

Level
Design
Portfolio

Nyctophobia

Portfolio of the planning and development
documentation of the level and steps done to lead to its
creation.

Craig Brown
2017

Contents

Planning of level concept and Research	3
Research	4
HUD	4
Level layout	5
Top floor.....	6
Basement	6
Objectives	7
Pickups.....	7
Player.....	7
Enemy.....	8
Previsualisation	8
Flowchart of level	9
Development	10
Notes taken during development.....	10
YouTube Link	13
Screenshot of HUD final design	13
Screenshots of blueprint code	14
ThirdPersonCharacter.....	14
Enemy	18
Pickup blueprints	20
Level Blueprint.....	21
Objectives	23
Widgets.....	23
HUD Widget and blueprints	24
Audio Cue	25
Evaluation	26
Resources and Sources.....	27

Planning of level concept and Research

The goal of the assessment was to conceive and develop an original game level utilising Unreal Engine 4. The level was to make use of game engine tools and my own scripting interactions. Although I could create my own art and audio assets the main focus was on that of the scripting but if I was to source anything that isn't my own then I'm too clearly credit the source.

The main aims of the level are to include the following;

- Triggers
- Scripting/Custom Logic such as;
 - Events/ Custom Events
 - Variables/ Arrays
 - Flow Control/ Loops
 - Functions
- Basic Animation/ Movement within the scene
- Collectable/Destructible/Moveable objects
- Implementation of basic, functioning UI
- Implementation of basic art and audio assets (sourced or my own) within the scene

The level is too based off of one of the following creative briefs;

- Autobiography – something that represents myself whether it be mundane tasks I do on a daily basis or an experience during the semester or another point in my life
- Memory – Something inspired by a memory of a space in time and uses this to convey how memories are often imperfect and be selective.
- History – a project that represents a historical event.
- Natural World – a level that can translate concepts, ideas and knowledge from the natural sciences such as physics and chemistry
- Psychology – a project inspired by psychological literature with the aim to get a reaction from the player with research into the likes of phobias and emotions.
- Adaptation – create a level adapted from a popular piece of media such as a poem, comic, film, ect.

In this case I have decided to base my level idea off of the Psychology brief. I will be doing some research into phobias and other psychological problems how they can affect a person and see how I can visually imply this. Any side effects of the psychological condition could be implemented as a game mechanic perhaps.

(UPDATED) As a way of saving time I primarily used assets found in the Starter Content pack in UE4 to help make things in the level as well as using Maya 2018 for making small 3D models for things such as the batteries and the keys as I tried looking online for good free mesh assets but I could only find ones that cost a lot of money, however I sourced some good sound assets and created some of my own.

Research

I've decided that my level is to be about someone that has a phobia of the dark, known as Nyctophobia (also known as Achluophobia, Lygophobia and Scotophobia to name a few). Nyctophobia is usually found in young children, normally around 10 years old or below, and at these ages it can be overcome by the use of nightlights but for past this age it can become more difficult and if untreated and worsen, can even dictate how a person lives. There are even physical symptoms of nyctophobia including;

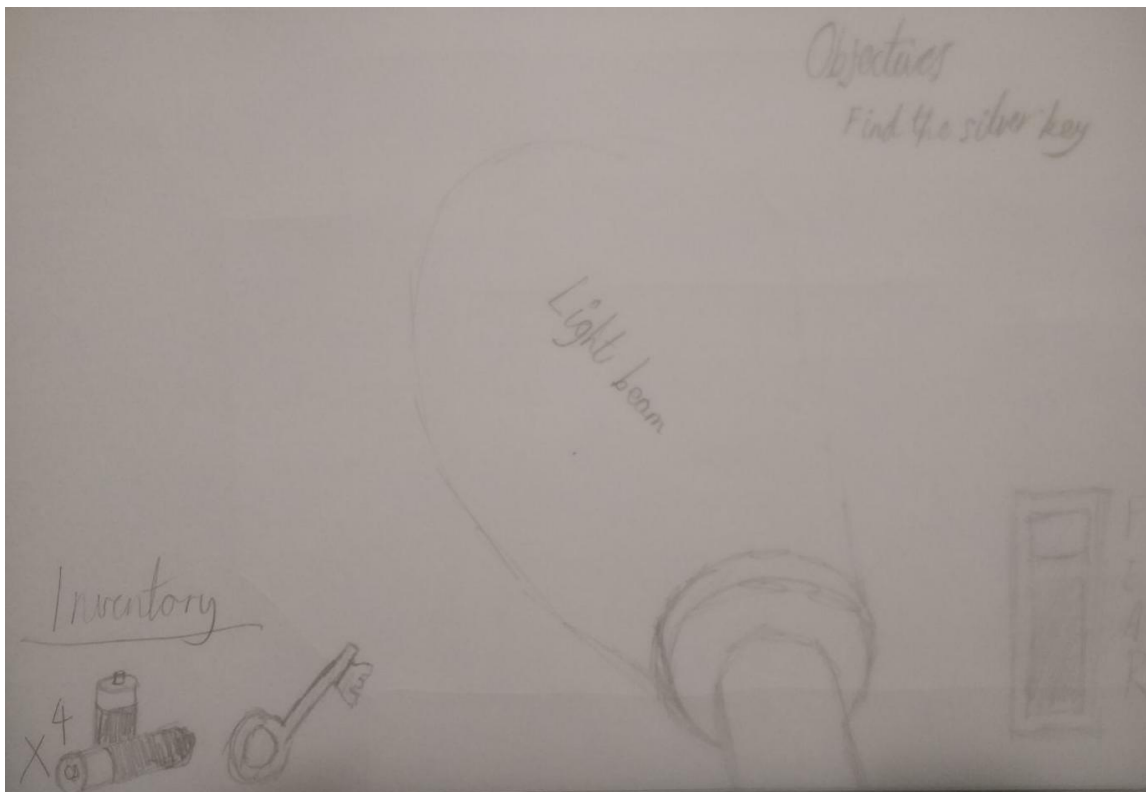
- Panic attacks
- Shortness of breathe
- Dizziness
- Trembling
- Sweating
- Accelerated heart rate
- Chest pains

As a way of trying to incorporate some of these symptoms I'll try and turn some of them in to game mechanics such a fear metre and maybe a weak stamina bar of sorts.

HUD

The HUD UI can be created using the widget blueprint that we were shown in one of our classes. For the keys to appear in the inventory section of the HUD then I'll need to link the widget and the pickup blueprint of that specific key, this can be done by adding the "Add to viewport" node in the key's pickup blueprint and a Boolean for if the key has been picked up or not.

Below is a rough idea of how I'm going to set up the players HUD:

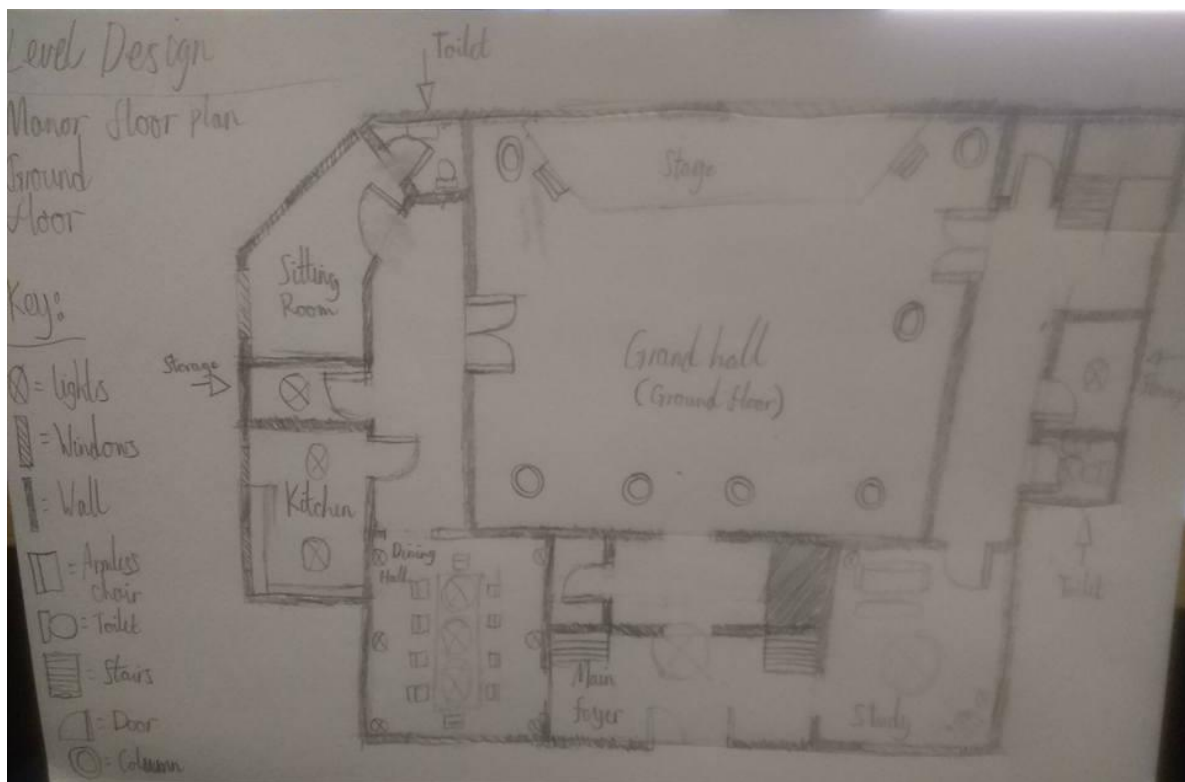


Level layout

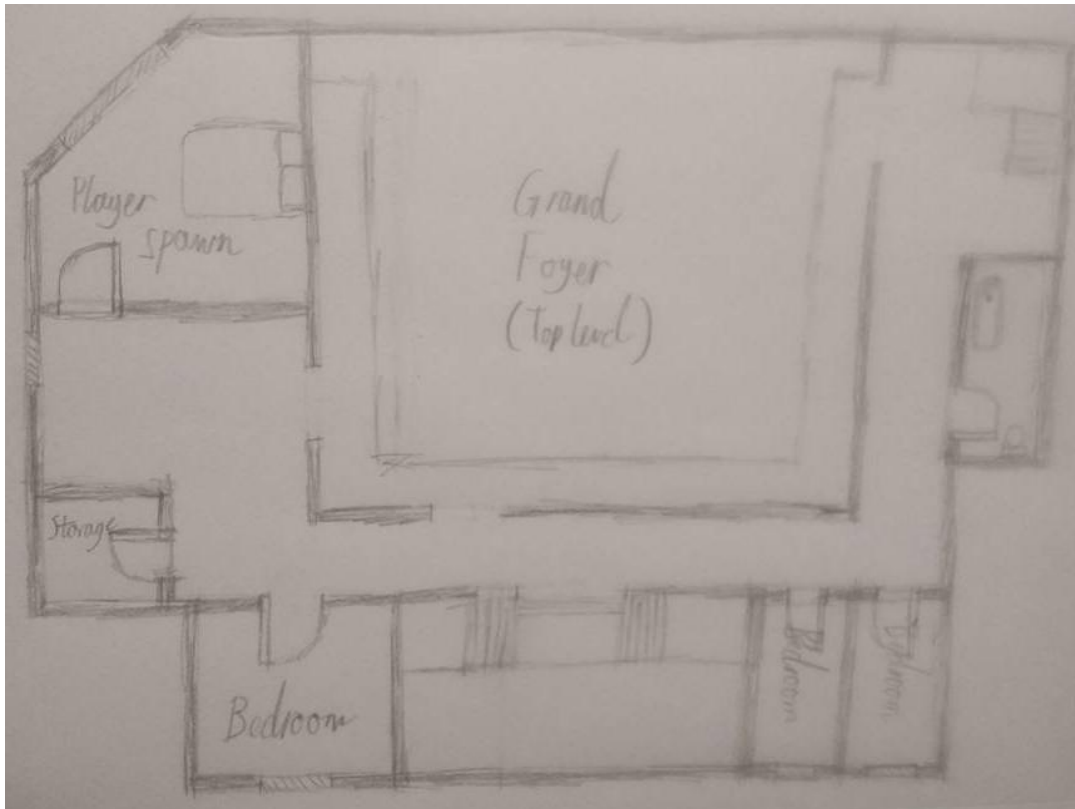
The level will be set inside an old manor to which the player, suffering from nyctophobia, has to stay in for the night however the lights go out and then they're fears kick in and they have to go restart the manors mains electricity which is located in the basement. Throughout the level will be batteries to keep the players torch lit and the player will have to collect 3 types of keys; a bronze, a silver and a gold key to open their respected doors with the final door being the gold one and then the player completing the level. In each room of the manor will be a set of batteries the player can pick up and a key will be in a random room so this will require the player the search every room. With this the enemy will be able to patrol around the level looking for the player and once the enemy sees the player they will give chase and the player must get out of the enemy's line of sight and hide until the enemy stops roaming around looking for the player and then returns to its patrol pattern. This can be done using custom events and a state switch. There will be enough rooms for the player to hide in and an open space to try and use to help lose the enemy in try and escape.

