemy nations' plane role to a model in the game files. Then, you would need to send plane. These options are present in the "Other" category of statements.

Mission scripts

Scripts: Sort by: None

Statements:

- AI templates
- Camera
- Events
- Group
- Message
- Multiplayer
- Objectives
- Other
  - bombardment
  - disappear units
  - object, create
  - object, remove
  - plane set desc
  - plane set spawn side
  - send plane
  - send plane from map
  - unload ship
- Player
- Script
- Spawn
- Variable

Parameters:

country	germany
plane	bomber
plane desc	german/ju_88s.dsc

Name: Enemy strikes

Time out (epoch): 3600

Start time (epoch): 2340

periodic script  don't generate lua-code

Script text:

```

...
if is units of "Player" near point "Waypoint 1" and is units of "Enemy" near point "Waypoint 1" ...
    use "bomber" desc for "germany" to "german/ju_88s.dsc"
    send "Enemy" bomber to "Waypoint 1"
...
end
...

```

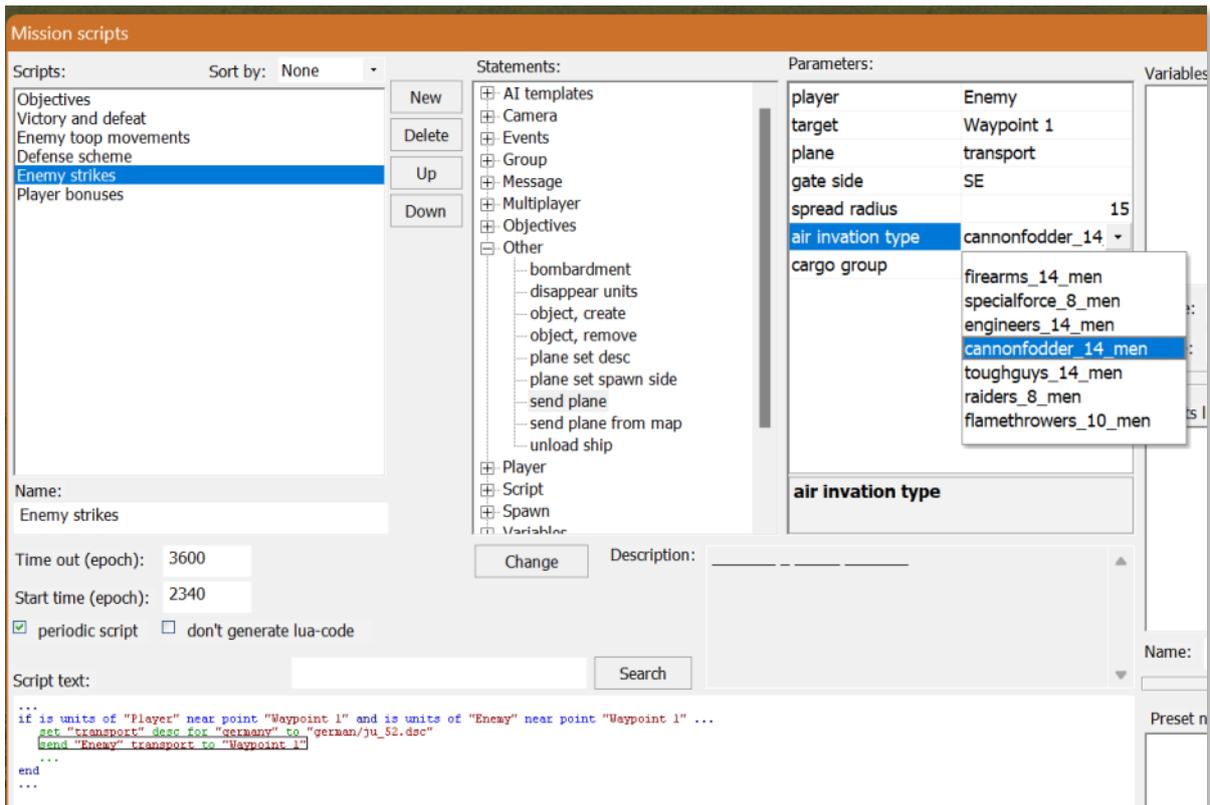
When sending the plane, you would need to choose parameters like the gate side for which side the plane will spawn into the battle.

Parameters:

player	Enemy
target	Waypoint 1
plane	bomber
gate side	SE
spread radius	15
air invasion type	
cargo group	

### Enemy paratroopers

Enemy paratroopers would be called from off-map to the location exactly as with an enemy strike. A plane and direction would need to be set to something appropriate. The difference is only in parameters, where you need to choose invasion type from the available list.



