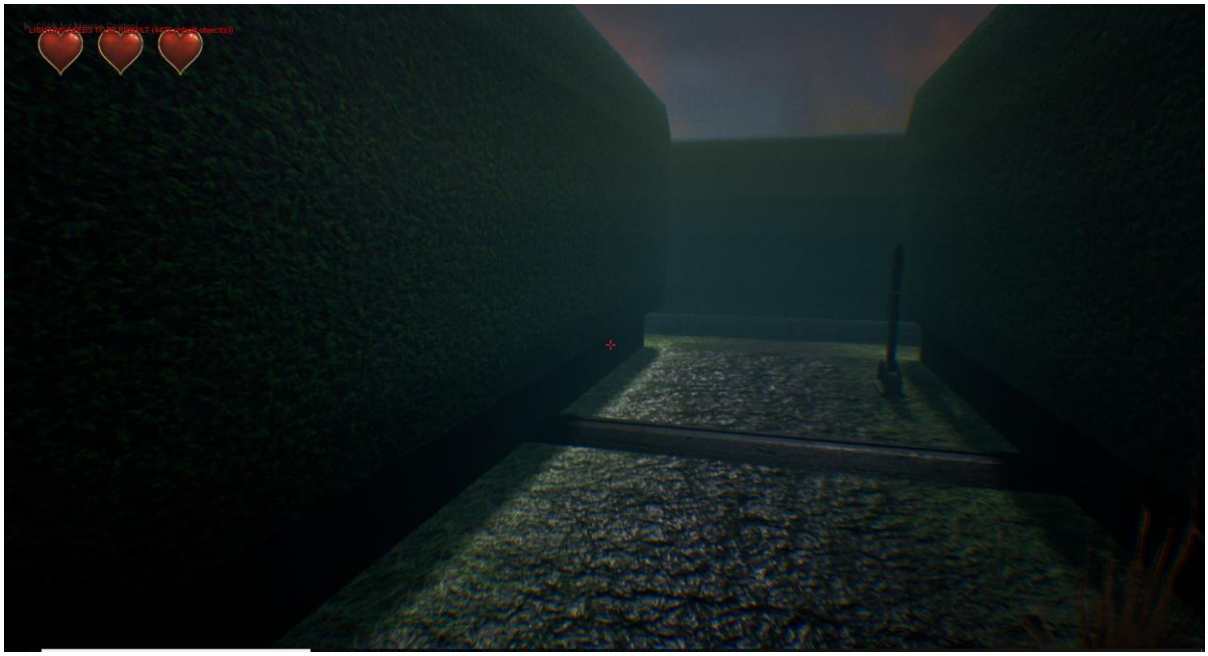
Devils in the detail

Unreal Project

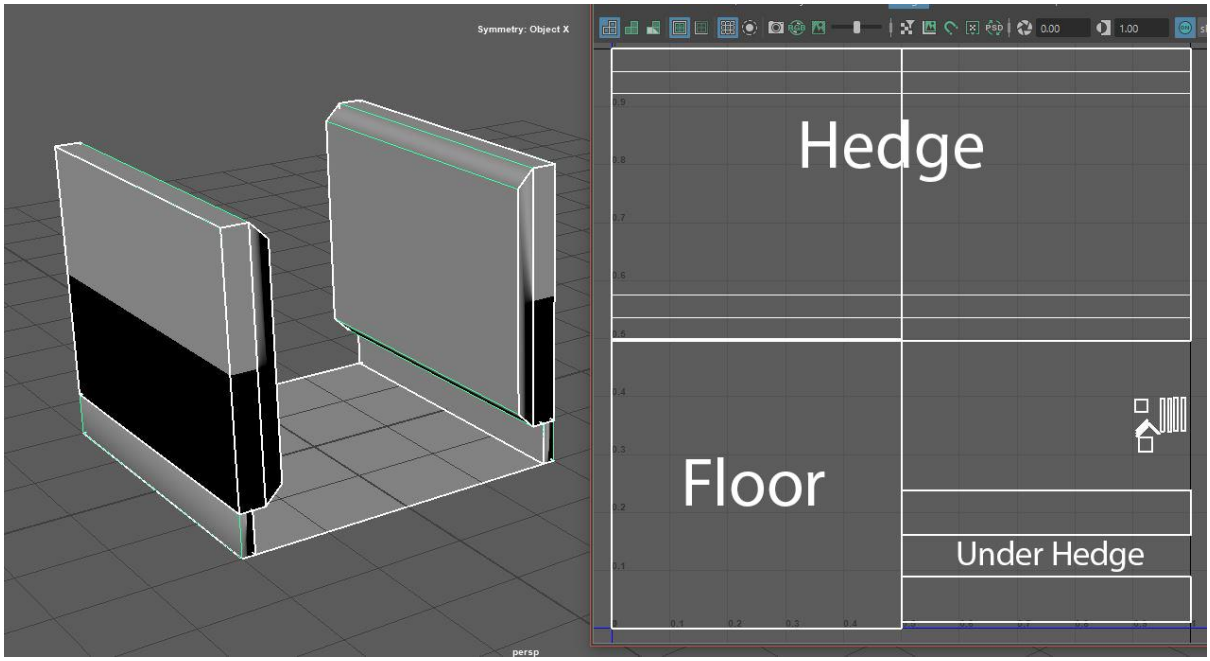


Art

- Modular design and textures
    - Lightmaps
    - Normal Map issues
- Initial Maze Design
- Creating one of the gnomes
- Creating the Monster

Modular Assets

At the beginning I started out blocking modular maze pieces for the maze. My plan was to create pieces that could be interchanged and altered