



PROJECT 1

Oscar Webb





UNIT 12 P01

Pebble Games



Game Pitch (P1, P2)

- In 'Bottled Up', you play as a test subject in an AI-run research facility. At the start of each level, the player is presented with a bottled liquid that they must drink; this then inflicts one of four emotions onto them. The four emotions included in the game are anger, fear, desire and happiness. Each level includes unique mechanics:
- Anger: Gives the player the ability to pick up and throw objects.
- Fear: The player's adrenaline is increased, making the character speed faster.
- Desire: The player must collect the objects in the level before they disappear.
- Happiness: Explores the idea of 'jumping for joy', and gives the player heightened jumping abilities.

Evidence for (P2)

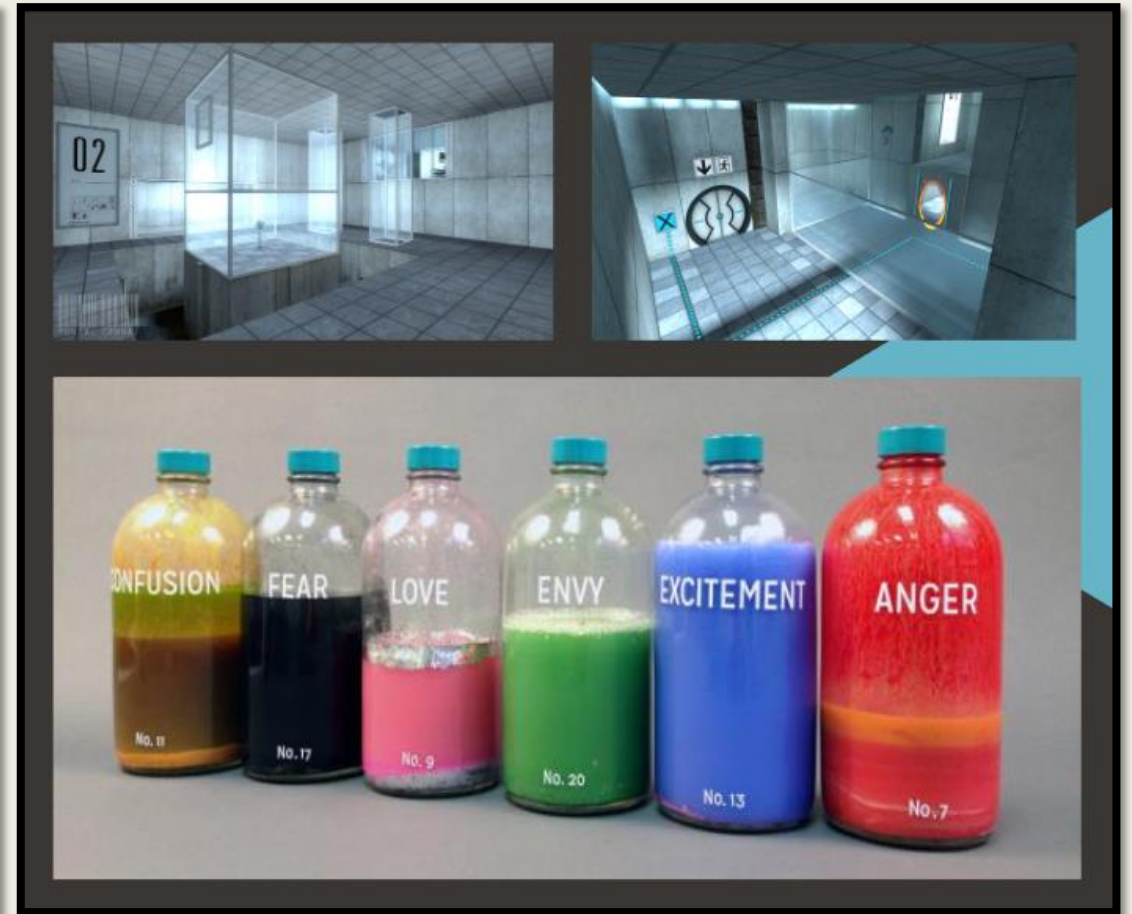
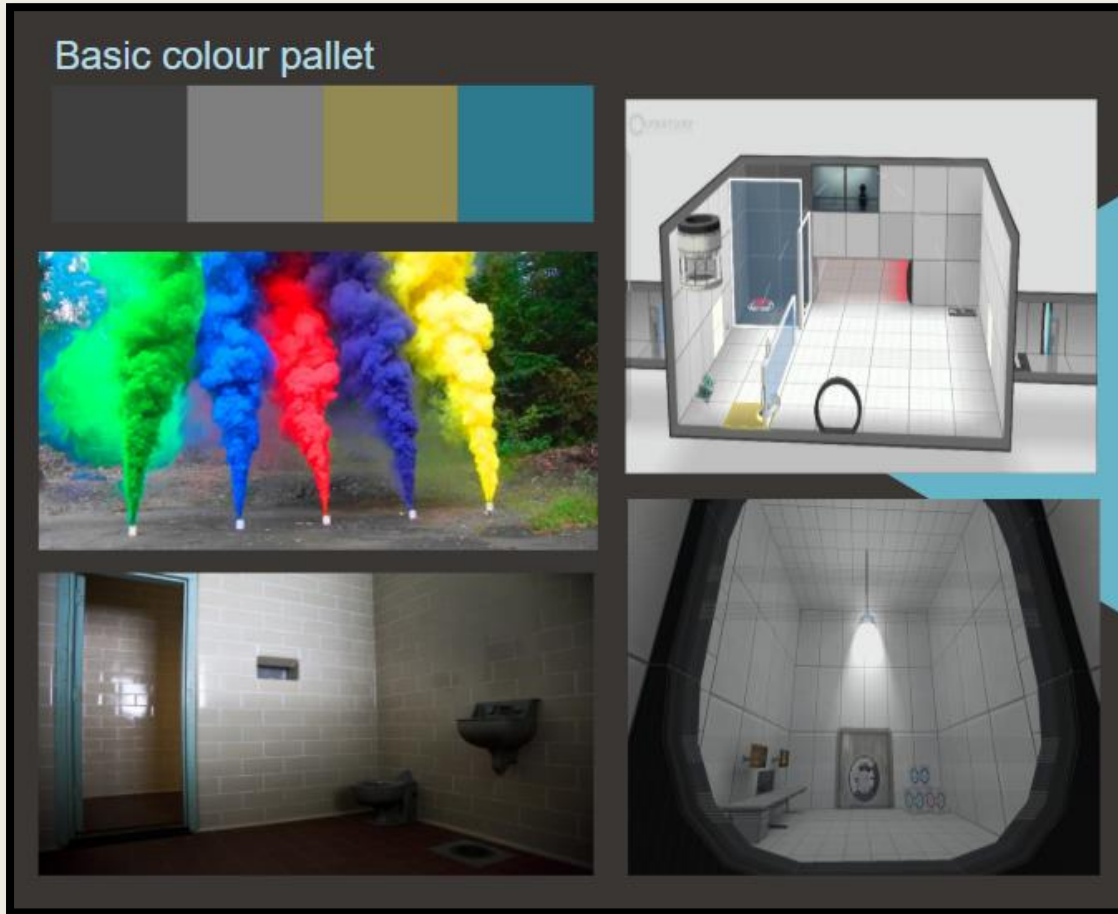


<https://wordpress.com/view/oscarwebbgamedesign.wordpress.com>

The above link leads to WordPress, which contains the past presentations our group has done.



Initial Ideas (P1, P2)



Influential Games (M1)

■ *Fallout 4*



- Metal interior.
- Industrial and worn surfaces.
- Markings and signs on way to go.
- Enclosed area (cannot reach above world).

Influential Games (M1)

■ *Portal 2*



- Enclosed environment.
- Worn and eroded surfaces.
- Mysterious graffiti on walls.
- Decaying facility.
- Mostly tiled and mesh surfaces.

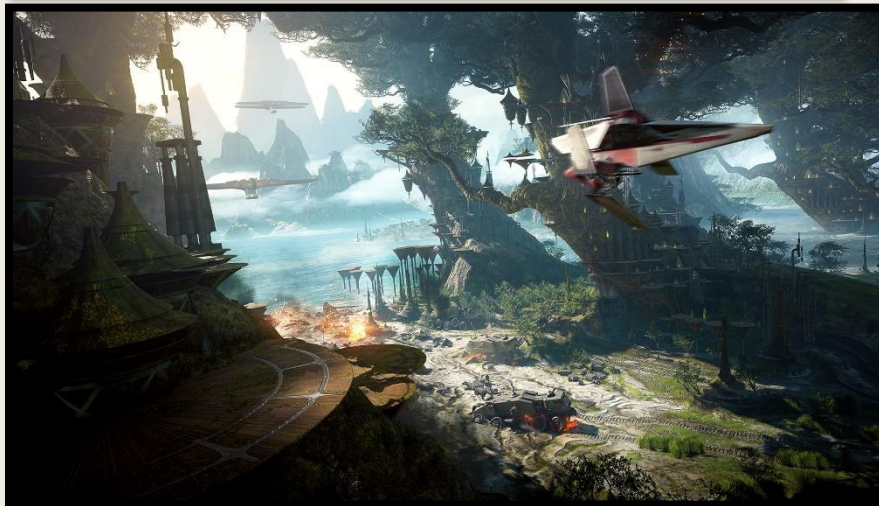


Influential Games (M1)

■ *Star Wars Battlefront II*

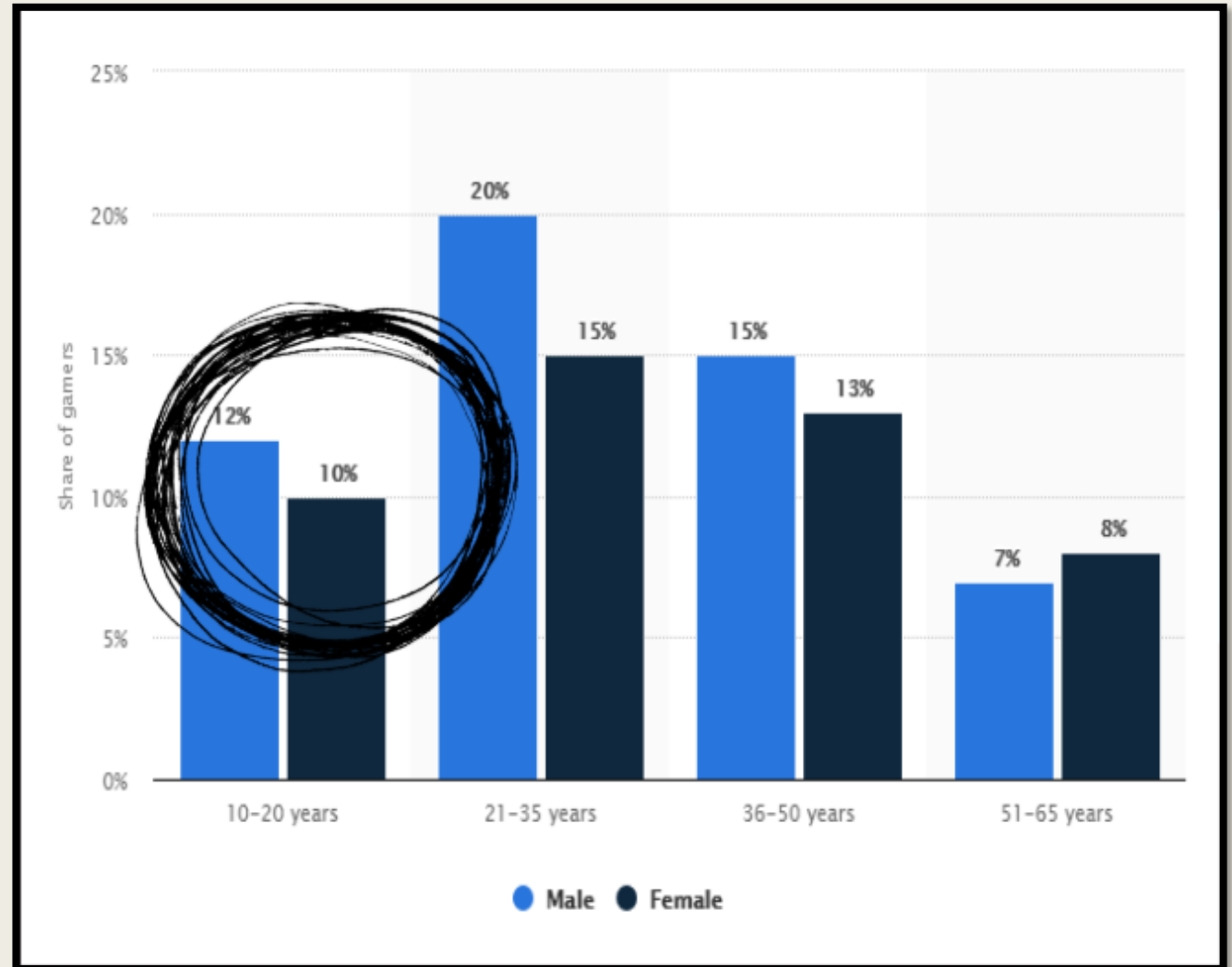


- Futuristic environments.
- Includes both new and decaying level maps.
- Wide use of metal textures.



