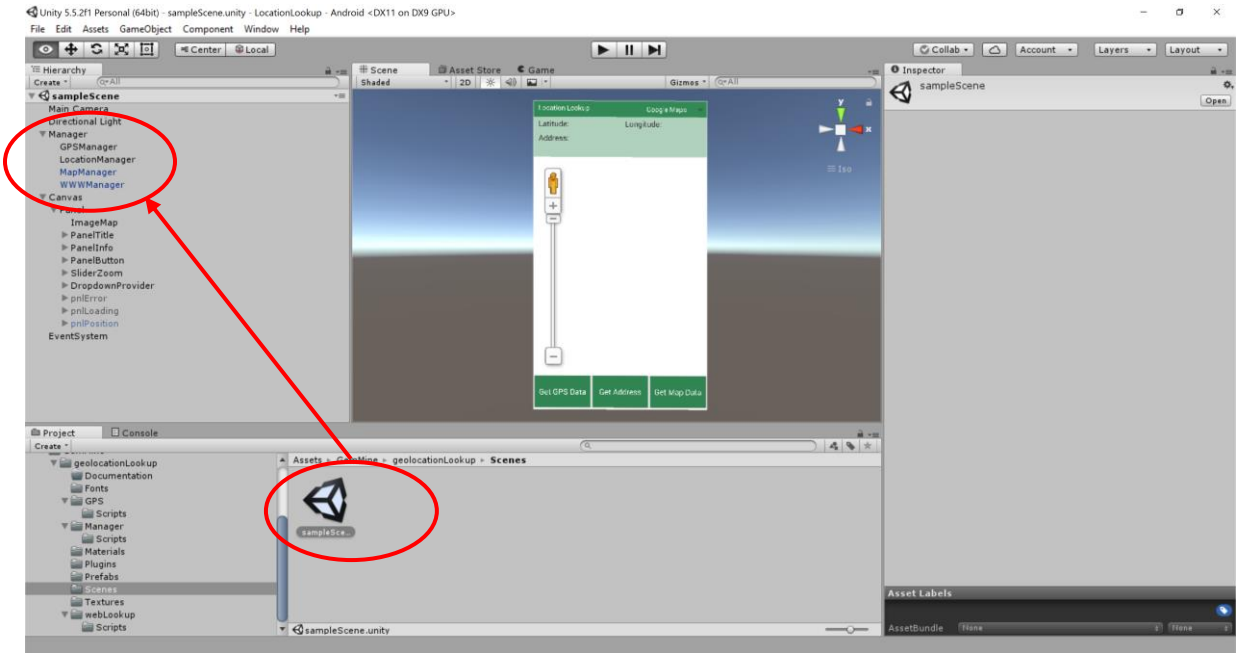
GEOLOCATION LOOKUP

Hello and welcome to the documentation of Project GEOLOCATION LOOKUP. This document should help you getting started with the project.

GEOLOCATION LOOKUP is a bundle of prefabs for locating and displaying your position in Unity. Open up the sample scene to see how it works.



If you start the scene, you will a preconfigured Manager object with four prefabs attached to it. These are the main prefabs in this package. If you want to learn about the interaction of these objects, please open up the Manager.cs script and follow the the workflow. Every step is commented and shows the necessary method calls to get results.

Nevertheless, let me introduce the basic functionality to you in the next sections.

GPS MANAGER

Purpose of the GPS MANAGER is to provide you with the actual longitude and latitude data from your GPS receiver. To start from ground up, follow these steps:

1. Create a new scene
2. Create an empty Game Object
3. Drag the GPS Manager prefab and child it to the empty Game Object
4. Create a new Script and attach it to the empty Game Object
5. Enter the code below
6. Drag the GPS Manager object to the script‘s inspector slot

To get access to the GPS Manager in your new script, you have to enter some lines of code – take the following class as a template:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GPSexample : MonoBehaviour {