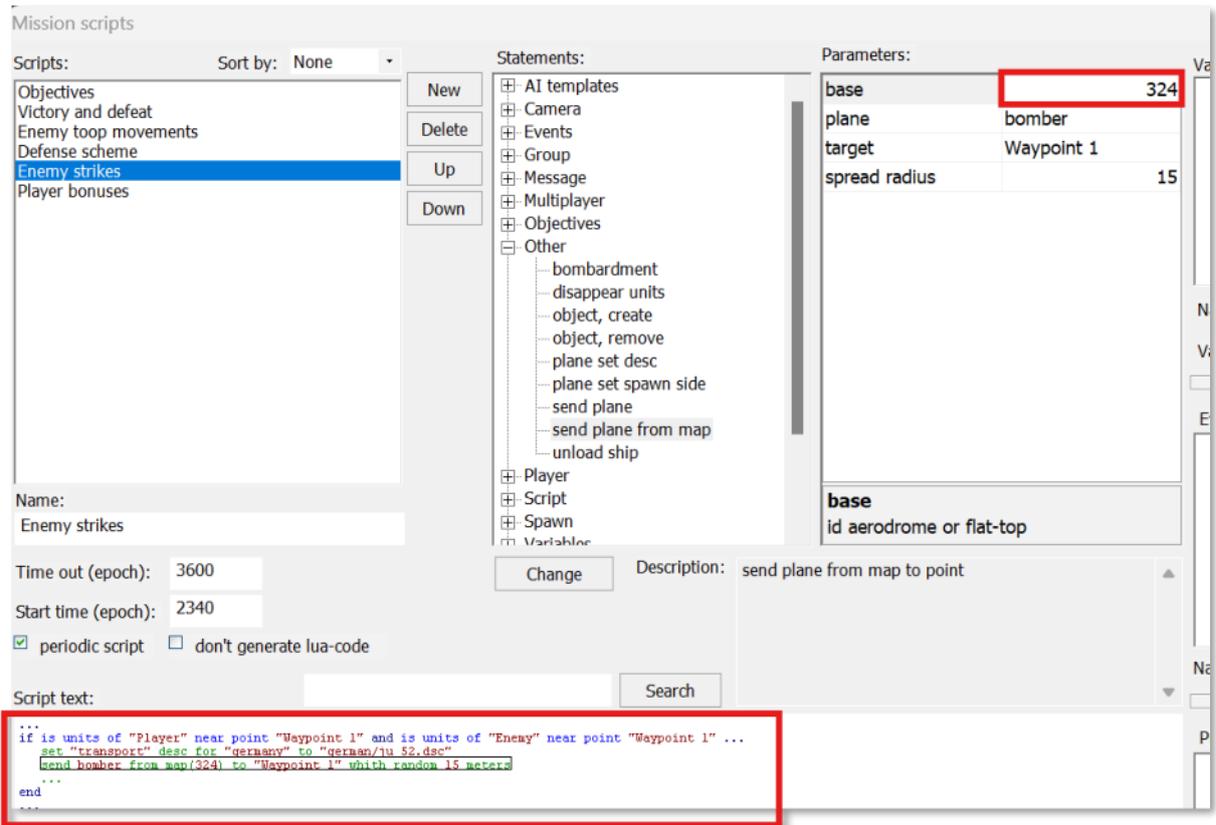
You would of course set the plane to something appropriate, too.

### Enemy airfields and aircraft carriers

Enemy strategic objects like airfields and aircraft carriers can be called by ID when carrying out a strike, so that the strike will be launched from the strategic object. ID can be retrieved the same way it is identified for a bridge (see "Destroy the bridge" under the "Objectives" section of the manual for an example). At least, this worked fine with the original airfields and units, but modded airfields and units may have difficulty.

A script for calling an enemy plane (or even a friendly plane depending on the ownership of the base and the parameters) will be set up as follows.

In the example, we identify the ID of the base or aircraft carrier as 324 and call in a strike from it to waypoint 1 instead of calling an off-map bomber to perform the strike. Aircraft ordered this way will return to base after performing the strike mission.