(UPDATED) I decided the manor idea wouldn't work out too well however I kept the layout and instead created more of an art gallery looking level. I removed the top floors front bedrooms and made them open spaces for art exhibits and changed the players spawn from a bedroom to a security room with a window looking into the large hall in the centre of the building. The kitchen was changed to be the stairwell down towards the basement and thought that rather than having the basement modelled to save time, I edited the gold door to play the ending screen instead. All of the smaller rooms have batteries in them and can be opened using the bronze key and the large storage room at the far right has a blinking red light to show the player that its the room with the silver key in it (an objective also appears when the player picks up the bronze key, indicating they have to go to the blinking red light). These are some basic sketches of the floor plan for the manor from top level to the cellar:

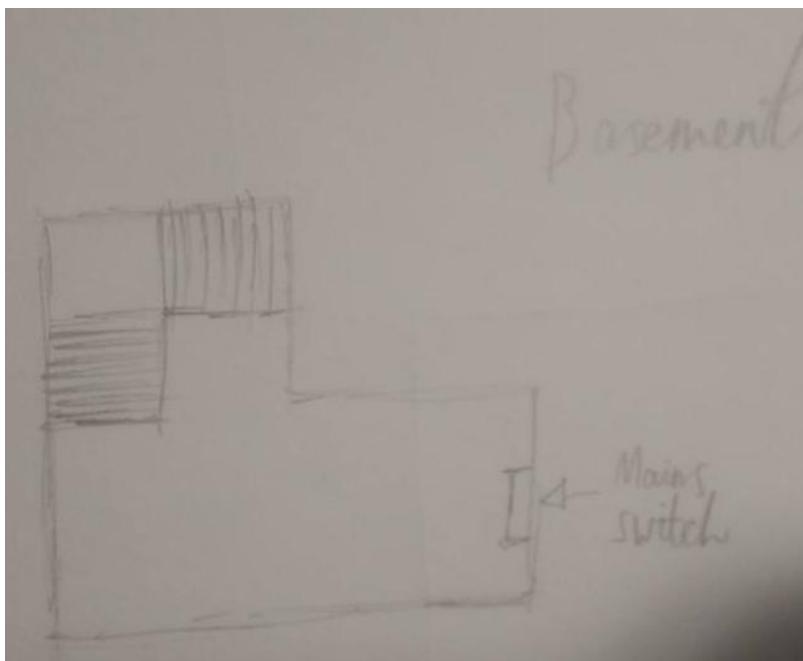
Main floor



Top floor



Basement



Objectives

The objectives will need to come right after one and another but this can be done by creating a blueprint that is linked with a text box underneath the objective tab in the players HUD UI. With changing objectives, the player will know what their task is and once the player completes a main task, such as picking up one of the keys, they will get an objective complete message that will appear in the centre of the screen. This can be done by using another widget blueprint and the same can be done with the player screen and the end screen.

(UPDATED) I wanted to have the current players objective to save if they were to exit the level and found out how to do the code but decided again to leave it out but left in the code for if I ever wanted it, I added in an objective just underneath the players spawn so that it corrects the objective each time the player spawns in.

Pickups

The pickups in the level will have their own blueprints for when the player enters their trigger box, the blueprints will all need to be linked to the thirdpersoncharacter's blueprint to allow for values to be exchanged. In the HUD widget, to help display the image of the correct key that has been picked up a switch on int will be required. For this, there will be 4 IDs;

- 0 = blank
- 1 = Bronze key
- 2 = Silver key
- 3 = Gold key

This will help with the linking of values in the Pickup blueprint.

Player

This player will need to have code that will allow them to sprint, the ability to lower and increase the fear metre and use a flashlight. For these actions all be using key pressed events so Left Shift for sprint, F for flashlight and E for when the player needs to lower the fear metre. When the flashlight is on the battery life will need to lower as well. Integers can be used for increasing and decreasing the fear metre and the variable MaxWalkSpeed can be used to help with the different speeds of the player's movement. In the players viewport I'll attach the camera to the player meshes head and adjust the camera to a fitting position to give the game a 3rd person aspect and include a shadow for the player. The player being the only human shadow because the enemy will invisible implying that there isn't anything physically there, noting back to my thoughts when I had a fear of the dark and the thoughts of others that have nyctophobia. If the fear metre reaches 100 then the level ends with the player having a heart attack due to fear, the same goes with if the enemy catches the player.

(UPDATED) I added in a rechargeable stamina system that can be charged faster using the E key when the player needs to breathe however if the player sprints they increase the fear meter. I also added in the ability for the player to move at a walking pace using the Left Ctrl button.

Enemy

The enemy AI will be a copy of the player blueprint with the camera, controls, the players own features and elements taken away. The enemy will need to chase the player, roam the level if they lose the player and follow a patrol route when if they don't find the player after a set amount of time. The patrol can be done using target nodes for the enemy to walk too and a NavMeshBounds Volume which, when the enemy roams and chases the player, can be used to come off of the patrol route and chase the player properly. The change in states can be done using a 'switch on' using an Enumeration where as the events themselves will be custom.

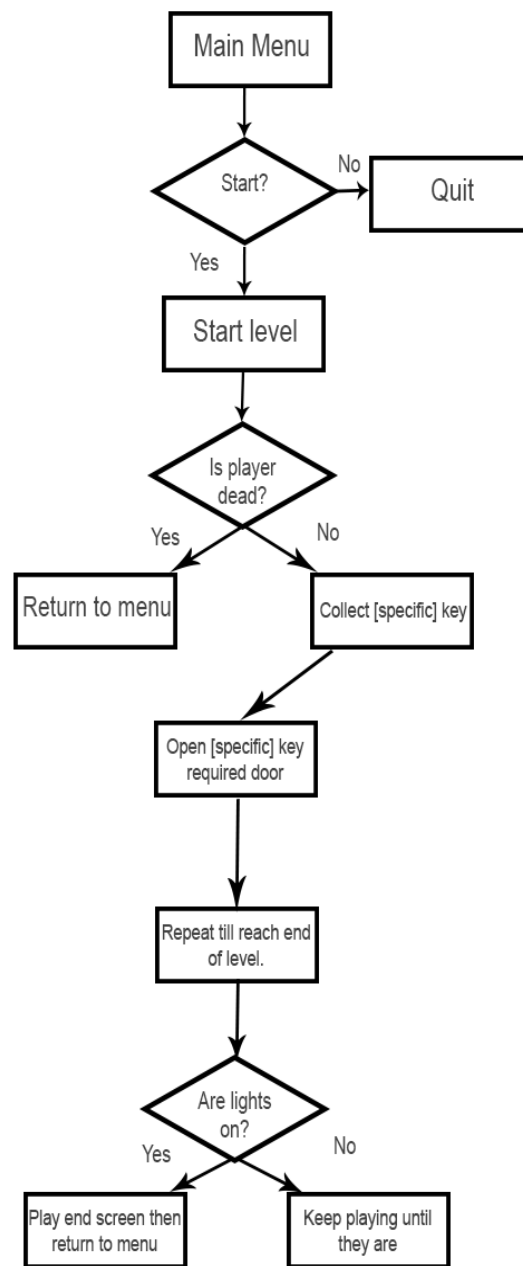
(UPDATED) The enemy doesn't roam as well as I had hoped however it still does roam none the less.

Previsualisation

NOTE: Everything talked about after this point will all be the updated version of the art gallery idea instead of the manor idea.

The player begins by spawning in the security office next their computer desk, the lights have gone out and they are startled. They get up and begin to collect some batteries and grab the copper key found in the room. The building is old, creaking with the wind wailing outside causing the players fears to begin to rise but they try to steady themselves by breathing long deep breaths. From the security room the player can see that not all the lights have gone out as a spotlight shines on a statue. As the player leaves the room, the spotlight short circuits and turns off and when the player turns on their flashlight they can see the statue it gone. When they look down the dark empty hallway as the fear rises, the player can see the yellow glow from the enemy down the hall as it begins to wander the halls, the floor boards creeping under each footstep. The player then walks down the hall to the blinking red room to collect the next key whilst trying to avoid the enemy. Once they collect the silver key they have to search for the silver door. If they're flashlight runs out of battery they have to try and locate batteries in complete darkness all while trying to still hide from the enemy. If the player is spotted the spotlights will aim at the player and blind them as the creaking floor boards suddenly turn into heavy thumping noises of something running. Once they find the silver door they can get the golden key and go down the stairs and finish the level.

Flowchart of level



Development

Notes taken during development

The development of the final level took around 12 days to make, during this time I took notes of what I created or had begun on specific days and about any problems I may have had and if I had managed to fix them. I used it as a way of keeping an idea of what I had done and if there was any errors from the previous night that I might forgot the next day or if I encountered a similar error I could look at the notes to find out how to fix it.

Day 1: Managed to create some code that allows a mesh to change into something else when the player walks into a trigger volume and when the player exits the trigger volume to mesh switches back to its original. I also added in code that allowed for lights to turn off as the player enters the trigger and a delay was added to all for the lights and mesh to change back to normal. This will work for when the lights in the house turn off and then the players fear meter increases.

Day 2: Created the HUD's final idea in photoshop (screenshot taken). Decided to switch from the first person blueprint to the third person blueprint as I found out how to make the camera go into a first person perspective and this allows a shadow of the fully body rather than there being no shadow at all. This also allows for some bobbing of the head as well to give the player a more immersive experience.

Day 3: Imported the elements that I'll be using for the HUD into the HUD_widget blueprint and locked them in their respected position so that when the player plays the level on a different resolution, parts of the HUD aren't in random positions. I've set up a basic 'fear' metre that increases by 5 every 5 seconds from 0 to 100 and once it reaches 100 the game pauses and a message appears. Changed the settings to the fear metre so it now increases by 2 every second and made the metre a progress bar to allow for the metre image to show the increase and to show the player how long they have until they get a game over. Edited some of the camera settings to allow for a motion blur effect as well as including a sprinting feature with bar that increases at a slow rate to create tension and lowers quickly when the player uses it (Left shift). Torch feature has been created, the torch is turned on using the F key but after a few seconds it runs out of battery so the player has to collect batteries and conserve battery life until they find it. A simple example of getting a new objective has been made as well. A 3D model of the batteries has been made as well for when the player needs to collect batteries. Imported the batteries and made some code to allow the player to pick them up and get 25% of battery life back. Enabled the batteries to rotate on the spot and scaled them up to see easier. Only problem with batteries is if torch is 100% and the player runs into new batteries, the new batteries disappear (trying to set it so the player can't pick them up and run through them instead until they need them).

Day 4: Edited the batteries so that when picked up they go back to 100%. Implemented the beginning of an inventory system for when the player picks up different types of keys but for now the only pick up is the batteries. Edited sprinting and the fear metres so that when the player sprints

the fear metre increases due to the player's character panicking. I also lowered the default speed to a more walking pace rather than a jog like pace this way the player will either walk or sprint when navigating. Edited the metres further so that the fear meter increases by 2 every 3 seconds but when the player sprints it increases to 1 every second to make it a bit fairer on the player (might try to implement a way to include the E key input to act as a coping mechanism like taking deep breaths to try and calm down).

