Hao Fang
Andi Grozdani
Amani Mitchell
Jordan Williams
Game Design 2.5D Project
Campus Racing Game 2.5D

Game Design Documentation

Purpose:

In this project., we still chose the racing game as our main gameplay mode. Firstly, we finished a 2D racing game in project 1, so we can base the original map and car models to update the game, what is more, the racing game is very appropriate for 2.5D game design. Therefore, we re-started to think about our own game idea and decide to create a racing game which relate to our daily life. In order to use the project1 materials, we still set up the theme as "Campus Racing", and we renamed it "Campus Racing II". We considered that our players are mainly from UMASS Dartmouth students and we know many of them want to drive at a high speed or maybe want to drift on the campus. But doing so would be against the laws and campus rules. Consequently, our game's feature is to simulate real racing on campus by inserting the UMASS Dartmouth campus map and 3D building models. Meanwhile, the purpose of this design and game is to give a better gaming-racing experience to our players and try to meet their requirements and expectations when they hear Campus Racing II!

Procedures:

After the idea set up, we started to turn generalities into and create this game step by step. Initially, following the professor's tutorial and requirements, we still be required to develop this 2.5 D game by the platform of "Unity". In Unity, we need to use a car' model for matching our subject (Racing game), after the model done, inserted the C# script codes to implement car's movement. Furthermore, due to our game is a racing game and it relates to background and map design, we use the "Blender" to implement these design features and insert the map and track to Unity. Next, we established some buildings and trees inside the map based on the UMASS Dartmouth that make the scene looks more realistic using free assets packs found online. Finally, we set up the timer for recording the time of each lap and imported the music and "coins" to provide better gaming experience to players.

Car Design:

For the car's design, firstly, we imported a vehicle's model from website, and call it "raceCarGreen". In "raceCarGreen", there are four 3D object: body, WheelColliders, Wheels and CamerStable.The parts of Body, WheelColliders and Wheels compose the entire car's model, after we inserted four spheres and place them into corresponding positions (coordinates) as the car's wheels. For implementing 2.5D gaming experience, we change the player's perspective, and the top-down camera is not useful for this game. Therefore, we create a "CameraStable" to replace the top-down camera for tracking car's movement in real-time. After that, we created a cube and scaled model it into a car.
For implementing new controller way (keyboard rather than mouse) ,we add a C# script file calls "CarMovement.cs". In the function "FixedUpdate()" to determine that car's turn and the speed.
After that, we went back to Unity and modified some attributes of operation and acceleration to make it control easily. For the car skin, we found a race car model online and we decided to use it because it best fit our game. We have multiple color options for the car color, which includes white, orange, red and green. For our game we decided to go with the orange color because it stood out in the map.

Map Design:

For the map design we wanted to keep the same track design which is based on UMASS Dartmouth campus. The track is the same but the design of the map looks much cleaner and better. First thing you will notice is that coins are now floating while spinning in the air and when you hit the coins they will added to your score and be destroyed. The dorms and facility buildings are now in 3D which looks much when viewed in 2D. We added a billboard with text to show which dorms is which. We also added the famous windmill on campus to the map. We added proper collision to the buildings and tree so when the player hits the tree or building it will stop the car. When the user drives to the parking lots, the parking lots will actually look like a parking lot because we added custom made texture to it. And finally the last design element we added to the game was visible checkpoints which are red round arch to make the player goes through the checkpoint. If the player doesn't go through the checkpoints, you will not be able to finish the game.

Game's Menu:

In the games menu we added an options tab, which will give you options like the master volume and the screen resolution. We believe that this was a nice option to have before playing the game. For our last game we did not have any of theses options so for our 2.5D game we wanted to make sure we had them. This was done by creating a script called menuManager.cs where all of the settings can be handled through the script. Once that script was done we were able to add screen resolution checkboxes and the master volume slider controls link them to the actual menu. For the master volume we had to create audioManager.cs script using a tutorial we found on youtube. This enabled us to link the background music with the master volume slider so we can control the games master volume.

