



# Games Laboratory SS 2023

**Supervisor: Prof. Rüdiger Westermann**

Team: Gravity Gladiators

Game: Chaos Coaster

### Team members:

Anil Celik Maral,

Erick Lorenzen Meneses,

Lorenzo Russo da Costa Auer,

Matija Jajcinovic,



## Summary of the final results

It all started with a vision of a fun shooter with a lot of chaotic elements. We wanted to incorporate physics as much as possible into the game to make it look and feel realistic and make the player immerse into the action. The idea was to have a gigantic arena in which you are fighting waves of enemies. During the fight there are these unexpected movements which switch up the whole fight and lead to epic Hollywood moments. The main technical achievement we had in mind was to have AI trained enemies, which would learn how to balance themselves in case of direction changes and hits. We wanted the enemies to learn how to walk with missing limbs as well, so the player could cut off limbs and they still would try to chase him.

We are proud to say that we achieved all of our main goals.

The game starts with immediate action and an easy to grasp objective: clearing the waves

of wacky enemies.

It stands out in the amount of fun (physics) interactions:

between the enemies (like the Grunt tipping over a spider), the versatile guns (enemies kicking away by chance your grenades ) and the Erratic Movement of the roller coaster throwing everything around, all generating a highly replayable gaming experience despite the limited setting and action-set of the player.

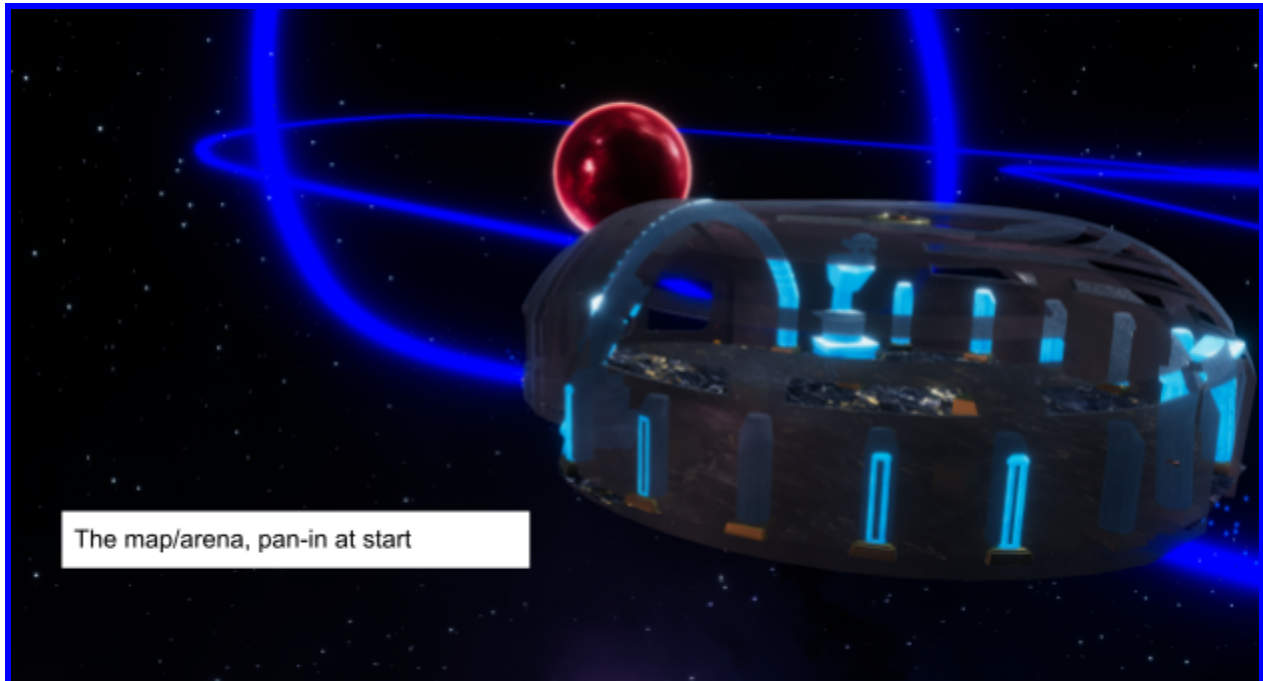
Already in our first playtesting people started to experiment with combos to kill the enemies faster, like shooting the Gravity Gun first to suckt the enemies in and then make a barrage of grenades to quickly clear the wave.

Shooting limbs at (or off) to tip them off balance, quickly became a fun hobby of the gameplay testers and us too.

Our technical achievement is also further boosted by the fact that we were using and experimenting with a technology which is neither well documented nor there are X tutorials to find for easy implementations, most of our big effort went into solving these problems (like navigation done only by Machine Learning).

