Group L Project

Introduction:

Our group project is a Festival Shopping Experience set in a virtual environment, where players can interact with NPCs and purchase items from different shops. We chose this project because it allows for creativity in designing various items using Blender, and festivals are known for their vibrant and diverse stalls and attractions. To ensure an efficient and effective workflow, we divided the tasks based on each team member's strengths. Three members focused on creating the 3D models while the other two focused on implementing the project using Godot game engine. As an additional feature, we included a user interface (UI) for players to interact with the NPCs scattered throughout the pier. We believe that this will enhance the immersive experience of our Festival Shopping project, and make it more enjoyable for players.

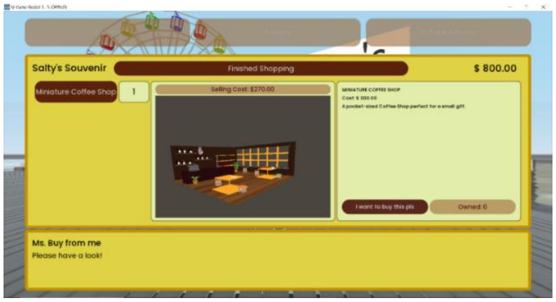
- ## Lariza Julia Bucao Rodrigo:

Role: Building Interactions, Locomotion, and Individual Systems.

Interactions: Shop System (Figure1), View point of character (Figure 2), and NPC variables (Figure3)

I have added a previously made Dialogue System and NPC. I made a classic RPG shopping system that activates whenever the shopkeeper boolean was turned on when NPC is instanced. I used previous sprites .png files from my Final Year Project (FYP) along with scripts. I tested the new features through a third person first before testing it for locomotion.

Figure 1



Salty's Souvenir

Finished Shopping

$ 800.00

Miniature Coffee Shop 1

Selling Cost: $270.00

MINIATURE COFFEE SHOP
Cost $ 300.00
A pocket-sized Coffee Shop perfect for a small gift

I want to buy this pls

Owned: 0

Ms. Buy from me
Please have a look!

Figure 2

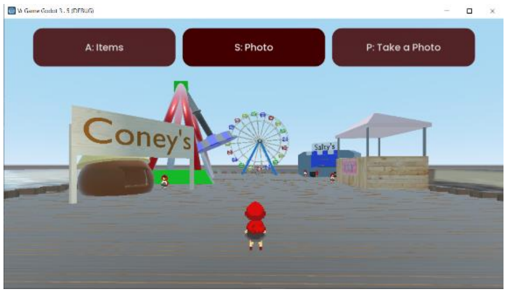
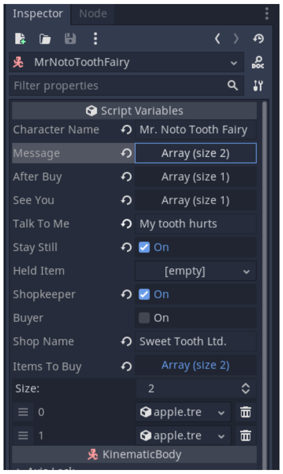


Figure 3



Inspector Node

MrNotoToothFairy

Filter properties

Script Variables

Character Name	Mr. Noto Tooth Fairy
Message	Array (size 2)
After Buy	Array (size 1)
See You	Array (size 1)
Talk To Me	My tooth hurts
Stay Still	☑ On
Held Item	[empty]
Shopkeeper	☑ On
Buyer	□ On
Shop Name	Sweet Tooth Ltd.
Items To Buy	Array (size 2)
Size:	2
0	apple.tre
1	apple.tre

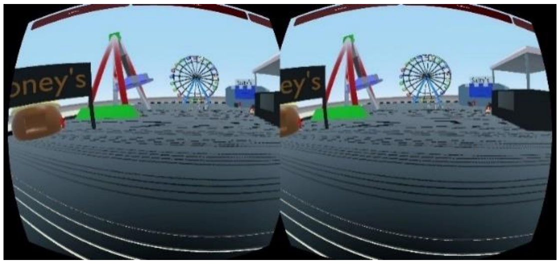
KinematicBody

Inventory System: Saved Item Resource (Figure4)

I used a premade Inventory System from my FYP. This uses Global.gd and Player.gd (AutoLoad scripts) to update the inventory on the UI.

The individual items show their image using a Viewport node, which has a child node with the item's PackedScene. Items are a custom-made Resource, containing the item name, prices, descriptions, scenes etc.

Locomotion on android (Figure5)



The android version for the app had a problem on the second update of the project due to large objects being imported to the folder. My other teammates reduced the file sizes but the android version lighting still did not work. After various testing, we found out that the Environment variable skies were not shown in the android export, resulting in a black environment but with interactable objects. We used a HDR sky clouds which solved the issue and the scene looked clearer.

I used the CS423 W10 VRPlayerLocomotion.gd script to make the Locomotion movement. I added the player into the scene made by my other team members and tested it on my Android Samsung. In the APK, I found out that the Photo System I made didn't work. I used buttons in the game to make the screen touchable. It worked, but it was hard to use during testing without any controllers.

