



Drive & Collect: AR Challenge

Documentation of the Second Assignment Intelligent Virtual Environments

Nipuna Weerasekara, Leduin Cuenca

Software and Systems Master Program
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Madrid, Spain
{nipunalakshithawe47, leduin.cuenca}@alumnos.upm.es

January 7, 2024

Contents

1	Introduction	2
2	Play Instructions	2
3	Project Structure	3
3.1	Game Scenes	3
3.2	Assets	3
3.3	Game Objects	4
3.4	Scripting and Game Mechanics	4
3.5	Interaction	4
	References	5

1 Introduction

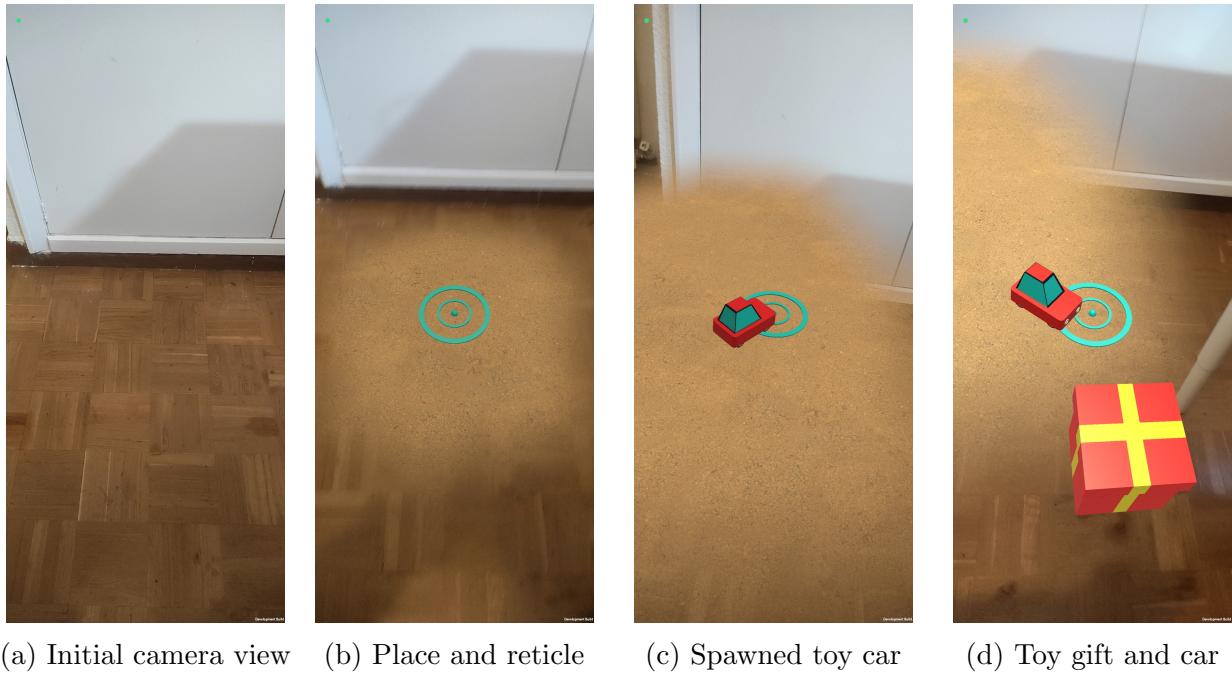
Drive & Collect: AR Challenge is an augmented reality game built using Unity3D and Google ARCore. This game is developed as a Unity game for Android devices. The aim of the game is to introduce the augmented reality to the player and play with augmented reality object while view it in the real world. In this game, the player is asked to point the camera to a horizontal flat surface and move the camera until the game recognizes the environment. Then the player will be able to see a toy car and then the player can move the reticle to move the car in the real world. This AR game is an amusing activity to do while the player is able to explore the real world.

2 Play Instructions

The AR game **Drive and Collect** is a Unity game developed for Android devices. This game is distributed as an APK file to the player. The player has to install the APK file on their Android device in order to play the game.

