



أُونِيْوْرِسِيَّتِيْ تِيْكُنِولُوْجِيْ مَارَا

UNIVERSITI
TEKNOLOGI MARA
MALAYSIA

**BACHELOR OF COMPUTER (HONS.) MULTIMEDIA COMPUTING
(CDCS253)**

**VIRTUAL REALITY
(CSC573)**

TITLE : TOYTOPIA TOY SHOP

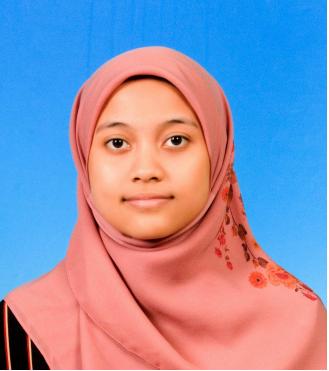
LECTURER : MADAM SITI NURAMALINA BINTI JOHARI

GROUP MEMBER	AISY BATRISYIA BINTI ABD WAHID	2023624314
	NOOR AISYA AINA BINTI AZAHAN	2023867516
	SYAFIQAH RUZANA BINTI SAZALI	2023686282
GROUP	GROUP 3	
DATE OF SUBMISSION	24 JANUARY 2025	

TABLE OF CONTENT

1.0 Team Members.....	3
2.0 Background of Project.....	3
2.1. Introduction.....	3
2.2. Theme.....	4
2.3. Concept.....	4
2.4. Interactivity Provided.....	4
3.0 Design and Development Process.....	4
3.1. Assets Creation.....	6
3.2. Scripting and Functions.....	8
4.0 Output.....	8

1.0 Team Members

	<p>Name: Noor Aisyah Aina Binti Azahan Role: Project Manager Description:</p> <ul style="list-style-type: none">• Plan and decide on the project's scope.• Create and manage the project team.• Monitor each member's progress and make sure each task is done exactly.
	<p>Name: Syafiqah Ruzana Binti Sazali Role: VR Developer Description:</p> <ul style="list-style-type: none">• Verify compatibility with various VR systems and hardware.• Enhance performance and graphics for an enjoyable VR experience.• Work together with designers to effortlessly integrate assets and graphics.
	<p>Name: Aisy Batrisyia Binti Abd Wahid Role: Project Designer Description:</p> <ul style="list-style-type: none">• Develop creative and appealing concepts based on its specifications.• Create attractive VR environments and components.

2.0 Background of Project

2.1. Introduction

This project aims to create a virtual toyshop environment by using virtual reality technology. It focuses on the journey of the player visiting a toyshop to

buy their desired toy from their collected coins. The environment will provide two scenes. Users can move between these two scenes by going in and out of the shop.

2.2. Theme

The theme for this project is a toy and hobby shop. This theme is chosen to create an immersive and interactive story in a virtual reality environment that addresses the process of purchasing a toy physically at a hobby shop.

2.3. Concept

The shopping experience served as the inspiration for the creation of this VR experience. Throughout the entire creation process, we primarily employ the Unity3D game engine as our scripting platform. In addition, we design and make in-game assets in 3DS Max and Blender which are loaded into Unity.

2.4. Interactivity Provided

In this project, player will start exploring an environment of a park to collect the coins , which later will be used to buy the toys from the shop. Player can go to the next scene which is inside of the shop. Player can drag an interactive object which is one of the toys and put it on the counter to buy it. The total coins at hand will be deducted.

3.0 Design and Development Process

The 3DS Max, Blender and Unity 3D game engine were used in the development of this project. The team members created the assets in accordance with the tasks they were given.

