Immersive Unity

Name: **Samiksha Modi**
Roll No: **2019331**
Email: samiksha19331@iiitd.ac.in

Link to Google Drive Folder (Containing apk, source code, demo presentation, assets developed):
https://drive.google.com/drive/folders/1ecA-EaC36DRBCdDvmeRtXz73NVGtL2zu?usp=sharing

I created a single Unity Project in the Unity Editor.

The Part 2 source code and things used can be found in the folder named “Part2” present inside the Assets folder and the Scene is “Week1 v2.unity” present inside the Scenes folder.

The Part 3 source code and things used can be found in the folder named “part3” and the Scene is “PersonalBranding.unity” present inside the Scenes folder.

Part 2

Link to Demo video: https://youtu.be/xnjLLZsCa2k

The prompt I chose

AR Game

Concept, Intent, Goals

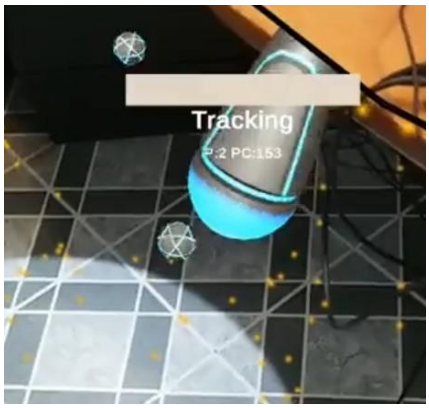
My idea was to place something in my room that I could hit with a ball. I often have a ball in my drawer and sometimes I just take it out and try to hit something in my room, maybe try to knock it over. But the problem was that once I threw the ball I had to get up and get it back. Which I did not like doing.

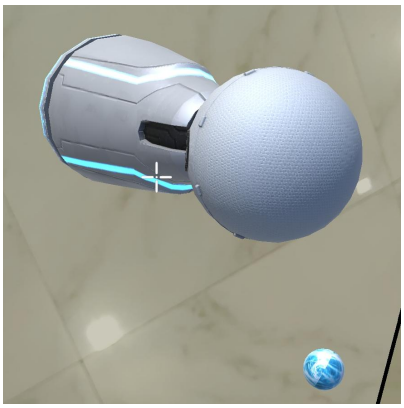
So I wanted to make an AR game for people like me in which I could place something in my room that I could hit without having to get up to get the ball back or worry about hitting and breaking something in the room.

Behaviour Developed from Scratch

- I used ARCore in part 2. My Scene had an **ARCamera** with **Directional Light** and **Point Light** as child of the AR Camera to light my scene properly. I changed the color of Point Light and I decreased the Intensity of Directional Light to 0.8 as the default value 1 was too bright.

- I created a **texture** for my ball using an image I found and changed it in the ball prefab which was **downloaded online and imported**





Before(left); After adding my texture (right)

- I wrote the code to shoot the balls at the direction of where the camera was looking

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ShootScript : MonoBehaviour
{

     public GameObject ball;

    public GameObject arCamera;

       public void shootButton()
            {
                RaycastHit hit;
                if(Physics.Raycast(arCamera.transform.position, arCamera.transform.forward, out hit))
                {
                    GameObject newBall = Instantiate(ball,hit.point,Quaternion.LookRotation(hit.normal));
                    Rigidbody rb= newBall.GetComponent<Rigidbody>();
                    rb.AddForce(5000*arCamera.transform.forward);
                }
            }

}
```

The shootButton() instantiates the ball prefabs and sets its position to where the camera is, and then uses the physics rigidbody addforce api to shoot the ball forward.

- I wrote the code to place the object on tapping on the detected plane.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;
using UnityEngine.EventSystems;


[RequireComponent(typeof(ARRaycastManager))]
public class ArTapToPlaceObject : MonoBehaviour
{

    public GameObject gameObjectToInstantiate;
    private GameObject spawnedObject;
    private ARRaycastManager _arRaycastManager;
    private Vector2 touchPosition;
    public static int cnt=0;

    static List<ARRaycastHit> hits = new List<ARRaycastHit>();
    // Start is called before the first frame update
     private void Awake()
     {
         _arRaycastManager=GetComponent<ARRaycastManager>();
     }

     bool TryGetTouchPosition(out Vector2 touchPosition)
     {
         if(Input.touchCount>0)
         {
             touchPosition=Input.GetTouch(0).position;
             return true;
         }
         touchPosition=default;
         return false;
     }

  /*   bool IsPointerOverUIObject()
    {
        PointerEventData eventDataCurrentPosition = new PointerEventData(EventSystem.current);
        eventDataCurrentPosition.position=new Vector2(Inpn.x, Input.mousePosition.y);
        List<RaycastResult> results = new List<RaycastResult>();
        EventSystem.current.RaycastAll(eventDataCurrentPosition,results);
        return results.Count>0;

    } */

    // Update is called once per frame

    void Update()
    {

        if(!TryGetTouchPosition(out Vector2 touchPosition))
            return;


            if(_arRaycastManager.Raycast(touchPosition,hits,TrackableType.PlaneWithinPolygon))
            {
                var hitPose=hits[0].pose;

                /* figure out if spawned object already. If already then move around else spawn */
                if(spawnedObject==null && cnt==1)
                {
                    /* then create it */
                    spawnedObject=Instantiate(gameObjectToInstantiate,hitPose.position,hitPose.rotation);
                    cnt=1;
                }
               /*   else{
                    spawnedObject.transform.position=hitPose.position;
                }  */
            }


    }

    public void rescanButton()
    {

        spawnedObject=null;
        cnt=1;

    }


}
```

