Project Isolation

RCGD6002 Final Major Project – Games Design Document University for The Creative Arts (BA) Games Design

A Games Design Document created by Michael Enriquez

Document	Contents
----------	----------

<u>3</u> PROJECT ISOLATION - DESIGN DOCUMENT	1
3.1 DESIGN VERSION	1
3.2 DESIGN GUIDELINES	1
3.3 GAME DESIGN DEFINITIONS	1
3.4 GAME FLOW CHART	3
3.5 PLAYER ELEMENTS	4
3.5.1 PLAYER	4
3.5.2 PLAYER PROPERTIES	5
3.5.3 PLAYER REWARDS (PICKUPS & HEALTH PICKUPS)	11
3.5.4 USER INTERFACE (UI)	12
3.5.5 HEADS UP DISPLAY (HUD)	13
3.5.6 PLAYER VIEW	17
3.6 SLENDERMAN	18
3.6.1 SLENDERMAN'S DEFINITIONS	19
3.6.2 SLENDERMAN'S PROPERTIES	19
3.6.3 ARTIFICIAL INTELLIGENCE (AI)	20
3.7 PREVIOUS SUCCESSFUL HORROR GAME TITLES	20
3.8 PROJECT ISOLATION'S STORY	21
3.9 RESEARCH	22
3.10 LEVEL DESIGN	26
3.10.1 LEVEL COPY	27
3.11 PRISON BREAK INTERVIEWS	30
3.12 GAME ARCHITECTURE	31
	37
3.12.1 HOW TO PLAY COPY	38
4 PROJECT ISOLATION'S TECHNICAL DOCUMENT	42
4.1 ANIMATION	42
4.2 PROJECT ISOLATION - AUDIO CONTENT	46
4.3 PROJECT ISOLATION - MUSIC	46
4.4 PROJECT ISOLATION'S C# SCRIPTS	47
4.5 3D MODELLING (MAYA 2021)	55
4.6 TEXTURING (SUBSTANCE PAINTER)	66

Game Design Document - Sign-off

CEO:	Michael Enriquez
Lead Designer:	Michael Enriquez
Lead Programmer:	Michael Enriquez

This Games Design Document's (GDD) sole purpose is to show the audience everything that is Project Isolation. Everything in Project Isolation, is inside of this document. The table above, can be clicked on and will go to the relevant sections within the games design document. Project Isolation was initially a two men job. However, circumstances changed very early on and then ended up being a one-man job. Project Isolation was created by Michael Emmanuel Enriquez for his BA Games Design – Final Major Project. Everything inside of his document was created and produced by him. Unless stated otherwise.

Project Isolation's Games Design Document was completed as of Thursday 20th May 2021.

Submission Deadline: Friday 21st May 2021

3 Project Isolation - Design Document

Welcome to the design section of Project Isolation. This section describes everything that has gone into making Project Isolation. This design document serves a purpose: to explain everything about this project thoroughly and in detail. This includes the following: Main menu, Tutorial Scene, Level 1 and Level 2 scenes. Each scene in this document will be laid out specifically to show how that scene was made, as an example this may include C# scripts which explain within the script what that script does but more importantly show you what each line of code does in relation to that scene.

In relation to my games development blog

(https://projectisolation12.blogspot.com/2021/04/rcgd6002michaelenriquez-project.html) everything that has been written in Blogger has been cross-referenced to this document in which there are further explanations of work and examples. The link above when clicked, you will see further information that relates to Project Isolation. You will see a 2-week period (or more) of work that has been produced at deadlines. The link above may go into further detail on deadlines that have been reached. However, you will find them in this document.

3.1 Design Version

In the pre-production stages, Project Isolation was going to be released on two devices. They were for MacOS and Windows 10. However, to keep things simple, I decided to keep to one device which was Windows 10 because this game was completed on PC. Every C# script inside the game has been hard wired and programmed to work with a standard PC running Windows 10.

3.2 Design Guidelines

When coming to the end of Project Isolation, I did run out of time and if I had time, I would have gone through models that I thought did not suit the style of the game. I would have gone back into Substance Painter and retextured them and texture them as high quality. There were a lot of things that I would have liked to of improved in Project Isolation, however, with the time constraint, I was restricted given that I was doing this project alone. However, did my best. For further information, please see the reflection document.

3.3 Game Design Definitions

Winning and Game Over

Project Isolation is a horror game based in a maximum-security prison. In short, you must find your way to the beast called Slenderman and you have a battle and while defending yourself, you have to try and kill him before he kills you. You go through Project Isolation finding clues as to where he may be but ultimately at the end, you try to kill him. If he ends up killing, you and taking your body for dinner a game over screen appears. Telling you, you have lost. However, this game over will not start you back at the start of the game. This game over screen will put you back to the point before you died so you do not have to go back to the beginning. This is in keeping with the original way the game over was designed from my previous games design document.

Once the player has completed Project Isolation and killed Slenderman, then he will be taken to the game win scene. The scene counts down from 15 seconds and once the counter has reached zero then the player will be taken to the final scene. This scene is the credits scene.

If the player ends up dying from Slenderman then the player will go to the game over scene. The game over scene has a counter which counts down from 15 seconds. Once the counter reaches zero, then the game restarts back to Level 1.

The Game Over Scene



The Game Win Scene:



Gun Properties

The way the gun script works is pretty simple: The script has different sections such as damage, range, fire rate etc. Each gun in Project Isolation will be different. However, for the rifle (below) this was the settings that I thought was correct (subjected to change). I will break down the following properties for the script:

- Damage
- Range
- Fps Cam
- Muzzle Flash
- Impact Effect
- Fire Rate
- Impact Force
- Max Ammo
- Reload Time

Damage

The damage section is how much damage the bullet will do on the guard once you shoot him. For instance, for this script, once the player fires a bullet at the guard then it will take 10 damage points away from the guard. This is due to change depending on the difficulty level the player chose on the main menu. If the player chose hardcore mode, then the gun script will change slightly. So, for arguments sake the damage will be divided by 2 on hardcore mode. So, every time the player fires a bullet at the guard then it will only take 5 damage points from the guard's health.

Range

The range is self-explanatory however, the range is how far the bullet goes once it has been fired. The lower the setting the shorter the bullet goes in the scene. The higher the setting, the further the bullet goes in the scene.

Fps Cam

The first-person camera is attached to this script because not only is the camera a parent of all the weapons that the player will hold, but also the main camera is where everything will be happening such as using weapons etc.

Muzzle Flash

Muzzle Flash is a particle system in which I created in Project Isolation for the guns. The muzzle flash is the effect between firing the gun and the bullet coming out. So typically, on guns when you fire them, smoke comes out. I have done a similar thing with the guns in Project Isolation.

Impact Effect

The impact effect is a bullet sprite. So, when you fire guns in the game you will see the bullet hit on guards, objects and around the scene. To make the bullet effect I made this in Photoshop. Then just import the PNG file in Unity Engine & converted the file to a Sprite 2D.

Fire Rate

The fire rate is the speed between the bullet firing and hitting an object. For the rifle this is set to 15. If the fire rate properties changed to a lower number, then the bullet will take its time firing. If you change the properties to a higher number, then the bullet comes out the gun quicker.

Impact Force

The impact force is the force of the bullet hitting an object. The higher the number, the higher the force will knockback the object / guard. The lower the number the less impact the bullet will do.

Max Ammo

The max ammo is how much ammo the gun will have when the player picks up the gun. The number for the rifle is set to 20. However, this can change depending on what difficulty mode the player chose on the main menu. In hardcore mode the number will change to a lower number so that it will be a challenge for the player.

Reload Time

The reload time is how long the weapon takes to reload. I plan to make a reload animation alongside an ammo pick up so when you walk or pick up the ammo, then the reload animation will play and the max ammo will change / update.



3.5 Player Elements

In Project Isolation, there are a few things you can do. Firstly, you are able to roam around the scenes at your own pace and explore what prison is like for prisoners. From there, the player can gather his/her own pistol and ammo or collect heath pickups. Which with all this will prepare him/her for the final showdown with Slenderman. The other interaction is with ammunition, which enables you to collect ammo once the gun has been equipped. Lastly, is the health pickup, which enables the player to gain health once he/she picks up the health pickup.

3.5.1 Player

In Project Isolation, the player can do a whole range of things. If we start with the main menu since this is the first interactable scene that the player can see. In the main menu, the player can choose different settings such as their preferred resolution screen size and as well as if they want the game to be in full screen mode or not. You can visit the 'Tutorial' page which allows you to gather more information such as the story of the game as well as how to navigate around the scenes. Now from the main menu you can play the game. As soon as you hit the play button, you will be transported onto the first level and you will end up in your prison cell. From there you are free to roam around the scene. You can a range of things such as collect ammo for weapons. Collect a flashlight and more importantly read prisoner's letters that tell you more about the prison. You also can collect health bar packs which increase your health if you end up having low health on your health bar.

The players one mission in Project Isolation, is to kill the beast which is called Slenderman. To kill him, you must shoot him multiple times with a gun. As you shoot him, his health bar will go down gradually and when his health reaches zero, then he dies, and you will be transported to the game win scene from there takes you to the credit's scene. However, if you are unable to kill him because you either have ran out of bullets or he ends up killing you by taking damage, then you will end up back at the start of the game and will have to try again.

3.5.2 Player Properties

Project Isolation is a single player game. Meaning you are only playing on your own, but more importantly, playing as First-Person. So, you will only see what the player sees from his eyes and won't see his body, arms and legs etc. In Project Isolation, the FPC (First Person Controller) has scripts attached to him. The player cannot see them but in Unity they are attached. These particular C# scripts do a whole range of different things. Below, you will find a screenshot of all the different players properties and what the different types of C# does:

r 😃 🗹 Character Controller			0	:
Slope Limit	45			
Step Offset				
	0.08			
	0.001			
	X O			
Height				
🖷 📕 🗹 Person Controller (Script)			0	
	PersonCont			
Player Camera	MainCamer	ra (Transform)		
Mouse Sensitivity	2			
	None (Guard)			
	None (Gun Sc	ript)		
Walk Speed	5			
Run Speed				
Max Health	100			
	None (Game (Object)		
Slenderman	None (Game (Object)		
Current Health	100			
Health Bar	🛯 Healthbar ((Health Bar)		
		•		
	~			
Min Damage				
	50			

The 'PersonController' C# script stores a lot of different codes and they all do different things. At the beginning of Project Isolation, my main priority was the way the first-person character walks. I ended up trialling a whole bunch of different movement scripts which I ended up storing inside of Project Isolation files and only recently removed them since they were no longer needed. The code that I went with was the one where you can move the character by using the 'WASD' keys to move around and the mouse to be able to look left and right. Also, in the 'PersonController' script is reference to the 'GunScript' which tells the player once the player has picked up the gun then you are able to shoot if you have bullets via the 'ammo count UI'. The 'Guard & Slenderman' variables are set in the 'void Awake' and set to find them components if there in the scene.

Players Health

The players health is also stored inside of the 'PersonController' script alongside another script which is stored on the players heath bar canvas called 'HealthBar'. This script sets the gradient colours and updated the slider. The health is stored inside the 'PersonController' script and determines what the players current health is.

Cell Door Open

The 'Cell Door Open' script which also attached to the player and what this enables the player to do is open the cell doors when you come inside a trigger. In Project Isolation, not every cell door that you can see, can be opened. So, there are only five different cell doors that the player can interact with. This script uses something called 'arrays' which enables me to use different doors at once and code each of them to do the same thing. Below you will see the full script and I have code commented the codes so it tells you exactly what it should be doing. There is also a cell door panel which is a UI canvas that tells the player what keyboard button he must use to interact with the cell doors.

	CellDoorsOpen	
Cell Door UI Panel	Cell Doors Panel	6
Cell Doors		5
Element 0	😚 CellDoor01	
Element 1	CellDoor02	
= Element 2	⑦ CellDoor04	
= Element 3	CellDoor06	
= Element 4	CellDoor07	
		+ -
Dooropen Sound	CellDoor Open Sound (Audio	Source) @

The cell door open script below shows exactly what it does as it has been code commented.



Desk Door

The desk door script is attached on the player and this enables the player to open the door for the desk. This also plays a sound, and a UI panel displays telling the player what keyboard button he/she needs to push in order to interact with the door.

🔹 Desk Door (Script)	0	ネ	
Script	DeskDoor		
Cell Door UI Panel	☺ Cell Doors Panel		\odot
Audio Source	Desk Door Opening Sound (Audio Source)		0
🔹 🗸 🗸 Dont Destroy (Script)	0	ᅷ	:
Script	DontDestroy		

The desk door script is roughly the same as the cell door open script as it more or less does the same. So, there is no need to produce a snippet of the code.