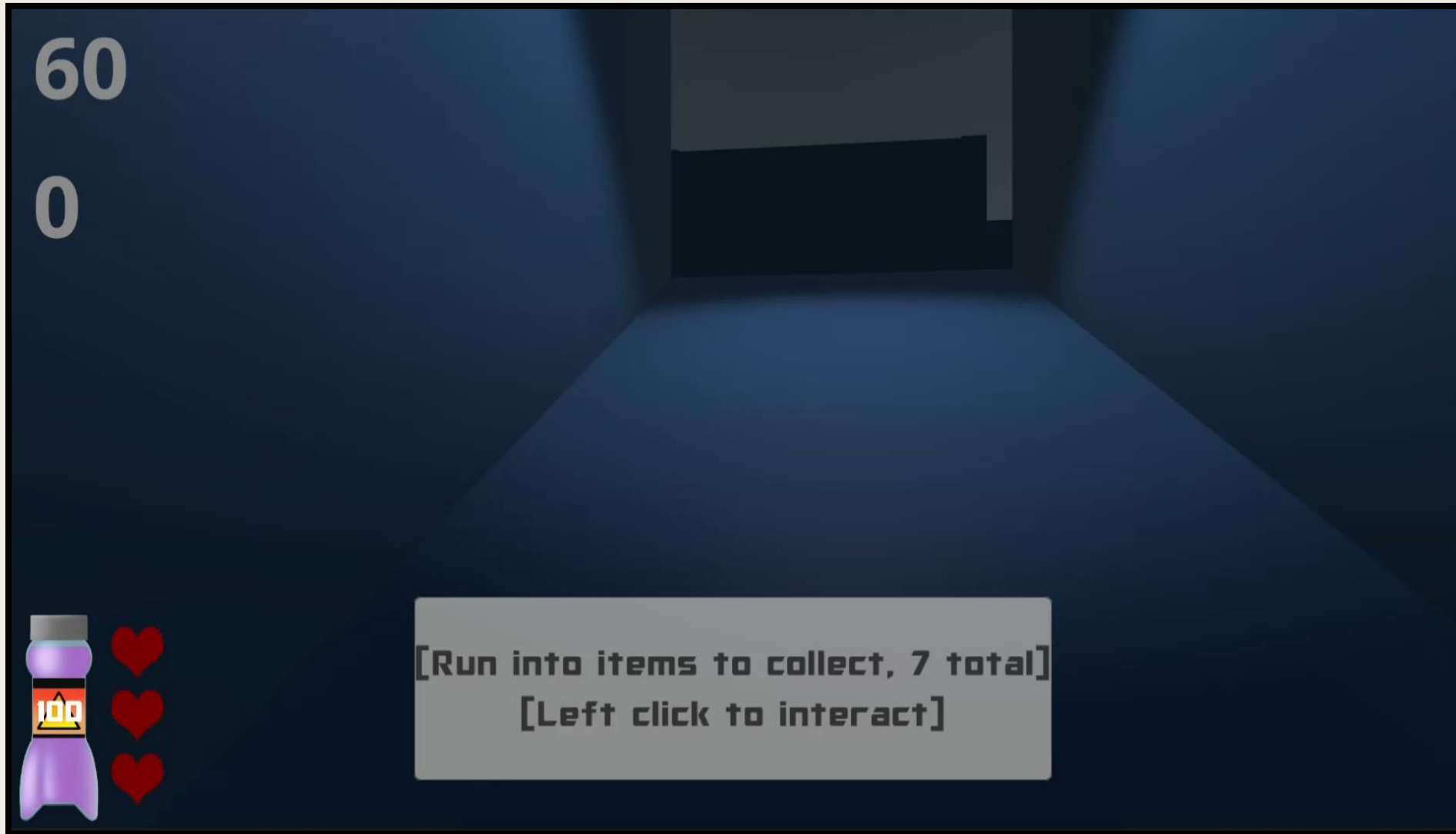
Target Audience (M2)

- The target audience for our game is aimed towards people aged 16-30 years; the male audience is also larger than the female audience. As well as this, we are targeting the game towards PC players, as it is much easier to run and test games, compared to console, VR or mobile. Additionally, the largest demographic of players who play the competitor's games play on PC.



Game Prototype (D1)

<https://youtu.be/aXirOWYpUWY>



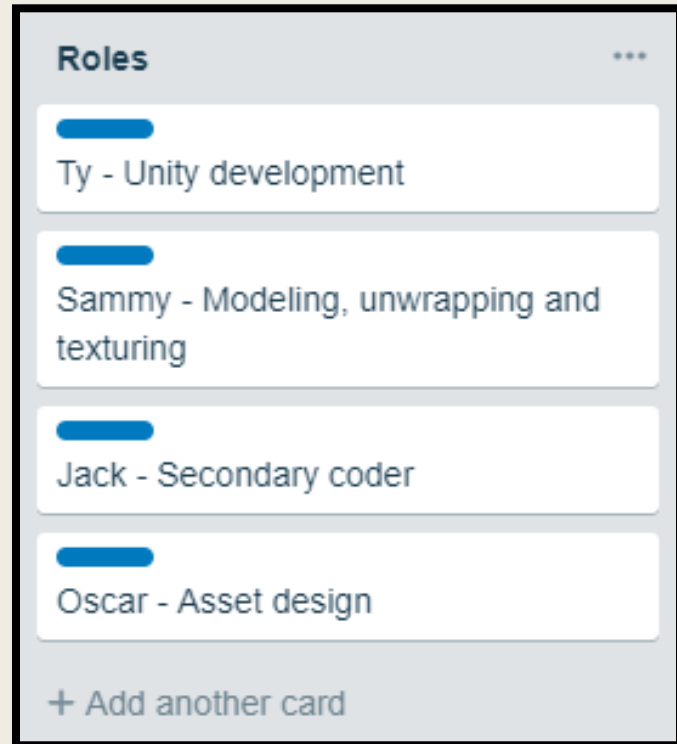


UNIT 12 P02

Pebble Games



Job Roles (P3) (P5)



- Here are our team member's job roles. This was created on Trello and serves as a reminder of each person's part in the team. Tyler is the lead coder, and has helped us all with mechanic errors in our games. Sammy is the lead modeller, and created the modular pack. As asset designer, I modelled most assets, such as collectibles and containers, barrels, and the laser turrets. Jack also helped with some of Tyler's coding and created the laser turret AI.
- These chosen roles complimented our team very well, as we all understood what we were meant to be working on. It also guided us on who to ask for help, if we needed another team member's knowledge, such as code or help with modelling. By having these team roles, we could work more comfortably within our game development team, and could progress through different parts of our game simultaneously while helping each other too.

CV (P4)

PDP (M3)

(D2)

Strengths:	
Modelling:	One of my greatest strengths within games design is modelling. I have modelled multiple objects throughout the course and have gained lots of experience using the program 3DS Max by doing this. Some examples of objects I have modelled are: <ul style="list-style-type: none"> • An apocalyptic dwelling. • A car (Plymouth 1970 Hemi Cuda). • A full western modular pack building, including assets. • An in-game survival asset.
Art:	Another strength of mine within game design is art/drawing. I have created numerous drawings during my time on the course, using pencil and also digitally, using a graphics tablet. I am aware of multiple drawing techniques using Photoshop and Illustrator, which was further improved by what I learnt in Graphics AS last year. With all these qualities combined, I have created a range of concept art pieces throughout my studies.
Education:	I currently study Computer Games Design BTEC Level 3 at Priestley College. I also completed GCSE Art, in which I acquired a C. I learnt multiple techniques within Photoshop, and other methods through completing projects in my sketchbook.
Interests:	In my spare time I enjoy modelling and playing games. Playing games inspires me when creating my own concepts and ideas. Modelling and concept art is my main focus within games design.

Skill Area	Current Level 1-5	Timeline For Improvement	Measure of Success
Modelling	3	5 Weeks	Improve the level of detail in models using 3DS MAX.
UI Menu Design	3	1-2 Weeks (1st Sprint)	Refine the UI menus within Photoshop, so they can be implemented into Unity.
Sound Design	1	Until Sound Design Project Is Finished	Learn how to use the sound design software during Iain's lessons.

I feel as though I completed my goals stated in my PDP. I have definitely improved my modelling skills over the course of this project, having modelled a number of assets for my game; I hope to further these skills over the next project. With the UI design, I completed it within the first sprint. I was pleased with how they turned out, however we used updated designs for our final game. Finally, the sound design aspect of my PDP went well too; having no prior experience with audition, I was pleased with my sound effects as well, and feel I have improved my current level.

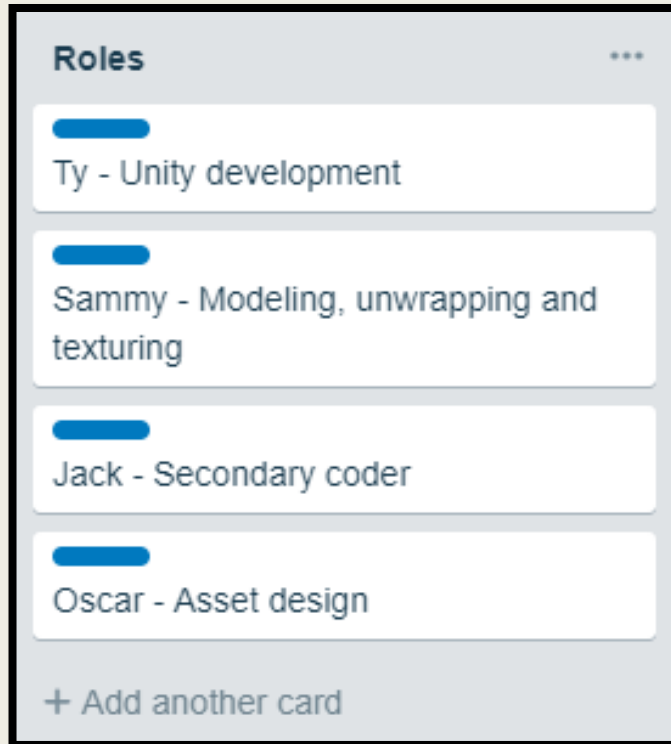


UNIT 12 P03

Pebble Games

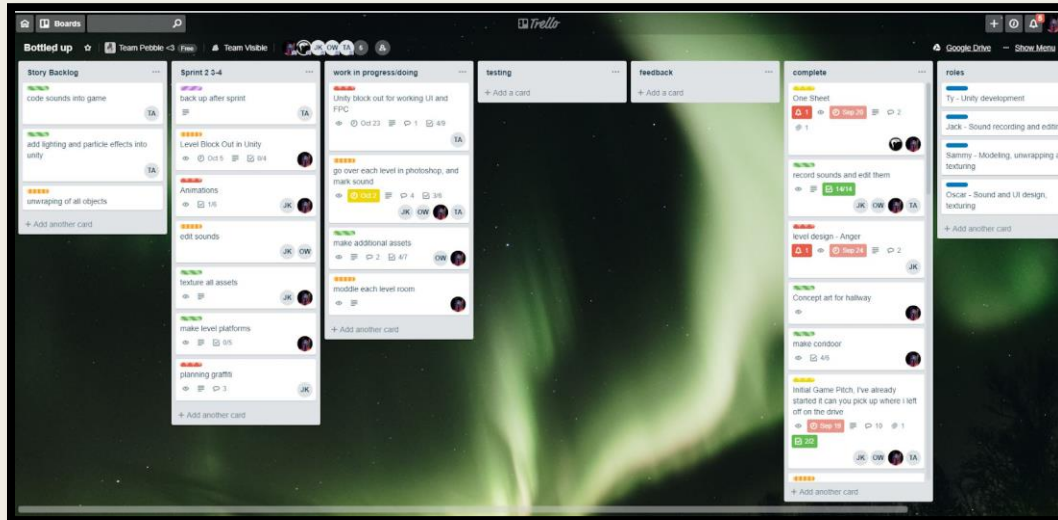


Development Tasks and Priority for my Role (P6, P7)

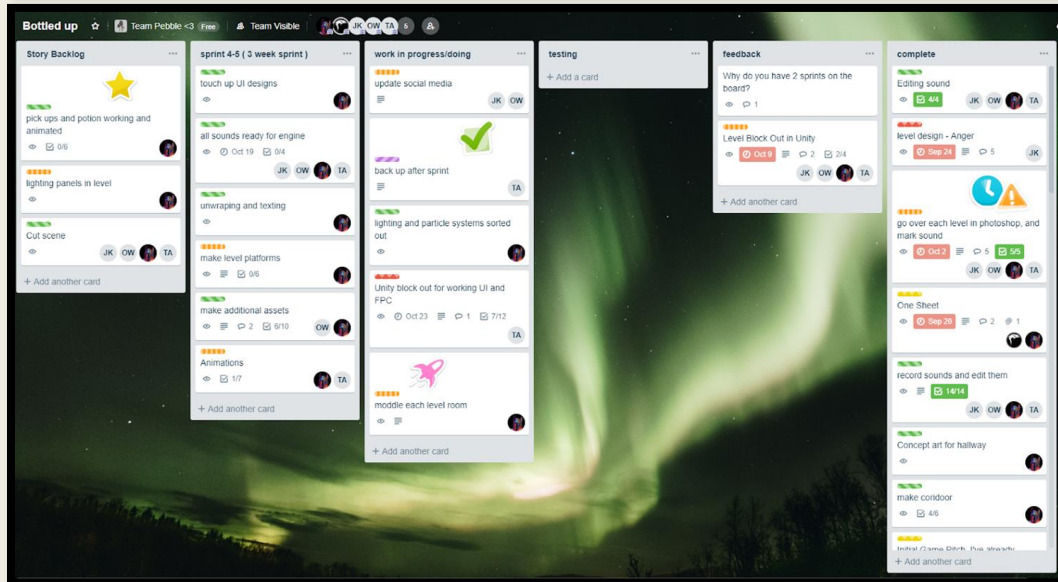


- As my role as asset designer, I have had a number of development tasks to complete. Naturally, I created a number of assets with textures; these were used as both static assets, such as containers, barrels and platforms, and also pick ups such as coins, pocket watch and vase. I also created a laser turret, which has been implemented with working code into each level. On top of this, I have also completed small quantities of code, and helped produce UI designs and sounds.
- The priority of tasks I have completed across each sprint has changed multiple times. This is due to when a task is complete, the tasks below it move higher in ranking and become the next priority; this is referenced in my sprint in sprint reviews.

Trello Sprint Reviews (P8)

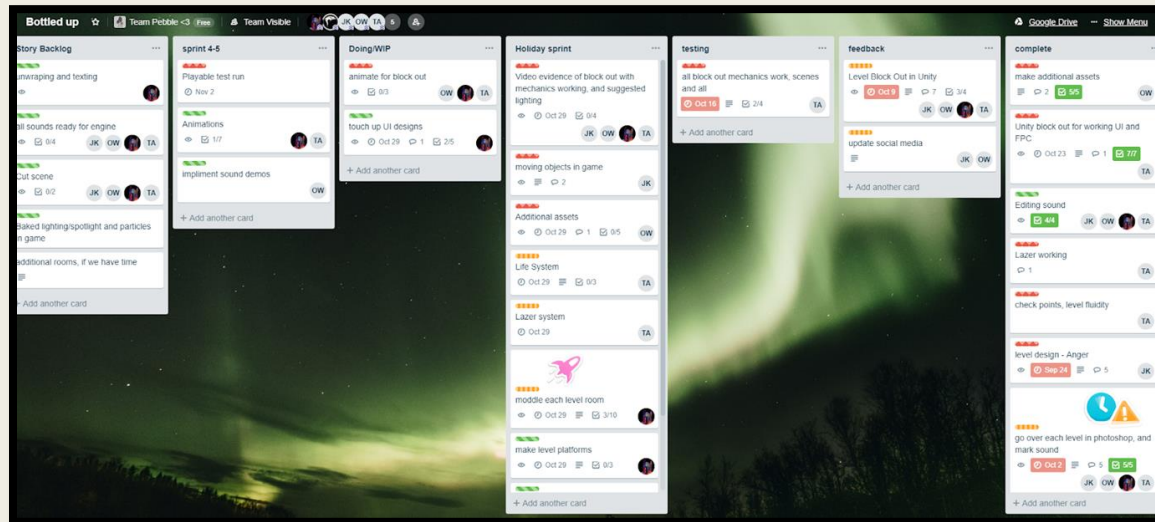


- This sprint was focused around starting our block outs, and creating our sound design maps. We also started to create assets, which proved to be too early on, as it meant I was behind in my block out to begin with.