I used ARRaycast manager method to place the robot on the plane using the screen coordinates of the touched point. If a hit was detected i.e. raycast hit a plane that the AR system has detected, then I add a robot prefab at the location that I hit.
To add the robot the user has to tap on crosshair which is the rescanButton() in the code.

Resources Utilised

- Libraries and Languages Used : C#, ARCore, XR Plugin for plane detection
- Robot and Ball imported from https://www.coursera.org/learn/handheld-ar
- Ball texture created using https://images.app.goo.gl/mpY3ZkanWjuNAhmw7
  I changed the texture of the imported ball
- Crosshair image imported from link give in the description of this tutorial https://youtu.be/RzmBnllRnh8
- C# code: for understanding how to use raycast to shoot the ball in the direction of the camera https://youtu.be/RzmBnllRnh8
- C# code: for tap to place object in ar https://youtu.be/xguiSueY1Lw
- Unity documentation to know about lights, plane detection, raycast, etc

Other Information I Want to Highlight

I know there are similar games like paper toss in AR already there on PlayStore but I really wanted to try this out so I started working on it and searching the internet to get started. I did not have any idea about how to get started with making augmented reality apps on Unity. So I went on Coursera on which I've completed 7 courses so far and looked for a course on AR in Unity to get me started.

I found the following and they had a similar idea so I used that for my Assignment in Part 2.
Handheld AR App Development with Unity
https://www.coursera.org/learn/handheld-ar
The course was released in 2018, and had tasks without the solutions to make this game. The code in the Coursera discussion forum could also not be used in my 2021 version of ARFoundation since it has received many updates and does not have the same classes, syntax, namespace to be used etc. So I had to understand some tutorials I found on the internet (Present in references) and adapt it for my needs.

The difference between their idea and mine was that I wanted to place multiple objects to place in my room and try different things like I normally do.
For eg, arrange 3 targets very close to each other and then try to pass the ball between them without touching anything, or simply just hitting the target. So I included the option of placing multiple targets in my room.

Part 3

Link to Demo Video: https://youtu.be/nSzxJ_GJqCw

Design Intent, Goals, Assets

My idea was to embed my personal brand into things which people see everyday and which can be given as gifts like a paperweight or a calendar. This will keep me on their radar without me being there.

So I took one example of a calendar and made it a showcase for my work. The person using that calendar can just scan the image and see my work come alive.

Behaviour Developed from Scratch

- I used an **AR Camera** from Vuforia Engine with **Directional Light** to light the scene.
- I **downloaded and imported** a crate and water texture from the unity asset store.I put the water **texture** on the crate and placed a **Point Light** inside it to give it a glowing effect. I then put a point light inside
- I created a button which on clicking opens my Portfolio on the web browser

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class part3 : MonoBehaviour
{
    public void visit()
    {
        Application.OpenURL("https://samikshamodi.github.io/Portfolio/index.html");
    }
}

```

- I added a video player on a Quad so that when a particular image target is detected by the camera, it plays the video in augmented reality.

Resources Utilised

- Libraries and Languages Used: C#, Vuforia for image target detection
- Crate imported from https://assetstore.unity.com/packages/3d/props/crate-31462#content
- Water texture imported from https://assetstore.unity.com/packages/2d/textures-materials/water/stylize-water-texture-153577

- For image target detection in vuforia https://youtu.be/khavGQ7Dy3c
- For playing video in AR https://youtu.be/MtiUx_szKbI