Don't Destroy On Load

The 'Don't destroy on load' is essential for Project Isolation. As this script tells Unity to not destroy particular items in the scene when you transition to another. As an example, when you go from Level 1 scene to Level 2 with the 'DontDestroy' script it won't destroys particular 'GameObjects'. The 'DontDestroy' script is on several items and UI canvases such as: The first-person character, the pistol, ammo UI, players health bar UI and Flashlight.

When it comes to the actual script, I had a lot of trouble with this as most of the time when the player moved between scenes, the scene would not load. After hours and hours of researching online and trialling different types of code, I finally managed to create this 'DontDestroy' script to say if the particular scene that the player is in IS NOT the game win or game over scene then do nothing but if it is then kill / destroy all the objects that this script is attached to.



The script above is self-explanatory. There is an 'int' which is a number and what that 'int' is doing is an index of the current scene in the game. Which is getting the build index of the scene. An if statement is saying that if the scene has a build index of either 7 or 8, then destroy the game objects. Since build index 7 & 8 is the game win and game over scenes. However, in the 'void Start' function, if the build index of the particular scenes is not index 7 or 8 then don't destroy.

<u>Flashlight</u>

The flashlight can be controlled by the player once he/she has picked it up. Once the flashlight has been picked it up it is accessible in the inventory by pushing either number 1 or 2 depending on what the player has picked up before the flashlight.

The flashlight has a Boolean that says if the flashlight has been turned on or off depending on an input which is the letter 'F'. There is a light source which is the light object so the player can see where he is going again this is toggled when the player pushes 'F'. We have a click sound which is the sound of the on and off. A 'fail safe' Boolean which allows the player to only be able to tun the flashlight on and off and not able to spam the button.

🔻 🗰 🖌 Flashlight (Script)		0 ‡ !
Script	Flashlight	
ls On		
Light Source	🗊 Flashlight Spotlight	⊙
Click Sound	Click Sound (Audio Source)	\odot
Has Flashlight		
Fail Safe		
Cell Door UI Panel	⑦ Cell Doors Panel	0

The flashlight script can be found below:



This whole script has been code commented to make my life easier when it came to adjusting a few settings for the flashlight.



Lights Off

The lights off script is exactly what it says on the tin. It turns off the lights. This script is attached to the player has is controlled for when the flashlight has been picked up. There is also light bulb smashes done with a 'wait for seconds'.

🔻 🛃 🛛 Lights Off (Script)		0 ‡ :
Script	LightsOff	
⊤ Lights		
 Element 0 	🕾 Hallway Lights 0	
= Element 1	C Hallway Lights1	
= Element 2	C Hallway Lights 1.1	
 Element 3 	G Hallway Lights2	
= Element 4	C Hallway Lights 2.1	
 Element 5 	C Hallway Lights3	
= Element 6	C Hallway Lights 3.1	
= Element 7	C Hallway Lights 3.2	
= Element 8	C Hallway Lights 3.3	
 Element 9 	C Hallway Lights 3.4	
= Element 10	C Hallway Lights 3.5	
= Element 11	⊕ Hallway Lights 3.6	
= Element 12	G Hallway Lights 3.7	
 Element 13 	C Hallway Lights 3.8	
= Element 14		
Element 15	G Hallway Lights 4.1	
= Element 16	Tellway Lights 4.2	
 Element 17 	⊕ Hallway Lights 4.3	
= Element 18	⊕ Hallway Lights 4.4	
= Element 19	Hallway Lights 4.5	
 Element 20 	C Hallway Lights 4.6	
= Element 21	⊕ Hallway Lights5	
= Element 22		
 Element 23 	G Hallway Lights7	
= Element 24	C Lamp Spotlight	
0		
⊤ Displays		
= Element 0	CCTV Display 1 (5)	
= Element 1	CCTV Display 1 (4)	
= Element 2	CCTV Display 1 (2)	
 Element 3 	CCTV Display 1	
Lightsmash	Bulb Smash (Audio Source)	
Lamp Sound	None (Audio Source)	
⊤ TV		
= Element 0	G Static TV Video 1	
= Element 1	Static TV Video 1 (1)	
Electric Particle	Electrical Sparks	

This script is self-explanatory as this gets arrays which are the different light objects in the scene, and I am basically telling the script to set them to off, so they are not visible in the scene and the scenes goes in total darkness. In between the lights going off is the 'yield return new WaitForSeconds' which waits an x number of seconds before continuing to turn off the other lights. This has been done so the lights go off in sections rather than altogether. Since this is an old prison and the electric has a slight few second's delay before turning off.

uł	lic void StartLightsOff()
	<pre>//Starts the Coroutine which is called turningLightsOff</pre>
	<pre>StartCoroutine(turningLightsOff());</pre>
23	
Fr	erence umerator turninglightsOff()
	//Sets the lights to be false - meaning the lights will turn off.
	//For this to work, you need to add the lights (GameObject in the Hierarcy) to this script
	Lights[0].SetActive(false):
	lights[1].SetActive(false):
	Lights[2].SetActive(false):
	Lights[3].SetActive(false):
	Lights [4].SetActive(false);
	//Plays the light smash audio
	Lightsmash.Play();
	//Yield return new WaitForSeconds(2) basically tells the script to wait 2 seconds before doing the next code.
	yield return new WaitForSeconds(2);
	Lights[5].SetActive(false):
	Lights[6].SetActive(false);
	Lights[7].SetActive(false);
	Lights[8].SetActive(false);
	Lights[9].SetActive(false);
	Lights[10].SetActive(false);
	Lights[11].SetActive(false);
	Lights[12].SetActive(false);
	Lights[13].SetActive(false);
	//Plays the light smash audio
	Lightsmash.Play();
	//Yield return new WaitForSeconds(2) basically tells the script to wait 2 seconds before doing the next code.
	yield return new WaitForSeconds(2);
	Lights[14].SetActive(false);
	Lights[15].SetActive(false);
	Lights[16].SetActive(false);
	Lights[17].SetActive(false);
	Lights[18].SetActive(false);
	Lights[19].SetActive(false);
	Lights[20].SetActive(false);
	Lights[21].SetActive(false);
	Lights[22].SetActive(false);
	Destroy(electricParticle);
	<pre>//Particle.SetActive(false); // Turns off the particle system</pre>
	//Plays the light smash audio
	Lightsmash.Piay();

Then once the script has finished, I have coded it to say destroy the lamp sound and also destroy the light smash sound. This script also turns off the TV's which are playing in the guard's room.

```
Lightsmash.Play();
yield return new WaitForSeconds(1f);
//Turns the displays to OFF
Displays[0].SetActive(false);
Displays[1].SetActive(false);
Displays[2].SetActive(false);
Displays[3].SetActive(false);
TV[0].SetActive(false);
TV[1].SetActive(false);
Destroy(Lightsmash);
Destroy(lampSound);
}
```

3.5.3 Player Rewards (Pickups & Health Pickups)

Health Pickup

Project Isolation is about survival. The player has a health bar which is in the top left-hand side of the screen and also has an ammo count. This ammo count only works when the gun is attached to the player. If we talk about the health bar first. If the player has zero health, then he dies. So, to give the players a chance to not die there are health bar pickups. These allow the player to gain health when his current health is not at 100%. The health bar pickup had to be created in Maya and then textured in Substance Painter and lastly imported into Unity Engine. However, for the health pickup this was created way before the final health pickup was modelled. Since I could make a basic cube and just add a box collider on the cube and click 'IsTrigger' meaning the trigger is set to allow. So, in the health pickup script I would code if the player went inside the trigger then set his current health to 'Random.Range' to set his health.

Problems: I ran into a few problems when it came to allowing other people to playtest Project Isolation. The feedback that I was receiving from the questionnaire documents that the health bar pickups were updating the players health but somehow was going over 100%.

Solution: Inside of the health pickup script I added line of code which was an 'IF' statement and it said that if the players current health was or reached 100 then the health pickup will remain 100. So, in theory this would work. As it didn't matter if your health was already at 100 because if you happen to walk into a health pickup then your health would not change. Inside of this script is a 'Debug.Log' which tells me what the health is when the player picks it up.

Ammo Pickup 'Loot box'

I wanted to do something a bit differently when it came to making a gun fully functional in Project Isolation. I did not just want the player to have unlimited ammo once the gun had been picked up and fire endlessly without stopping so, I had to create an ammo count UI system (which you will find further information about under **3.6.5 Heads up Display (HUD)**). The ammo pickups also known as a 'loot box' allow the player to collect ammo for the gun. However, since the ammo count UI was capped at 50 bullets, I had to make sure that inside the loot box script the bullets would be capped at 50. If I am being honest, this script took way too much time then it needed to be. As for some reason I tried to make the code complex rather than simple. Which in the end, simplicity was needed? If the script works which was my main mission.

Problem: I asked my friend to playtest the game first before I allowed other people to play. When he played Project Isolation, one of the main problems was that the loot boxes were not adding to the ammo count UI. The player would collect the gun and end up not being able to shoot because it would tell him that you have zero ammo.

Solution: I realized throughout playtesting Project Isolation that using a 'Debug.Log' was very useful. As not only does a 'Debug.Log' display in Unity's console but also checks if the lines of codes are working and for the ammo script this was the case. Some of the lines of codes were not working. After a few hours trying to figure out why this script was not working, I decided to start from fresh and make sure everything works before I uploaded the game online for play testers. The fixed script looked like this (below):

it ((other.gameObject.tag == "Player")
{	
	if(boolean.isPickedUp == true) //Checking to see if the gun has picked up
	Debug.Log("Player has picked up Ammo and Gun"); //Debug
	AmmoCount(); //AmmoCount
	ammoPickupSound.Play(); //ammoPickupSound
	Destroy(ammo); //Destroy ammo
	<pre>// hasPistol = true;</pre>
	<pre>//ammoPickupSound.Play(); //ammoPickupSound</pre>
	<pre>// Debug.Log("Working");</pre>
	<pre>//AmmoCount();</pre>
	The second s

3.5.4 User Interface (UI)

In Project Isolation, there are many different UI elements within the game. A UI element is everything that the player can see in the scene. As an example, a health bar is a UI because it is a canvas which is displayed on the screen or on an enemy that the player can physically see. Normally, these UI are not interactable, meaning the player cannot interact with them, but these UI elements may update depending on what the player does.

<u>Health bar UI</u>

The health bar is fixed to the screen. Meaning in the first level and the second level, players health can be monitored. The health bar is very important as this determines the players current health and progress through the scene. As soon as the health reaches zero, then the player dies. However, the health bar can increase with health pickups.



The health bar UI

The health bar needed to be very visible to the player so when it came to creating the UI for it, I made sure that this was in the top left-hand side of the screen and the scale was relatively big so that the health bar stood out.

Problems: When it came to playtesting Project Isolation, I soon realised that after the tutorial page that all of the UI's that were supposed to be on the screen were not and had disappeared.

Solution: After tampering around with some settings, I soon realised that the display resolution for the engine was not set at 1920x1080 instead the resolution size was set at the default setting.

Ammo UI

The ammo UI displays the ammo count to the player. The ammo UI is displayed on the bottom left-hand side of the screen and tells the player how much ammunition they have in relation to bullets when the gun has been equipped. To allow the ammo UI component to update in Unity, the player must first walk over to an ammo box. The ammo boxes are in certain locations within the first scene. To be able to collect the ammo and for the ammo to update the ammo count UI, the player must first equip the gun. Without equipping the gun, the ammo cannot be picked up. Which in turn, will not update the ammo count UI component.

This is the ammo count UI component:



Other UI Components

In Project Isolation, there are a few other UI components that display on the screen for the player. However, these UI elements **DO NOT** update any script unlike the health UI or the ammo count UI. What these other UI components do is display information in which the player can interact with. For instance, from the main menu page, the player is first put into his cell. This is where the player starts the game. When he walks towards his cell door, a notification will pop up on his screen. This will prompt him/she to push the letter 'E' on the keyboard. As soon as 'E' is pressed, the cell door will open, and the prompt will disappear. These UI's just give the player extra information on how to interact with certain doors. When the flashlight is picked up then all the lights explode / turn off. While this is happening, all the cell doors have been coded to open. So, when the player has collected the flashlight, the UI's for the cell doors are no longer there and the player cannot interact with them.

