

Device Map

Operation Manual



All information provided in this document is subject to change without notice and does not represent a commitment on the part of &U ASSETS. The software described by this document is subject to a License Agreement and it is not meant to be copied to other media. No part of this document may be copied, reproduced or otherwise transmitted or recorded, for purposes other than the explicit by the customer, without prior written permission by &U ASSETS.

© Copyright Assets, 2016. All rights reserved.



Content

1	Overview.....	4
2	Project Structure	5
3	Building the Demo Project	6
4	Adding MapKit.framework in XCode	9
5	Script Examples	12
5.1	Place an Annotation	12
5.2	Show and Hide the Map	12
5.3	Go to Address	12
6	Unity Interface.....	13
6.1	Native Map Settings	14
6.2	Native Map Settings	14
6.3	Map Orientation.....	15
6.4	Map Positioning.....	15

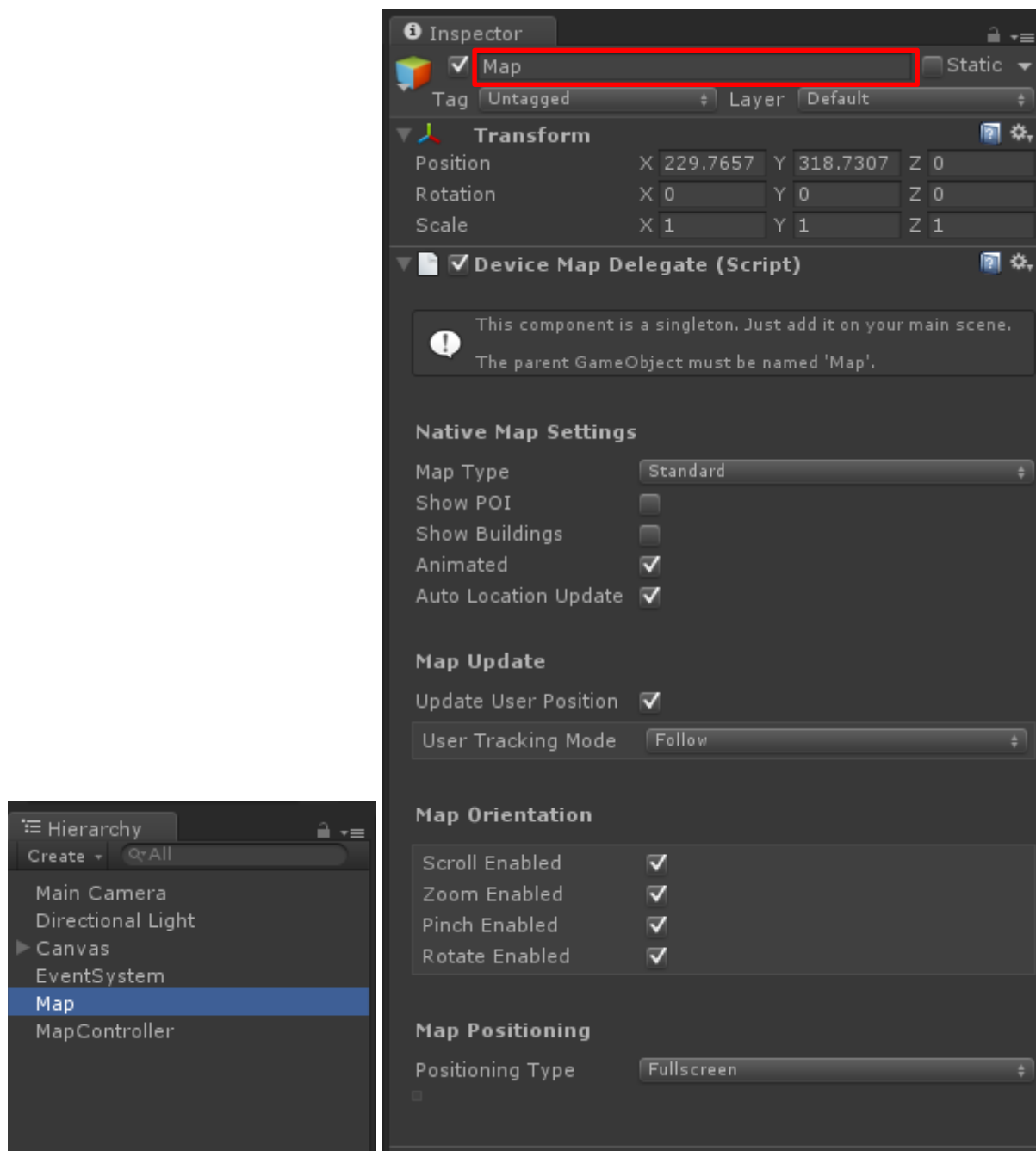
1 Overview