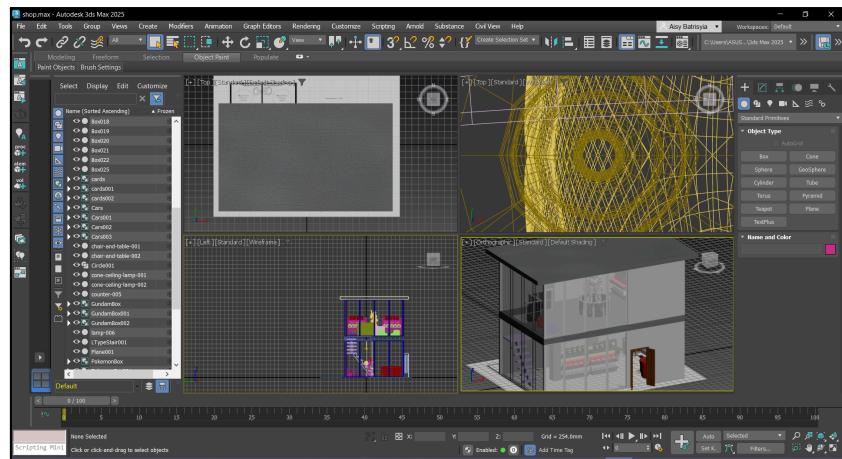


Figure 3.1 3DS Max Interface

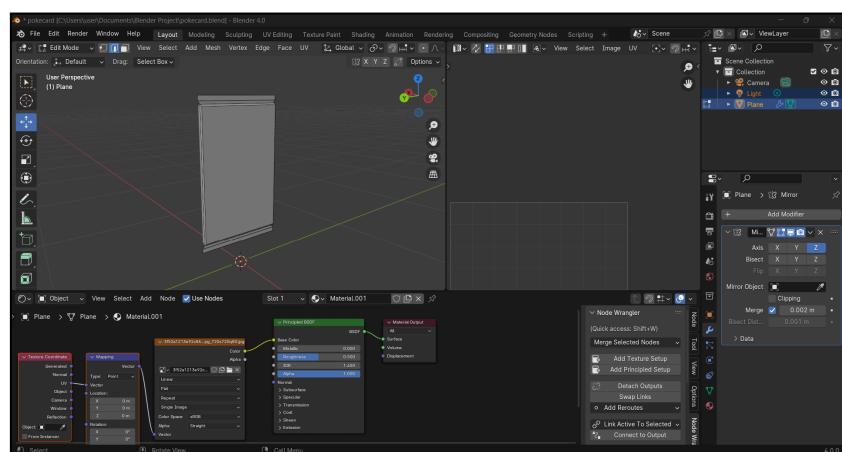


Figure 3.2 Blender Interface

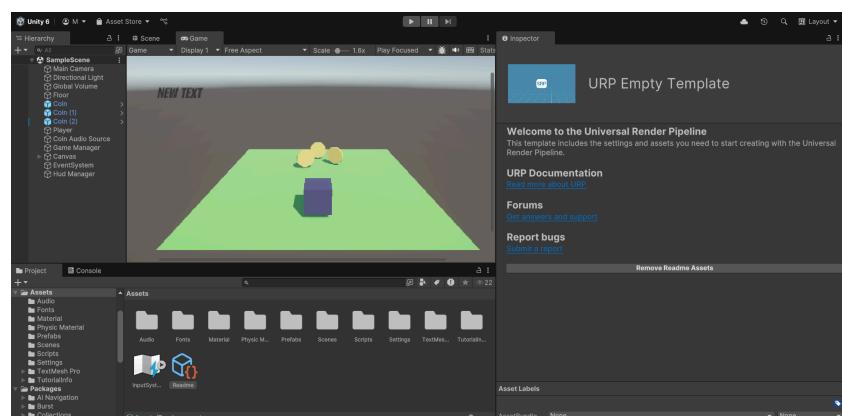


Figure 3.3 Unity Interface

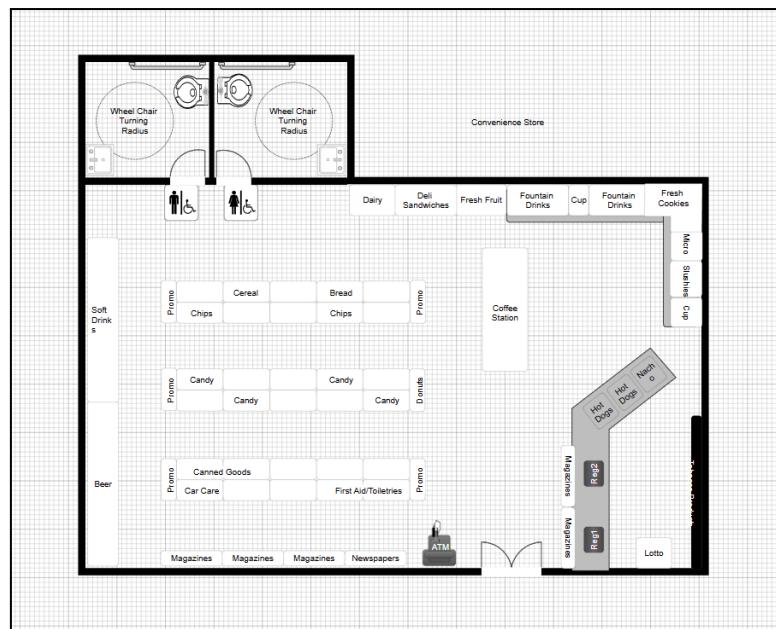


Figure 3.4 Floor Plan

3.1. Assets Creation

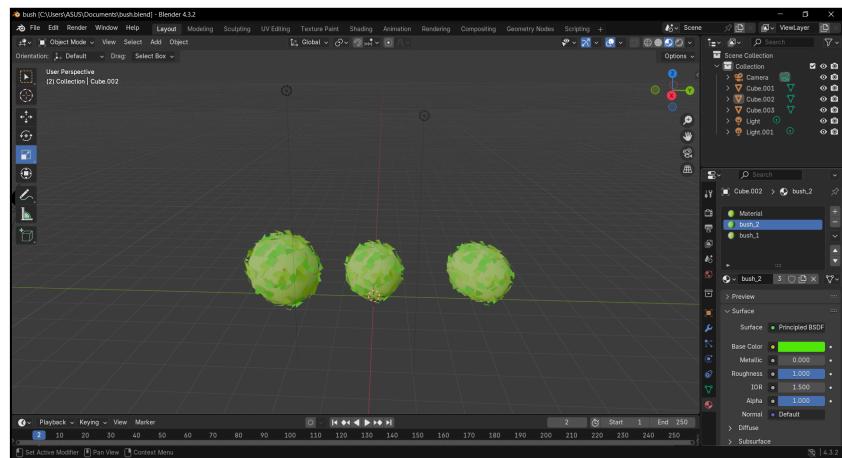


Figure 3.5 Bush Creation

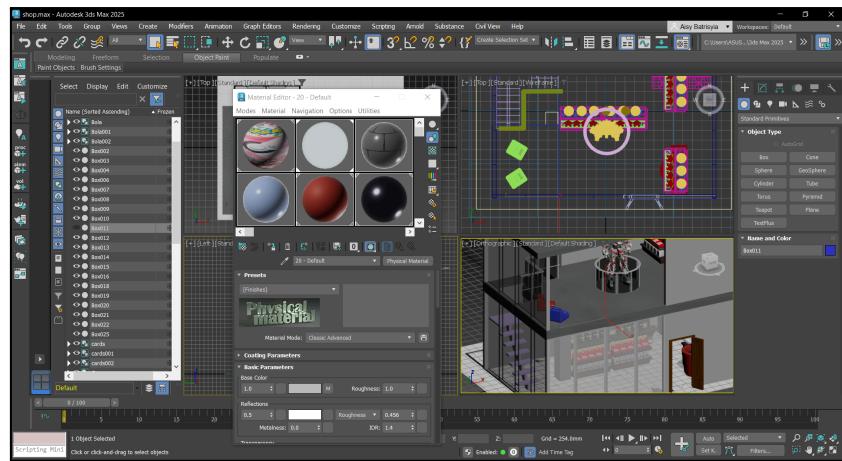


Figure 3.6 Shop Creation

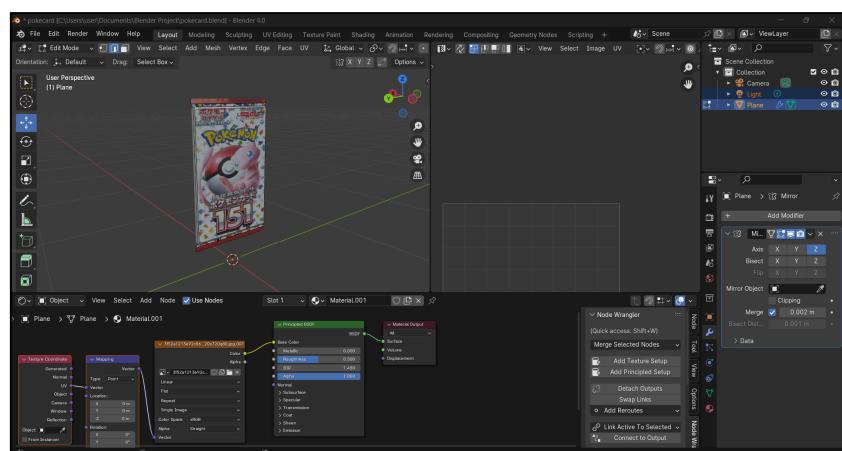


Figure 3.7 Pokemon Card Creation

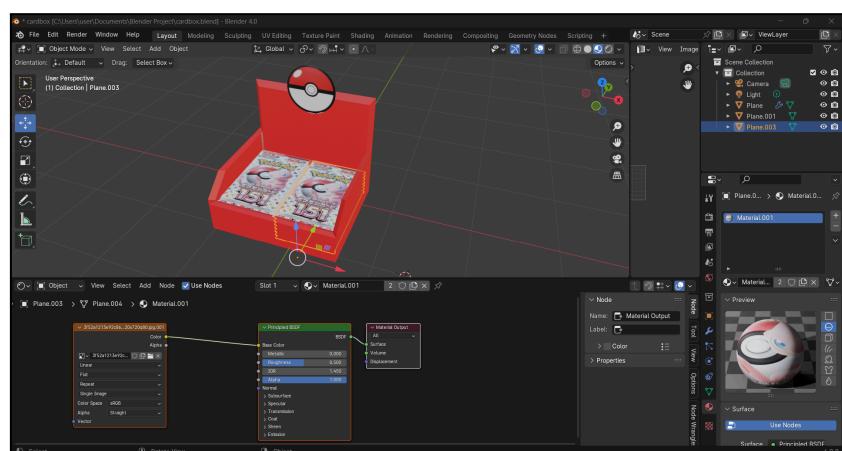


