SEARCH FOR A STAR

Their Support

Design and Development Documentation

Nathan Cooper

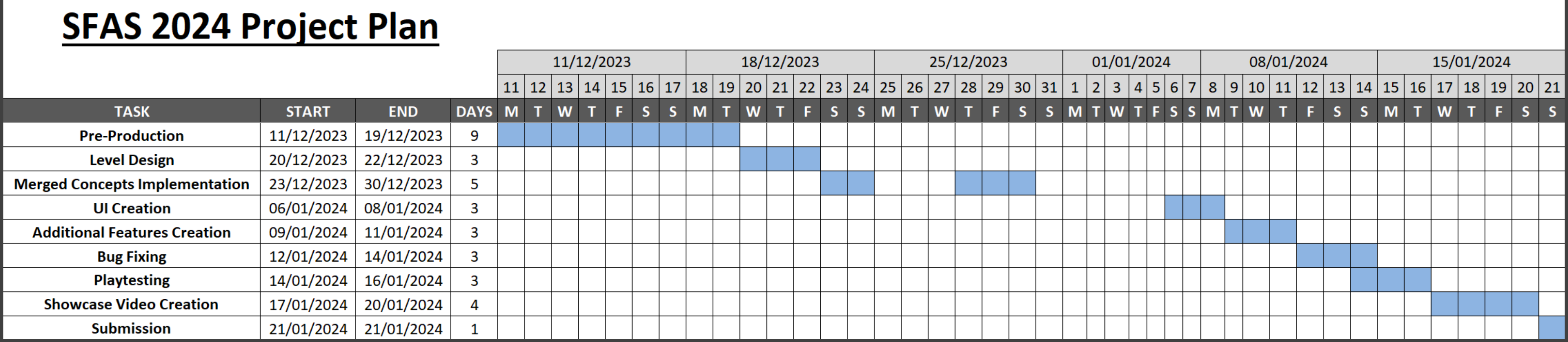
Contents

Project Plan

Gantt Chart

The Pre-Production stage made it easy to create my project as I had set out the foundations for what I needed. For example, I conceptualised ideas for the prototype, constructed a Level Blockout, formulated which mechanics, features and puzzles I would include, how I would create them, the length of the prototype and what enemies I would include. Additionally, leaving time near the end for any major bug fixing and playtesting. As these are essential for the development of any project.

I decided to create a Gantt Chart for my project plan. I chose this method because I believe this to provide the best visualisation for the production schedule. As I can clearly see how long each milestone will take me and inform me if I am behind or ahead of my schedule. Below is my Gantt Chart.



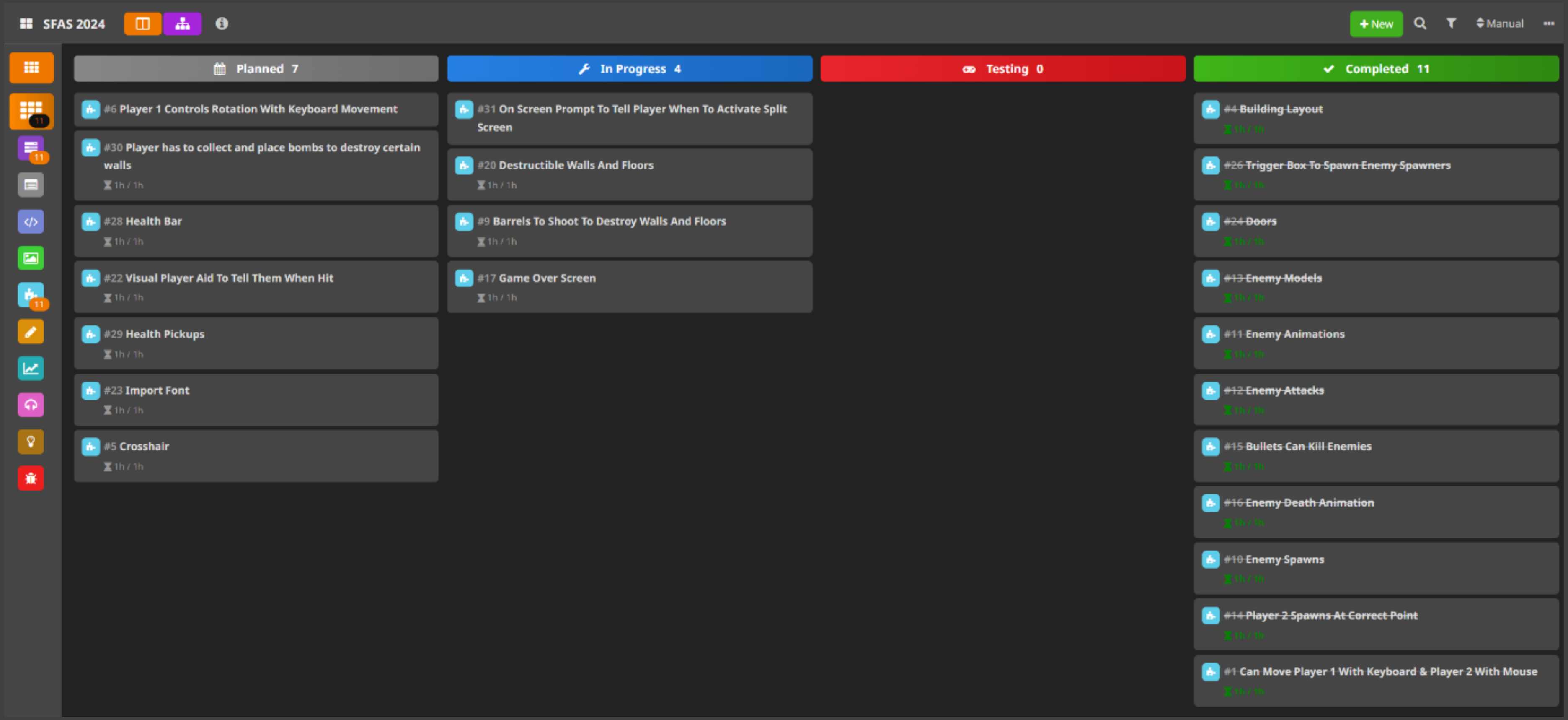
SFAS 2024 Project Plan

				11/12/2023							18/12/2023							25/12/2023							01/01/2024							08/01/2024							15/01/2024						
				11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
TASK	START	END	DAYS	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
Pre-Production	11/12/2023	19/12/2023	9	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																																	
Level Design	20/12/2023	22/12/2023	3										🟦	🟦	🟦																														
Merged Concepts Implementation	23/12/2023	30/12/2023	5													🟦	🟦				🟦	🟦	🟦																						
UI Creation	06/01/2024	08/01/2024	3																											🟦	🟦	🟦													
Additional Features Creation	09/01/2024	11/01/2024	3																														🟦	🟦	🟦										
Bug Fixing	12/01/2024	14/01/2024	3																																	🟦	🟦	🟦							
Playtesting	14/01/2024	16/01/2024	3																																			🟦	🟦	🟦					
Showcase Video Creation	17/01/2024	20/01/2024	4																																						🟦	🟦	🟦	🟦	
Submission	21/01/2024	21/01/2024	1																																										🟦

Figure 1 - Gantt Chart

HacknPlan

Furthermore, I created a HacknPlan page for my project plan. This secondary method allowed me to create the bitesize tasks and visualise them in greater detail, as I could break down each task. This method made it easy for better project management, and ease of development. Below is my HacknPlan page.



