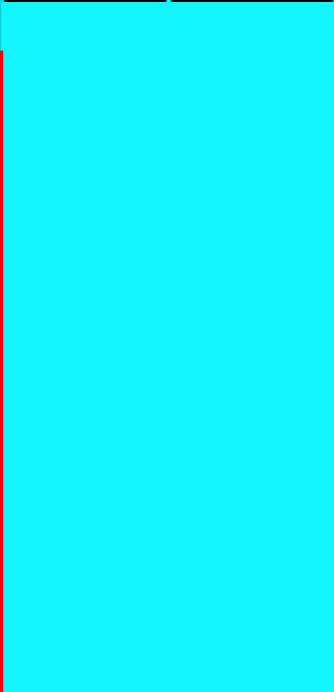


DYNAMIC DIFFICULTY SYSTEM

By Leo Clarke

Games Development Project



CONTENTS

INTRODUCTION

AIMS

DELIVERABLES

RESEARCH METHODOLOGIES

DOCUMENTATION OF PRODUCTION

EVALUATION

FURTHER IMPROVEMENTS

BIBLIOGRAPHY

Introduction

Games have developed the length of their gameplay through the addition of difficulties, seen since the 1970/1980's in arcades and at the dawn of home consoles. However, how to present these difficulty options has been an enduring question in the gaming industry.

How to design a game that *changes* based on the player's **skills**

Luckily, this question has already been answered and been presented in several ways. I seek to understand the **Dynamic Difficulty System** and how different games present it in different ways. Using these, I believe I can build from their achievements and blunders, developing my understanding on **gameplay management** and how gameplay connects with **user experience**.

Aims

The development of difficulty systems has become rather mainstream - using the regular Easy, Normal, Hard. This project seeks to develop on that through involving the player's skills in gameplay to define the current difficulty, and to also make it dynamic – presenting change from start to finish of a session.

My Objectives:

1. Research DDA/DDS in games to find watched variables and counters to loopholes
2. Design a difficulty system that analyses in-game variables to change difficulty
3. Create a looping gameplay experience to visualise change in gameplay and different final outcomes.
4. Obtain and iterate on feedback through testing.

Deliverables

This is a **2.5D Unreal Engine** Project with at least 5 minutes' worth of gameplay that resembles a **Horde Game**.

It should be **replayable** and change in difficulty depending on the player's **quality of gameplay** during the playthrough.

The project will include a gameplay level, main menu screen and game over screen.

The flow of the game should hopefully introduce **tutorialisation** through gameplay and enemy design, and UI features.

Sound and VFX will exist but can change during research, testing and iteration.

By the end of my project, my testers should receive a **different experience** compared to the others, in at least one different way.

RESEARCH METHODOLOGIES

DDA

Several games have already used this system, so my primary source of research comes from analysing their DDA/DDS and see the benefits.

The key concepts I want to understand is **what is being changed** and **what is being observed**.

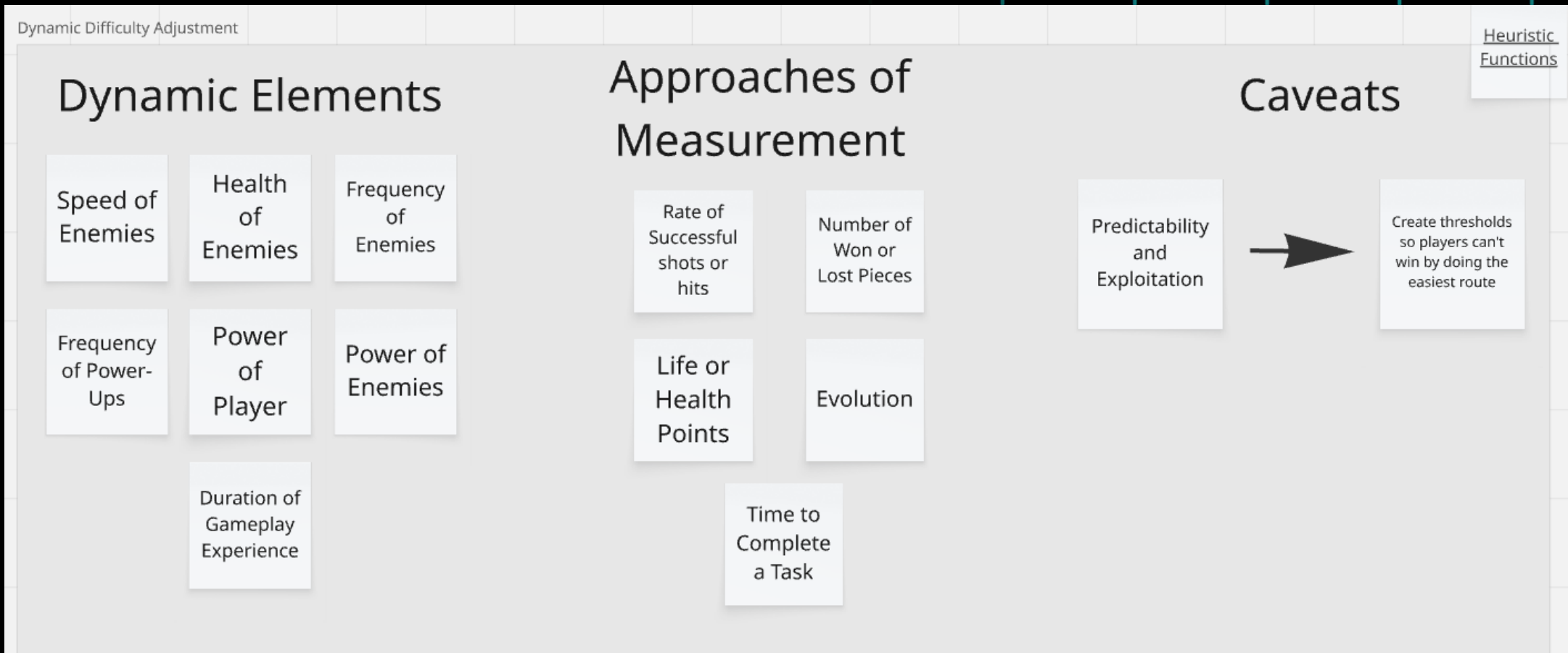
Looking at existing games is the best source of research on this topic. It has already been presented in the games industry so I should use this source to study its pros and cons.

These were studied to be common variables such as health, wins/losses or position/rank.

However, research also proved that it could also be accuracy, quality of items or the time it takes to beat an area.

DDA Research Summary

Using research sources, I can create a base study of most DDAs that exist in games. I can plan out which variables to observe during gameplay and what to change as well.



Resident Evil 4 - DAS

RE4 has a Difficulty Adjustment System which is more developed than other game series. It follows a score and rank system.

Variables contribute to the total score, and the score places itself within 1 of the 11 ranks.

It was a major source of influence in designing a DDA, offering a variety of variable sources and variable alterations.



Gameplay

Another important source I needed to investigate for the project to function is the gameplay experience that the DDS would be built into.

My focus – **Replayability**.

This game needs to be something repeatable and something quick enough for players to get a different experience each time they play it. In this style, I can gather data quicker than a longer narrative experience and the data should also be different between testers.

Because of, I looked towards Arcade Games or Horde Games.

DOCUMENTATION OF PRODUCTION

MoSCoW

This MoSCoW presents the necessities in my project. Must have's be practically steps that I need to complete for my project to work and the Should have's contain features done in polish.

I wanted to include a few of the Could have's, yet they were never a priority.

[\[LINK\]](#)