afterwards, allowing for early drafts of the maze to be made in engine; allowing geometry to be updated later down the pipeline and for easier layout alterations.

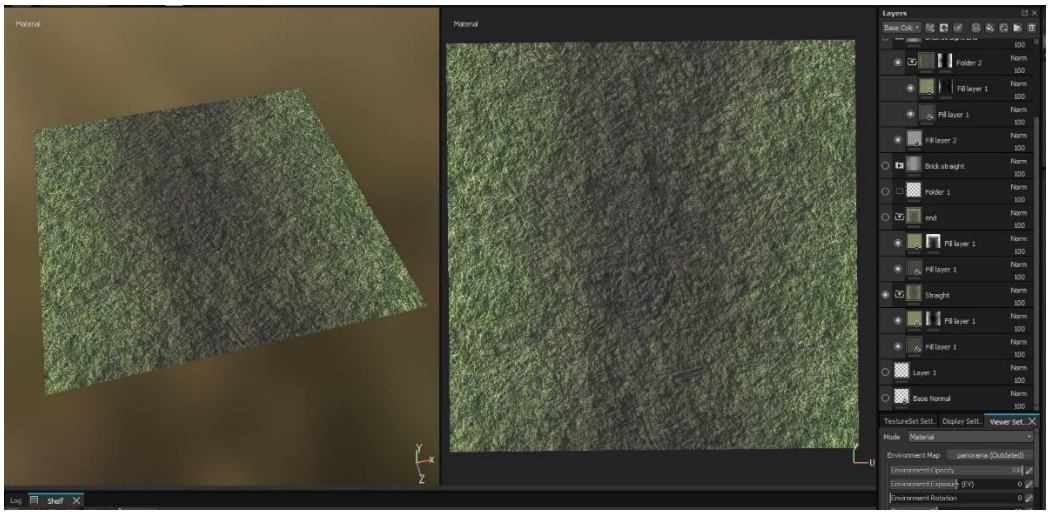


Example of A modular asset

I divided the UV's space into quarters, so that after creating the base materials I could composite them in photoshop in such a way to maintain tillable modular pieces while not using multiple material ID's.