The **Device Map** plugin for Unity provides a full map integration for iOS Devices. The plugin uses the built-in Apple Map, which allows access to all functionality that apple provides. Device Map allows you to create fully responsive (drag, zoom, pan, pinch, scroll...) Standard-, Satellite- and Hybrid-Maps, with the ability to change the map's appearance at runtime.

2 Project Structure

Every project that includes the Device Map plugin must contain a game object named **Map** with an attached DeviceMapDelegate component.

The component is a singleton instance and persists over the complete lifetime of the application.



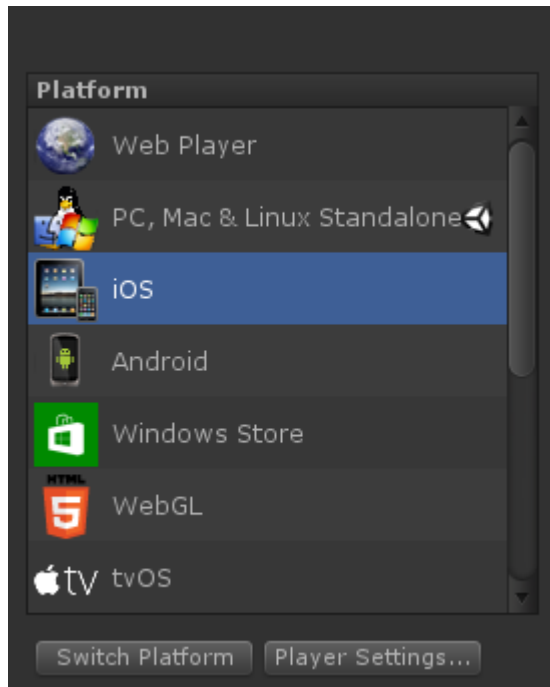
It's important to name the game object **“Map”** because all native messages get sent to this component. No other game objects can exist named **“Map”**.

3 Building the Demo Project

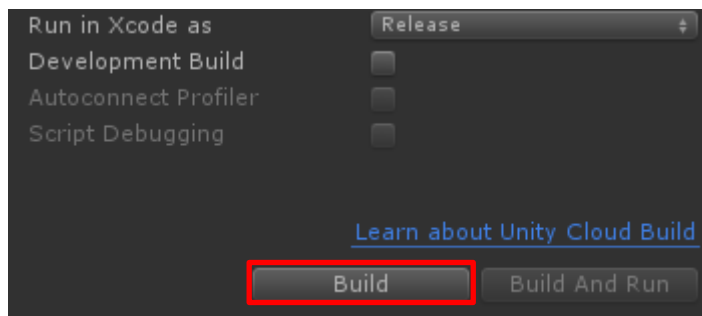
Follow the steps below to build the included demo project.



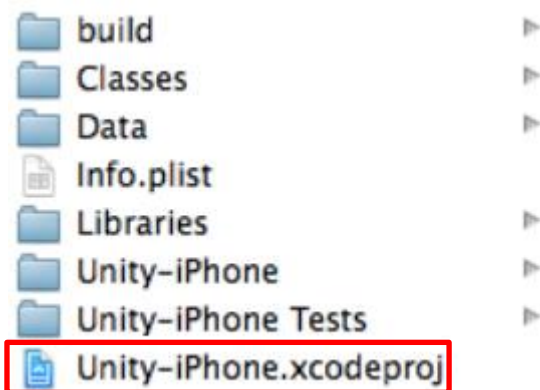
- 1 In Unity, open the demo scene located in the Assets\Demo\Scenes Folder.
- 2 Open the build dialog File->Build Settings and select the iOS platform as build target.