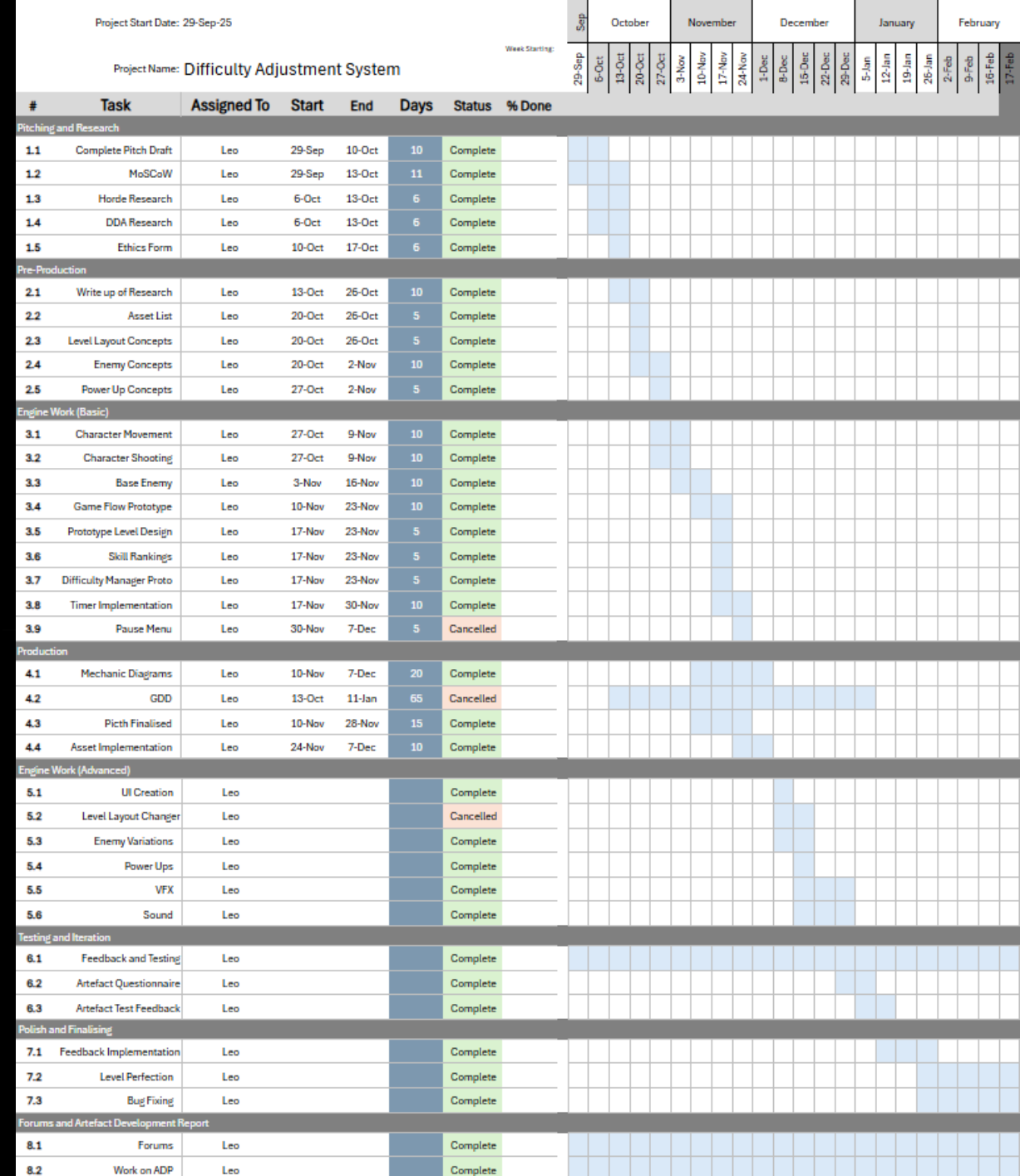
MoSCoW	Concept	Detail	Completed?
MUST	Variable Manager	A manager within engine to watch the player's variables then change the enemies variables, through a summed ranking.	Yes
	Player Skill Ranking	Judged by the player's health, current kills, power-ups used and the total time they have played. It will roughly go through the rankings of 1 - 11	Yes
	Player Movement	The player will move on a 2D axis in all directions through arrow keys or joystick on controller	Yes
	Player Projectiles	The player will be able to shoot projectiles, aimed using either mouse or another joystick	Yes
	Player Health	A float-based health system instead of Lives to better work alongside the Difficulty Manager	Yes
	Enemies	The game will introduce enemies of different varieties, be they harder, better, faster or stronger.	Yes
	Enemy Open Variables	The enemies' health, damage, speed and spawn rate will all be editable during gameplay for the Difficulty Manager	Yes
	Start Screen	The game will have a start menu, hopefully introducing a settings option and possibly a scoreboard.	Yes
	Game Over Screen	Upon death, there will be a Game Over screen that will tell you your score and allow you to return to the main menu.	Yes
SHOULD	UI Pop-ups	UI visuals for damage done, certain mechanic locations that need to be dealt with and directions where the player might need to go.	Yes
	VFX	Particle systems for projectiles, movement and death to make the game more flashy.	Yes
	Repeatable Game Loop	The playthrough of the game should be repeatable and insite replayability.	Yes
	Timer	A time of the current playthrough from start to finish, possibly used within ranking.	Yes
	Pause Menu	A way to pause the game to offer a setting change or to return to menu, all while pausing the timer as well.	No
	Map Sections	The level layout can be broken up into smaller sections that the player will be forced into, directing them to areas with more or less obstacles	No
	Power-ups	Pickups within the level to increase, durability, damage, projectile speed, projectile count, etc... These pickups will be connected to the Difficulty rank and will contribute to it.	Yes
	Sound Effects	Audible Notifications for projectiles, enemy deaths, damage taken, movement (possibly), power-ups and in-game events or warnings.	Yes
Settings Menu	A way to change resolution, fullscreen, learn controls or change sound.	No	
COULD	Emissive Assets	For polish, each asset could glow and offer lighting to the game instead of a plane 2D design.	Yes
	Accessibility	A way to turn off functions like VFX, add colourblind mode, TTS for blindness	No
	Rank Visuals	A visual Way to see your current ranking like in games such as Devil May Cry.	Yes
	Easter Eggs	Possibly add a small easter egg within the game either referencing old arcade easter eggs or something along those lines.	No
	Procedural Generation	Level could be procedurally generated, or at least just in the designetd rooms, which is influenced by Difficulty Ranking.	No
	3D Assets	Going from a 2D concept, 3D assets could be introduced to allow a 2.5D experience	Yes
WON'T	Narrative	This game is arcade-style and won't likely have any narrative progression. The most would be a background story like Pacman.	No
	Copyright Material	All used assets will be under permission of the creator, possibly with crediting in a credit page such as a list for sounds.	No
	Voice Acting	This game will have no need for VA nor any form of subtitles or dialogue boxes.	No
	Save System	This game is intended to be short and replayable yet I won't design a save system so it remains like an arcade game.	No
	Bugs	This game should hopefully be playable and any possible bug won't effect the experience.	No

Gantt Chart

This is my planning for each step I'd need to take. A big hiccup in this was the push towards a demo for my presentation. As such, a few tasks were delayed yet it didn't entirely rush my process.

I decided to at least give every task a week to complete and that task could only take a couple hours. Because of this, the Gantt Chart isn't exactly exact to the steps I took. Some concepts like Enemy Variations was designed before a Timer.

[\[LINK\]](#)



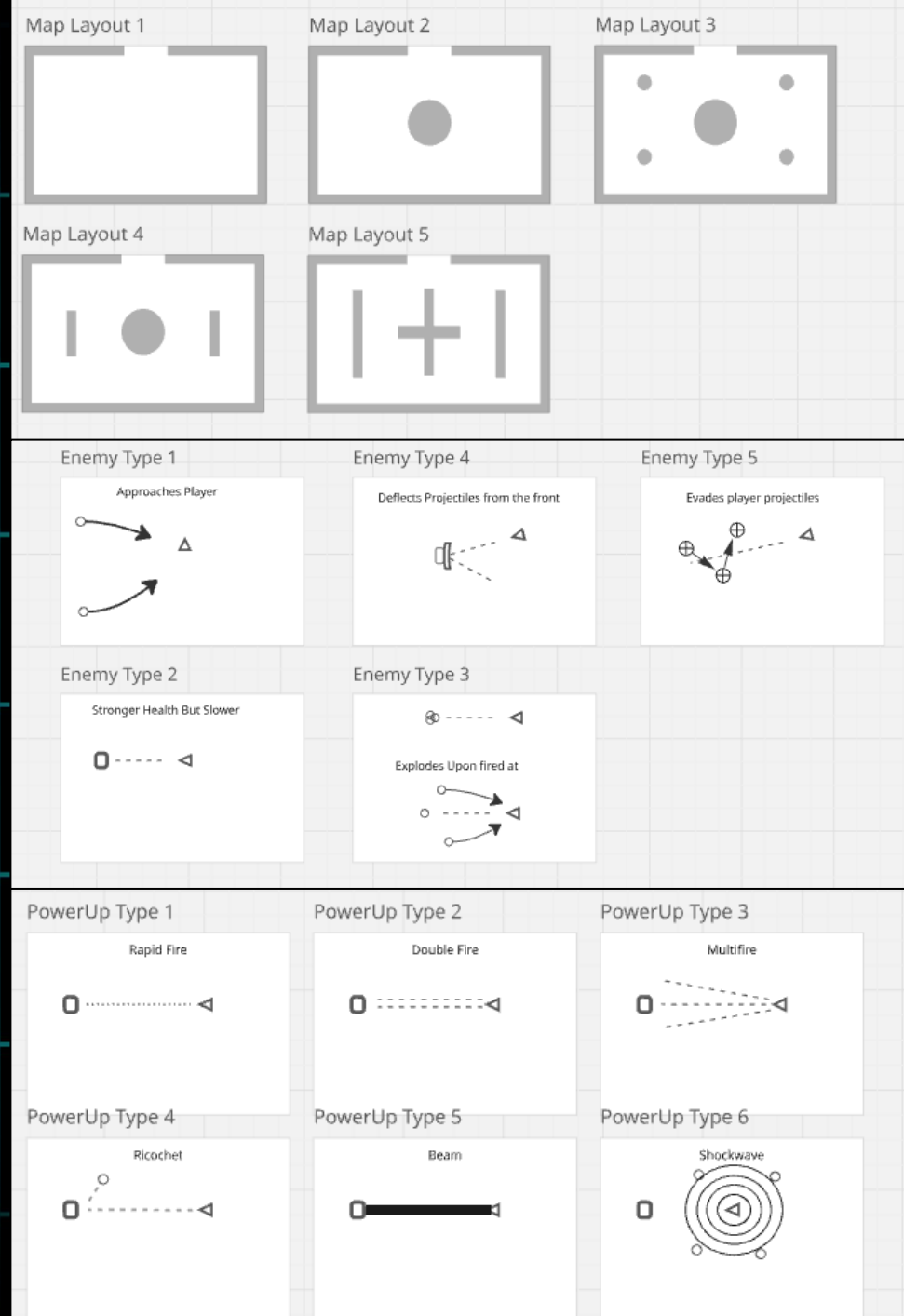
Gameplay Planning

Via Miro Board, I began planning out features that would be necessary for both the gameplay and the DDS.

These assets would be variables whose spawn chances would be affected by the DDS so I ranked them from 1-5 to present this.

The map had several concepts – finally choosing Map 5.

The enemies and power ups were assets that were affected by the DDS so these needed numbers and differences. The main change that occurs are their spawn chances.



DDS Plan

Taking a lot of influence from RE4 DAS, I created my DDS to follow a rank system.

To better present my chosen variables, I listed all the changes that occurs with each rank.

I did my best to avoid making the enemies' health as the main change of the DDS, I did include it as a consideration, and it only offers 1-3 extra hits which take a matter of seconds.

