

(Old pages, still useful for reference)

- [Ai Escorts \(and Special Objectives\)](#)
- [Applying Custom Instances](#)
- [Cubemaps, Cubelights and Lighting your level](#)
- [Limitations and Issues with mapping in Diesel](#)
- [Making and adding Environments](#)
- [Navigation in diesel](#)
- [Optimisation: Occlusion and Portals](#)

Ai Escorts (and Special Objectives)

A test map is provided in the Attachments.

Escort Characters

All Escort characters will stop moving when enemies block their path, different radiuses apply for different hostages. Escort unit data typically contains the following (using Matt as an example, This is not important unless you want to make a custom character act as an escort)

```
<!--Escort Animations-->
<anim_state_machine name="anims/units/escort_guy/escort_undercover_machine" />

<!--Escort Extensions-->
<extensions>
  <extension name="base" class="CivilianBase" >
    <var name="_tweak_table" value="escort_undercover" />
    <var name="_spawn_state" value="escort_undercover/spawn/loop" />
  </extension>
  <!--Most important, makes the escort not die.-->
  <extension name="character_damage" class="CivilianDamage">
    <var name="immortal" value="true" />
    <var name="_no_blood" value="true" />
  </extension>
</extensions>
```

Escort Special Objectives/Patrols

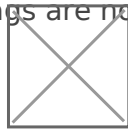
Escorts in heists are fairly simple preset tracks for Escort civilians to follow.

First step is to choose your patrol path/s and add those positions to the level. Keep in mind you may need multiple patrol groups if you want your Ai to take different paths for different reasons.



Once you have a path for the Escort to take you need to add a SpecialObjective element for them to use that patrol group and use it like one. Not all of these settings are required, but most are not necessary for what we are doing here. (Access Flags are not shown in the following image but use

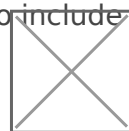
civ_male/civ_female to make it work for escorts.)



Telling Escorts to use a path

When spawning the Escort character you will want to include an EnemyDummyTrigger element to

tell the escort to use the SpecialObjective we made.

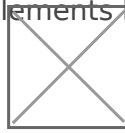


Making things happen along the way

Since were using a SpecialObjective element to make these escorts we can include triggers and branches to them.

- **SpecialObjectiveTriggers** set to Administered trigger type will execute once your escort has been told to follow the path, In the example level this is used to spawn an enemy that will block his path if you do not kill the enemy. A Complete trigger can be used for when your escort reaches the end of his pathing.
- **SpecialObjectiveGroup** With a follow up to take 2 different SpecialObjective escort paths can be used to start of connect branches together, in the example level its used to

go left or right and it activated using the Followup Elements list in the 'escort_start'



SpecialObjective (Use Instigator is required for this)

Applying Custom Instances

Note: Newer versions of Beardlib Editor includes dedicated custom instances, this workflow is largely unnecessary and outdated. Information on the page is still useful past the main.xml example.

Instance Example (In the Main.xml)

```
<instance id="level_id">
  <add file="levels/level_id/add.xml"/>
  <include directory="levels/level_id">
    <file file="world.world" type="generic_xml"/>
    <file file="world/world.continent" type="custom_xml"/>
    <file file="world/world.mission" type="generic_xml"/>
    <file file="editor_only/editor_only.continent" type="custom_xml"/>
    <file file="editor_only/editor_only.mission" type="generic_xml"/>
  </include>
  <packages/>
</instance>
```

With this setup you can work on the instance as a level without needing to move any files around

CONTINENT

Example: table continent="world" folder="levels/instances/unique/hoxestatesecurityroom/world" indexsize="1000" name="hoxestatesecurityroom001" position="0 0 0" rotation="0 0 0 0" script="default" startindex="11000"

```
continent="world"
```

- The name of the continent file its in.

```
folder="levels/instances/unique/hox_estate_security_room/world"
```

- The path to the "world" folder containing the continent and mission_script files.

```
index_size="1000"
```

