# Programming HW 4

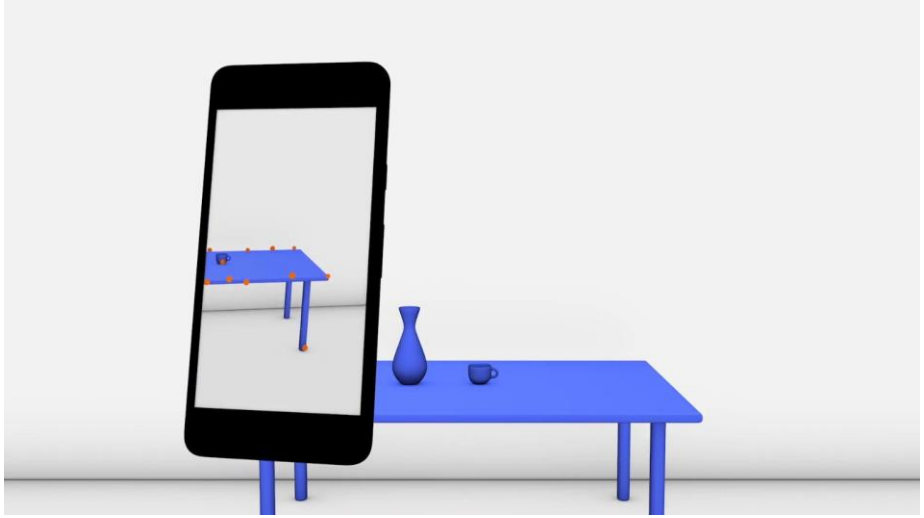
Cloud Anchors

# What is a Cloud Anchor?

- Using Cloud Anchors, your app lets users add virtual objects to an AR scene.
- Multiple users can then view and interact with these objects simultaneously from different positions in a shared physical space.
- To enable these shared experiences, ARCore connects to the ARCore Cloud Anchor service to host and resolve anchors. This requires a working Internet connection.

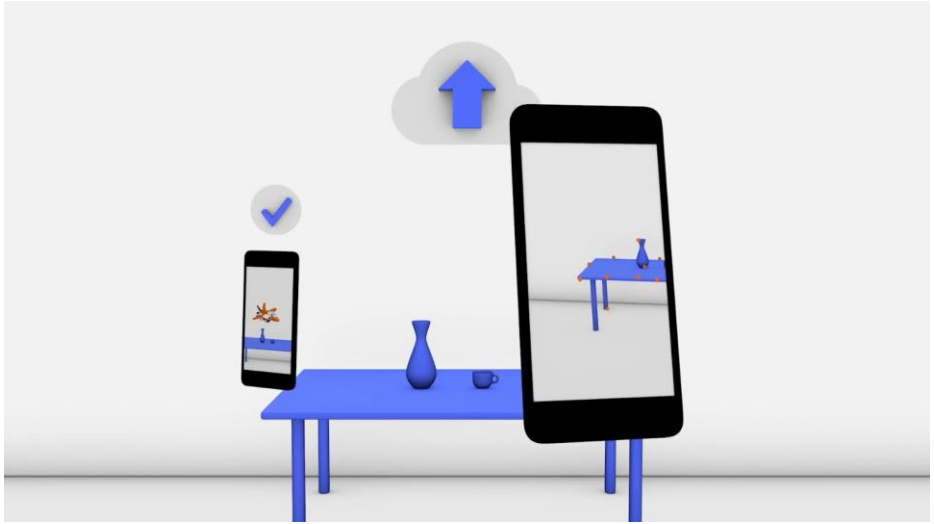
# How to host a cloud anchor

1. **Wait a few seconds after the session starts** to give tracking time to stabilize before attempting to host an anchor.
2. When selecting a location to host the anchor, try to **find an area with visual features that are easily distinguishable from each other** -- for example, a corner with visually distinct features.
3. **Point the rear device camera at the center of interest**, that is, the area surrounding the point where you want to place the anchor.
4. While keeping the camera trained on the center of interest, and while roughly maintaining the physical distance between the device and the center of interest, **move the device around to map the environment from different viewing angles and positions for up to 30 seconds**. Walking around in the space while keeping the device camera trained on the center of interest will enable capturing visual features of the area of interest from all angles, making resolving more robust.



