

Prototyping Report Group 10

Table of Contents

Prototyping Report Group 10	1
Player Controller Prototype.....	2
Challenges:.....	2
Conclusion:	2
Arena Prototype	2
Challenges:.....	2
Conclusion:	2
Network Prototype	3
Challenges:.....	3
Conclusion:	3
Tron Minigame Prototype	3
Challenges:.....	3
Conclusion:	3
Invasion Mingame Prototype	4
Challenges:.....	4
Conclusion:	4

Player Controller Prototype

Since our game is going to be a racing game you will be driving a car. We want the car movement to feel natural and realistic.

Challenges:

Creating a player controller proved to be quite difficult. The first prototype could only move a cube, using forces we tried to create a natural feel of acceleration and deceleration. But using just forces isn't enough we also want mimic the suspension of a car but this cannot be done using a single body. After a lot of research we found examples using wheel colliders so we decided to use those in the prototype. The prototype isn't finished and there is a lot that still needs tweaking, but we now have a good base to start from.

Conclusion:

Creating a natural feel isn't easy and creating the player controller for the actual game will take a lot of time, but we think it is definitely worth it.

Arena Prototype

The arena needs to be procedurally generated and change during gameplay.

Challenges:

We started off prototyping the arena by trying to create a script that procedurally changes the height map and thus changing the terrain. This was hard so we tried a different approach, combining multiple terrains and let them slide over in each other.

This didn't look smooth and was of course far from ideal so we decided to give up on the smooth terrain and wrote a script to generate a circular, kind of blocky arena by adjusting the height map. We want to make the arena change size during the game but rendering was too slow to be able to change the entire terrain at runtime.

We therefore decided to focus on the procedural generation of assets in the arena which works fine by now. Objects disappear and appear at random locations and orientations within in the arena and they check whether the spot they want to be placed is empty before the placement happens.

On the side we managed to create a wall of fire to place in the arena using a particle effect.

Conclusion:

Creating a procedurally generated arena has some difficulties but we will focus on elements that work for the arena in the actual game.

Network Prototype

Our game will be a LAN multiplayer game. The network prototype was made to find out how we can deal with the challenges we encounter in making a multiplayer game.

The network prototype allows a host to create a game using the create button. Other people can join this game by typing the host's IP in the IP box and click the find button. In game each person has his own player which they can move (blue cube for the host, red cube for clients) and a 3rd person camera that follow this player. When someone enters a collider placed a bit ahead of player spawn point an enemy is spawned (with 50 hp), this enemy is controlled by the server and follows the nearest player. Players can shoot using the spacebar and kill the enemy (shots deal 10 damage).

Challenges:

One of the first problems we encountered was giving each player their own camera, but we managed to solve this problem by giving each player their own camera as a child and enable it on spawn. Later on we had trouble trying to smooth the movement of the players you are viewing. We solved this problem by using the Vector3.Lerp function.

Another problem was shooting, first we used ray cast but it was hard to hit moving enemies since you can see the ray. We solved this by using a particle system instead of the ray cast.

Conclusion:

Based on this prototype we think we can make a good LAN multiplayer game.

Tron Minigame Prototype

The Tron minigame is a minigame in which the players create a trail that damages/kills players on touch.

Challenges:

The most challenging part was trying to rendering a single smooth 'tunnel' between points and update this render depending on the player's movement. Getting the mesh right took a few attempts, first we tried to add width and height to y and x coordinates, but this only worked in specific directions. In the second attempt we tried to render the mesh by calculating the cross vector for the width which resulted in 'gaps' when turning. In the last attempt we calculate the proper vectors for every point and rendering between those points, so we would never render to the last point but only the before last point. Finally we fixed the light rendering by removing unneeded vertices.

Conclusion:

The trail turned out really good in the prototype so we will include this minigame in the actual game.

Invasion Minigame Prototype

Invasion is a minigame in which multiple enemies spawn and go after the players when they get within a certain range, the players earn points for killing the enemies.

Challenges:

One of the most important parts of this minigame is the AI of the enemies, they must follow the players when they get within a certain range. So this became the most important part for this prototype. We managed to get the enemies to follow to players and this worked really well.

Conclusion:

The invasion minigame prototype worked well so we will include this in the actual game.