- How many units and/or elements it takes up in memory. beardlib sets all instances to 1000 by default even if it only uses 3, you can check official maps files for exact index counts but there doesn't seem to be any issues with it being set to 1000.

```
// name="hox_estate_security_room_001"
```

- In level name and id, cannot be the same as any other instance or things get mixed up.

```
// Position/Rotation
```

- The usual positional information.

```
// script="default"
```

- Hard to explain but I think it's like a sub-folder for element code, doesn't need to be anything but default for custom heists.

```
// start_index="11000"
```

- If you have 3 instances each taking 1000 spaces for the index_size this decides where the elements are stored. starts at 0 each instance adds its index_size from the last instance

MISSION

Example: entry index="12" type="string" value="hoxestatesecurityroom001"

```
// entry index="12"
```

- The order of instances, if this is the 5th instance on the map it's 5.

```
type="string"
```

- Magic code stuff, don't touch it.

```
// value="hox_estate_security_room_001"
```

- This is the name id mentioned previously in the continent.

Cubemaps, Cubelights and Lighting your level

Cubemaps

A cubemap is a six-sided texture used for reflective surfaces to reflect stuff. Here's an example:

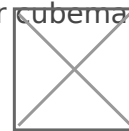


“ This looks wrong! Why are there buildings in the reflection even though we are indoors?

That's because diesel does not support realtime reflections, so map makers have to create cubemaps manually

To do so you will need to spawn a `cubemap_gizmo` from the editors spawn menu. Position it however you want. Then head over to the env manager and press "Build Cubemaps" (this will fail if you don't have python3 and PIL module installed) If successful, the editor will tell you the path to the cubemap relative to your map's directory. Remember this path!

Reload the map, click on the env tab in the upper menu and find your `cubemap` texture in the



cubemap dialog box. Once selected, the global cubemap will change:

Eh, not too good looking, but it will do!

The game will keep track of changes to the cubemap texture so you won't have to reload the map each time you build cubemaps for the gizmo you're working with. Keep in mind that you will have to reload the map every time a new gizmo is added to the map!

“ This cubemap is global, how can i make certain areas have different cubemaps?

For that you'd need to create a separate env area, tick the cubemap option and include an environment file with a different cubemap texture. You'd need to do this for every area

“ Why are reflections unchanged for some units?

That's because these units override the global cubemap. To fix that you need to extract the unit, change the `global_texture` parameter to the path of your cubemap in the `seq_manager` and load it from `extract`. That way you can have per-unit local cubemaps!

Cubelights

Cubelights are a six-sided texture used for projecting static shadows from lightsources onto the

environment Here's an example:



You can do that too! All you have to do is to spawn a `light_omni_shadow_projection` in the spawnmenu. All lights with a `shadow_projection` prefix can build projection textures. Now just select the light, head over to the env manager and press "Build Projection Lights" (this will fail if you don't have python3 and PIL module installed)

Same as with cubemaps the projection texture will only start working after a reload. Let's compare

the two: Before:



After:



Building projection lights can also fix light leaking through map geometry

“ Projection lights don't build for a specific light unit!

If your light unit does not have options such as shadow resolution or doesn't have any light settings at all you would need to extract it to build projection lights

Limitations and Issues with mapping in Diesel

Here is a list of potential issues you may have when working on custom maps.

Out of Memory Crash

Application has crashed: C++ exception
Could not load texture because IDirect3D9::CreateTexture call failed.
Direct3D could not allocate sufficient memory to complete the call.

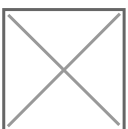
This issue only happens on maps that have too many assets loaded, through overbloated packages or high file size assets. Temporary workaround to this issue is to lower texture settings to Medium or lower, however in most cases this can be avoided by loading assets that only exist in larger packages from Extract.

If you use more than 2000MB of memory in editor you will likely crash due to team Ai using a very large amount of memory, even more likely players will crash when using mods in the heist also. The White House heist uses ~1600MB. The Safehouse uses ~1600MB.

You can figure out how close you are to 'OOM' (Out of memory) crashing by using the `-qa` steam launch option.



Ingame it will look like this.