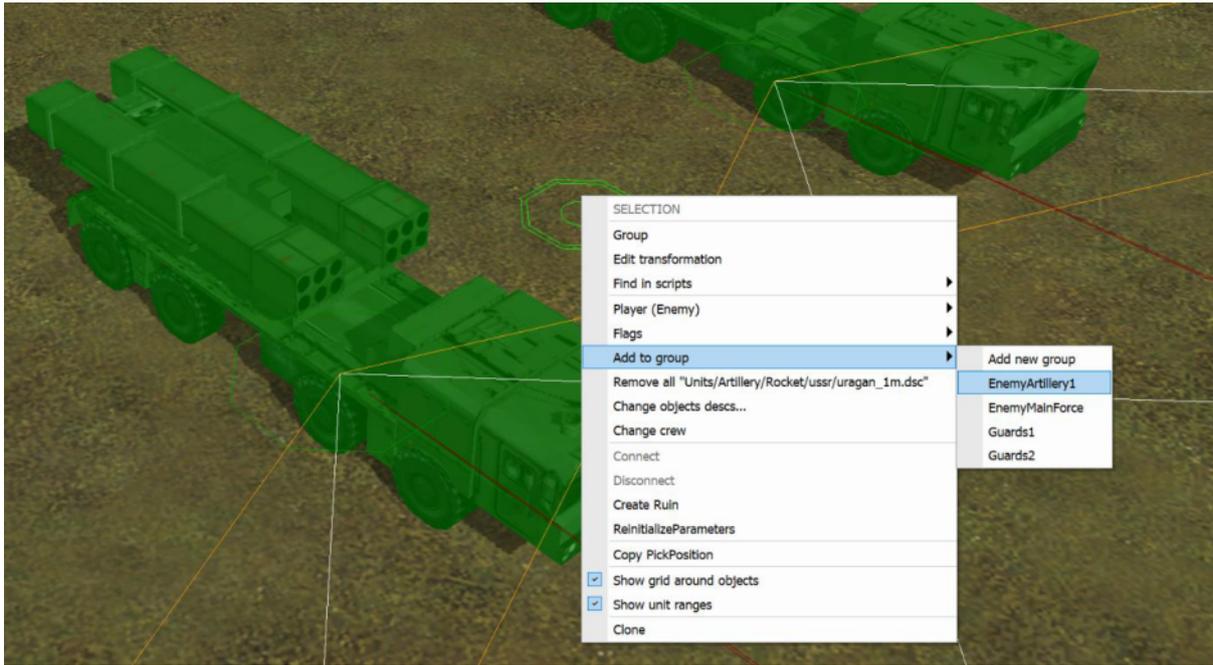
Be warned that enemy airfields cannot be destroyed, only the planes can be destroyed on them, and the airfields count as units in the game, so they can interfere with completing objectives if you set the objectives up incorrectly.

You would add planes to an airfield or aircraft carrier just as you would add crew/passengers to a vehicle or building object in the map. See "Placing units" section on "Soldiers in buildings" for how this works.

### In-map enemy artillery

Enemy artillery placed in the map can be called upon to strike waypoints. They should be units that only fire shells in ballistic fire mode and do not use machine guns or a lower-calibre weapon as another weapon type (for example, you can use the German railway guns or howitzers of the original game or the Russian Katyusha launchers).

The artillery should be set up as a group (see "AI guard groups and zones"). You can add enemy artillery pieces to a group by selecting them on the map, right-clicking, and choosing "Add to group".



The units must belong to the right player the group was assigned to, for this option to appear.

In the script covering enemy strikes (or another script if you prefer it to work on a different, more frequent timer) you can set a group to attack place, choosing spread radius, group, and waypoint.

Mission scripts

Scripts: Sort by: None

Statements:

Parameters:

Name: Enemy strikes

Time out (epoch): 3000

Start time (epoch): 2340

periodic script  don't generate lua-code

Script text:

```

...
if is units of "Player" near point "Waypoint 5" and is units of "Enemy" near point "Waypoint 5" ...
    group "EnemyArtillery1" attack "Waypoint 5" spread radius 20
...
end
...

```

group, attack place

group	EnemyArtillery1
target	Waypoint 5
spread radius	20

target  
target point

Description: Group receive command attack place with random radius

## Placing units

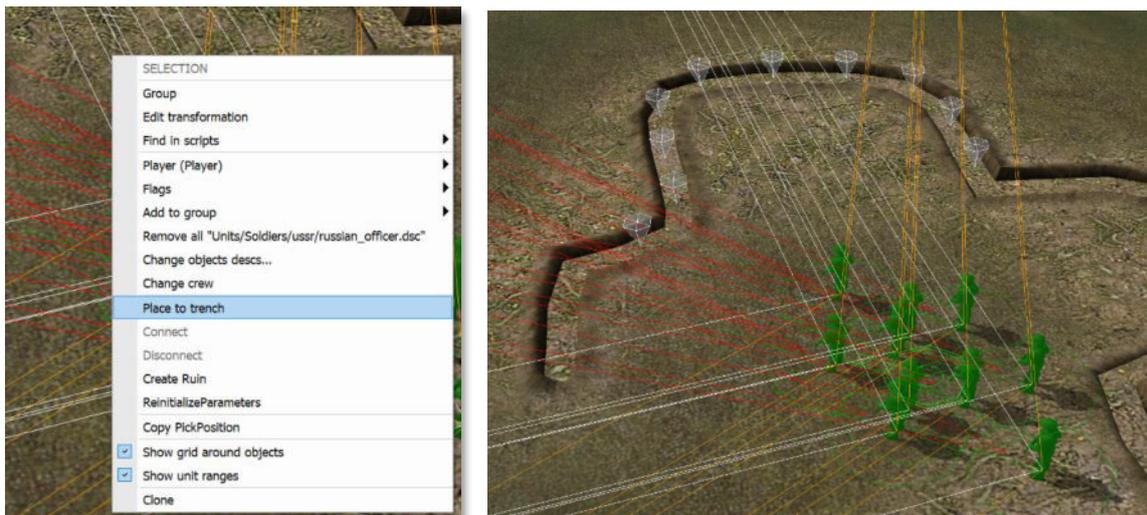
Units can be placed exactly like with any objects, in Object mode (see "Map-making" guide and "Forest and rock". The difference is that you would choose which player they should belong to when placing them.