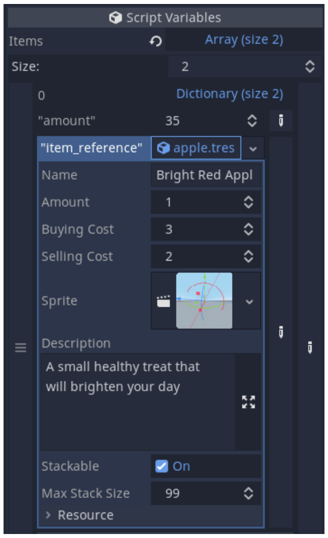
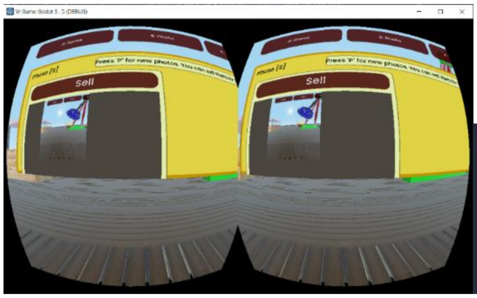
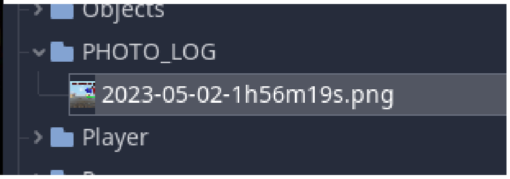

Script Variables
Items: Array (size 2)
Size: 2
0: Dictionary (size 2)
"amount": 35
"item_reference": apple.tres
Name: Bright Red Appl
Amount: 1
Buying Cost: 3
Selling Cost: 2
Sprite
Description: A small healthy treat that will brighten your day
Stackable: ☑ On
Max Stack Size: 99
Resource

Figure6 and Figure7: On Game Engine (photo works with file path created shown)





Item Resource Variables (Figure4)

- ## Abderahman Haouit:

Work Done: Ferris Wheel, Roller Coaster, and Hammer Ride.

Steps:

Ferris Wheel (Figure1)

First, a circle was added and turned into a face by pressing 'f'. The face was then inset by pressing 'I' twice, with the inner circle selected. A wireframe modifier was added and left unchanged so that it could be rotated by 90 degrees and duplicated. Two cylinders were added, one fitting inside the hole in the middle and the other with a mirror modifier whose object was set to the first cylinder. All modifiers were applied, and everything was joined by pressing 'ctrl+j'. A cube was added and rotated by 25 degrees, with a mirror modifier whose object was set to the wheel. Two more mirror modifiers were added, and the legs were mirrored, and a plane was added. A cylinder was also added and scaled, and the top and bottom faces were scaled down on the X and Y axes. Finally, the cabins were manually placed and parented to the wheel using a constraint called limit rotation.

Roller Coaster (Figure 2)

To make the rail for the roller coaster, two small cylinders, two big cylinders, and a cube were added and edited in edit mode to create a segment of the railing. Everything was joined together, and a circle was added using 'shift+a'. An array modifier and curve modifier were added to the segment of the railing and the array count was increased to make the railing longer. The curve object was set to the circle, and the deform axis was set to Z to make the railing bend along the curve. The route of the roller coaster was edited by scaling the circle on one axis and increasing the array count. The points were subdivided and moved to create the desired shape. Finally, a cube was added, and a "follow path" constraint was added to make the cart move along the rails.

Hammer Ride (Figure 3)

To create a Hammer ride, a cylinder was added and scaled on the Z-axis. The bottom face was extruded a few times using E to extrude, S to scale, and G to move. One of the bottom edges was clicked while holding Alt, Shift+D was used to duplicate it, and S+Shift+Z was used to scale it up. This process was repeated, and two loop cuts were added using Alt+R. Two lower faces were selected and right-clicked to choose "bridge faces." This process was repeated with every other set of faces. One more loop cut was added from the outside, and E and S+Shift+Z were clicked to add a cube. One face was selected and extruded, and another cube was added and treated in the same way.

Figure 1

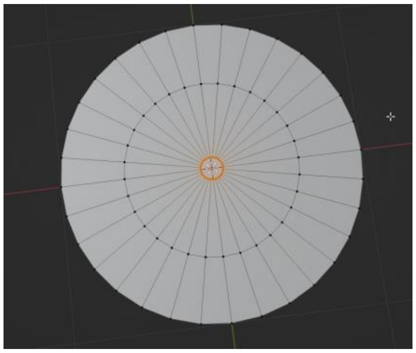


Figure 2

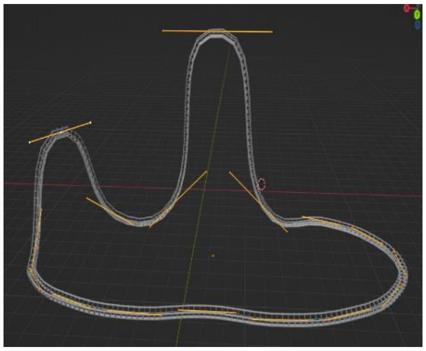
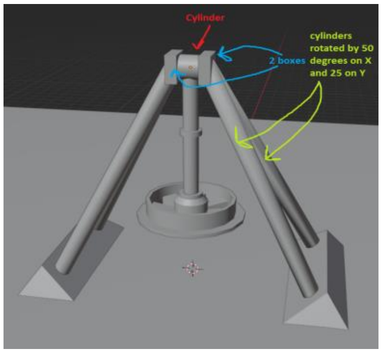


