POLITÉCNICA

Assignment 1

AR/VR Foundations

Intelligent Virtual Environments:
Technologies, Architectures And Applications

Birk Bregendahl

23/12 2025

1 Introduction

Assignment 1 consists of two tutorial-based projects intended to introduce core Unity and XR foundations:

- Unity project structure (scenes, assets, prefabs, packages)
- Scene composition (cameras, lights, GameObjects, hierarchy)
- XR setup for:
    - VR on mobile via Google Cardboard XR Plugin
    - AR on mobile via AR Foundation (ARKit/ARCore through Unity XR)
- Basic interaction and feedback mechanisms in XR contexts

Scope clarification

The goal of Assignment 1 is not to design a custom VR/AR experience from scratch, but to follow, explore, and document two provided tutorials (VR and AR), and demonstrate understanding of their structure and mechanics.

Assignment parts:

- **Assignment 1.1:** VR (Google Cardboard Quickstart)
- **Assignment 1.2:** AR (AR Mobile Template + AR setup tutorial)

2 VR Tutorial (Assignment 1.1)

2.1 Overview and Purpose

Assignment 1.1 focuses on introducing the fundamentals of Virtual Reality (VR) development in Unity using the Google Cardboard XR Plugin. The purpose of this assignment is not to design a custom VR experience, but to follow, configure, explore, and document an existing tutorial project in order to gain familiarity with Unity's XR workflow, scene structure, camera setup, and basic interaction mechanisms.
The assignment is based on the official HelloCardboard tutorial project provided by Google. Through this project, the core concepts of mobile VR are demonstrated at the project and configuration level, including stereoscopic rendering, head tracking, and gaze-based interaction as implemented by the Cardboard XR Plugin. The outcome of the assignment is a compiled VR application together with documentation describing the project structure and mechanics.

2.2 Project Description

The project is built around the HelloCardboard scene, which is included as a sample in the Google Cardboard XR Plugin for Unity. The scene places the user inside a simple virtual room containing several geometric objects that are configured for gaze-based interaction. Interaction is achieved through a reticle

fixed at the center of the user's view, which responds when the user's gaze remains focused on an interactive object.
The project demonstrates, through its configuration and provided scripts, how a mobile device can be used as a VR platform by rendering stereoscopic images for each eye and updating the camera orientation based on the device's physical movement. The tutorial also covers the necessary project and build configuration required to deploy a Cardboard-compatible VR application.

2.3 Project Structure

Game Scene
The project consists of a single main scene named HelloCardboard, located in the Google Cardboard sample directory. This scene serves as both the entry point and the complete VR experience for the application. It is preconfigured by the Cardboard SDK and does not require manual assembly of XR components.

Assets and GameObjects
The scene contains several key GameObjects that together form the VR experience. At the center of the project is the Main Camera, which represents the user's viewpoint. The camera is configured to be driven by device orientation sensors via the Cardboard XR Plugin, which would allow head-tracked viewing when run on a supported mobile device.
Attached to the camera is the CardboardReticlePointer, which acts as a gaze-based interaction tool. The reticle is configured to interact only with objects placed on a custom-defined Interactive layer. This ensures that only specific objects respond to user focus.
The interactive elements of the scene are grouped under a Treasure GameObject, which contains several child objects with mesh renderers and colliders. These objects serve as visual targets for interaction. The environment itself is defined by static geometry forming a room-like space, providing spatial context and depth cues for the VR experience.
Additional utility objects are included by the SDK to manage internal systems and debugging, but no custom assets were added beyond those provided in the sample.

2.4 Scripting

No custom scripts were written as part of Assignment 1.1. All functionality is handled by scripts included in the Google Cardboard XR Plugin. These scripts manage head tracking, stereoscopic rendering, gaze detection, and interaction logic.
The only scripting-related changes made during the tutorial were configuration-based. This included assigning interactive objects to the Interactive layer and configuring the reticle to detect only objects on that layer. This approach separates interaction logic from scene content and keeps the project modular and easy to extend.