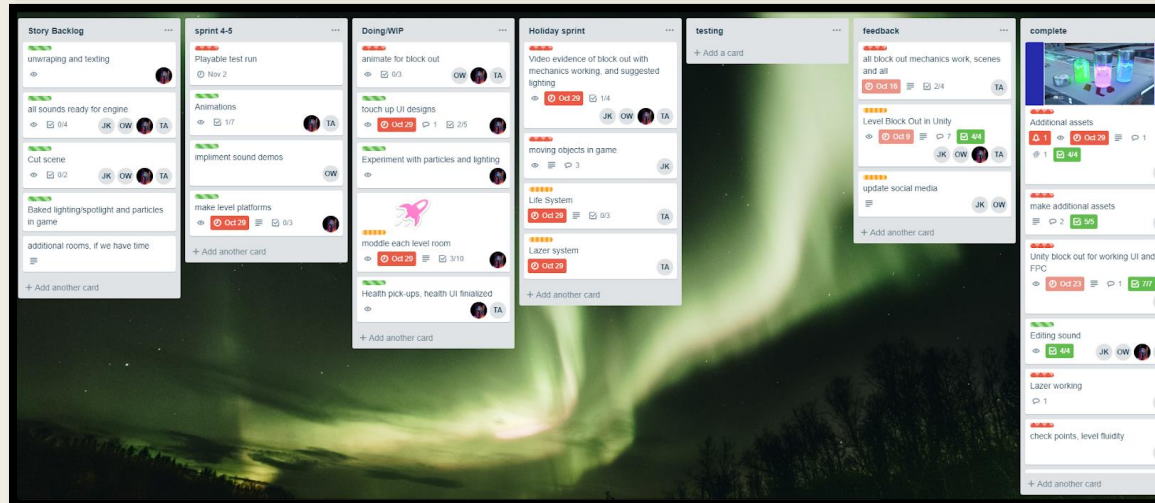


- This sprint mainly consisted of block out completion. It took me longer than others, because I carried on with asset design rather than finishing my Unity block out. Next time, I will prioritise the mechanics working first.

Trello Sprint Reviews (P8)

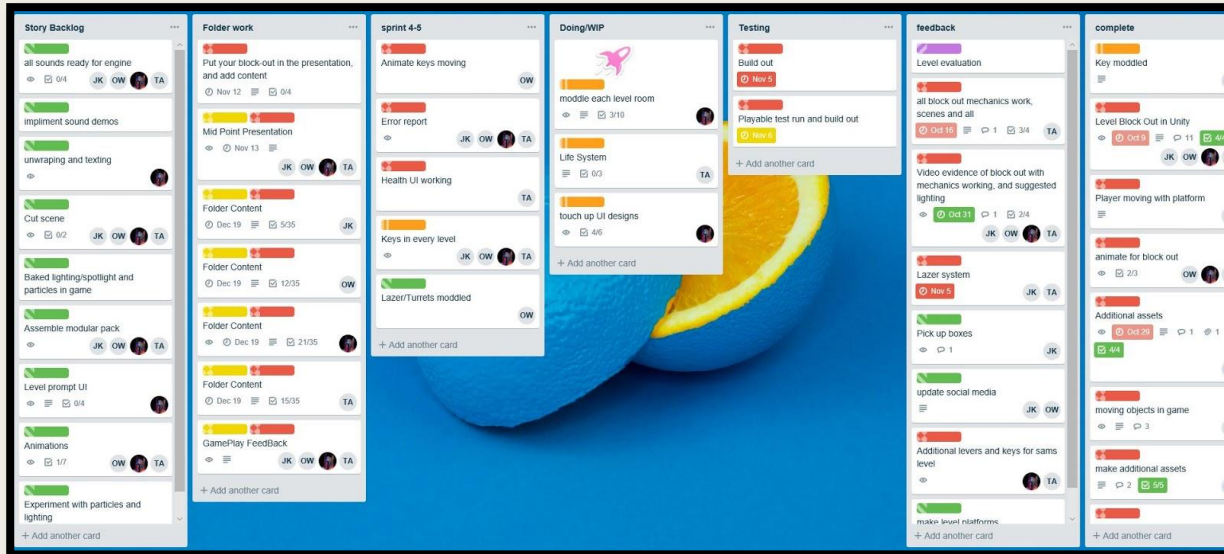


- During this sprint, I returned to modelling additional assets, as I had mainly completed my block out. We also began to ready our sounds for the game engine.

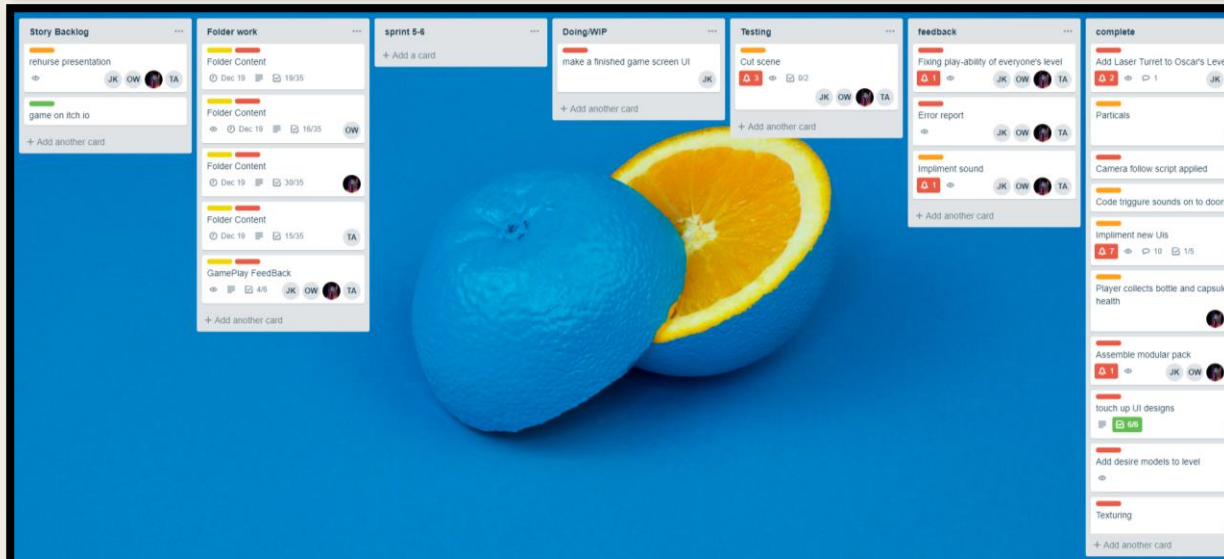


- This sprint involved us demonstrating our levels through video walkthroughs. We later used these videos in our mid-point presentations to the class.

Trello Sprint Reviews (P8)



- For me, this sprint was focused on modelling. I modelled more assets such as bottles, platforms, laser turrets and keys. We also started progressing through our folder work checklist.



- This is a screenshot of our final sprint. The last week is focused mainly on creating our end presentation, and also finishing any folder work.

Contingency (M4)

- Over the duration of the project, some aspects of the game have been changed; this is due to both time constraints, and issues with the modular pack. One change made, was the mechanic listed in my level. To add difficulty to my level, we planned to make the 7 desired objects spawn in different locations each time. However, because of time constraints, we could not do this, although it worked out by adding laser turrets and making the objects harder to collect.
- Another change I made was with my level layout. I kept my initial level layout until the first build out with the modular pack. Aside from separate corridors, I realised the modular pack didn't allow corners on walls, so I had to adapt my level layout, making it smaller and have less hallways. I did this as it was much easier than going back and changing the modular pack.



UNIT 12 P04

Oscar Webb



Modelled Assets: Desired Objects (P9)



Ring



Pocket Watch



Vase



Diamond



Cash

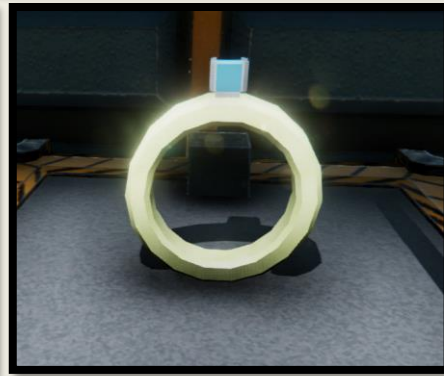


Gold Bar



Coins

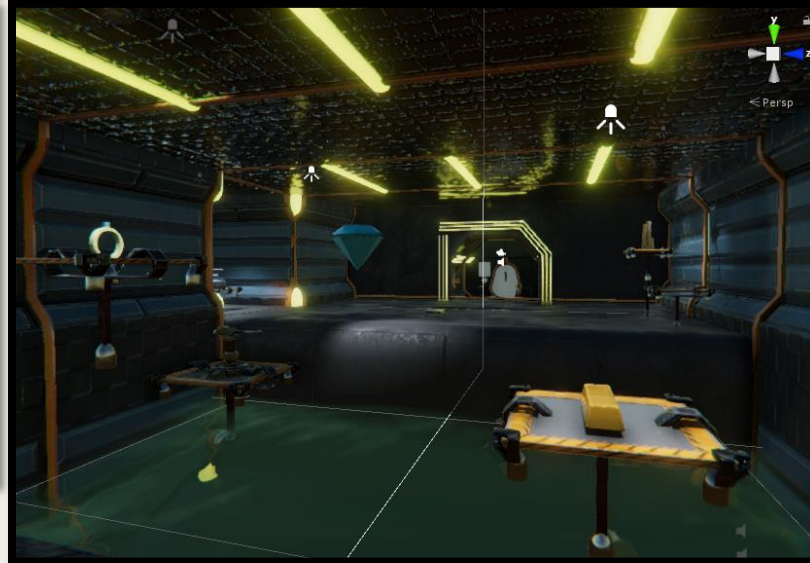
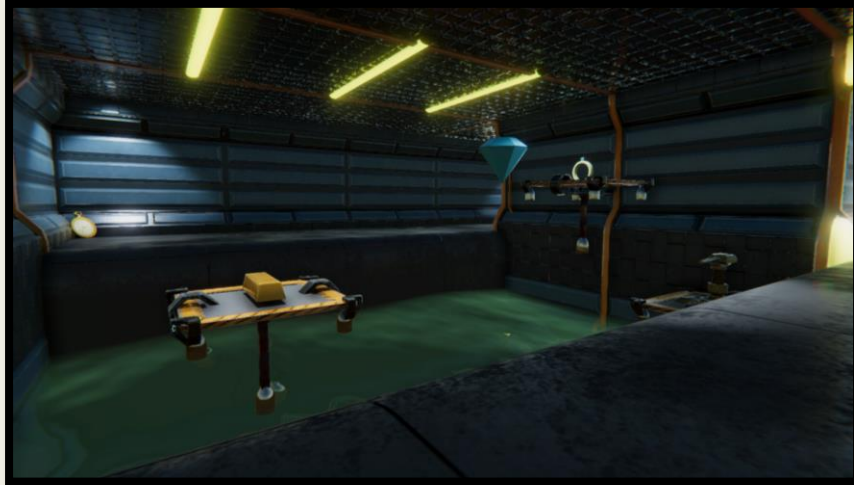
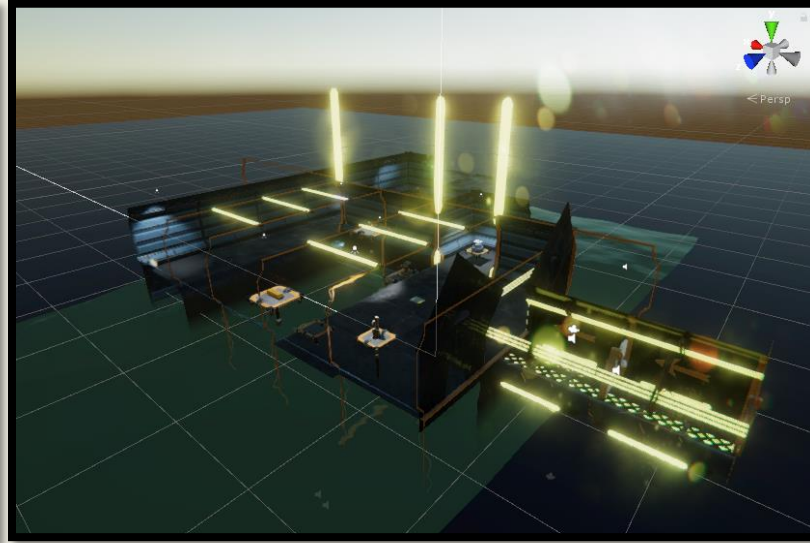
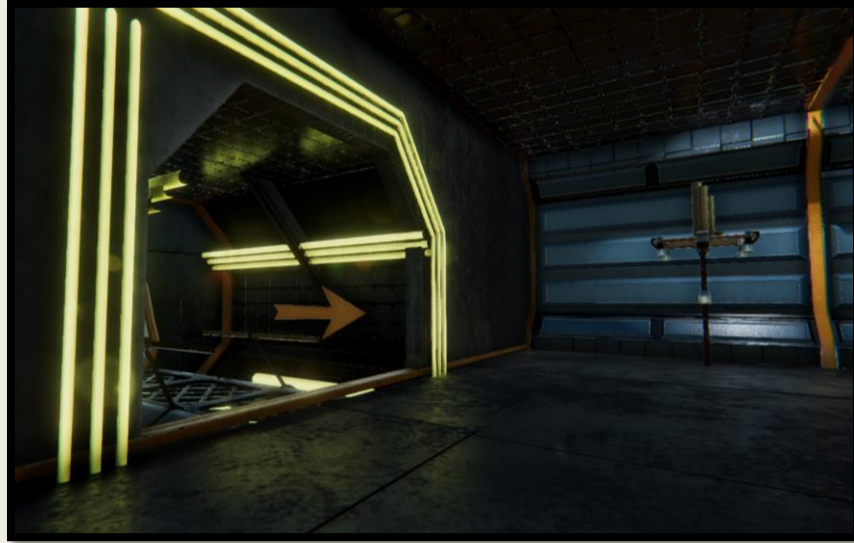
Implemented Assets (P10, P11)



Implemented Assets (P10, P11)



Implemented Modular Pack (P10, P11)



Error Reports (P12)

Mechanic	Problem	Resolution
Moving Platforms	The collider was not detecting the player and was being moved into a different place. Both the collider object and the platform has the animation attached.	Removed the separate collider and added it to the original platform.
Modular Assets	The player would walk up to an asset piece and get stuck, being unable to move.	Made the object's mesh collider impacting the player convex, with some objects having temporary box colliders.
Key Reader	At the end of the level, the key reader would not activate the door even when the player had collected the key.	Changed the raycasting interaction to activate when the object was tagged as a reader instead of named as one.
Laser	The laser object would not be avoidable as it was not always visible and would not collide or damage the player effectively.	Replaced the laser with a turret that would shoot at the player.
Laser Turret	The laser turret would not reach the player when they were in sight.	Extended the box collider on the turret.

Feedback

Class Feedback (From Presentations)

- Fix game lighting
- Possibly make it harder

-
- Nice level design (2D)
 - Nice consistent art direction
 - Horror? Tighter Corridors
 - No UI for pickups
 - Yellow on black hurts eyes
 - Nice Puzzle/Platforming
 - How do I know I have hit a checkpoint?