Figure 3



• Richard Ogbonnaya:

Work Done: Pier scene, WaterPlane scene, Dirt scene

Steps:

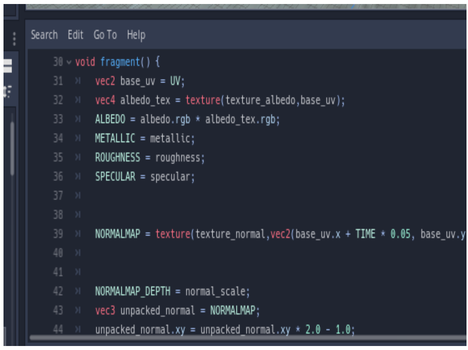
WaterPlane: Begin with a planer mesh created with Godot 3.5. To begin creating the water effect, we add a shader material to our plane. We now have a playground with this shader material to alter the water look, such as adding refraction components to the water, roughness, UV1 Scale adjusting the x and y to 10, texture refraction, and texture Normal.

Once we get the general desired look we then convert this shader into a GD.script to add waves to our planer in contrast of time.
In our void fragment() function Between the Specular and NORMAL_DEPTH values we add a line of code which gets us these waves which correspond with time in the x and y direction.

“ NORMALMAP = texture(texture_normal,vec2(base_uv.x + TIME * 0.05, base_uv.y) * 1.2).rgb * texture(texture_normal,vec2(base_uv.x + TIME * -0.05, base_uv.y)).rgb; “.

Earth/Sand: Again, for our sand we used a planer mesh following the exact transform of the water to almost overlap but act as a seabed for our WaterPlane. A shader was used to edit material to get our basic look. Which is depicted in the above image.

Pier: The last foundation object was the pier this was created in blender. It starts with a plane mesh which is scaled down to a rectangle. An array modifier is then added to this mesh we change the Y factor to 1.1 to give a gap in our plank look the count is then changed to the desired amount of how many planks is need for the boardwalk, this is then scaled and reposition to the cursor for further editing. While in edit mode we change to face selection where we will begin to select every 5th or 4th face in our X view, this will be the legs of our pier. This is first extruded in to give a nice square gap and then saved then we select those same faces to extrude back to its normal position the leaves a square attached to our plank leaving a face in the y direction which is extruded to give us our legs for the pier. We also extrude the top faces to give more detail for the legs in the surface of the pier. A mirror modifier is then added to get these legs on the other side of our pier. A structural mesh box was added and scaled carefully to act as a stability block for each side of the inner legs. Pier was the copied and some faces were deleted to give the connected pier look.



```
void fragment() {
	vec2 base_uv = UV;
	vec4 albedo_tex = texture(texture_albedo,base_uv);
	ALBEDO = albedo.rgb * albedo_tex.rgb;
	METALLIC = metallic;
	ROUGHNESS = roughness;
	SPECULAR = specular;


	NORMALMAP = texture(texture_normal,vec2(base_uv.x + TIME * 0.05, base_uv.y


	NORMALMAP_DEPTH = normal_scale;
	vec3 unpacked_normal = NORMALMAP;
	unpacked_normal.xy = unpacked_normal.xy * 2.0 - 1.0;
```

Figure 1: GDscript

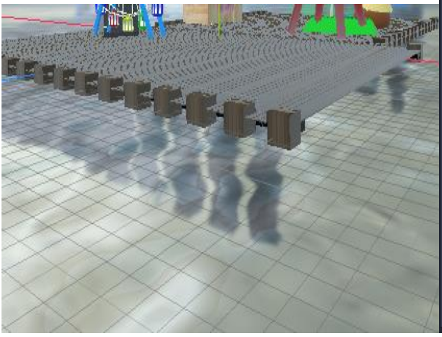
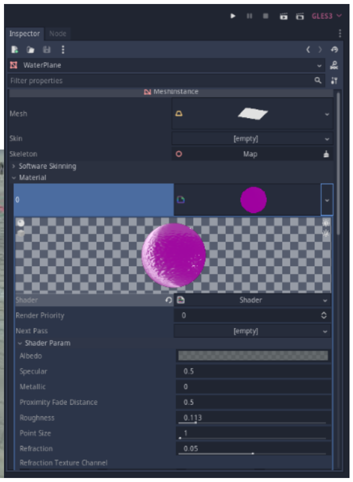


Figure 2: Waterplane/Dirt/Pier



WaterPlane
MeshInstance
Mesh
Skin [empty]
Skeleton Map
Software Skinning
Material
0
Shader Shader
Render Priority 0
Next Pass [empty]
Shader Param
Albedo
Specular 0.5
Metallic 0
Proximity Fade Distance 0.5
Roughness 0.113
Point Size 1
Refraction 0.05
Refraction Texture Channel