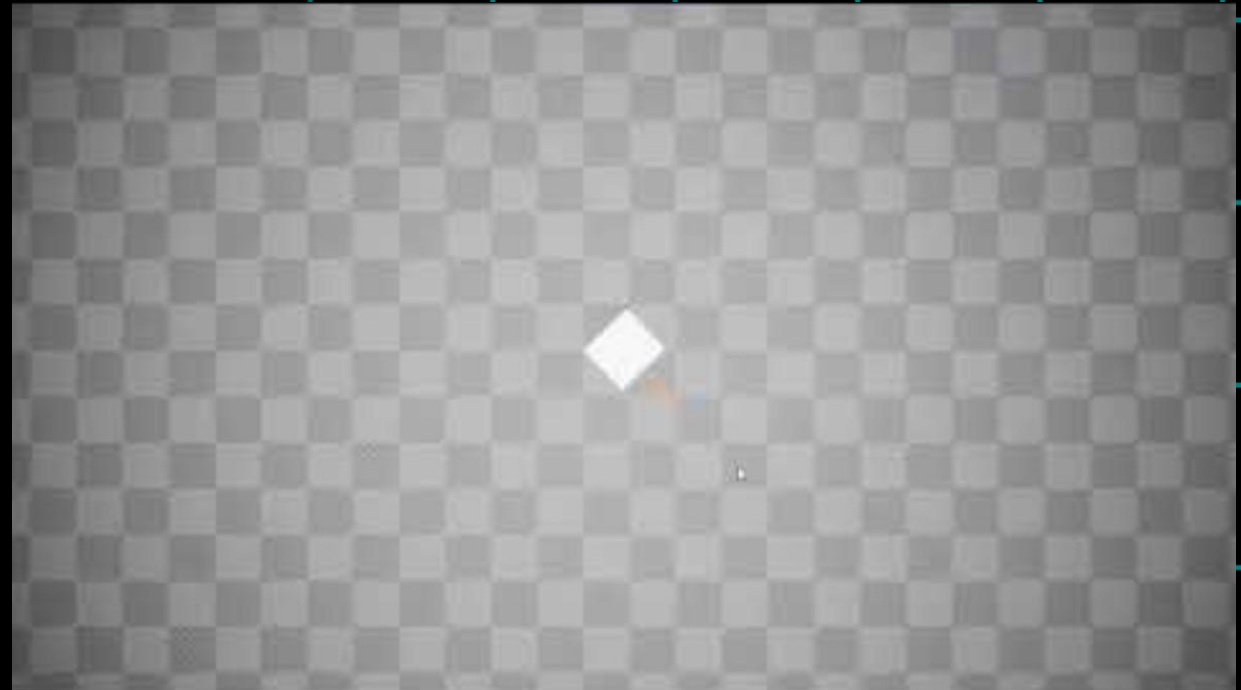
Rank	Wave Size	Speed Boost	Enemy Types	Enemy Health	Power Ups
Rank 1	10	0% Bonus	1	Basic	40% Chance
Rank 2	13	5% Bonus	1	Basic	40% Chance
Rank 3	16	10% Bonus	1,2	Basic	40% Chance
Rank 4	20	15% Bonus	1,2	1 Extra Hit	30% Chance
Rank 5	24	20% Bonus	1,2	1 Extra Hit	30% Chance
Rank 6	28	25% Bonus	1,2,3	1 Extra Hit	30% Chance
Rank 7	32	30% Bonus	2,3	1 Extra Hit	30% Chance
Rank 8	36	40% Bonus	2,3,4	2 Extra Hits	20% Chance
Rank 9	40	50% Bonus	2,3,4,5	2 Extra Hits	20% Chance
Rank 10	45	60% Bonus	3,4,5	2 Extra Hits	20% Chance
Rank 11	50	100% Bonus	4,5	3 Extra Hits	10% Chance

Player Movement

Originally, I intended to create the game entirely in 2D, yet a 3D world offered me more creativity in design.

It uses a basic WASD movement that can move diagonally. The main body of the player actor rotates to face the position of the mouse in the 3d world.

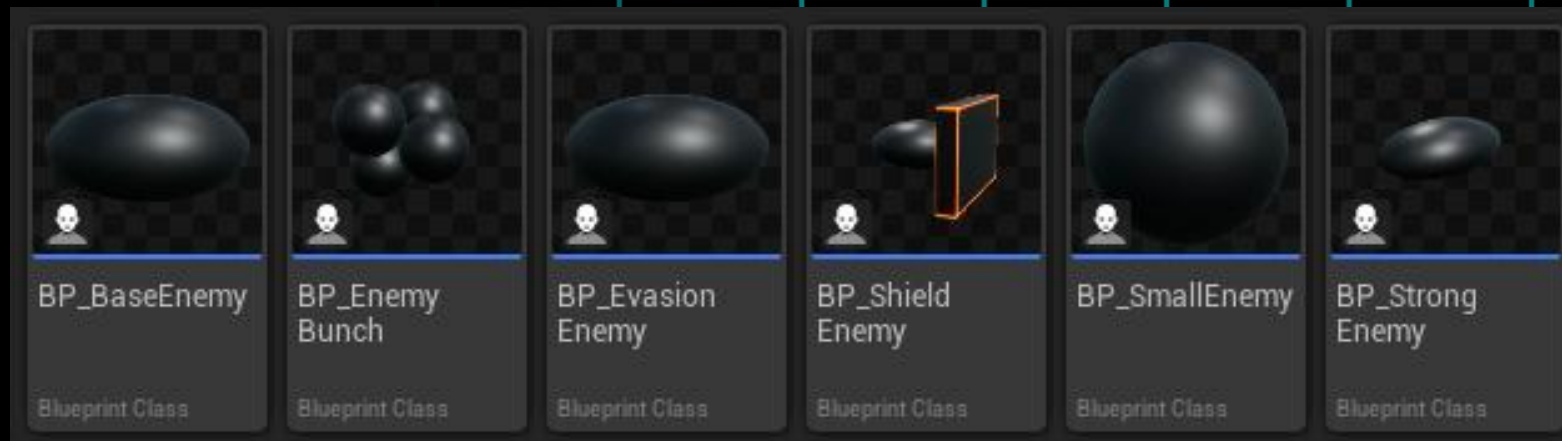
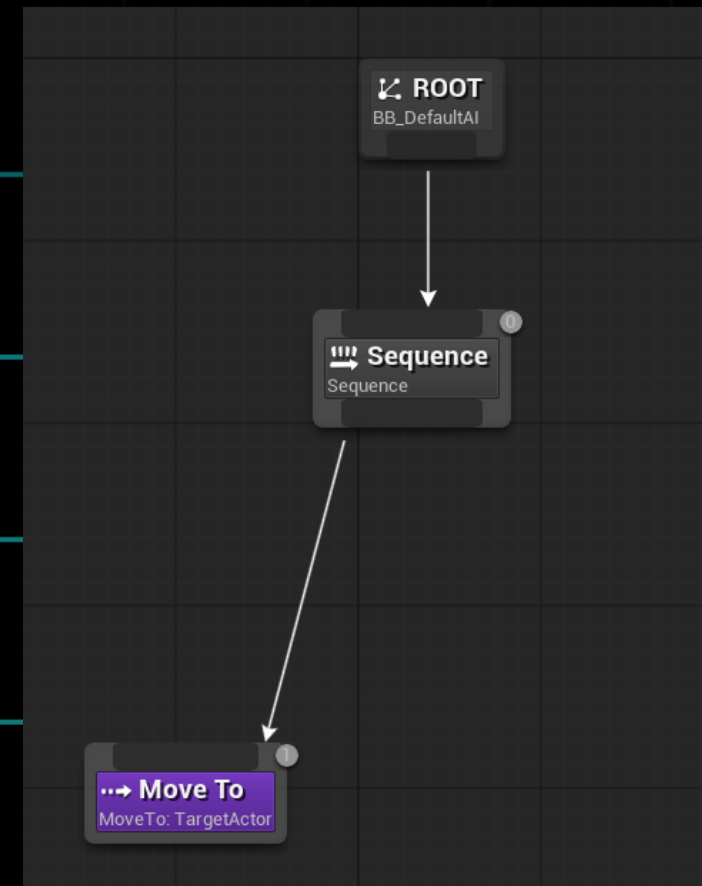
[LINK]



AI and Enemies

I designed the enemies to follow the player, commonly seen in horde games such as COD Zombies. It is a basic behaviour system but it allows the AI to easily access the nav mesh and quicken rotation options.

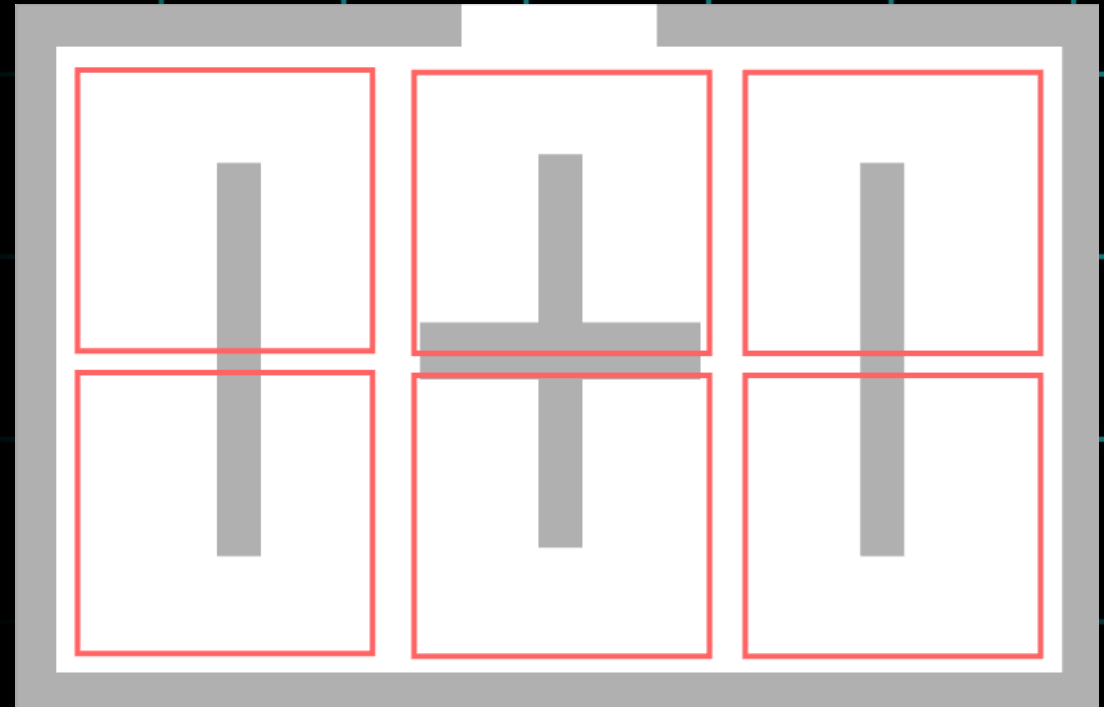
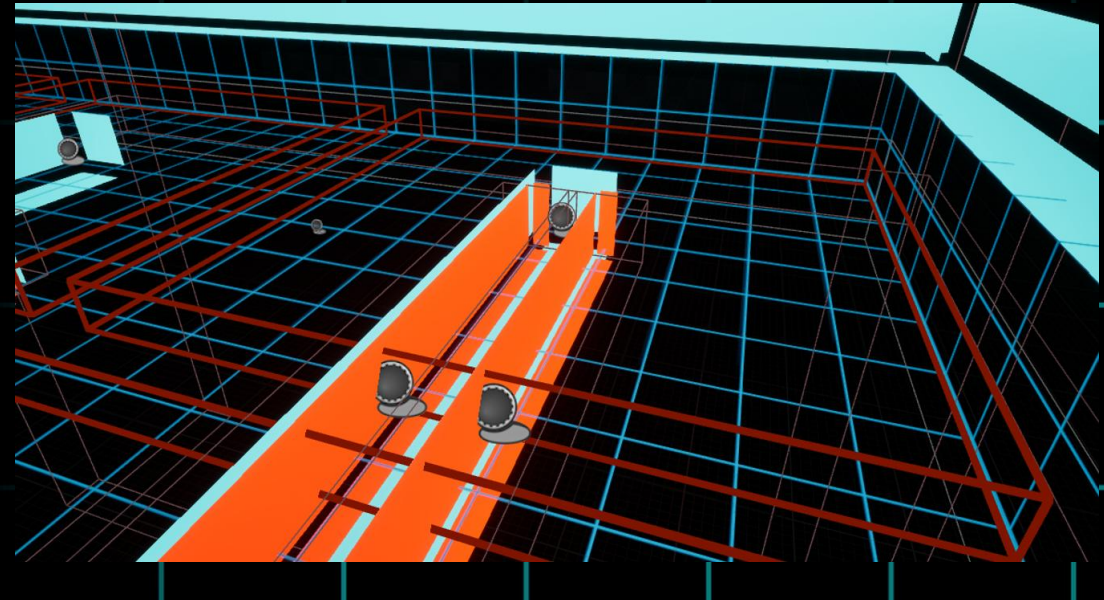
Below is the Parent Enemy and the Children. All their repeating systems such as health, death and damage are all found within the parent.