-
- Take out procedural spawnings. Help yourself, Oscar.
 - I'd like to have seen a level with all mechanics in place.
 - Shrink levels more.
 - SOLID! Well Done

-
- Still very open and very empty
 - Bit confusing
 - What's the objective?
 - No run part too long and boring
 - Any visuals yet?
 - Jack's level is pointless and doesn't make sense

- Add more visuals
- Nice animations :)
- No real threat
- Could improve lighting
- Health doesn't seem to affect you too much
- Levels quite empty
- End bit not needed

-
- Lack of detail in the game overview
 - Lots of detail in the game style overview
 - Lack of visuals in the level
 - Good lighting
 - Good UI Design

-
- No Textures

-
- Make Jack's level more useful

-
- Very nice art progress
 - Decent acknowledgement of past feedback
 - Player feedback improved
 - Unclear UI, White Text was a poor choice
 - Sam's level improved

The above list of feedback is taken from after each presentation we gave to the class. We have now responded to and thought about all of this feedback, and have made changes to our game, for example:

- The lighting has been fixed in our game, making it fluent throughout. We did this by adding the lighting capsules into the modular pack.
- We have added UI for the pickups in my level, and added checkpoints into other levels.
- One suggestion was to have random spawning of objects within my level, so it was harder; this was too much work for us to complete, as I had modelling to finish, so I instead added laser turrets and made the objects harder to collect.

Team Development Evaluation (P13, M5)

- This project has helped me understand what it means to work within a team when creating a game. Before this project, I hadn't worked collaboratively within game design before, and I have learned how important it is to be a practical and applied team member. At the start, in September, I was lacking with work slightly, as I assumed the workload would be less than that of a solo project; I was wrong, and realised I had to produce an equal amount of work to my other teammates, so I did so immediately. This is the key to working efficiently within a team, and next time I will do this from the start.
- While working collaboratively, Trello helped massively; the website allowed our team to manage our workload evenly, and let each of us know when we had work to complete. It also allowed us to tag team members on specific tasks, add due dates, and add importance labels. Unity Collab also helped greatly, as it allowed our team to work simultaneously on the same Unity project; we all worked on our individual scenes within the same project.
- Overall, I believe the collaborative work in our development team has been successful. We have communicated well, both at college and at home, and have learnt a multitude of skills in different areas from each other. To improve collaborative work in the future, I aim to start the project with a head start, and keep up to date with my other team members; this improves team communication and ensures all work is being completed to a high standard. It also gives us extra leeway to complete further work if necessary, towards the end. I would also carry on using both Trello and Unity Collab, despite some minor bugs and complications with Unity.

Final Game Analysis (M6)

- I am mostly pleased with my team's game as a whole. A lot of work has been put into our game, and I would say it has turned out as good as I initially expected. The only downfall of our game, in my opinion, is the incompleteness of our teammate Jack's level; the first three levels somewhat match in quality, and definitely in style, so when the fourth level loads it looks instantly incomplete and unfinished. Aside from this, the levels transition fluently with opening door animations, and I am highly pleased with the visuals, mechanics, and playability of our game.
- Compared to our initial game pitch, linked below, not much has changed. We have kept the same art and modular style throughout, as well as the story. Our game has stayed similar to the same competing games we originally listed, such as 'Fallout 4', 'Portal 2', and 'Star Wars: Battlefront II'. Bottled Up falls into the same genre as Portal 2, as a puzzle/adventure game, and also has a similar art style. Although the gameplay and mechanics are different to Fallout and Star Wars, our art style was heavily influenced by these games.
- I think our game is still most appealing to our original target demographic (aged 16-30, PC gamers); this is because the competitor games' fan bases we listed have this age range, and could be drawn to our game, due to its similar style. As well as this, the market for puzzle/adventure games is quite dry, potentially attracting Portal fans.

<https://wordpress.com/read/blogs/135491902/posts/116>

Final Game Analysis (D3)

- I am satisfied with the assets I have produced across this project, and feel as though I have improved my modelling skills highly. After the summer holidays and returning to college, I was slightly unfamiliar with 3DS Max; however, I now feel much more confident with the software again, and am excited to develop my skills further. With the UI design, I also feel more confident, and have learnt from mistakes, despite my designs not being used in the final game. Lastly, I am now familiar with sound editing and the Adobe Audition software we used to do so. With no prior knowledge, I am now able to successfully record sounds using a proper audio recording kit, develop and edit these sounds in Audition, and implement these into my game. These are other useful skills that will come in to play during my next project.

Skill Area	Current Level 1-5
<i>Modelling</i>	4
<i>UI Menu Design</i>	4
<i>Sound Design</i>	3

- I am excited to further develop my skills within modelling, UI design, and sound design over my FMP. I also aim to practice in these areas outside of college projects, to build my portfolio to a higher standard.



UNIT 11

Sound Design



Condenser and Dynamic Microphones

What's the difference?

Condenser Mic:

- Condenser mics are usually used indoors. This is because they are more sensitive than dynamic microphones, meaning it is ideal for studio recordings and podcasts.

Dynamic Mic:

- Dynamic mics are for outdoor use, as they have lower frequency; this means they don't pick up as much background noise. They are good for live performances and interviews.

Large and Small Diaphragm Condenser Mics

What's the difference?

Large Diaphragm:

- Large diaphragm condenser microphones make a sound source appear bigger and more engaging.

Small Diaphragm:

- Small diaphragm condenser microphones have very good transient response, extended high frequency response, and a very consistent pickup pattern.

Microphone Polar Patterns

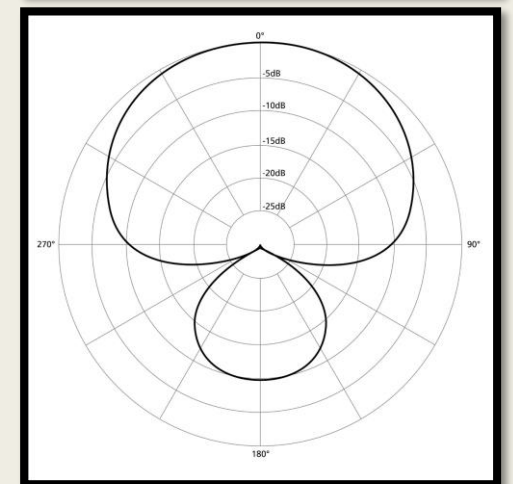
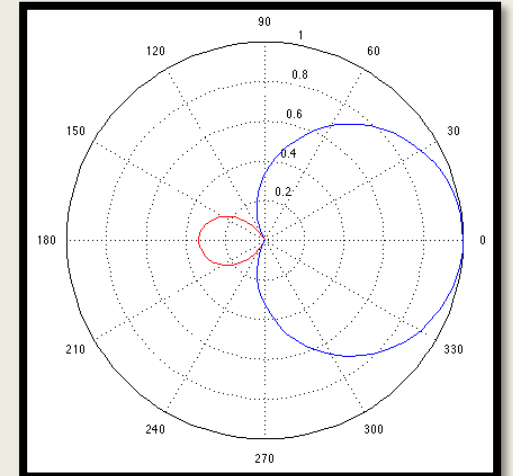
How do they affect sound recording?

Cardioid Mic:

- Cardioid microphones have most sensitivity at the front and least at the back, making them ideal for loud stages. This gets rid of unwanted background noise, as they are much more resistant to feedback than omnidirectional mics.

Super Cardioid Mic:

- Super cardioid microphones offer a much more narrow pickup compared to cardioids, as well as a greater rejection of ambient sound. These microphones are most suitable for when a single sound source needs to be picked up in a loud environment, as they are very resistant to feedback.



Microphone Polar Patterns

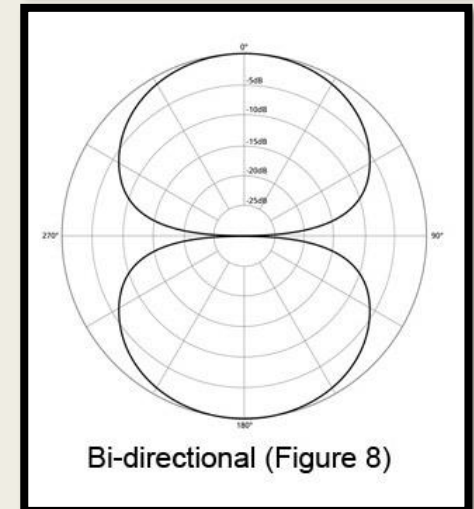
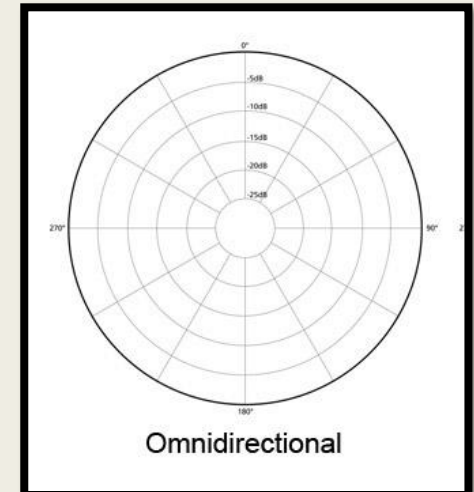
How do they affect sound recording?

Omni-Directional Mic:

- Omnidirectional microphones have equal sensitivity on all angles, meaning they pick up sound evenly from each direction, meaning the microphone must not be aimed in a certain direction. A disadvantage is that omnidirectional mics cannot be aimed away from undesired sources.

Figure 8 Mic:

- A microphone with a figure of eight polar pattern picks up sound from a frontal source, and from the rear, but not from the side. Microphones with this figure of eight polar pattern are usually large diaphragm microphones.



Microphone Polar Patterns

How do they affect sound recording?

Shotgun Mic:

- The polar pattern of shotgun microphones is the most highly directional. It has a very narrow lobe in the frontal direction, rejecting sounds from other directions. They have small lobes of sensitivity on the sides and rear, but are much less sensitive than other directional mics.
- Due to the narrowness of their sensitivity area, shotgun mics are commonly used on TV and film sets, stadiums, and in the wilderness.



Types of Sound

And their use in sound design for games.

Ambient Sound:

- Ambient sounds are used in the background; for example, wind, water, birds, etc. Ambient sounds create a sense of immersion and realism, by providing audio between scenes, preventing an unnatural silence.
- Different types of ambient sounds include:
- Matching Ambient Sound- recordings that match the ambience of a scene.
- Wild Sound- background noise with a distinct sound.
- Buzz Track- A term for ambient sound.
- Room Tone- The sound of an empty room.

Types of Sound

And their use in sound design for games.

Directional Sound:

- Directional sound is sound technology aimed in a particular directional field, rather than spreading naturally. Directional sound can be used in many different ways in home entertainment.
- Some directional sound technologies are created using speaker arrays; these technologies are commonly used for acoustics in public installations, such as performance venues.
- Another common application of directional sound is within the development of sonic and ultrasonic weapons. Sonic weapons have been part of sci-fi culture for decades, but are now being developed inside the military. The idea behind them is to create a focused beam of sound using directional sound, which will affect the human body in some way.
- In games, directional sound can be used to create more focused and specific forms of audio. For example, it can be used to signify interaction with an object or asset. Additionally, it can be used in a spatial way within 3D environments.

Types of Sound

And their use in sound design for games.

Vocal Sound:

- Definition of vocal sound: A sound produced by vertebrates by means of lungs, larynx, or syrinx.
- Within games, vocal sound is used to communicate narrative and human emotion between characters. It can also be used to create unnatural sounds or sound effects, such as machines, animals, monsters, etc.
- Vocal sound is usually mixed and balanced to be more projecting during cinematics and cutscenes.

Game Audio

Sound Recording: Microphones:



- These attachments were used to muffle the sound slightly, so the microphone does not pick up as much background noise.

- This cross-polar microphone was used to record the directional and spatial sounds.
- This microphone is used for vocal sound; we used this to record the heavy breathing sound effects.



Game Audio

Sound Recording: Environments:

- Here are some examples of environments I used to record my sounds in. I chose to record in these areas as they are small, quiet, and have low levels of echo, reducing the level of background noise in my sounds. However, some sounds required me to travel around the college, which brought problems. For example, while recording my door sound, students kept coming through the door.

Atmospheric Humming



Lift

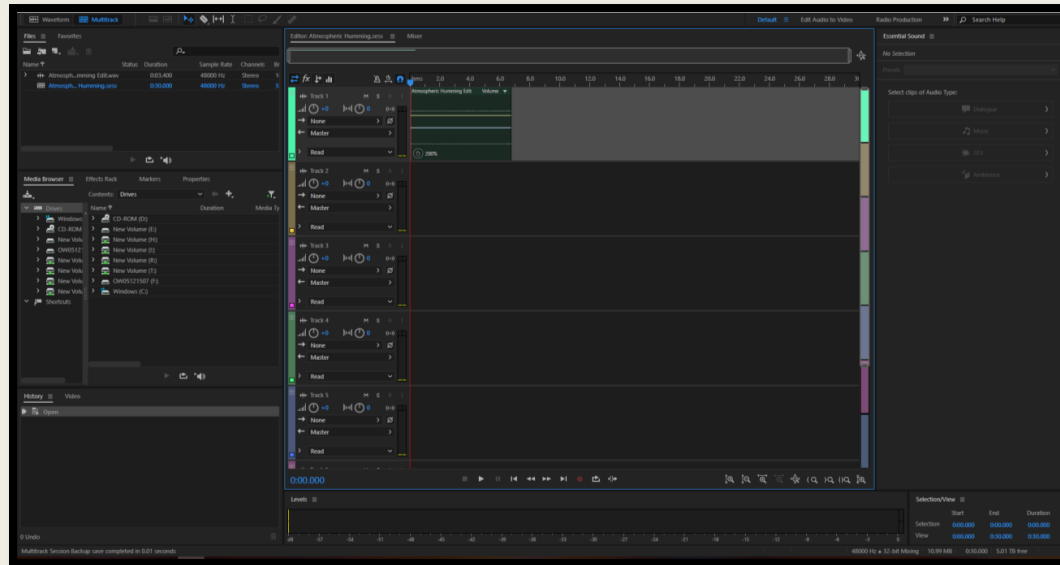


Atmospheric Wind

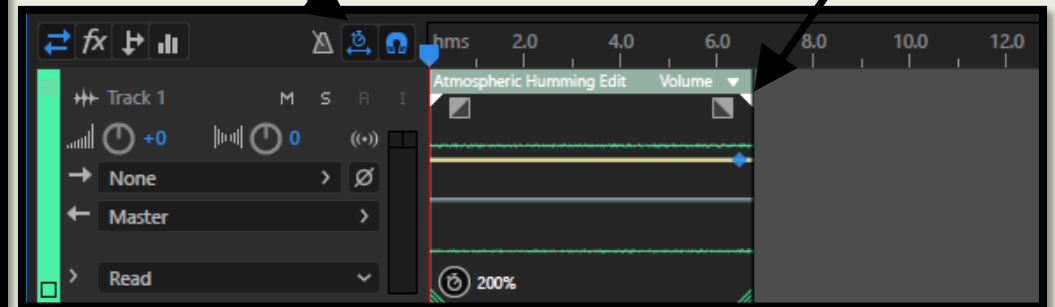


Game Audio

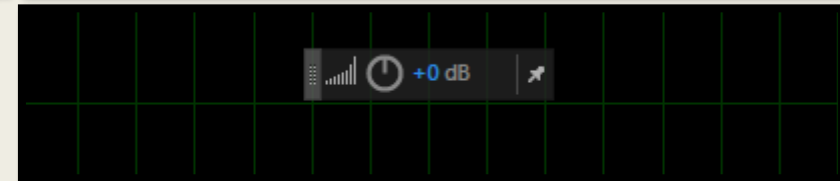
Multitrack Sessions:



- I also increased the duration of the sound file by 200%, making it twice as long. This slowed down the sound making it deeper in pitch, and sound more eerie.