Alternatively, you can right click after selecting the units in the editor and change their allegiance to player or enemy.

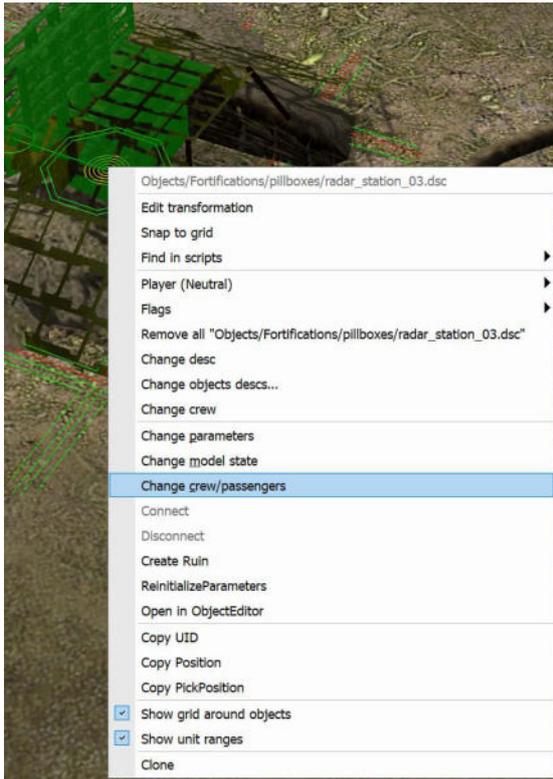
## Soldiers in trenches

Soldiers can be placed to trenches by right clicking after selecting them in the editor (in Object mode) and then choosing the place to trench option, then hover your cursor over the trench and left click to place them.



## Soldiers in buildings

Soldiers can be placed in buildings by right clicking a selected building in the editor, choosing "crew/passengers" and adding the soldiers.



However, it is best to just put them outside the building and they will automatically go inside it with their default behaviour (which itself can be altered by adding them to a group and editing the group's behaviour, see "AI guard groups and zones" section for information about it).

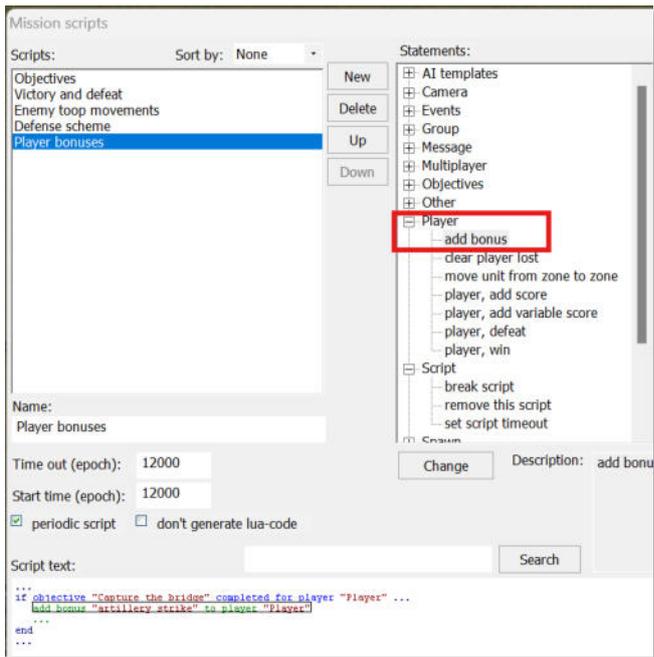
Soldiers put into buildings via the building parameters do not shoot when enemies come nearby, instead holding fire and acting like spies for the enemy. Those who enter buildings during runtime will defend the buildings properly.

## Player bonuses

Go to Mission scripts and make a new script called Player Bonuses.

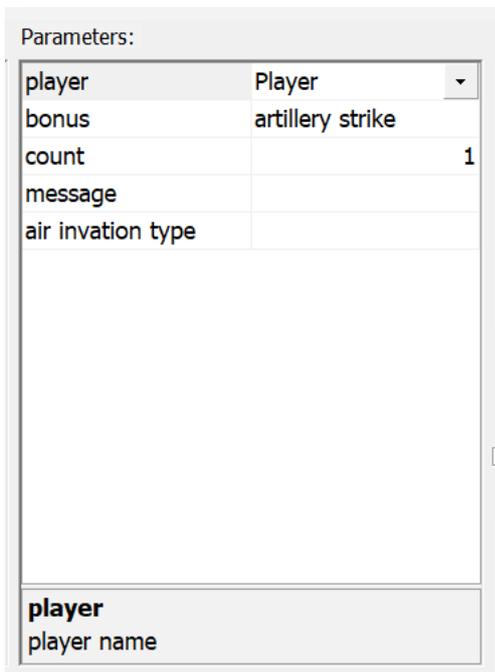
The easiest way to set up bonus conditions is to give them a long timer (20 minutes, or 12000 epochs, would be reasonable to wait as you won't want the player winning through bonuses alone, they are only support). Also checking as an additional condition if a certain flag is controlled is another good and easy way.