Planned 7

- #6 Player 1 Controls Rotation With Keyboard Movement
- #30 Player has to collect and place bombs to destroy certain walls — 1h / 1h
- #28 Health Bar — 1h / 1h
- #22 Visual Player Aid To Tell Them When Hit — 1h / 1h
- #29 Health Pickups — 1h / 1h
- #23 Import Font — 1h / 1h
- #5 Crosshair — 1h / 1h

In Progress 4

- #31 On Screen Prompt To Tell Player When To Activate Split Screen
- #20 Destructible Walls And Floors — 1h / 1h
- #9 Barrels To Shoot To Destroy Walls And Floors — 1h / 1h
- #17 Game Over Screen — 1h / 1h

Testing 0

Completed 11

- ~~#4 Building Layout~~ — 1h / 1h
- ~~#26 Trigger Box To Spawn Enemy Spawners~~ — 1h / 1h
- ~~#24 Doors~~ — 1h / 1h
- ~~#13 Enemy Models~~ — 1h / 1h
- ~~#11~~ Enemy Animations — 1h / 1h
- ~~#12 Enemy Attacks~~ — 1h / 1h
- ~~#15 Bullets Can Kill Enemies~~ — 1h / 1h
- ~~#16 Enemy Death Animation~~ — 1h / 1h
- ~~#10~~ Enemy Spawns — 1h / 1h
- ~~#14 Player 2 Spawns At Correct Point~~ — 1h / 1h
- ~~#1~~ Can Move Player 1 With Keyboard & Player 2 With Mouse — 1h / 1h

Figure 2 - HacknPlan Page

Project Pitch

Their Support allows the user to control two different characters at the same time, while evading enemies and traversing the building, with varying obstacles in their way. The player must control each character simultaneously. One with the W, A, S, D keys and the other with the mouse, and each character has their role to be able to progress through the level. The player must be wary of the oncoming dangers, and keep an eye on both characters simultaneously, which produces the main challenge of the prototype.

Project Concept

Inspirations

My main inspirations for this project were from *Grand Theft Auto V* (Rockstar Games, 2013), *Fireboy & Watergirl in the Forest Temple* (Oslo Albet, 2009) and *Oddworld: New 'n' Tasty!* (Just Add Water, 2014).

I gained inspiration from *Grand Theft Auto V* from the ability to play as different characters. Additionally, *Fireboy & Watergirl* contained the mechanic to control two players on the same device. However, this game is intended for two players, whereas my prototype is intended for one player. This inspiration led me to creating this prototype with this ability but controlling two characters with the same device.

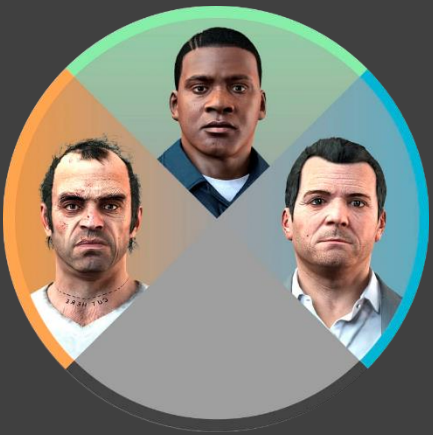


Figure 3 - GTA V Character Switcher

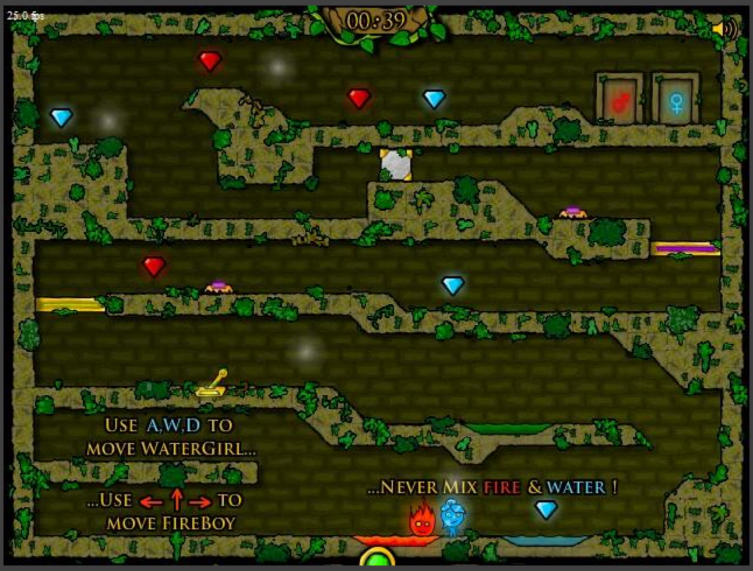


Figure 4 - Fireboy & Watergirl

The visuals for the project were inspired by *Oddworld: New 'n' Tasty*. As I wanted to include a similar perspective in my prototype. However, *Oddworld: New 'n' Tasty* includes a 2.5D graphic style and gameplay. Whereas in my prototype, the player is still able to move in a three dimensional space and has no restrictions to their movement.