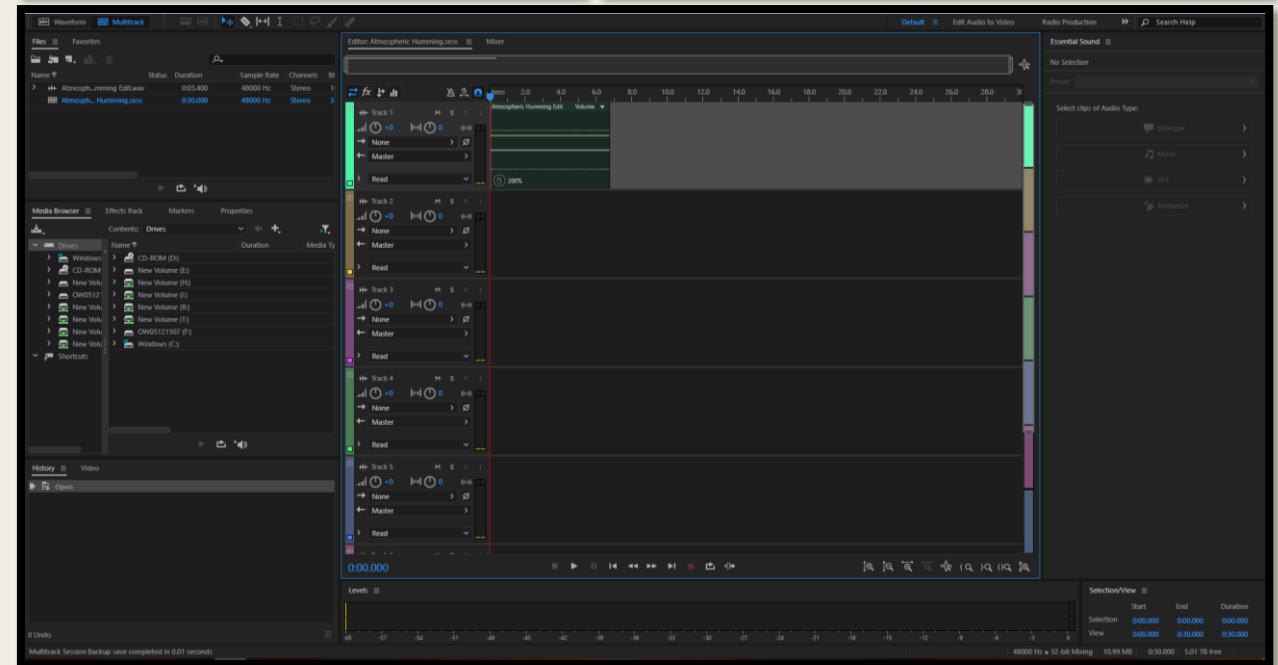
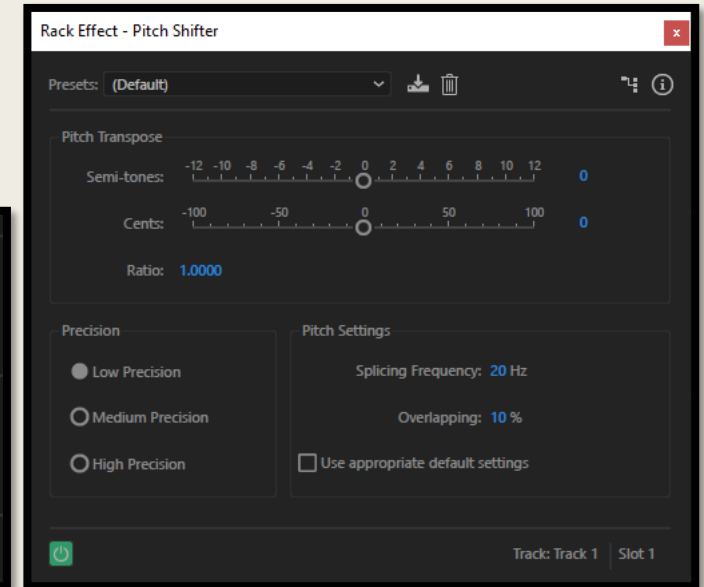
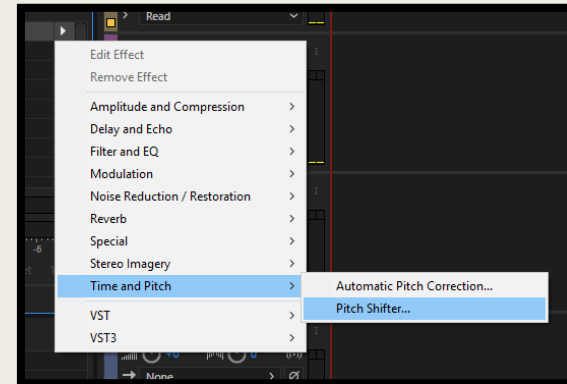
- This is the multitrack session for my atmospheric sound; in this, I lowered the sound levels to prevent it from coming distorted, by dragging the dB lower.



Game Audio

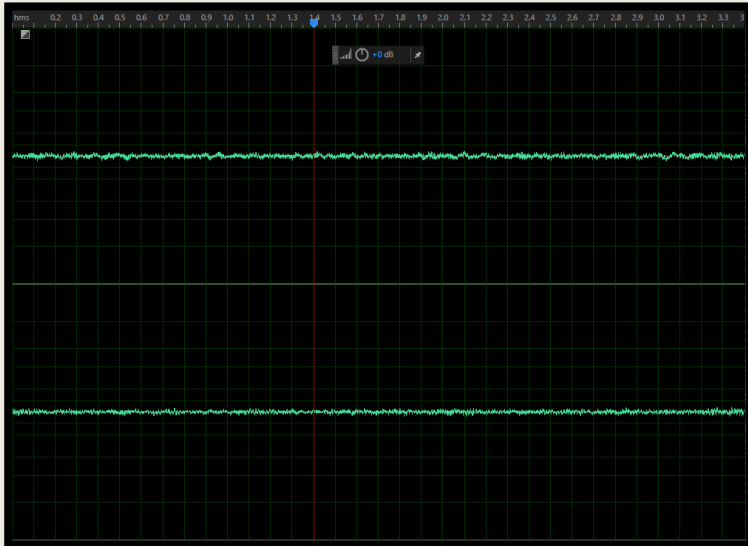
Multitrack Sessions:

- Another way I increased the pitch with the atmospheric humming sound, was by applying the pitch shifter effect. I then created a much deeper pitch by dragging the semi-tone down. This was more efficient than increasing the track duration, as it lowered the pitch but kept the duration the same.

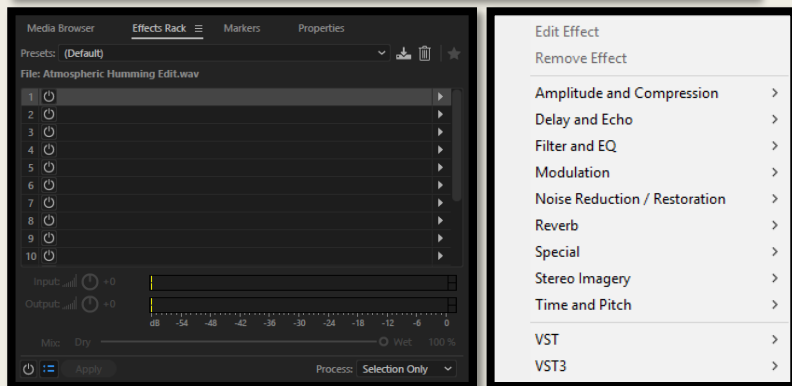


Game Audio

Multitrack Sessions:



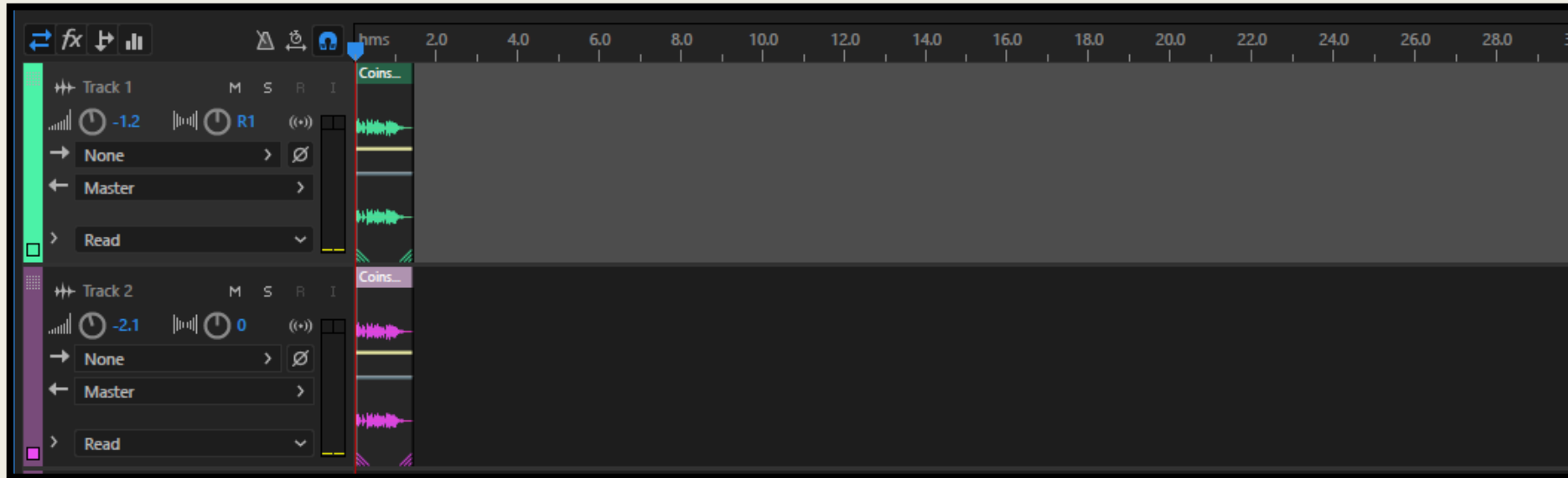
- Here, the sound file can be directly edited. You can cut parts of the file out, and bring the levels up and down at specific points. I did this to make the tracks seamless, so they can loop.



- Effects can also be added to sound files to change the overall sound, or adding slight alterations such as chorus, echo, etc.

Game Audio

Multitrack Sessions:

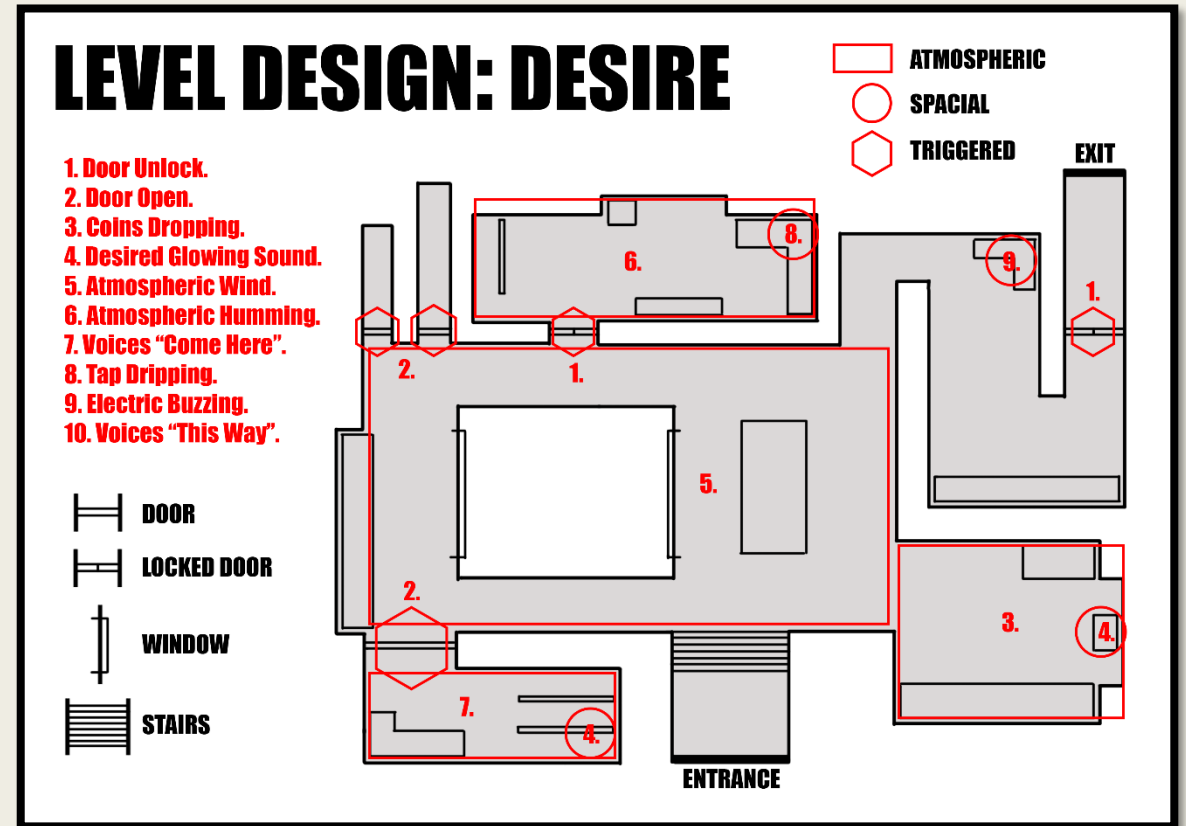


- Here, I created the dropping coins sound effect. It was simple to record, as I only had to drop coins onto a hard floor. I edited this by duplicating the sound file and offset it very slightly, making it sound as if there are more coins dropping.

Game Audio

Audio Plan:

- To the right is my level plan; it shows the basic layout and placement of assets for my level, but also displays the appointment of each sound. The plan lists all ten sounds I initially expected to record, however as I progressed through recording the sounds, I changed some of them, which is evident on the next slide.



Game Audio

Edited Recordings and File Types:

- I have saved my ten edited recordings as both .wav files and .ogg files. The .wav files are lossless, meaning no quality is lost; I used this file type to upload my tracks to SoundCloud. I imported the .ogg file format sounds into my game, as it compresses the file size.