# How to resolve a cloud anchor

1. **Wait a few seconds after the session starts** to give tracking time to stabilize before attempting to resolve an anchor.
2. In the same environment as the hosted anchor, **scan the original area of interest**, ensuring that:
    a. The device camera has a clear line of sight to mapped area
    b. The device camera is a similar distance from the hosted anchor as the device that originally hosted the anchor.
3. ARCore continuously polls the ARCore Cloud Anchors API, sending visual data to the ARCore Cloud Anchor service.
4. The ARCore Cloud Anchor service compares visual features from the scene against the 3D feature map that was created. When it finds a match, **the service returns the pose of the Cloud Anchor**.



# Task

- Open Unity, and load the project at the root of the given folder. Open scene ‘PHW4/PHW4_scene’
- Install ARCore Extensions Package (ver 1.19, not 1.20)
  - https://github.com/google-ar/arcore-unity-extensions/releases/
- Check the unity setting refer to TA’s ARCore Tutorial & Additional Guidance
- (Cloud Anchor works on both Android and iOS)

1. [TO DO #1] Set up development environment for Cloud Anchors

1. [TO DO #2] Spawn target prefab

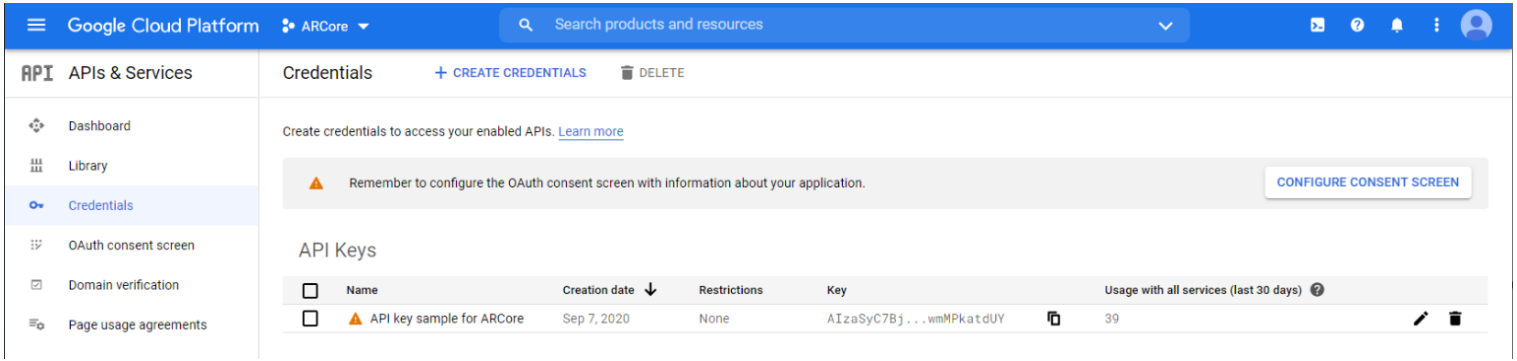
1. [TO DO #3] Launch the app

# 1. Set up development environment for Cloud Anchors

- Import version >=1.0.4 of the Multiplayer HLAPI package
  - Choose Window > Package Manager, and install Multiplayer HLAPI package
  - (Select Unity Registry in the dropdown menu of Package Manager if you don’t see the package)
- Enabled the developer options and debugging on your phone
  - https://developer.android.com/studio/debug/dev-options
  - Instructions may differ depending on your device