Figure 3: Shader

- # Laurence Buggy:

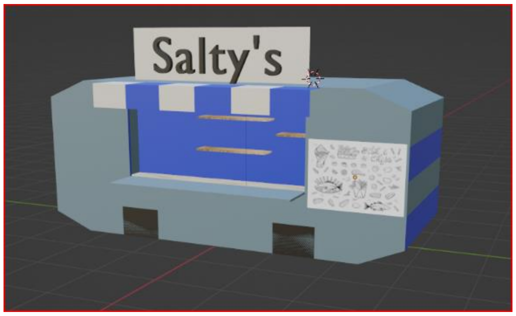
Work Done: Salty’s fish and chip shop, Coney’s ice cream stand, Ice cream cone stand and props for shops: Ice cream and fish shop.

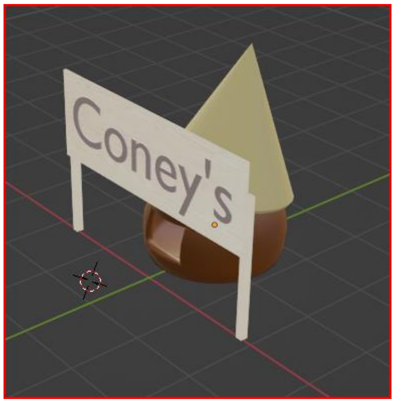
<u>Salty’s Fish and Chips</u>

To create Salty's fish and chip stand, I began with a simple box shape. Using Blender, I transformed the box into a van or cart-like shape by pulling out the sides and manipulating the edges. Next, I duplicated the object and scaled it down by a factor of 0.1. By using a boolean operator on the original cube and the scaled-down object, I was able to create an interior space for the food truck. To create a window on one side of the van, I added a cube and used a boolean to cut out a window-shaped hole. I added loop cuts on all sides of the van to add depth to the outside and create a space for the wheels underneath. To complete the truck, I added a pane to the side and added text to it, as well as a pane to add a texture to the front of the van. Finally, I added textures to the rest of the van to give it a more realistic appearance. This allowed me to create the perfect Fish and Chip "truck" for Salty's stand in our virtual festival shopping experience.

<u>Coney’s Ice Cream Stand</u>

To create the Ice cream stand I first started with a UV sphere and copied what I did before with boolean operators to hollow out the sphere. I then added a cone to the top of the sphere and used the boolean modifier again to hollow it out. I then placed it on top of the sphere at an angle and cut the bottom of the sphere to have it sit on the ground. I then added a sign to the front of the shop to display the name. I created the sign by getting two boxes and manipulating them to be sticks, I then got a pane and added a solidify modifier to it to give it depth. I added a wood texture and text to the sign.



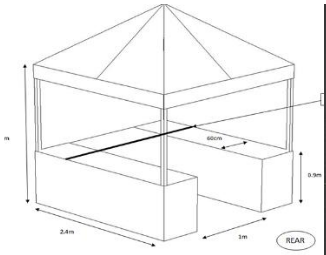


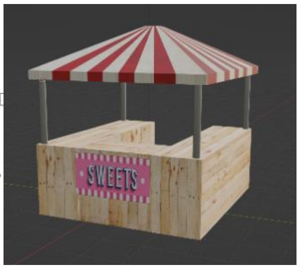
- **Lola Crowley**

Work done: Sweet Stand, fence, lollipop, chocolate bar, Rubix cube

Sweet Stand:

To build the sweet stand I started off using three boxes that I measured out to the dimensions in the figure down below and then put together, I found a wooden box texture that I used as the material for the base of the stand. I added in a cylinder that I modeled to the appropriate size and height then I duplicated it so that I ended up with four of them. Then for the "roof" of the stand I brought in a plane then added two faces of appropriate size to each of them to create the lifted tent top. I then found a texture that I thought suited the build and added that as the material. I also added a smaller plane and fixed it to the front of the model to work as the sign.





Fence:

The fence was very simple to build, I used the default box to create the two middle parts, then added two more boxes for the fence posts and shaped them accordingly. I then found a wooden grainy texture to use as the material. Since this is part of a Pier, I didn't want the fence to look "new" so I purposely picked a grainy weathered looking texture for it.

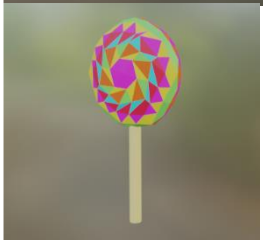


Figure 4: lollipop

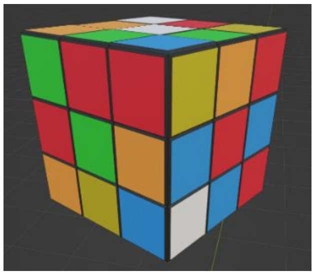
Lollipop:

I made the lollipop to go on the sweet stand, I started with a cylinder for the lollipop stick, I shrunk it down and then beveled the end of it so it wouldn't look so clear cut. For the sweet part of the lollipop, I used an Ico sphere and flattened it a bit, to give it the correct shape. I then subdivided the faces on the Ico sphere so that I could make some sort of pattern with the coolers.