3.5.5 Heads Up Display (HUD)

In Project Isolation, there are quite a few scenes. It is estimated that there are roughly 7-9 scenes and only two of them scenes the player can interact with. The first scene that was created was the company logo scene. This is displayed at the beginning of the game. Unfortunately, I was unable to include it in the project settings under the 'graphics' settings as it was not letting me fade in and out the 'micky studios' image. So, to get around this, I made a scene and imported the image onto the canvas and created a C# script which allowed me to fade in and out the logo and with a 'yield return new WaitForSeconds' which allowed me to pause / add a delay into when the fading had finished.



Micky Studios logo



The Logo Script

This script gets the Image and the next level. This script also changes the way the image is displayed, by changing the canvas renderer and cross fade which is the fading in and out.

The Introduction

Now, I wanted to do something very different and new when it came to the introduction scene. This introduction scene plays right after the splash scene in Project Isolation. What this intro scene does is basically gives you an introduction into the game. However, I wanted this to be a video that is being played. I was able to use my skills in Adobe Premiere Pro and create a 30 - 45 second video that is a horror introduction into Project Isolation. Thankfully, there was a template online (logged in the 'free asset' document) in which I could use and change in the software. The music I used was from my work colleague's music production degree that he is studying, and he was willing to let me use it for Project Isolation. From there, I added a few other different sounds like the cell door slamming shut and a prison alarm. Which I felt was suitable for the introduction since this game is based in a prison.

The Introduction script is very basic and what this script does is play a video but, in the background, it is waiting 13 seconds 'yield return new WaitForSeconds(waitTime)' 'wait time' being 13 seconds. 13 seconds is how long the introduction video is and as soon as it has stopped playing, then the scene changes to the Accessibility scene.



Accessibility

I want everyone who plays Project Isolation to have fun playing the game. However, I totally understand that this game is not best suited for everybody. For instance, the target audience for this game is for anybody of the age of 12 onwards. Anyone younger than the age of 12 should not play the game since it contains horror scenes and low weapon violence. The accessibility scene was set up because in Project Isolation, there are scenes where the camera has this 'glitch' effect where you (the player) are looking through the game from a CCTV camera perspective. How this works is that the camera has a C# script which uses a 'camera glitch shader'. This shader changes the colour of the camera as well as the timing of the video. The video is a static vhs tape effect. This script is on roughly 3 - 4 different scenes and works very well with the credit's scene. However, because of the way the camera has this static 'broken feel' look and jumps around in particular scenes, the accessibility page had to be created because this may affect people who suffer from epilepsy. Since Project Isolation contains flashing scenes, it is only fitting and suitable to include an accessibility page as it gives people time to exit the game before the game continues onto the main menu in which is where the first glitch effect is. I want Project Isolation to be able accessible to everyone however, in the unlikely event it is not, then people should be able to know what this game contains and if it affects them.



This is an earlier version of the accessibility scene. The latest scene has an icon which warns the player.

The accessibility scene needed an accessibility C# script as this scene needed to be told when to start and finish and where to go from there. Saying that, like the splash logo scene, I have faded in the writing for the accessibility scene, and this is done by animating the text is the UI. I change the transparency of the text from zero and after time it goes up to 200. Lastly, I added a countdown timer to the scene as this represents how long the scene will last. The countdown starts from 15 seconds and when it reaches zero then the scene will change to the main menu scene. The script below has been coded and code commented to tell you exactly what every bit of line does in relation to the script.



The next scene after the accessibility scene is the main menu. However, I have already explained the main menu and information can be found in the section called '**3.13** – **Game Architecture**'. Which you will find everything to do with the main menu and the other settings it has.

Credits Scene

The credits scene is the last scene where the player will go to if he/she has completed the game. The credits scene takes a main camera and moves you around the scene. While doing that, credits start playing which tells you everyone who has made helped make Project Isolation the way it is. While this is all playing, the music starts as well and at the end the game's title comes up and as soon as that stops then the game quits. There is a camera animation which makes the camera go through the scene and there is also an event on the animation which tells the script to play. Below you will see the C# script for the credit's scene:

I have code commented everything in this script, so it tells you exactly what each piece of code does.

```
public class Credits : MonoBehaviour
     // References the creditsCamera which has the animation on
     public GameObject creditsCamera;
     //References the credits canvas GameObject
     public GameObject creditsCanvasGameObject;
     public GameObject doubleDoors; //DoubleDoors GameObject
     //AudioSource for the horrorMusic
     public AudioSource horrorMusic;
     // Start is called before the first frame update
           Message | 0 reference
     void Start()
     {
         horrorMusic.Play(); // Plays the horror music
         StartCoroutine(StartAnimation()); // On start run the Coroutine which is called StartAnimation
     3
     // StartAnimation IEnumerator
     IEnumerator StartAnimation()
     {
         creditsCamera.GetComponent<Animator>().SetTrigger("CameraAnimation"); // Tells the camera's animator to play the next t
         yield return new WaitForSeconds(5); //Waits 4 seconds before playing the next script
         vield return null:
         creditsCanvasGameObject.SetActive(true); //Sets the credits to active
         creditsCanvasGameObject.GetComponent<Animator>().SetTrigger("Finish"); //Plays the 'Finish' trigger on the animator
         StartCoroutine(DuringFinishAnimation());
     3
     1 reference
     IEnumerator DuringFinishAnimation()
         yield return new WaitForSeconds(42.5f); // waits 42.5 seconds. Which is the length of the camera animation
         doubleDoors.SetActive(true); // Sets the doubleDoors to active
         yield return new WaitForSeconds(3); //Waits 3 seconds
        // SceneManager.LoadScene("Main_Menu"); // Goes to the Main Menu scene
         //Destroy(creditsCanvasGameObject); //Destroys the Credits Canvas
```

3.5.6 Player View

The player in Project Isolation, is a first-person character. Meaning, you will not ne able to see his face or body. The only view you will see, is through his eyes. Below is the view of what you will see through the player:



View from the player in Project Isolation

The way the camera has been set up, is that whenever the player moves (keyboard) then the camera follows. Also, whenever, the player moves left or right by using the mouse, then the camera's position also changes as well. The video below explains the way the camera moves better:



Project Isolation's screen size has been set up in Unity Engine to be 1920 x 1080 resolution screen size. Meaning, the screen will take up the whole of the monitor and be full screen. Initially, this was not set up but once the project was coming to the end of the production, I then changed the resolution size.

# Scene	🕶 Game	
Display 1 🔻	Full HD (1920x1080)	

1920 x 1080 screen size

3.6 Slenderman

In Project Isolation there is only one enemy in the entire game. This is Slenderman. A 6ft tall, cloaked creature who lurks throughout Project Isolation. Even though Slenderman was an asset used externally online he comes already animated with a few animations which are an idle, walk and run animation. Since Project Isolation is a horror game, it is only fitting to add jump scares through the game. I will not go into detail about the jump scares i.e., where they are located however, they are in the first scene. The first jump scare, Slenderman jumps out of nowhere and takes damage from the player which decreases his health. While he does that, he also screams. Now, the second jump scare is a bit different. When the player pushes a particular keyboard button in a specific spot in the level, then the players main camera suddenly becomes temporarily disabled. A second camera then plays. This camera is fixed on the cut scene and when this happens, most of the lights turn off and Slenderman suddenly appears and runs at you and screams. When this happens, he also takes damage from the player. If the player ends up with zero health after the second jump scare, then he dies and goes to the game over scene.

3.6.1 Slenderman's Definitions

The only enemy in Project Isolation is Slenderman. Slenderman is 15-feet tall and always wearing a black suit. Carrying on from the story, Slenderman has taken over the prison and the main objective is to find him and go one to one with him to kill him. However, to do that, you must go around the first scene and collect the ammo pouches. You must also find the gun as without it you cannot kill Slenderman. Once you have collected enough bullets and the weapon, on the second scene you must kill him before he kills you. Slenderman is in his walk state at the start of the second scene.

3.6.2 Slenderman's Properties

There is currently only one property on Slenderman in Unity Engine. That is the animator. The animator controller allows Slenderman to have various animations within his properties. Slenderman has currently 4 different types of animations. These are idle, walk, run and attack.

In the second scene for Project Isolation, Slenderman has the 'guardAI' script attached to him. This allows Slenderman to walk on his own using 4 different sized paths. These paths are positioned in the scene (point a, b, c and d) and he walks to each of the paths. The 'guardAI' script has also been coded that when Slenderman reaches one of the paths, he rotates and continues to walk.

The 'guardAI' script has been specially coded in a way that when Slenderman walks then the walk animation also starts. The walk animation is on a loop so it will never stop as long as the animation is being called in the 'guardAI' script. So, in the section in the C# script where I code to say the guard follows the paths, I have added 'Slenderman.GetComponent<Animator>().SetTrigger("Walk");' which basically tells the script to run Slenderman's animator if the trigger is called 'Walk'. He then walks whenever he follows the paths. Slenderman also has a spotlight attached to him which when the player comes closer and in range of Slenderman, his spotlight colour changes from yellow to red. When the spotlight is on red, then the player has a split second to get out of the Slenderman's spotlight, otherwise Slenderman will disable the players controls and the Game Over UI screen will appear.

🖷 🗶 Guard (Script)				0	콽	
	Guard					
Slender Man	Slenderman Level 2					
Path Holder	APath (Transfor	m)				
Gun Script	None (Gun Script)					
Health	100					
Spotted Player						
Speed	3					
Wait Time	0.3					
Turn Speed	90					
Time To Spot Player	0.5					
Spotlight	📥 Spot Light (1) (L	ight)				
View Distance	10					
View Mask	Mixed					
🙀 🗹 Box Collider				0		
Edit Collider	ሌ					
Is Trigger	~					
Material	None (Physic Mat	erial)				
Center	X 0.162674	YO		-0.21096	88	
	X 3.293945	Y 1		2.06036	4	
	Add Component	Add Component				

In the script which is called 'Guard' I have referenced the health, Gun Script and 'SlenderMan' to all find the 'GameObjects' or relevant scripts when the scene is automatically loaded.

Inside of the 'Guard' script you will find everything to do with how Slenderman functions. In this script is everything from the way he walks to how he takes the player's damage if you happen to go near him. Briefly, Slenderman walks and is able to follow invisible paths which are inside the game (just not in view of the player) as inside the 'Guard' script, I have code commented most of the lines of code which says what exactly that line of code is doing in relation to how Slenderman functions.

Slenderman's Health

The way I did Slenderman's health bar system is very simple. I wanted him to have the exact same health bar that the player has. So, for that, it was finding the health bar canvas for the player and then duplicating it for Slenderman. The only difference I had to make was to re-position where I wanted Slenderman's health bar to display. As the player's health is visible on the screen positioned in the top left-hand corner. So, for Slenderman's health bar, I had to re-position it so it was above his head and only visible when the player goes near him. I ended up doing all of Slenderman's health functions and variables inside the 'Guard' script as I only wanted Slenderman to only be having one script which held everything in. This would have been fine if the 'Guard' script was not packed with a lot of lines of code all doing different things. In the end, things started to look messy even with the code comments that state what the code was doing. When doing the health bar system for Slenderman, his current house variable was the same as the players. So, when I was doing 'Debug.Log(currentHealth);' I was getting confused and thinking that this was using the players current health. Since the player had the same variable which was called 'currentHealth'. This was also confusing me when using Debug to try to find out why Slenderman was taking health but was not updating his health bar canvas.

Resolution: In the end, I decided to abort his health functions and variables in the 'Guard' script and ended up creating a separate script on Slenderman which was called 'Health'. Even though this took a bit of time to do because I was having to change different variables to say 'slendermanCurrentHealth' instead of just 'currentHealth' as I wanted the Debug to only pick up what Slenderman's current health was. This made it clearer for me to find out why exactly Slenderman would take damage but would not update his own canvas. The reason behind this was that his damage was d with the player's canvas. This was because all the same functions and health variables were the same as the player's and Unity was using the player's health canvas instead of Slenderman's.

By creating a separate new script on Slenderman it made it easier for me to locate where the problems were. As not only were the functions all correctly named but also did not havie to look at a lot of other different code from the 'Guard' script.

3.6.3 Artificial Intelligence (AI)

The programming of Slenderman has been designed with difficulty in mind. I have designed one (as others are not done yet) so that he is not too difficult to kill. Also, not too easy to kill either. This will depend on what difficulty you choose at the beginning of the game.

Slenderman's states are as follows:

Normal State: He walks along the waypoints and rotates when he needs to. He does not come off these waypoints as he has only been instructed to follow them.

Detection State: If the player goes into his field of view as the guard is moving, then the guard's spotlight will turn from an ordinary default colour to red to indicate danger. However, the player has a few seconds to get out of his field of view and if he does then the spotlight changes back to the default colour. If the player does not get away from Slenderman's field of view, then Slenderman will follow the player.