Choose the Player category and "add bonus" Statement to add when the condition such as an objective is met.



### Artillery strike

Chosen as a parameter. You can also choose the number of bonuses under "count".



### Air strikes

You can choose the type of plane, direction it comes from, and the number of bonuses.

You would need to start with the type of plane. After making the condition statement (for example, your second objective is completed for the player), choose Other, then "plane set desc" as the statement for what to do. Fill in the Parameters there using the drop-down options for your nation as set up at the start of the mission design process, the role of plane that will

be used for the strike e.g. "bomber", and the exact model of aircraft in the game files.

Mission scripts

Scripts: Sort by: None

Statements:

Parameters:

country	ussr
plane	bomber
plane desc	ussr/a_20_ussr.ds

Name: Player bonuses

Time out (epoch): 12000

Start time (epoch): 12000

periodic script  don't generate lua-code

Script text:

```

...
if objective "Capture the bridge" completed for player "Player" ...
  add bonus "artillery strike" to player "Player"
...
end
...
if objective "new objective 1" completed for player "Player" ...
  set "bomber" desc for "ussr" to "ussr/a_20_ussr.ds"
...
end
...

```

Now set the spawn side. In this example, it is changed from the default direction to South, where the player's units may be coming from.

Mission scripts

Scripts: Sort by: None

Objectives  
Victory and defeat  
Enemy toop movements  
Defense scheme  
Player bonuses

New  
Delete  
Up  
Down

Statements:

- AI templates
- Camera
- Events
- Group
- Message
- Multiplayer
- Objectives
- Other
- bombardment
- disappear units
- object, create
- object, remove
- plane set desc
- plane set spawn side
- send plane
- send plane from map
- unload ship
- Player
- Script
- Spawn
- Variables

Parameters:

player	Player
gate side	S

Name: Player bonuses

Time out (epoch): 12000

Start time (epoch): 12000

periodic script  don't generate lua-code

Script text:

```

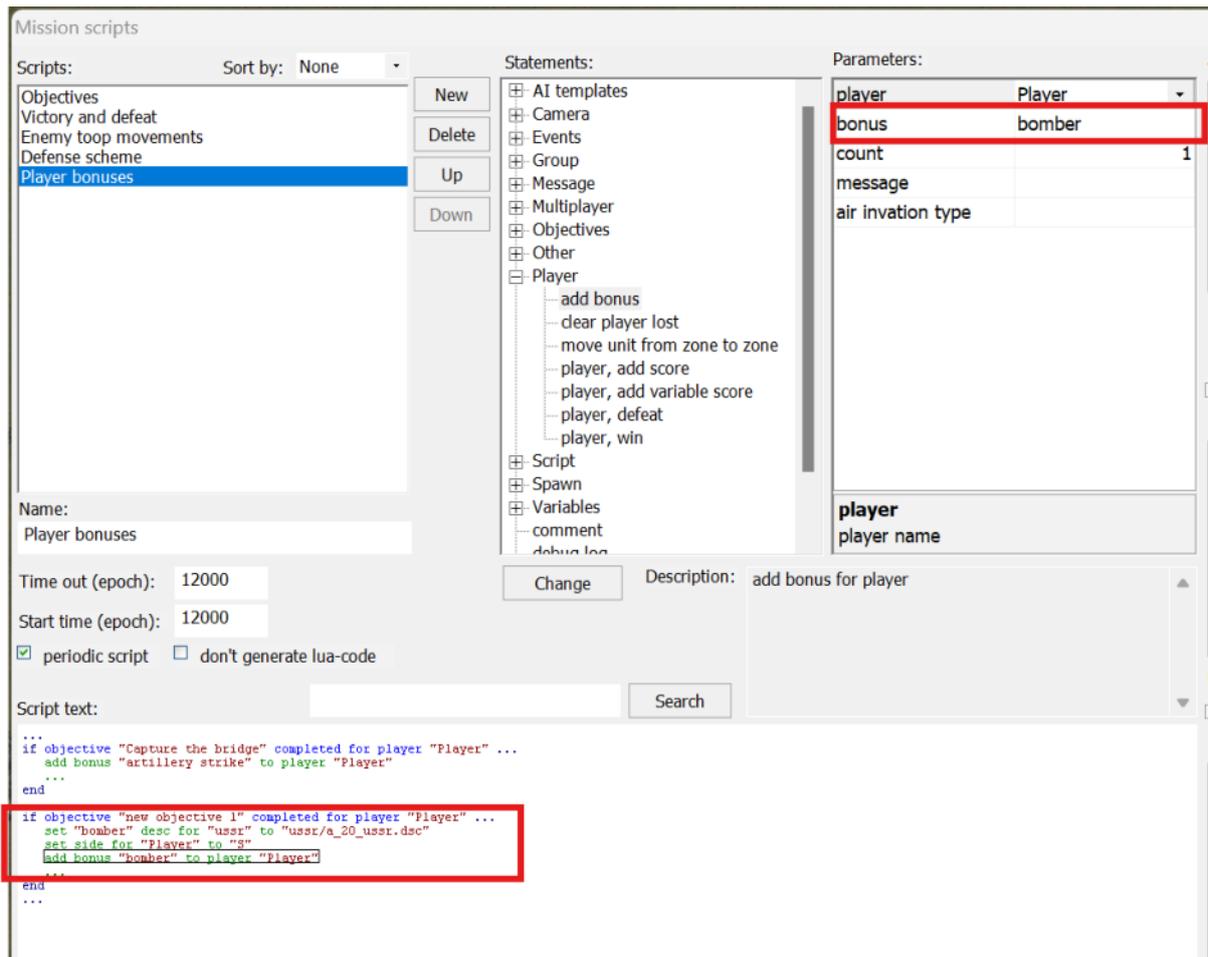
...
if objective "Capture the bridge" completed for player "Player" ...
  add bonus "artillery strike" to player "Player"
...
end
...
if objective "new objective 1" completed for player "Player" ...
  set "bomber" desc for "ussr" to "ussr/a_20_ussr.desc"
  set_side for "Player" to "S"
end
...