Spawning and Level

I broke up the spawning locations into 6, splitting them evenly across the map. The intention is to deactivate the spawner than the player is currently in.

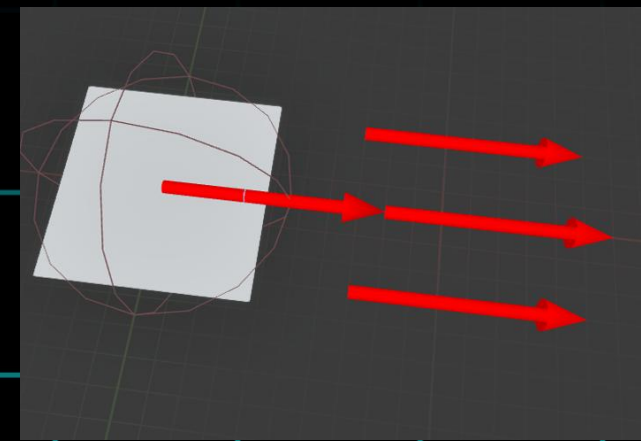
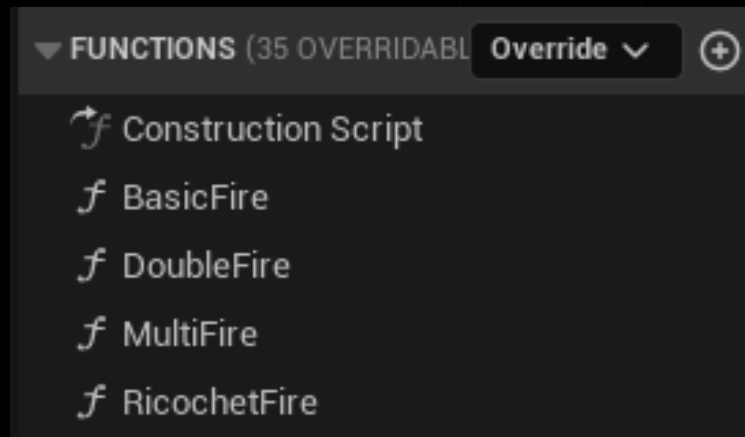
This makes gameplay feel less intensive than it would be if enemies could spawn right next to the player. My biggest issue with this was discovering why enemies spawned on the central walls. It was an easy fix which needed one node to be changed.



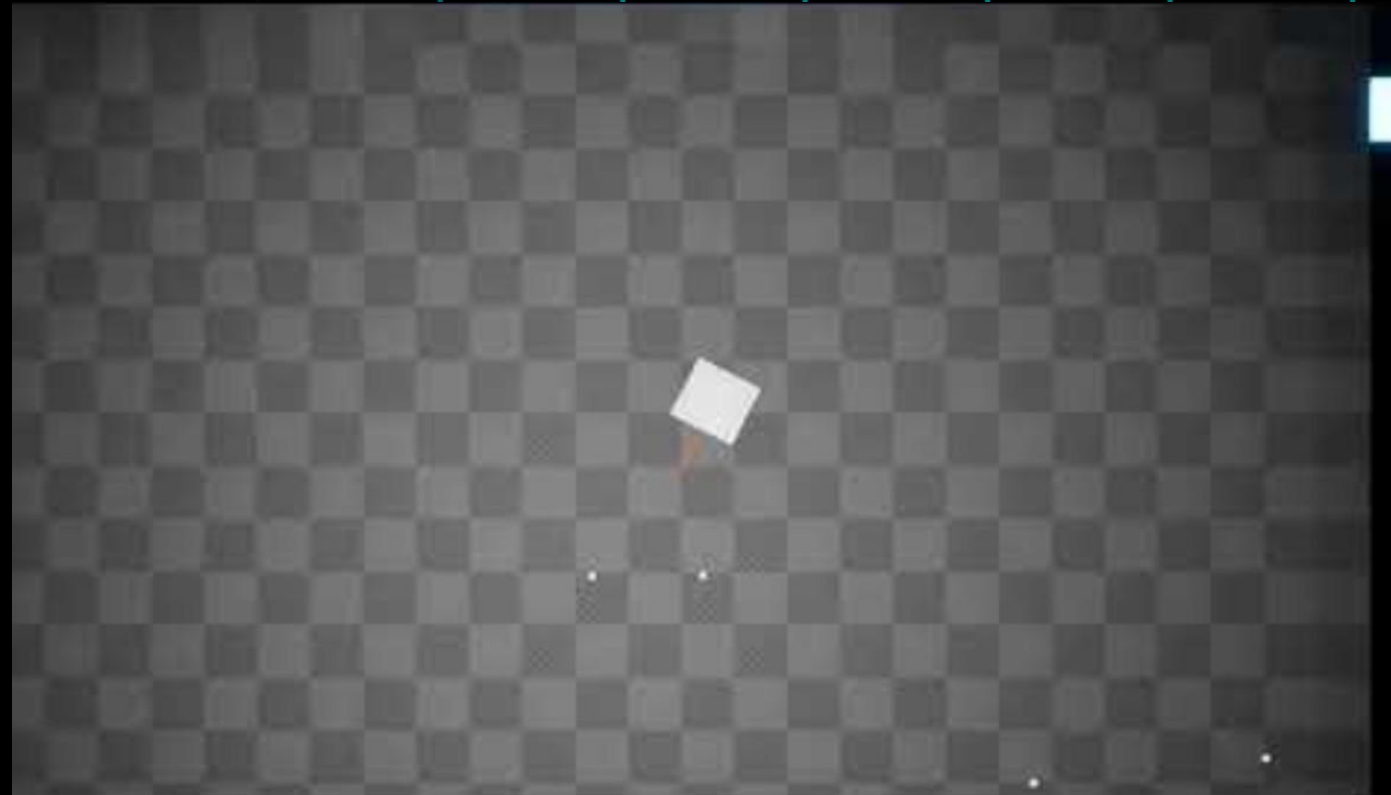
Player Firing

Instead of using a component, I designed the player's firing system through the player's blueprint as functions.

I do this to keep player variables all in one place. It is also something I am more experienced with which allows me to focus on the DDS manager and the systems it requires.



[\[LINK\]](#)



Art Implementation

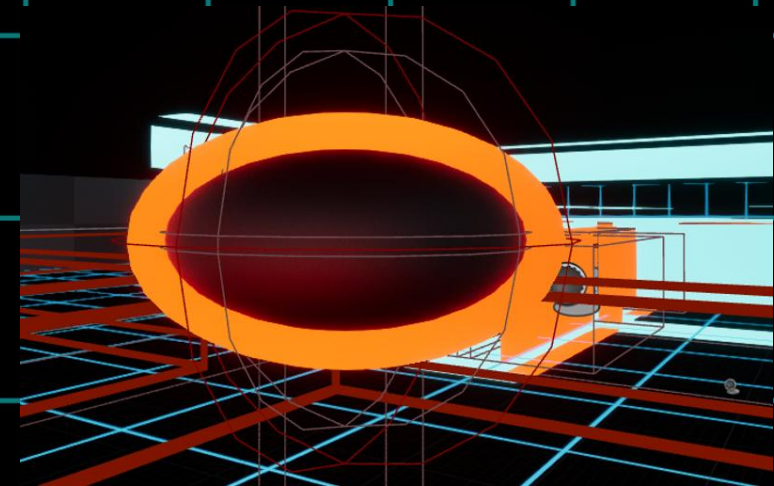
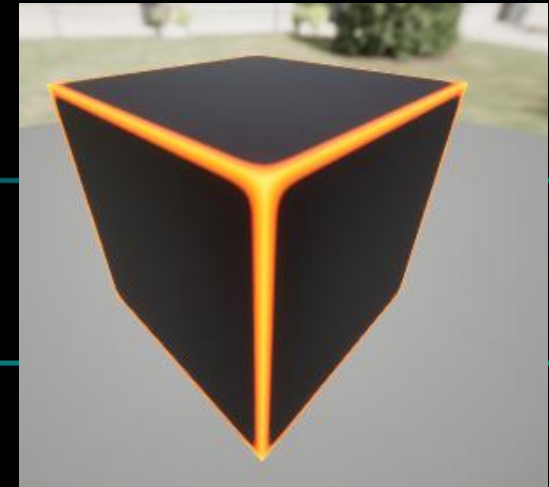
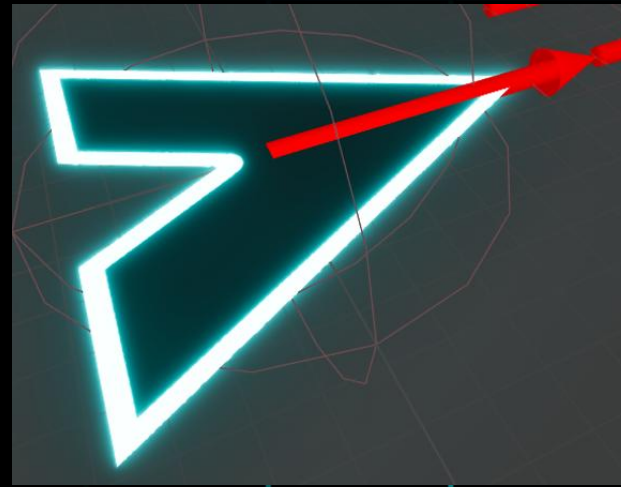
I wanted to test several material types.