Loading units

TLDR: two of the solutions are near perfection but both have some drawbacks, adding them from extract cannot load some units and packages cause slow loading times and unstable maps.

Loading units for maps was an issue I tried solving for a long time.

Loading from extracted files

Phase 1:

With overkill adding a new function to add files(DB:create_entry) I found out you can force existing assets to be loaded by loading them from the map itself but sadly this has a few issues, first of all it takes a lot of space and secondly biggest issue is that it cannot load effects from what I know which can cause some units to crash.

Phase 2:

After releasing the editor I had that idea floating in my head that maybe we don't need the model and texture of the asset to make it load in the map(had the idea after finding out you can mostly only read these files inside the game using DB:open even if they're not loaded) and I was right, after testing that it had no issue what so ever to load the units.

With phase 2 we cut the file size of the map by a lot(for example I tried doing that for a big custom heist and it was only around 8mb) this makes that method a more viable option now the only remaining issue is that DB:create_entry does not support effects which is a big issue and can cause some units to fail to load, this option is not really viable for characters. DB:create_entry also doesn't support fonts from what I know(pls ovk)

In future versions of the editor you will see an improvement to the load from extract system.

Loading packages

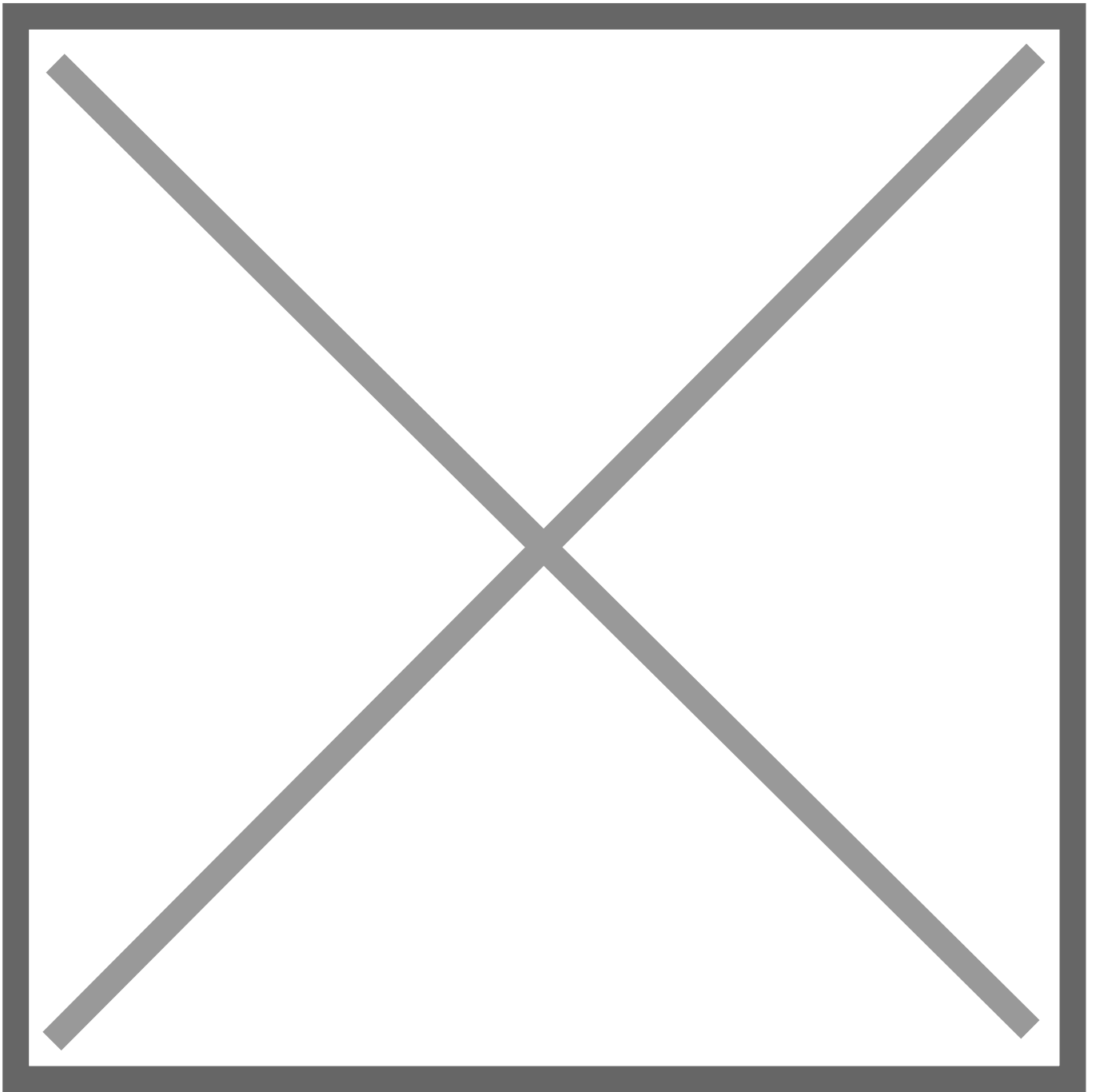
After fiddling around with packages I found out you can load as much as you want as long as you monitor the size of the packages(packages are actually bundles just non hashed! so we can see their actual size by using core lua functions) this method a limitation, it can be quite hard to load packages for big maps resulting in the map loading too much packages which can also cause the loading to be very slow and make the map unstable.

