CSE 366
Spring 2024

Assignment 3: Augmented Reality
Due date: Thursday, May 2, 2024, 5:29 p.m.

In this assignment, you will be exploring the realm of augmented reality (AR). Similar to your previous assignments, you are required to build a virtual world, but by "augmenting" the real world on planar surfaces in the real world. However, in this assignment, you will be holding your device in your hand instead of mounting it onto your head. You will be using Unity's **AR Foundation** as the platform for AR application development (AR Foundation 4.1.13 & Tutorial). You will use your smartphone or tablet for this assignment. Note that the specification of the assignment assumes that you are using our recommended version of Unity (2020.3.25f1). Also, we strongly recommend that you meet the following ***minimum*** requirements for this assignment to support AR Foundation and its core library (ARCore for Android, ARKit for iOS).

Overview

For this assignment, you will have to design an AR minigolf application in which a user can interactively build and play an augmented 3D scene on planar surfaces in the real world. Your application will have two modes: a ***build*** mode and a ***game*** mode. Therefore, your application should have a simple UI to toggle between each mode.

The application should start in the build mode by default. To begin building, the user should first be prompted to tap and select a horizontal plane from their real-world surroundings, such as the floor, table, etc. After selecting the plane, your application should prompt a UI that will ask the users to place **four** markers for the game: starting point, par hole (i.e., endpoint), and 2 markers for teleportation holes (details explained below). Once the markers are placed, users should then be able to choose from a list of virtual objects, manipulate the position and rotation of the selected object, and confirm its placement in the real world.

In the game mode, the user should be able to play the mini-golf game. To play the game, the user will need to orient the AR device (tablet or phone) and use a tap gesture swipe to hit the golf ball. The objective of the game is to complete the mini-golf course in as few strokes as possible.

Reminder: You may move while working or playing with this assignment. Please be careful about your physical environment (e.g., stairs, steps, walls, desk corners) to avoid injury.

Selecting the game planes

As mentioned in the overview, the user should first be prompted to select a plane in their scene. This can be achieved using AR Foundation's ground-plane detection capability**.** Implement it so that your application shows a visual cue, such as a mesh, to display all detected surfaces. Once a user taps and selects a plane, the UI should show the selected plane using any color.

Physical-world interactor preparation

To present the users with a fascinating AR experience, your application must provide a number of interactions between the virtual world and the physical world:

1. Physical world setup
   Printed markers are often used in AR applications to place virtual objects on absolute physical locations. In such a setup, the application places virtual objects exactly on the marker's physical location whenever the marker is in the user's view. In this assignment, you are required to print and place four physical markers to (1) mark the start position of the golf ball / game, (2) determine two locations of the teleportation holes, and (3) mark the par hole. A marker can be any image. However, there are a set of rules that need to be followed in order to have the stable registration of virtual objects. It is recommended to use this "QR-code-like AR marker" generator if you are not able to achieve a satisfying result. Each physical marker must be unique. The following are the markers you have to print and place in your application:
   - Two A4-sized QR-code markers - These physical markers will be used to mark the starting point and the par hole position.
   - Two half-A4-sized image markers - These physical markers will be used for the teleportation holes.

   Extra credit: Create a UI that tracks the markers and prompts "good" or "bad" marker placement based on their distance from each other. For instance, good placement is when all markers (or marker pairs) are at least 2m apart.

2. Virtual setup
   The application needs to be configured so that the application recognizes the physical world markers and places virtual objects on them accordingly. The virtual object for the starting point should have a white flag with any visual cue, the teleportation markers should have a green flag and a hole each, and the par marker should have a red flag with a hole. These virtual objects need to be rendered on the user's device whenever the device detects the physical markers.

Fig. 1. Example of an image marker for the par hole.

For an effective game experience, make sure that the markers are not placed close to each other.

Build mode (Placement)

After selecting the plane and placing the markers, your application should start the build mode. In this mode, users should be able to **(B1)** draw a boundary around objects in the real world and (**B2**) augment virtual objects onto the real world.

The goal of (B1) is to treat real-world objects like obstacles. That is to say, when the ball collides with these objects, the ball should bounce off their surfaces. While this is challenging to accurately implement based on the exact shape of the real-world object, we will assume assigning a simple cube bounding box around each physical object. To achieve this, you will need to design the following steps:

1. For a real-world object on the selected planes, the user should be able to tap on the object to initiate a transparent cube.
2. Display the following UI controls on the screen: scale up, scale down, rotation about the z (up) axis, and done.
3. On tapping the scale and rotation buttons, the transformations should be applied to the cube. Make sure the base of the cube is always touching the plane. Namely, in scaling up/ down, the cube should not sink in the plane or shrink to float in the air.
4. When satisfied, the user should tap the done button to finalize the obstacles.