- 3 Press Build.



- 4 The folder you built the project in should pop-up and contain the xcode project file. Open the *.xcodeproj file in XCode.



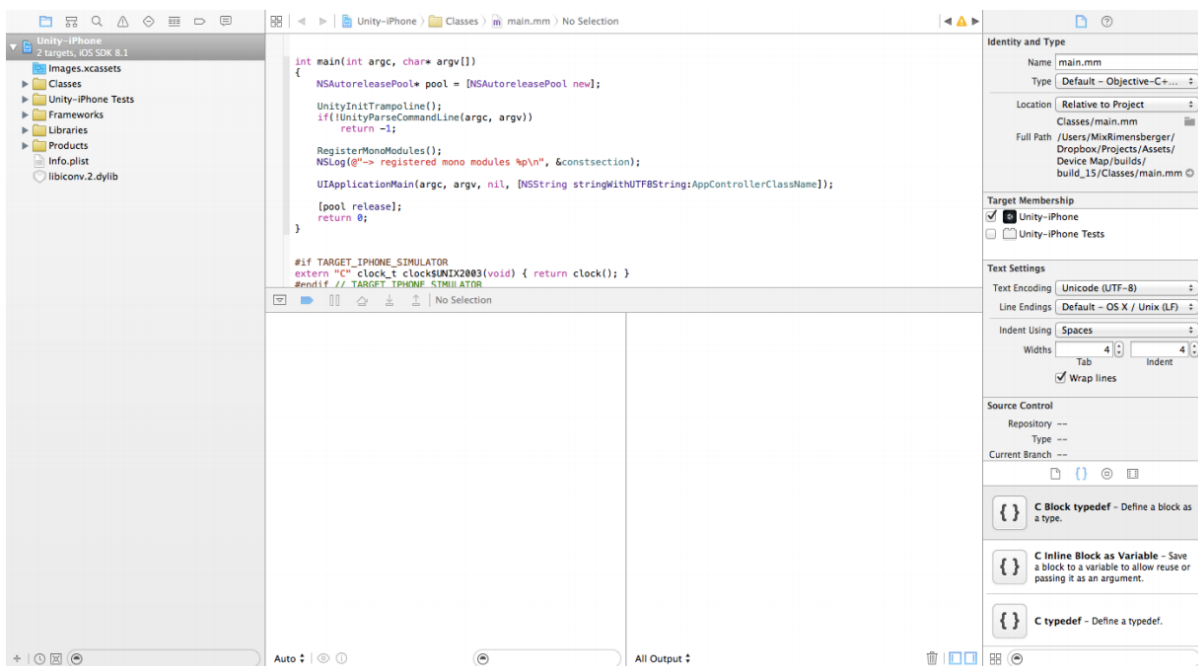
All needed frameworks are included automatically. If the MapKit framework should be missing for some reason read the [MapKit](#) chapter.

4 Adding MapKit.framework in XCode



Note that this step is normally not required as the MapKit framework should be automatically added to your XCode project.