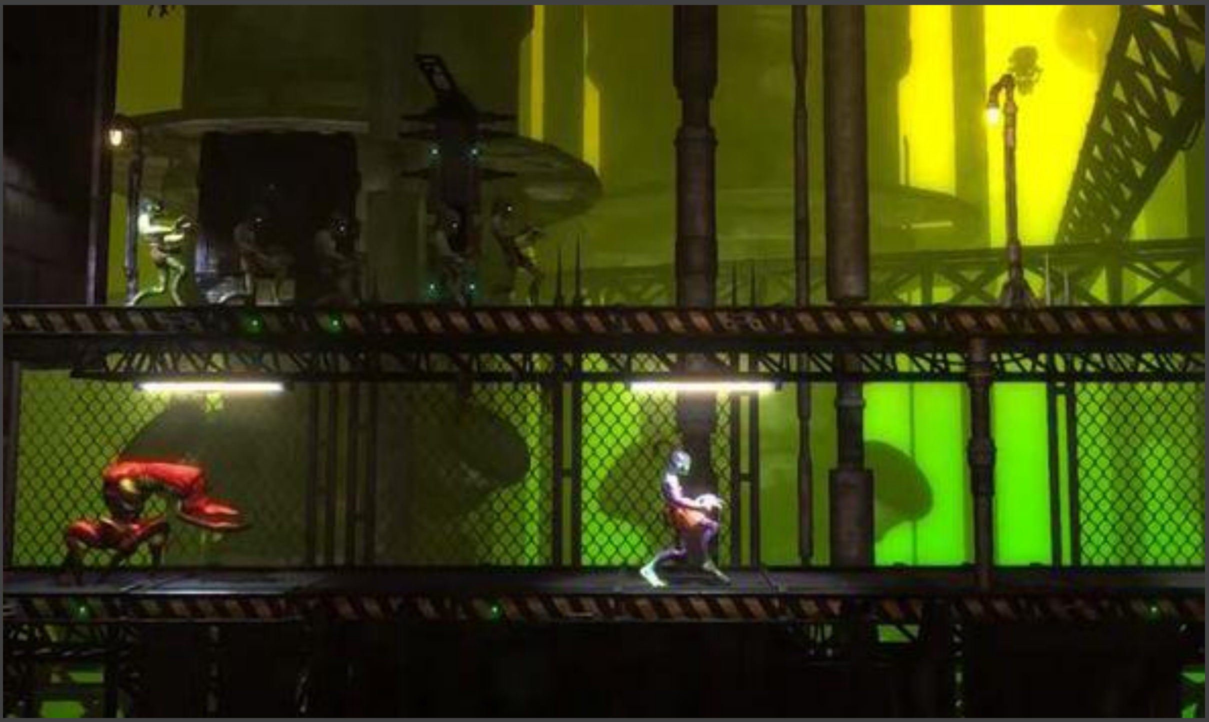


Figure 5 - Oddworld: New 'n' Tasty

USP's

- Simultaneously play as two characters with the same device
- Mixed First-Person and 3D Platformer perspectives

I believe these points to be the USP's as simultaneously playing as two different characters in a single player game is a rare mechanic found in games and shows the uniqueness of the prototype.

Similarly, the mixed First-Person and 3D Platformer perspectives is not as rare but still uncommon, and the player will have the ability to switch from this perspective at any time.

Conceptualisation

Idea Generation

For this project I was contemplating a couple of other ideas. I knew I wanted my project based around single player and cooperative game modes. One other idea I had in mind was a game where you also can play as two players simultaneously. However, the player would need to reset puzzles which have already been completed before they start the game. They would play as workers in a puzzle factory and would need to complete puzzles of varying difficulties in order to reset them for the game to be ready for the people actually completing it.

Another similar idea I had to this one was that the player splits into two characters. However, one of the characters turns into a defected player and must try and destroy the other, while the other character tries to complete tasks.

Furthermore, I had the idea of the player taking the role of a sniper and must fire into a crowd of enemies. However, one of the enemies is the player and if they shoot themselves then the game will end. The player would also be controlling the character in the crowd.

I settled on the idea of my project including two characters, but each character has a certain role to play in order to complete the game. With an unorthodox mechanic that is rare in games. I felt this idea to be the most sensible, as the gameplay makes sense, and includes game modes that do not typically feature together at the same time in games.

Merged Concepts

Justification

I chose to create this project with the concept of playing as two characters simultaneously, because I felt this was the most unique concept through my ideas, and quite an unorthodox idea too. Having the ability to play as two characters simultaneously in a game does not typically feature and presents more of a challenge than a Cooperative or Single player game. As the player would have to be mindful of the dangers of the game and their environment. With the example of the enemies being the dangers in this case. To conclude I have merged the genres of Single Player and Cooperative game mode into a Single Player game, as this concept does not typically feature together in video games.

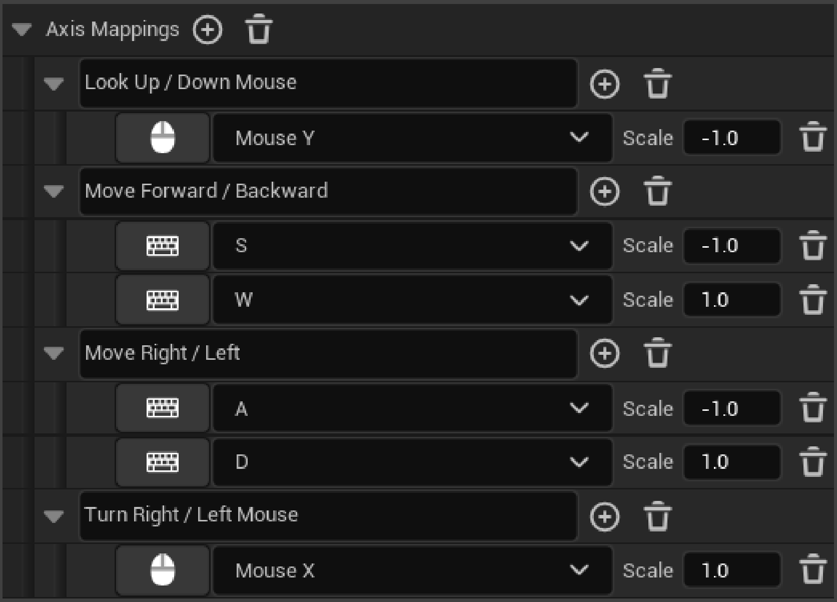
Project Name

Originally I wanted the name of the game to be *It Takes Two*. However, upon research I found there was a game of this name already and decided to be sensible and avoid any plagiarism. I then deliberated what another name could be. As I wanted the name to include a pun or wordplay. I gathered the theme of the game and eventually came up with *Their Support*. As one character has the role of the Air Support, and they are both each other's support in the game.

Mechanics Design

Spawn Second Character Mechanic

Figuring out how to control two characters simultaneously was the most complex mechanic. I needed to create the spawning events in the ‘First Person Game Mode’ Blueprint, create the ‘Axis Mappings’ for the Movement, and the nodes which would control which character should have the correct inputs.



Axis Mappings

- Look Up / Down Mouse
  - Mouse Y — Scale -1.0
- Move Forward / Backward
  - S — Scale -1.0
  - W — Scale 1.0
- Move Right / Left
  - A — Scale -1.0
  - D — Scale 1.0
- Turn Right / Left Mouse
  - Mouse X — Scale 1.0

Figure 6 - Axis Mappings

Firstly in the ‘First Person Game Mode’ Blueprint, the spawning events were created. These events spawn and destroy the secondary character, with the ‘Create Local Player’ node, setting the ‘Spawned?’ variable as ‘True’, and then setting the ‘Player Index’ variable as 1 and adding a ‘Tag’ as ‘Player 2’. The ‘Player Index’ was set to allow the second character to access the nodes, and the ‘Tag’ was needed to be able to cast to the second character. As the second character would not be able to call any nodes or inputs without this.

When the ‘DestroyPlayer2’ event is triggered then the ‘SpawnPlayer2’ is effectively reversed, with the local player and tag removed, and the ‘Player Index’ reset.

