LAR

LOCATIVE AUGMENTED REALITY TOOLKIT

VERSION 0.23

mariovskiii@gmail.com

1.UNITY3D VERSIONS

2020.1.0b

2019.3

2018.4

2.TARGET PLATFORM

Android

(IOS not tested, may work)

3. LAR OVERVIEW

This is an experimental Toolkit for using locative Media in your Unity3D project, resorting to GPS, Gyroscope and Compass as a registration method for Augmented Reality gaming experience.

Create a game or an Augmented Reality experience similar to Pokémon Go AR game or Google's Magical Park, in which a virtual world will emerge at a predetermined geographical location.

The user explores the physical world environment around him holding the smartphone and interact with 3d Media which is overlaid on the camera image.

4. KEY FEATURES

- Standalone Toolkit. For exclusively Locative AR projects it is not required to install other libraries such as VUFORIA or AR Foundation.

- Play mode inline simulator.

- Integration with VUFORIA Image Targets. LAR projects can easily combine Locative Targets with VUFORIA Visual Markers in the same scene.

- Option to Integrate with Native Toolkit (free asset) in order to provide better GPS accuracy (Double) instead of Unity3D's poor Input.Location method (Float).

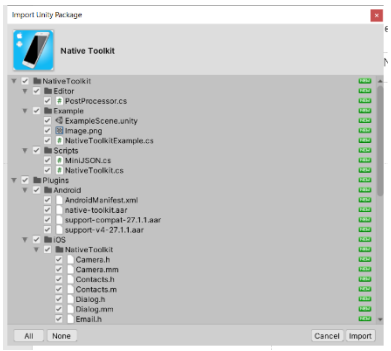
- Option to cast shadows from Unity3D GameObjects into the Image camera's floor.

5. KNOWN ISSUES

- Due to the nature of GPS and Gyroscope typical experience provided by locative AR toolkits is not comparable to visual marked based augmented reality, in terms of stability, lag and accuracy.

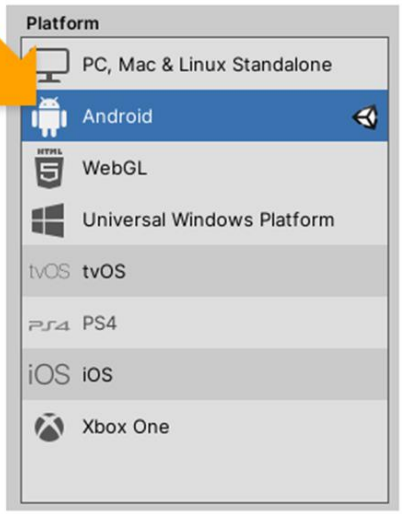
- For certain angles, directional light cast shadows wrongly into the Image camera's floor.

6. HELLO WORD! CREATING THE DEMO SCENE.

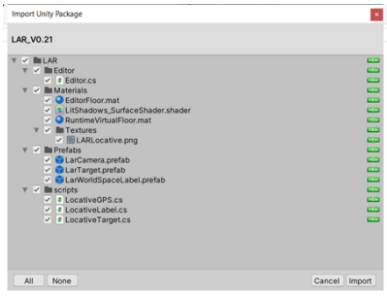


1. Create a new project with the name PRProj.

2. Install the Native Toolkit package from the AssetStore [Optional]. If Native Toolkit (free asset) is imported intor the project, LAR will use it in order to provide better GPS accuracy (Double) instead of Unity3D's poor Input.Location method (Float).

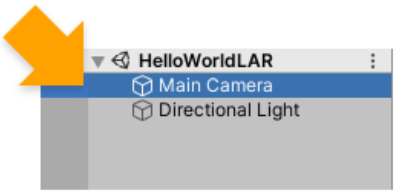


3. In the build settings, change the application target to Android.

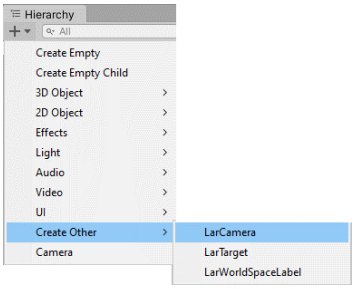


4. Install and import the LAR package

5. Create a new scene with the name HelloWorldLAR



6. Delete the Main Camera

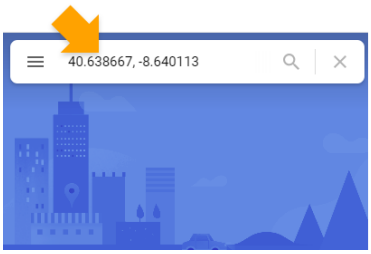


7. Add a new LARCamera. Click on add GameObject, then open the "Create Other" category and finally choose LARCamera option.