```

Change Description: Set default plane start side for plane spawn

Now, include another line awarding the Bomber to the player after setting the side it should spawn from.

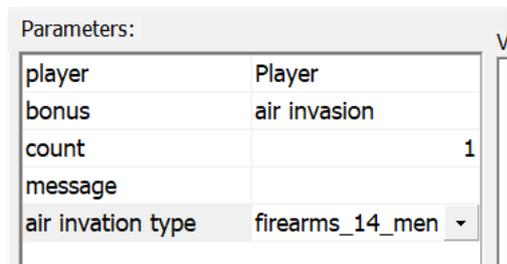
Just copy your other bonus line if you want, but change the type to Bomber.



Optionally, a one time script can be made at the start of the mission to decide what planes a player will have and where they will spawn from, instead of setting these every time a plane is awarded.

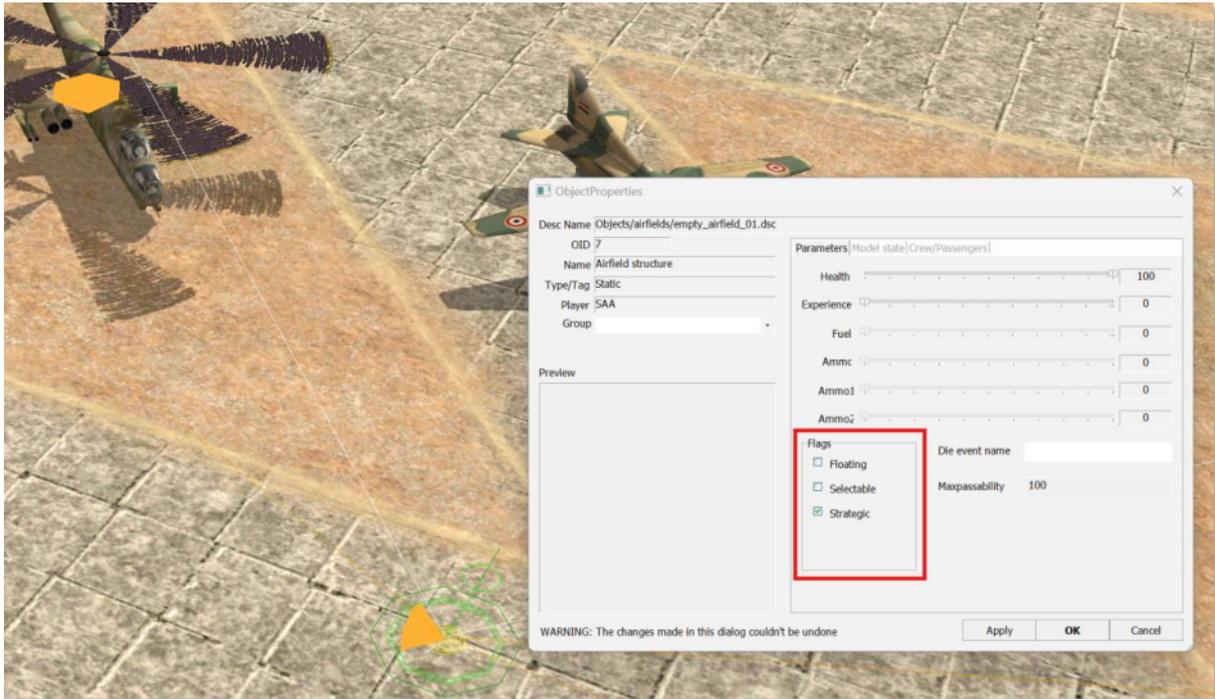
### Paratroopers

You can choose the same settings as with air strikes but must choose air invasion and an invasion type parameter from a dropdown list.