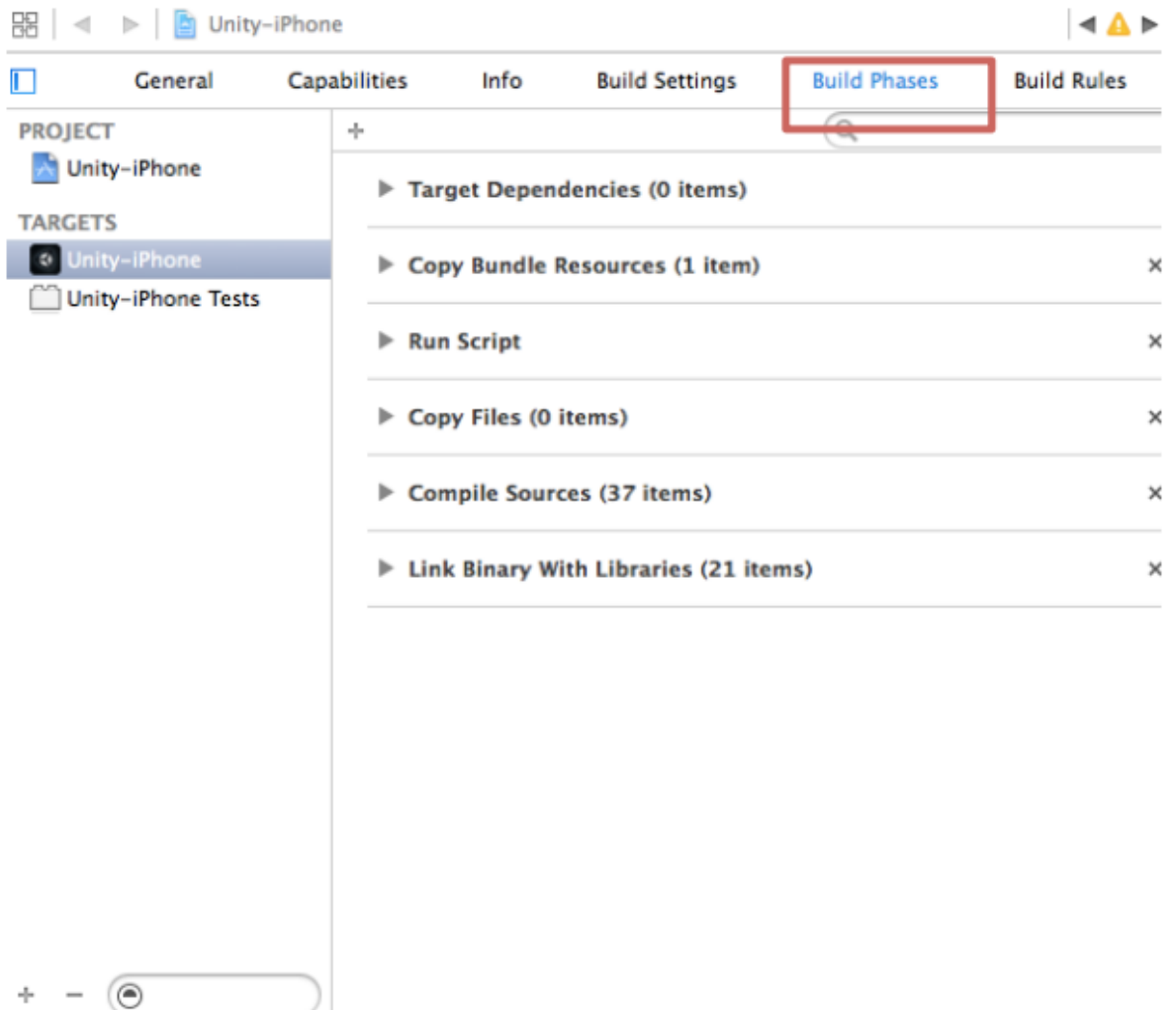
1 Open the xcode project. Your xcode view should look more or less like this:



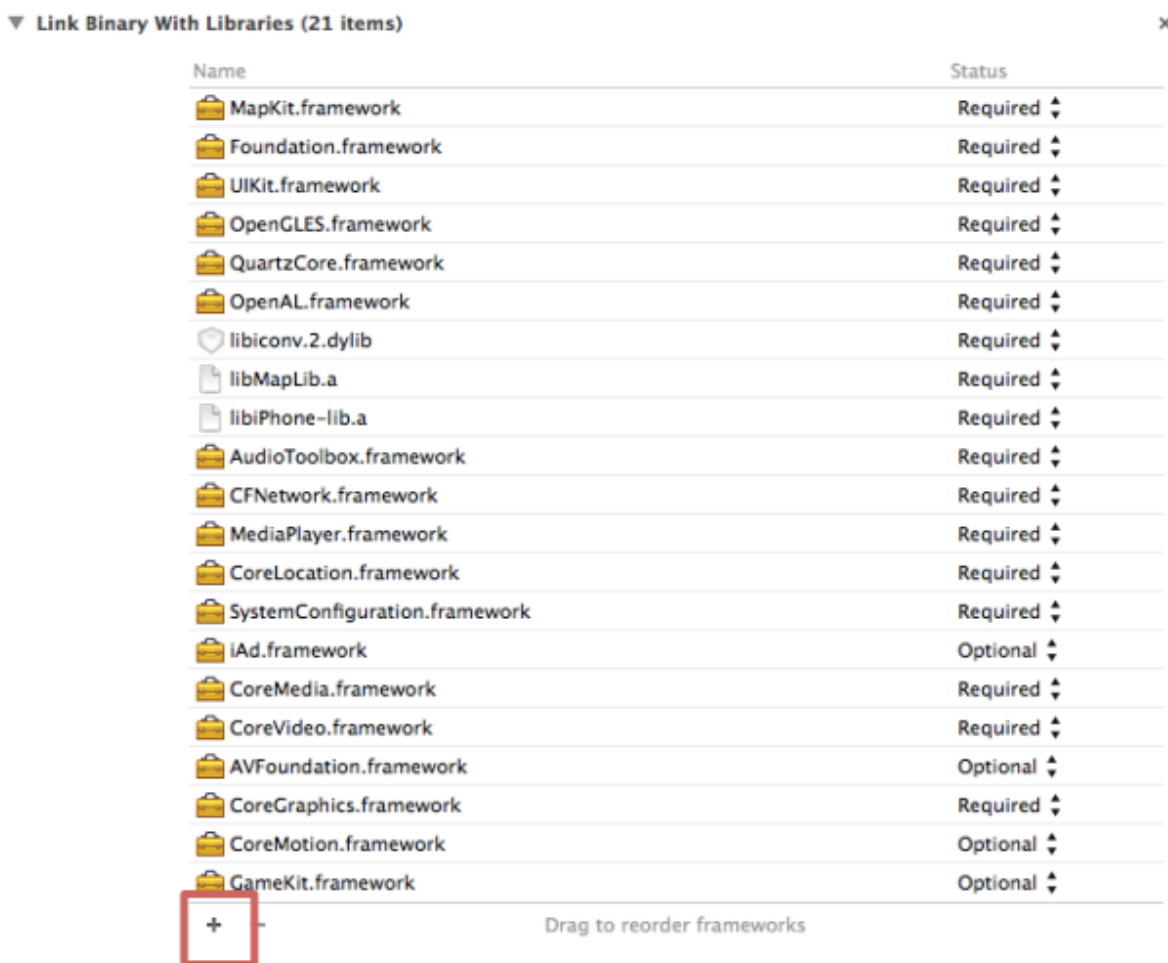
2 Open the Project-Editor by double-clicking on Unity-iPhone in the left panel



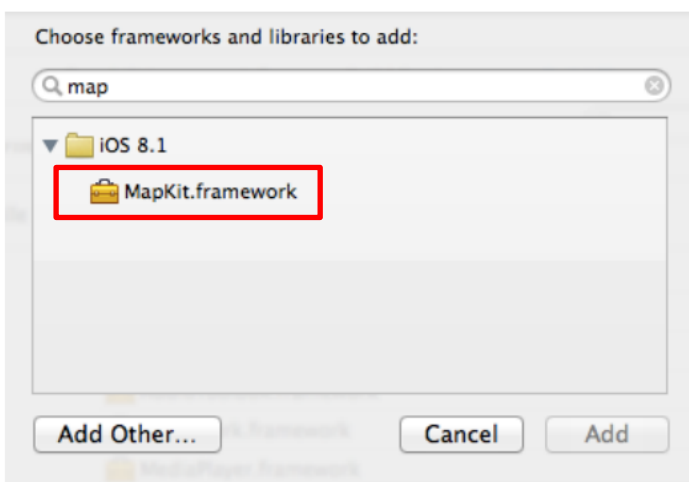
3 In the horizontal Menu, navigate to Build Phases



- 4 Open up *Link Binary With Libraries* – a list of the attached frameworks opens up
- 5 Click to the + icon in the bottom-left corner of the dialog



- 6 Search for MapKit.framework, select it and click the Add Button.