Figure 3.8 Pokemon Cardbox Creation



Figure 3.9 Stuffed Bear Creation

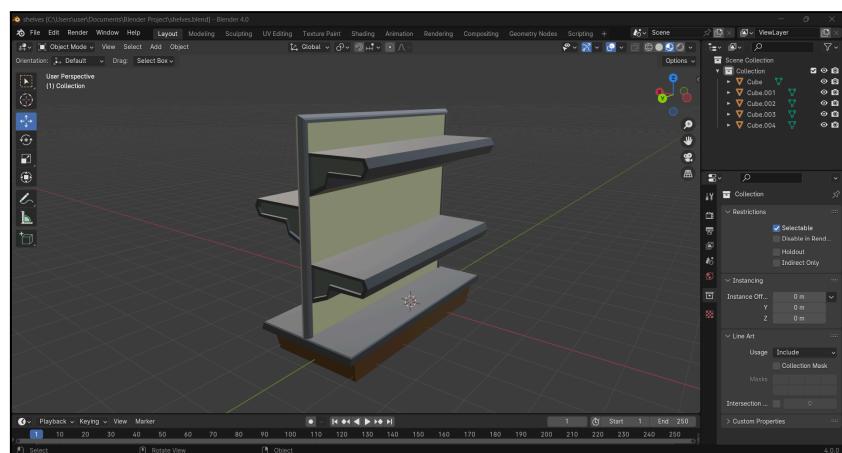


Figure 3.10 Shelf Creation

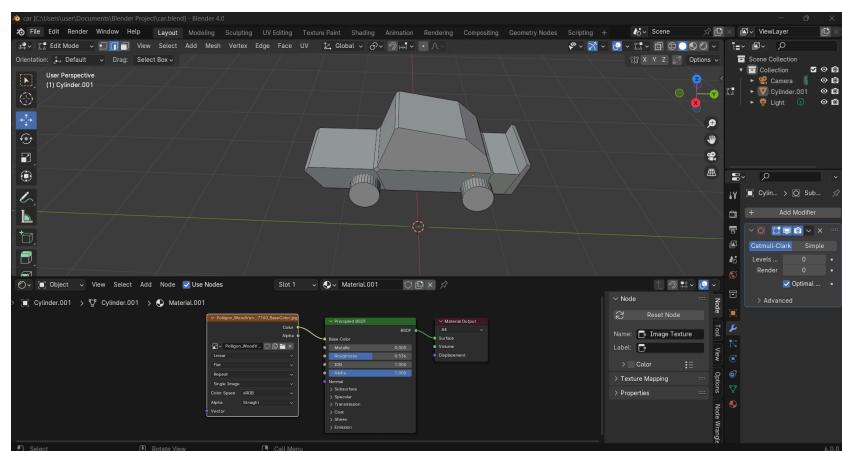


Figure 3.11 Toy Car Creation

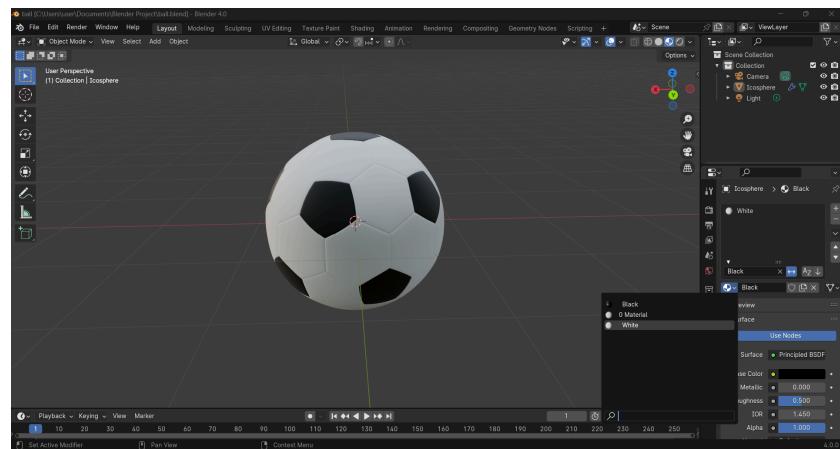


Figure 3.12 Soccer Ball Creation

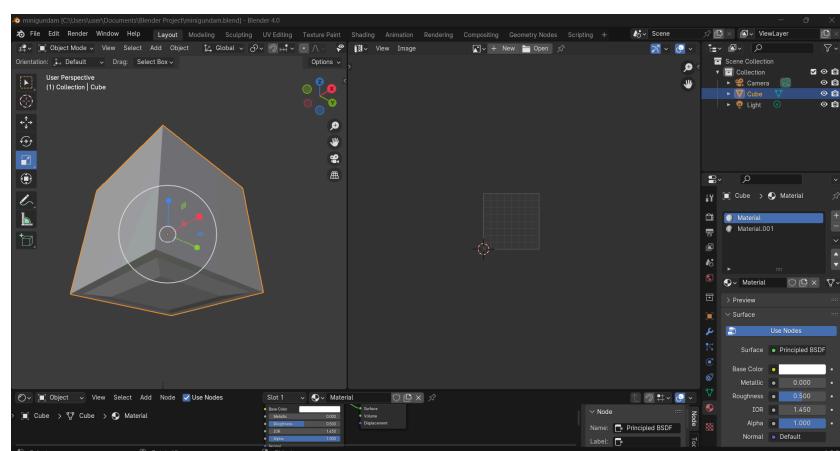


Figure 3.13 Mini Gundam Box Creation

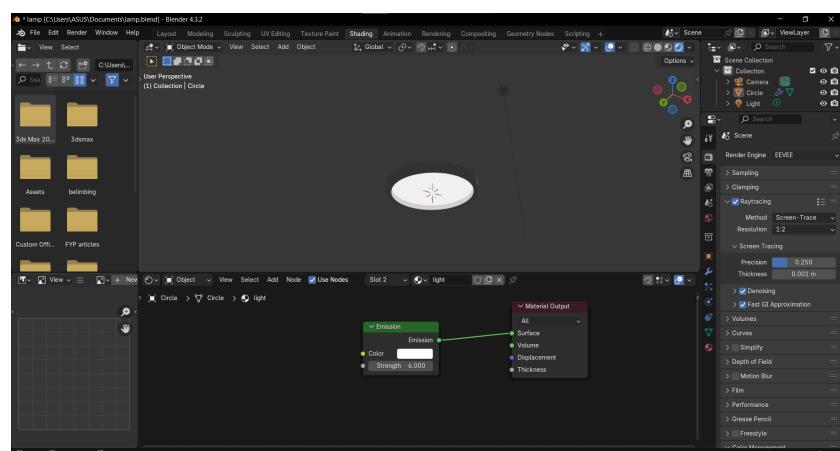


Figure 3.14 Lamp Creation

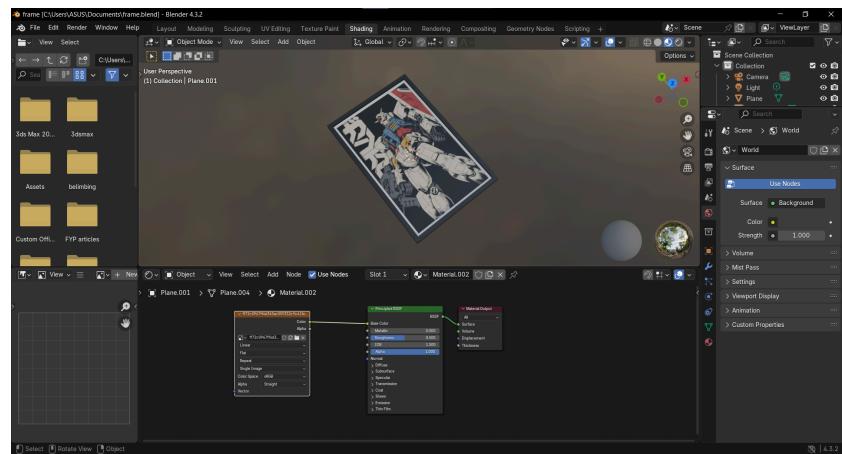


Figure 3.15 Poster 1 Creation

