CMP3754M Virtual and Augmented Reality

Workshop 10 : Plane Detection and Object Placement

Objectives:

1. Add plane detection
2. Use ray casting to detect screen touch point on plane
3. Create and delete objects on a detected plane

Duration: 1 hour
Platform: *Unity 6 (version = 6000.0.55f1 LTS)*

By completing this workshop you will build functionality that you can use to complete your second assessment for this module

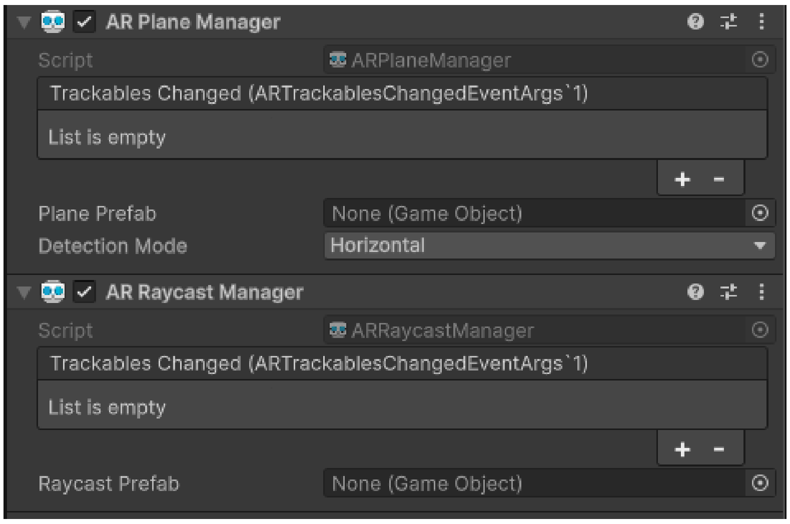
1. Introduction

In this workshop we are going to learn how to detect planes in the real world, and then use them as an anchor for placing virtual objects. Later we will add more functions to control and interact with these objects.

2. Detecting the Planes

Let's start by creating a new folder: **Assets>CMP3754>Planes**.

Next, select the XR Origin object in the hierarchy, and add an **AR Plane Manager** component, and an **AR Raycast Manager** component. In the Plane Manager, set the Detection Mode to Horizontal:


AR Plane Manager
Script: ARPlaneManager
Trackables Changed (ARTrackablesChangedEventArgs`1)
List is empty
Plane Prefab: None (Game Object)
Detection Mode: Horizontal
AR Raycast Manager
Script: ARRaycastManager
Trackables Changed (ARTrackablesChangedEventArgs`1)
List is empty
Raycast Prefab: None (Game Object)

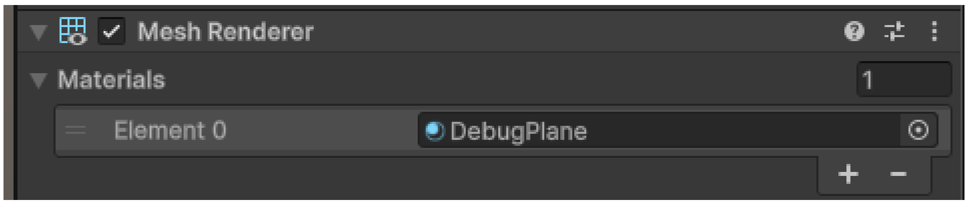
Next we need a plane object that we can use to represent the detected planes. Right click in the Hierarchy, and select **XR > AR Default Plane**. This creates a default plane object in the scene.

Drag the new AR Default Plane object into the Project directory **Assets>CMP3754>Planes** to create a prefab. Delete the object in the Hierarchy. Select the XR Origin object again, and drag the new plane prefab into the Plane Prefab field of the AR Plane manager component.

Run this in the editor. As you move the view around, you should see a yellow plane expand across the floor in the simulated environment.

You can now also test this on the device. You should see the same yellow plane expand across the floor, or other horizontal surfaces, as you move the phone around.