    // slot for the package prefabs
    public GPSManager gpsManager;
    // If the GPS Manager does not get a result after
    // 10 seconds, it will cancel the process
    int timeoutInterval = 10;
    // latitude and longitude
    float latitude, longitude;

    void Start () {
        // Startup the GPS Manager
        // If the Manager finishes its work, it will call onGPSResult
        StartCoroutine (gpsManager.initGPS (onGPSResult, timeoutInterval));
    }

    public void onGPSResult(bool result) {
        if (result == false) {
            // something went wrong
            // handle the error here
        }
        else {
            // everything is fine
            // get the latitude and longitude from the gpsManager's properties
            latitude = gpsManager.Latitude;
            longitude = gpsManager.Longitude;
        }
    }
}
```

Besides the Latitude and Longitude property, you can query what error occured. Please use this property fort hat purpose:

```
gpsManager.Errorcode
```

LOCATION MANAGER

Purpose of the LOCATION MANAGER is to provide you with geocoding and reverse geocoding functions. In case you don't have a GPS or your GPS is not working properly, you may start up the LOCATION MANAGER with an address. It will do a quick lookup at google and send you the correct address and latitude/longitude information. To start from ground up, follow these steps:

1. Create a new scene
2. Create an empty Game Object
3. Drag the Location Manager prefab and child it to the empty Game Object
4. Drag the WWW Manager prefab and child it to the empty Game Object
5. Drag the WWW Manager prefab to the Location Manager's empty slot
6. Create a new Script and attach it to the empty Game Object
7. Enter the code below
8. Drag the Location Manager object to the script's inspector slot

To get access to the Location Manager in your new script, you have to enter some lines of code – take the following class as a template:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LocationExample : MonoBehaviour {

    public LocationManager locationManager;

    int timeoutInterval = 10;
    float latitude, longitude;
    string address;


    void Start () {
        // get the info by providing latitude and longitude
        locationManager.getLocationData (onLocationResult, timeoutInterval,
          latitude, longitude);
        // get the information by providing an address string
        locationManager.getPositionData (onLocationResult, timeoutInterval,
          address);
    }

    public void onLocationResult(bool result) {
        if (result == false) {
            // something went wrong, handle the error here
        }
        else {
            // parse the JSON object google delivered
            locationManager.parseLatLong(locationManager.wwwManager.Result);
            // get access to the properties by
            // locationManager.Latitude; locationManager.Longitude;
            // locationManager.Address;
        }
    }
}
```

MAP MANAGER

Purpose of the MAP MANAGER is to provide you with static maps. It will do a latitude/longite lookup at google, bin gor mapquest and send you the correct map back. To start from ground up, follow these steps:

1. Create a new scene
2. Create an empty Game Object
3. Drag the Map Manager prefab and child it to the empty Game Object
4. Drag the WWW Manager prefab and child it to the empty Game Object
5. Drag the WWW Manager prefab to the Map Manager's empty slot
6. Create an UI Image and drag it tot he Map Manager's empty slot
7. Create a new Script and attach it to the empty Game Object
8. Enter the code below
9. Drag the Map Manager object to the script's inspector slot

To get access to the Map Manager in your new script, you have to enter some lines of code – take the following class as a template:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MapExample : MonoBehaviour {

    public MapManager mapManager;
    public int timeoutInterval=10;
    float latitude, longitude;
    int zoom, mapProvider;

    // Use this for initialization
    void Start () {
        mapManager.getMapData (onMapResult, timeoutInterval, latitude, longitude,
        zoom, mapProvider);
    }

    public void onMapResult(bool result) {
        if (result == false) {
            // something went wrong
            // handle the error here
        }
        else {
            // everything is fine
        }
    }
}
```

As you may notice, this works exactly as the GPS- and Location Manager do.

You have to provide a latitude and a longitude value which you may get from the GPS Manager, an address lookup from the Location Manager or you may provide them manually.

The zoom factor may vary from provider to provider, but mostly it is in the range of 1..21.

MapProvider is an int value which you get from the Dropdown selection in the demo scene. Valid values here are 1 (google maps), 2 (mapQuest) and 3 (Bing Maps).

API KEYS

Please be aware, that most online Provider take money for the offer of their services. Some services are free, again some provider want you to register and get an API-Key for use of theri services and potential billing. Same applies here:

google maps offers a static map api for free where the use of an API Key is optional. MapQuest needs an API-Key which is free, same does Bing.

So, if you want to use different provider in this product, you have to register with them and get an API Key. The Map Manager offers three slots in the inspector where you can put your keys.