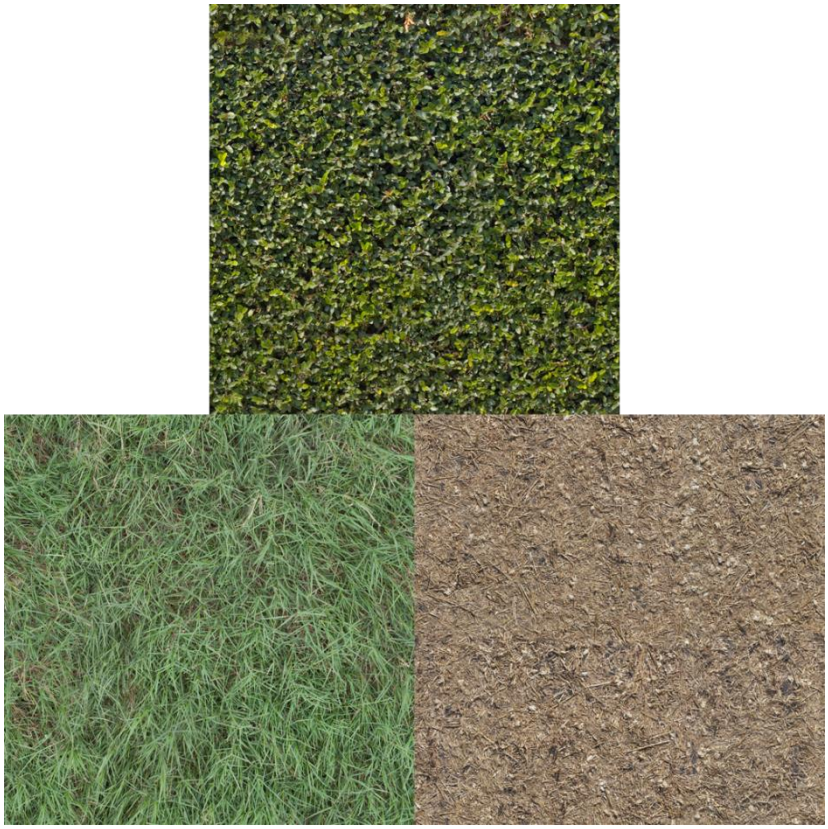
I initially created the modular assets without being water tight, but light would bleed from the opposite side of the mesh. To correct this, I created a Lightmap.

I used based/ reference textures from Textures and from these images I derived Normal, and Roughness maps through BitMaps2Material. I then added these newly created textures into substance Painter to create the desired tiling size. For the ground I used two different materials grass and dirt, blending them together within Substance Painter, I used the mirror symmetry to keep the textures tillable



Original textures

The hedge textures were taken from https://www.textures.com/ . I used the following textures as a base to edit and create the textures I needed (all Royalty free, free to use textures).



Initial Maze Design

I created the original maze design, and showed my group how the modular design works together. I also tried to experiment with maze designs, to created disorientation in the player. Luis remade the maze for the final project, but some of the level design made it into the final design.

The red maze pieces were an idea to make the maze react to completing objectives, by reorganising its lay out (turning to close off routes and create new ones. This idea was later scrapped but turned into the portcullis idea.

Characters

Gnome- **https://skfb.ly/6uWTt**

I created one of the gnomes, used in the puzzle (look below). I made him in Z-brush from dynamic mesh in a T-Pose, I then re-topologised the mesh in Maya, and then created the textures in Substance Painter (including baking normal and ambient occlusion maps). As part of the initial design/ idea each gnome focused on a hear no evil, see no evil, and speak no evil; my model was the see no evil.