- <https://soundcloud.com/user-428144390/sets/edited-level-sounds>

- <https://soundcloud.com/user-428144390/sets/raw-audio-recording>

- The top link leads to my edited sounds; the bottom link is my raw recordings.

.ogg Files

Name	Date modified	Type	Size
Atmospheric Humming Edit	29/11/2018 12:18	VLC media file	190 KB
Atmospheric Humming Edit	29/11/2018 12:18	Adobe Prelude XMP	8 KB
Banging Thud Edit	29/11/2018 12:19	VLC media file	57 KB
Banging Thud Edit	29/11/2018 12:19	Adobe Prelude XMP	8 KB
Coins Dropping Edit	29/11/2018 12:19	VLC media file	99 KB
Coins Dropping Edit	29/11/2018 12:19	Adobe Prelude XMP	9 KB
Door Edit	29/11/2018 12:19	VLC media file	205 KB
Door Edit	29/11/2018 12:19	Adobe Prelude XMP	8 KB
Dripping Water Edit	29/11/2018 12:20	VLC media file	285 KB
Dripping Water Edit	29/11/2018 12:20	Adobe Prelude XMP	8 KB
Footsteps Edit	29/11/2018 12:20	VLC media file	526 KB
Footsteps Edit	29/11/2018 12:20	Adobe Prelude XMP	8 KB
Heaving Breathing Edit	29/11/2018 12:20	VLC media file	245 KB
Heaving Breathing Edit	29/11/2018 12:20	Adobe Prelude XMP	8 KB
Heavy Breathing 2 Edit	29/11/2018 12:20	VLC media file	362 KB
Heavy Breathing 2 Edit	29/11/2018 12:20	Adobe Prelude XMP	9 KB
Running Water Edit	29/11/2018 12:20	VLC media file	708 KB
Running Water Edit	29/11/2018 12:20	Adobe Prelude XMP	11 KB
Wind Atmospheric Edit	29/11/2018 12:21	VLC media file	222 KB
Wind Atmospheric Edit	29/11/2018 12:21	Adobe Prelude XMP	8 KB

Oscar Edited Level Sounds 8 minutes ago Private

0:00 0:04

- 6 Footsteps Edit.WAV
- 7 Heaving Breathing Edit.WAV
- 8 Heavy Breathing 2 Edit
- 9 Running Water Edit.WAV
- 10 Wind Atmospheric Edit.WAV

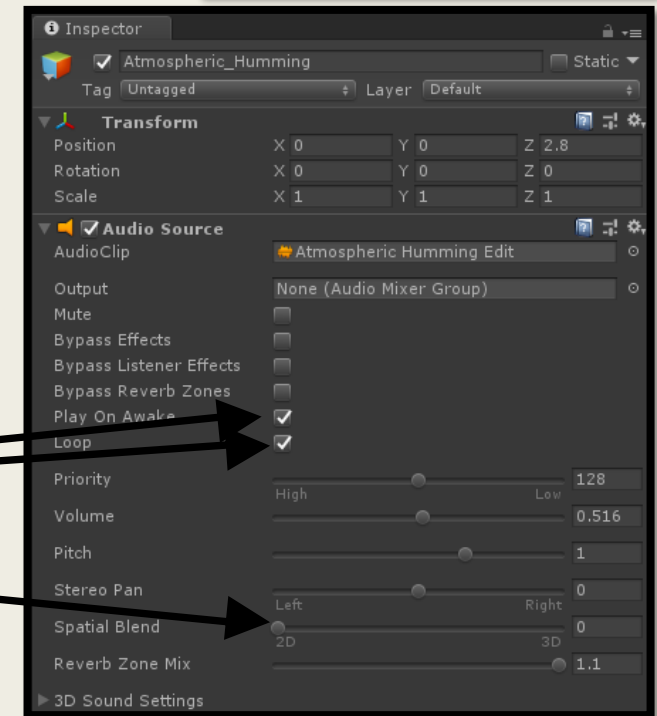
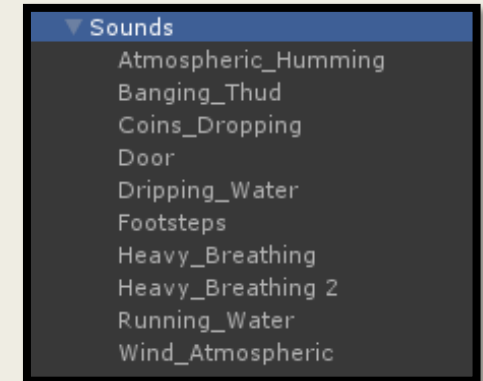
View 10 tracks

Share More

Game Audio

Unity Implementation (Importing and Ambient):

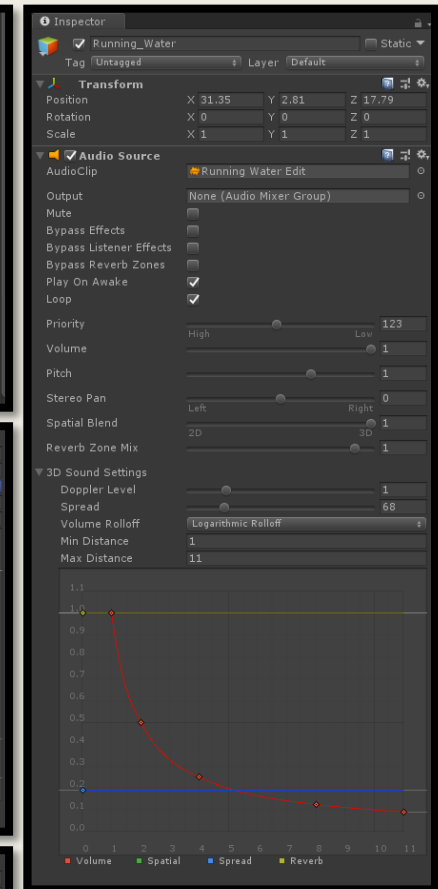
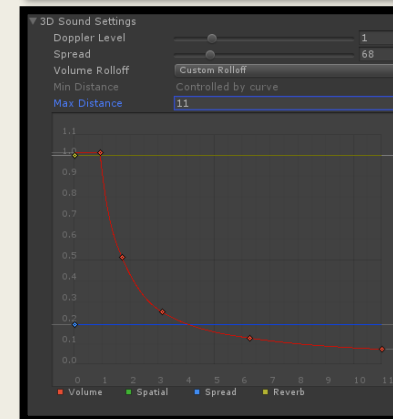
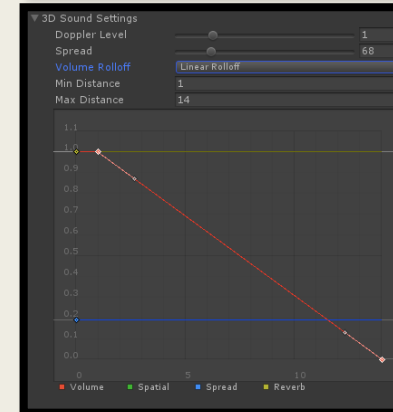
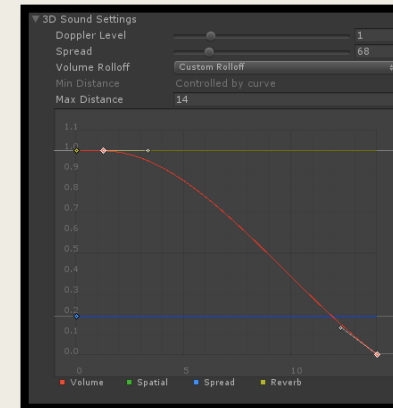
- Firstly, I imported my .ogg sound files into my level's assets. I then created an empty object called 'sounds', which I created audio sources within, attaching my sound files into the 'AudioClip' section on each.
- With my atmospheric sounds, they were simpler. I just clicked loop, play on awake, and left them as a 2D sound, meaning they stay the same volume as you move around the level.



Game Audio

Unity Implementation (Spatial):

- For my spatial sounds, it wasn't very different to atmospheric; again, I clicked loop, play on awake, but turned the spatial blend up to 3D. This is what changes the sound to be spatial, rather than atmospheric.
- I then increased the spread, min distance and max distance. This essentially changed the size of the spatial sounds. This allowed me to fit the sound into the correct size of my death drop.
- I also customised the fall-off, which is the distance of the sound travelling away from the source. This made the spatial effects sound more believable in-game.



Game Audio

Unity Implementation (Trigger):

- I made my trigger sounds by bringing in the scripts from the test level. I added the 'AudioPlay' script onto my collectible items, then dragged the sound effect into the 'Audio Source' section. Then all I had to do was add the 'PlayerTrigger' script onto the player.

AudioPlayScript

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class audioPlay : MonoBehaviour {
6
7     public Transform audioSource;
8
9     public AudioSource source;
10
11     // Use this for initialization
12     void Start () {
13         source = audioSource.GetComponent<AudioSource>();
14         source.Stop();
15     }
16
17     // Update is called once per frame
18     void Update () {
19
20     }
21 }
```

PlayerTriggerScript

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class playerTriggers : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14
15    }
16
17    void OnTriggerEnter (Collider other) {
18        if (other.tag == "") {
19            other.GetComponent<audioPlay>().source.Play();
20        }
21    }
22 }
```


Game Audio

Unity Implementation (Audio Mixer):

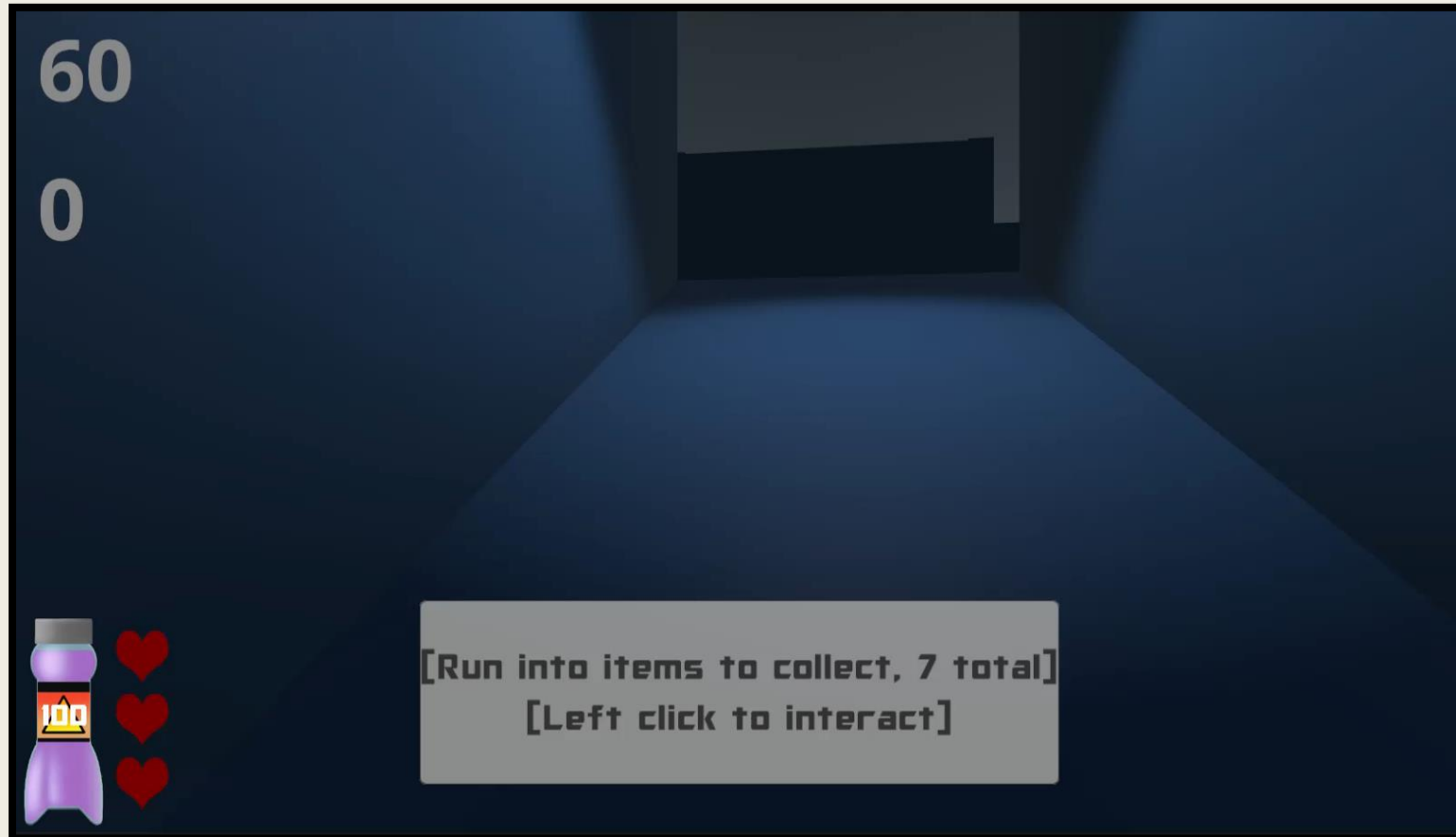
- I created an audio mixer within my scene. This allowed me to properly mix and adjust the volume of each sound effect within my level. By doing this, I could easily fix sounds being overpowered by others, make the overall sound quality of the game better, and create an accurate depiction of my game environment.