Chocolate Bar:

This was a very simple one again. I used the default box and turned it in a rectangle, then I textured to look like a chocolate bar.

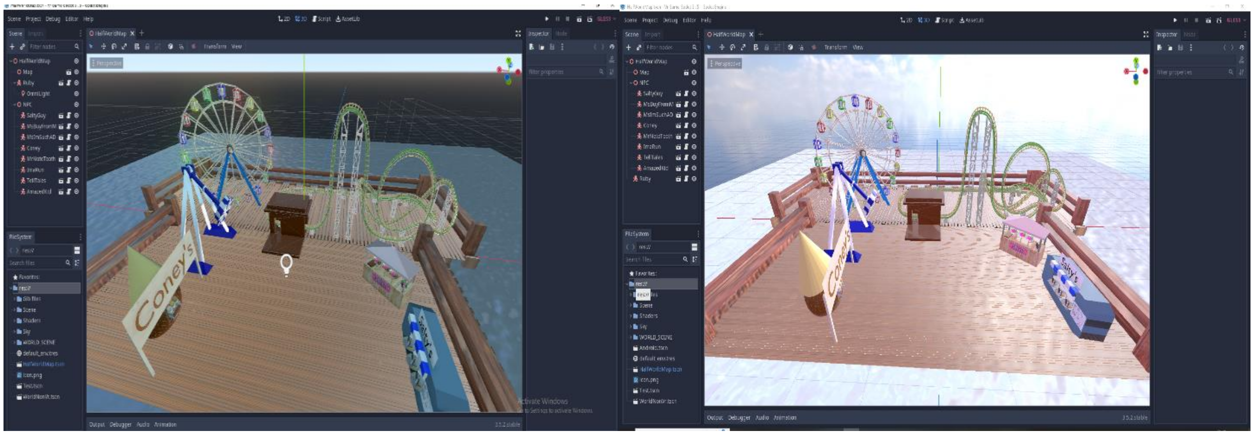
Rubik's Cube:



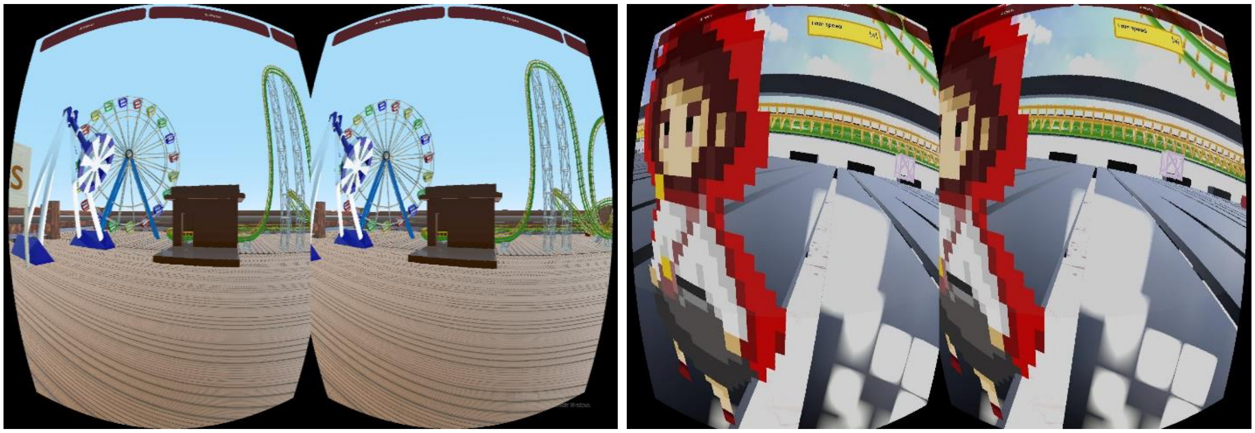
This was quite fun to make. I started with the default cube then used loop cuts to separate each face of the cube into nine faces. I then added a face to each of the buts and pulled them out slightly to give it some depth. I used my real life rubix cube to make sure the colors are all in a place that makes sense too.