5 Script Examples

5.1 Place an Annotation

```
public void PlaceAnnotation(string title, string desc)
{
    MapAnnotation annotation = new MapAnnotation();
    annotation.AnnotationTexture = AnnoationTexture;
    annotation.Title = title;
    annotation.Description = desc;
    annotation.Offset = Offset;
    annotation.Coordinate = DeviceMap.LastScreenCenterLocation;

    DeviceMap.AddAnnotation(annotation);
}
```

5.2 Show and Hide the Map

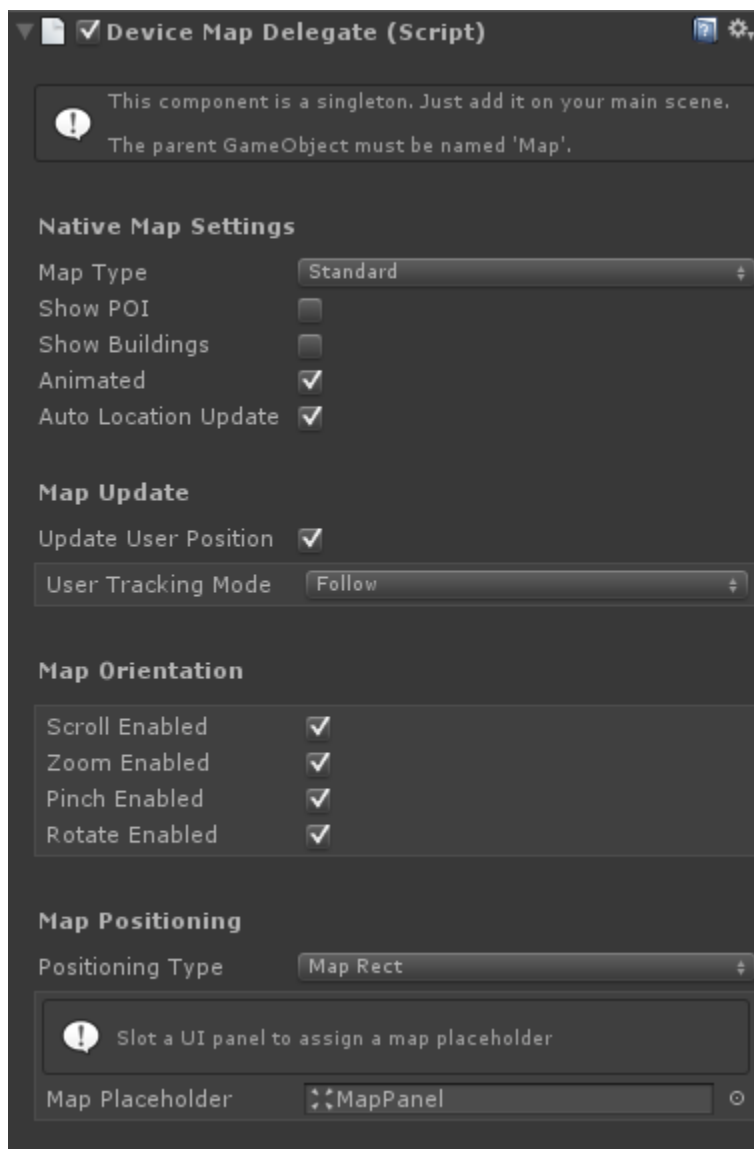
```
public void SetMapVisibility(bool visible)
{
    if(visible)
    {
        DeviceMap.ShowMap();
    }
    else
    {
        DeviceMap.HideMap();
    }
}
```