Day 5: Created a key model using Maya 2018 and created a bronze and silver material (there is a gold material already made on UE4) for said key using the material editor in UE4, the same key model will be used. The keys will be saved in the inventory so they only will have to be picked up once and open multiple doors. Added controls to the left Ctrl button to allow the player to move at a walking pace, this can help when looking for keys. Edited the inventory system so that when one of the keys is picked up it shows the picked key in what inventory slot is free (this can be useful for future games or if I want to take the game further after the hand in date). The keys now rotate on the spot much like the batteries and were scaled up a bit further so they are easier to collect. Edited the code for when the player presses the E key that they player is slowed down and has to press the key multiple times to lower the fear bar and increase the stamina bar. Edited the objective code to save when the player completes an objective rather than just changing the text for when the player quits the game.

Day 6: Implemented and edited the code for when the player picks up the keys they will get a notification as well as when they complete an objective they will also get a notification indicating that they have completed it. Commented most of the code to show what each piece does and screenshots have been taken for evidence. Duplicated and edited the GameObjectives blueprint to allow for me to start making the level properly. Edited bronzekey_pickup blueprint to play the objective complete animation, I'll do this for the other objectives in the future. Begin creating the AI, it's originally a copy of the player blueprint with the same mesh but I've edited the mesh to appear more menacing when in a dark corridor. The enemy, since it is a player copy, includes the mannequin which I made invisible instead and to allow the player to see where the enemy could be I included two spot lights coming out of two glowing balls that will pretend to be the eyes of the enemy. I believe this is a good way of interoperating the enemy as the player isn't supposed to know what the enemy looks like, this can add on to the thought of paranoia of the player as well as make it easier rather than just creaking footsteps. Made it so that the AI can roam around the map using NavMeshBoundingVolumes.

Day 7: Rather than roaming around randomly, I've changed the code to allow the AI to follow target points instead. This will be much better since the AI will be roaming the halls. The AI can now chase the player and when the player is out of its line of sight, it returns to following the target points. Set it so that when the AI is patrolling at a walking speed (Movement speed: 100) but when the player is spotted the AI runs after them (Movement speed: 600).

PROBLEM: When the AI is chasing the player sometimes they "give up" for a second and run to what I'm guessing is one of the target points but then they come right back. It's not a major issue considering the player will be running away most of the time but it's still a bug.

FIXED?: (AI no longer seems to give up and is rather persistent now that it genuinely scared me a few times by catching me off guard).

Day 8: Created manitees for when the player opens doors. I've made some code to allow for certain doors to open only if the player has a certain key. I've added a spotlight over the pickup items to allow them to be seen easier in the level. I've also made it so that the AI now has a spotlight so that when the player is in a room they can see the spotlight from underneath the door, this can help if the door doesn't have a window and see what way the enemy has gone past and too look cool. I've created an end screen and button for returning to the Main Menu which I have also created with the ability to exit the game and start the first level. I've added sound to the doors and music to the level and main menu. I've also changed the material of the doors that need to be opened with key to correspond with the required key.

PROBLEM: Door animations work the first time then seem to just cut to either being open or closed when you run into the trigger_box. They seem to respond with the correct action but the animation and sound just stops. I tried changing the door manitees to just be one so the animation plays the door opening then closing but they only seem to open once.

FIXED: Found out about the reverse node that is used in the for manitee actors in blueprints so I edited all the door animations to just open so when the player hits the trigger box, the door opens and then after certain delay the door manitee is reversed, closing the door. With this the animations work fine and the door can be opened again.

Day 9: I've been trying to implement more audio to the game and editing and mixing some of the audio clips I sourced from freesound.org to create an appropriate walking sound effect. I've went into the third person animations and added in sound so with each step the player and AI now make a creaking noise. Mixed and edited some more sound effects and implemented them into the level using the sound cue blueprint and attaching it to an Ambient. Created my own sound effects such as wind, collecting keys and the enemies chasing wrasp and added them to the level.

- 1. PROBLEM:** While trying to find a way to slow down the animation of the walk cycle I've managed too total wreck the animation as a whole so now have had to remove the players and the AIs ability to move slowly as trying to find a way to get different audio tracks to not lap over with each other in game is too time costly so the AI will just run from now on.

FIXED: I copied the third person walk from a new third person project and used the file manager to copy over the .usset as trying to drag it into the current project wouldn't work.
- 2. PROBLEM:** Trying to destroy the enemy when the player hits a certain trigger box. I worked the first time but the enemy would then just stand in its spawn location and wouldn't do anything. The second time the enemy wouldn't destroy and on my third try the enemy doesn't destroy itself but it will spawn a new enemy (I could maybe use this in the future but for now it would make the level too difficult, was also rather scary).

Day 10: Built the level using the standard content pack since I couldn't find any good free assets because all the good ones cost money. Decided to change the story of my level to be about a security guard in an art gallery which was once built on an old burial site rather than just a person in a haunted manor, this way I can make the level more interesting rather than being blank since I don't have any housing assets like beds and kitchen appliances. The level has now been made much more interesting. I fixed some lighting and texture messes, made use of the materials and meshes from the starter content folder to make some things like statues and modern art.

Day 11: Finished creating the level now to implement the objectives, target points and some small editing to the code. Implemented the key specific doors into the level and made a trigger that when hit will turn off all of the lights and make the statue disappear. Edited most of the spotlights to be moveable and deleted most of the point_lights to make the map much darker. Went back to the first person project we made for at the start of the year and implemented in the blinking light code for the red light. Implemented and edited the objectives of the level and asserted them into their correct areas.

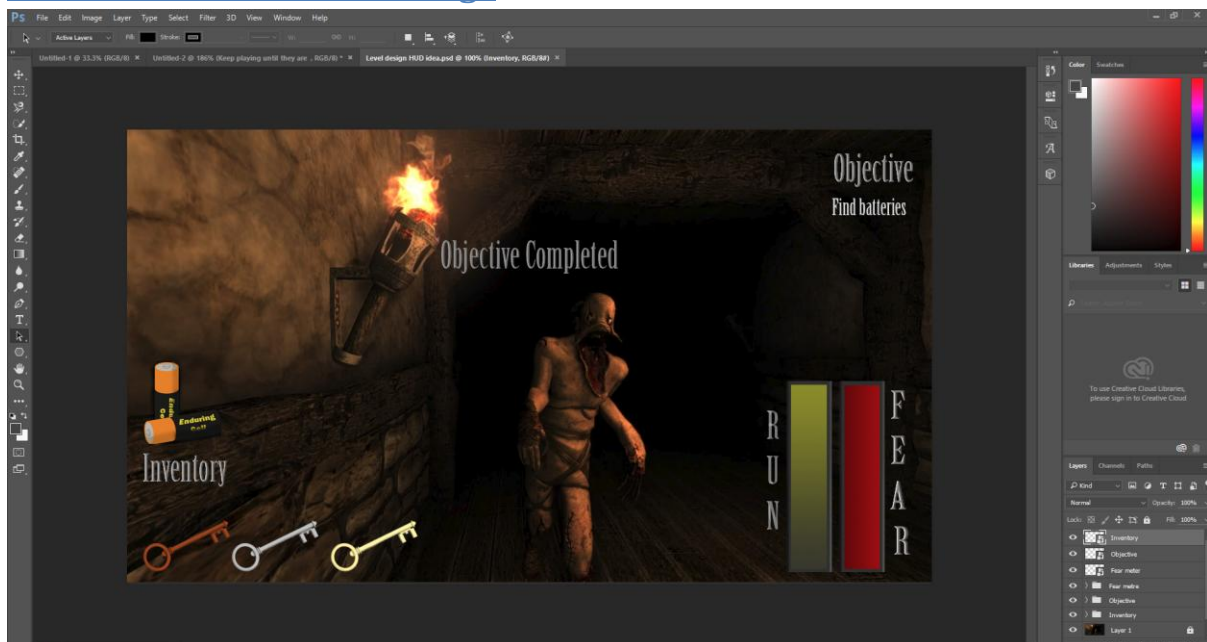
Day 12: Edited the size of the door frames and lessened the battery life decay to give player more of a chance to see in the dark. Added in a lamp to show off the statue in the dark and added in particle effects such as sparks then when the lamp goes out, smoke appears and a sound effect is played. Created a video of the level, in the video I've shown a play through of the level, the player being chased and killed by the enemy AI and the player death by the fear meter.

PROBLEM: For some reason the silver key is automatically picked up when the level plays. Unknown what causes this because the key and the player are on separate sides of the level.

FIXED: I'm still not sure what was causing the problem but I think it might have had to do with something in the room with the key, perhaps the batteries were conflicting with the key or something but I moved it further to the front of the room and it seems to be working properly now.

YouTube Link: <https://www.youtube.com/watch?v=8MKIGzEx1WM>

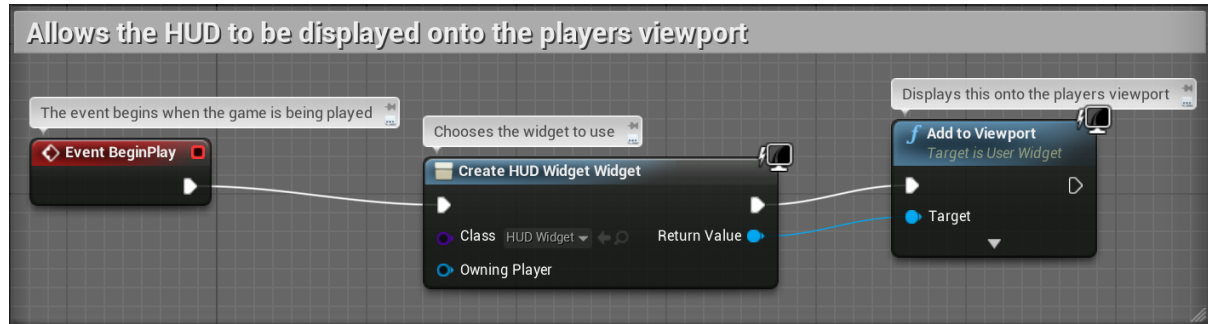
Screenshot of HUD final design



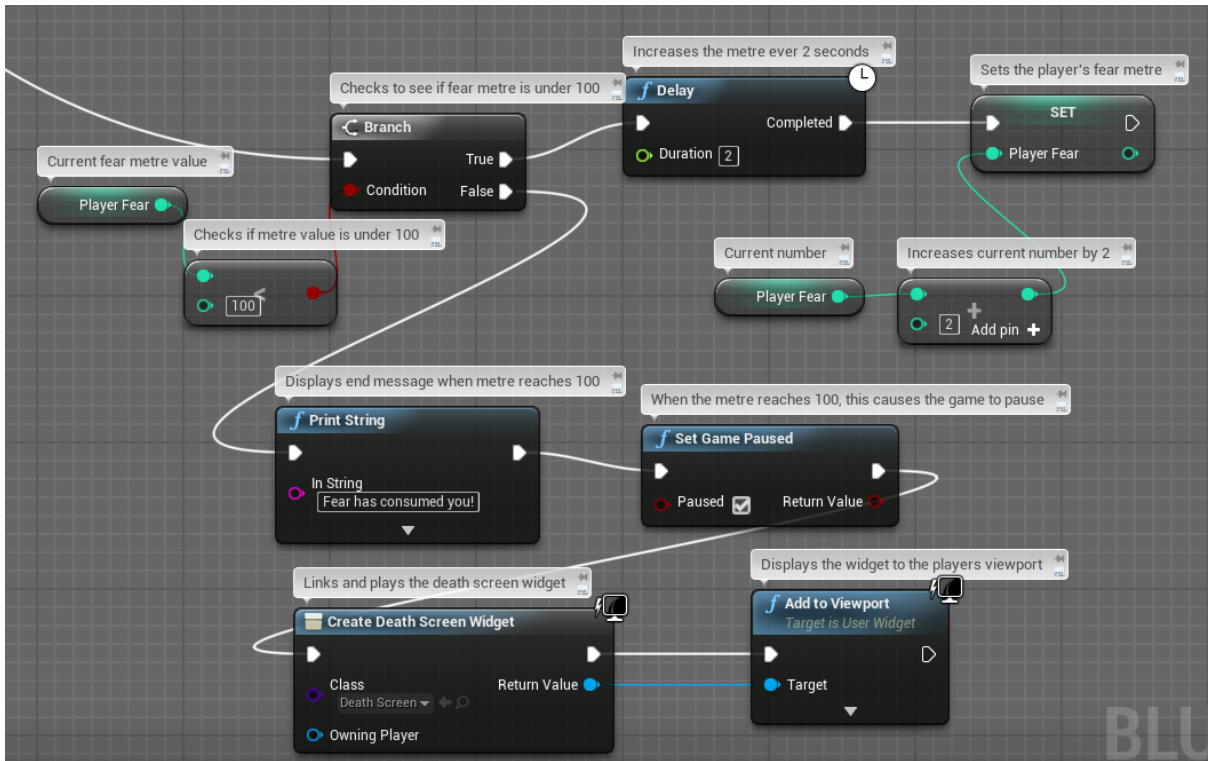
Final HUD example: This is the final HUD design that I made for the HUD widget, with the inclusion of the stamina metre and some repositioning of images from the original sketch idea.