### Player airfields, battleships and aircraft carriers

Player airfields, battleships and aircraft carriers are all strategic-level objects that can be set up as such in their settings. This is done by selecting them in Object mode and changing their parameters so they are strategic. Also disallow player control on them (untick "Selectable"), as moving them at a tactical level may cause errors.



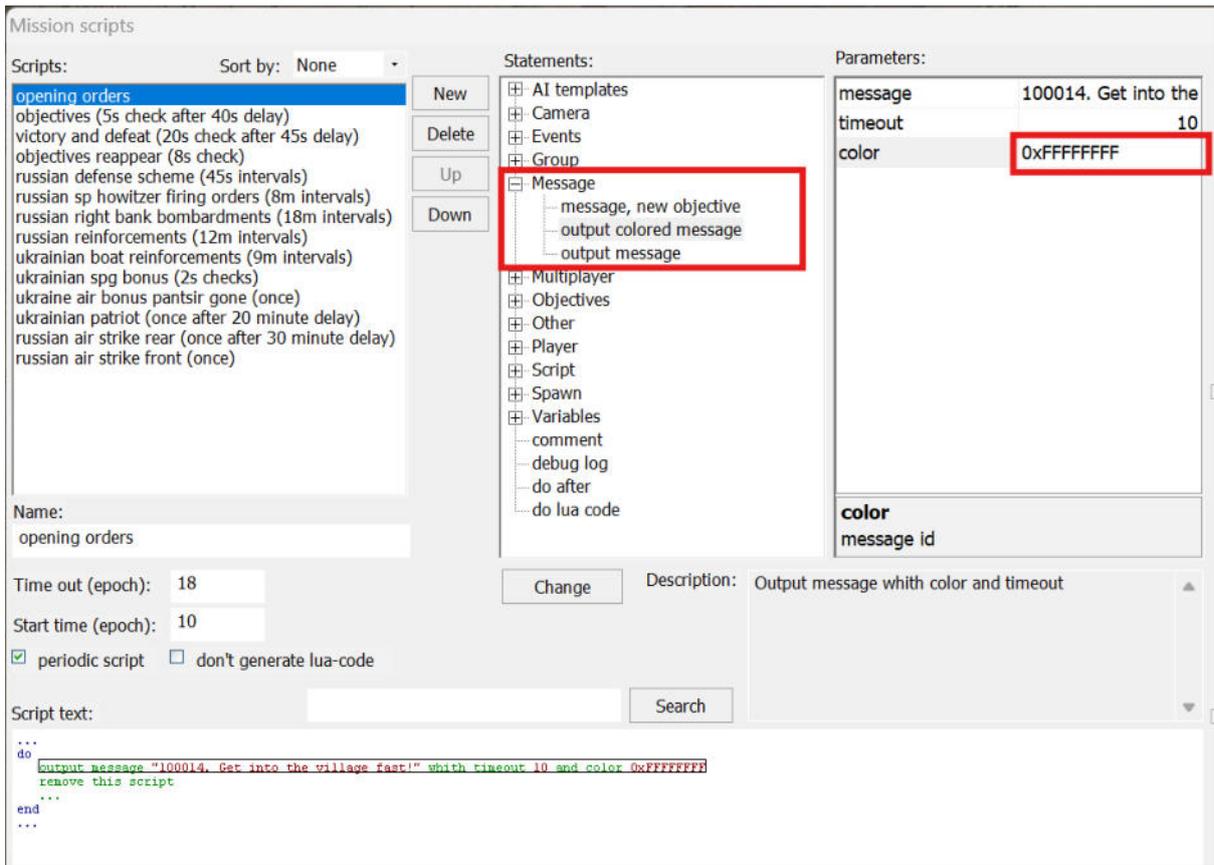
Doing this allows these objects to be controlled in the strategic map, but they can take no action at the tactical level.

Aircraft can be added to an airfield just as passengers can be added to a unit or troops can be added to a building, again through the parameters.

## Coloured messages

You can set up a message to display, but it is best for it to only display once or very infrequently.

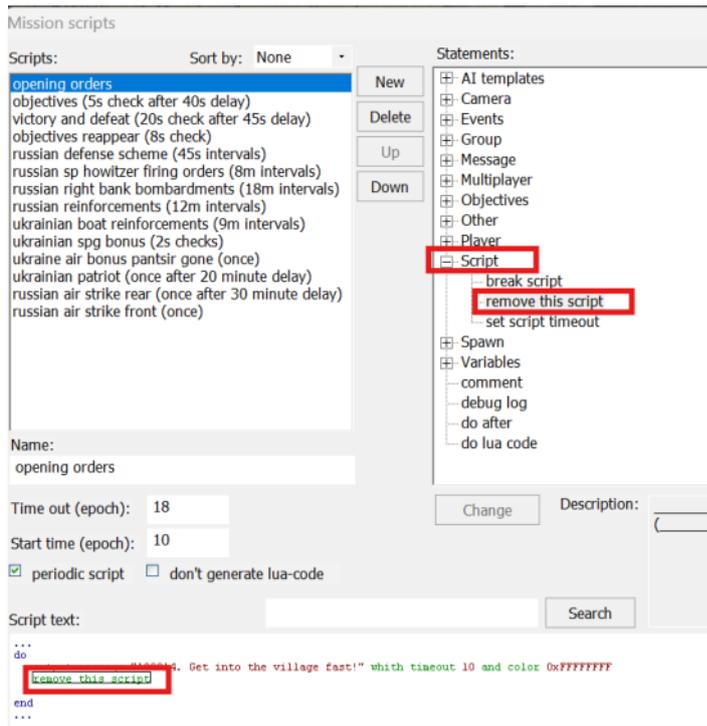
We can try an opening message in the below example.