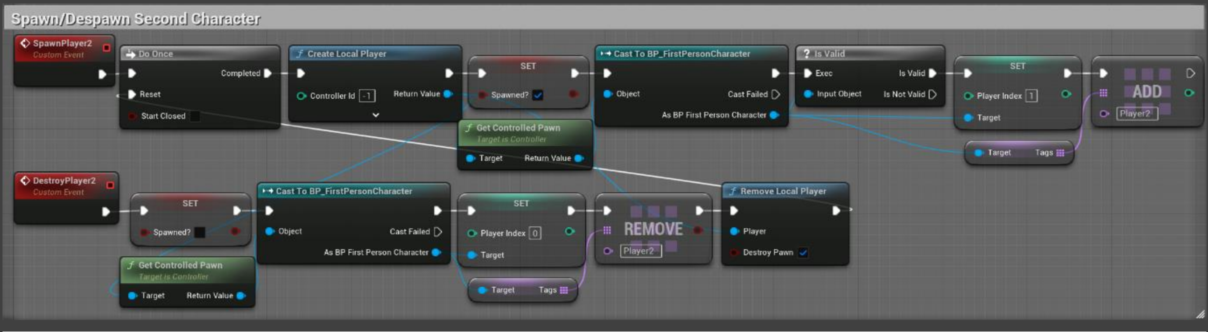


Figure 7 - First Person Game Mode Blueprint

Additionally, these events are triggered by the player pressing the ‘R’ key. I chose the ‘R’ key for this trigger as it is close to the W, A, S, D keys, and makes it easy for the player to quickly spawn the second character.

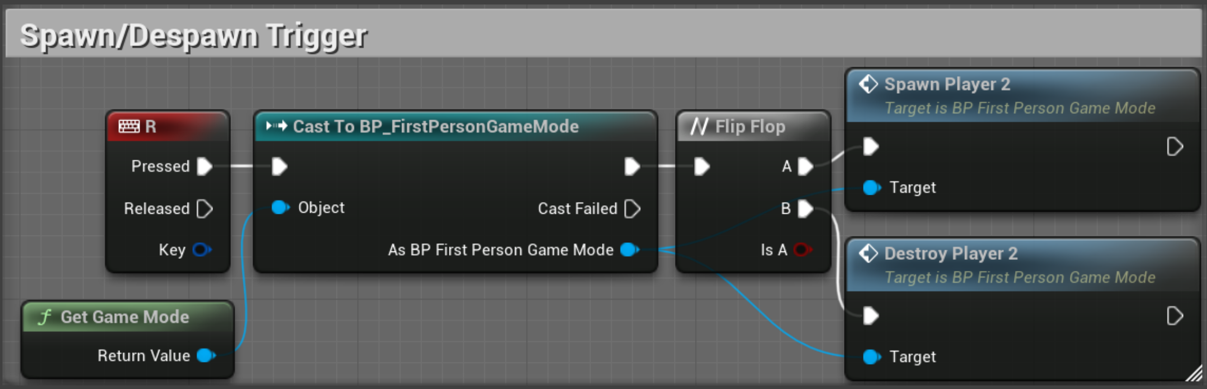


Figure 8 - Mechanic Input

Next, the inputs in the ‘First Person Character’ blueprint needed to be created, as these would allow the player to control each character simultaneously, and also switch the inputs of each character. These events would allow me to call them when the second character had been spawned and switch the inputs of each character. Since I would only have to switch the mouse input, I created the two custom events below to control the input switch.

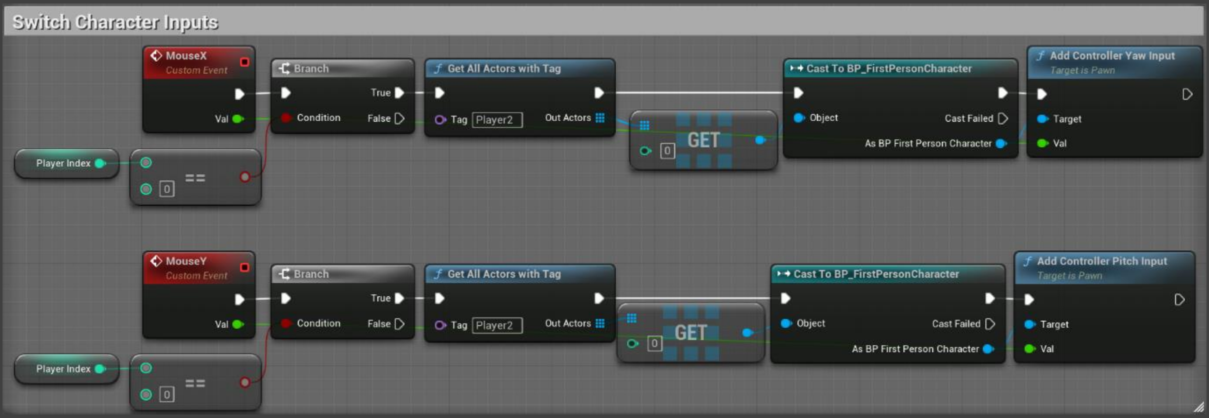


Figure 9 - Mouse Input Events

I was then able to call the ‘Mouse X’ and ‘Mouse Y’ events in the input nodes, by checking if the second character had spawned and then switching the input. If the second character had not spawned, then the mouse input would remain the same. Furthermore, I wanted the second character to spawn the rifle. This required me to set the rifle to ‘Hidden in Game’ along with the ‘Character Mesh’, which I did not want to be constantly visible.