**Our custom models for the enemies, map, and weapons, custom sound effects round up the unique experience you can enjoy with our game.**

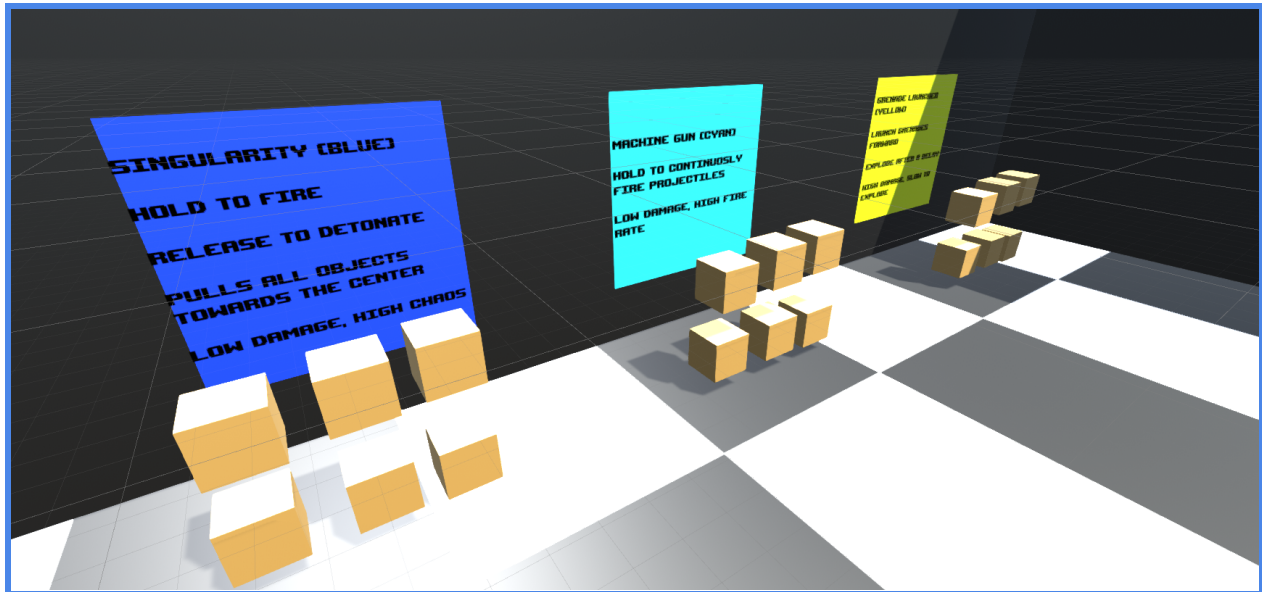




In the following chapters we go into more detail summarizing the evolution of the individual parts of the game.

## Tutorial

The current version of the tutorial was made relatively quickly for the playtesting session and is far from ideal. Our game is fairly complicated and there is no simple way to explain everything quickly, beyond trial and error from the player. Now it is a small sandbox environment with billboards on the walls explaining how the weapons and controls function. It is not extremely interesting and involves a good amount of reading, but it should provide the player with a small environment to mess around and grasp the basics and then figure out the rest. We received some feedback that a more structured tutorial where mechanics are introduced in a linear fashion, like a tutorial level where new weapons are introduced slowly, instead of everything being available from the start. This would be a better way to onboard the player if we had more time to do it.



Tutorial billboards describing the secondary weapons



## Weapons

Our original intent was for each new weapon to be unique and different from each other, not just traditional weapons. In the final build there are 5 different weapons (1 main and 4 secondary), the player can swap between the different secondary weapons and the main weapon is always available to use. There is some good amount of variance between them, the main weapon can be charged and has more force the longer it is held, the grenade

launcher (yellow) projectile bounces around and explodes after a while, the singularity (blue) flies straight ahead slowly and explodes when the button is released, the machine gun (cyan) is more traditional but has slow projectiles that need to be aimed well and finally the pulse shot (red) shoots sticky projectiles that can deal solid damage if they all stick. There were more weapons and unique interactions planned, but we feel the original objective was met.

As for the assets of the weapon, the sounds and the particle effects of the projectiles are not self made, but the models and their animations were made by us. The model itself is fairly low poly, but we gave each weapon unique animations and a part of the model changes to the color of the current weapon.