Reaction State: Once Slenderman has caught the player, then he will continue to move around the waypoints as instructed. If the player manages to move away from Slenderman's field of view before it turns a solid red colour then Slenderman will ignore it and move accordingly.

3.7 Previous Successful Horror Game titles

In this section you will find a review of successful horror game titles that has inspired me to create Project Isolation into a horror game. Aside from the fact that horror is my favourite genre, Project Isolation was inspired by the following games: Until Dawn (2015) and LIMBO (2010). Two completely different games since one is driven by story and the other is a puzzle game.

<u>Until Dawn</u>

Until Dawn is a horror game that is about choice. When I say choice, I mean your choices in game lead to different consequences. Depending on what you choose through the story, can lead to fatal outcomes later. Until Dawn portrays horror very well. The use of sound, dark scenes and lighting build up atmospheric horror. The game leaves you on the edge of your seat wondering what will happen next. The use of jump scares has been developed in such a way that builds tension and, in some instances, draws the player in closer making them believe that there is going to be a jump scare, but in fact this is not the case. With jump scares sound plays an important part. Until Dawn uses sound also to build tension. In a particular scene at the beginning of the game where Beth (one of the main characters) is trying to find Hannah as Hannah has run off following an argument and she has gone into the forest. Beth gets scared because she thinks someone is following her from behind and the way Until Dawn incorporates sound into making us (the audience) believe that this is the case but in fact it turns out to be something completely different. It is having that feeling of the unknown as you anticipate one thing, and it happens to be something totally different.

<u>Limbo</u>

Unlike Until Dawn, which is a story driven game, Limbo is still based on a story but is a puzzle adventure game. First, the main colours throughout Limbo is black and white. You would not consider Limbo a typical horror game. In some cases, you could not say it was a horror but, the way they use black and white, and the use of atmospheric fog, story and environments could suggest otherwise. The key for Limbo is the nature of 'the boy's' quest and the procession of environments, dangers, and enemies he encounters along the way. Dangers and enemies including shadowy spider-like creatures, humanoids, and other typical creatures. Going back to horror, the game leaves me on the edge of my seat since at any given time a second monster of

obstacle would come out of know where and kill me. It left me wondering if I was going to be able to complete the game as Limbo has its own gruesome ways such as the main character can die which only adds to the intensity.

Now the target audience for limbo is very interesting as yes, it is a horror game but not your typical horror like Until Dawn which age rating is 18+. However, Limbo's age rating is for 15-year old's and older.

As I said in the introduction, these games were my inspiration into making Project Isolation. Not everything in these games, I have used in Project Isolation. However, these games inspired me to use different elements to portray horror such as lighting. Lighting plays a huge part in Project Isolation. From the get-go lights have been used not only to see where you are going but give a sense of atmosphere to the scene. When you collect the flashlight, all the lights turn off. Some lights flicker and sound plays. With the lights being blown out, this already gives off an eerie feel to the game.

Audio is another key element that I have used in Project Isolation. Typical in game objects where the player will be interacting with assets generally have audio on them. Cell doors as an example. As the player has to push a keyboard button to open or close the doors, as soon as the player does this a sound plays which indicate the door is opening or closing.

In game music also plays a part in how the game feels in terms of atmospheric horror. Project Isolation also has music in the game which represents the whole game and story feel. For further information and explanations go to '3.12' which is the audio and music section.

3.8 Project Isolation's Story

A lot has happened since the pre-production stage of Project Isolation. Our overall goal / aim was to follow the story. The story was going to be our main priority into how the game starts and how the game ends. However, this unfortunately was not the case. The story was changed. Given Project Isolation was completely made on my own instead of in a group, I had to justify a story that I was not only happy with, but also, I was compatible of doing in a given deadline. I wanted Project Isolation completed before deadline and didn't want to submit an unfinished project. So, the story was thoroughly looked into.

Project Isolation is still based in a maximum-security prison however, instead of escaping and finding your prisoner file, you are now having to defend yourself and protect yourself from this evil creature that is lurking inside of the prison. Some say he captures his prey. Tortures them before killing them. A lot of murders have happened in Project Isolation.

You are a prisoner in this prison serving a life sentence. Something happens to the electric that is unexplainable and as you exit your own cell, things start happening around the prison. Your own source of light is a flashlight. In which you have to find to progress through the game. However, trying to find the flashlight is one thing and you trying to survive is another. You have to be very careful where you are in the prison as Slenderman may come after you at any time. Be very careful what you open, what you touch. If by a miracle you end up surviving to the next scene, you and the beast is one to one with one another and you have to kill him before he kills you. Can you survive throughout Project Isolation? Time will tell. Goodluck!

3.9 Research

Research was supposed to be Marion's field since this was originally a group project work. However, unfortunate circumstances took place, and this project is now a one-man team. Having said that, I needed to still research a few things in terms of Project Isolation. I wanted to know what kind of assets needed to be modelled and textured in the prison cells. I did not want to rush into Maya and model assets without first looking at examples online. I also wanted to design my models a particular way that when I import them into Project Isolation, that they do belong there. When players play the game, they look at the objects in the scene and know what they are.

During the pre-production stage, the main inspiration for doing a game based in a prison was the TV show called Prison Break. Not only is prison break one of my favourite TV shows, but the way the prisons are designed in the series and also how Michael Scofield who has a very rare condition that allows him to think of ways in which he can be sent into a particular prison to escape with his brother Lincoln Burrows. Inspiration from the TV series I looked at was not only online images but also watching the series as well. This gave me a better insight of the type of things I could use when it came to 3D modelling. Cells specifically and assets inside the prisons I was looking at in which could use with Project Isolation.

Cell doors research

So, a cell door for example. I needed to know what type of material these would have been made from. The common material was steel. So, I looked at images of steel prison doors and this give me an idea on what type of materials or textures I could add into Substance Painter for my doors.



A cell door used for research https://eu.usatoday.com/story/life/allthemoms/2018/12/12/woman-gave-birth-alone-locked-prison-cellinexcusable-incident/2287533002/



Another cell door example https://robertpaterson.com/prison-cell

Another example I used was the Joliet, Correctional Centre, which is in Illinois, United States. Which was Fox River State Penitentiary in Prison Break. The image below is of the abandoned prison which was shut down in early 2002. Since Project Isolation is about an abandoned prison these were the type of images that I was looking for. However, not necessarily the prison cell interior but the way the prison bars are. There is one prison door which slides open to the right and closes to the left and one prison cell that is stable and does not move. Using these types of images would help me give me a general idea on how to make these prison cell doors in Maya when it came to modelling them.



Joliet Correctional Centre prison abandoned Link - https://www.mirror.co.uk/news/world-news/eerie-photos-show-remains-abandoned-9256144

Prison bed research

As for the interiors inside of the prison cells, I continuously researched what kind of objects you would find. I wanted to add a few objects into Project Isolation but not cram a lot of objects into one cell as I wanted to make sure that the player could move. I looked at prison beds specifically, as you would get some sort of bed even if it ended up being just a mattress. With the way that the prisons were designed in Project Isolation, I wanted to design a bed that wasn't bunk beds or any beds that were on top of one another. So further to my research, these were the type of beds that I was looking to model.



Singular bed for a standard prison cell. Link - https://www.irishtimes.com/culture/books/bobby-sands-bed-and-long-kesh-maze-s-afterlife-1.2814608

Now, in terms of 3D modelling a bed, I was looking at previous examples of other people's work which gave me a general idea on what a bed would look like with it being modelled and also textured. I had to bear in mind that when I began modelling, I wanted to make sure that the bed was fairly low poly since a bed is just a rectangle shape on top of other different shapes that are not too complicated to produce. Having a high poly modelled bed would be unnecessary and just use up more processing power in Unity and for my computer.



A 3D modelled of a prison bed and textured. Link - <u>https://www.turbosquid.com/3d-models/prison-bed-3d-model-1179919</u>

Prison toilet research

The next asset that I was looking at was the toilets. Now, the majority of the prisons have toilets in the prison cells. Unlike most assets in prisons, you can model one, texture one and just duplicate them as many times as you want since they are more or less the same as each cell. However, to even think about modelling a prototype of an asset, I needed to do further research into what a prison toilet would look like. As they are totally different to say a house toilet as an example. The examples I was looking at online were the following images:



Prison toilet. Link - <u>https://www.prophire-backdrophire.com/product/0770050-stainless-steel-prison-toilet-h-41cm-x-57-x-52-x-1-off-2/</u>

Another prison toilet I was researching was the following:



Another prison toilet example I researched Link - <u>https://ddimitrov.weebly.com/prison-toilets.html</u>

Prison Desk lamp

The prison lamp was next on my agenda to research as somewhere in Project Isolation, there will be a lamp which could possibly be activated by the player to toggle on and off. Or a desk lamp that is broken but still turns on but flickers. The lamps that were online were different to what I was wanting however, can use them as reference when creating the model. Lighting is very important in Project Isolation as lighting is key for atmosphere. Since this game is a horror game, lighting plays an important role. What I plan to do is have the scene to be lighted with all different kinds of lighting such as: Celling lights, alarm lights and desk lights and eventually have all the lights turn off with a C# script.



Prison lamp

Prison Desk

Another asset that I was researching was the prison desk. A prison has some sort of desk inside the cells which allow the prisoners to have valuable items placed on.



A prison desk example http://mangoandginger.blogspot.com/2010/08/pollard-weekend-eating-antiquing.html

3.10 Level Design

Project Isolation has been changed quite a lot. From the pre-production stage, production started straight after. This section is everything and anything to do with level design and the levels that are in Project Isolation. Specifically, this will be Level 1 and Level 2.

A lot of thought process had to be done into creating the first and second scene in Project Isolation. I didn't want to make the scenes seem boring and not have much going on in them. This is when I began to get creative. I started designing and modelling different types of assets that could be used in the game as well as being duplicated throughout the scene. My main goal was not making the scenes seem to long or too short. I know in most prisons, they have loads of prison cells. However, I wasn't going to be making a particular scene that had over hundreds of different types of prison cells. In the end, I only made a handful since this is what I felt needed. My main focus was working in Unity Engine and focusing on programming however, needed to have different models inside the project which then would enable me to program them to do different things. I wanted the first scene to have a CCTV section in it, since in prisons, cameras are everywhere! You cannot cough without someone knowing about it or seeing it.

Then first scene was done by Jack Venn, who kindly helped me 3D model the scene and then I textured it. The second scene I 3D modeled and textured.

This is the second scene without texture:



Second scene



Another view of the second scene in Maya

3.10.1 Level Copy

In Project Isolation, I would have liked to of done cut scenes. Especially with using the plugin 'CineMachine' which allows you o record your own cut scenes using cameras and animations. However, as I was doing the project on my own, I did not get round to playing with CineMachine. However, in Project Isolation, the player can get from one scene to the next, as there is a C# script that is being called. Often, when the player enters an invisible trigger. Now, since there are 9 different scenes in the game, and the player only plays in two of them scenes, I had to create different C# scripts that allowed the game to go from one scene to another. The script would automatically run when something had been finished, or when the player clicks a button. Below, you will find all the C# scripts that allows the game go from one scene to another:

Micky Studios C# Script

pub	olic Image splashImage; //Splash Image
put	olic string loadLevel; //Load Level
0 L	Inity Message 0 references
IEr	numerator Start()
{	
	<pre>splashImage.canvasRenderer.SetAlpha(0.0f); //Changes the alpha on the imag FadeIn();</pre>
	<pre>yield return new WaitForSeconds(2.5f); //Waits 2.5 seconds FadeOut();</pre>
	vield return new WaitForSeconds(2.5f);
	SceneManager.LoadScene(loadLevel): //Loads the next level
}	
1 re	ference
voi	id FadeIn()
{	
	//Changes the cross fade for the image
	<pre>splashImage.CrossFadeAlpha(1.0f, 1.05f, false);</pre>
}	
1 re	ference
voi	id FadeOut()
{	
	<pre>splashImage.CrossFadeAlpha(0.0f, 2.5f, false);</pre>

UnityScript Oreferences Jblic class Wait : MonoBehaviour
// Float waitTime which is 13 seconds
<pre>public float waitTime = 13f;</pre>
// Start is called before the first frame update
@ Unity Message 0 references
void Start()
1
<pre>StartCoroutine(WaitForIntroToFinish()); //Starts a WaitForIntroToFinish Coroutine }</pre>
//IEnumerator WaitForIntroToFinish function
1 reference
IEnumerator WaitForIntroToFinish()
yield return new WaitForSeconds(waitTime); //Waits whatever the waitTime is which is 13 seconds
<pre>SceneManager.LoadScene("Accessibility"); //Loads the Accessibility scene</pre>
}