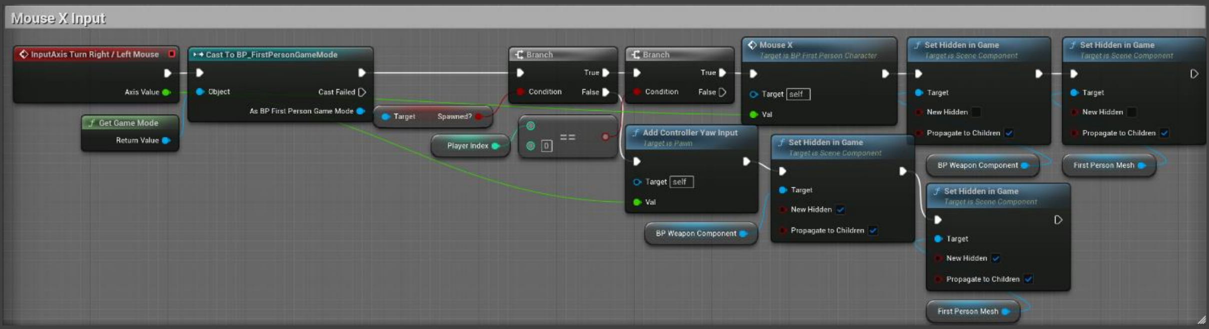


Figure 10 - Mouse X Input

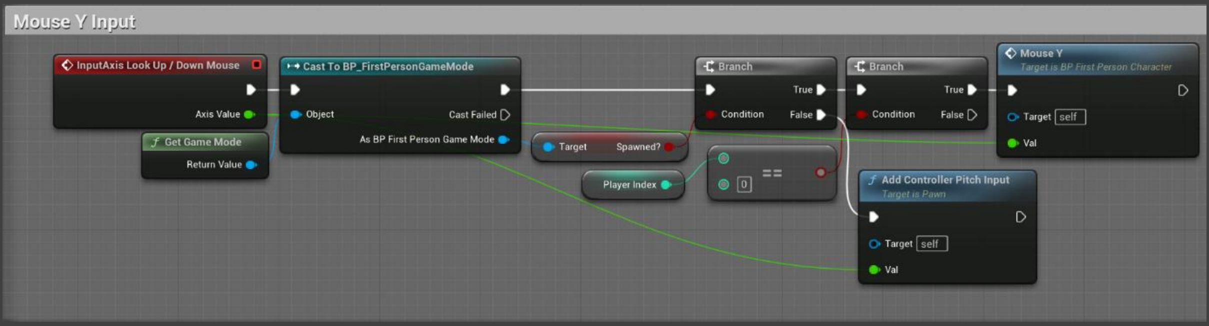


Figure 11 - Mouse Y Input

Helicopter Mechanics

The Helicopter required two main mechanics to function. These are to follow the player's movement, and end the game once the player has reached the Helicopter. The 'Event Tick' runs to follow the player's movement with no delay. I did not want the Helicopter to lose sight of the character in the building. As the player would need a constant view of where they are. Secondly, once the player has exited the building they must jump towards the Helicopter, and then win the game.

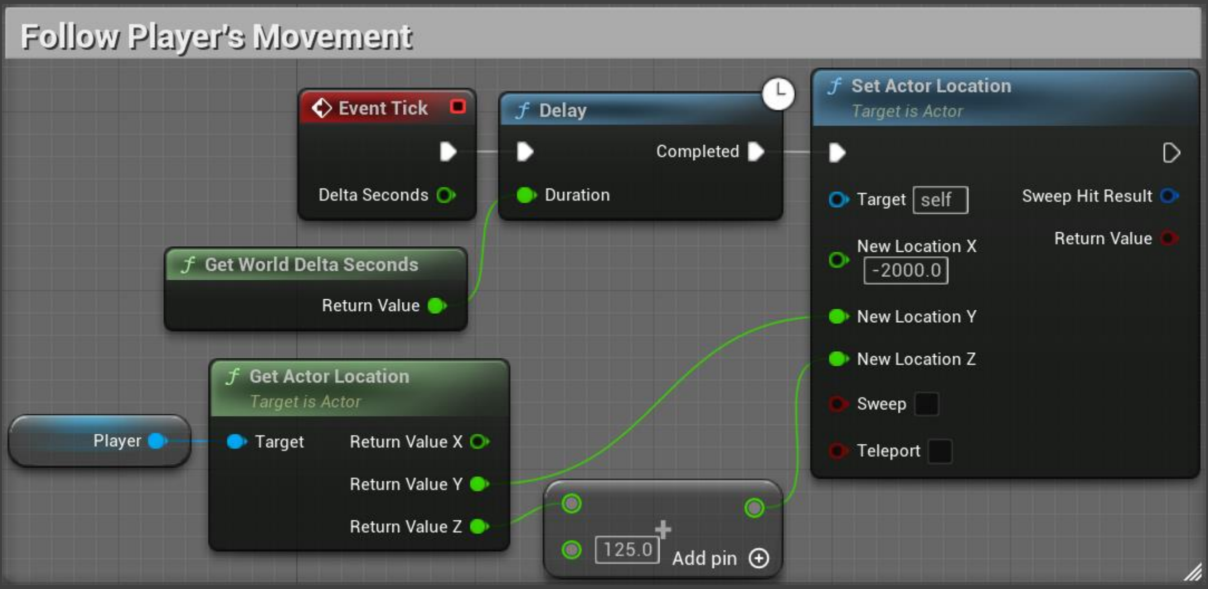


Figure 12 - Helicopter Movement

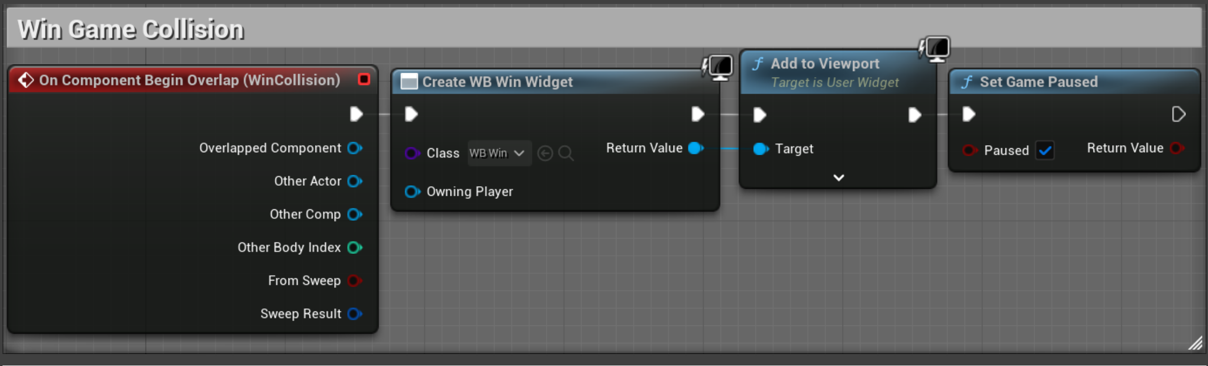


Figure 13 - Win Game Condition

Bomb Mechanics

The Bomb required more blueprints than the Helicopter. Firstly, I needed to check on player overlap if the player had already picked up a Bomb or not. As I only wanted the player to be able to carry one Bomb at a time. This was achieved with a simple collision check and then placed the Bomb into the player's inventory.

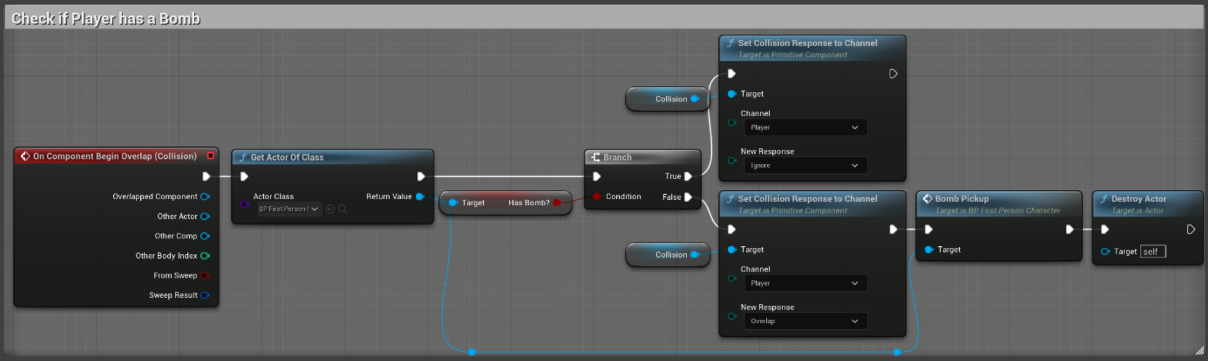


Figure 14 - Bomb Check

Once the player collected a Bomb, I then checked on the 'Bomb Prompt' Blueprint where the bomb should be placed, as there were multiple locations for this prompt. This blueprint was the visual prompt to assist the player where the bomb should be placed.