The player is a sprite texture with emissive parts.

The power ups are basic cubes with emissive bevels.

The enemies are shapes with overlays that make an emissive border.

I prefer the sprite texture design but yet I like the effect the overlays give to the enemy. They both serve their purpose pretty well, but the power ups are only a place holder.



Game Demo

This demo is what I presented to explain the purpose of my development project.

It includes all the necessary functions:

DDS Rankings

Enemy Spawn Changes

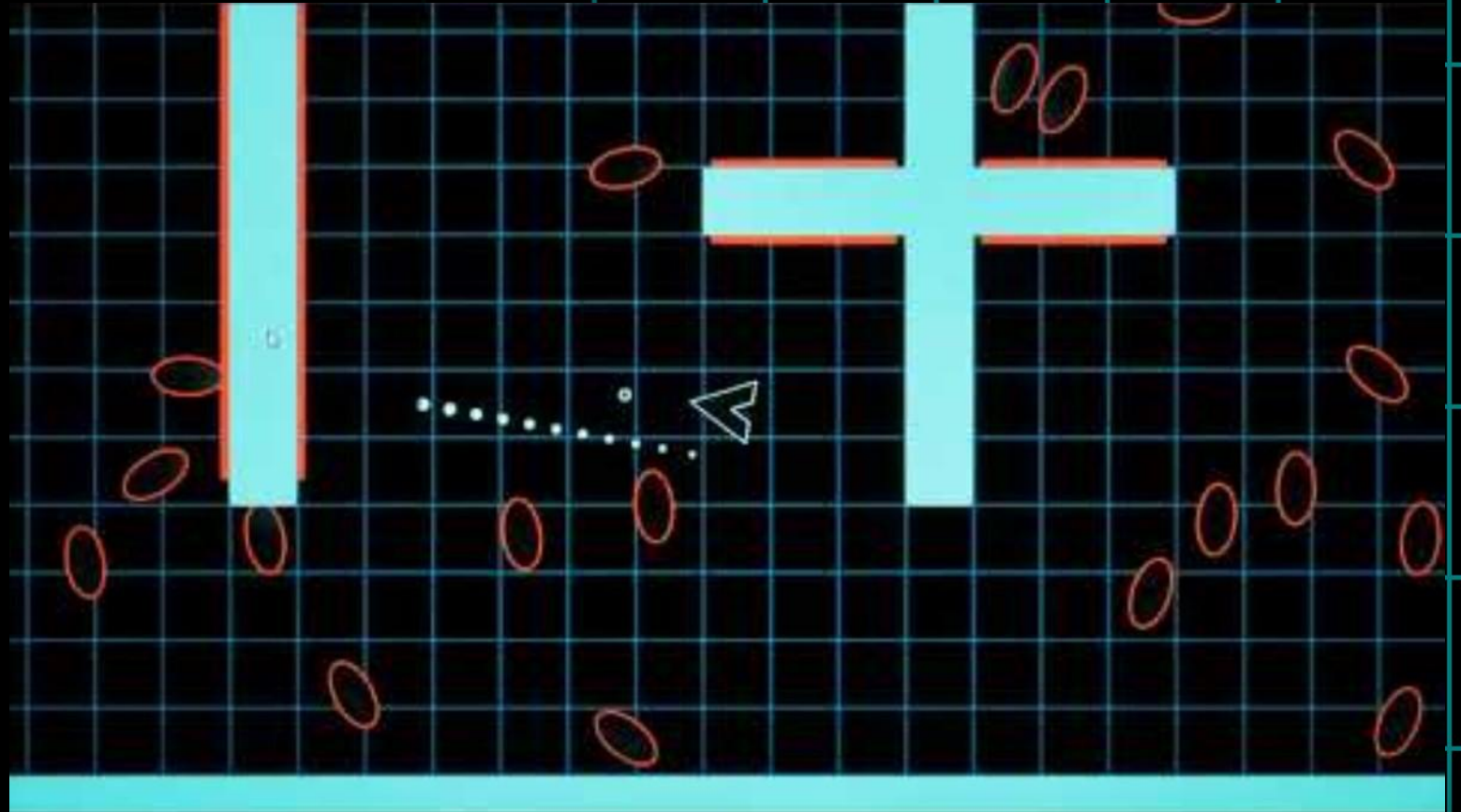
Power Up Spawn

Probabilities

Projectile Deflectors

However, no player health.

[LINK]



Bug Fixes 1

I discovered an issue during personal testing where picking up a power-up matching the current one active, the duration doesn't increase. Fixing this, I made it that the power up duration resets when another power is picked up.

This very same system allows me to also avoid the issue of testers having **2 power ups at once** as it automatically deactivates the previous power up.

During this time, the coding for the Beam and Shockwaves were designed though only in the appearance of traces or collision spheres. This will be hidden behind VFX yet it serves as a good placeholder to allow a tester to see the range of the ability.

“Run Away” System

Starting off with a Launch Character method, I created a new system that made the enemy run away with a better movement flow. This way it doesn't look janky and glitchy, however I did keep a small Launch Character to add a burst of speed to the enemy.

How it works is getting the **forward vector** between the player and the enemy then inverting it to go away from the enemy instead of towards.

I added a **Decorator** to my **AI Behaviour Tree**, allowing me to keep it in the **BT** instead of using On Tick or a Timer node, and to judge whether to use it or not based on a Boolean.

This method has also been used for the **Evasion Enemy**, but I need to adjust my randomisation numbers as they don't spawn as much in the later Difficulty Ranks.

AI Runs Away from Player



Spawn Limits

I discovered an issue, more of an oversight, with the design of my spawning. It correctly works yet during higher ranks, the number of enemies can become quite overwhelming, especially when including the Bunch Enemy which explodes into 4 Small Enemies upon death.

I only needed personal testing to note this as an issue for user experience.

It took a while to discover a fix, resulting a lot of logic testing.

Now, it will only spawn a maximum of 20 enemies upon the field at one time. When that number decreases below 20, it will add in the remaining while keeping under the 20 limit.

UI

Ready for testing, I introduced basic UI into the game and managed to design the timer.

Currently, this timer works as a threshold - clamping the possible lowest rank the further they play. I wanted this timer to be present for testing so I can compare end times and possibly see a correlation with the ranking. If they are all the same time and rank, then those thresholds need to be spread further or maybe I need to overhaul the entire reason for the timer.

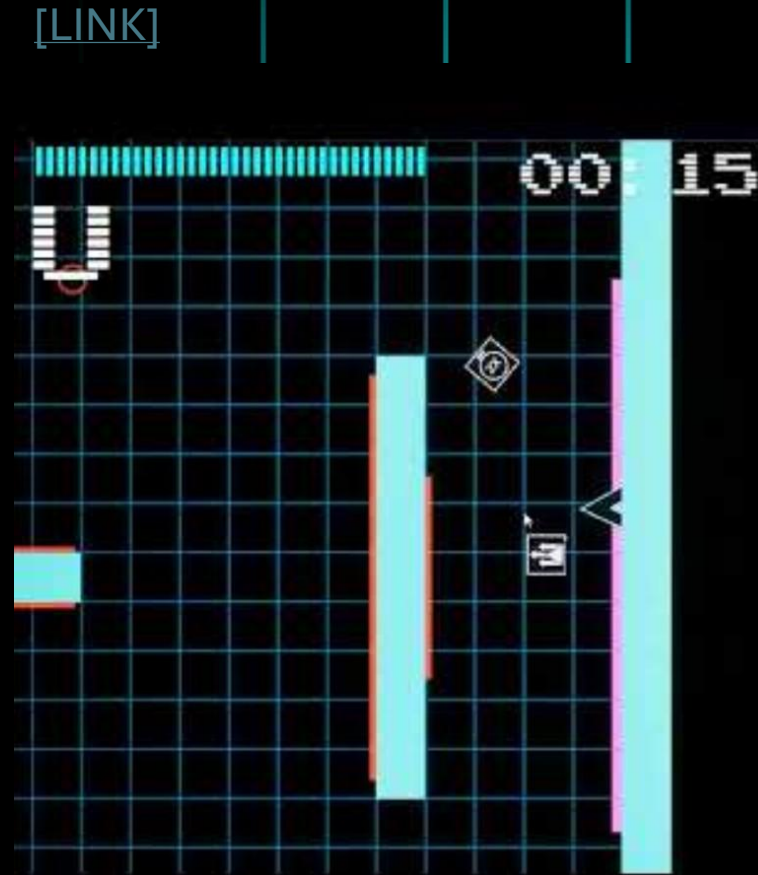
Currently the UI is basic, offering the player the name of their power up just below their rank in place of future icons I will add. Perhaps, I might keep the text for the UI so the player can match the icon with the power up after picking one up.