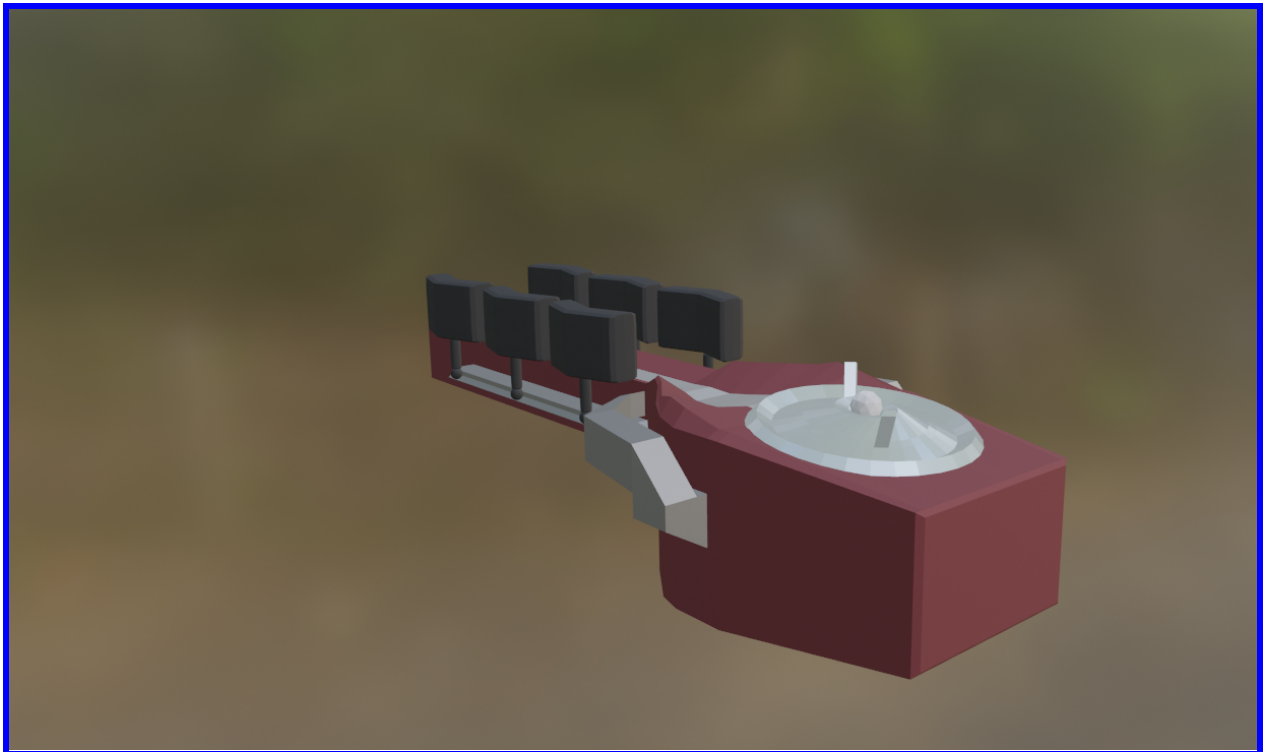


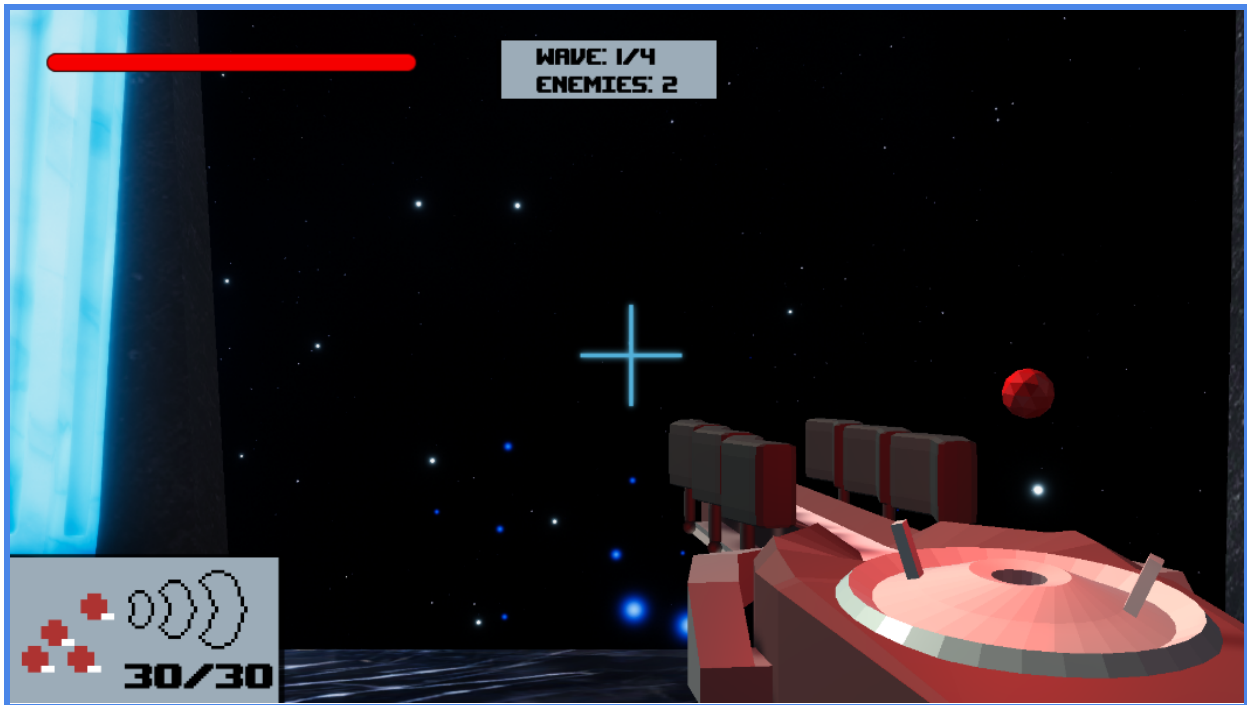
Image of the main weapon on Blender, originally we wanted to have multiple models but due to time (and skill) constraints we ended with just one.