This bounding box will serve as your collider object in Unity to enable the ball to "bounce" off when it hits the real-world object. For the scope of this assignment, you will have to prompt the user (using a UI design of your choice) to mark at least **two** real-world objects.

For (**B2**), users are presented with a UI to select and place 3D virtual objects on the ground plane. There should be a library of predefined objects similar to what you used to design your Assignment 1. When a virtual object from the library is selected, the user should be able to choose a position on the selected plane in the real world, using the view through their AR device. Your application should then place the selected 3D object at the specified position. Your library should include at least the following: (1) trees, (2) a cart, and (3) buildings.

Build mode (Transformation)

When a 3D object from the library is selected and the user taps on the device screen, the user should be provided with a UI similar to Fig. 2. For full points, the following minimum functionalities should be implemented:

> A joystick to translate the virtual objects in the x-z direction on the planar horizontal surface and another joystick to rotate them about the yaw axis of the surface.

The reasoning behind the joystick UI is to allow the user to freely modify the position/rotation of a virtual object even after the screen tab action. Once satisfied, the user can tap the 'Finish edit' button on the screen to finalize the placement.

Fig. 2. User Interface example look

Play mode

When a user finishes building the virtual scene augmented on real-world surfaces, the user should be able to tap the 'Mode' button to go to Game mode and play the game

Interaction - Unlike the build mode, the user will play by tapping on the screen. Therefore, when the user locates the ball in the real-world, there should be a UI to guide them to tap the ball. Also, there should be **two** buttons (Up and Down) to adjust the theta angle (θ) of your golf swing. The angle should vary from 0 to 45 degrees. When the ball is hit while the angle is 0 degrees, the ball should simply roll on the ground (putt). The green trajectory in Fig. 3 indicates the example of this functionality when the angle was set 45 degrees. The animated arrow should guide the user to swipe on the screen along its direction. The speed of the swipe can be set as constant.

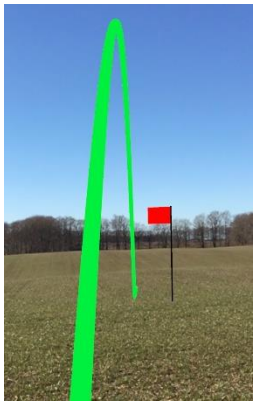
Fig. 3. User Interface example look

Gameplay instructions:

- The play mode should start with the ball placed on the starting position virtual object.
- The ball should follow basic Unity Physics. That is, the ball should cover a natural distance based on the programmed force and the direction selected. The ball should bounce off the bounded real-world objects and the virtual objects placed in the scene.
- If the ball enters any of the teleportation holes, it should teleport and be placed next to the alternate teleportation hole. Therefore, it should be able to both teleport up and down the planes.

Extra credit: Implement the force based on a "swiping" gesture: the longer the finger swipe across the screen, the more the force of hitting the ball.

Submission:

We will be testing your assignment in class. However, you still must submit all your code by the assignment deadline. It is strongly recommended that you test your application and come to class prepared to show your work. If you are unable to show it in class, we will consider the uploaded submission as final and grade the uploaded version. In that case, you must ensure all code and dependencies are sufficiently submitted and documented for the graders to compile, run, and understand your assignment.

For this assignment, include the following for full credit: your Unity project folder, a report, and a video. Your Unity project folder should contain your Unity scene and all the Scripts and Assets you will need to rebuild your project. Please do not include your project executable in the folder.

You will also need to upload a video recording of your application. To do that, turn on the screen video record on your phone and run the application.

Interact with your application and make sure you show **all** the aspects of the assignment, including, but not limited to, position/orientation change, build/play mode, and all required AR objects for this scene. Please note that you must demonstrate your application in all angles with varying distances. To elaborate, the objects must remain in the initial placed position with the same orientation unless the user brings the physical AR device closer or rotates the orientation of the device.

For your report, you should include *at least* the following:

- A title: "CSE 366 Virtual Reality, Spring 2024, Assignment 3: Augmented Reality"
- Your name and Stony Brook ID
- Unity and AR Foundation version
- Hardware used
- Directory hierarchy
- Any extra functionalities/ features that you implemented for this assignment
- Details on implementation: references to the downloaded 3D models or how you designed your own model; how the rotations in your scene were implemented; how you designed your own model; how the controls were implemented; how the golf game was implemented, to mention but a few.