Rank	Letter
Rank 1	U
Rank 2	G
Rank 3	F
Rank 4	E
Rank 5	D
Rank 6	C
Rank 7	B
Rank 8	A
Rank 9	S
Rank 10	SS
Rank 11	SSS

Teleporters

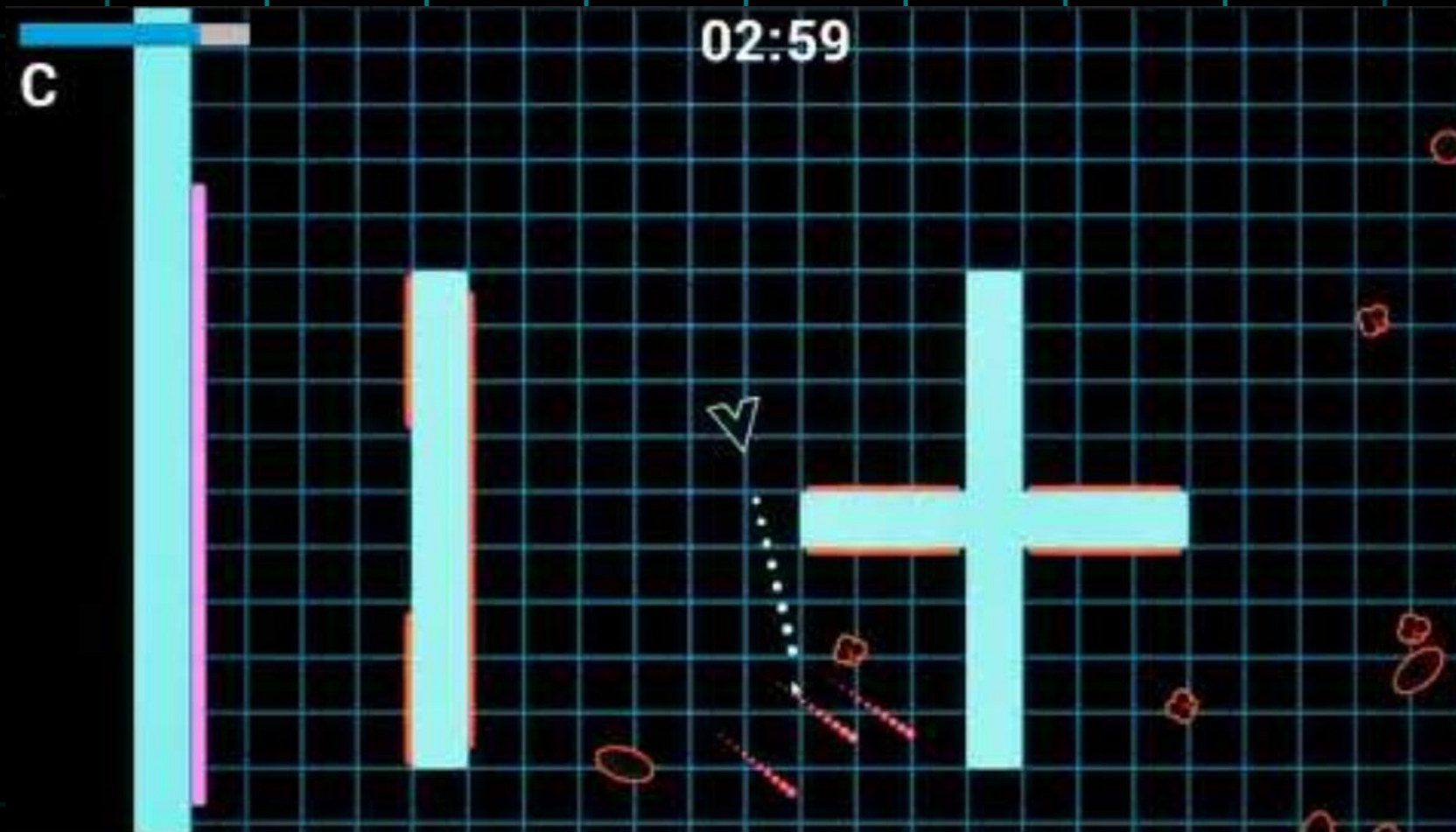
I have managed to introduce a portal for better manoeuvring across the level and to avoid the player being blocked and pincered on the side walls.

I originally intended these portals to have a 1-2 second cooldown to balance gameplay yet I decided to keep them on constantly and allow enemies and projectiles to also pass through them.



Pre-Testing Video

[LINK]



Pre-Testing Issues

Deflectors - I have noticed I need to shrink the collision box which the projectiles overlap. Currently, you can align yourself with the red wall and fire red bullets which bounce off the first object. This is how I coded it to function, but it should only bounce off the red wall which uses that bounce token up.

Health issues - I am unaware of HOW or WHY the enemies seem to have more health than they should have. The enemy that explodes into smaller ones should have **10-20** health depending on the rank and the player's bullets should deal **5 per hit**. As such, it should only take **2-4 hits**. I might reduce this for this enemy but other enemies seem far **too tanky** despite an intended small amount of health. This bizarre concept makes the highest ranks slow and far too annoying.

Enemy's Sometimes Teleporting - This isn't really a bug but a bad design on my part. To create a failsafe to stop the wave system from breaking, the Killwalls used to simply **kill the enemies** that somehow escaped confinement. Now, they **teleport** them to one of **4 locations** which are right in the middle of the + and I instead of being in the corners. As such, they can luckily appear on top or next to the player if they somehow manage to interact with the Killwalls.

Testing

I received a decent amount of testing, being able to compare their times and final ranks.

The rough estimate of a timed playthrough is **6-7ish** minutes and ending in roughly **11 waves**.

The biggest issue is the **health bug** where more health was given to enemies in the higher ranks. The Evasion and Bunch Enemies have roughly **double** the amount of health they should have, and it results in them taking far too long to defeat.

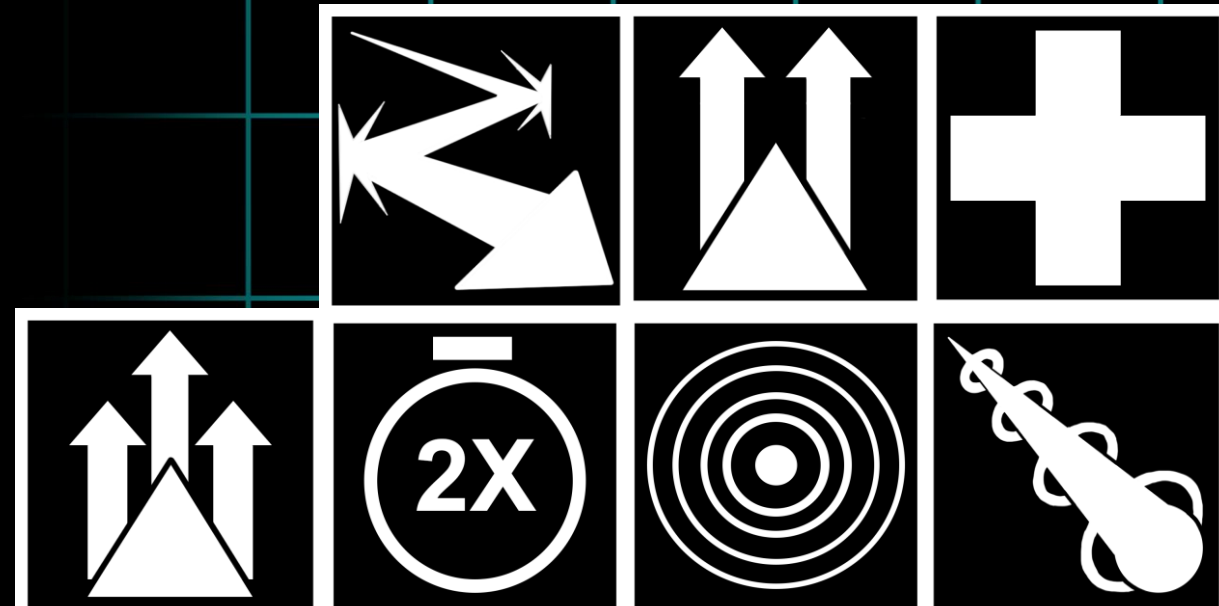
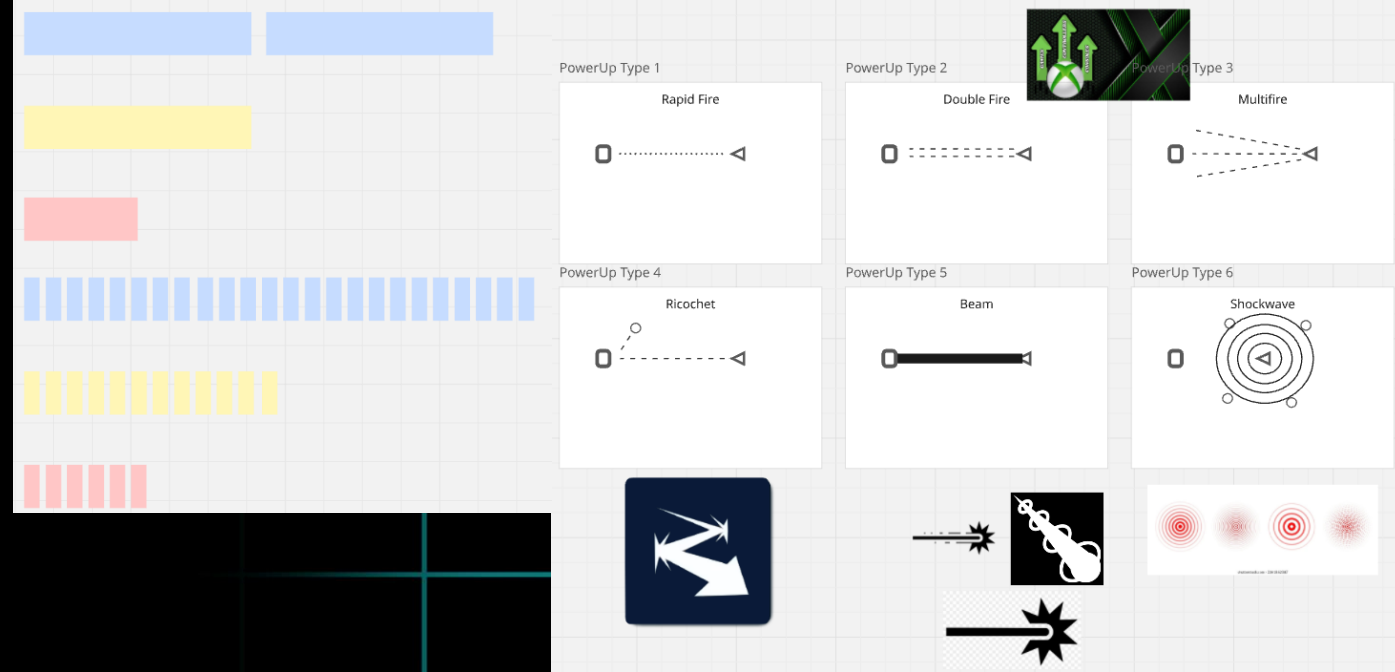
I received a bit of feedback about **UI** changes and **adding visuals** to know what each power-up does. This was already in development

Support items were said to be **too far** for the player. Due to this, I changed their spawn positions to be **around the player** rather than randomly spawning **around the map**.