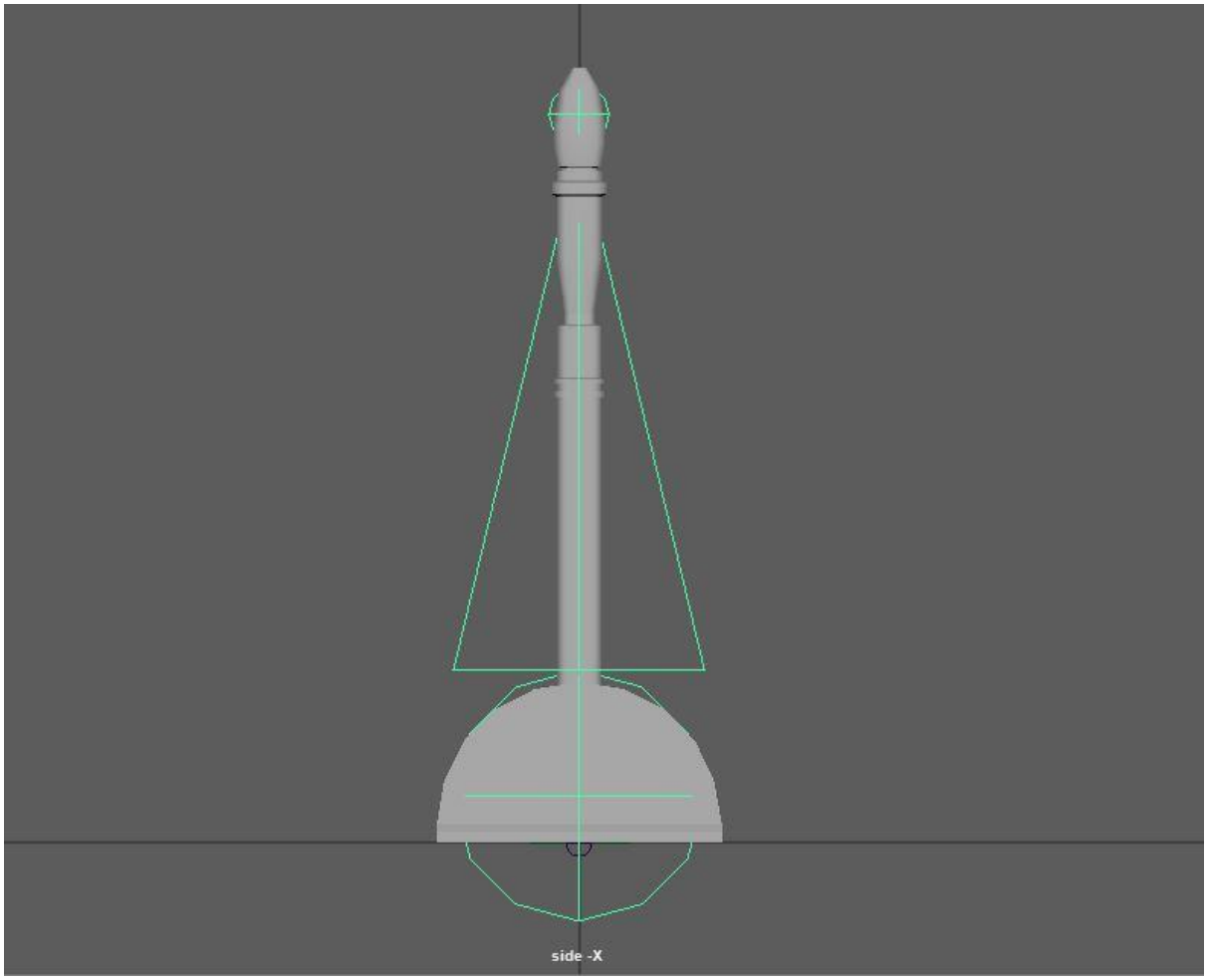
Monster- **https://skfb.ly/6voYM**

I created the main antagonist in Z-brush, re-topologise and UV unwrapped in Maya, and textured in Substance Painter. I created hollow eye sockets for eyes to re-create a nightmare I use to have, and took further inspiration from Death stranding, creating a metallic oil effect seeping from its body using the particle brush

Misc.

***Lever Rigging and Animation*- https://skfb.ly/6vQqT**

As part of the portcullis (see more down in the blueprint section) I had to rig, animate and export an FBX animation of the activation of a lever. I rigged the lever in Maya, initially creating a contrain parent controller which I had to remove as the exported animation didn't show in unreal. So I created an origin (Root) joint and a binding joint atop the root that drives the deformation; this means that the joint on top its parent is on its origin and has 0's on its attribute editor and so is much easier to use as a control.