This AR game relies on the actions of the players in the real world. In another words, the player needs to move the device (in this case, their Android device) to move the car in the game. As for the detailed playing instructions,

1. First, the player opens the game from their Android device.
2. Then, the game will start and show the game splash screen to the player.
3. Then, the player will ask for camera permissions on their Android device.
4. After the player accepts camera permissions, they will see the camera screen (Figure 1a).
5. The player will then point the camera towards a flat surface until a virtual flat surface appears on their screen (Figure 1b).
6. Then, the player will see a green-blue reticle in their screen where it is used to move the car around the real world (Figure 1b).
7. The player will then touch any point on their screen to spawn a toy car and then use the reticle to move the car (Figure 1c).
8. Toy gifts will appear randomly in the screen and the player can move the car and touch the gifts with the toy car and collect the gifts. Once the gift is collected, a new gift will appear randomly in the screen (Figure 1d).



(a) Initial camera view (b) Place and reticle (c) Spawning toy car (d) Toy gift and car

Figure 1: Screenshots from the 'Drive and Collect' game

3 Project Structure

In this section, we discuss the structure of the Unity3D and ARCore project. For developing the AR mobile application, we used **Unity 2022.3.15f1** for Android.

3.1 Game Scenes

In this AR game, we use the **SampleScene** provided by the Google Codelabs for ARCore [1]. In this scene, it access the **AR Session** and **XR Origin** game objects to access the camera of the Android device. The game scene would be the camera view of the device and the real world horizontal surface will be mapped to Unity playing field and over that playing field, we spawn a virtual surface, a toy car, a reticle, and collectable gifts.

3.2 Assets

As the assets in this AR game, we use the assets imported from Codelab starter package [2]. In addition, for the Android APK, we use an icon (Figure 2b) and a splash screen (Figure 2a) designed using DALL-E 3 AI image generation model [3].



(a) Splash screen of the Drive and Collect AR game



(b) Android APK icon for Drive and Collect: AR Challenge game

Figure 2: Designed images for aesthetic purposes

3.3 Game Objects

In this game, we use one game scene **SampleScene** and in that, we use **AR Session** game object to control the lifecycle of AR experience. This game object is part of AR Foundation framework [4]. In addition, we use **XR Session** game object where it is used to access the main camera for the game scene. In this game, this is directly refers to the Android device camera. The next game object we use is **Driving Surface Manager** where it is used to create a virtual flat surface over the real world so that the player can move the virtual car around. The **Car Spawner** game object is used to spawn the car in the augmented screen of the game. We use **Reticle Prefab** to spawn a reticle in the center of the game screen and the user can move the reticle to move the toy car. Finally, **Package Spawner** game object is used to randomly spawn gift packages in the game screen.

3.4 Scripting and Game Mechanics

In this AR game, the player will point the camera of their Android device to the real world and then they will move the camera around to move the toy car to collect gift packages. For this we use Google ARCore and Unity AR Foundation frameworks. In addition, the script **ReticleBehaviour** is used to manage the reticle in the game screen and when the reticle moves, the toy car will follow the reticle within the game screen. For this we use Raycasting feature in Unity. The script will be used to perform a Raycast from the Android device camera's center point each frame, positioning the reticle at the raycast hit location, and it ensures that a child game object is active only when the reticle is positioned on the virtual surface.

3.5 Interaction

In **Drive and Collect: AR Challenge** game, it uses the players camera movements to track the in-game reticle and then move the toy car according to the reticle movement. This game mixes the real world and the virtual world where the real world surface is transformed in to a game surface and the player can drive a toy car towards a point of their choice in the game

surface. To play this game, the player has to move the device camera, effectively making the game an amusing outdoor activity.

References

- [1] *Create an AR game using Unity's AR Foundation.* <https://codelabs.developers.google.com/arcore-unity-ar-foundation#0>.
- [2] *Codelab Unity Starter Package.* <https://github.com/googlecodelabs/arcore-unity-arf-game/raw/main/codelab-starter-package.unitypackage>.
- [3] *DALL·E 3.* <https://openai.com/dall-e-3>.
- [4] *AR Foundation Framework.* <https://unity.com/unity/features/arfoundation>.