Screenshots of blueprint code

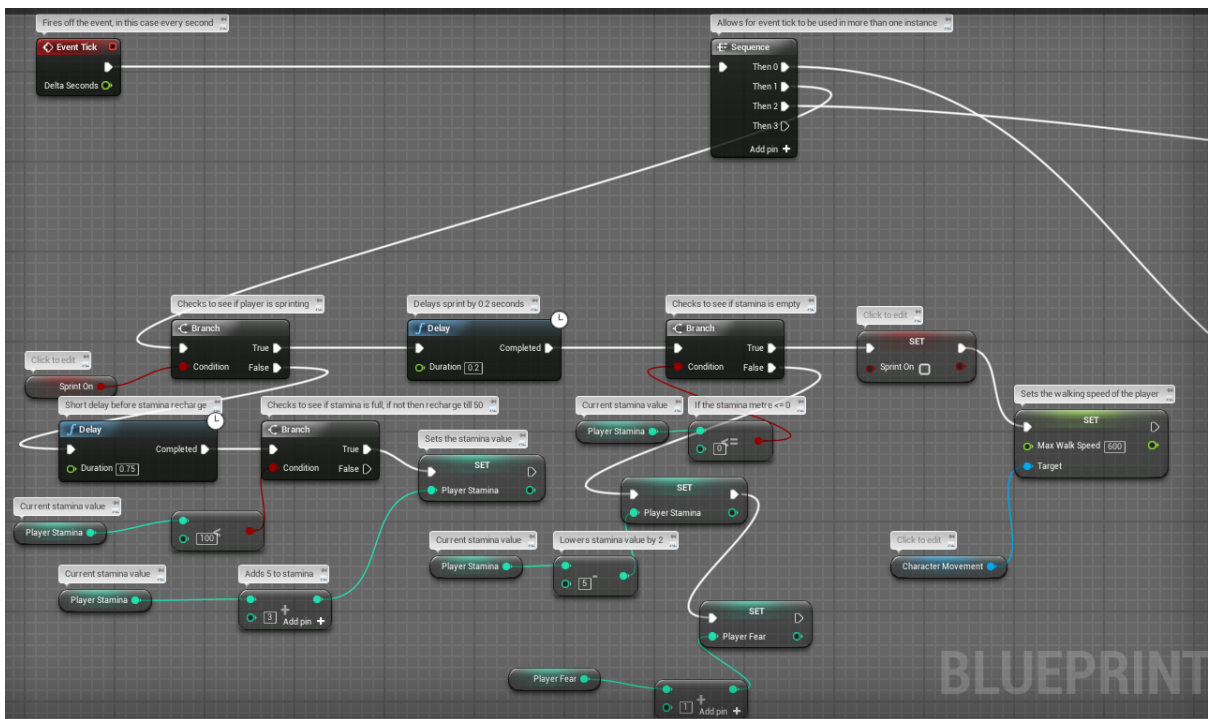
ThirdPersonCharacter



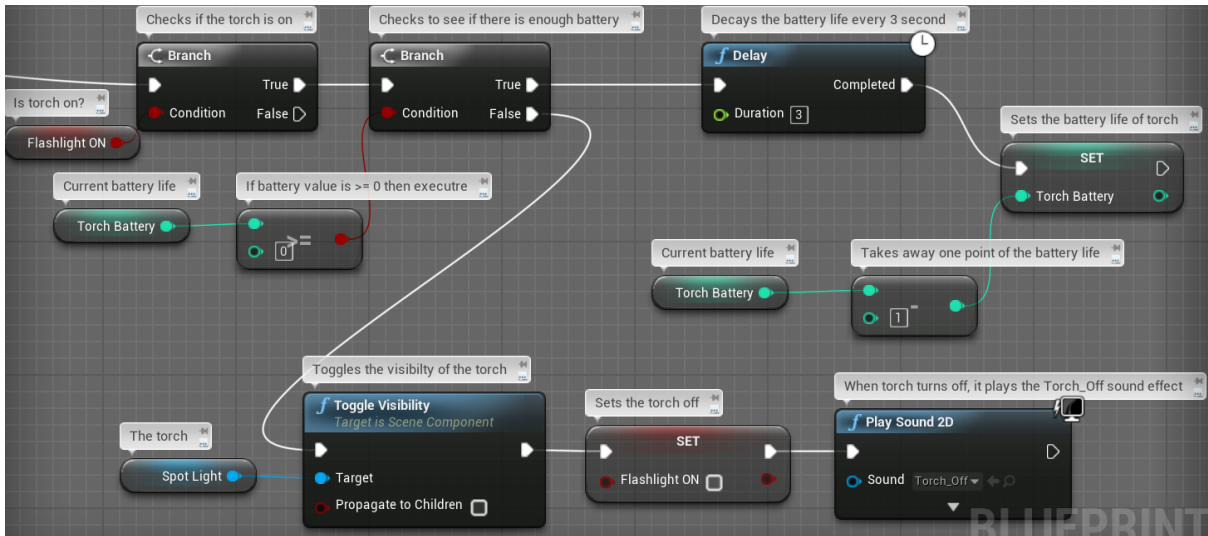
Player HUD: This allows for the players HUD widget to be displayed at the beginning of the level.



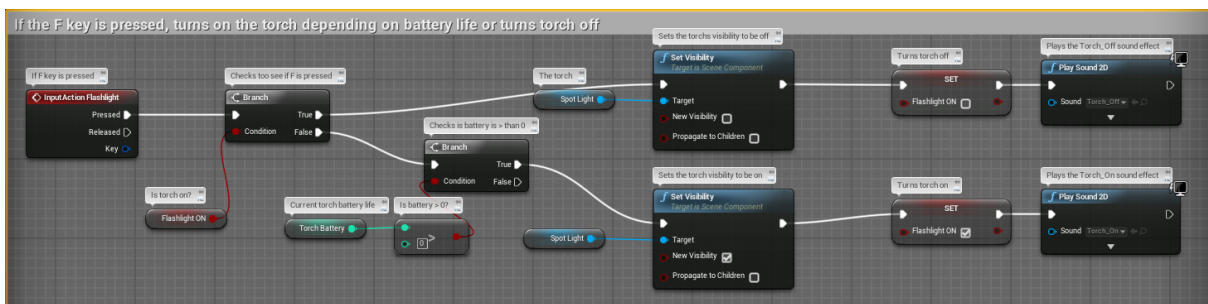
Fear metre bar: This code allows for the fear meter to increase by 2 every 2 seconds and if it reaches 100, it will pause the level and then displays the death screen using similar code to the HUD. The branch at the start is linked to a sequence which is linked to the Event Tick which causes the event to update every frame.



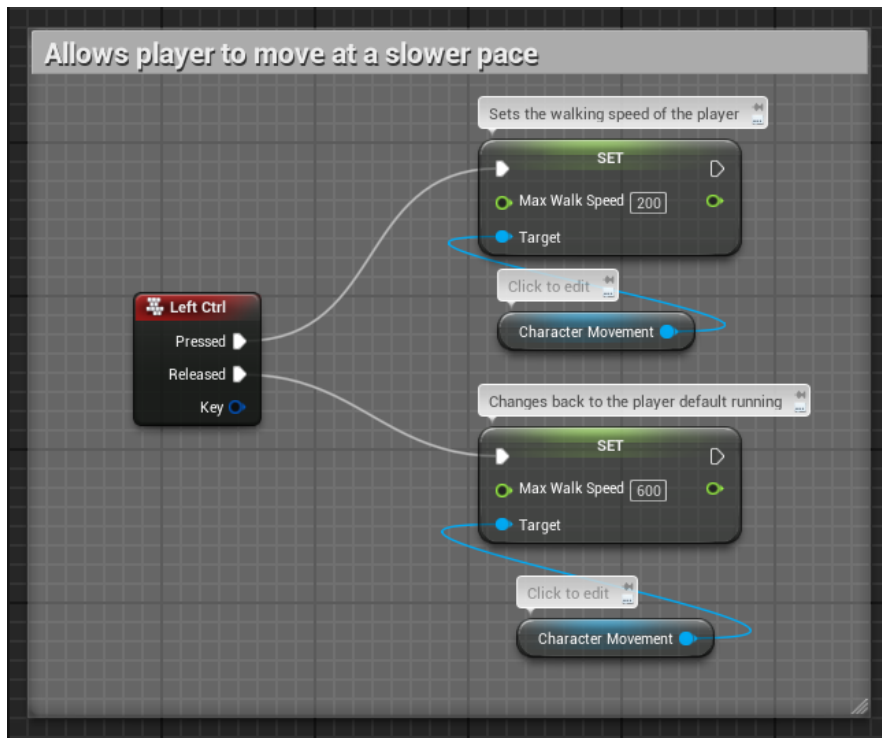
Player Stamina bar: This code is linked to the same sequence which is also linked to the event tick. If the player isn't sprinting then it checks to see if the stamina bar is less than 100 and recharges by 3 every second. If the player is sprinting then there is a delay of 0.2 seconds before the stamina lowers by 5 and increases the fear metre by 1 every second. If the metre is less than or equal to 0 then the player returns to their normal running speed.



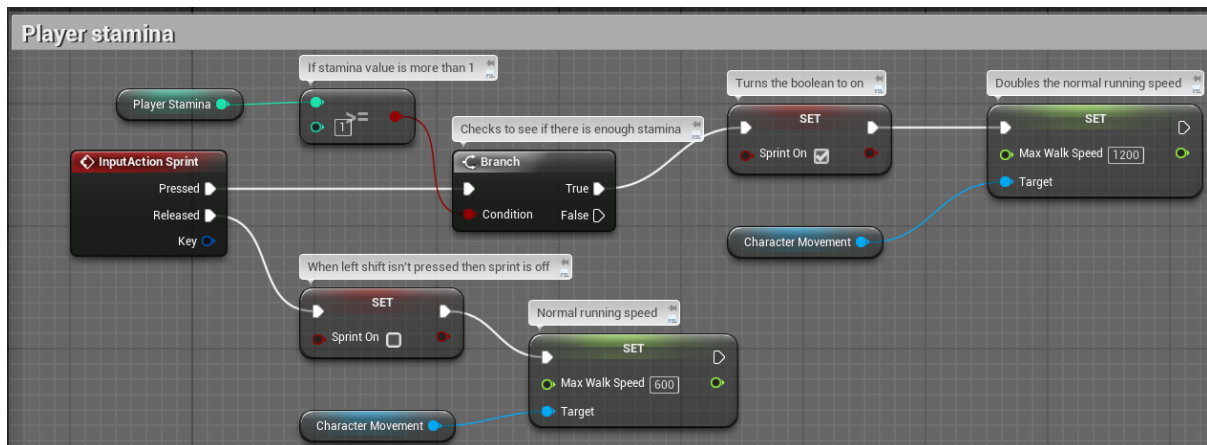
Torch Battery metre: This is the final piece of code connect to the sequence linked the event tick. This allows too see if the battery life has been reduced to 0 or not. If the player tries to turn on the flash light without battery it won't turn on. However if they do have battery and the battery is on, it will lower by 1 every second and once it hits 0 it will turn off.