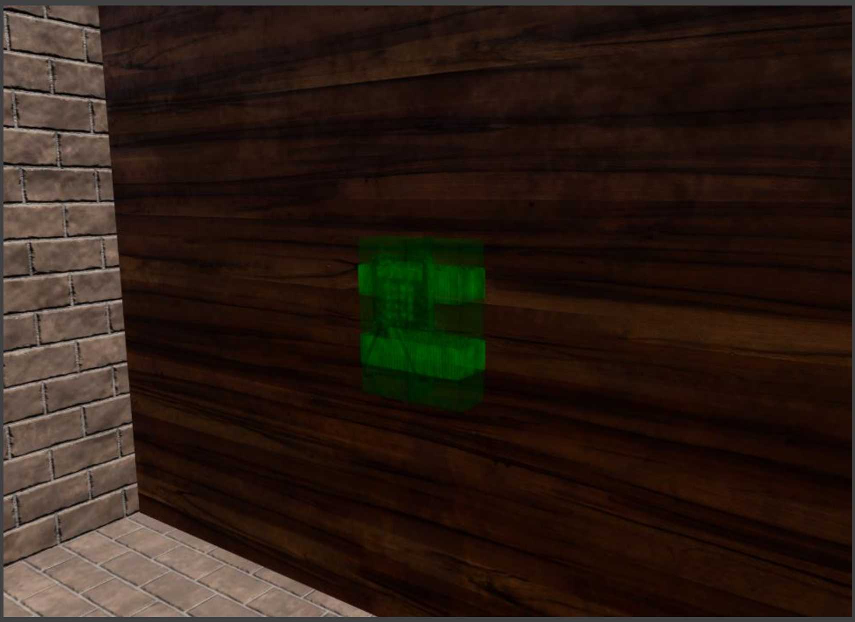


Figure 15 - Bomb Overlay

This then led to the ‘Activated Bomb’ Blueprint to be spawned. This blueprint spawns the explosion and fires the impulse to crumble the destructible walls. Thus allowing the player to progress.

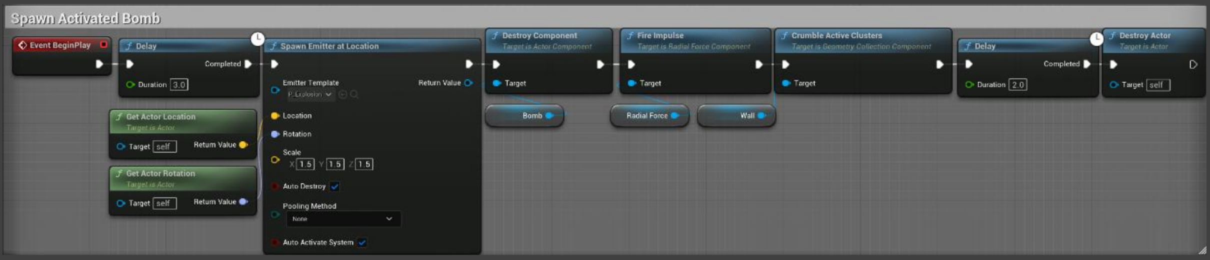


Figure 16 - Spawn Activated Bomb

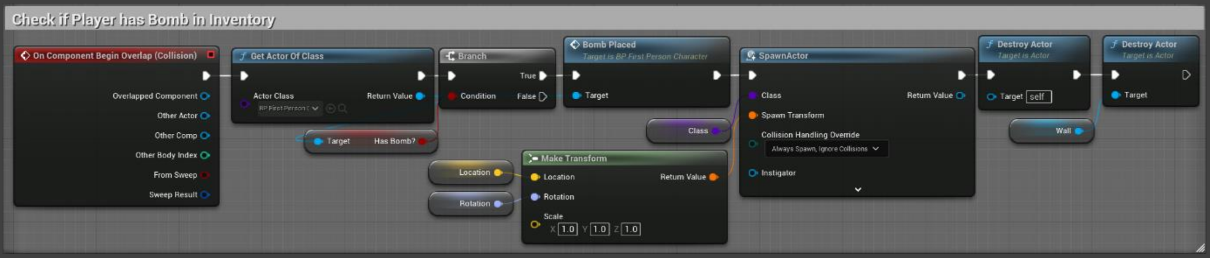


Figure 17 - Bomb Inventory Check

Level Design

Layout

I wanted to include a small playable level, with puzzles that progressively become more complex. This level would also have to restrict the players view into a 2.5D perspective, with accompanying three dimensional movement. The level would replicate that of a block of flats, with the player having to escape with the building to win the game. The layout includes features such as the explosive barrels and the destructible walls. With this layout, I could then place the enemy spawn points, health packs and bombs. Whilst making room for affordances such as the barrel and destructible walls. Below is the 2D perspective of the level layout.