Accessibility C# Script

QU	inity Script 0 references
pub	vlic class Timer : MonoBehaviour
{	
13	public int countdownTime; //Makes countdownTimer a number by using an Integar
	public Text countdownDisplay;// Grabs the Text
	@ Unity Message 0 references
-	public void Start()
	<pre>StartCoroutine(CountdownToStart()); //Starts a Coroutine called 'CountdownToStart'</pre>
-	}
	1 reference
1	IEnumerator CountdownToStart()
	<pre>while (countdownTime > -1)// If countdownTime is greater than -1</pre>
	<pre>countdownDisplay.text = countdownTime.ToString(); //Uses the countdownDisplay and converts the integar to a string</pre>
	yield return new WaitForSeconds(1f);//Waits 1 second
	countdownTime; // Decrement
	}
	SceneManager.LoadScene("Main_Menu"); //Loads the main menu
1	}
}	

Main Menu C# Script

1	DIC CLUBS CALCUL . HONOBERGATOR
1Ê	<pre>public GameObject exitUIPanel; // exit panel GameObject</pre>
	<pre>public GameObject tutorialButton; //Tutorial button GameObject</pre>
	<pre>public GameObject settingsPanel; // Settings panel GameObject</pre>
	Dreferences public void PLAYGAME()
	SceneManager.LoadScene("LVL1"); }
	Oreferences
71	{
	exitUIPanel.SetActive(true); // Set exitUIPanel to true
	tutorialButton.SetActive(false); // Sets the tutorialButton to false
	<pre>settingsPanel.SetActive(false); // Sets the settingsPanel to false</pre>
-	
	// For the EXIT Yes button
	public void ExitYesUI()
ŦĿ.	
	Application.Quit(); // Quits the application when you click the yes button }
	// For the EXIT No button
17	0 references
밑	f
	exitUIPanel.SetActive(false); // Sets the exitUIPanel to false - Turning the panel off
	settingsPanel.SetActive(true); // Sets the settingsPanel to true - Turning it on
L	
[}	

Tutorial C# Script



Level 1 Next Scene C# Script



Level 2 C# Script - Player Game Over



Level 2 C# Script – Player Win

```
2 references
public void TakeFurtherDamage()
{
    TakeDamage(UnityEngine.Random.Range(10, 80)); //Take Damage takes a damage range
    if(currentHealth == 0)
    {
        //SceneManager.LoadScene("GameOver");
        SceneManager.LoadScene(6);
    }
}
```

3.11 Prison Break Interviews

The main reason I wanted to make a horror game that was based in a maximum-security prison, I was inspired with Prison Break. I have always been interested in watching prison documentaries which focused on different prisons around the world and how them prisons deal with a whole range in mates all doing different sentences for crimes that they have committed. The documentaries first started out with watching 'The world's toughest prisons' which is a Netflix original series. Then onto 'women behind bars. Which again, is on Netflix. Thanks to Netflix, I stumbled upon Prison Break and after a few episodes I fell in love with the series. It became a show that I would continuously 'binge' (watch every episode one after another) watch. I guess the excitement behind it was that someone (Lincoln Burrows) had been wrongly committed into killing the Vice President's brother (at the time) and he was sanctioned to the death penalty (electric chair). Watching Lincoln's brother (Michael Scofield) do heavy research into the prison his brother was in and then ending up committing a crime to be able to be in the same prison as Lincoln to just eventually break him out, just fascinated me on the dedication and commitment that Michael was willing to go for his brother just shows the amount of respect and love that the brothers have for one another.

In this section of the Games Design Document, you will find extra information regarding the film producers who created Prison Break. The reason this section is here is that it will give you a better in site into how Prison Break became to be a global success. Currently as of time of writing, there are only 5 seasons of Prison Break and if I am honest, probably will not be any more. Prison Break's main influence and producer is Paul T. Scheuring. In one of his interviews on YouTube when talked about Prison Break and the upcoming season 5 release date of Prison Break, Paul described the series of having a massive worldwide fan base who had a strong desire into watching the series and following the storyline. Paul also stated that there was a desire among the actors and producers which enabled them to continue filming Prison Break. The Prison Break producers wanted to continue on with production however, still wanted to produce quality and especially for the audience, something for them to be engaged into the story and not see a 'bad version' of Prison Break. When asked what it in Prison Break's story was that made the producers come back and think about doing another season, their response was that there was a desire among the actors to inhabit those rules for more episode's. Paul also went onto to explain that Prison Break season 5 only had 9 episodes. Paul's main goal was to tell the story of Prison Break in a new way that the audience has never seen before and that was to tell the story in 9 episodes. Instead of 20 - 25 episodes that the other seasons have.

This interview was hold with the executive producers Paul T. Scheuring, Vaun Wilmosts & Michael Hororwitz. The YouTube video can be found here: <u>https://www.youtube.com/watch?v=gMXDrIPkxZI</u>

In another interview with Sarah Wayne Callies who plays Dr Sarah Tancredi & Wentworth Miller who plays Michael Scofield, Sarah talked about how Dr Tancredi's family ended up being killed because her role into helping to save the brothers in Fox River State Penitentiary. She says that when someone is falling apart and has nowhere to go and things around you starts going wrong, that you start blaming things on people that are not necessarily their problem and that in Prison Break she does not blame Michael Scofield for her father's death. In a separate interview with Dominic Purcell (Lincoln Burrows) he was asked if he gets fan mail from the fans around the world who watch Prison break and his response was that Wentworth Miller gets the most. Then when Wentworth Miller, gets asked that same question he says the Dominic is the most popular. It appears that throughout the interviews of the two brothers, that they have this very strong connection outside of the series as well as in Prison Break. In my opinion, I find them two inseparable.

The YouTube link can be found here: <u>https://www.youtube.com/watch?v=qM87tCGx9n4</u>

As I said that the main inspiration for this game was from Prison Break. Describing the interviews as well as watching them, tells the audience more than just the series. It tells them what the actors are like with one another during set, how strong their friendships are and, in most instances, actors making lifelong friends.

3.12 Game Architecture

Project Isolation has a range of menu screens in game. These are: Main Menu, Options Menu and the Credits Scene and UI. Carrying on from the Pre-Production games design document, these menu screens have been further developed into creating a horror atmosphere ideally for horror enthusiasts. You will find anything and everything to do with menu screens that are in Project Isolation. This section has been broken down into different sections to make this clearer.

<u>Main Menu</u>

Carrying on forward from Project Isolation's pre-production games development where I created a fully functioning main menu, further work has been put in and the main menu is officially completed. Everything inside the menu is working such as buttons. When the player clicks a button then the button will either go to a different scene in game, go to the settings page or quit the game with 'Application.Quit();' which has been coded in a C# script which is attached to the quit button UI element.

Now, Project Isolation is a horror game based in a maximum-security prison. It would only be fitting to design the main menu in such a way that it contains a horror atmosphere. Each button in the main menu has been coded in a way that has audio. In a separate document you will find external sounds, assets and packages which have been used that is not my work for Project Isolation. Having said that, some audio sounds I had created and one of them being was for the main menu buttons. This was recorded from my PS4 start up sound. I needed a sound that was relevant to a CCTV camera audio and needed audio that only lasted around 1 second.

Now, I have talked about the quit button and how that functions. I haven't talked about the play button. Below, you will find a script which the play and quit button is attached to. As you can see there is a play function within the C# script. When the player pushes the play button then he will be transported to the first level of the game, which is thanks to the code 'SceneManager.LoadScene(SceneManager.GetActiveScene.buildIndex +1);'. Which tells the code to go to the next active scene that is the current build index plus 1.

```
public class MenuMain : MonoBehaviour
     //Slenderman GameObject
     public GameObject slenderMan;
      //When game Awake Function
     public void Awake()
          StartCoroutine(SlendermanAnimation()); // Slenderman Coroutine
          slenderMan.SetActive(false); //Sets slenderman to false
     //PlayGame Function
     public void PlayGame()
          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1); // Goes to the next scene with a build Index
     3
     //Quit function
     public void QuitGame()
          Application.Quit(); // Quits the Applicatin
     3
     //SlendermanAnimation Function
     IEnumerator SlendermanAnimation()
          yield return new WaitForSeconds(1); // Waits one second
           slenderMan.SetActive(true); // Turns on Slenderman
          SienderMan.SetActive(true), // furns of Sienderman
SienderMan.GetComponentKAnimatory(); //Gets the slendermans Animator
yield return new WaitForSeconds(1.5f); //Waits 1.5 seconds
slenderMan.SetActive(false); //Sets Slenderman back to false
3
```

Main Menu 3D Scene

Instead of a generic canvas UI with an image UI with a background, I wanted to change things and add something in the main menu scene that I personally have done before. This was adding a 3D scene.

The scene was created in Maya, initially I was going to create a cell room as part of the main menu. However, I wanted to spice things up and add something that was different to Project Isolation and that was the entrance scene to the prison. This is a room that has a computer, some lockers, lamps and windows. The camera in Unity is positioned outside of the scene so you cannot really see a lot as when you are on the main menu, you are looking at the entrance.



The main menu for Project Isolation

Post Processing

This scene did not need to be textured as I had planned to use Post Processing in Unity to change the way the scene looked in terms of colours and effects. There are a lot of effects for the main menu thanks to Post Processing. Which I was luckily able to install as I used it for a previous project as now Unity have discontinued this. There are around 5 - 10 different effects and they all do something different. My main goal was to change the main menu scene colour to black and white. To give it a CCTV main menu scene look.



Post Processing on the Main Menu

Shader Effects

I am more than happy with how I managed to get the main menu looking like a CCTV screen. This was done using a shader in Unity and I managed to get a video online of a static TV without signal. This was then imported into Unity Engine and relevant C# scripts attached and coded so that when the game starts, the video of the static TV plays. On the camera is the script for the shader and there are properties that can be changed such as the position and speed.



Camera Animations

Following the Pre-Production camera animation this was altered for the production stages. In the Pre-Production stage, I created an animation that animates the camera from one position to another and that animation is activated when the player clicks the settings button. I have kept this animation in Project Isolation, however, had to change the way the camera rotates from one canvas to another.

On the first animator on the camera, I have also attached an animation that allows the camera to jump up and down which is named camera Idle in the animator. This allows a camera stutter to happen and works very well with the CCTV shader. Luckily, nothing had to be scripted. The animations play automatically when the scene starts. When the player moves from the main menu scene to the settings scene. The Idle animation for the camera stops and only plays again when you go back to the main scene.


Camera Animator for the main menu

Settings Page

Lastly is the settings page which is in the same scene as the main menu. Further development has been made with how the settings looked previously from the Pre-Production document. A lot has changed. Changing the texts, graphics, sprites and horror elements have been imported into the settings.

Now, for the horror element side of things, I have imported a protagonist which is Slenderman. He is positioned in the settings canvas UI and is also animated. However, I have realised that you cannot really see him moving but he is. He has been animated on an Idle state with an animator element attached to him and also has been coded in a C# script which allows the animation to play when the settings button is pressed in the main menu. His animation is on a loop since he is not walking or running.



Slenderman positioned in the settings page. He has an Idle animation, so he moves whenever the menu scene appears.

The settings page has been changed. The whole Main Menu has the new glitch shader on which gives this glitchy feel to the game. This works very well as it gives off a 'CCTV' look. There are roughly 3 buttons in the settings page: Game, Controls and Video. The Game button takes you to the game canvas which tells you further settings for Project Isolation. This allows you to change the game difficulty from normal to hard as well as the music volume and various other game settings. The Video button allows you to change the full screen mode of the game. However, this will be changed to 'Story' which will tell you about the story of the game. Lastly, the Controls button allows you to change the controls to Project Isolation. However, I will remove this later on.

		SETTING	
normal is hardcore	GAME game difficulty music volume show hud tooltips	Controls Video	
		Return	

Settings page for Project Isolation

Game buttons

Most of the buttons on the main menu scene do things. Buttons have two different sounds on them. One of them is a hover sound; if you hover over the button then a sound plays. You also have a click sound on the button so when you click the button a sound plays. Lastly, the buttons have a 'GameObject' on them. Which allows me to tell the buttons where they should go to when the button is clicked. An example is when you click the return button, you go back to the main menu and this is thanks to the way the camera is set up via animations.