Making and adding Environments

Environment is like a config which decides how light and shadows look in your level, this is what makes the level a night map or a day one, you can also use environment areas to make rooms look better, here we will talk about making them generally.

All of the actions are done in environment menu which is the world icon at the top menu.





Editing

You can edit the values of the environment freely, for now we won't have an exact explanation for each value in there, you can play around with it until you feel comfortable with the controls and the environment itself.

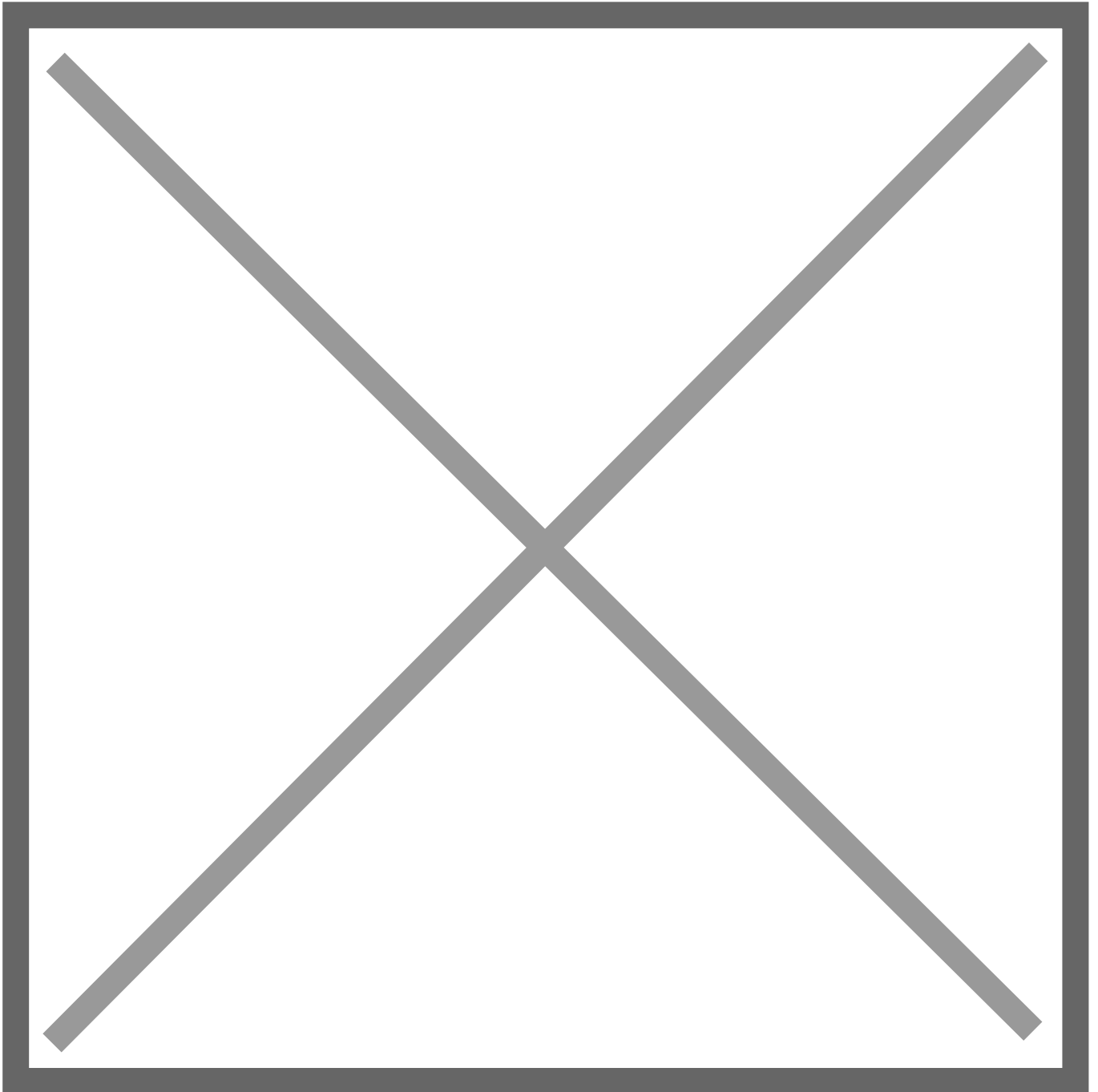
Some points:

- Colors are buttons for color selection dialog

- Intensity sort of controls how much the color is being applied.
- Sky texture is the texture of the sky, you can use moon texture to make it a night map(play with color top, color low and sun color also)

Saving

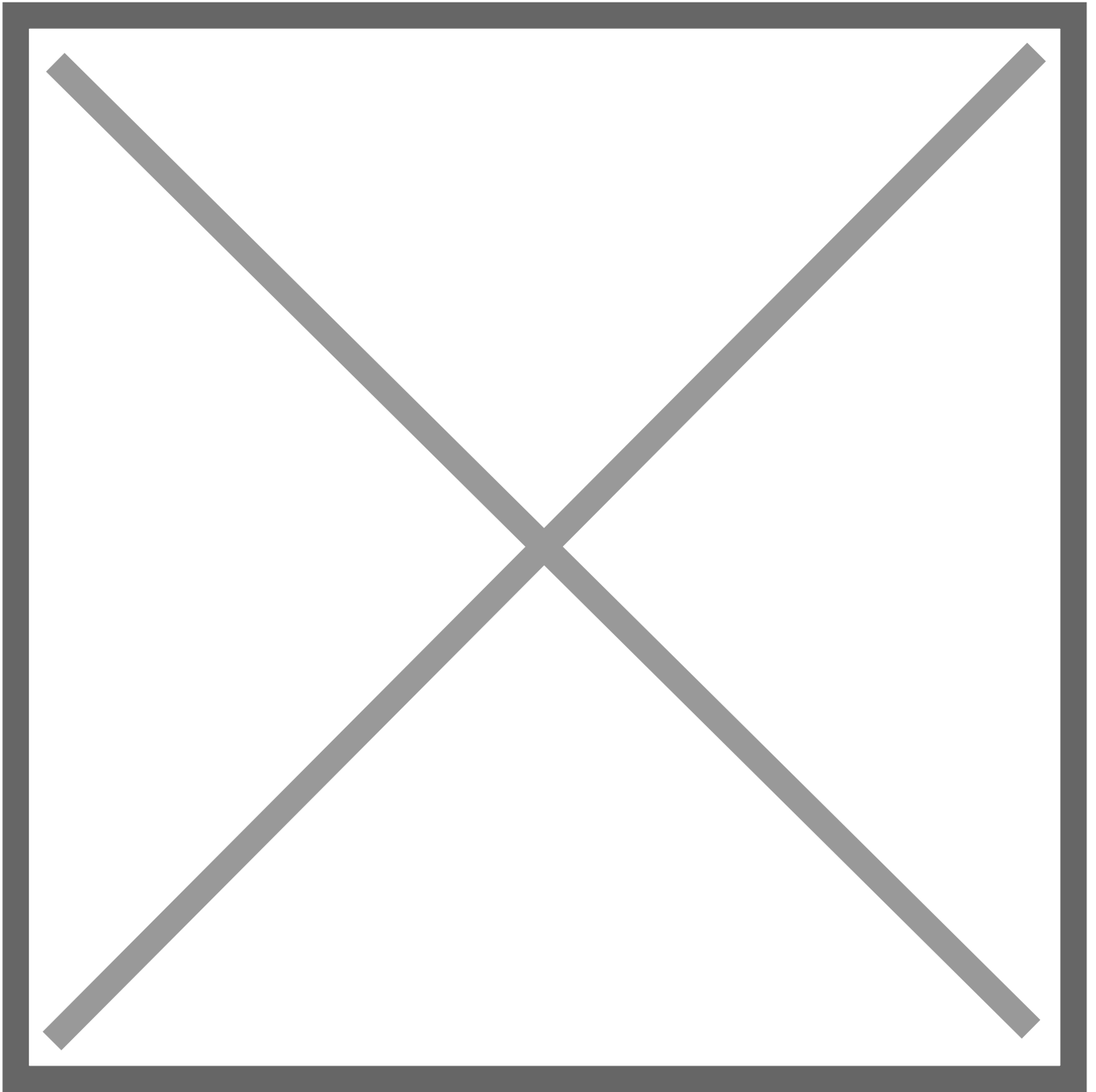
You can save the environment for any use by pressing save, this will open a dialog to fill a file name for the environment(the directory starts from your game directory)



