

Immersal City-scale Demo App Tutorial

This is a simple Demo App for Unity, using the Immersal SDK, AR Foundation and a real-time database server. The purpose of the app is to demonstrate persistent AR with basic multi-user capabilities. The app is divided into two parts: the Automatic Mapper view, where any real-world environment can be quickly scanned, and the Content Placement view, where simple AR content can be placed in the previously scanned environment. All devices see the same content at the same place. The content can also be moved by dragging, and the new position is updated to all devices in real-time.

The app is currently using Immersal's proprietary localization service, but will be later integrated to work with the Open Spatial Computing Platform (OSCP).

We recommend using the Unity 2019.4 LTS version. The sample code should build with newer versions, too, but hasn't been thoroughly tested. This tutorial won't go into details about how to use Unity, it is assumed that the user has some prior knowledge of it.

To build the project in Unity, follow the following steps:

Step 1 Create a free account at

<https://developers.immersal.com/>

After creating an account, you will be logged in to the Immersal Developer Portal.

Step 2 Clone the Demo App repository from GitHub

Go to <https://github.com/immersal/city-scale-demo-app-2021> to download the repository.

Step 3 Open the project in Unity

Open the cloned / downloaded GitHub project in Unity. Load the *Assets/Scenes/CityScaleDemo* into the Unity Editor. Switch the build platform to either iOS or Android, and you are ready to build the project.

1 Using the Demo App

When the app is first launched, you are faced with a login screen. Enter the same credentials you used to register on the Developer Portal. Leave the server URL as is.

After logging in, you can select between the automatic mapping view, or the content placement view. If you have already created some maps, you would probably go straight to the content placement, but as this is our first time using the app, we will start with the mapping view.

Before we start, here's a video of the demo app (a slightly older version) in action:

<https://youtu.be/GI9s13zpiAg>

1.1 Automatic Mapping

Because a map construction on the cloud service can sometimes take a lot of time, this demo app features the “automatic mapping” mode, which will only capture 40 images of your surroundings. This way the map construction will only take about 10 seconds, so the map will be ready to use for content placement almost immediately. This way of mapping is suitable for room-size environments, such as your living room or something similar. For capturing larger (even city-scale) areas, you’d use the Immersal Mapper app, see Chapter 3.

Tap the “Start Capture” button to start a mapping session. The app will automatically capture a new image after you move a bit. Walk around the room, quite slowly and holding the smartphone steadily, and try to capture images from the areas where you would imagine people using the app and viewing AR content. Do not move the device abruptly; the images need some overlap between each other to construct a usable map. 40 images is not a lot, you may run out of time if you try to capture the whole room. If you feel like you could do better, start over and give it a few runs before actually saving the map. You can cancel or stop capturing during the session using the “Cancel” and “Stop Capture” buttons. During mapping, the feature points of the environment are represented as a *point cloud*.

When you are ready to save your map, give it a name and tap “Submit”. If you want other people to be able to use your map, you can also make it public.

After the map construction has been started, you can switch to the content placement mode. The map may not be immediately ready for use, so give it 10 seconds or so -- the map name will appear in the dropdown menu, where you can select it and load it when it’s ready. You can also see the map and its processing status in the Developer Portal.

1.2 Content Placement

When entering the Content Placement view, it first says “Please wait while loading”; it is trying to find matching feature points from your camera data. The text will go away when the map is localized the first time. You are now ready to place content in the scene.

The view has two buttons at the bottom of the screen: the left one adds an AR object (a rotating cube) into the scene, and the right one clears the whole scene (that is, removes all the placed objects). If you start the app on a different device and log in with the same credentials, you will see the same content at the same place. Or, if you made your map public, it doesn’t matter which credentials you use. You can also hold and drag to move the object; it will be updated in real-time on all devices watching the AR content.