On Click ()		
Runtime Only 👻	MainMenu.Position2	
🛯 Camera (Main Menu 💿		
Runtime Only 👻	AudioSource.Play	
Button Click Sound 💿		
		<u>+ -</u> [
🐞 Event Trigger (Ren	noved)	
🔻 🐞 Event Trigger		0 ‡ ∶
Pointer Enter (BaseEvent	ata)	H
Runtime Only 👻	AudioSource.Play	
Button Hover Sound 💿		
		+ - [
	Add New Event Type	

Inspector Collaborate Image Type	Simple	a :
Use Sprite Mesh		
Preserve Aspect		
	Set Native Size	
🔻 🔘 🖌 Button		0 ⊒ :
	2	
	Color Tint	
Target Graphic	⊠Btn_Return (Image)	
		1
Highlighted Color		1
		1
Selected Color		ø
Disabled Color		8
Color Multiplier	•	
Fade Duration	0.05	
Navigation		
	Visualize	
On Click ()		
Runtime Only 👻	MainMenu.Position1	
🗈 Camera (Main Menu 📀		
Runtime Only 🔹	AudioSource.Play	
📕 Button Sound (Audi 💿		

Credits Scene

Every game and every film that is produced has a credits scene of some sorts which outline key game roles and names of the people who have either worked on the game or film. To do this, was like the way I produced the Main Menu but did things a bit differently. I had to import the prison scene from Level 1 and add the prefabs into the scene. I then had to animate the camera, so it moved around the scene.



First example of the credit's scene for Project Isolation

With thanks to Veeshane who gave me advice over the way the credits scene looked; I added the following to the scene:

- The glitch effect shader from the main menu onto the credit's scene camera
- 'Cam 02' to the credit's scene canvas
- Horror music into the scene

The way the credits scene works is that the scene will only appear when you have completed Project Isolation. If you happen to lose then the Game Over screen appears and you will have the options to try again or go back to the Main Menu.

Cam 02 BA(Hons) Games Design - Year 3 Final Major Project University For The Creative Arts Project Is Michael En

3.12.1 How to play Copy

Project Isolation has two Menu screens. The first one has already been spoken about at length which gives the player three options: Tutorial, Options & Quit. When the Options button is pressed then the menu changes with camera animation. When the player clicks the tutorial button, the game will go to the tutorial scene. The tutorial scene in Project Isolation describes how to play the game, how to use the inventory system and item picks up that this game has.

The way the tutorial page has been designed, is with horror in mind as this is a horror game based in a maximum-security prison. I used some of the models from the Level 1 scene for the tutorial scene. Such as the prison toilets, bed and the main scene. Instead of manually adding them to the tutorial scene, adding materials, setting up the lighting etc. It was quicker to add the models in Level 1 as prefabs and just drag and drop the prefabs in the tutorial scene. This was done effortlessly and with no hassle. With the way I wanted the tutorial scene to work, is when the player pushes the start button then something happens in one of the prison cells. That is where Slenderman comes into play.

Slenderman is positioned in one of the cells within the tutorial scene and when the play button is pressed the cell door opens and the door has an opening sound then Slenderman animates. Before he animates, an audio file plays first then Slenderman walks out of the cell.



An earlier version of the tutorial screen

This is all done by using one C# script. The C# script is attached to the play button and is done by using arrays. Arrays help me collect more than one object. Arrays have been used a few times in Project Isolation. When the lights turn off and when the cell doors open when the player pickups the flashlight. There probably were other ways I could have done this script however, arrays seemed to be the quickest and most efficient. Even though technically only two things were happening in this tutorial scene which was the door opening and Slenderman animating. I still chose to do arrays simply because I like C# scripts to be well structured and less confusing when I code. If there is a simple way of doing something, I will do it as I do not want to commit all my time to making something work but making my script look confusing.



Tutorial Scene updated 20/05/21

```
∋public class ButtonClick : MonoBehaviour
     // Inherits the Start Button UI
    public Button startButton;
    //finds the cellDoors GameObject
    public GameObject cellDoors;
    public GameObject slenderMan;
    // An AudioSource for the Door Open Sound
    public AudioSource DooropenSound;
    // AudioSource for the Creepy Laugh
    public AudioSource creepyLaugh;
    // Start is called before the first frame update
    (9) Unity Message | 0 references
    void Start()
    {
         //Gets the button component
         Button btn = startButton.GetComponent<Button>();
         btn.onClick.AddListener(TaskOnClick);
    }
    // On button click
   public void TaskOnClick()
    {
         //Plays the cell door animation
        cellDoors.GetComponent<Animator>().SetTrigger("OpenClose");
         // Plays the doorOpen sound
        DooropenSound.Play();
         StartCoroutine(SlenderManAnim());
        // Destroy(DooropenSound);
    3
```

```
1 reference
   IEnumerator SlenderManAnim()
    {
        //Waits 1 seconds
       yield return new WaitForSeconds(1f);
        // Plays the creepy Laugh AudioSource
        creepyLaugh.Play();
        //Waits 2 seconds
        yield return new WaitForSeconds(1f);
        StartCoroutine(WaitSecond());
        slenderMan.GetComponent<Animator>().SetTrigger("GO");
        yield return new WaitForSeconds(5f);
        LoadScene();
   }
    public void LoadScene()
    {
       SceneManager.LoadScene("Main Menu");
    }
   public IEnumerator WaitSecond()
   {
        yield return new WaitForSeconds(1);
        Destroy(DooropenSound);
   }
}
```

The above script is very self-explanatory. Since I have added coding comments however what the script is essentially doing, is when the button has been clicked then a sound plays, animation plays and then goes back to the main menu.

The cell door animation and Slenderman animation codes are very similar. The way I got the cell door to open when the start button is pressed was adding an 'OnClick' method. Inside this OnClick method was everything I wanted to happen when the button was pressed which includes the animations. Since I was using arrays, I labelled them as 'celldoors[1]' (meaning the first cell door) which upon further programming, I could expand that line of code further and add 'celldoors[1].GetComponent<Animator>.SetTrigger("OpenClose")'. This told the computer that when the start button is pressed to start the cell door animation only if the trigger in the animator is referenced 'OpenClose'. This was the same method I used for the Slenderman run animation.

Then once the doors open, Slenderman leaves his cell there is a 'Coroutine' function in the C# script which basically suspends an execution along with an 'IEnumerator' method. Inside that 'IEnumerator' is a 'yield return new WaitForSeconds(2)' the '(2)' tells the computer to wait 2 seconds before going to the next line of code. After the Coroutine finishes, the C# script is programmed to go to the next scene which is called 'LVL1'.

System Requirements

Project Isolation has been optimised to perform without problems such as slow performance. The way Project Isolation has been optimised is that there are no unwanted files within the project folders. Meaning, everything in the project is stored within the projects folders and any files, folders, audio, assets, scenes, C# scripts, images and Unity Engine packages that are not being used for Project Isolation are removed.



Unwanted first-person controller scripts which have been deleted as they are not used. This frees up space in Unity.

Regarding C# scripts, every script that is being used in Project Isolation has also been optimised. With the way Unity Engine works, if a particular script has been coded incorrectly or does not make sense, then Unity Engine checks that script and if the code does not comply with the engine, then the Unity's console will flag up red which indicates a particular C# script is not working and tells me exactly what coding line(s) in visual studio that Unity Engine does not like. When this happens, these particular scripts that are red from the console, mean that I cannot play or build the game until these scripts are working.

Project Isolation can still be built or tested within Unity Engine if the console displays yellow in the console. These are called yellow warnings. Yellow warnings are very helpful in programming within Unity. Even though you can choose to ignore these warnings and forget about them, I do not. These warnings are essential part of games design and shaping the way this project looks. These warnings indicate pieces of code lines within your C# scripts that **could** indicate a potential bug. I highlight the word could in bold because it may or may not happen. The computer checks the code and realises that there may be a potential bug within the script. However, the computer is 50/50 whether or not it is.

Another type of yellow warning which has come up when creating this project is codes that I have defined such as audio. In the C# script I have told the script what to do with a particular audio source and I have not yet set that audio in Unity to the correct C# script that it is referenced. Unity will check this and display a console message saying I have declared what the audio should do but have not assigned it. Again, even though I can ignore this, I do not as I want Project Isolation to run flawlessly and without problems. Even if I did leave this and the console message displays this, the computer is using processing power to check every C# script in that project so adding unnecessary yellow warnings in the console is just building up. By minimising this, Unity and the computer uses its processing power efficiently.

In my previous Games Design Document for the Pre-Production unit, I was doing the project on my MacBook Pro and was coding as usual but also adding code to support PC as the control input was different for MacOS and Windows 10. However, as the project grew, I did not think things through and was having performance issues with the project on my MacBook. So, from there on I transferred the project files to my PC and changed the code to only work with Windows 10. As I initially left the code in the C# scripts and occasionally Unity Engine would give me several yellow warnings in the console telling me that the codes are for a different operating system. Therefore, to avoid any potential hassle later on in the development stage of Project Isolation, I removed them.

4 Project Isolation's Technical Document

Welcome to the technical document of Project Isolation. The technical section is everything technical that went on in the production stages of this game. In this section you will find in-depth explanations and resolutions of how things have been made and created.

The technical section will also describe the modelling and texturing process that shaped the way Project Isolation looks. Bear in mind, when it came to texturing the models within the project, this was all new to me. So, some models were difficult to texture and that could have been how I was texturing the models in Substance Painter or how the UV looked on the models in Maya. In some instances, I had to create the model again.

4.1 Animation

In this section, you will find everything to do with animation. A lot of things in Project Isolation required animation. Animation made my life easier as I did not need to code anything when animating game objects. However, if I wanted the animation to work with a button press or if a player went into a particular trigger in game, then I would have to do a few lines of code. This section goes into depth how particular game objects were animated. In this section also, you may find a 'Problem / Solution' this is a problem that I found when animating game objects and a solution in which I found.

For more information into animation please read this blog post:

<u>https://projectisolation12.blogspot.com/2021/05/rcgd6002-animation.html</u> which goes into more detail on how I went about animating Project Isolation.

Cell Doors Animation

The cell doors had three different animations on the cell door animator. The first was an idle state which is basically what the animator plays first. The second is the open animation which opens the door within 2 seconds and lastly is the close animation which closes the cell door. All the animations are connected to the 'Cell Door Animator' with the idle animation being the default state, followed by the open animation and close animation. However, the open and close animations in the animator have a trigger which is called 'OpenClose' and this is referred to one of the C# scripts that the first-person character is carrying.

Problem / Solution

Now, with the cell doors. I had been animating every single one of the doors that I wanted the player to be able to open and close. That means I had a folder in Unity named 'CellDoor01, CellDoor02, CellDoor03' etc, and inside these folders had the 'idle, open and close' animations in each and every folder. When I told Trev about this as he asked me how I was animating the doors and when I told him about my method, he told me there was a quicker and easier way to animate selective doors. He told me to just add the 'CellDoor01' animator to all of the doors that you want the player to open and that all the doors that I have had to animate I could delete. Not going to lie, I was a bit annoyed and frustrated as I spent quite so many hours animating all the doors and when he told me that there was an easier way, I felt cheated. Having said that, I was very thankful for his advice and changed my way and used his method.

The idle animation for the cell doors is just a static position. Meaning, the idle animations do not move as these are played as soon as the game starts waiting for next instruction. I just made a start and end for the animation.



Cell Door idle animation

Now, the cell door open animation is different to the idle animation. With the door opening, I set a start and end animation state. However, I changed the end state to a different position, so it left the door open. So, when the open animation plays, the door goes from being shut to open. All I then did was turn off the looping as the player was going to open the door with a button interaction. Then in the animator I set a trigger and referenced that trigger to play every time the player pushes a particular button.



Cell door opening animation.

Lastly, it was the close animation state. The close animation state is almost the same as the opening animation for the cell doors, however, the position of the doors is opposite to one another. So, the close door animation is the cell door open but then closed when being played.



Camera Animation

In Project Isolation, there are 3 different cameras that are animated. The first camera is the main menu camera that animated from left to right and back. The second is the jump scare camera. Followed by the second jump scare camera and lastly the credits camera. The cameras will go in order of appearance first.

Main Menu Camera Animation

The main menu camera has a few properties on it; however, we are focusing on the animator. When you play the game, you will notice that there are a few animations on the menu. This is because the camera has 2 different animations all connected to 1 animator. We have the camera jump animation which moves up and down at a particular speed. I have created this animation in relation to the way I want the main menu to look. I want the camera to move and up and down to give off a 'broken' feel. Since the main menu has been designed so that the player is looking through a CCTV view. The second camera animation is the way the camera goes from the main menu to the settings page.

The first jump scare is done via a trigger and when the player steps on it then Slenderman appears.