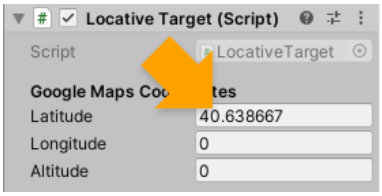
8. Create a new LARTarget. Open the Create Other category again choose "LARTarget" GameObject.

9. Open google Maps in the Browser.

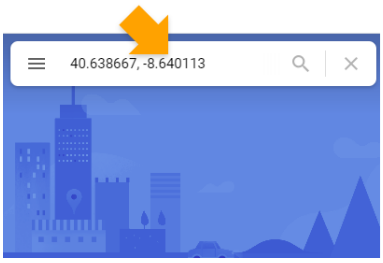
10. Choose a location near you and then click on a point on the map (1). An overlay with a link in the lower area, click on that link to open the geographical coordinates (2).



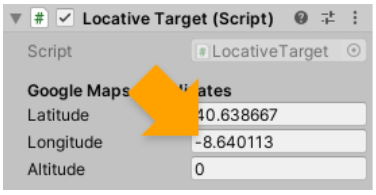
11. A tab appears on the right side of the browser window. Copy the first number regarding the latitude.



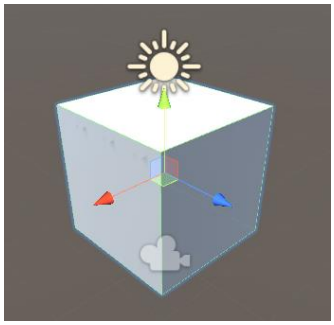
12. Back in the Unity3D editor, select the LARTarget GameObject and find the the input box for the Latitude variable in the inspector and paste the number. Make sure you don´t miss any number, the last digit may represent dozens of meters.



13. In the browser, copy the second number regarding the longitude.



14. In the Unity3D editor paste the number into the public variable Longitude from the LocativeTarget component. Again, make sure you copied and pasted all digits.



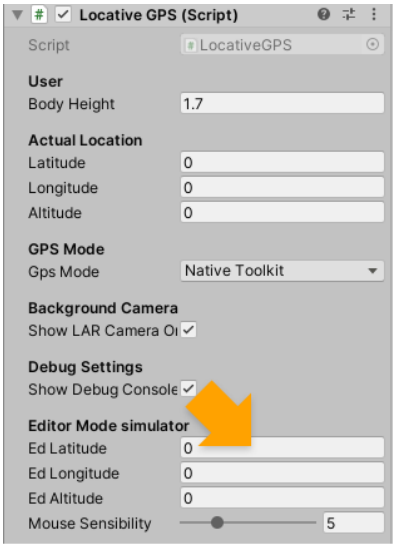
15. Create a cube-type primitive and make it as child of the LARTarget at location 0,0,0.

16. Assign a scale of 2m on all axes an d in the position enter the Y axis enter the value 1.5m.

17. Build the application to an Android smartphone (any API will do) and then, walk to the physical location to explore the scene.

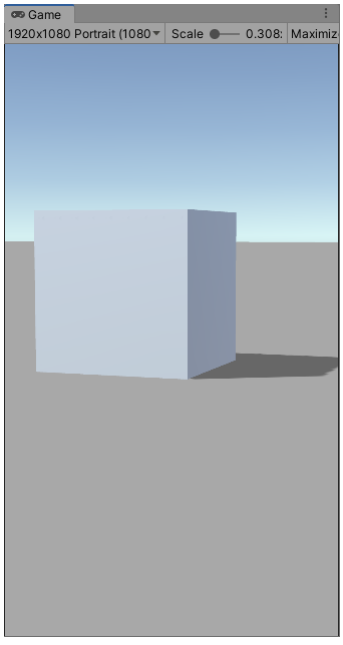
7. USING THE SIMULATOR.

Testing the scene during the development phase can be a very frustrating task for locative projects. In addition to the time it takes to produce the builds, the physical locations represented by our targets can be located geographically very far away or even be locations inaccessible.



1. To use the simulator, first go to google maps and copy the latitude and longitude from the starting point location. Choose a location close to the previously defined target, maybe 5 or 10 meters away.