When you have finished testing this, return to the Project window, and inspect the AR Default Plane prefab. If you expand the Materials list of the Mesh Renderer Component, you will see a material listed called "DebugPlane":


Mesh Renderer
Materials 1
Element 0 DebugPlane

This is the yellow plane material, located in **Packages > AR Foundation > Assets > Materials**. If you want, you can:

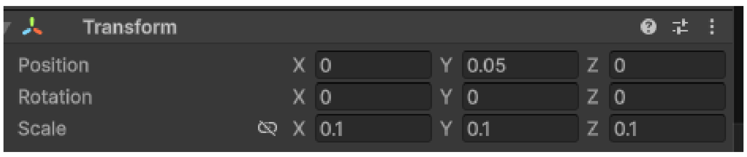
- Create a new material of a different colour, and use it in place of the DebugPlane material. This works best if you use a transparent surface type for the material.
- Or, remove the DebugPlane material completely: in this case, the planes will just be rendered as the edge lines.

3. Creating Objects

Next, we are going to use touch control inputs to place objects on detected planes. To do this, first add an Event System object by right-clicking in the Hierarchy, and selecting **UI > Event System**.

Next, we will create an object to instantiate.

In the Hierarchy, create an empty object called PlaneObject. Create a Cube as a child object. Scale the cube to (0.1, 0.1, 0.1), and offset it by 0.05 along the y-axis:


Transform

	X	Y	Z
Position	0	0.05	0
Rotation	0	0	0
Scale	0.1	0.1	0.1

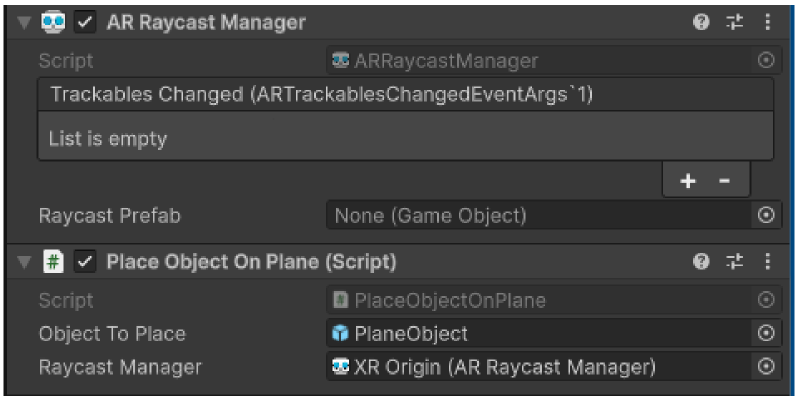
In the **CMP3754M>Planes** directory, create a material called PlaneObjectMat, and pick a colour for it. I made mine purple. Drag it onto the cube.

Disable the Box Collider on the Cube, and instead, add a new Box Collider component to the parent PlaneObject. You'll need to scale this box collider to (0.1, 0.1, 0.1), and offset it by 0.05 on the y-axis, so that it is aligned with the child cube object.

Drag the PlaneObject from the Hierarchy to the **Assets>CMP3754>Planes** directory in the Project window, to create a prefab. Delete the original object from the Hierarchy.

I have provided a C# script to handle the touch input detection, and object creation. Download the **PlaceObjectOnPlane.cs** script, and place it into the **Assets>CMP3754>Planes** directory.

Add this script to the **XR Origin** object. Drag the AR Raycast Manager from the XR Origin object into the **Raycast Manager** field of the script. Drag the PlaneObject prefab from the **Assets>CMP3754>Planes** directory in the Project window to the **Object To Place** field.


AR Raycast Manager
Script: ARRaycastManager
Trackables Changed (ARTrackablesChangedEventArgs`1)
List is empty
Raycast Prefab: None (Game Object)
Place Object On Plane (Script)
Script: PlaceObjectOnPlane
Object To Place: PlaneObject
Raycast Manager: XR Origin (AR Raycast Manager)

Now have a look through the code in the script. It is important that you understand how this works, as you'll likely want to edit it to complete your assignment.

The editor AR simulator can't simulate the phone touch inputs, so you'll need to deploy it directly to the phone for testing. You should find that:

- If you touch a position on a detected plane, a cube object is created.
- If you touch an existing cube, it is deleted.

4. Conclusion

We now have the building blocks of an AR application. In the next tutorial, we will add some UI elements.