Art and Icons

From testing and for a step in polish, adding icons and extra visuals in the place of UE5's basic assets, I began designing icons for power ups and styles for the health bar.

I developed the icons in the same way as the character sprite, so the white parts of the asset have emissive materials.



VFX and Sound

I used Niagara effects for my VFX, starting with Directional Burst as the preset for my enemies' explosions, the projectile impacts and also for the teleporters when something passed through.

The beam and shockwave were more complex, needing additional tutorials. The beam required 2 tutorials as the first created the beam but didn't help me add it over my trace. The shockwave only took one.

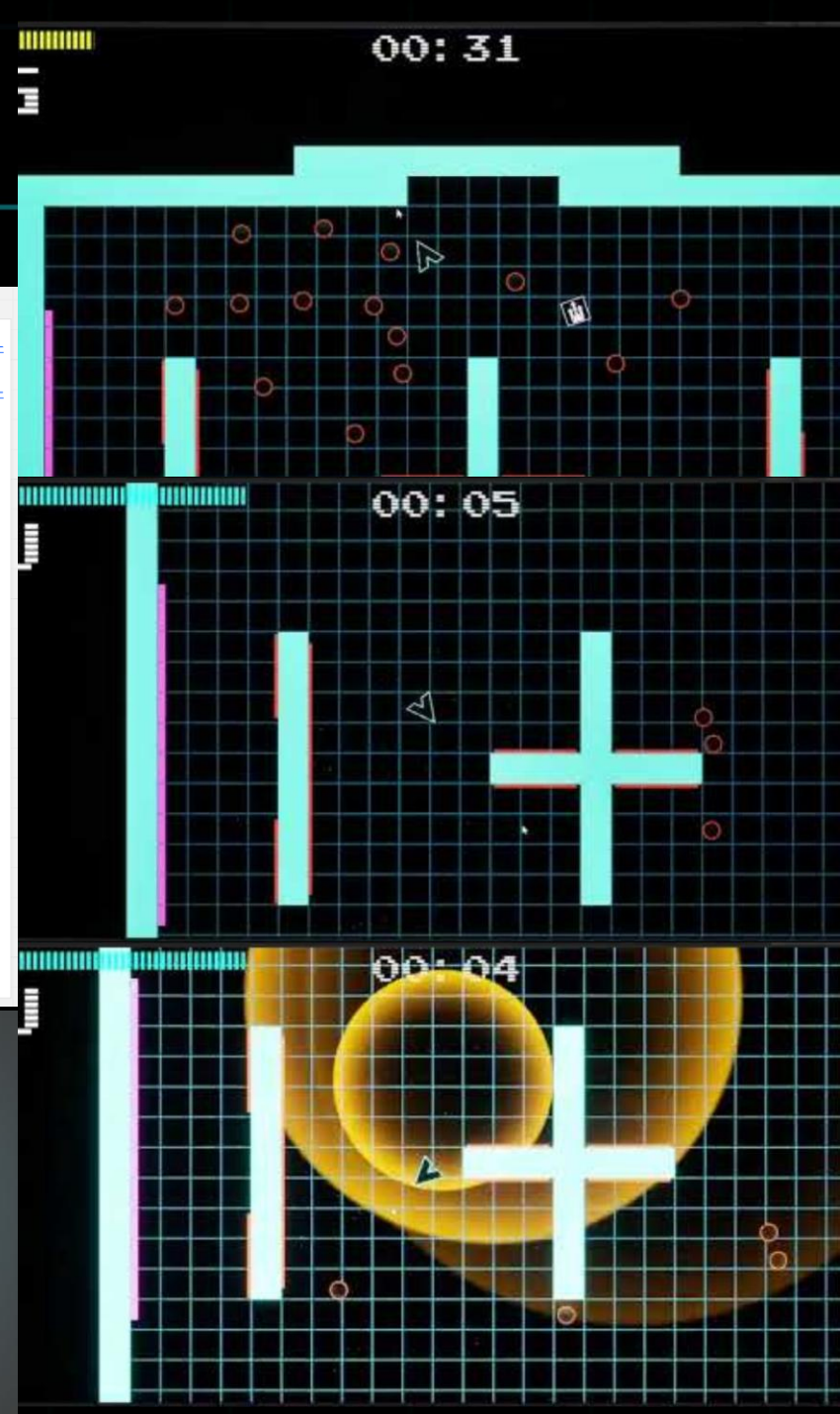
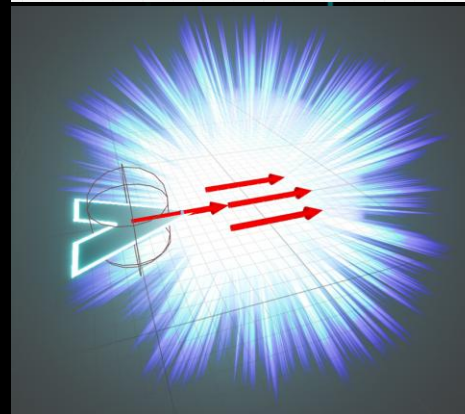
The beam ended up being a constant vfx which I changed the visibility of during gameplay

For sound, all my sources came from Pixabay which I used in Sound Cues.

[LINK] [LINK] [LINK]

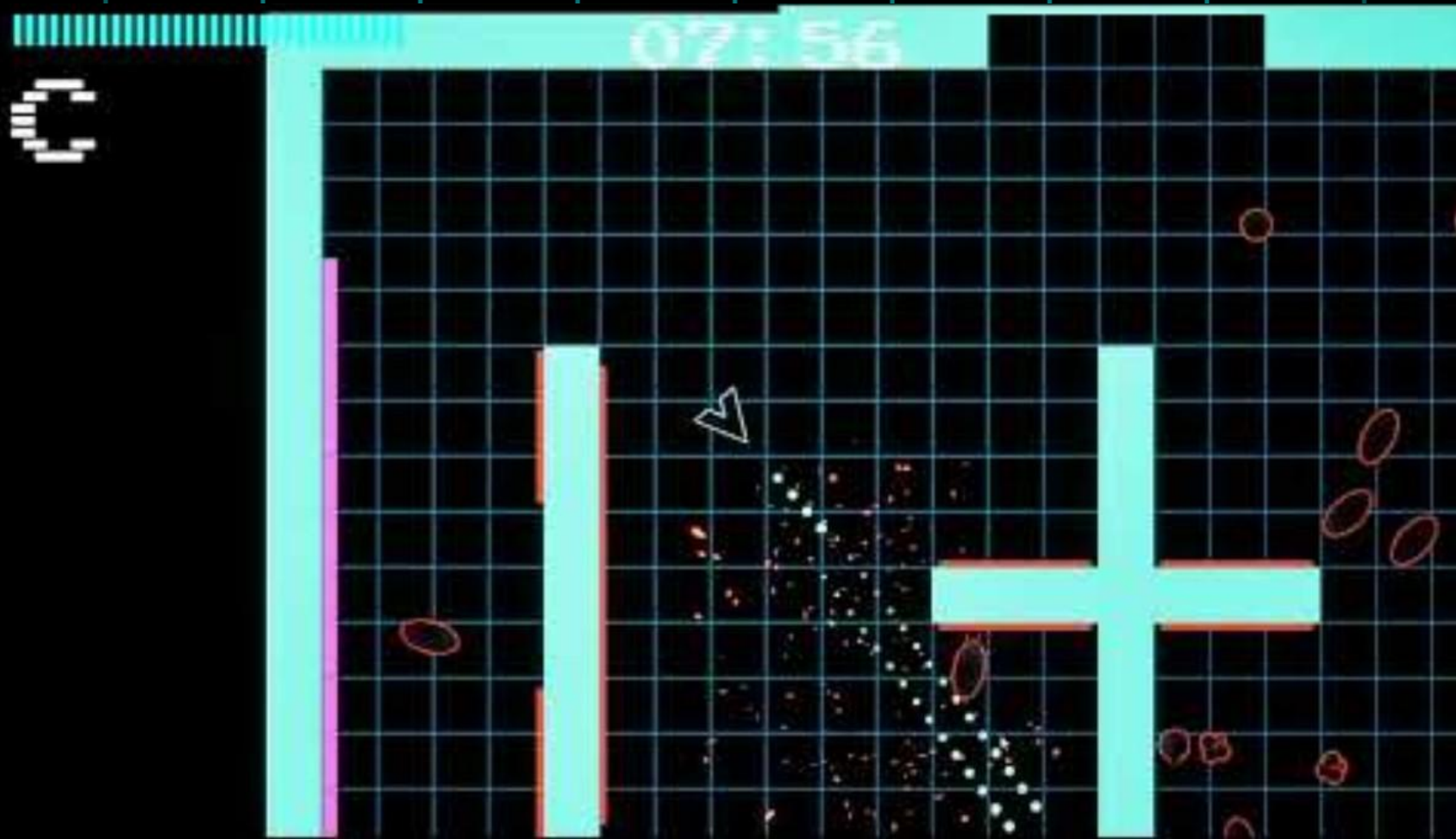
Wall of Sounds

- <https://pixabay.com/sound-effects/film-special-effects-sonar-ping-95840/>
- <https://pixabay.com/sound-effects/film-special-effects-sonar-ping-289721/>
- <https://pixabay.com/sound-effects/zap-hit-474055/>
- <https://pixabay.com/sound-effects/film-special-effects-laser-45816/>
- <https://pixabay.com/sound-effects/film-special-effects-laser-312360/>
- <https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-8-314694/>
- <https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-1-314687/>
- <https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-14-314686/>
- <https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-11-314691/>
- <https://pixabay.com/sound-effects/film-special-effects-zap-c-07-82067/>
- <https://pixabay.com/sound-effects/film-special-effects-heavy-beam-weapon-7052/>
- <https://pixabay.com/sound-effects/film-special-effects-heal-up-39285/>
- <https://pixabay.com/sound-effects/musical-hurt-c-08-102842/>



Final Artefact

[LINK]



EVALUATION

Evaluation

Overall, I am happy with what direction the project took me and the final outcome greatly resembled the concept I had at the beginning. The designs of the assets are simple yet perfect for this arcade-style game and the DDS has come into fruition, allowing me to dynamically manipulate the probabilities of certain features like enemies and power ups.