## UI

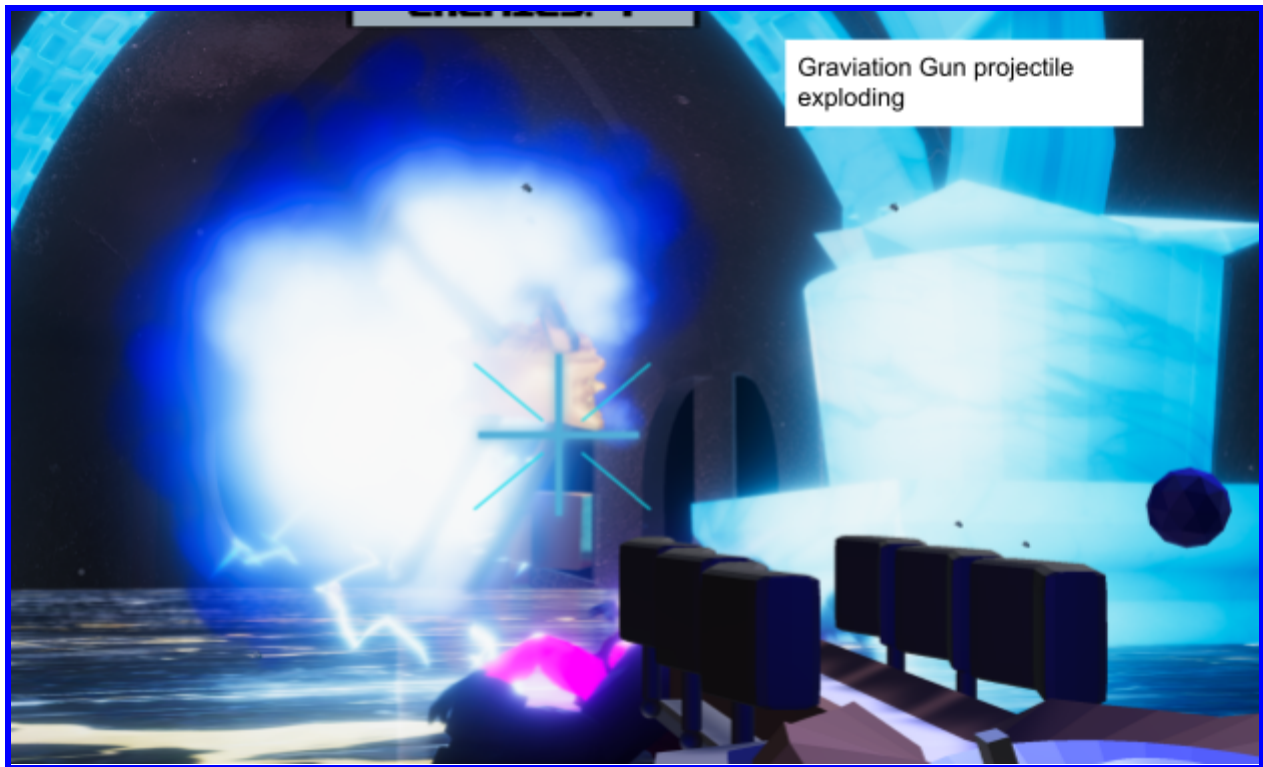
The current UI during gameplay is very basic and focuses on functionality. Creating a more stylized UI that is more cohesive with the art direction of the game would be a fine next step if development continues. For now it shows :

- Which weapon is currently equipped and how much ammo it has left

- The main weapon's current charge level (a gauge that fills up the longer it is held)
- A health bar for the player
- A crosshair that shows feedback when hitting enemies
- An indicator of which wave the player is in and how many enemies are left in that wave
- A warning sign which shows up when the player is approaching a more intense section of the track