The first jump scare

This is the second jump scare. On a button press Slenderman will run at you:



Jump Scare Animation (1 & 2)

The jump scares are done by a trigger in Unity Engine. When the player goes inside one of the triggers, the trigger disables the first-person character, and a second camera comes from being inactive to active. Meaning, you cannot move during this jump scare and have to wait until it is finished to carry on with the game. Then Slenderman appears and has an idle animation attached to his animator.

The second jump scare is very similar to the first. However, a few things are different and have been changed. First of all, there is a canvas UI which is animated so the display goes from bright to low. All that needed to be animated was the canvas' colour and that is animated at around 20 - 40 seconds. Then in the second jump scare C# script the animation is referenced.

For the jump scare camera, all I did was slightly change the position of the camera, so it moved up and down quickly. Then once I was satisfied with this, I duplicated it for around 1 second 40. As this was as long as Slenderman running towards the player.



Credits Camera

The credits camera is the last of the cameras that have been animated. The credits camera when played, moves around the scene, and displays the credits of the game. This is done by a credits canvas which is also animated which shows a list of information relating to the production of Project Isolation. A few text objects start at the bottom of the canvas (out of view) and move to the top of the canvas and disappears and while doing this the camera moves around the scene. The credits canvas UI animation is referenced along-side the camera animation in a C# script and to get them both to work at different times I use the 'yield return new WaitForSeconds();' which allows me to tell the script wait x amount of seconds before doing the next line of code. For the credits camera animation there is an animator on the camera and the animation is roughly 40 seconds to 1 minute long.



■ Project >> Animator	Animation	
Preview 🥥 🚧 🖌 🕨 树 🛛 🔘		
Credits Animation 🔹 💠 🖡 🖡		
Credits GameObject : Anchored Positic -680 •		
Credits GameObject : Size Delta.y -476 •		
Add Property		

4.2 Project Isolation - Audio Content

<u>Audio</u>

Audio in Project Isolation is very important. Not only can it scare the player, but also using audio in any game, it creates atmosphere and tension. Every piece of audio used in Project Isolation has been well thought through and I have tried different types of audio to see if the audio fits in with the game. In both the level 1 and level 2 scene there is a sound manager 'Empty GameObject'. What this does, it holds every audio source. I have done this, so these audio files are not scattered everywhere around the scene. I have made sure that even the audio is neat and tidy in the game. As often, I would want to change a particular audio and then have to go through everything in Unity's Hierarchy and try to find that particular sound. So having it all in one place just organizes everything and is my preference. With the introduction scene, the audio is already built into the .MP4 file. So, I didn't need to add the audio into Unity Engine for that scene as the video already has it included.

Below is a setup that I created in Unity which just keeps the audio files all in one place:

🔻 😭 Sound Manager	
😭 AmmoPickupSound	
😭 GunShotSound	
😭 Guard Door Open Sound	
😭 CellDoor Open Sound	
💮 Desk Door Opening Sound	
💮 Bulb Smash	
💮 Lamp Sound	
🕅 Radio Sound	

Sound Manager which includes all the audio files for the first level.

4.3 Project Isolation - Music

Music is very important in games as well as audio, music can enhance the players perspective on the game. Having the right mood / atmosphere in a game can often lead from how the music has been created and if the type of music fits in with the genre of the game. In Project Isolation, music has been thought through very carefully taking into consideration the genre of the game. Since the game is based in a maximum-security prison, then the music should relate to different types of prison sounds such as doors opening and alarms going off. Since, generally speaking, this is normally what you would hear in a standard prison as well as general chatting and voices. Now since the genre is horror, the music needed to be dark and fast paced.

Thankfully, my work colleague was able to help me out with the music for Project Isolation, since he is studying music production at Christ Church University in Canterbury. One of his units was to make horror music, this went very well with my game since this is horror as well so we could kill two birds with one stone. The music I received was very good. I loved it a lot. The music needed tweaking here and there since not all of the music could fit in the scenes.

This was the music for Project Isolation:



The music was fantastic however, since the music wasn't able to fit in with the entire scenes, I had to shorten it. I used a software called 'Audacity' which allowed me to edit audio files and add in a range of different audio effects. If any, I used the fading in / out effects more than any of the other special effects. The only scenes that this music was going to be in was the introduction scene, fight scene (which is the second scene) & lastly, the credits scene. However, with the credit's scene, I was able to use the full length of the music file and made sure that the camera animation was timed perfectly within the music audio length.

The introduction video within Project Isolation, had this music. However, it needed to be shortened since the the introduction video was only roughly 30 - 45 seconds long. With the music, I also managed to find audio files online of cell doors slamming shut and a prison alarm. Which came in very handy with my game.

The fighting scene has fighting music. To do this, all I did was cut out different sections of the music which I felt didn't relate to any fighting combat. Then once I was happy with the final music, I imported into Unity and made sure to click 'on awake' & 'looping' which sets the music to turn on when the scene is loaded and also when the music finishes, then it will be on a continuous loop cycle.

4.4 Project Isolation's C# scripts

This section is for the C# scripts that are in Project Isolation. Every C# script in this game is 100% working and with no console errors. The C# scripts do a variety of things. This can range from the first-person character to how doors open, lights flicker in game or to how the player goes from one scene to another. This section showcases different game objects that have C# scripts on them and go into further detail on how these work in relation to the game. To make my life easier, in most of the C# scripts I have code commented on them to explain what each line of code does in relation to that specific game object. Since I am a visual learner, having codes that have been commented on helps me because if anything goes wrong with the particular code or I want to use the C# script again in another project it makes it easier to identify. Please note that not every C# script is going to be in this document as I have created potentially over 50 different C# scripts that all do different things and if I listed every single script then this document would potentially go on forever. So, below is a list of the C# scripts that were very important to Project Isolation. Some C# scripts may go into further detail other than the actual script as some scripts were not working correctly when creating them for the game and will explain the problem I had and how I fixed it. Not every game object will need an explanation as to what a particular script is doing, as I code commented what each line of script does in Visual Studio.

Cell Doors

There are two different cell door C# scripts in Project Isolation. The first is the prison cell doors which allow the player to enter a trigger near a prison cell and interact with the door allowing the player to go inside the prison cell and the doors open and close with animation. The second is the C# script for the cell doors using the flashlight. So, when the player collects the flashlight, all the lights and cell doors open. Please refer to '4.1 Animation' of this games design document which has in-depth information on how I animated in Unity for Project Isolation.

The first-person character has attached on him the cell doors C# script. This allows only the player to open and close the cell doors in Project Isolation. To do this, I first needed to create an UI canvas which was renamed to 'Cell Door Canvas UI'. Re-naming my Game Objects and UI panels are very important. As, if I have to refer back to them or change them, I can just locate them by name which saves time. Going back to the cell door canvas UI, this needed a panel attached to it followed by a text object. This text object was then changed to 'Press 'E' to interact'. The panel was then hidden from the scene, as in the C# script this would have been toggled on when the player enters the trigger. Of course, a trigger was set in place near the cell doors and the tag for the trigger has a tag called 'Door' then open the cell doors. Below is the script that was used to make this happen.

The Pistol

The pistol was 3D modelled in Maya and then imported into Unity Engine. The gun had to have a script on it which tells the engine what properties, variables and functions this gun would have. The gun script in short allows the player to fire bullets and shoot from the gun.

Now, on the gun script you will see below a whole range of different variables. These all do different things that is essential into making the pistol work when the gun has bullets:

The damage and max damage basically tell you how much the bullet will cause in terms of damage on impact.

The text variable is the reloaded text UI which is the ammo count UI component. So when the player picks up ammo from the ammunition box then the text variable will update the ammo count UI.

The fps cam is the first-person camera and what this does is position the gun on the right-hand side where the player would be holding the gun with his/her right arm.

The muzzle flash & impact effect are particle systems which one controls the actual bullet and the other is the after effect of firing a gun.

The fire rate is simple, this is just a number in which when increased will shoot the bullet out faster or slower depending on the fire rate.

The impact Force is the force in which the bullet will hit another object.

Maximum ammo allows the gun to have a x amount of ammo before the gun ends up running out and the current ammo just says how much ammo the gun has currently.

Then lastly, you have the audio source. The audio source is the gun shot sound. Which is played every time the player shoots the gun by clicking the left mouse button.

Inspector P Lighting Collaborat	e					a :
Pistol						c • 🕯
Tag Pistol	✓ Layer					•
Prefab Open	Select	Overrides				•
🔻 👃 Transform				0		:
Position	X -6.1488	Y 1.221	z -	15.101		
Rotation	x o	YO	zs	90		
Scale	X 0.01205935	Y 0.01205935	zo	0.012059	35	
🛛 🧮 Semi auto Handoun:Semi Auto I	Pistol (Mesh Filter)			0		
	Bemi_autoHandg	un:SemiAutoPistol				•
▶ 🖽 🖌 Mesh Renderer				ø		1
V B Cun Script (Script)						
Seriet	B GunSerint					
Damage	4					
Max Damage						
Range						
Text	ReloadText (Tex	ct)				⊙
Fps Cam	MainCamera (Ca	amera)				\odot
	WuzzleFlash 1 (P	Particle System)				⊙
Impact Effect	BulletImpactMet	talEffect				•
Fire Rate	1.63					
Impact Force	20					
Max Ammo	50					
Delead Time						
Shot Sound	GunShotSound (Audio Source)				0
Box Collider				e		
Edit Collider						
	~					
	None (Physic Mate	rial)				⊙
	X 10.86896	Y 2.204635	Z 1	3.3413		
	X 27.95502	Y 38.63779	Ζ 5	57.79139		
🔻 式 Pickup Item (Script)						:
	D pickupltem					
	🗊 Gun Image					⊙
	None (Lights Off)					
Cell Door	None (Cell Doors O	pen)				
I ngger Lights						
🔻 🚜 🛛 Has Gun (Script)						
				COLUMN TWO IS NOT		

Next Scene

The next scene script is in the first level (level 1) of Project Isolation. You will find this script near one the the black double doors in the scene. Essentially, what happens is as soon as the player enters an invisible trigger, then a countdown begins from 15 seconds to zero. However, during this, everything in the scene goes dark and the player is unable to move. A footsteps sound begins to play to give the audience a feel that something is happening while the second level begins to load. Below, you will find a code commented C# script that is the next scene.





The scene manager loads the second level 1 instead of the code commented 'Main Menu'.

Notes System

In Project Isolation, the player can read notes that have been made by other prisoners as well as, their responses from family relatives. The way this works is that each letter has a box collider on it so that when the player enters his box collider then I have a C# script that allows the player to read the notes. This had to be created as a 'Note System UI' canvas. The canvas in the scene is turned off and is only turned on when the player enters the box collider of the note. When the player leaves the box collider then the canvas UI turns off. The C# script for the note system is below:

```
M Unity Scrint | 0 rafe
public class Note : MonoBehaviour
 {
     [SerializeField]
     public Image noteImage; //Image
     //OnTrigger enter
     public void OnTriggerEnter(Collider other)
     {
         if(other.CompareTag("Player")) // If its the player with a tag called player
         {
              noteImage.enabled = true; //Sets the noteImage to true
         3
     }
     //On Trigger exit
     public void OnTriggerExit(Collider other)
     {
         if(other.CompareTag("Player"))
         {
              noteImage.enabled = false; //Sets the noteImage to false when left trigger
         3
}
```

Note System C# script

Below, you will find screenshots of the UI components that work with the note system. The note's canvas has an image on it which is checked off and is referenced to turn on when the player has gone inside of the note's box collider. The note's script is attached to every letter:



Note's canvas

The letters were created to give the game more atmosphere. Often, when you go to prison there will be prisoner letters / replies. I created this with the help of my family, so that the game would have more direction. On the letters, you will find the individual prisoner ID's. Below you will find the original letters that were created for Project Isolation:

Jane Smith To ID No1 573 963 235 ISOLATION PRISON Jate 13/03/2004 Dear Jane, How and you keeping You are missed out here very The kids are doing well, I read to them every night, using voices just you do? like With visiting times coming up in a month or so, we should Fingers crossed! yal! 1 will bring pichrel with me that the children have drawn for you. next time wee you. Mose to visit soon. Love of lare, Mahilda Smith