Banner/Overview

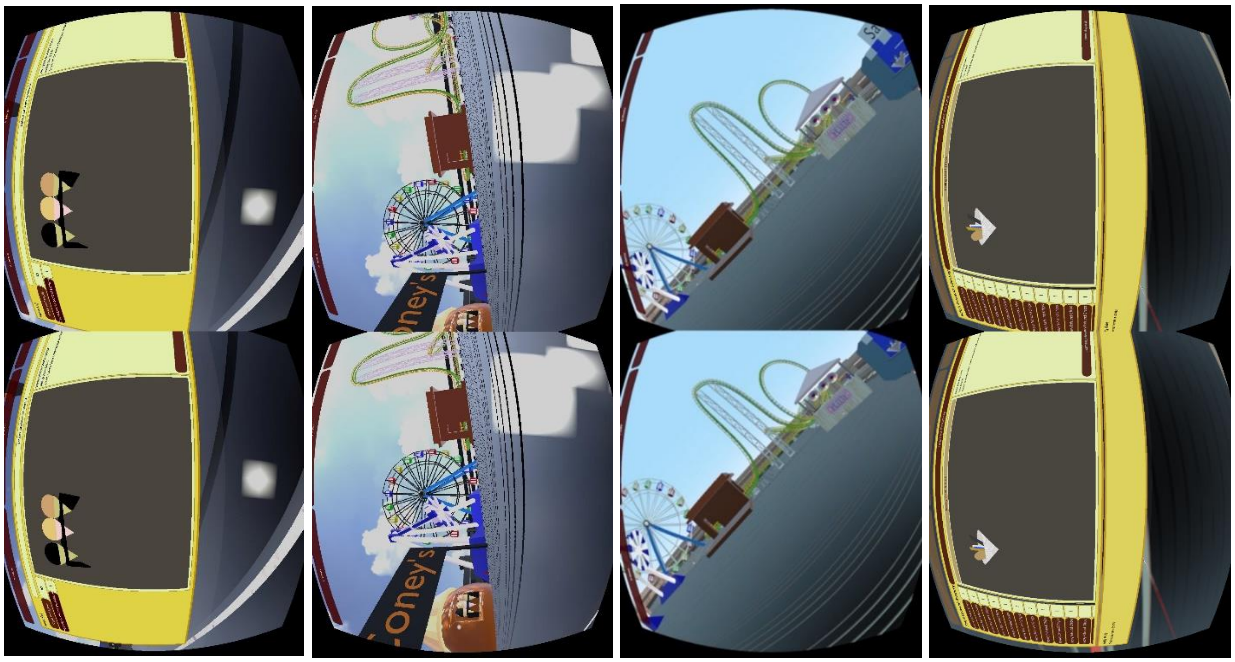
Final scene Godot 3.5



Final scene vr



Other related vr screenshots captured on ios/android.



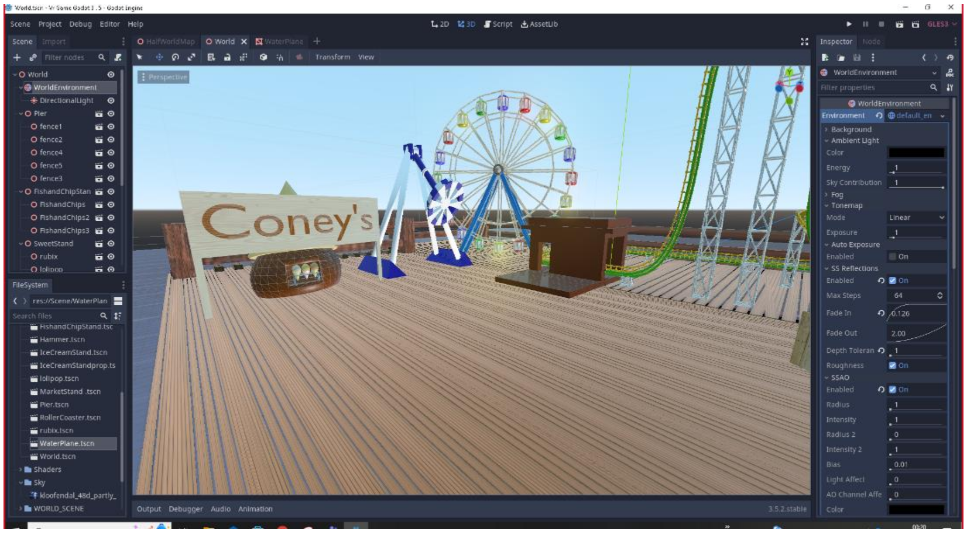
LARIZA JULIA BUCAU RODRIGO, LAURENCE MICHAEL BUGGY, ABDERAHMAN HAOUIT, RICHARD OGBONNAYA, LOLA KATE CROWLEY

Appendix:

Extra tasks/work:

We all done extra tasks apart from making objects such as:

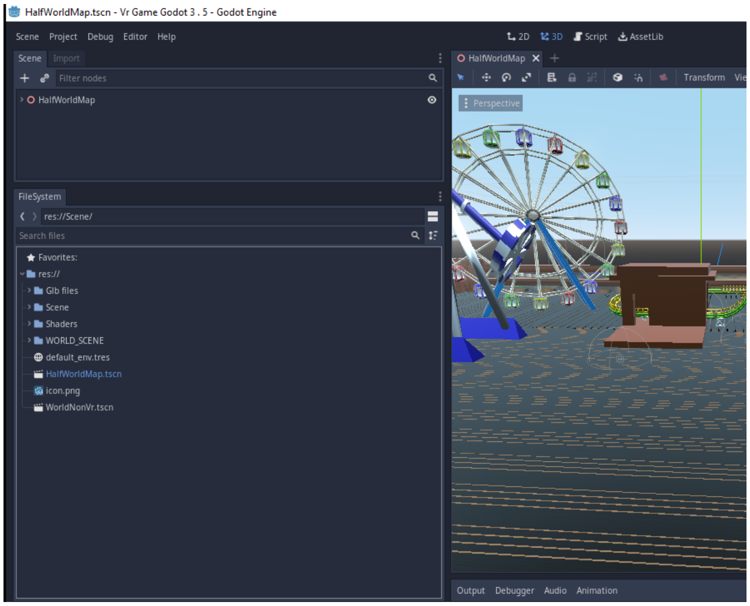
Abderahman worked on adding Lighting, glow, reflections etc on the environment , finalizing the document (Report) and exporting project to ios.