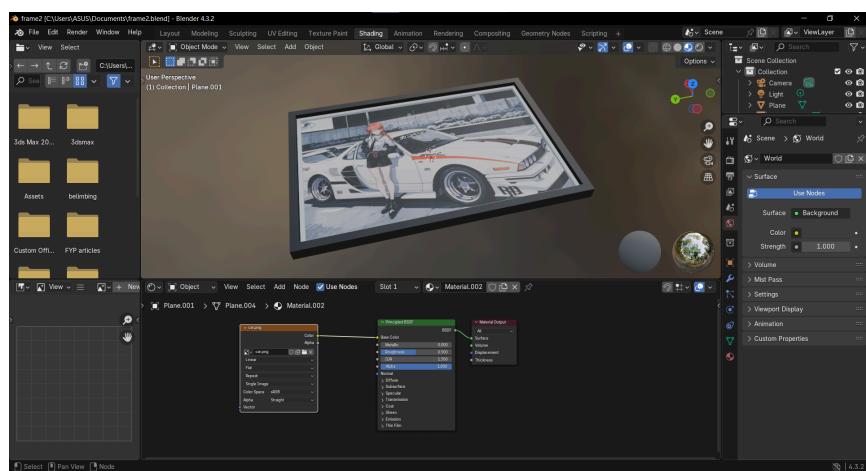


Figure 3.15 Poster 2 Creation



Figure 3.16 Park Environment Creation

3.2. Scripting and Functions

```

6 public class PlayerController : MonoBehaviour
7 {
8     void OnTriggerEnter(Collider collider)
9     {
10         if (collider.gameObject.tag == "Coin")
11         {
12             print("Grabbing coin..");
13             coinAudioSource.Play();
14             Destroy(collider.gameObject);
15
16             // Call CoinCollected
17             playerInventory.CoinCollected();
18
19             // Ensure coinText is not null before updating