The UI of the game



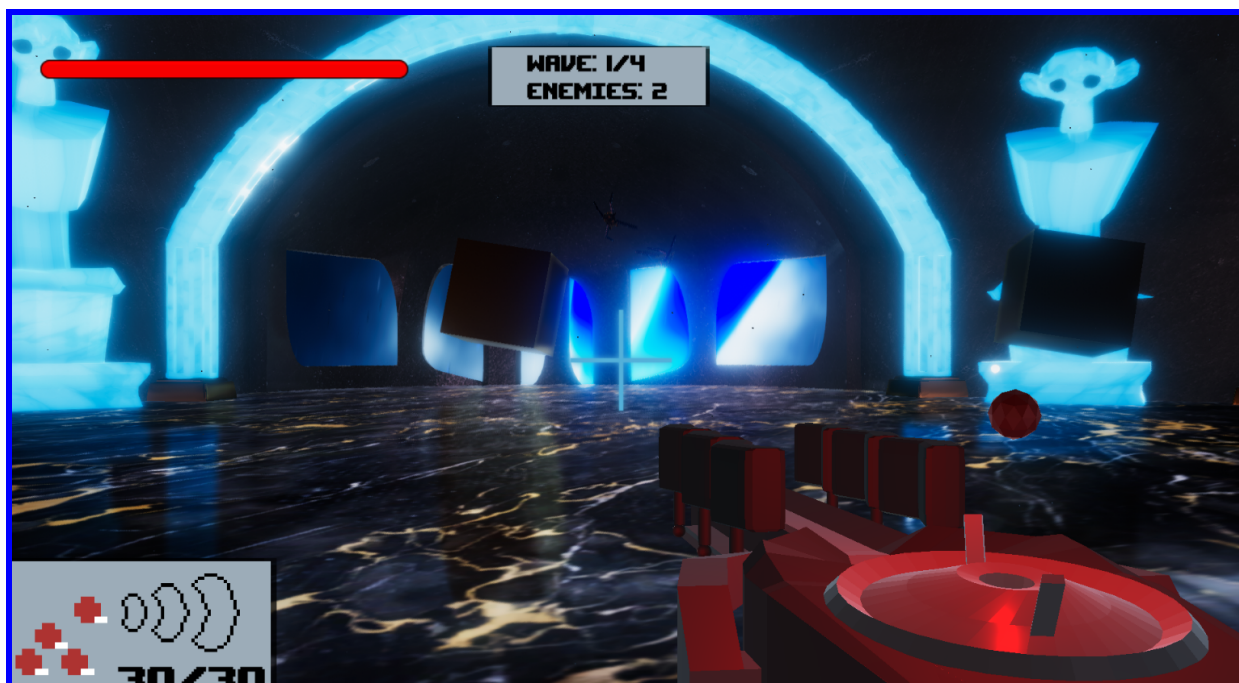
## Movement Alert

Since the beginning we were worried about the arena movement. A lot of mechanics you cannot interact with became frustrating. In order to counteract this we had a few options. Make the player not be affected at all or give him options to move during the movement, with grappling hooks for example or magnetic boots. The option to completely leave the player unaffected by the game mechanics was too boring gameplay wise and adding all those movement options was not possible in the time frame we had. So we decided to at least prepare the player for incoming movement. We played around with a lot of options. We added a train model that mimicked the movement of the arena 5 seconds ahead of it happening. A sound cue and warning signs were also added. After the playtesting we were certain that the train model needed extreme rework. It was hardly seen at all and when somebody stumbled upon it, they were mostly confused. Thus we removed it completely for the final version and only kept the street signs and the audio cue, since those actually helped a little in preparing. For the future we probably would implement some other version of the train model, included in a minimap with the track as context, so players can better grasp what the information is showing. But this would need further testing and a few iterations, so this was not possible to realize for the final release.



## Player Movement

The movement was a particularly difficult point to get right, as the way the map moves and turns makes it really tough to implement. We tried several different ways to do it, but in the end we decided, despite the name of our game, to go for movement which is easy to understand for the player. The camera rotates with the map, which makes it difficult to tell when it rotates but it would have been really confusing to do it otherwise. This part would need more work to truly deliver on the fantasy of fighting in a roller coaster track and the chaos that entails, but for now it is good enough we feel.



The level during a flip, it is hard to tell what is happening beyond the objects flying to the roof.

## The Map and Enemies

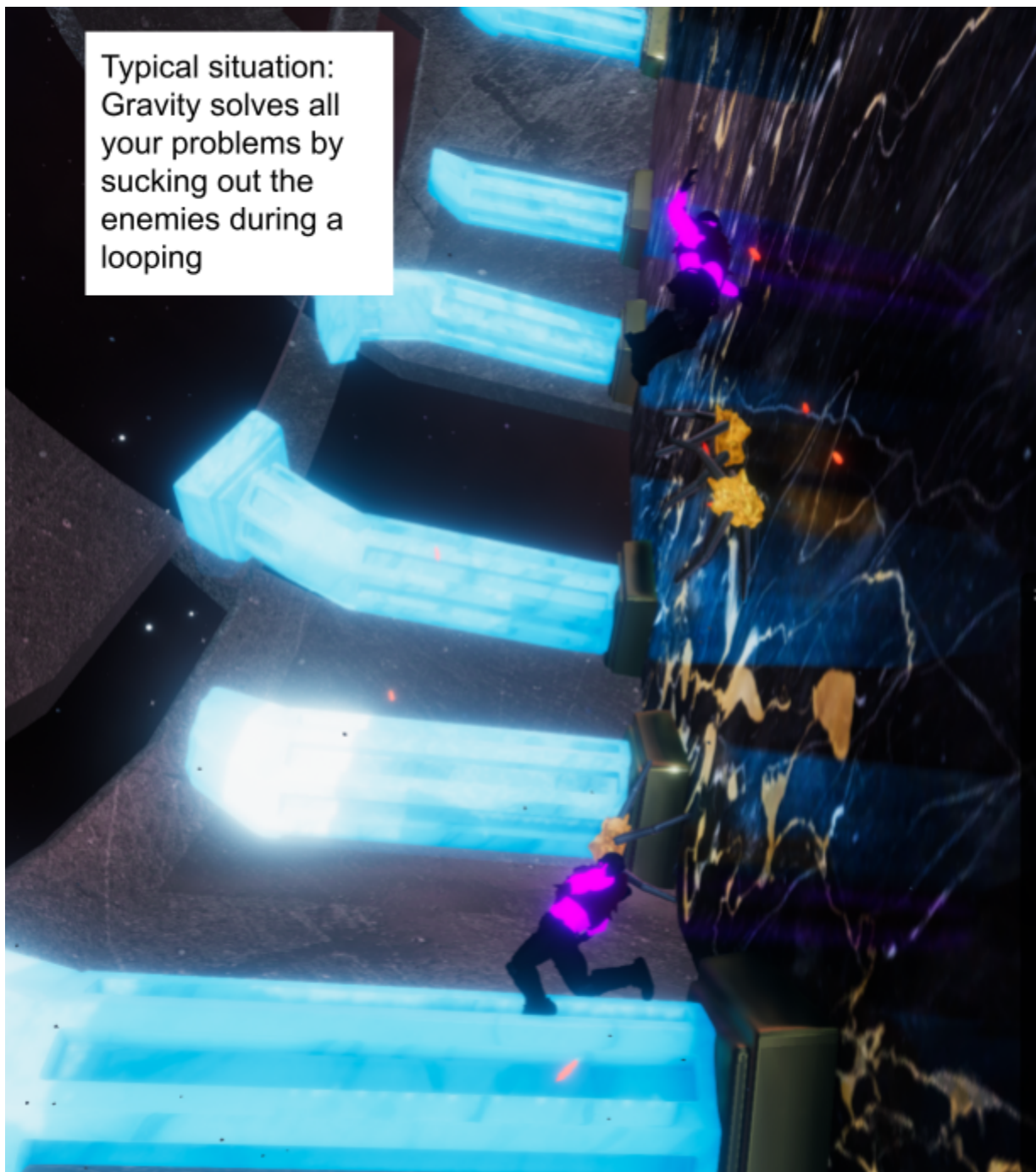
The map is not convex and has obstacles, which the enemies can avoid. They sense it with a voxel sensor and raycasting. The input for the PPO Reinforcement ML are all the poses and physics properties of each limb plus global data like the speed and rotation speed of the map, player position and so on. The movement also works on slopes and with pushing them by the centrifugal forces of the map movement.