WorldEnvironment
Environment: default_en

- Background
- Ambient Light
  - Color
  - Energy: 1
  - Sky Contribution: 1
- Fog
- Tonemap
  - Mode: Linear
  - Exposure: 1
- Auto Exposure
  - Enabled: ☐ On
- SS Reflections
  - Enabled: ☑ On
  - Max Steps: 64
  - Fade In: 0.126
  - Fade Out: 2.00
  - Depth Toleran: 1
  - Roughness: ☑ On
- SSAO
  - Enabled: ☑ On
  - Radius: 1
  - Intensity: 1
  - Radius 2: 0
  - Intensity 2: 1
  - Bias: 0.01
  - Light Affect: 0
  - AO Channel Affe: 0
  - Color

Richard worked on the migration of objects/Project Structure: The main objective here was being able to import all the objects from my team partners and then formulate and structure the various objects and materials. So, if an outsider viewed the project, they could easily find any resource. This took a lot of scaling, folder creations, organization, keeping the workspace clean and identifiable and getting everyone's input on how everyone desired the project to look like. Which we all thought was a success. Our communication was mainly through teams having group calls and sharing screens to give visual representations.



LARIZA JULIA BUCAU RODRIGO, LAURENCE MICHAEL BUGGY, ABDERAHMAN HAOUIT, RICHARD OGBONNAYA, LOLA KATE CROWLEY

Julia worked on UI, infographic, android app fixes, inner code, code for attractions, app testing and vr!

Laurence worked on creating the fish and chip shop, Ice cream shop, small items for the fish and chip shop UI and working on completing the report.

Lola worked on the sweet shop and lollipop, fence post, chocolate bar and Rubics cube.

Instructions on how to use the app:

The app is made to be used on an android/iOS device. It is touch-screen and therefore does not require VR controllers. The app was made and tested on VR mode through touch-screen and pressing buttons.

1. Move the screen of your phone to move the character.
2. To enter interaction, go beside an NPC and see the pop box for interaction. Tap on the middle of the screen to start talking. Keep tapping the screen to follow the conversation. There is a video showing this in the .zip file.
3. If you talk to a shopkeeper, you will be able to see the items available in the shop. You can buy the items by clicking on the items list to see the information change. A way to check the items is to hold on the screen and drag you finger down. You can see the UI buttons changing colors as you hover on the items.
4. Buy the item by hitting the 'I want to buy this pls' button.
5. You can exit the shop by clicking on 'Finish Shopping' or buying all the items in that shop.
6. When you buy an item, you will be able to see it in your inventory. Click on the 'Item' button on the top screen. You should be able to see your newly bought item. Get rid of the item for more money.
7. The photo function on the app does not show the image (unfixed bug), but you can take photos by clicking on the 'Take a Photo' button. Look at your photo list by clicking on the 'Photo' button. You will see that a new instance of a photo tab has been made. You could sell a photo to get more money.

Screen Issue

Since the app is tested on a screen with longer length than width, you are not able to see the amount of money you have on the right-hand side (since the app is cutting off on the edge). You can run out of money and no longer be able to buy anything. But to account for this, you can sell pictures and the items you own for more money.

Problem faced:

The team faced a variety of difficulties throughout the course of the project, including lagging in the VR environment, black or dark screens, and the need to troubleshoot numerous scripts and pieces of code for VR and environment interaction. However, with cooperation and efficient problem-solving methods, the team was able to overcome these obstacles.