The colour has been changed from default black to 0Xffffff or white. You can look up Hex colours online to decide other colours you might want to use. A timeout of 10 seconds is included, and the message is selected from a dropdown of available options (additional messages can be made at the editor's Message editor, explained in the "Objectives" section).

However, this message would play indefinitely, so the script must be removed after it is used once. This is why a "remove this script" line is included under the output message line.

This is found under the "Script" category in the Statements box.



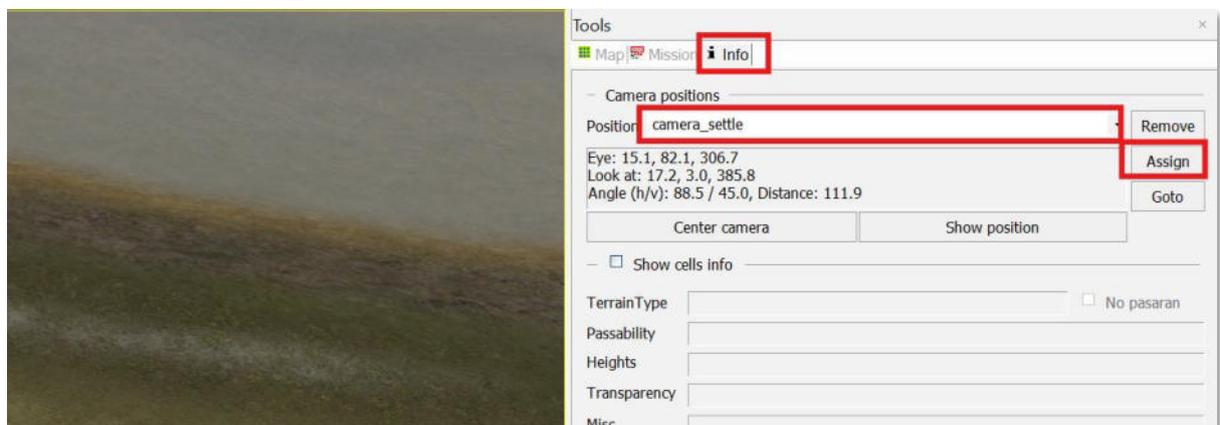
Adding a “remove this script” line can be useful for one-time events, one-time reinforcement bonuses, etc.

## Cameras

You can set up the player camera to interpolate from one view to another view, preventing the player from controlling the camera while it moves, for a more cinematic experience.

Again, we can use an example where the camera only moves at the start of the mission and this does not happen again (other missions may be designed do the same after you completed an objective.)

You would go into the “Info” mode at the Tools panel. Then you need to position the in-editor camera where you want the camera to go, and save this information by typing a name for the camera position at the “Position” box, before clicking “Assign” to store the name of the camera angle.



You can save multiple camera positions like above and then use one or more of them in the Mission scripts mode to tell the camera to go there.

Go into Mission scripts at the one-time script where you would like the camera to lock and move for the player in the mission.

Before the "remove this script" line is added to stop the camera from being controlled by the scripts and be free for the player to use, you would set camera lock to true.

Mission scripts

Scripts: Sort by: None

Statements:

Parameters:

Name: vdv\_deploy x168 men (once)

Time out (epoch): 105

Start time (epoch): 0

periodic script  don't generate lua-code

Script text:

```

do
  camera lock (true)
  camera at "camera_settle"
  do after 12 sec camera at "camera_settle_3"
  do after 25 sec camera lock (false)
  remove this script
end

```

Camera lock is first used and is set to true in parameters. The camera is then set to be "at" a location, and then sent to be "at" another location after a "do after" delay of 12 seconds, with its own timer in parameters of 13 seconds.

After the "do after" delay of 25 seconds (in other words, the remaining 13 seconds mentioned above), camera lock will be used again but it will be set to false in parameters, releasing control to the player.

The script is then removed so it will never be used again in the mission.

You can combine these one-time camera shots like this with reinforcements spawning in and messages at the opening of a mission, to make the start of a mission more entertaining and cinematic.

Note that the editor can handle much different camera angles than the game itself, so only use camera positions that are appropriate for the in-game camera.