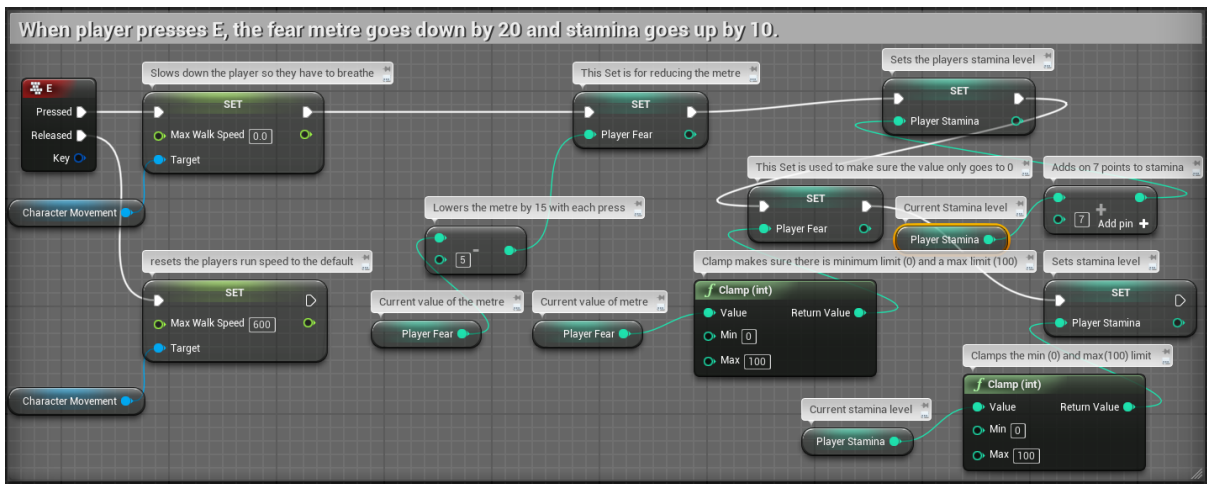
Flashlight activation: This allows for the flashlight to be activated. When the F key is pressed, depending if the torch is on then it will toggle the visibility of the spotlight (the torch in game) off. If the torch is already off then the spotlight's visibility will be turned on, as long as there is enough battery life.



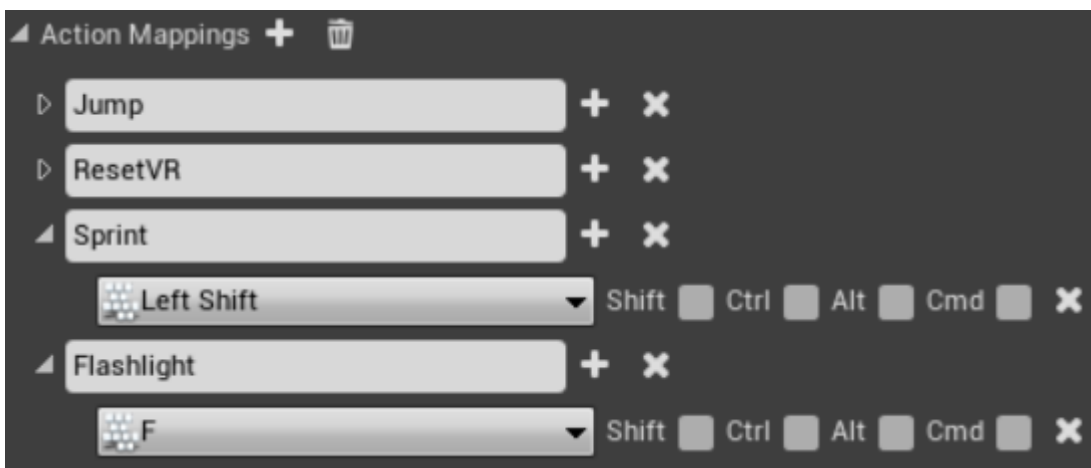
Player walk: When the player presses and holds down the left ctrl key, the player's movement speed is reduced down to a slower pace to allow them to carefully move around the level rather than running around. When the key is released the movement speed is set back to its default value.



Player sprint control: When the sprint key, the left shift, is held down then the branch checks if there is enough stamina. If there isn't enough then it the player won't be able to sprint but if there is enough, the movement speed is doubled.

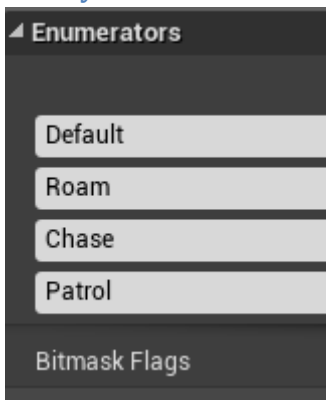


Lowering the fear metre: The player is able to press the E key to lower the fear metre so they can survive in the level. When the key is pressed, the player stops moving and the fear metre is lowered by 5 and the stamina bar is increased by 7. When the key is released the player movement is set back to its default.

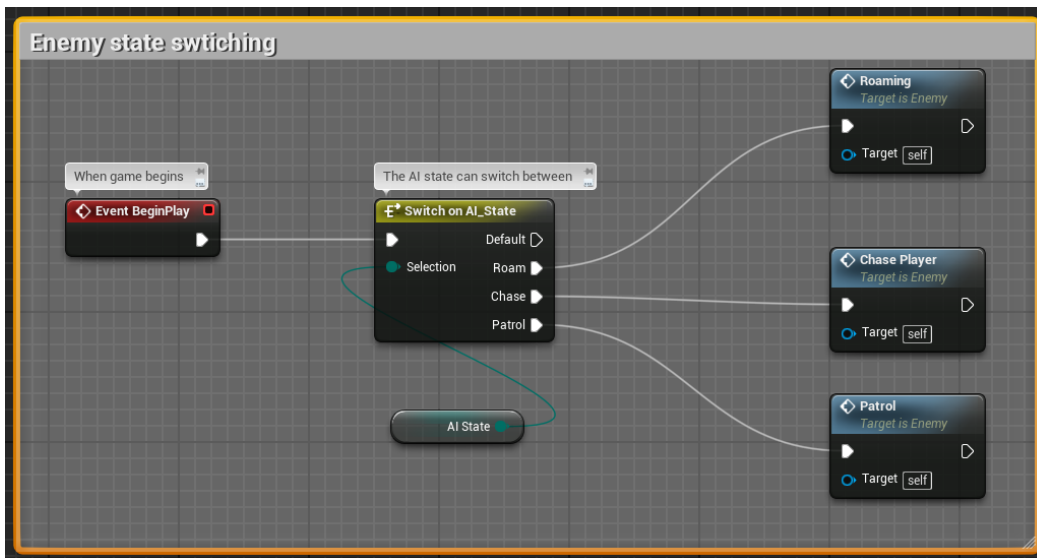


Evidence of the Action mapping: This is just to show that the ActionInputs are associated with the keys mentioned in the sprint code and the torch controls mode.

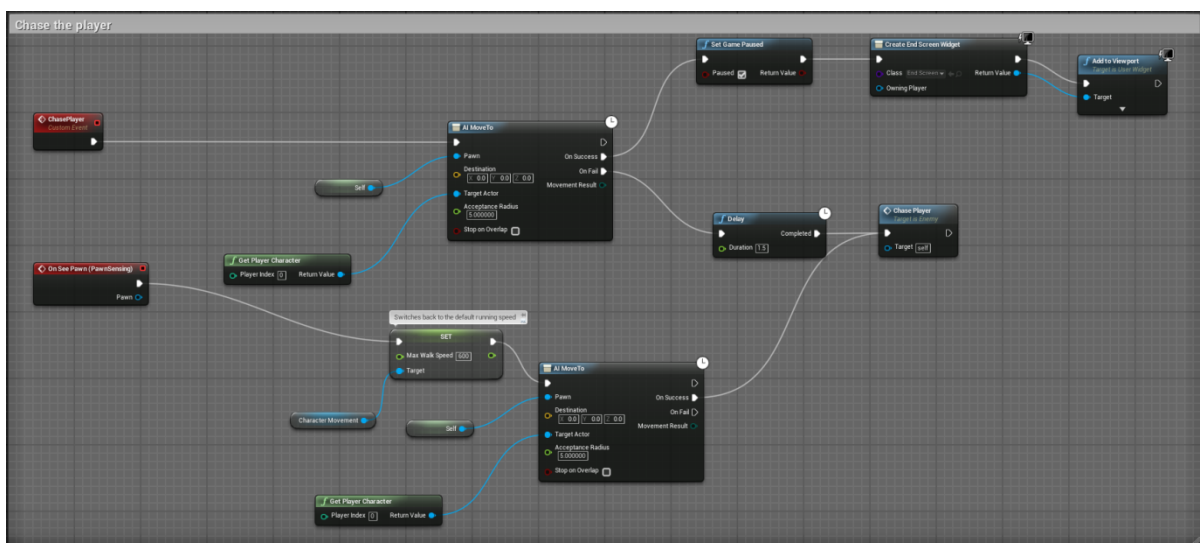
Enemy



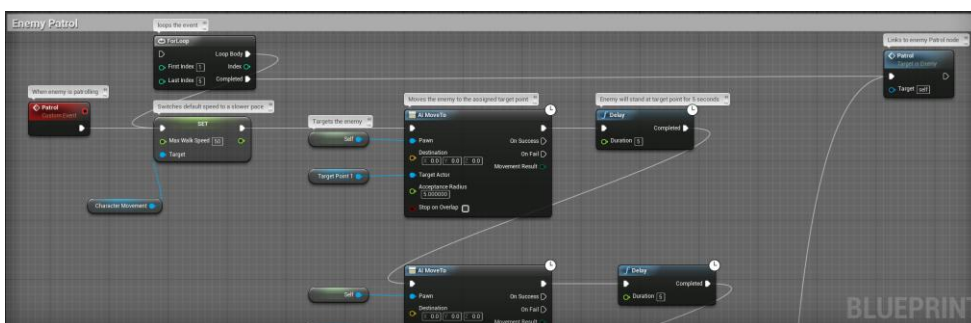
Enumerator: The enumerator allows for different states that the AI can be in during the run time of the level.



AI_State change: The switch node is activated every frame too allow for the seamless changing during the game. The AI_State is the numerator and AI State variable is linking it to the switch node. The Roam, Chase and Patrol are all attached to nodes which are all part of larger pieces of code which are expanded after this.



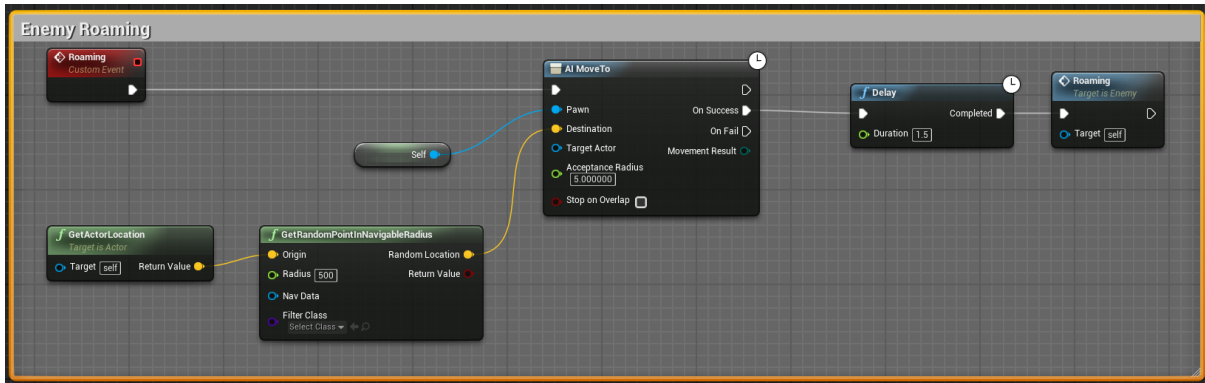
Player Chase: This allows for the AI to chase the player. The ChasePlayer is a custom event that when they are successful on catching the player then the game is paused and the death screen is played however if they haven't caught the player then there is a slight delay before they can chase them again, this is then linked to the Chase Player node at the end. The On See Pawn event works when the player enters the AI's line of sight the AI's movement speed is increase a walking pace to a run and chases after the player by using the target actor. This is then linked to the Chase Player node which will be used in the 'Switch to AI_State'.