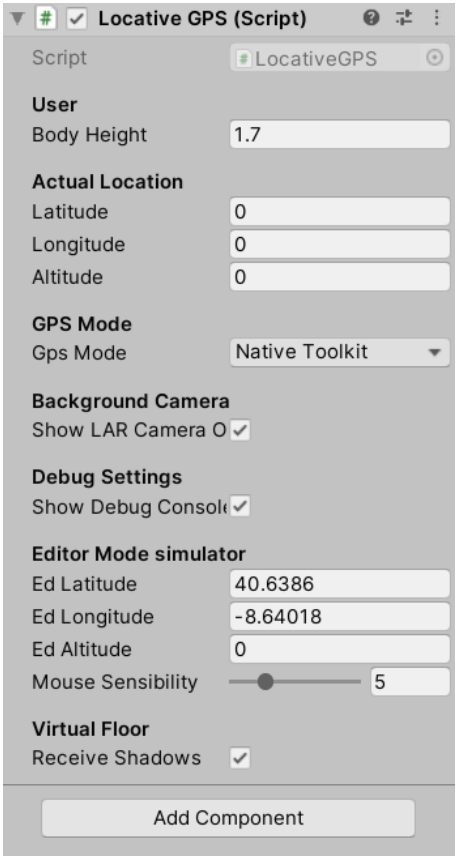
2. Select the GameObject LARCamera and in the Locative component, paste the coordinates in the EdLatitude and EdLongitude fields in the Play Mode Simulator section.



3. Now you can test and explore the scene without having to compile the project for Android.

Press Play in the editor. Use the mouse while holding the the right button to orient the camera. For moving around the world press W, S, A, D keys and hold the Shift key for running.

8. LARCamera OVERVIEW.



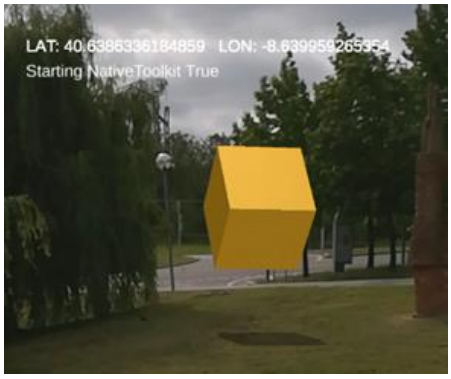
BodyHeight the height of the Camera. For a User 1.8m tall user the camera will be placed at his eyes level roughly 10cm below.

Latitude A double precision public variable that holds in real time the actual user latitude location.

Longitude A double precision public variable that holds in real time the actual user longitude location.

Altitude A double precision public variable that holds in real time the actual user altitude location. This value is always 0. GPS altitude data is very inaccurate, so LAR fixes the ground to 0.
If a LARTarget is defined in the Altitude field with a value of 10 then, in runtime, the user will see the object floating in air at 10 meters height.

GPSMode defines the interface to where LAR is fetching th GPS data. The valid constants values are Unity3D, NativeToolkit or OFF.
If one tries to use NativeToolkit without being installed, LAR in runtime will switch automatically to Unity3D mode.



ShowLARCameraOnBackground A boolean to show or hide the Camera in the background. Uncheck the checkbox in case a camera from another toolkit, such as VUFORIA, is needed.

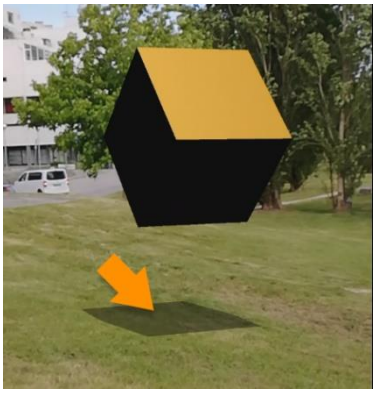
ShowDebugConsole A boolean value to enable a text field that is used by LAR to output data during runtime for debugging purposes, such as current GPS coordinates and other status messages.

EdLatitude A double precision public variable that holds in real time the actual user simulated latitude location in play mode.

EdLongitude A double precision public variable that holds in real time the actual user simulated longitude location in play mode.

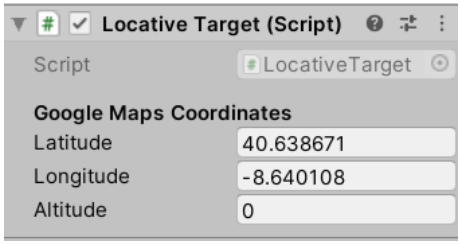
EdLatitude Unused. Fixed to zero in play mode.

MouseSensibility Mouse movement sensibility in play mode. Use the mouse while holding the the right button to orient the camera. For moving around the world press W, S, A, D keys and hold the Shift key for running



Receive Shadows When this option is on, shadows from Unity3D GameObjects are visible into the Image camera's floor.

9. LARTarget OVERVIEW.



Latitude A double precision public variable that holds latitude the location of the target.

Longitude A double precision public variable that holds the longitude location of the target.

Altitude A double precision public variable that holds the altitude location of the target.

10. ACCESSING LAR OBJECT.

To access all global variables of the LARCamera component a static object `LocativeGPS.Instance` is made available for convenience.

Example:

Reading the actual latitude of the player:

```
Double latitude = LocativeGPS.Instance.latitude;
```