Game Audio

Unity Implementation (Videos): <https://youtu.be/aXirOWYpUWY>

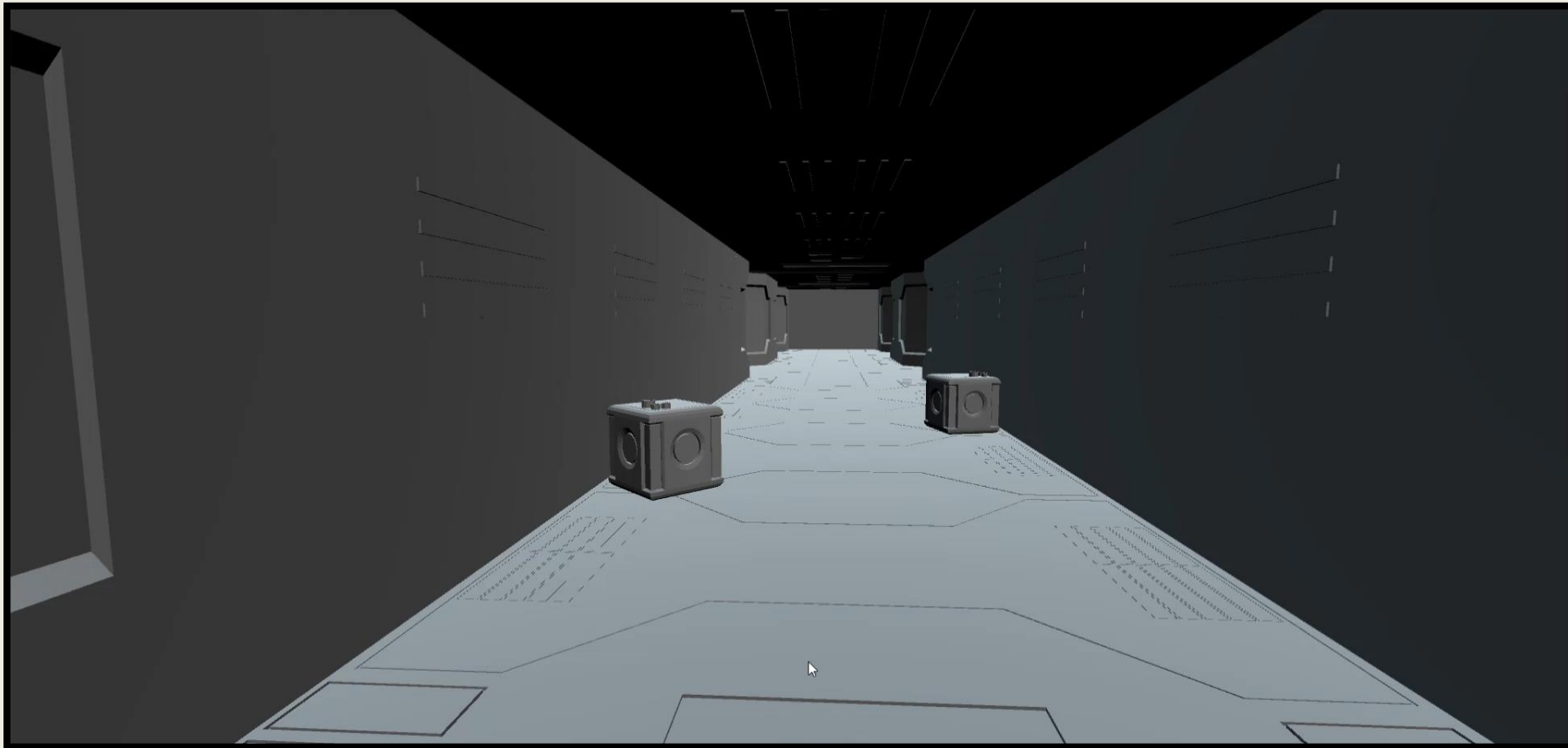
- This video displays the sounds implemented into my own level:



Game Audio

Unity Implementation (Videos): <https://youtu.be/AU2uuB8sI90>

- This video displays the sounds implemented into my test level:



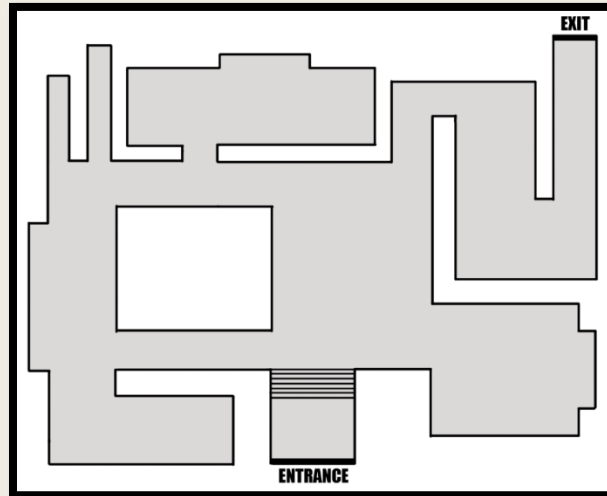


UNIT 9 P01

Oscar Webb

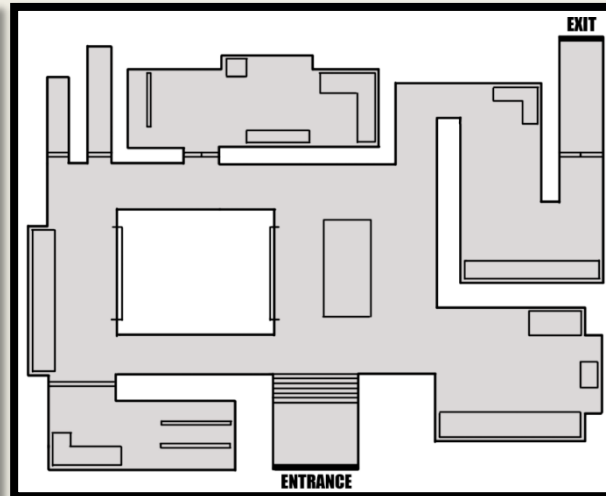


Level Design (P1)



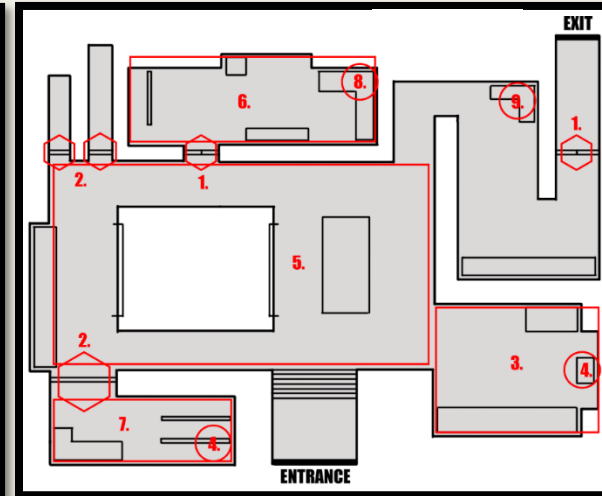
Primary Layout

- The primary layout is the basic shape of the level; this includes the entrance and exit to the level, each room, and stairs.



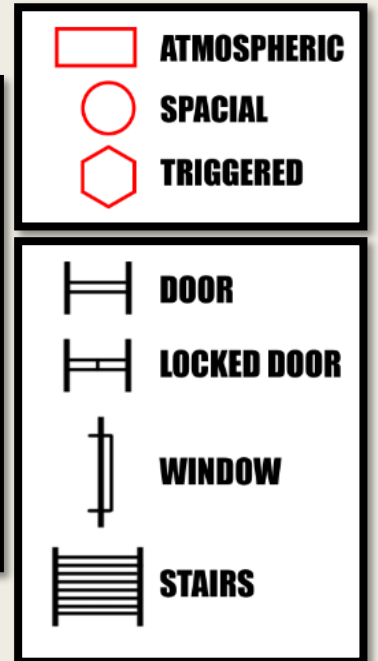
Secondary Layout

- The secondary layout adds assets such as doors, windows, tables, shelves, etc.

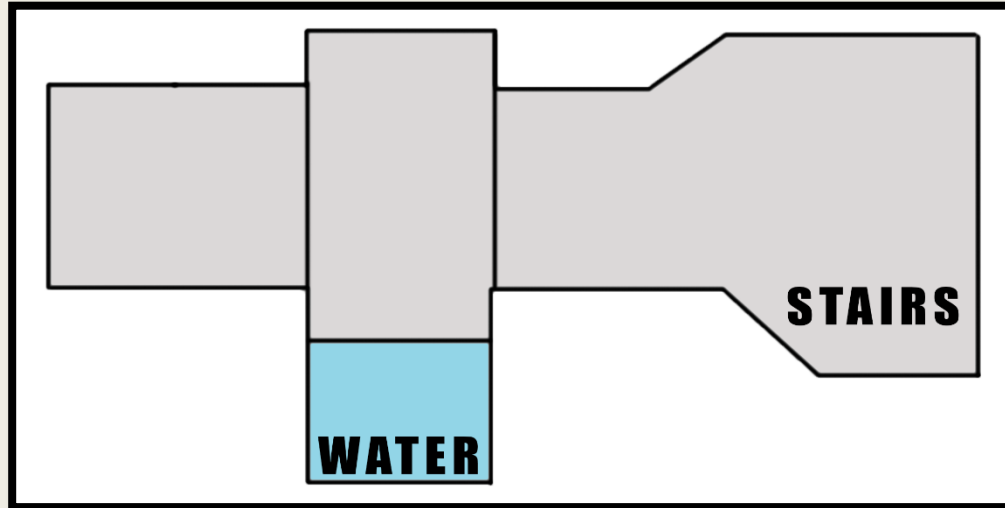


Sound Layout

- The sound layout displays each sound that will be in the level. The rectangles are atmospheric, circles are spacial, and hexagon is triggered.

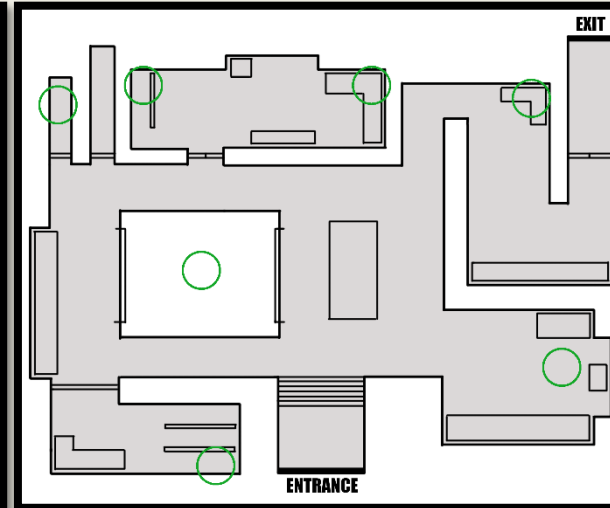


Level Design (P2, M1, D1)



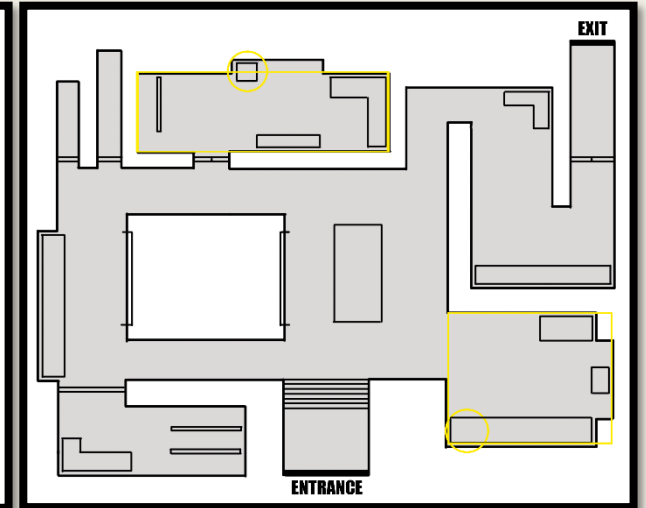
Side View Layout

- The side view displays different height variations, although there aren't many throughout my level.



Pick Ups Layout

- This layout shows the location of each pick up asset.



Turret AI Layout

- This layout shows the location of every turret, and their detection zones (box colliders).



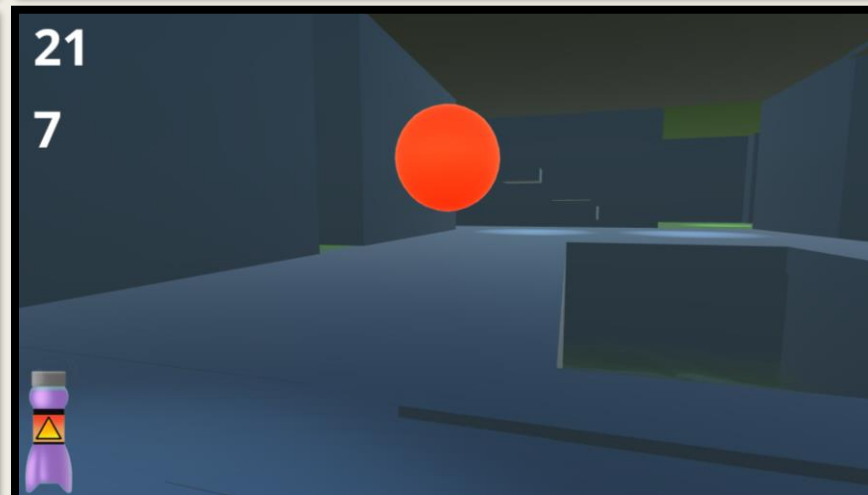
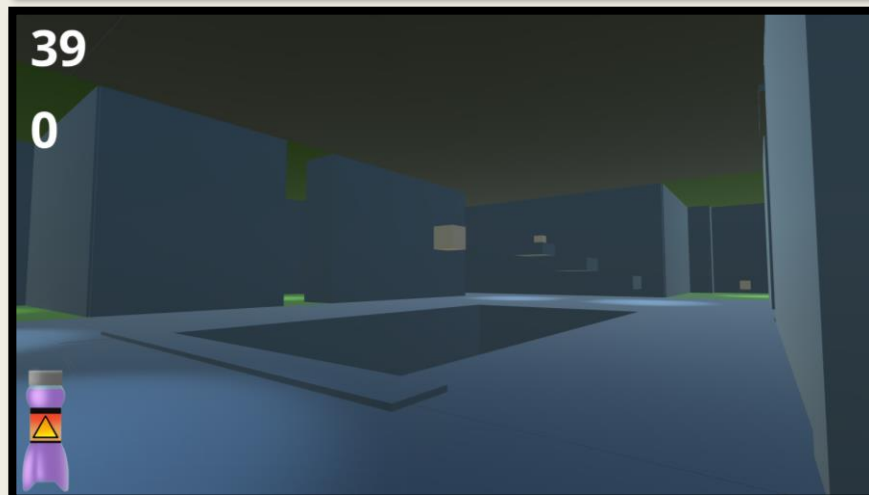
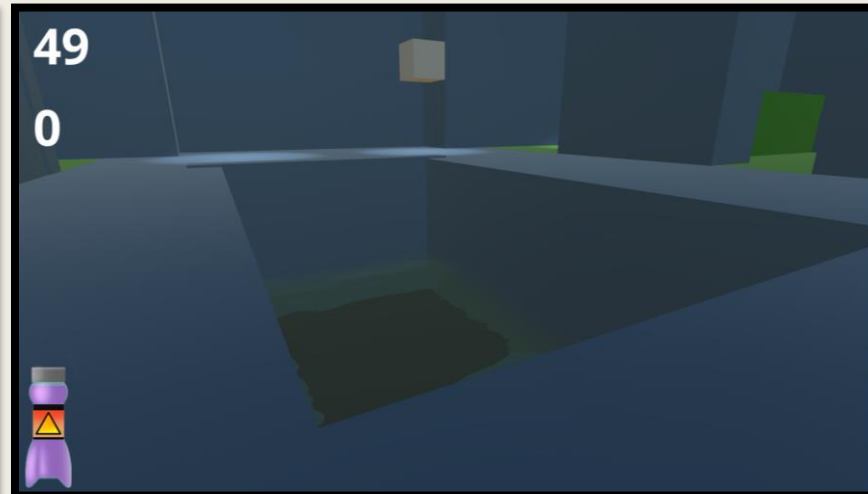
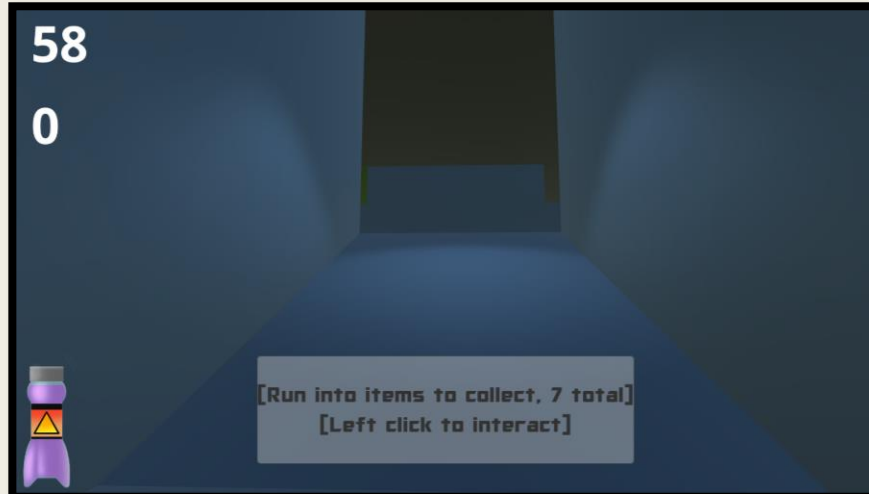
UNIT 9 P02