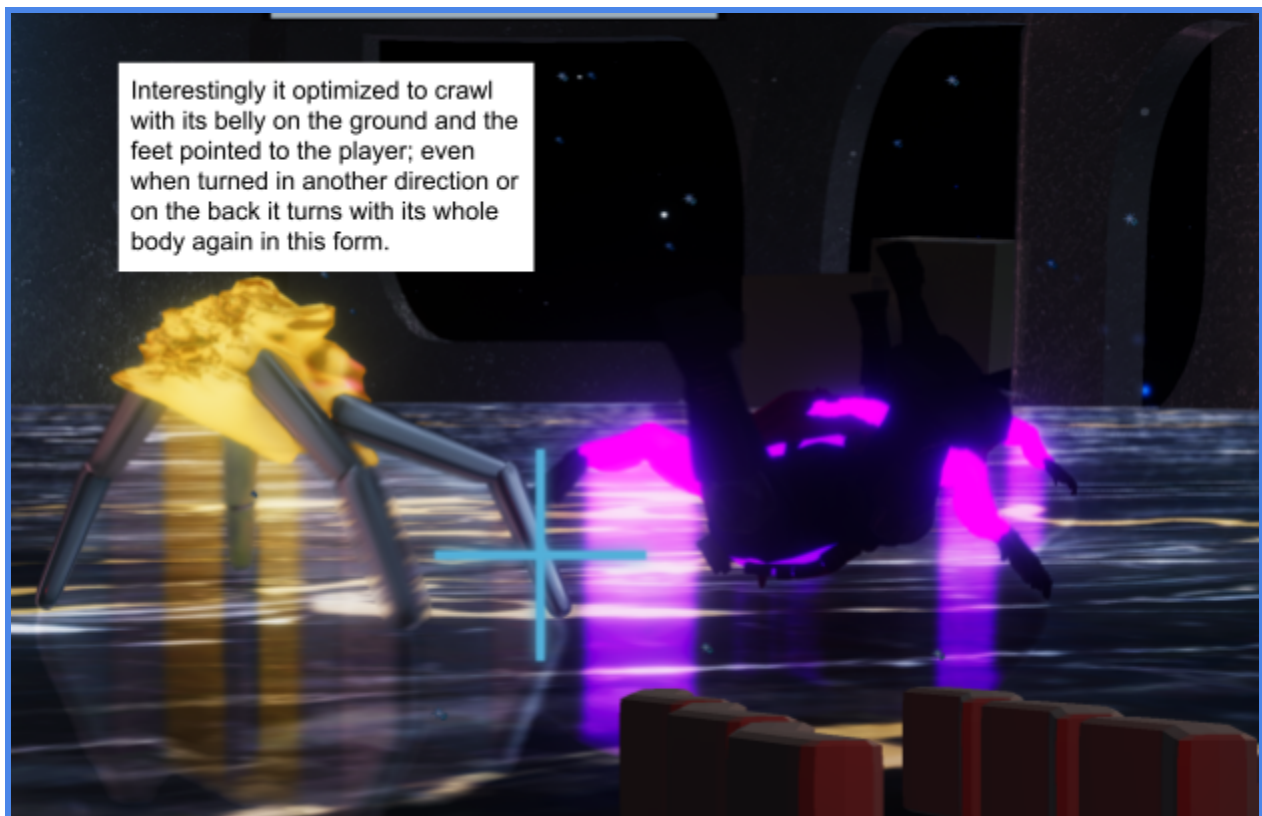
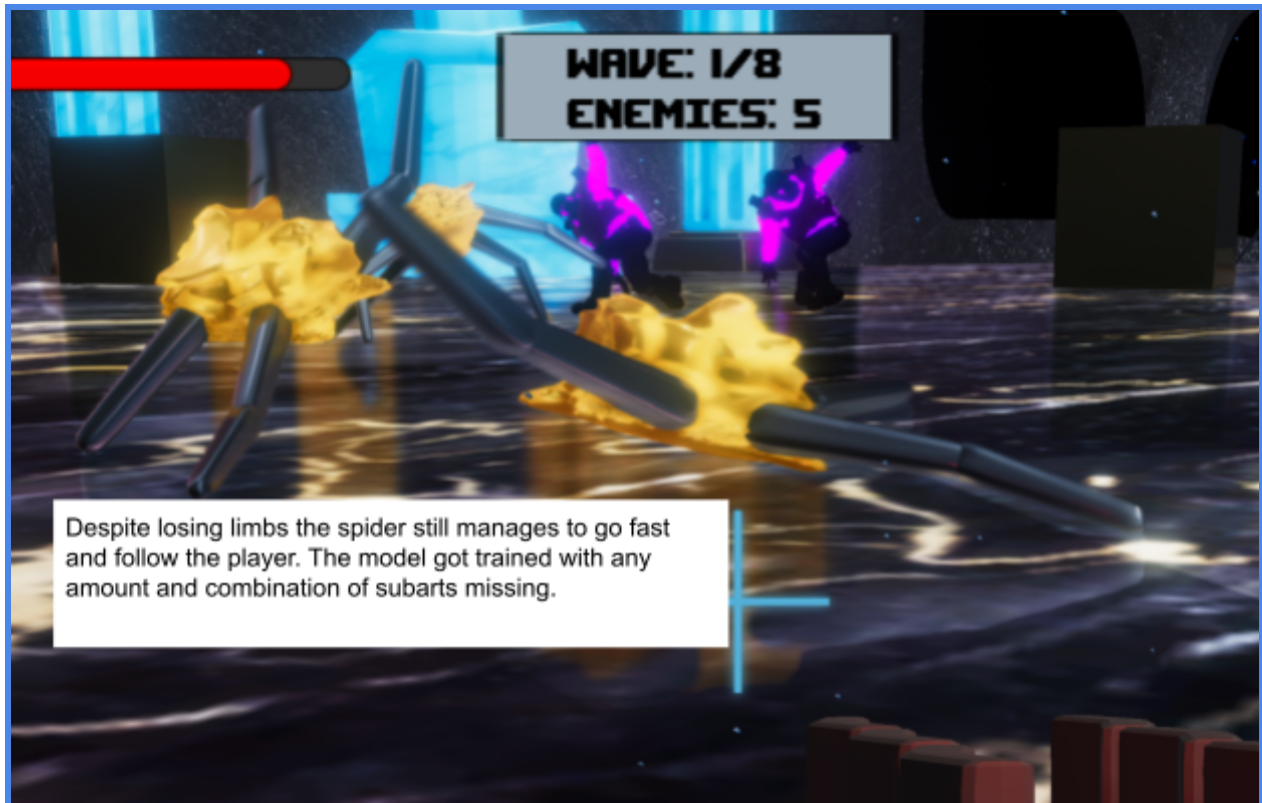
The limb separation is a big feature: the spider has been trained to optimize its speed while having a random chance to miss any combination of limbs.