Collision:

For the collision detection we wanted to have accurate enough hitboxes such that the player does not lose immersion but also not too accurate to the point where it affects system performance. As such, different forms of collision were used for different objects. The two types of colliders used were box and mesh. Box collision has faster performance, but can only be in cuboid shapes. Mesh collision, on the other hand, matches the 3D model it is attached to at the cost of some performance.

Things close to the track, like the checkpoint arches, used the slower mesh collider since the player is more likely to crash into them compared to objects far from the course. The trees all used a box collider for their hitbox, since it is unlikely the player will ever get high enough to touch the leaves and the detail for the tree trunks' collision was less important. The barrier walls used box colliders for a similar reason; as they are on the edge of the map the player should not be interacting with it too often. The hitbox also extends extremely high vertically so the player cannot escape. Due to both of these objects being simpler and not requiring as much detail, we opted with the higher performance option. The Dell buildings used a box collider as well, since they were already cuboid shaped on the bottom-half of their structure. In contrast, the Chestnut buildings had to use a mesh collider since some of them are angled to be parallel to the course, and the box collider does not support angles. The larger buildings in the middle of campus used a mixture of box and mesh colliders. This was so the player could drive underneath the roof that extends past the rest of the building, but the grooved door on the front of the building's model caused awkward physics, so we put a box collider to cover that so it acts like a flat wall. Lastly, the billboards were given a mesh collider so that players could drive underneath them in between their legs. The remaining miscellaneous objects were given mesh colliders since their numbers were few and unlikely to significantly affect performance.

Before adding collision, we had to force all the trees (per type) to the same Z-height as the car was hitting some trees, but sliding under others. This removed some of the nice aesthetic variation that the trees had, but was necessary to not ruin immersion and let players drive directly through the trees.

Sources/Links Used To Help Make Game:

START MENU in Unity
https://www.youtube.com/watch?v=zc8ac_qUXQY

How to make delayed start in Unity 2D game? Pause before start in Unity game.
https://www.youtube.com/watch?v=3MlauoiahvI

How to create a simple coin counter in your Unity game? Easy Unity 2D tutorial.
https://www.youtube.com/watch?v=-EIXQHxoicg

How to make coin pick up feature in your Unity 2D arcade game with sound effect | Unity 2D tutorial
https://www.youtube.com/watch?v=lNXDK8a75l0

Low Poly Forest Pack
https://jaks.itch.io/lowpolyforestpack?download

Kenney Racing Kit
https://www.kenney.nl/assets/racing-kit

Countdown Timer In Unity - Easy Beginners Tutorial/ Guide
https://www.youtube.com/watch?v=o0j7PdU88a4

https://www.youtube.com/watch?v=ZZD1cg8xDsI – MiniMap Tutorial

https://www.youtube.com/watch?v=pAsCXXsuB1M - Top Down Tutorial

https://www.youtube.com/watch?v=BKCsH8mQ-lM - Music Tutorial

https://www.youtube.com/watch?v=0Quv9U9_a8c - Car Physics

Camera tracking:

https://www.youtube.com/watch?v=ehDRTdRGd1w

Windmill object:

https://www.turbosquid.com/FullPreview/Index.cfm/ID/642388

How To Make A Driving & Racing Game For FREE - Unity Tutorial #01 - BEGINNER BASICS
https://www.youtube.com/watch?v=MQ5GJPlAGS4&list=PLZ1b66Z1KFKgkE9ji0tF2iDO0LGxmlwIm

Unity 2017 Beginner Tutorial - Making a spinning coin
https://www.youtube.com/watch?v=s1opSKwApLU
Create a menu for any game

https://www.youtube.com/watch?v=EA-tBcTxE8M

PAUSE MENU in Unity:
https://www.youtube.com/watch?v=JivuXdrIHK0&t=524s