AI State	Patrol
Target Point 1	TargetPoint1
Target Point 2	TargetPoint2
Target Point 3	TargetPoint3
Target Point 4	TargetPoint4
Target Point 5	TargetPoint5
Target Point 6	TargetPoint6
Target Point 7	TargetPoint7
Target Point 8	TargetPoint8
Target Point 9	TargetPoint9
Target Point 10	TargetPoint10
Target Point 11	TargetPoint11
Target Point 12	TargetPoint12
Target Point 13	TargetPoint13
Target Point 14	TargetPoint14

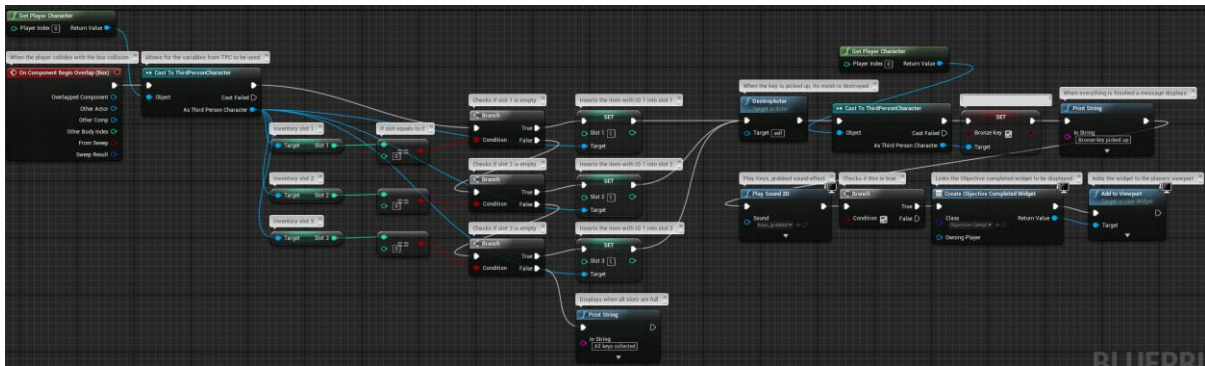
Patrol: The patrol event works having the AI walk at a walking pace and following target points when it isn't chasing the player. The AI will walk towards a target node in the amount of time given in the delay and once the delay is finished,

the AI will then move onto the next target point until it reaches the last point it then loops back to the first one. The final target point in the code is linked to the blue Patrol node. In this image you can only see 2 target points but in the level there are 14 target points in the level, the only reason for this is because the image would be too large to fit all 14 target point code into. The image on the far right shows the default AI State and all the target points and their associated target points in the level.

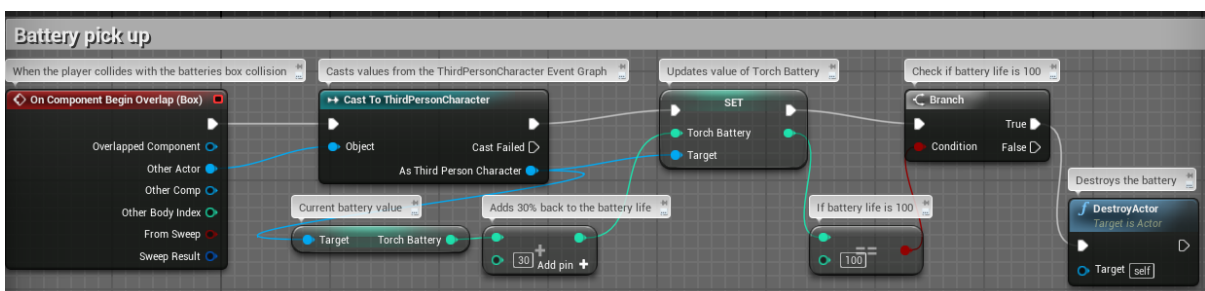


Roaming: The roaming event allows for the AI to roam the level when the player is lost.

Pickup blueprints

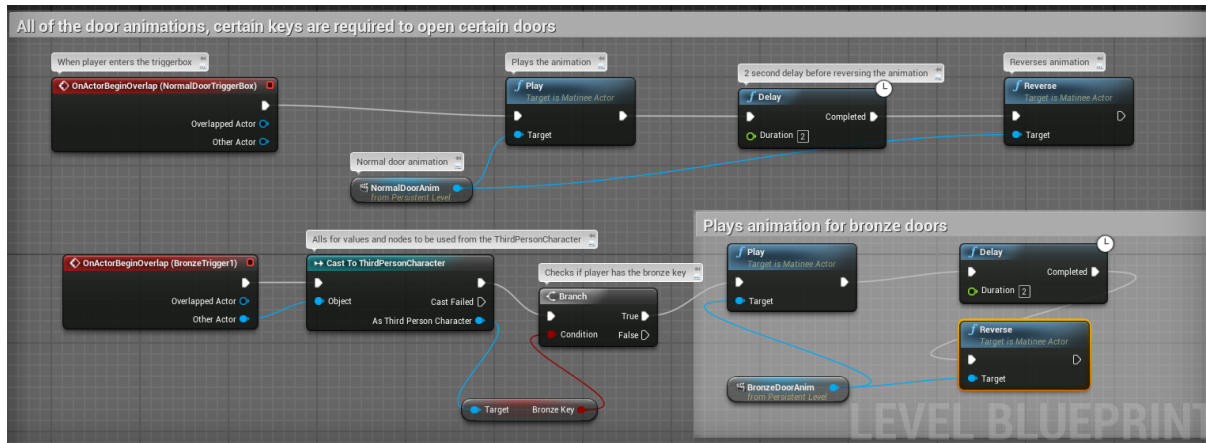


Specific Key pickups: This code here is similar for all 3 keys (bronze, silver and gold) but I'll be using the BronzeKey_PickUp blueprint for the example. Each of the keys has an image and ID which is associated with it; in this case the bronze key's ID is 1 as it is the first key to pick up. When the player hits the key's trigger box, the code checks to see if an inventory slot is empty, if not then it checks the next slot and so on until it displays a message saying inventory full. If a slot is open then the slot equals 1 (The Bronze Key's ID) and then destroys the actor. It then updates the Set Boolean (Bronze Key), plays a sound effect and then displays the objective complete animation using its respected widget and 'Add to Viewport' node. The key ID's will be explained in the HUD widget.

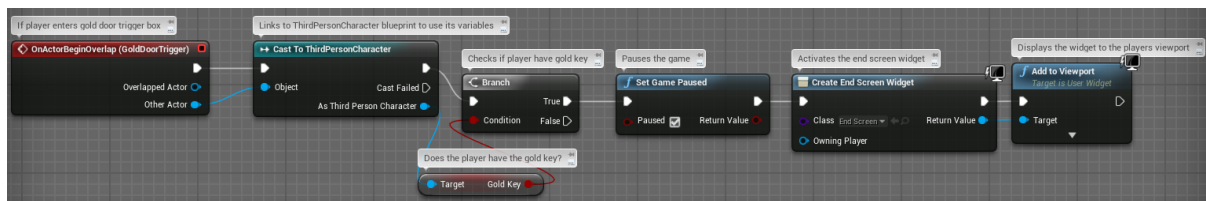


Battery pick up: When the player enters a battery's trigger box, it adds 30 onto the battery life's current value, if the battery life is 100 then the battery's actor is destroyed.

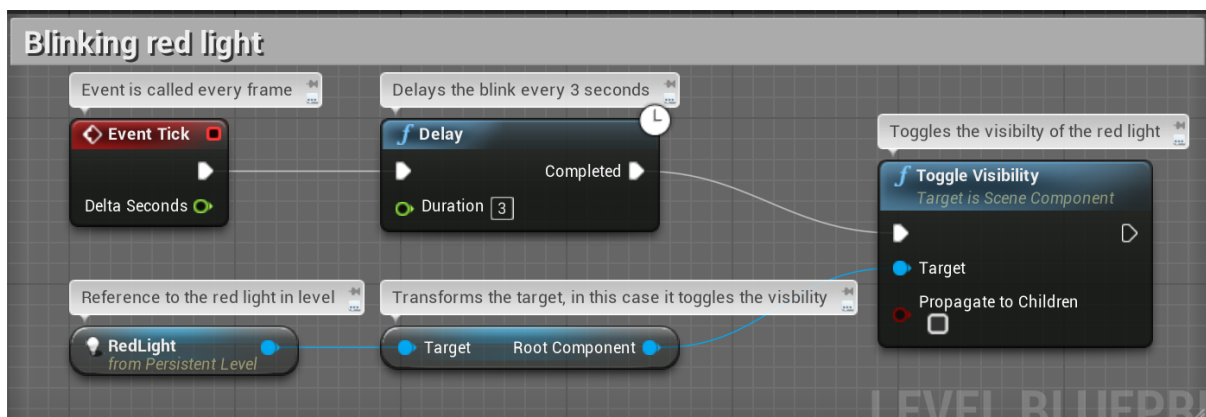
Level Blueprint



Normal, bronze and silver Door activation and animations: What is scene in the image isn't all of the door and key code. For doors requiring the bronze and silver key, the code is similar with the only difference being the Get [specified] key linked to the branch and ThirdPersonCharacter cast. For doors that don't require keys, when the player enters the trigger box the animation plays to open the door and after a delay it is reversed to close the door however when they try to walk up to a key required door, the door won't open. For key specific doors, when the player enters the trigger box the ThirdPersonCharacter variable will check if the [specified] key Boolean is true and if so it'll play the door animation for that door.

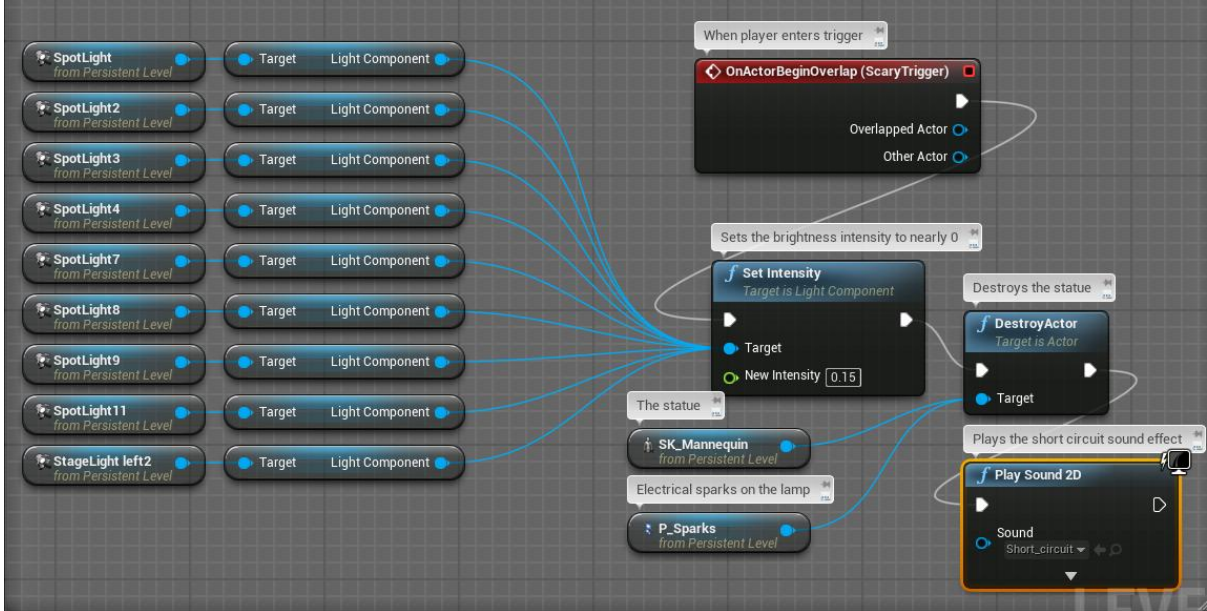


Gold door activation: When the player activates the gold door, the start of the code is similar to the door code above but instead of playing an animation of the door opening, the game is paused and the end screen widget animation is played and added to the player's viewport.