We couldn't figure out in the limited time how to make the Grunts stand up when falling so they just continue in an Exorcist mode crawling to you.

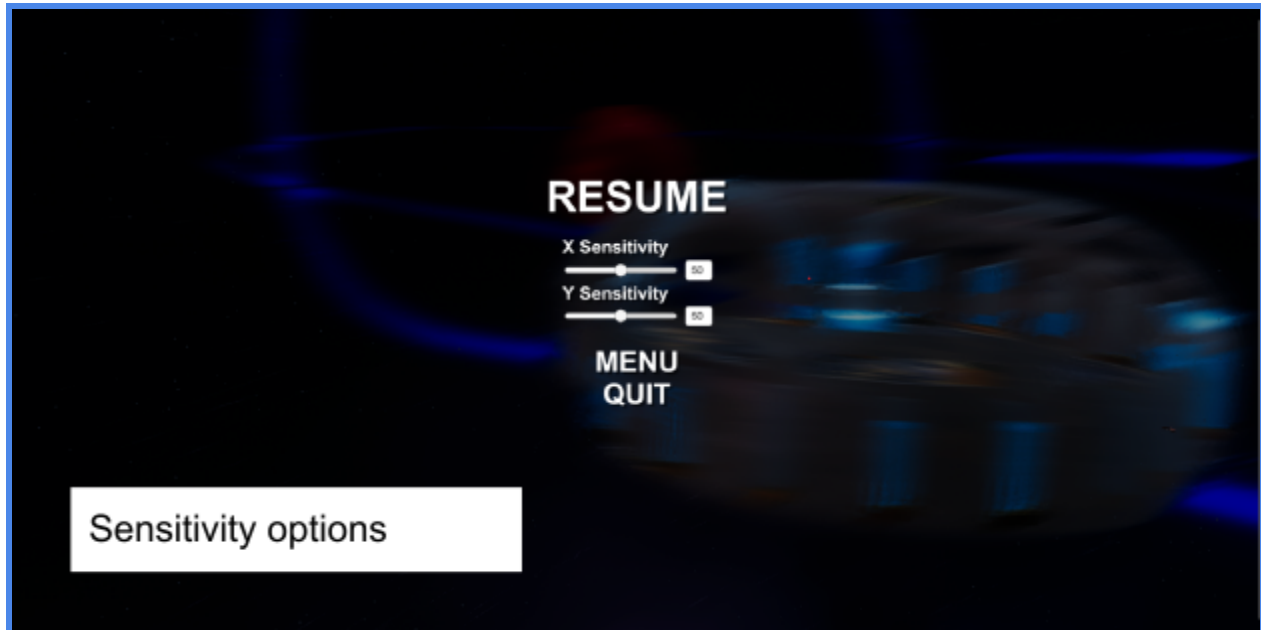


Typical situation:  
Gravity solves all  
your problems by  
sucking out the  
enemies during a  
looping