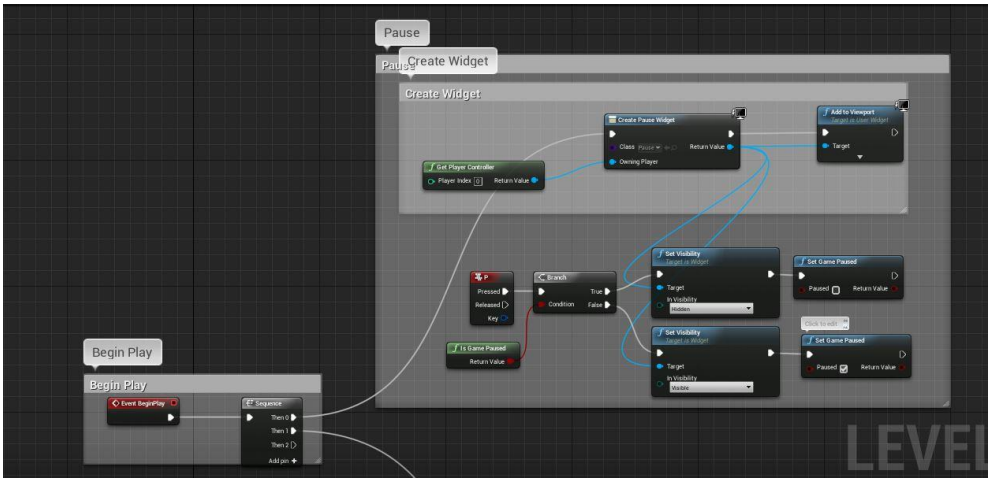
Coding

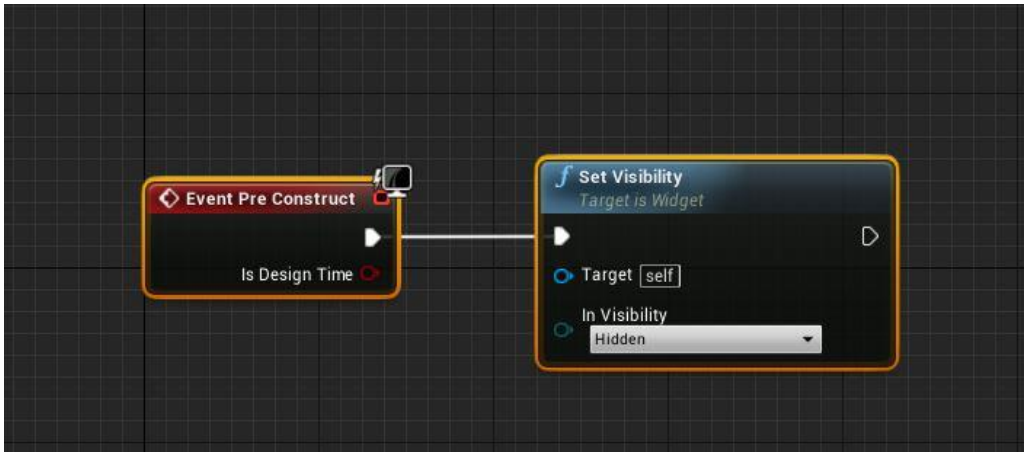
- Pause
- Portcullis

Pause

Implementing the pause system was first blueprint I have created. This was meant to be a minimalistic pause, where it would just pause the game

1. I had to use a sequence node to split the *Event BeginPlay* as the node was already in use for other blueprints.
2. I then used the *Event BeginPlay* Sequence to ready the custom pause widget, which I create, and set to Invisible (Event Pre-Construct->Set Visibility{Hidden})
3. Set an event to the player pressing the “P” Key- which is connected to a branch node, which checks if the game is paused.
4. If the game is not paused, then the blueprint is paused and the widget (Which is set to the player) is made visible. If the game is already paused, then the game un-pauses.





Portcullis

Part way through development we saw that the monster was too relentless, and hard to shake off as a player. We decided that an object needed to be created to block the monster during a close chase.

The portcullis is formed of three meshes (Lever (Modelled by Gabby), Gate (Modelled by Gabby, textured by myself), and the wooden block the gate appears from (modelled and textured by me. In gameplay you approach a lever where a prompt will come up telling you can “Press E to raise the portcullis”. The blueprint was divided into trigger, widget, and animation.

Trigger

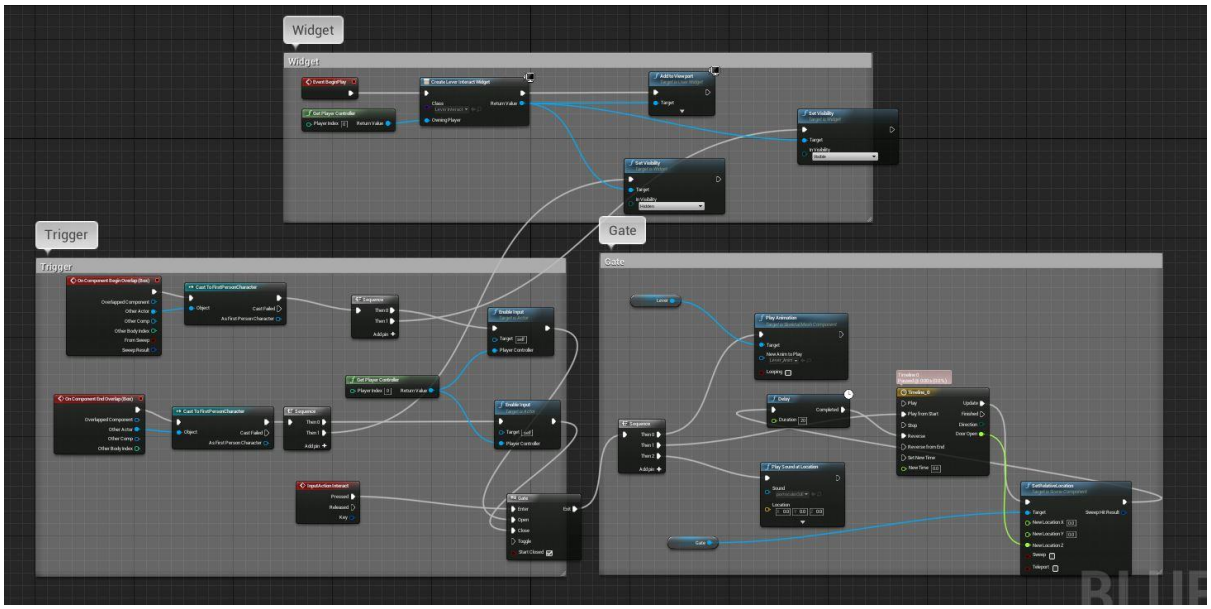
Run on a trigger box start and end overlap, which triggers the widget to be visible (set up similarly to the pause widget above), the triggers are also connected to a gate node, the start overlap opens the gate and the end overlap closes the gate, with the “Interact” button press activating the gate and the Gate animations. The interact button was added by going Project Settings -> Engine -> Input -> Action Mapping. Defining these inputs would be good to use in future game development workflows as it’ll make it easier to re-map a lot of game mechanics, instead of re-entering various blueprints to individually alter them.