# 1. Set up development environment for Cloud Anchors

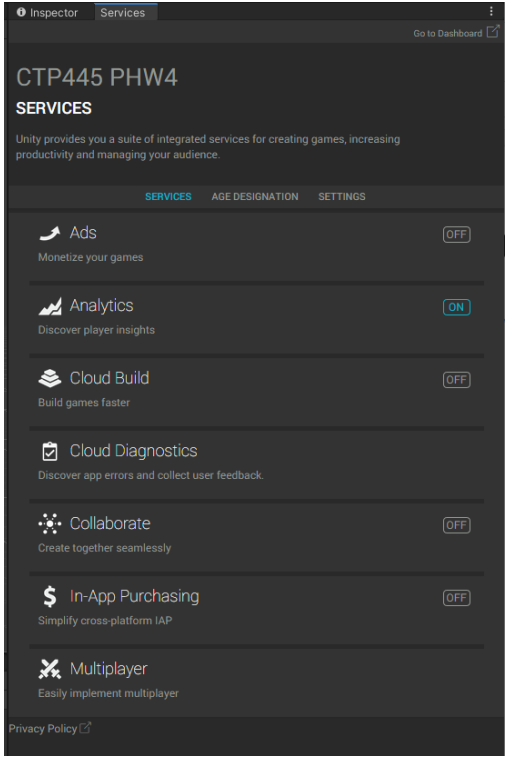
- Add an API Key
  - Obtain an API Key (link)




- Enable the ARCore Cloud Anchor API for your Google Cloud Platform project
  - https://console.cloud.google.com/apis/library/arcorecloudanchor.googleapis.com
- Add your API key to your project:
  - In Unity, go to Edit > Project Settings > XR > ARCore Extensions
  - Add your API key to the Cloud Anchor API Keys field

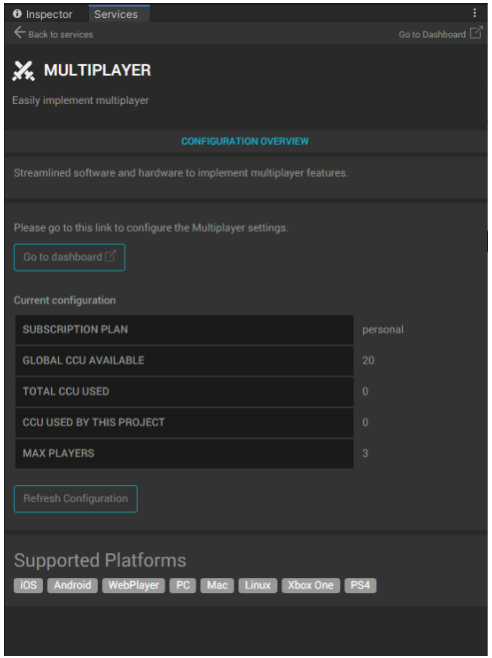
# 1. Set up development environment for Cloud Anchors

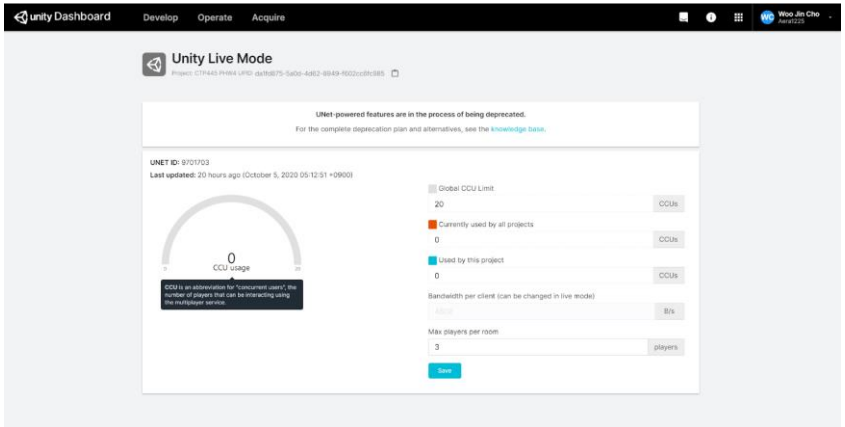
- To be able to use multiplayer services, you must set up project services and set up multiplayer.
- https://docs.unity3d.com/Manual/SettingUpProjectServices.html
  - Go to Window > General > Services
  - Create a new Project ID
  - Follow the instruction
  - You might see Service tab like figure (for version 2019.4.9f1)



# 1. Set up development environment for Cloud Anchors

- https://docs.unity3d.com/Manual/UnityMultiplayerSettingUp.html
  - Selecting Window > General > Services in the menu bar. In the Services window, select Multiplayer.
  - Click **Go to dashboard** button and set up the new multiplayer configuration
  - Once you have clicked Save, the Multiplayer Services Dashboard reflects your current project.