Oscar Webb



Unity Block Out (P4, P5)

Screenshots



Unity Block Out

Mechanics

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CollectItems : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9     }
10
11
12     // Update is called once per frame
13     void Update () {
14
15
16     private void OnTriggerEnter(Collider other)
17     {
18         if(other.name == "Player")
19         {
20             Destroy(gameObject);
21             ScoreScript.ItemNo += 1;
22         }
23
24         //if(ScoreScript.ItemNo == 5)
25         //{
26             // CountdownScript.currentTime = CountdownScript.currentTime;
27         //}
28
29 }
```

collectItemsScript

The collectItemsScript script allows the player to pick up items. If the player walks over the objects, then they disappear. The scoreScript makes the ItemNo value increase by one each time an object is increased.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class ScoreScript : MonoBehaviour {
7     public static int ItemNo;
8     public Text scoreItemText;
9
10
11     // Use this for initialization
12     void Start () {
13         ItemNo = 0;
14     }
15
16     // Update is called once per frame
17     void Update () {
18         scoreItemText.text = ItemNo.ToString();
19     }
20 }
```

scoreScript

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.SceneManagement;
6
7 public class CountdownScript : MonoBehaviour
8 {
9     static public float currentTime = 0f;
10    float startingTime = 60f;
11
12    [SerializeField] Text CountdownText;
13
14    private void Start()
15    {
16        currentTime = startingTime;
17    }
18
19    private void Update()
20    {
21        if(ScoreScript.ItemNo < 7)
22        {
23            countdown();
24        }
25
26        if (currentTime <= 0)
27        {
28            currentTime = 0;
29            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
30        }
31    }
32
33
34
35    private void countdown()
36    {
37        currentTime -= 1 * Time.deltaTime;
38        CountdownText.text = currentTime.ToString("0");
39    }
40
41 }
```

countdownScript

The countdownScript creates a counting-down timer which lasts one minute. It also stops the timer if all seven items are collected before the timer runs out. If the timer runs out, then the level will restart.

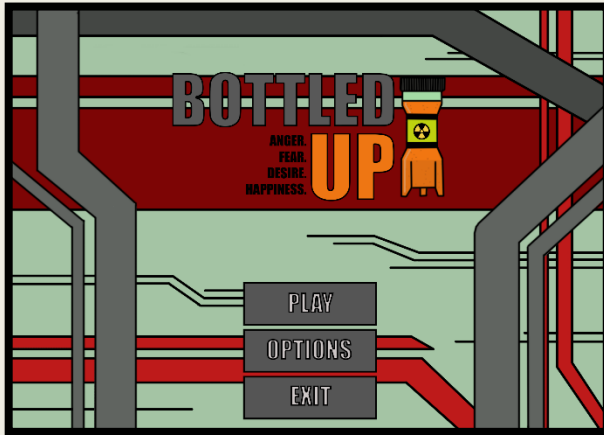
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FallScript : MonoBehaviour {
6     public Transform checkpoint;
7     public GameObject player;
8
9     // Use this for initialization
10    void Start () {
11
12
13
14    // Update is called once per frame
15    void Update () {
16
17
18    public void OnTriggerEnter(Collider other){
19        if (other.name == "FallPoint") {
20            HP_Script.HP = HP_Script.HP - 10;
21            player.transform.position = new Vector3(checkpoint.position.x,checkpoint.position.y,checkpoint.position.z);
22        }
23
24    }
25 }
```

fallScript

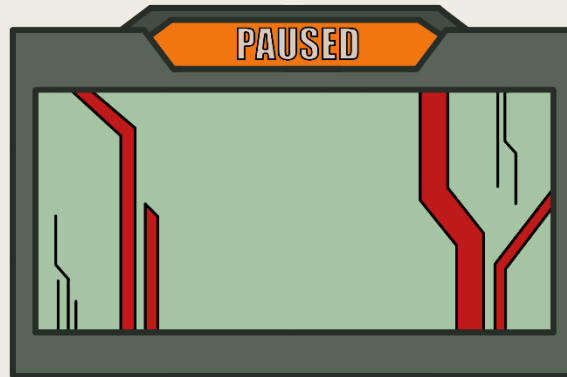
The fall script kills the player when they fall into the death pit. It restarts the level, and transforms the player back to the start of the scene.

Initial UI Designs

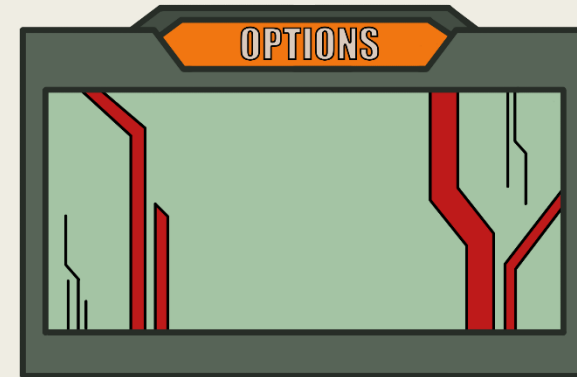
Main Menu



Pause Menu



Options Menu

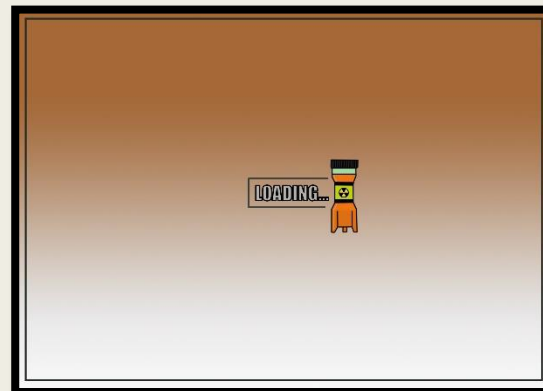


Both the pause and options menus appear on top of the paused game, which is blurred out in the background. I also created a button template which is to be applied with text in unity.

Button



Loading Screen

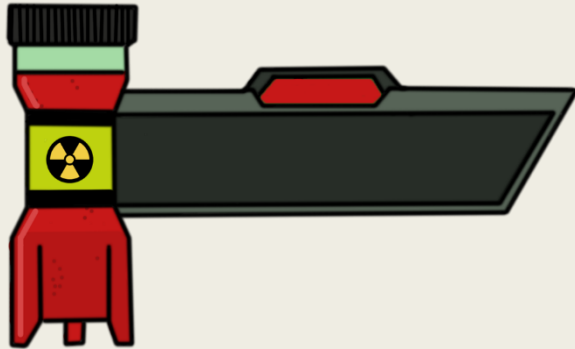


Death Screen

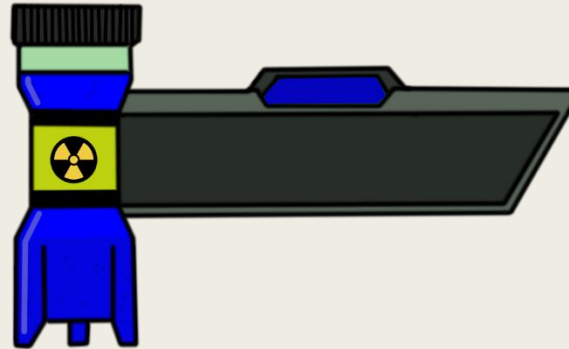


Initial UI Designs

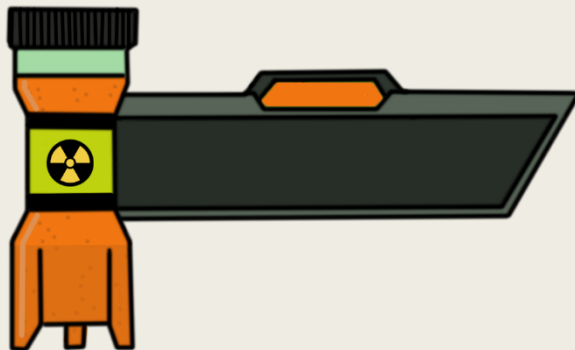
Health Bar Anger



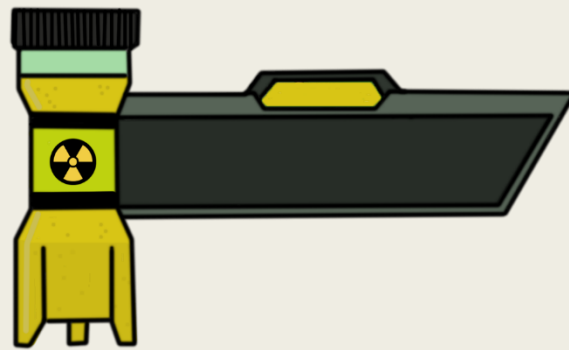
Health Bar Fear



Health Bar Desire

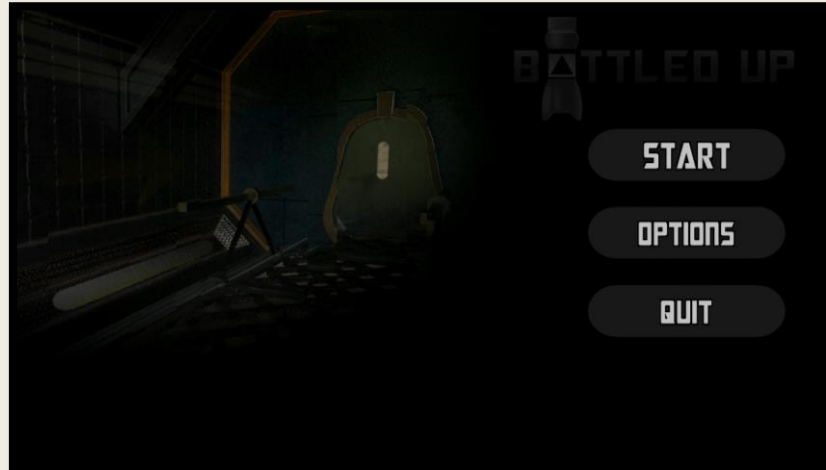


Health Bar Happiness



Updated UI Designs

Main Menu



Pause Menu



Options Menu



Health Bars



Button



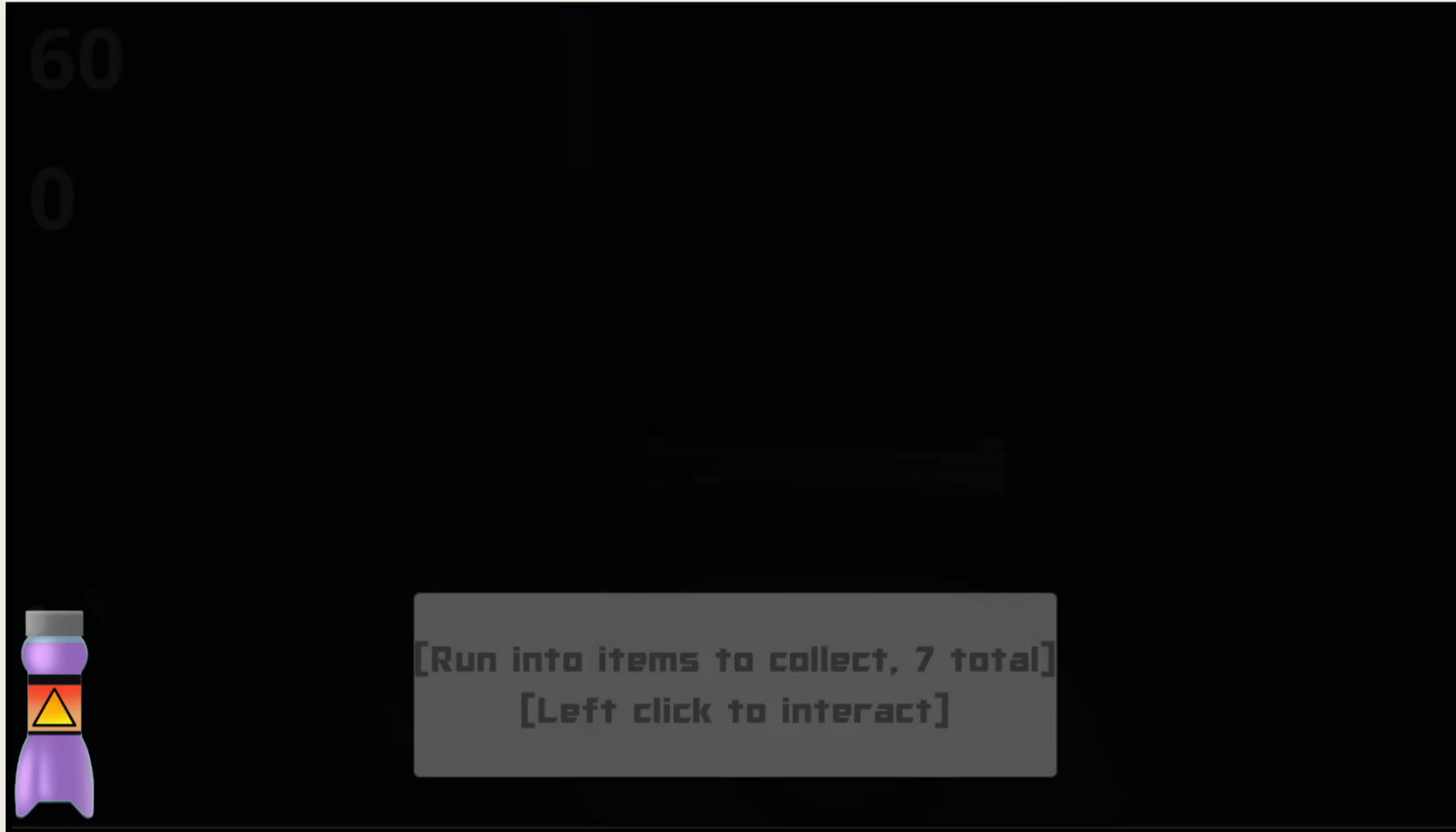
Death Screen



These are the new UI designs implemented into our game. We felt these designs fit our game's style and theme better than the initial ones.

Unity Block Out (P4, P5)

Video <https://youtu.be/aXirOWYpUWY>



Unity Build Out (P5, P6, M2, M3)

Video <https://youtu.be/s4eCnynWI3U>