5.3 Go to Address

```
public void GoToAddress(string address)
{
    if (DeviceMap.Initialized)
    {
        DeviceMap.GoToLocationByAdress(address);
    }
}
```

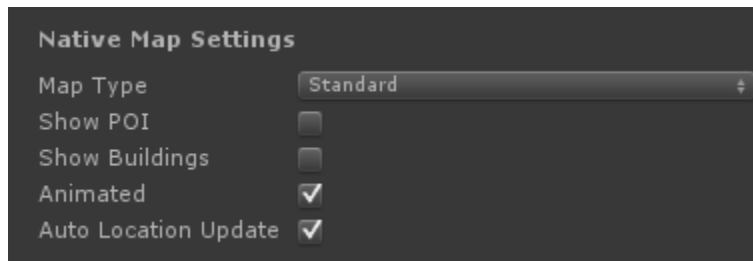
6 Unity Interface

The appearance of the map gets controlled by the DeviceMapDelegate component.



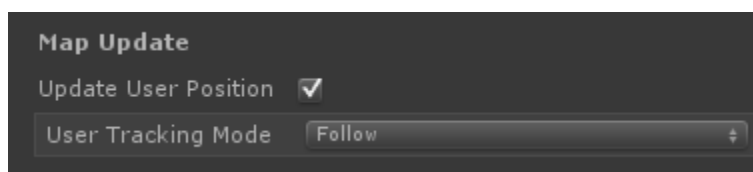
Note: The DeviceMapDelegate is a singleton component and therefore it only has to be placed in the start scene.

6.1 Native Map Settings



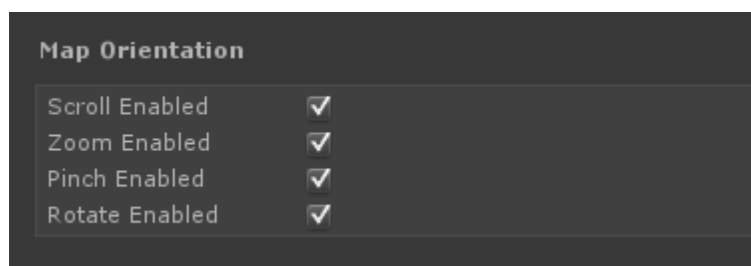
Hotkey	Description
Map Type	The visual type of the Map.
Show POI	Determines whether Points of Interest should be displayed on the map.
Show Buildings	Determines whether buildings should be displayed on the map.
Animated	Determines whether map actions should be animated (e.g. Go-To-Address-Fly-Ins)
Auto Location Update	Rotate the decal along the projection direction by 90 degrees (Clockwise).

6.2 Native Map Settings



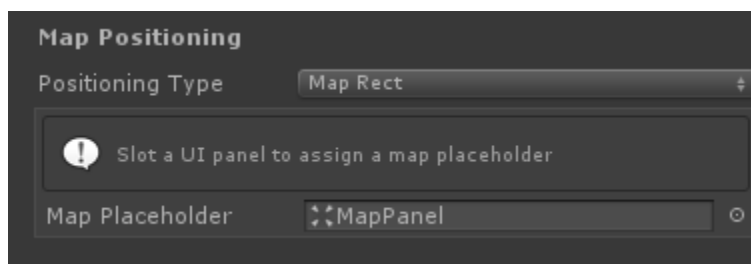
Hotkey	Description
Upadte User Position	Determines whether the user position should be frequently updated.
User Tracking Mode	Determines whether Points of Interest should be displayed on the map.

6.3 Map Orientation



Hotkey	Description
Scroll Enabled	Enables or disables map scrolling.
Zoom Enabled	Enables or disables map zooming.
Pinch Enabled	Enables or disables map pinching.
Rotate Enabled	Enables or disables map rotation.

6.4 Map Positioning



Hotkey	Description
Map Rect	Use an UI placeholder to position the map.
Absolute	Use absolute screen coordinates to position the map.
Margin	Use a border margin to position the map.
Fullscreen	Display the map in fullscreen mode.