# 2. Spawn target prefab

- Prior to PHW4, we provided target script, target function for implementation.
- Now you need to find proper script to edit and proper class function yourself.

- As shown in the example figure,
  Find and implement the necessary code for the star prefab to be created by tapping the screen after hosting the cloud anchor.

# 3. Launch the app - Host

- Create a room with the host device. The app will display a room code which can be used to join this room from another device.
- Move your phone around until ARCore starts detecting and visualizing planes.
- Tap a plane to create an anchor object on it. A host request is sent to the ARCore Cloud Anchor service. The host request includes data representing the anchor's position relative to the visual features near it.
- After the Cloud Anchor is successfully hosted, subsequent taps will create Star objects, which will be synchronized across devices using Unity Multiplayer Services.

- If the app failed to host the anchor, re-install the application, and execute it again.

# 3. Launch the app - Client

- **Launch the example from a new device, and join the room you created previously.**
- **View the same location as the host device did.**
    - **Check slide #14**
- **A request is sent to the ARCore Cloud Anchor service to resolve the pose of the cloud anchor. When this request returns successfully, the sample app renders the anchor using the pose.**
- **Both devices will now be able to tap the screen to place Stars.**

# Deliverables

1. Record each screen running the app on your host & client device
2. While running the app, include features as below,
    a. Create cloud anchor in host device
    b. Create multiple stars in host device
    c. Observe the star being created by the host in the client device

1. Submit video file as .zip file on KLMS

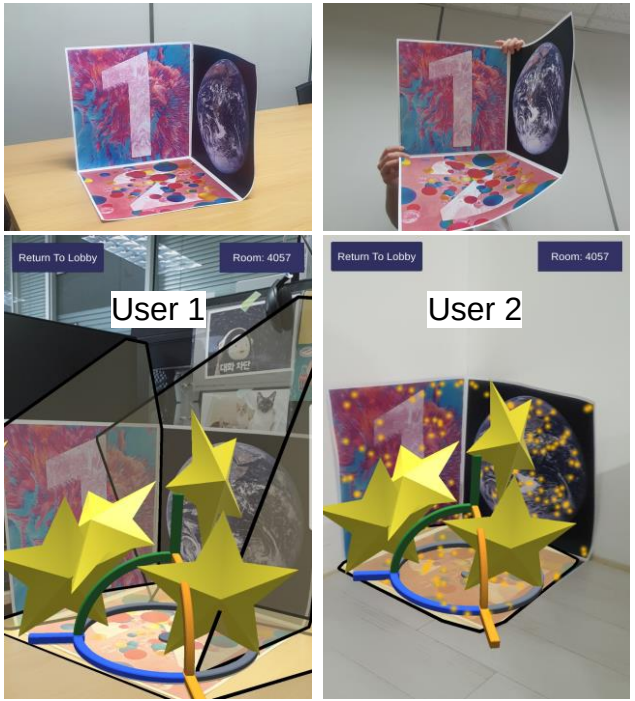
**Release date : 10.12**

**Due date : 10.18 11:59**

If you can’t test the cloud anchor on the same site as your team (i.e. you are abroad or such), there are two options.

1. (On your own) Use 2 different devices, each being host and client.
2. (With your team) Make a half cube out of printed images with strong features and use it as if it is a small shared space →
    - The images you used in PHW3 will do.
    - Minimize blank areas on the paper.

This is just a workaround. Testing on the same site is strongly recommended.



# References


- https://developers.google.com/ar/develop/unity-arf/cloud-anchors/overview