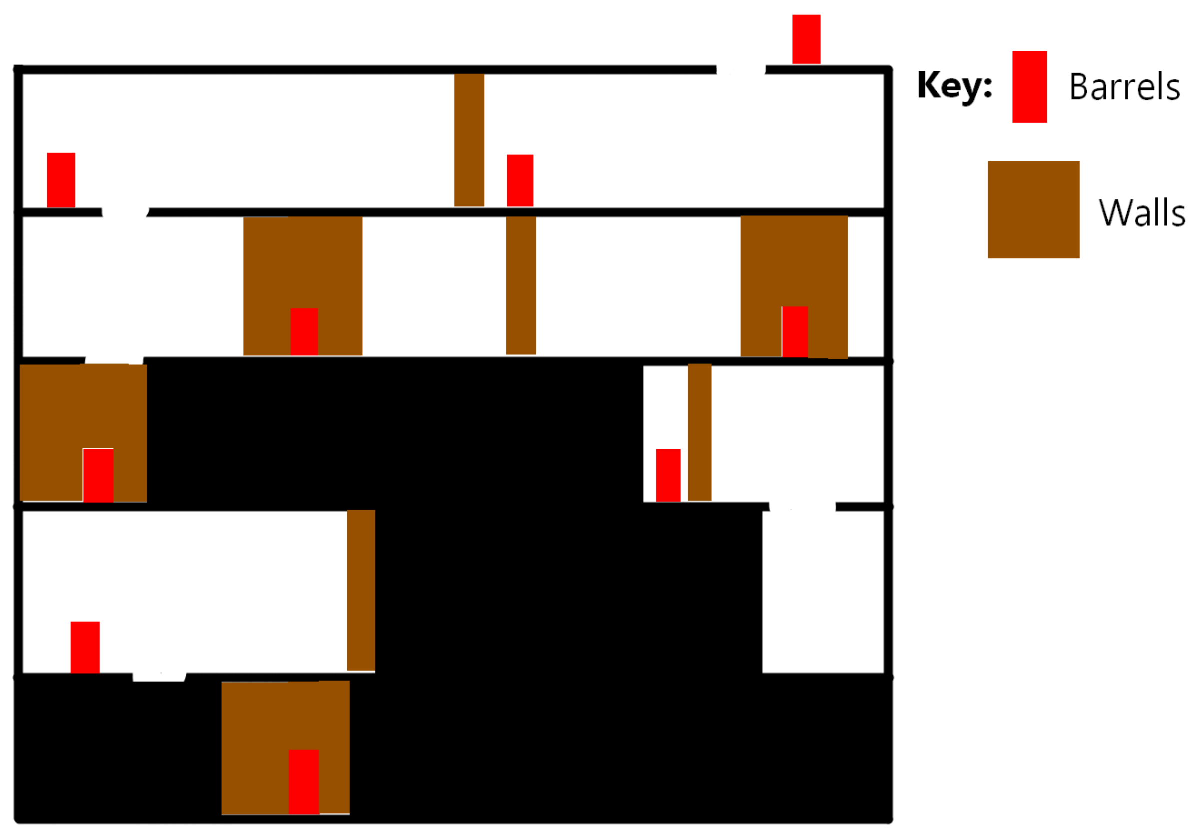


Figure 18 - Level Layout

UI Design

Purpose

Each UI has its own specific purpose to assist the player of the game conditions, or items they have acquired. The UI is visible on the screen when the second player has been spawned. The reason being because I wanted the second player to have this as their head-up display, as if they are wearing a helmet which shows this information to the player.

Health Bar

The 'Health Bar' shows on the first characters side of the screen to indicate this is the character that can be damaged. It will show when the player has been damaged and replenish once the character has collected health.

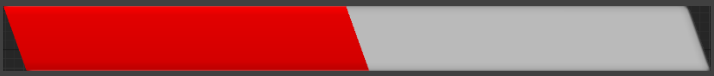


Figure 19 - Health Bar

Bomb Icon

The 'Bomb Icon' will appear once the player has collected a Bomb. This icon will then disappear once a Bomb has been placed. This visual aid gives a confirmation to the player to allow them to see when they have collected a Bomb.

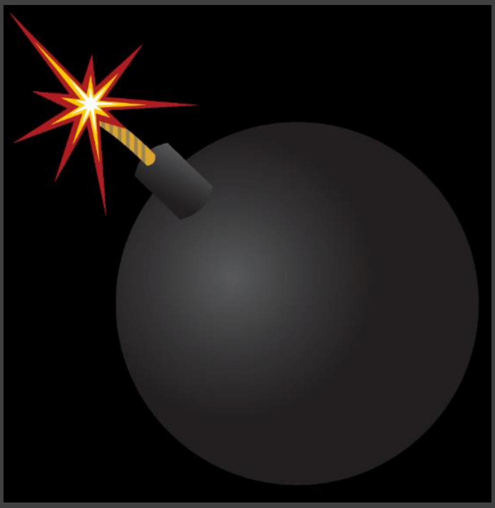


Figure 20 - Bomb Icon

Crosshair

The ‘Crosshair’ will show on the second characters side of the screen, to assist the player where they are aiming. Without this it would be very difficult to aim correctly, and it is not too large to distract the player from anything else on the screen or block the players visibility.

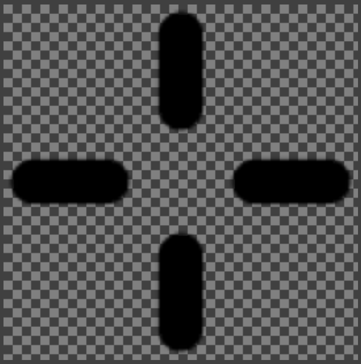


Figure 21 - Crosshair

Damage Indicator

When the player is hit by an enemy, then the border of the screen will flash red to indicate to the player that they have been damaged. This is a quick flash, and will not stay on too long, in order to not be an annoyance to the player.

Furthermore, the first character materials start as green, and will briefly turn red to serve as another visual aid to the player, to let them know they have been damaged.

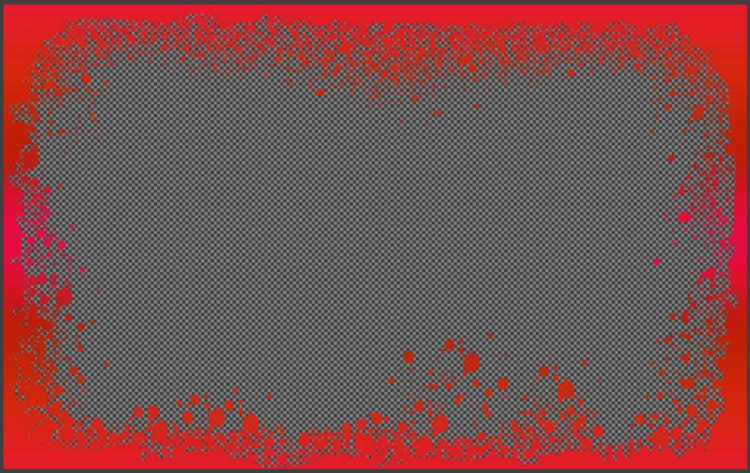


Figure 22 - Blood Overlay

Win Game Screen

Once the player has won the game, then the Win screen will appear. This will then allow the player to quit the game or restart.

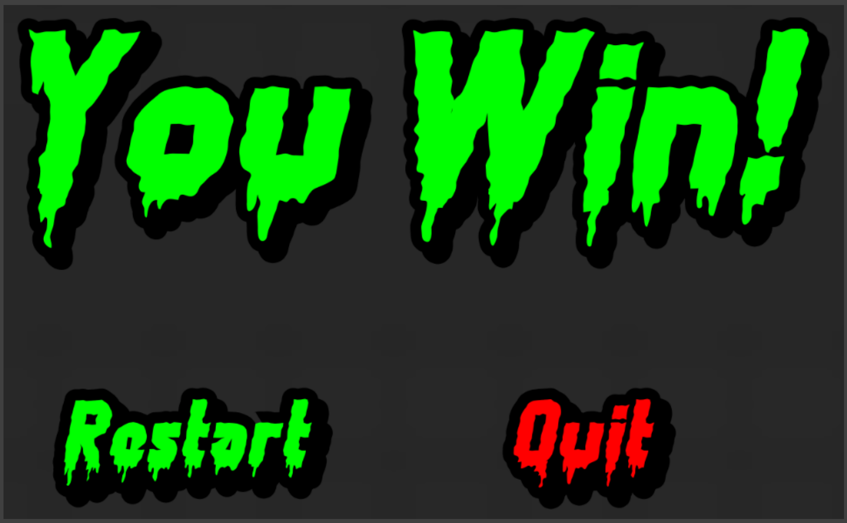


Figure 23 - Win Screen

Game Over Screen

The Game Over screen will appear once the player has died and will allow the player to restart or quit the game. Both the Game Over and Win screen follow the theme of the game with their font, which is a zombie style theme.