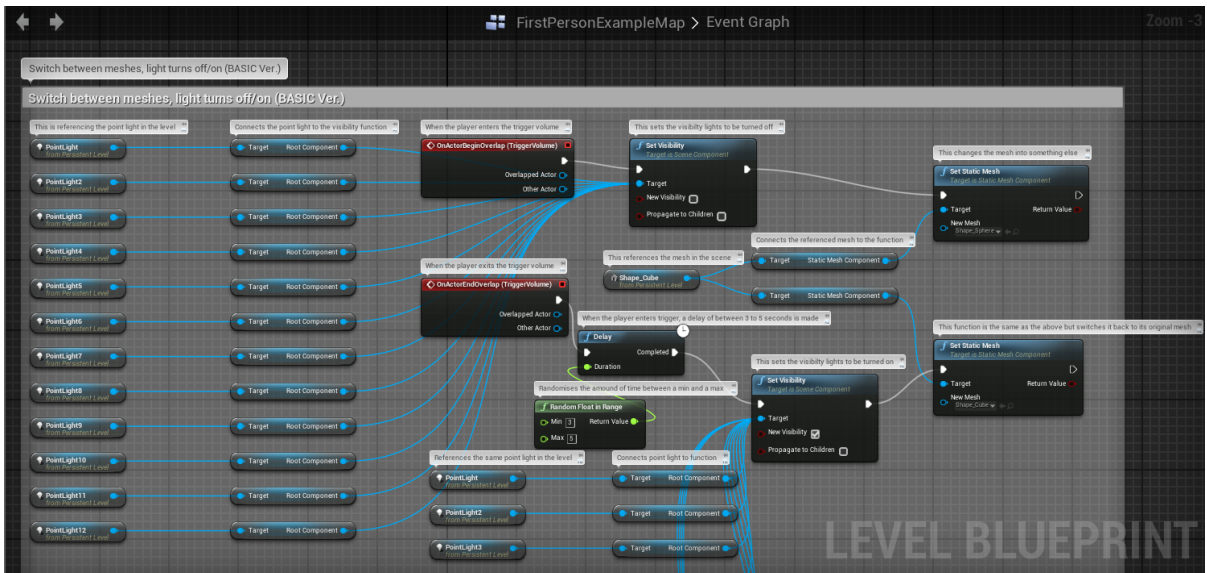


Blinking Red light: When the level begins, the visibility of the red light is turned on and off with an interval of 3 seconds in-between.

Turns down light intensity of the lights and makes statue disappear

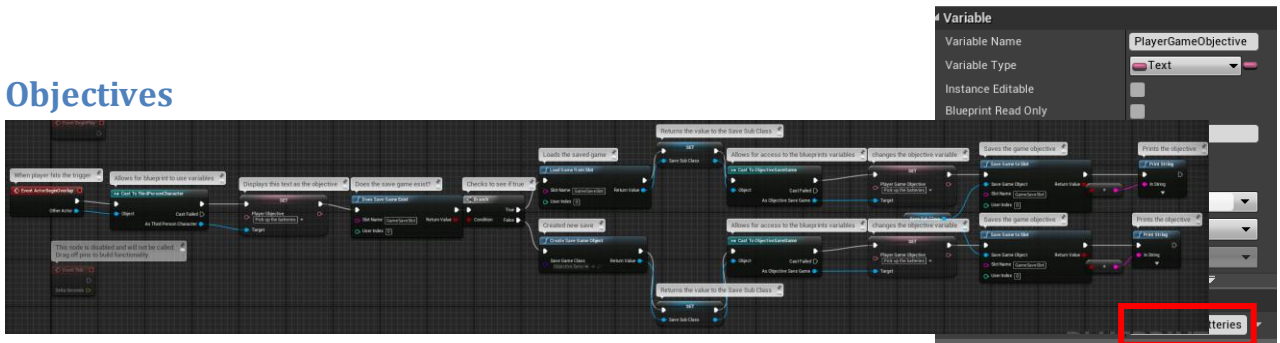


Spotlight, statue and effects: When the player leaves the office at the start of the game, they'll see some small simple lights and a lamp, with sparks shooting out of it, shining a light at a statue. Once the player enters the trigger box outside of the door, the spotlights intensity will turn down incredibly low and the mannequin will disappear and the sparks as well leaving only smoke. A short circuit sound effect will also play when the player enters the trigger box.



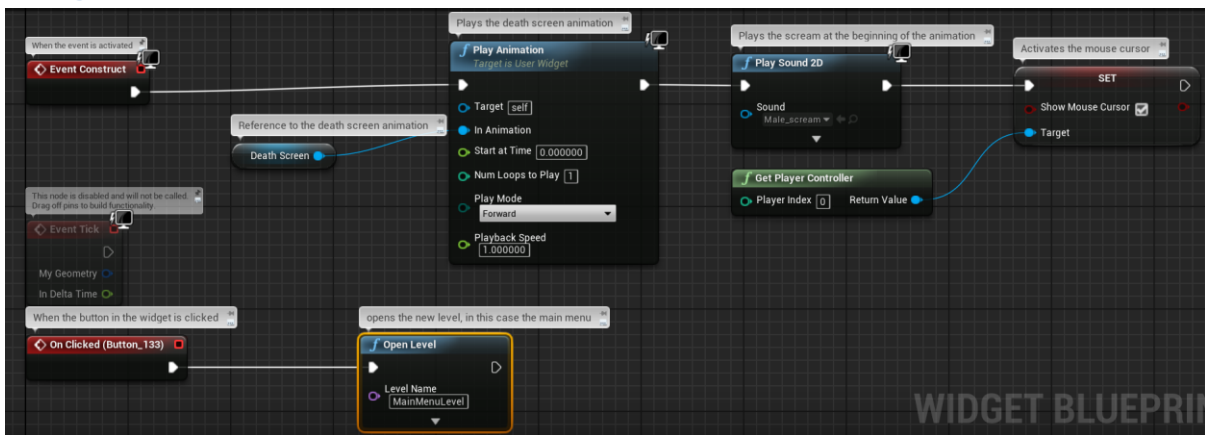
First example trigger activated mesh swap and light deactivation: All though this code isn't in the final level, this is the code I had created that when the player entered a trigger box the meshes linked with the trigger box would change their mesh and all lights linked with the trigger box would deactivate. When the player left the trigger box, there would be a random delay between 5 to 8 seconds and after that the mesh would change back to its original mesh and the lights would be reactive. I removed it from the final level because I couldn't get it the AI to disappear and re-appear where I needed it as it would only make copies of itself and not despawn.

Objectives

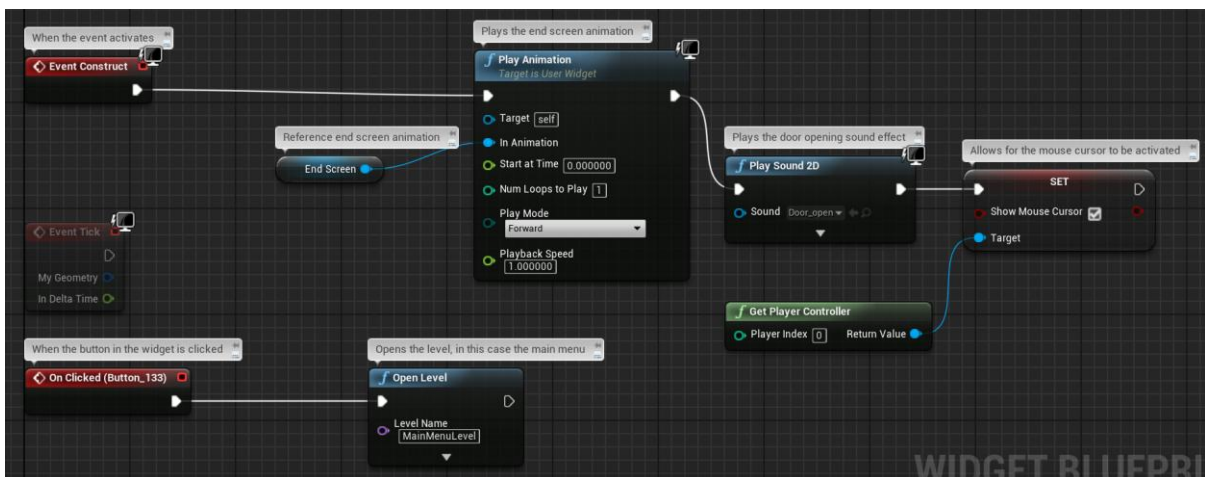


Objectives: All objective blueprints have the same code besides whatever is written within the Player Objective and Player Game Objective nodes. When the player entered the trigger box of an objective (which would be placed in an area or item of importance) event would cast to the third person character and then have the game update the text set into the Set Player Objective. It would then link to Does Save Game Exist (I wanted to include maybe a save feature but then thought about removing it as it only saved the objective. In the end I decided to just leave it in anyway so not to mess up the objective code that did work in my favour. A default objective is placed on the players spawn as to counter the save) and if true, it loads the saved game and sets the save sub class to carry over this information to the ObjectiveSavedGame cast which then updates the Player Game Objective (If level has been started fresh then the default objective will be as seen in the read until a new objective is picked up). If this isn't true then it will create a new save game and then update the Player Game Objective.

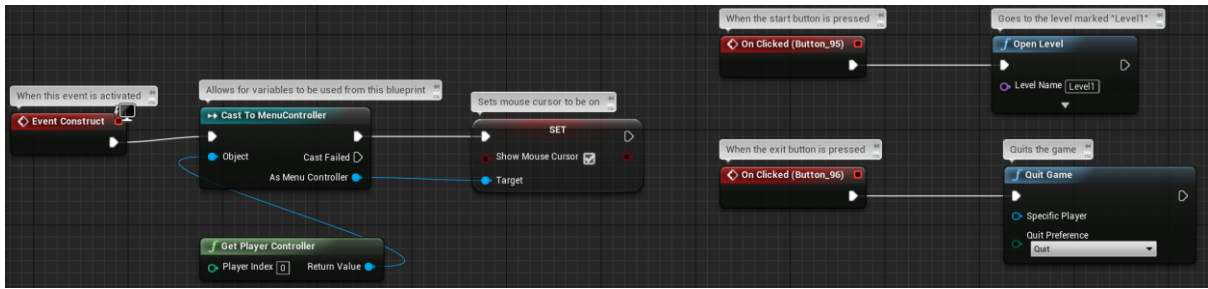
Widgets



Death screen: When the player dies, the animation within the death screen widget is played using Play Animation, a scream sound effect is played and the player is then able to move around the mouse cursor. When the Return to Menu button is clicked on, the main menu level is then displayed.

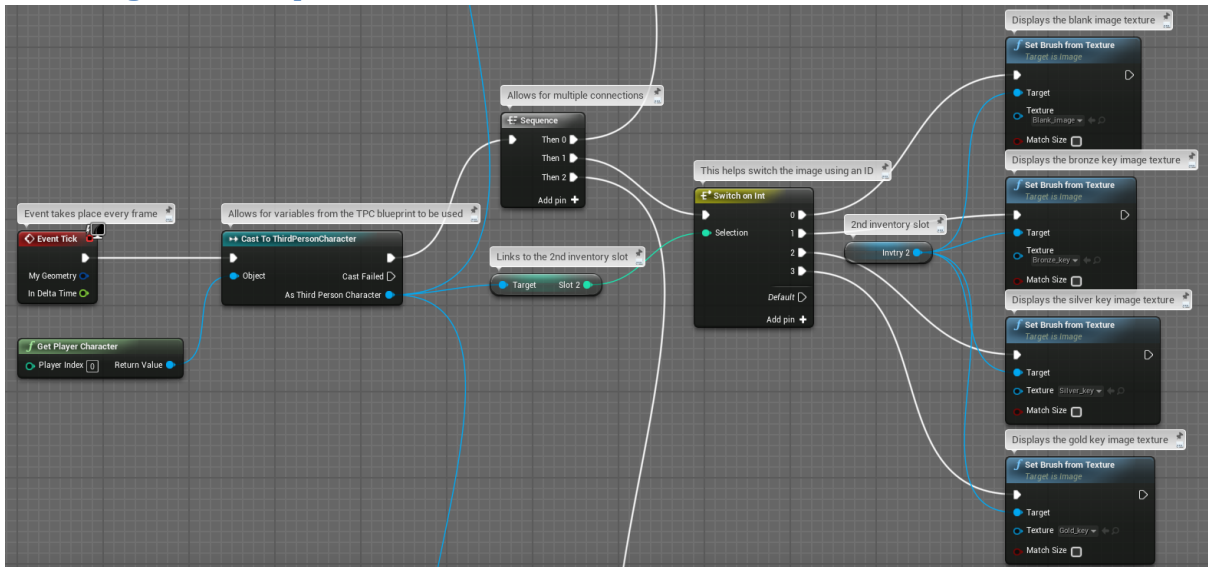


End screen: Very similar to the death screen in coding but the sound effect is changed to the door opening sound effect.