2.5 Interaction and VE Mechanics

Interaction in the HelloCardboard project is designed to be entirely gaze-based. The reticle at the center of the camera continuously emits a ray forward into the scene. When this ray intersects with an object on the Interactive layer for a sufficient duration, an interaction event is designed to be triggered. This method is commonly used in Cardboard applications due to the absence of handheld controllers.

The project does not implement artificial locomotion. The user remains stationary within the virtual environment, and all viewpoint changes are driven by physical head movement. This design choice reduces motion sickness and aligns with best practices for mobile VR.

Feedback in the project is primarily visual. The reticle provides continuous feedback about the user's point of focus, and interactive objects respond visually when targeted. Depth perception is reinforced through stereoscopic rendering, lighting, and shadows within the environment.

2.6 Experimentation

Limited experimentation was performed to better understand the structure and behavior of the VR scene. The dimensions of the room geometry were modified, which changed the perceived scale of the environment and the relative distance to interactive objects. This demonstrated how spatial scale plays a significant role in user perception within VR.

When running the scene in the Unity Editor on a desktop system, the application appeared static, with no ability to look around or move. This behavior is expected, as the Cardboard XR Plugin relies on mobile device sensors such as the gyroscope and accelerometer to update head tracking. These inputs are not available in a standard desktop environment, resulting in a fixed camera orientation when tested outside a mobile device.

2.7 Build and Output

For Assignment 1.1, the project was successfully built and exported as an Android application package (.apk) for use with a Google Cardboard viewer. A desktop standalone build (.app) was also generated for inspection and limited execution. Additionally, the complete Unity project sources were packaged for submission.

2.8 Summary

Assignment 1.1 provided a practical introduction to VR development in Unity using the Google Cardboard XR Plugin. By following and configuring the HelloCardboard tutorial project, the assignment demonstrated core concepts at the configuration and project-structure level, such as XR setup, scene organization, gaze-based interaction, and mobile VR deployment. The project establishes a foundational understanding of Unity-based VR workflows that can be expanded upon in subsequent assignments.

3 AR Tutorial (Assignment 1.2)

3.1 Overview and Purpose

Assignment 1.2 focuses on exploring Unity's AR Mobile Project Template and understanding how mobile augmented reality applications are structured using AR Foundation. The work follows two official Unity tutorials: Configure your AR development environment and the AR Mobile Template Quick Start Guide.
The goal of this assignment is not to design a custom AR application, but to explore a pre-configured AR project, understand how AR sessions, tracking, interaction, and UI are organized, and to experiment with the provided interaction mechanisms. The project was explored primarily in the Unity Editor, using the provided template scenes and XR Simulation, with built-in support for deployment to mobile platforms.

3.2 Project Setup and Environment

The project was created using Unity's AR Mobile Core template, which automatically configures the required packages and project settings for mobile AR development. The template includes AR Foundation, ARKit XR Plugin, ARCore XR Plugin, XR Interaction Toolkit, and XR Plug-in Management, enabling cross-platform AR development for both iOS and Android devices.
The development environment supports both iOS and Android workflows. Since the template abstracts most platform-specific details through AR Foundation, the same project structure and scenes can be explored in the Editor and later deployed to a physical device if needed.

3.3 Scene Structure

Two scenes were used during the assignment.
The first scene, AR Template Scene Birk, is a copy of Unity's default AR template scene. This scene serves as a minimal AR starting point and contains a rotating 3D object placed relative to the AR camera. It demonstrates the core structure required for an AR scene, including an AR Session and an XR Origin (AR Rig), without additional UI or interaction layers.
The second scene, SampleScene_BirkCopy, is based on the AR Mobile Template's sample scene. This scene contains a fully configured AR experience with interactive object placement, screen-space UI, coaching prompts, and debugging tools. It was used as the primary exploration and experimentation environment.

3.4 Core GameObjects and Components

At the heart of both scenes is the AR Session, which manages the lifecycle of the AR experience, including communication with the underlying AR subsystem and session configuration. Only one AR Session is active at any time.