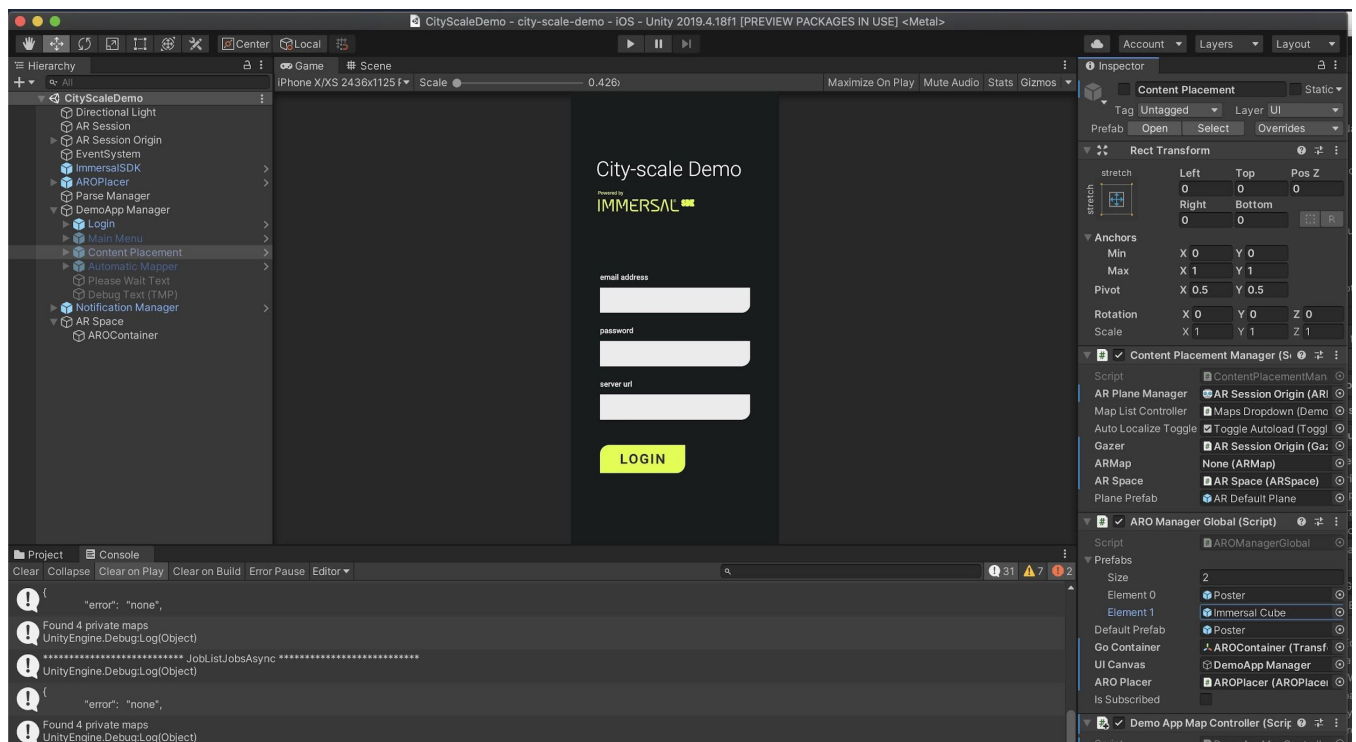
The map, or captured feature points, will be shown as a point cloud in the scene. This can be toggled off in the Unity project, editing the *DemoAppMapController.cs* script (*LoadMap()* function).

On the top-right corner you can see a tracking status indicator; when it's green, it means that the device has been successfully localized against the map.

2 Customizing

This is where the fun starts! The app contains a simple 3D asset, a rotating cube. You can change this to be your own object. Or you could add more buttons to the UI, where each button would spawn a different object.

In the picture below, you can see where to change the AR object prefab. You can create your own content in Unity or use an existing object.



3 Mapping larger areas using the Immersal Mapper

You can capture a lot larger areas using the Immersal Mapper app, available for both iOS and Android.

App Store: <https://apps.apple.com/us/app/immersal-mapper/id1466607906>

Play Store: <https://play.google.com/store/apps/details?id=com.immersal.sdk.mapper&hl=en&gl=US>

If you use the same credentials when logging in to the Mapper app, you can use these maps in the demo app also.

Please refer to Immersal's online documentation about using the Mapper app, and best practices when mapping:

<https://immersal.gitbook.io/sdk/unity/sample-scenes/mapping-app-sample#using-the-mapping-app>