However, this DDS system can be improved greatly. I understand the tempting reliance on making things difficult with increasing the health of enemies alone. I tried my best to avoid this through only adding a couple extra hits to kill and limited the enemy spawn limit to only 20 at a time.

On the topic of making things difficult, I had thought yet not managed to concept a new method of managing the DDS through the speed of playthrough. Because of this, I realised there are several paths in the DDS I could have investigated.

I think I would have benefited from an extra set of testing to help develop this concept more and it would have given the project the complexity that it needs. I still have a list of stretch goals I had wanted to complete which would have benefited the gameplay greatly.

Compared to when I started, my skills and understanding of the engine has improved greatly and this project has brought me closer to the level of the games that influenced this problem such as Resident Evil 4 and hopefully Left 4 Dead through developing the project further. Beyond just the DDS, my development of gameplay has also reached a better level, and I hope this game experience is only steps away from the road games like Geometry Wars has taken.

I have learnt a lot during this project, such as time and project management, complex blueprinting and I've managed to get a better view on overscoping. I will use everything I have learnt and develop better projects in the future.

FURTHER IMPROVEMENTS

Power Up Spawn Chance

A change I might investigate is to **repurpose the timer**. I was given the idea of using the timer to check the time it takes to kill a certain wave.

If it takes longer, power ups will spawn more frequently, and if it is too quick, the rank will be higher and there will be fewer power ups.

Through this design, the DDS won't just take in a ranking based on the player's progress but also how quick they are – being able to increase the difficulty to slow them down.

Perhaps this design could work together with my DDS Ranks for more than just the power ups yet it might require an entire overhaul of the current design.

This is a similar system that **Left 4 Dead's AI Director** uses, observing the players' progression.

Changing Level Layout

Intentionally, I planned for the level itself to have changes that the DDS would manage.

Whether I switch the layout of the current level layout or make the player move towards the next, I have concepts that could work for this. However, the latter design would require the managers to differ between several sets of spawn points which is another layer to the code.

This design would be nice as it would refresh gameplay. I could always simply start with deflectors and teleporters then build my way up to changing the layout of the level itself.

BIBLIOGRAPHY

Wall of Sound

<https://pixabay.com/sound-effects/film-special-effects-sonar-ping-95840/>

<https://pixabay.com/sound-effects/film-special-effects-sonar-ping-289721/>

<https://pixabay.com/sound-effects/zap-hit-474055/>

<https://pixabay.com/sound-effects/film-special-effects-laser-45816/>

<https://pixabay.com/sound-effects/film-special-effects-laser-312360/>

<https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-8-314694/>

<https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-1-314687/>

<https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-14-314686/>

<https://pixabay.com/sound-effects/film-special-effects-explosion-8-bit-11-314691/>

<https://pixabay.com/sound-effects/film-special-effects-zap-c-07-82067/>

<https://pixabay.com/sound-effects/film-special-effects-heavy-beam-weapon-7052/>

<https://pixabay.com/sound-effects/film-special-effects-heal-up-39285/>

<https://pixabay.com/sound-effects/musical-hurt-c-08-102842/>

Research Materials

- Survival Mode by Wikipedia, the free encyclopedia. [\[LINK\]](#)
- Replayability in Game Design by Josh Bycer. 2018 [\[ARTICLE – LINK\]](#)
- Dynamic Game Difficulty Balancing from Wikipedia, the free encyclopedia [\[LINK\]](#)
- Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review by Mohammad Zohaib. 2018 [\[ARTICLE – LINK\]](#)
- General Overview of DAS in RE4 Remake, Reddit [\[LINK\]](#)
- Game Rank (RE4), Resident Evil Wiki [\[LINK\]](#)

Tutorials

2D Setup [\[LINK\]](#)

3D Top-Down [\[LINK\]](#)

Projectile VFX [\[LINK\]](#)

Basic Enemy AI [\[LINK\]](#)

Cursor Movement [\[LINK\]](#)

Random Spawn Locations [\[LINK\]](#)

VFX Collision [\[LINK\]](#)

Fleeing Enemy [\[LINK\]](#)

Shockwave [\[LINK\]](#)

Emissive 2D Sprites [\[LINK\]](#)

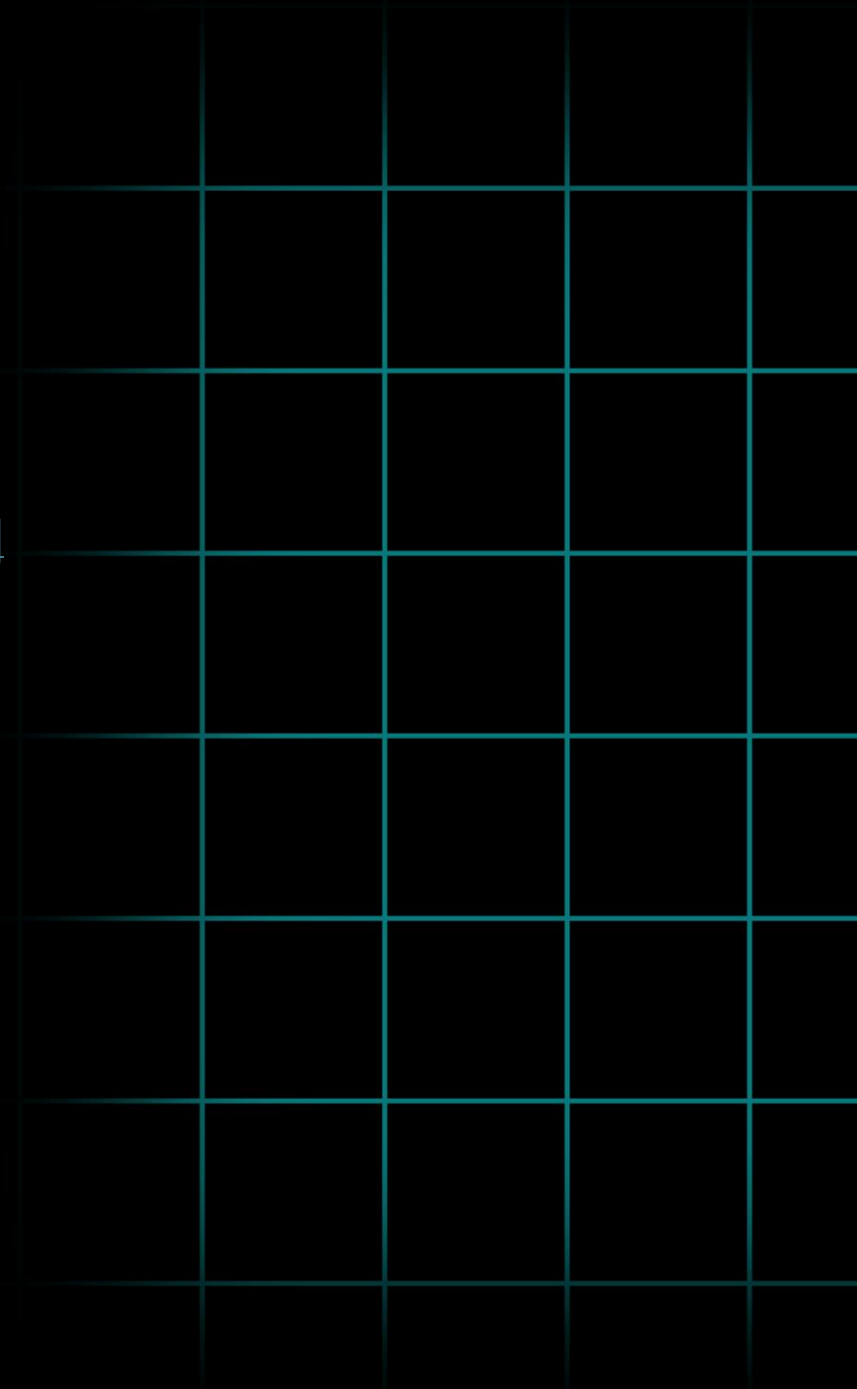
Beam VFX [\[LINK\]](#)

AI Pathing Fix [\[LINK\]](#)

AI Movement Idea [\[LINK\]](#)

Uobjects [\[LINK\]](#)

Font [\[LINK\]](#)

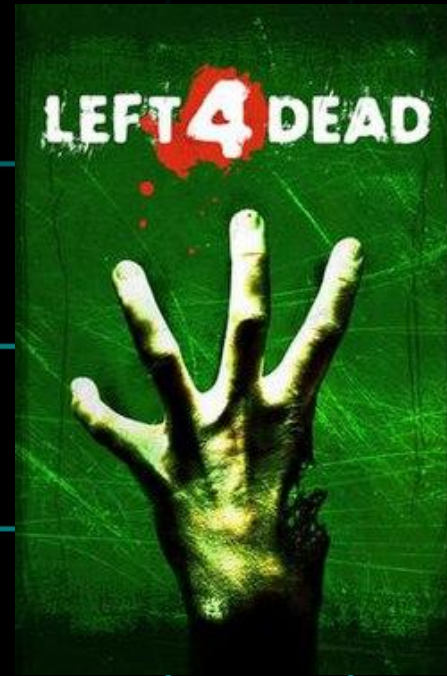


Influences

Resident Evil 4 – Dynamic Adjustment System

Left 4 Dead – AI Director

Geometry Wars





THANK YOU FOR LISTENING

Games Development Project

By Leo Clarke