The XR Origin (AR Rig) represents the mobile device in the virtual scene. As the user moves the device in real space, the XR Origin is designed to update its position and orientation accordingly. The AR camera is a child of the XR Origin and is responsible for rendering the camera feed and virtual content.
Detected real-world elements, such as planes and surfaces, are stored under the Trackables GameObject. These trackables are dynamically generated during runtime based on sensor input or simulation data.
The Object Spawner component enables the placement of virtual objects into the AR environment. It works in combination with raycasting and input events to instantiate selected prefabs at valid positions in the scene.

3.5 Interaction Design and VE Mechanics

Interaction
Interaction in the AR Mobile Template is primarily screen-based. Users interact with the scene through tap gestures, which are processed via the XR Interaction Toolkit. Tapping on the screen triggers raycasts into the AR scene, allowing objects to be placed on detected surfaces.
Once placed, objects can be selected, moved, and deleted using on-screen UI controls. Selection is visually indicated, and contextual UI elements appear when an object is active.

Locomotion
There is no artificial locomotion system in the project. Movement through the AR environment is conceptually achieved by physically moving the device in real space. When running in the Unity Editor, this behavior is simulated using XR Simulation, which allows the camera to be moved using mouse and keyboard input.

Feedback
Visual feedback is provided through multiple channels:

- Plane visualization indicators show detected surfaces.
- UI prompts guide the user through scanning, placement, and manipulation actions.
- Selected objects are highlighted, and interaction states are clearly communicated through UI changes.
- The AR Debug Menu provides runtime feedback such as tracking quality and performance information.

3.6 Experimentation and Modifications

Several controlled experiments were conducted to explore the flexibility of the template.

In AR Template Scene Birk, all three optional challenges were completed. The rotation speed and direction of the default object were modified by adjusting the parameters of the rotation script. The rotating cube was then replaced with a Torus primitive by copying the relevant components and configuration from the original object, demonstrating how behavior can be transferred between GameObjects.

In SampleScene_BirkCopy, the interactive AR experience was explored extensively. Multiple objects were placed, selected, moved, and deleted using the provided UI. The AR Debug Menu was enabled to inspect runtime information and understand how tracking data is exposed during execution..

As an additional experiment, one of the placeable prefabs in the Object Spawner was replaced with a PushButton object. After updating the prefab reference, the new object could be successfully spawned and interacted with in the AR scene, confirming that the spawning system is modular and easily extensible.

3.7 Build and Output

For Assignment 1.2, the following deliverables were produced:

- A standalone desktop build (.app) demonstrating the AR Mobile Template functionality
- The complete Unity project source, including modified scenes and assets

While the project supports deployment to mobile devices, exploration and testing in the Unity Editor were sufficient to meet the learning goals of the assignment.

3.8 Summary

This assignment provided a practical introduction to Unity's AR development workflow. By working with a pre-configured AR template, it became clear how AR sessions, tracking, interaction, and UI layers are structured and connected. The experimentation tasks demonstrated how existing AR behavior can be modified and extended without building systems from scratch, highlighting the modular design of AR Foundation and the XR Interaction Toolkit.

4 Brief Summary of Work Performed

For VR (1.1), the Google Cardboard HelloCardboard sample was imported, configured, and inspected to understand how a mobile VR scene is structured in Unity: stereoscopic VR rendering and head tracking are provided by the Cardboard XR Plugin, while gaze-based interaction is implemented through the CardboardReticlePointer and an "Interactive" layer mask. Full head-tracking behavior is device-sensor dependent and therefore not fully observable in the desktop Editor.

For AR (1.2), Unity's AR Mobile Template was created and examined to understand AR Session / XR Origin setup, plane/trackable structure, screen-based interaction flows, and UI feedback (including the AR Debug Menu). Minor experimentation was conducted (scene edits and prefab swapping) to verify understanding of scene structure and the modular interaction/spawning setup.

Both projects were evaluated primarily at the configuration, scene-structure, and interaction-design level, in line with the tutorial-based learning goals of Assignment 1. Desktop builds were provided as macOS .app files, which the instructor confirmed as acceptable alternatives to Windows .exe builds.