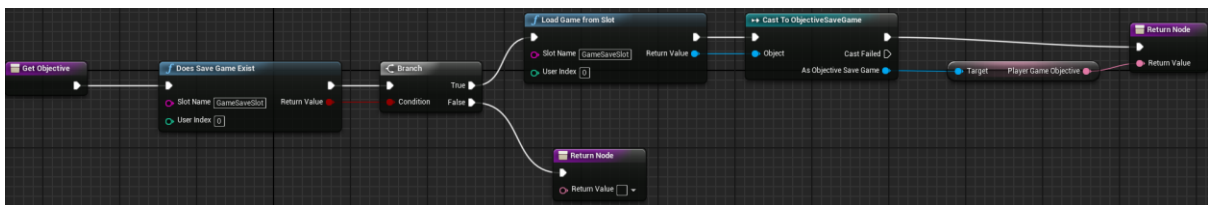


Main menu: The main menu code allows for the mouse cursor to be active. When the start button is pressed, the playable level is then opened. When the exit button is pressed, it quits the game and returns to the desktop.

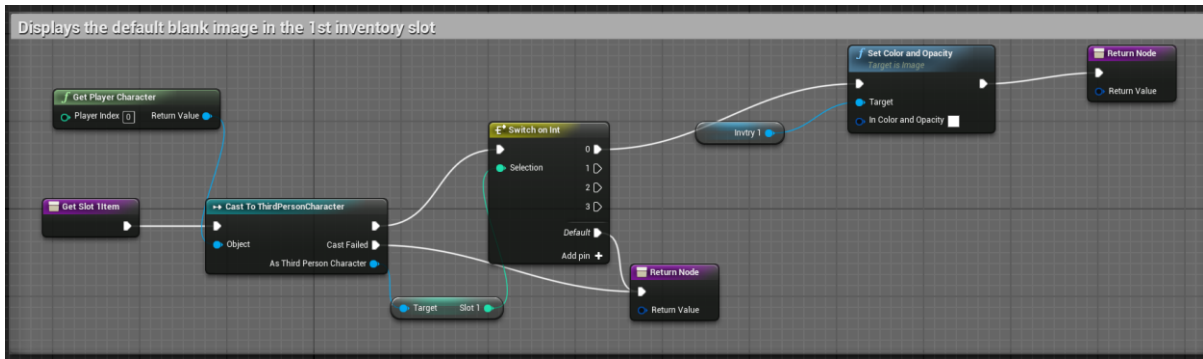
HUD Widget and blueprints



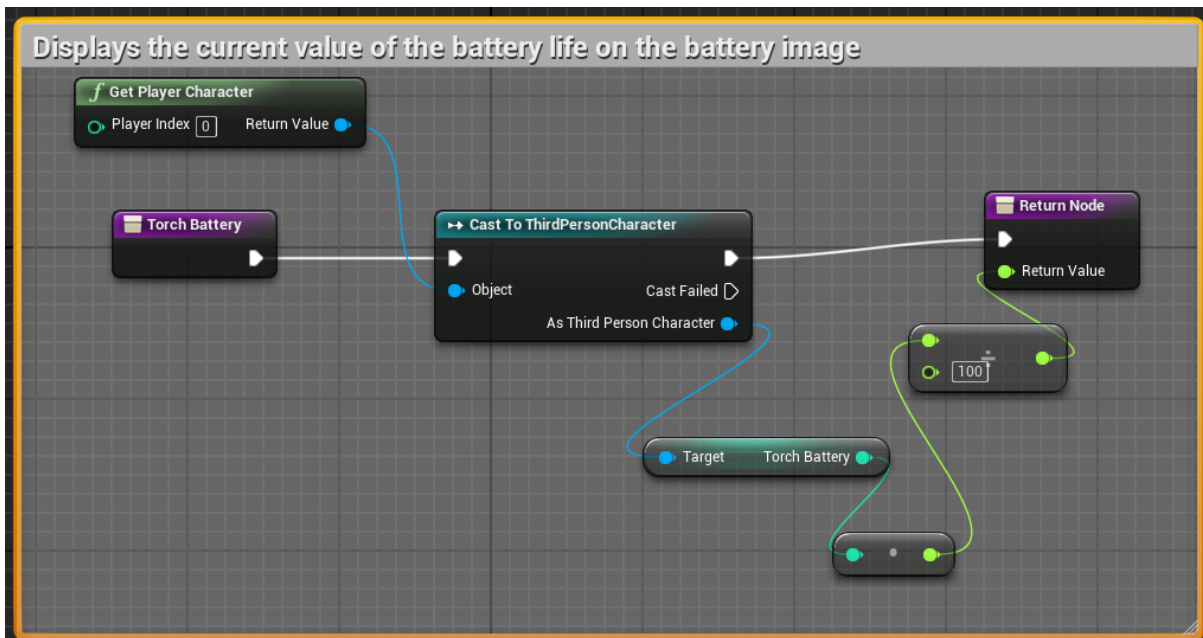
Displaying the inventory images: In this image is one of 3 Switch on Int nodes in this part of the HUD's event graph. The event graph updates the code every frame, the inventory slot number variable is taken from the ThirdPersonCharacter cast and linked to a Switch on Int node. The Switch on Int node links to Set Brush from texture nodes which allow for whatever image it displays to be linked to the ID number on the Switch to Int.



Get Objective: This allows for the objective to be updated and displayed. If there's a save game then it'll load the game from the save slot and then display the objective using the text box in the HUD.



Displays default image in the inventory: This code is the same in all 3 inventory slots with the only difference being the inventory slot variable number and the inventory image number. This makes the inventory slot default as the blank image when the game begins.



Display battery life metre: This code is similar to the fear metre and the stamina metre as all three at progress bars. This allows for the battery metre to update and show the correct progression. In this case, battery life value has to be divided by 100 because the progress bar only goes from 0 to 1. E.g. If the battery value is 73% and to display that onto the progression bar it'll be converted to 0.73.

Audio Cue

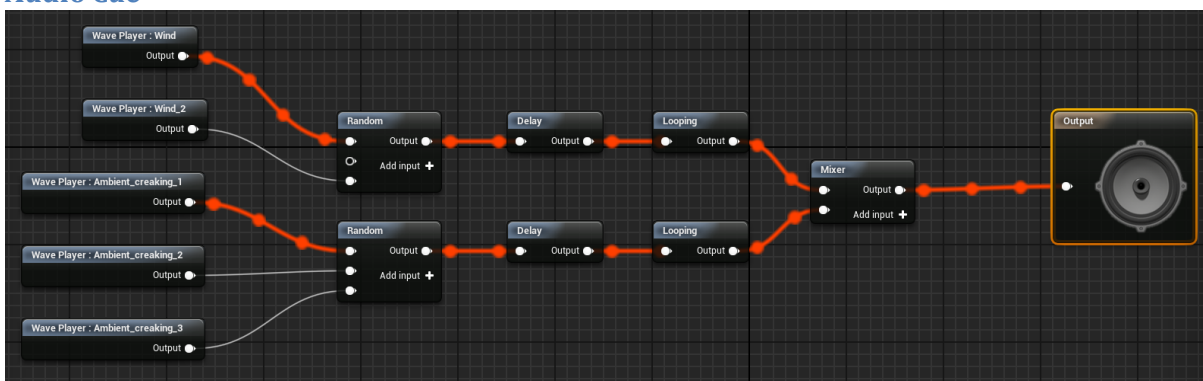


Figure 1 Audio Cue: This allows for me to loop and randomise ambient noises such as wind and creaking throughout the level.

Evaluation

For this assessment I was to create a functioning level utilising Unreal Engine 4, demonstrating the use of triggers, animated objects with the scene, a basic functioning UI system with collectable/destroyable/moveable objects and implement my own or potentially sourced art and audio. If anything within the level was sourced from outside then I am to credit it and its source and make sure that it complies with the copyright law. Within the blueprints I was to show the use of variables and arrays, events, functions, flow control and loops. I was also to create a video, with a limit of no more than 10 minutes, showing gameplay within the level and displaying what I had created. I was also to plan, develop, implement, test and evaluate of the level.

I believe that I managed to get just about everything in the requirements implemented into my level. I wasn't sure whether or not that the "/" next to the listed requirements was an "And/or" like situation so I tried to implement as much as I could in the allotted time. There were triggers in the game in just about everything from just walking in the hallway to the collection of items. Events had to be in the code either way to allow for things such as the HUD to even work as well as for the player and custom events were made for the AIs different states. There is a loop for the implemented into the AIs patrol state as well as in the Audio cue for the ambient sounds. Functions were made within the HUD to allow for the progress bars to work properly and for the objective to update when needed. Manitees were created for each individual door. I even created my own assets in the form of the batteries and keys as well as their textures and some of the audio, I made the wind by blowing into my microphone, the AIs rasping is a mix of 4 audio clips of my breathing/rasping and the key collection sound effect was me catching my accommodation keys. There are variables are used throughout the coding as well. I truly believe I've managed to implement just about everything that is required into this level.

There are some strengths and weaknesses that I believe I could mention in this as well. One of the big strengths I believe is AI itself since I have never created AI before in UE4, the only time I ever created AI was in my college course on Unity and even then it wasn't as good as the one I've for this level so I'm rather proud of it. Another strength I believe was when I was building the level itself and how I got creative with making the different pieces of art within the level using the Starter Content props and shapes plus all the default materials in UE4. It made me realise how much can be done for creating a level even with the starter content. One last strength I'd have to mention is how confident and how much I've managed to learn from creating the level, from reading online forms, re-reading all of the tutorials from the level design class to using the tutorials online I feel much more confident when it comes to programming in blueprints and believe I could make even better things in the future with the knowledge I've managed to gathered. However, I believe a weakness of my level is the fact that I did use only the Starter Content pack as allow though I made objects to decorate the map, the walls are rather bare and plain looking as well as when it comes to the hallways. Another weakness I can mention is that my level design of the building isn't the best in the work because this is the first time I've created a level in 3D. As a student coming from college when the last year we barely learned anything I could say that I've improved a lot but I know I could have done maybe a bit better on the designs.

For future development I believe I should concentrate on bug testing and playtesting much more. Although I got lucky with some of the code, it would still be much better to take more notes in case I encounter a bug that I could have noted down the fix for future reference. In the end I am rather proud of this level and all that I managed to both learn and create during these past few weeks. With this experience I'm confident not only in just myself when making content in UE4 but if I work with a larger group of people in the future producing a larger game.

Resources and Sources

Music: 003_Spelunking_PT1 - Created by Craig Brown (myself) and Enzo Satera (2017)

Various creaking noises – Sourced from freesound.org.

Creators include;

- Herbert Boland
- Maximus Dragonus
- Ben Boncan
- Joelaudio
- Alienxxx
- Jarred Gibb
- Loafdv
- Miekjy
- Apollo Aiello
- Petewyer2

Rasping and wind sound effects – Craig Brown, creating using headset and Audacity

3D Battery mesh and texture – Craig Brown, created using Maya 2018 & Photoshop 2015 CC

3D Key mesh and bronze and silver material – Craig Brown, created using Maya 2018 and the UE4 material creator

All of the HUD assets – Craig Brown, created using Photoshop CC 2015

Main menu mansion image: <http://architectureimg.com/haunted-house-darkened-shady-mansion-dark-scary-creepy-ominous-shadowy-black-gloomy-sunless-image-gallery/haunted-house-darkened-shady-mansion-dark-scary-creepy-ominous-shadowy-black-gloomy-sunless-picture/>

Creator of image: Helaine Chardon

Various other 3D meshes, materials and textures – Unreal Engine 4.17 Starter Content Pack

Research sources:

<https://www.allaboutcounseling.com/library/nyctophobia/>

<https://www.healthtopia.net/disease/mental-health/phobia/nyctophobia-fear-darkness>

<https://www.healthline.com/health/nyctophobia#outlook>