20             UpdateCoinText(playerInventory); // Update coin UI after collection
21
22             // Ensure coinText is not null before updating
23             if (coinText != null)
24             {
25                 coinText.text = playerInventory.NumberOfCoins.ToString();
26             }
27             else
28             {
29                 Debug.LogError("CoinText is not assigned.");
30             }
31         }
32     }
33 }

```

Figure 3.16 Grab coin scripting

```

7 public class ToyInteraction : MonoBehaviour
8 {
9     void HandleToyPickup()
10    {
11        // Trigger pickup using the E key
12        if (Input.GetKeyDown(KeyCode.E) && hoveredToy != null && pickUpScript.GetHeldObject() == null)
13        {
14            Debug.Log("Picking up: " + hoveredToy.name); // Debug log to confirm that the key press is registered
15            pickUpScript.PickUpObject(hoveredToy); // Pickup the hovered toy
16            heldToy = hoveredToy; // Set the heldToy to the hovered toy
17            currentToyData = hoveredToy.GetComponent<ToyData>(); // Get the ToyData component
18            if (currentToyData != null)
19            {
20                Debug.Log("Toy data: " + currentToyData.toyName);
21            }
22            else
23            {
24                Debug.Log("Toy data is null!");
25            }
26            HideToyDetails(); // Hide the details after pickup
27        }
28    }
29
30    void DropToy()
31    {
32        // Trigger drop using Q key
33        if (Input.GetKeyDown(KeyCode.Q) && pickUpScript.GetHeldObject() != null)
34        {
35        }
36    }
37 }