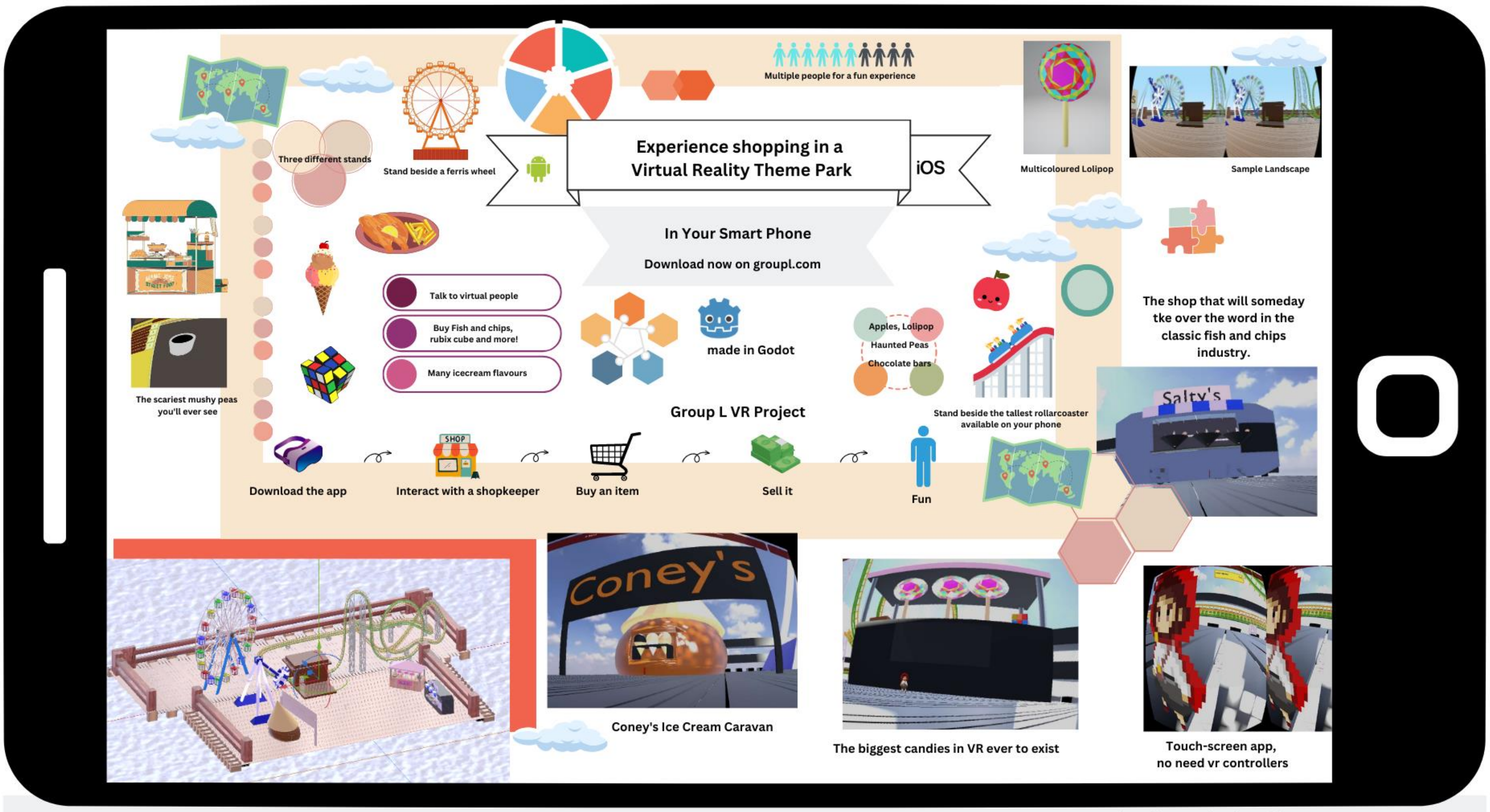
For instance, to improve efficiency, the team found files that were taking up a lot of space and eliminated them. We also investigated errors and established the root of the issue, enabling us to resolve it and proceed. One significant challenge the team encountered was with three textures that held up 100MB each of video RAM, which contributed to the overall size of the project folder.

The project folder size was cut in half when these textures were eliminated, which increased efficiency. In order to streamline the process and enhance overall organization, the team collaborated throughout to identify duplicate files and reorganize the folder structure. The team overcame the different challenges we ran across and finished the project successfully by cooperating with one another.

References:

- Code with Tom. (2020d, June 19). INVENTORY & ITEM SYSTEM in Godot [Video]. YouTube. https://www.youtube.com/watch?v=hYRN0eYttLc
- Blaise Guy. (2020, February 5). Godot - 3D Game with 2D Background (Super Basic Viewport Use) [Video]. YouTube. https://www.youtube.com/watch?v=oMOqKRcI1uw=
- Emi. (2020c, February 14). RPG or Visual Novel dialog box for Godot 3! [Video]. YouTube. https://www.youtube.com/watch?v=kkLqW8WhCgg
- (2023) *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=z6z00wwvk1A (Accessed: May 1, 2023).
- (2022) *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=cyA8gViQVn0 (Accessed: May 1, 2023).
- https://www.youtube.com/watch?v=7rGNoGZfRYg -- **Waterplane**
- https://youtu.be/1c3pQUTTFg8 -- **Pier**
- https://www.canva.com/ - Infographic

Infographic for the Theme Park:


Multiple people for a fun experience
Three different stands
Stand beside a ferris wheel
Experience shopping in a Virtual Reality Theme Park
iOS
Multicoloured Lolipop
Sample Landscape
In Your Smart Phone
Download now on groupl.com
Talk to virtual people
Buy Fish and chips, rubix cube and more!
Many icecream flavours
made in Godot
Apples, Lolipop
Haunted Peas
Chocolate bars
The shop that will someday tke over the word in the classic fish and chips industry.
The scariest mushy peas you'll ever see
Group L VR Project
Stand beside the tallest rollarcoaster available on your phone
SHOP
Salty's
Download the app
Interact with a shopkeeper
Buy an item
Sell it
Fun
Coney's
Coney's Ice Cream Caravan
The biggest candies in VR ever to exist
Touch-screen app, no need vr controllers

Group L VR project: Theme Park Infographic

LARIZA JULIA BUCAU RODRIGO, LAURENCE MICHAEL BUGGY, ABDERAHMAN HAOUIT, RICHARD OGBONNAYA, LOLA KATE CROWLEY