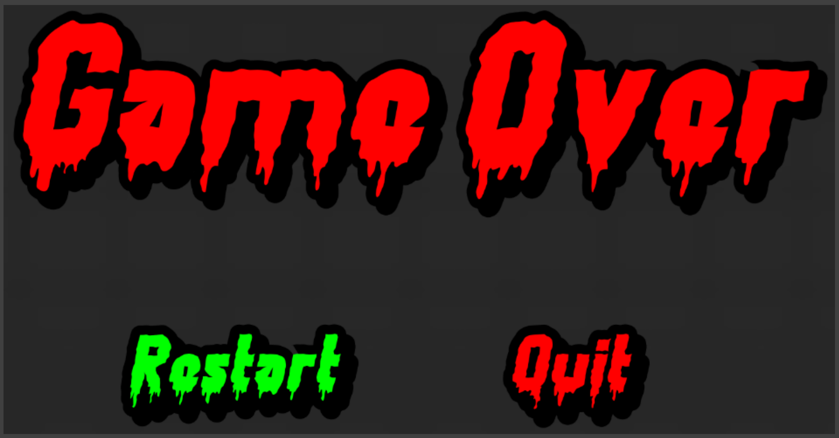


Figure 24 - Game Over Screen

Additional Features

Enemies

I chose to use Zombies as the enemies for this prototype. Since I wanted a slow moving enemy, I felt these characters were the most suitable choice. I created their statistics to not be too difficult for the player. Additionally, to show the player that the Zombies have spawned I created a red mist, which indicates that the spawners have been activated. Below are the Zombies statistics.

Spawn Rate	7 Seconds
Movement Speed	150 cm/s
Attack Damage	5 - 10
Attack Speed	1.05 secs
Health	10

Below are the player's statistics.

Movement Speed	600 cm/s
Health	100
Damage	10

Health

I decided to include health pickups, as the first character can lose health, to aid the player in finishing the game. The health packs are picked up when the first character is close to them, and they immediately fill the characters health to max health.

Explosive Barrels

Explosive Barrels have been logically placed for the player to shoot, in order to progress through the building. These barrels will explode on impact of the rifle projectile and cause the corresponding wall or planks to crumble due to the explosion. These barrels serve as an affordance to the player, as they stand out against the environment, and the player is taught how to use them to their advantage.

Breakable Walls

The Breakable Walls are features for more complex gameplay and puzzles. The player must use the second character to assist the first character to destroy the barrels near the walls to destroy them. Without the second character, then the first character would not be able to progress. Once the walls and planks have been destroyed they will disappear after 2.5 seconds.

Bombs

The Bombs are another additional feature which serve as another small puzzle for the player. They can only carry one Bomb at a time, and then find the correct position to place the Bomb. The player will have a visual aid to allow them to see where they can place the Bomb. Once the Bomb has been placed, the wall which the Bomb is placed will crumble and then disappear.

Player Death

If the player has taken enough damage from the Zombies, then they will die, and the game will end. This will then show the Game Over screen, which will then allow the player to quit the game or restart.

Player Win

Similarly to the Game Over screen, the Win screen will appear once the player has reached the Helicopter. This will also allow the player to quit the game or restart.

Play Testing

After numerous playtesting sessions I found a couple of issues, that did slightly affect the gameplay, and implemented some late additional features. One issue I encountered was that Character One's camera cannot rotate when Character Two has spawned. I had experimented with this issue and was able to rotate Character One's camera. However, this rotation was at a 90 degree angle each time, and the forward movement of the character did not adjust to the rotation angle. Meaning the W, A, S, D keys still moved the character in their original directions.

Another issue I encountered was the destructible walls automatically crumbled at the start of play. This issue was frustrating, as I had set the 'Damage Threshold' to a high limit which should not crumble automatically. To fix this issue, I turned off 'Auto Activate' and crumbled the walls once the relevant barrel had been exploded.

After playtesting, I then implemented additional features for the player, which would assist them with the gameplay. Such as prompts which inform the player which button activates the spawning second character mechanic, and how to win the game.

Conclusion

Future Development

I would plan to develop this prototype into a full playable experience in the future, by making the gameplay experience longer, adding more features and puzzles and general polishing of the project.

One feature I would implement would be to make the second characters camera more dynamic. As it currently moves with the first character instantly, instead of having a small delay. This camera was inspired by *Oddworld: New 'n' Tasty*, which includes the type of camera I would ideally like to implement into the prototype.

Another feature I would like to implement would be able to have the characters move around the side of the building, instead of limiting the gameplay to two axes. This would allow me to create a better gameplay experience and show more creativity within the prototype.

I have enjoyed creating this project and have gained more skills to create more complex mechanics with each project. Giving me more confidence with each project, and I hope to be able to continue the development of this project in the future.

Asset References

3D Models

3DDomino (2020) *Helicopter bell* [3D Asset]. Available online: https://sketchfab.com/3d-models/helicopter-bell-20d80ac1480b42eb855b8136a4c411b4

Mixamo (n.d.) *Romero* [3D Character]. Available online: https://www.mixamo.com/#/?limit=96&page=1&query=romero&type=Character

TheoClarke (2020) *First Aid Box* [3D Asset]. Available online: https://sketchfab.com/3d-models/first-aid-box-edaba58075ff4796a3ff5c4a43a765a7

Thunder (2021) *C4- Bomb* [3D Asset]. Available online: https://sketchfab.com/3d-models/c4-bomb-b994b09748254607a8382e51d23f4c86

Animations

Mixamo (n.d.) *Zombie Attack* [3D Animation]. [Description: Zombie Swipe Attack]. Available online: https://www.mixamo.com/#/?page=1&query=zombie+attack&type=Motion%2CMotionPack

Mixamo (n.d.) *Zombie Death* [3D Animation]. [Description: Zombie Getting Hit And Falling Onto Back]. Available online: https://www.mixamo.com/#/?limit=96&page=1&query=zombie+death&type=Motion%2CMotionPack

Mixamo (n.d.) *Zombie Idle* [3D Animation]. [Description: Zombie Alert Idle]. Available online: https://www.mixamo.com/#/?page=1&query=zombie+idle&type=Motion%2CMotionPack

Mixamo (n.d.) *Zombie Walk* [3D Animation]. [Description: Zombie Walking]. Available online: https://www.mixamo.com/#/?limit=96&page=1&query=zombie+walk&type=Motion%2CMotionPack

Images

1up Indie (2019) *Blood Overlay* [Image]. Available online: https://opengameart.org/content/blood-overlay

588ku (n.d.) *Cartoon Black Bomb Illustration* [Image]. Available online: https://pngtree.com/freepng/cartoon-black-bomb-illustration_4737604.html

Brandon Shields (2017) *Crosshair* [Image]. Available online: https://thenounproject.com/icon/crosshair-1022053/

Font

Display Studio (2021) *Zombie* [Font]. Available online: https://www.dafont.com/zombie-3.font