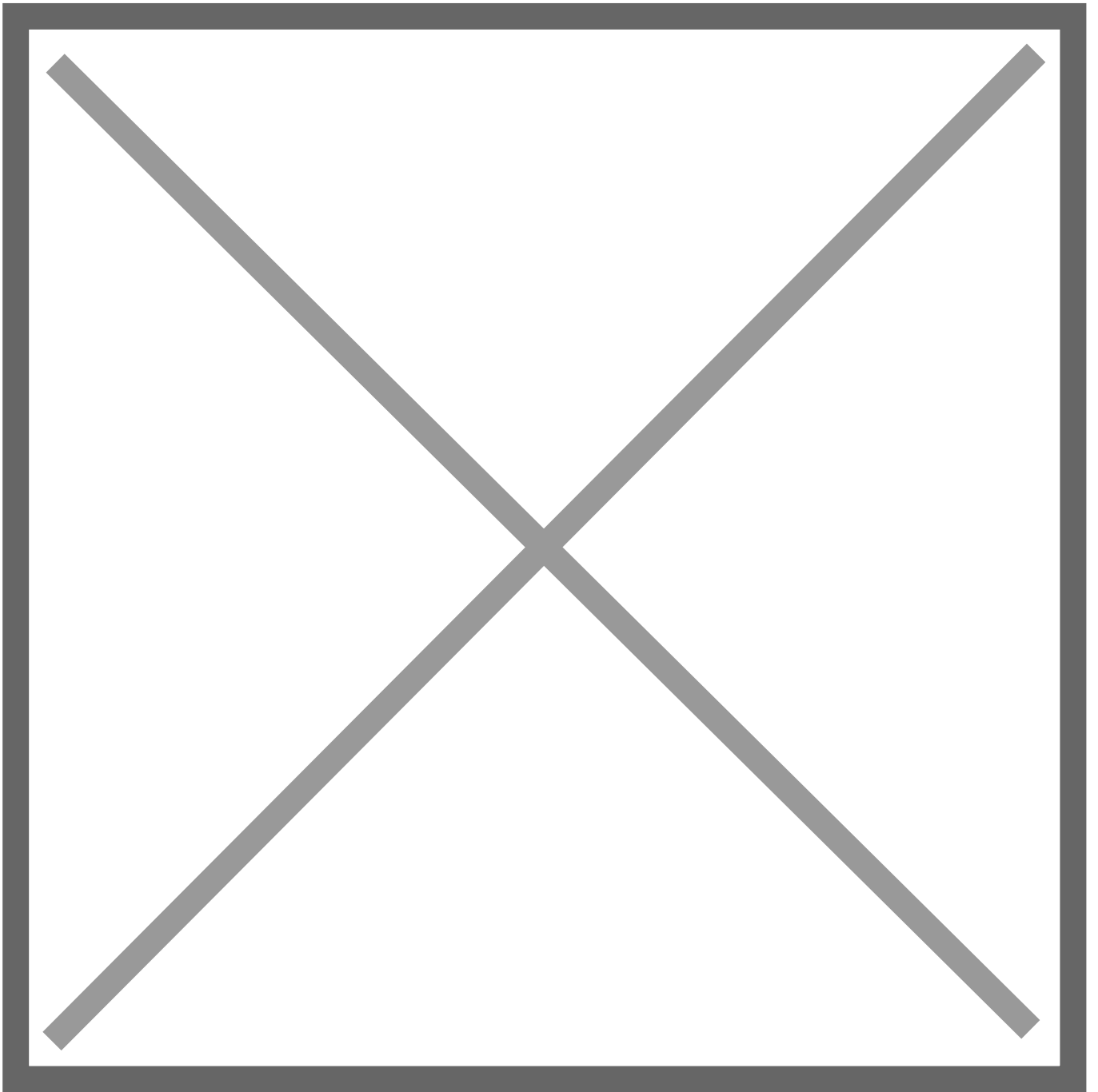
Press and you're done.