To Jane from Matilda

Mahba Smith
34 Riversdaler
Kent

Date 03/04/2004

Dear Matilda,

lam so somy imised your
Visitation last week, I have been
in solitary confinement.
weind things are happening
neve Matilda. I can't stand it.
Luce wat h scilory for maching
hallos when heavy to cheaning
Have whet saw the manger
thing in prodo dayugur 1
saw a lovig, bark how in a
suit havery in the courtyard.
The gravas thaught I was daily.
The nights are warse than the days.
Muttue screams empt from all.
comes of the prison, and ratting
keys can be neard in all hars
of the night. Something
At first itraght omething was
wong with the electricity, as
doors swing open, and liquits plicker
Canstartly But after seens that
dave shacow, Hear something
much more prister is happening.
If you don't hear from me, send help.
Jane xx

Letter from Jane to Matilda

6 MARTIN JANES ID NOV . 429817639 ISCHATION PRISON, Date S 3/04 Dear Mantin, I hope you are keeping well and healthy, we are both well over here. Country is doing well at school, and tan still working. I hope all being well we stabuld be up to see you nest nionth, I will let you know the date nearer the time when Sorted. We do both niss you and hope it want be too long before you are back hone with us, starting a new life and gitting back to named. I nuist que voue, but will let you know the date when we will be up to see you. bye for now Lots uf have Stella & Countries

To Martin from Stella & Courtney

Stella
23, Godstore Rd-
Kenley _
surrey.
Date: 23/3/2004 _

DEAR STRALA

Thank you for your letter, I was Pleased to hear from you, giad you and courtney are well, I can't wait to see you both, strange things have been happening here. Phisoners are bearing side all of a sudden and at night you can hear Strange noises, and Screens from a distance, you can hear the rattling of keys but this is after Midnight so there is no body around, its as though the prison is haunted, and I'm not one to believe in paranormal conspiracies, but now in beginning to think its all : true, especially when I was talking to a couple of prisoners from another wing and they have been telling ma the thigs they see and hear, plus they have experienced lights flashing and all and thing also said furthe Scare on Strandows Hooma would, when then that ally good. I can't wait to see you both, must go now. Loss Jave Mark wart in XXX0

A response from Martin

4.5 3D Modelling (Maya 2021)

This is the 3D modelling section of the Games Design Document. In this section, you will find everything to do with 3D models and how they were created. All models were created in the following 3D software: Maya 2021. All assets used in Project Isolation were 3D modelled and following a successful model they were then ready for the texturing stage. Which you will find further information in the next section (4.7). Now, once the model was created in Maya, I then imported some textures onto the model in Maya and exported them. This made my life a bit easier, as for smaller objects which would have been positioned on the floor – A needle for example. I did not need to go into Substance to texture it. I did not do this with every model but with smaller objects I did.

Prison Toilet model

Following the research section, it was then time to use the research as knowledge and begin to think about how I would be modelling these assets to use Project Isolation. The first asset was the prison toilet as I was unsure how to model it. However, I did have an image online that I could use as a reference in Maya, and this would have been my template to model a toilet. Once I completed the model, I then started to UV map the model which would let me export the model from Maya and allow me to import into Substance Painter ready for texturing.

UV mapping was all new to me. I did not really understand how UV mapping worked and how you could use this in substance painter. All I knew is that you could texture in Adobe Photoshop. Unfortunately, it is not all fun and games when it comes to UV mapping as on the right-hand side you see squares on the object. You must make sure that these squares are all aligned properly and are the correct size for each one of them. If they are not, then this could affect the way the texture looks on the model.

Another important thing to be wary of is the way the models are aligned in the UV editor (left hand side). If models overlap one another then the texture will not look good on the final model in Substance Painter. So, for this model, I first used the automatic button in Maya which automatically aligned the models into the view, and I just repositioned them, so they were not as close to one another.



Prison toilet UV mapped (16420 Tris)

Prison bed model

The next asset on my list was the prison bed. Again, like the toilet model I was using images that I researched previously online. Which helped me understand where to start in terms of modelling. Now, the bed model itself has been made using 3-4 different shapes in Maya. There is the bed frame, which is where the mattress would lay, there is the mattress, the duvet and the sheets. Just like with everything that I do, I make sure to save my work regularly just in case something unexpected happens and I lose my work.

The most difficult part when it came to modelling this bed was the bed sheets. That is because I used the bend tool in maya which allowed me to bend the sheets / blanket for the bed. Which gave a 'wrinkly' look to the model. I did not just want to copy a model that I researched online, I wanted to create this model differently but keep the whole 'prison bed' feel since this asset was going to be imported into Project Isolation.



Prison bed model (Tris 23,700)

Prison Chair Model

The next model was the prison chair. This model was not complicated to make as it was just a square being extruded in Maya and then other objects being attached to the model. Below is the final model and a UV mapped version.



Chair model in Maya (228 Tris)

Desk lamp model

The desk lamp model initially sat on one of the guard's shelves in Project Isolation. This model was in some ways complicated to make since there was more than one object to this 3D model. However, using references online helped me with shaping the way the lamp would look like completed. I then UV mapped the model and imported the model into Substance Painter.



3D model of the desk lamp for Project Isolation and UV mapped (6046 Tris)

CCTV Camera model



CCTV camera in Maya (Tris 1840)



Guard Door Model

The guard door needed to be modelled and this was straight forward. I modelled a door frame as well as the main door and door handle.



Guard Door 3D model in Maya (460 Tris)



Guard door UV mapped

Ashtray Model



Ashtray UV mapped



3D model of an Ashtray in Maya (14680 Tris)

Chemical Bottle Model



3D modelled chemical bottle (1996 Tris)



Needle Model



Needle model (1200 Tris)



Desk Shelf Model



Desk Shelf (24576 Tris)



Desk shelf UV mapped in Maya

Television Model



Television model in Maya (1956 Tris)



Television model UV mapped

<u>Bullet</u>



Bullet (Tris 262)



First Aid box (pick up)



First aid box (Tris 4478)



<u>Radio Model</u>



Radio (Tris 14868)



4.6 Texturing (Substance Painter)

This section focuses on texturing. I used Substance Painter to apply textures to the models that were created in 3D Maya. Please note that this was my first-time texturing models using Substance Painter. Briefly touching upon our Pre-Production presentation, we were talking about our textures being 512 pixels. However, after careful consideration I changed the resolution size of the textures. Most textures in Project Isolation are 1k maps or 4k maps. These depend on the 3D model. As an example. The cell toilets have a 4k texture on them. The reason for this is, that in Substance Painter under the 'texture settings' page, you can change the way the textures look at different resolution sizes.

So, before I export any textures from Substance Painter to Unity Engine, I look and see which texture resolution is better and use my preferred size. Smaller models such as an ashtray can be textured at 512 resolution since this is a smaller model object and in Unity, the player will not necessary be looking at that model for a long time. Due to the processing power that my PC has when making Project Isolation, I could include 4k texture maps in the game.

For further information and techniques that I used when I was in Substance Painter, look at this blog: <u>https://projectisolation12.blogspot.com/2021/05/rcgd6002-texturing.html</u> which gives you a general idea on how I textured the models and what techniques I used.

Initially, in the pre-production stages we (myself & Marion) were going for a 2004 - 2013 feel for Project Isolation. Below, is the type of resolution we were going for:



GTA San Andreas

Even though we are not going for the same style as Grand Theft Auto (GTA) San Andreas, I was looking into the type of resolution the textures were. GTA San Andreas, was created in 2004. This gave me a general idea / vision of what I wanted Project Isolation's textures to be.

First Scene

The first scene of Project Isolation was produced by a year 2 student who was helping me with the game. However, when the model was finished. I had trouble with texturing this in Substance Painter. Any textures I included onto the model looked awful and out of scale. I tried to UV map the first scene in 3D Maya but was having real difficulty and that is when I decided to ask my lecturer for advice, and he helped me with the model. After that, it was plain sailing and was able to add texture maps to the scene.



First look at Level 1 textured in Project Isolation

Second Scene

Level 2 scene in Project Isolation was textured a bit differently. After importing the model from Maya to Substance, I broke the model down into 3 different sections. The sections were the floor, bars and walls. This made my life a little bit easier. So, instead of applying one texture which would have covered the whole model, I was able to disable which sections that I did not want texture in a particular material.



Second scene floor in Substance



Prison cells and bars for second scene in Substance



Whole model in Substance Painter



Prison cells and bars in Substance Painter texture



Level 2 fully textured in Unity Engine

Prison Desk

Following the UV mapped prison desk model, I imported this into Substance Painter and began texturing the model. Once the model was textured, I then exported the textures and imported them into Unity. This was the first texture I imported into Unity Engine.



Textured prison desk in Substance

I was not particularly happy with the way the textures looked and my brother did notify me that in general prisons they would not have a wooden desk since wood can easily catch fire. So, with that in mind, I changed the way the desk door looked and in Substance Painter changed the texture. I added a rust material onto the overall texture to make the desk look like it has rust which adds aging to the model. Below is the updated version of the model.



Updated texture to the Prison desk

Lamp



Lamp fully textured in Substance Painter

TV


Television model in Substance Painter

<u>First Aid Box</u>



<u>Radio</u>



Radio fully textured in Substance Painter.

Play-Testing Project Isolation

Play-testing allowed me 1-2-1 full on with my own game. Playtesting allowed me to find faults with the game that I may have not noticed otherwise. As well as finding faults with my game, I had to log the faults in a separate document (see the playtesting document) and had to indicate how I went about fixing the type of problems / faults that occurred.

Play-testing gave me the ability to not only see what was wrong with the game, but also able give me the ability to see how far I had come in terms of production. I tried to complete the playtesting coming to the end of the project as the next step after playtesting was releasing Project Isolation online. So, I did not want to release the game to other people for them to be bugs and glitches with the game. So, I got on top of that quickly. Once I did my own playtesting, I then had to release Project Isolation for other people. This gave me an insight of not only what other people thought of the game but also if there were able to pick up anything that was wrong with the game, that I was not able to spot.

I found this stage very exciting as I had friends and family, who were very excited to get a firsthand look into Project Isolation. I ended up creating a questionnaire which laid out all sorts of questions like: Who is the target audience for the game, was the game enjoyable, was there anything in the game that stopped working? Etc. The questionnaire document was created so that I could receive feedback and act upon the feedback that I had received from other people. As well as a document in which I could review from time to time.

I published the build files as well as a 'README' document on OneDrive. The 'README' document explained the story of Project Isolation, and as well as how to treat the game in terms of playtesting. The document outlined what I was wanting people to look for when playing the game. I then also created a feedback log which was also published to my OneDrive. Inside of the feedback log was the log of every update that I did for Project Isolation during the playtesting stages. This was to show people of any updates that I did for the game and if they wanted to play the game again then they were more than welcome to.

Further information regarding the playtesting stage can be found in my reflection document.

Project Isolation Before & After

When I first created Project Isolation in Unity Engine, I kept a separate scene with everything in, except the textures as this had not been done at the time of creation. After going through the completed game, I stumbled upon the scene where I was able to capture a before (pre-production) & after (production) images of everything. Below, is the before and after images as well as other level images that I captured in Project Isolation.



Before Production - Cell View



Before Production – Hallway View



Before Production - Guard's Room



Before Production - Guard's View



Before Production – Further into the Guard's Room



Before Production – Start View.

After Production



After Production - Cell View



After Production – Start View.



After Production – Player View.



After Production - Guard's Room.



 $After \ Production-Guard's \ View \ Room \ 2$



After Production – Project Isolation.

Project Isolation - Scenes in order of appearance



Company Name - Micky Studios logo (Build Index 0)



Project Isolation – Introduction Scene (Build Index 1)



Project Isolation - Accessibility Scene (Build Index 2)



Project Isolation - Main Menu (Build Index 3)



Project Isolation - Settings (Build Index 3)





Project Isolation – Tutorial Camera View (Build Index 4)

Project Isolation - Tutorial Slenderman (Build Index 4)





Project Isolation - Tutorial Players View (Build Index 4)

Project Isolation - Level 1 (Build Index 5)



Project Isolation - Level 2 Cell View (Build Index 6)



Project Isolation – Health pickup view (Build Index 6)



Project Isolation - Player Position* (Build Index 6)

*Player position is where the player will begin when he enters level 2 from the first level.



Project Isolation – Level 2 Slenderman's view (Build Index 6)



Project Isolation - Level 2 Overall view (Build Index 6)



Project Isolation - Game Over Scene (Build Index 7)



Project Isolation – Game Win Scene (Build Index 8)



Project Isolation - Credits Scene Cell Door View (Build Index 9)



Project Isolation - Credits Scene Hallway View (Build Index 9)



Project Isolation - Credits Scene Guard's Room (Build Index 9)



Project Isolation - Credits Scene Hallway View 2 (Build Index 9)

Project Isolation

RCGD6002 Final Major Project – Games Design Document University for The Creative Arts (BA) Games Design

END OF DOCUMENT