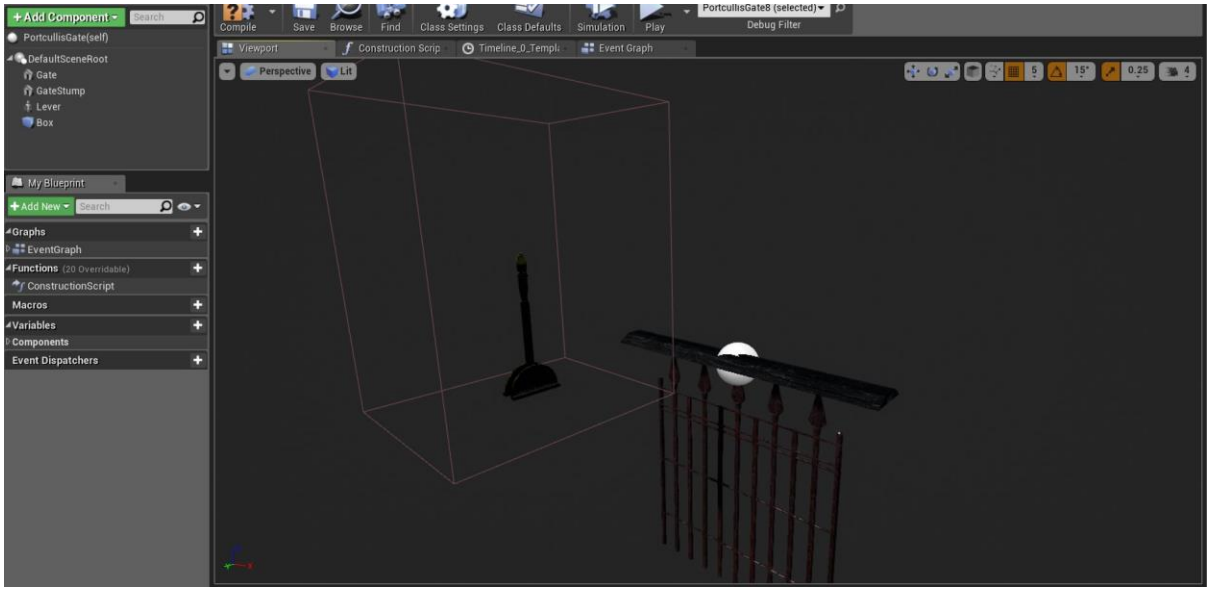
Widget

I reused the Pause widget method, but with a new widget.

Gate

I used a sequence to split the triggered action, one triggers the lever animation (which I rigged and animated), another to play the timeline (animation for the gate), and the last for the sound. The timeline is linked drives the *SetRelativeLocation* of the gate, causing it to raise, which in turn is linked to a delay node which drives the reverse of the same timeline resetting the gate to open, and making it easier to iterate how long the gate stays close for.




+Add Component
Search
PortcullisGate(self)
DefaultSceneRoot
Gate
GateStump
Lever
Box
My Blueprint
+Add New
Search
Graphs
EventGraph
Functions (20 Overridable)
ConstructionScript
Macros
Variables
Components
Event Dispatchers
Compile
Save
Browse
Find
Class Settings
Class Defaults
Simulation
Play
PortcullisGate8 (selected)
Debug Filter
Viewport
Construction Scrip
Timeline_0_Templ
Event Graph
Perspective
Lit