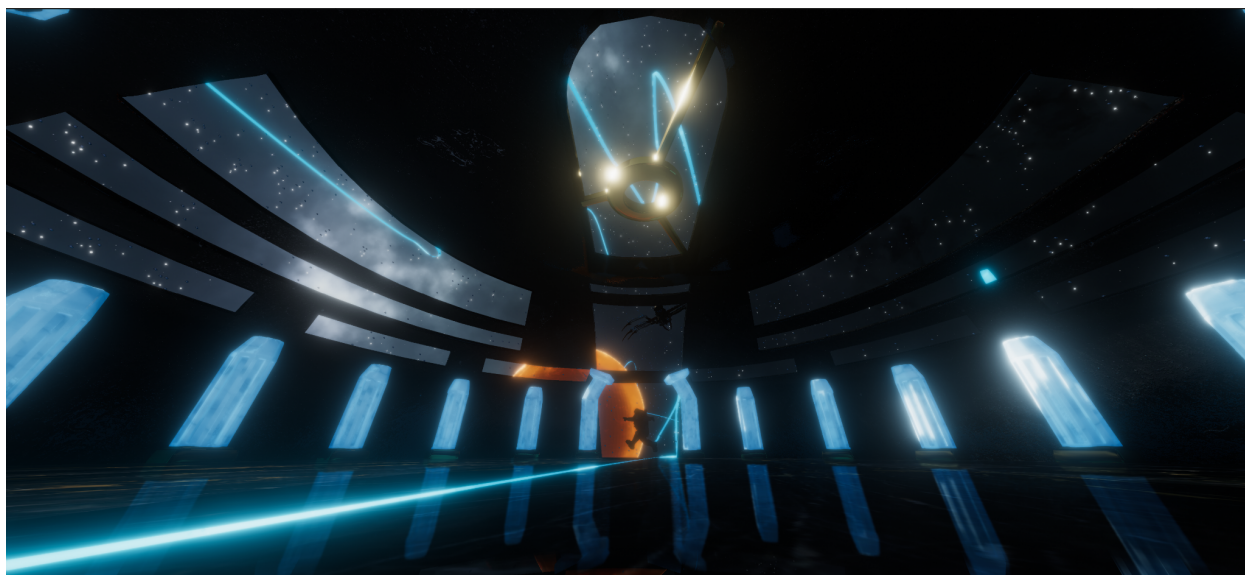
## UI



## Further thoughts about our results

All in all the outcome was close enough to what we imagined. Due to technical difficulties

and unforeseen problems we didn't manage to make it as polished as we would've liked to, but the base idea is incorporated well. Our main technical achievement was also our main difficulty throughout the process, since changes in the arena or the movement meant multiple hours or days of training, to see how well the enemies adapted. Every small bug or misbehavior could only be detected after the training had completed, so we had a lot of time intensive iterations to get to the state we are now. Furthermore the enemy brains started to produce unexpected performance issues, which led to more iterations and a limit of how many enemies can be present at a given time. But all in all the enemies are now moving with a respectable speed towards the player and can definitely pose a threat. Their behavior doesn't completely resemble our visions, but as explained, due to technical and time constraints we were not able to further improve it. Nonetheless there are already other enemy types in the making which barely didn't make the cut into the final game version. The other big issue we had is to incorporate the course theme into the game. In the game idea it was well discussed and obvious that the whole level was a really big roller coaster, but the implementation lacked cues to make the player feel like they were on a roller coaster. The first iterations and during the playtesting most players wouldn't notice that they were in a roller coaster at all. They acknowledged that the movement was resembling a roller coaster movement, but that they were playing inside of a big roller coaster was never in their mind. So we tried to add more visualization to the track and the surroundings to improve the roller coaster feeling.



## Changes made according to feedback from playtesting:

### Graphics:

- + added proper Screen Space-reflections and Post-Processing
- + added glowing Cross Hair
- + added Enemy Hit Crosshair feedback
- + added Enemy Killed Crosshair feedback
- + made arena more transparent to make the track and planet more part of the game
- + recolored the planet to be more visible

### Gameplay:

- + added Grunt Enemy (Humanoid)
- + Improved controls for weapon swapping, now they can cycle forward and backward and use the numbers 1-4 for choosing specific weapons
- + added camera shake when getting hit by an enemy
- + retrained the enemies limbs-off AI

### Sound:

- + selfmade sound for limbs separated
- + sound for getting hit by enemies
- + changed music for main menu and game scene
- + fix machine gun sound bug

### UI:

- + Added an indicator of how many enemies are left in the current wave
- + changed main menu
- + Added sensitivity slider to the pause menu
- + removed train movement indicator since it was only confusing for most players
- + increased the size of the movement warning signs
- + added a credits scene
- + fixed continue button on win screen
- + made main menu buttons more visible

### Tutorial:

- + Fixed typo and changed some explanations to be more accurate
- + Weapon ammo respawns after grabbing it
- + added dummy enemies