Adding to your level(including)

If you wish to include that environment in your level you have to press `Include current`, this will add the environment to your level, which will allow you to use it in the level in an environment area.

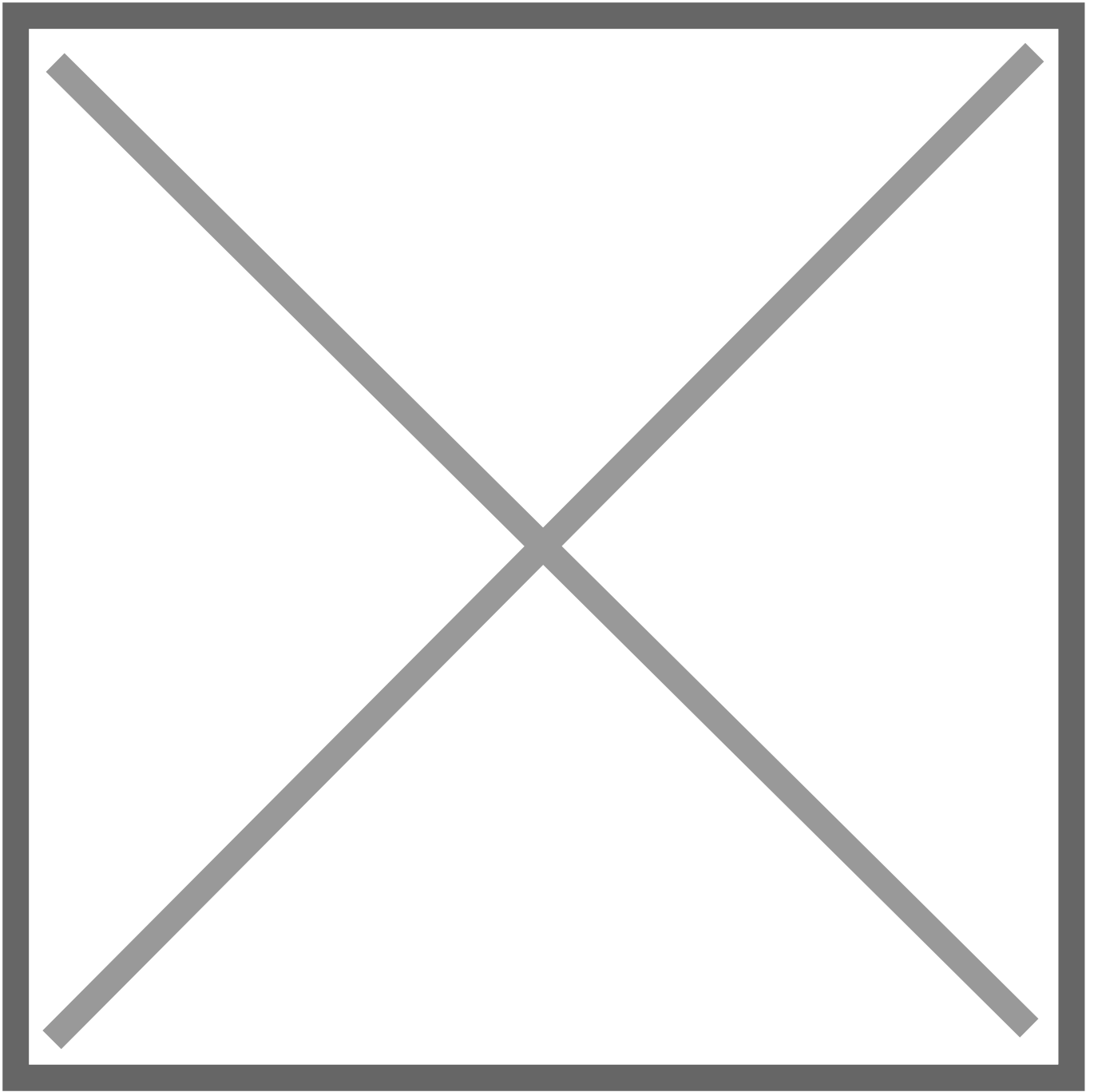


After pressing it, the environment should be added to the `Included Environments`



Using it

Navigation in diesel



Navigation links or navlinks for short are **ElementSpecialObjectives** that have "Is Navigation Link" set to true.

They are used to allow enemies to get from one navigation part to another.



When they are executed they can be used freely by enemies/ai teammates. You need to go to "Manage Access Flags" and select which units should be able to use this navlink.

"Search Position" needs to be somewhere in the navigation area that the enemy should go to after doing the animation.

Optimisation: Occlusion and Portals

What is an Occluder?

Occluders are simple planes that are used to hide props (specifically objects of a unit).

In the model itself `oc_planes` (the commonly name for them) are a one-sided plane pointing its occluding face in the Y-Positive direction.



Occluder planes are set in the object file of a unit.

```
<dynamic_object>
  <occluders>
    <occluder name="oc_plane_001" />
    <occluder name="oc_plane_002" />
    <occluder name="oc_plane_003" />
  </occluders>
</dynamic_object>
```

Warning, occluders are limited to 5 active at a time decided by what occluders are nearest to the camera.



What is a Portal?

Portals are groups of units that are made visible by standing inside shapes.



You first make a Portal in the Portal World menu, Then once a portal is made you can select that Portal. Having a Portal selected and then selecting units in your level will allow you to Add To Current Portal (Or Remove).

Units inside a Portal group will show a red bounding box over that unit to indicate that its in the active Portal.



Units do not have to be inside a Portals shapes to be included, they can be for example Backdrop props that get hidden once you are not outside.

Useful information

Occluders do not have to be in a dedicated occluder unit, if you believe its neccisary they can be included in a prop unit.

Occluders have a hard time updating in the level, You cannot use an Enable/Disable/Move/Rotate Unit Element to toggle or reposition them, but you can use the Delay Load setting on the occluder unit and have it show later using Load Delayed Elements.

Occluders may be limited to 5 closest planes at a time, but if you arent looking at an occluder plane it will not be used.

Occluders will ignore units that use `<network sync="spawn" />` or other forms of network sync. This includes most interactable units or enemies.

Multiple Portals can be used on a single unit, you are not limited to 1 Portal per unit.