```

Figure 3.17 Item Pickup Scripting

```

5 public class ToyPurchase : MonoBehaviour
6 {
7     private void Update()
8     {
9         Debug.Log("Update called");
10        // Check if the player is near the counter and has pressed the B key
11        if (isNearCounter && Input.GetKeyDown(KeyCode.B))
12        {
13            Debug.Log("B key pressed, checking held toy...");
14            GameObject heldToy = toyInteraction.GetHeldToy(); // Get the held toy
15            Debug.Log("Toy Purchase Held toy: " + (heldToy != null ? heldToy.name : "None"));
16
17            // Try to get the ToyData and process the purchase if it's valid
18            if (heldToy != null)
19            {
20                ToyData toyData = heldToy.GetComponent<ToyData>(); // Get the ToyData component from the held toy
21                if (toyData != null)
22                {
23                    Debug.Log("ToyData found: " + toyData.toyName); // Debug log for ToyData
24
25                    Debug.Log("Attempting to deduct coins for toy: " + toyData.toyName);
26                    Debug.Log("Toy price: " + toyData.price);
27
28                    // Check if playerInventory is assigned
29                    Debug.Log("Player Inventory is: " + (playerInventory != null ? "Assigned" : "Null"));
30
31                    Debug.Log("Coins before deduction: " + playerInventory.NumberOfCoins);
32                    bool purchaseSuccessful = playerInventory.DeductCoins(toyData.price);
33                    Debug.Log("Coins after deduction: " + playerInventory.NumberOfCoins);
34
35                    if (purchaseSuccessful)
36                    {
37                    }
38                }
39            }
40        }
41    }
42 }

```

Figure 3.18 Purchase Scripting

4.0 Output

This section consists of the final output of the project.

Environment

Scene 1



Figure 4.1 Park Environment

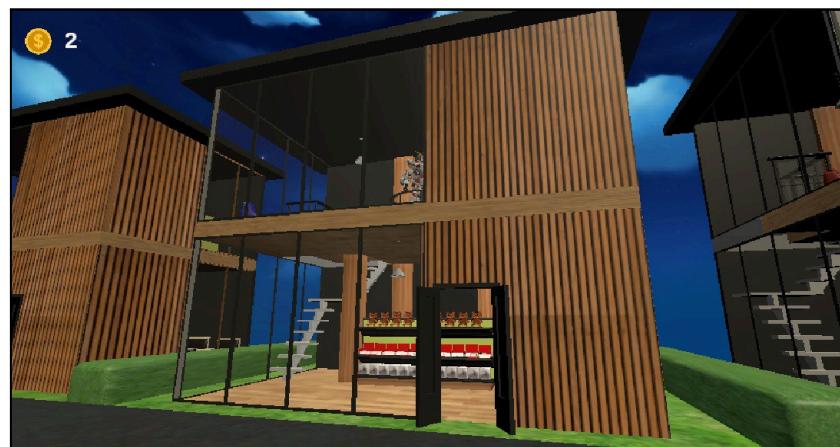


Figure 4.2 Park Environment Outside of Shop

Scene 2



Figure 4.3 Shop Environment 1st Floor



Figure 4.4 Shop Environment 2nd Floor

Interactivity

When the player hover, it shows the name of the toys and its price, and the player can grab the toys and throw them. To buy the toys, the player have to put the toy in the counter.

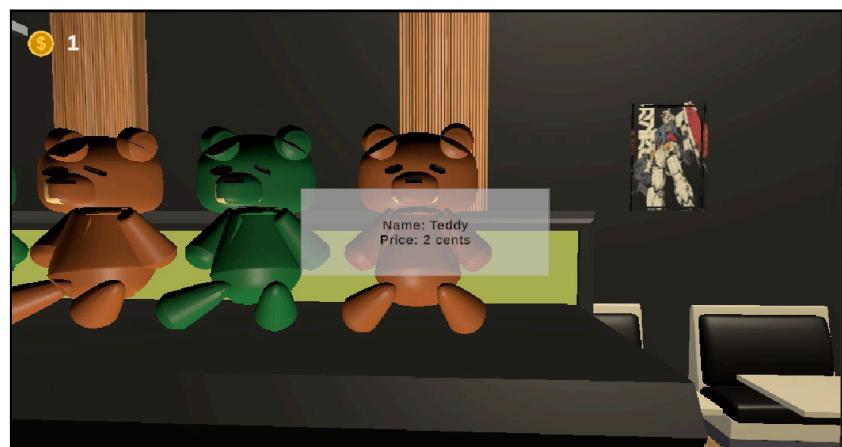


Figure 4.5 Hover



Figure 4.6 Grab toys