

Online maps

Version 2.3



Infinity Code, 2013-2016

<http://www.infinity-code.com/>

Contents

Overview	4
Introduction	4
Controls	4
Modes of the map	5
Quick creation of map using Wizard	5
Manual creation of map	5
Controls	6
Saving State of map	6
Work on mobile devices	7
Licenses and API keys	7
Description of fields	9
Online Maps Component	9
2D Controls	11
3D Controls	11
Tileset Control	12
Location Service	13
Tiles	14
Source of map tiles	14
Providers of tiles	14
Using «Provider - Custom»	15
Using MapBox	15
Styled Google Maps	16
Using offline	17
Preparation of tiles using GMapCatcher	17
Preparation of tiles for offline use manually	17
Caching the current state of map	17
Other methods for offline usage	17
Markers	18
Working with markers	18
Working with 3D markers	18
Animated markers	18
Using the Location Service (GPS)	19
Elevation data	19

Generation of buildings.....	19
Third-party assets and services	20
Using with NGUI, DF-GUI and iGUI.....	20
Using uGUI and SpriteRenderer	20
Working with Playmaker	20
Integration with Real World Terrain	21
Google API.....	21
Working with Open Street Map Overpass API	22
Other third-party API.....	22
Management of C# code	23
Changes to source code of Online Maps.....	23
Updating versions.....	23
Troubleshooting	24
Known issues	24
About KML.....	25
Your problem is not listed	25
Support.....	25
Links.....	26

Overview

Introduction

The component is designed for viewing online / offline maps in Unity.

Supports Unity v4.3+, Unity v5+.

Supported platforms: Standalone, Webplayer, WebGL, iOS, Android.

Important: Support for WebGL is experimental. Requires Unity v5.3+. If you find that something is not working properly, please write to support.

Online Maps contains:

- **Online Maps Component** – it is core. This component is fully controls the creation and operation of the map, and should always be used. To add this component, select «**Component / Infinity Code / Online maps / Online Maps**».
- **Controls** - those components are used to display map on various sources and user interaction. Be sure to use only one control. More information you can find in the section «**Controls**».
- **Plugins** - additional modules that serve to extend the functionality of the map. For example: GPS (Location Service), Google Maps API modules.

Online Maps has a powerful API, using which you will be able to solve any task, without modifying the core code.

Important: you can use only one instance of map in the scene.

Controls

Controls - are used to display map on various sources and user interaction.

Types of controls:

- **2D:** GUITexture, SpriteRenderer, UIImage, UIRawImage, NGUI, iGUI, DF-GUI.
- **3D:** Texture, Tileset.

All controls, except for «**Tileset**», working in «**drawing in texture**» mode. Tileset Control, works in «**Tileset**» mode. Learn more about modes, you can in «**Modes of the map**».

To use UIImage Control and UIRawImage Control must use Unity v4.6+.

To use NGUI control, iGUI control and DF-GUI control, you must have the relevant assets in the project. These assets you can buy in the Unity Asset Store.

Important: Do not use «**OnlineMapsControlBase**», «**OnlineMapsControlBase2D**» and «**OnlineMapsControlBase3D**». It is the base classes for other controls.

Important: You can use only one control.

If you want to display a map on the source for which there is no suitable control, please contact us. We will try to add the desired control in the shortest time.

Modes of the map

«Online Maps» supports two modes:

1. **Drawing in texture.** In this mode, the map and any additional elements are drawn in texture. It is a universal mode. You can display the map in any source, using the desired control.
2. **Tileset.** It is procedurally generated mesh. This mode is faster and requires less memory than drawing in texture. Supports 3D markers and elevation data. **Recommended for use on mobile devices and WebGL.**

Quick creation of map using Wizard

Select «Hierarchy / Create / Maps» or «GameObject / Create Other / Map».

Note: In Unity 5, select «Hierarchy / Create / Create Other / Map».

Select the type of map, specify where you want to use the map, additional settings and click «Create».

Manual creation of map

Select the menu item «Component / Infinity Code / Online maps / Online Maps», to add a component to a selected GameObject.

In section «**Create texture**» select size, type of texture (SpriteRenderer and UIImage – Sprite, other – Texture) and click «**Create**». Will create a new texture in the root of the project named «**OnlineMap**». If a file with the same name already exists, it will be replaced.

Select a texture provider and type that you want to use.

Add created texture (OnlineMap) to the source display. For example, if you use GUITexture, then in the field «**Texture**», if you use the material, then «**Renderer / Material / Main Texture**».

Important: If you use the «**Tileset**», then create texture unnecessary.

To become an interactive map, click on the menu item «**Component / Infinity Code / Online Maps / Controls**».

In section «**Advanced**» select «**Redraw on Play**».

Run the scene.

Move the map area to the right place.

Add markers (for details, see «**Markers**»).

To save the current state, press «**Save state**».

Stop the scene.

IMPORTANT

«**Online maps**» uses WWW class, for which there are security limitations. For this reason, for Webplayer and WebGL, all request of tiles and services works through the site **infinity-code.com**.

Read more about the limitations of security WebPlayer, you can read here:

<http://docs.unity3d.com/Documentation/Manual/SecuritySandbox.html>

For WebGL server should return a response with a header that is not implemented in most of providers:

```
header("Access-Control-Allow-Origin: *");
```

Currently, there are no restrictions on the number of queries. But with a strong increase in the load on our servers, we can add such restrictions.

All requests to the server are cached.

If you want to get a server-side scripts, and use them on your server, please contact support. In this case, you will not depend on the work of our server.

Controls

Map control is similar to that in other online services and applications.

Click and drag the mouse to change the position.

Use mouse wheel to zoom.

Double click zoom in map under the cursor.

Press «**M**», to add a new marker.

Press «**N**», to add a new 3D marker.

Hold down «**Left Control**», to enable movement of markers.

If you are using 3D control, with enabled «**Allow Camera Control**», you hold down the right mouse button to turn on the rotation of the camera around the map.

Saving State of map

You can save the current state of the map in play mode.

To do it, click on «**Save state**», select the items that you want to save, and click again «**Save state**».

After you stop the scene, the map will restore saved state.

Work on mobile devices

Online Maps works on iOS and Android.

Gestures:

- **Drag** - moving map.
- **Pinch** - change zoom.
- **Double tap** – zoom in, at the tap point.
- **Spread** - the rotation of the camera around the map. Requires 3D control with «**Allow Camera Control**».

Important: in current version smooth pinch-to-zoom is only possible when using «**Tileset**». For use smooth pinch-to-zoom, enable «**Smooth zoom**».

For optimal performance on mobile devices, we recommend that you use «**Tileset**» mode.

If you want to draw a map in the texture, we recommend using the texture size not more than 512x512.

If you see on iOS big red question marks instead of the texture, enable the software JPEG decoder (**Online Maps Inspector / Troubleshooting / Software JPEG Decoder**).

Licenses and API keys

Online Maps provides you with the technical ability to use the map in your application, but does not give you any permissions and privilege. When working with Online Maps, you must comply with the terms of the license agreement of chosen provider.

License Agreement:

Google Maps: <https://developers.google.com/maps/terms?csw=1>

ArcGIS: <http://resources.arcgis.com/en/help/main/10.1/index.html#//00qn00000030000000>

Nokia (here.com): <https://legal.here.com/en/terms/serviceterms/us>

MapQuest: <http://info.mapquest.com/terms-of-use/>

Virtual Earth (Bing Maps): <http://www.microsoft.com/maps/product/terms.html>

Open Street Map: <http://opendatacommons.org/licenses/odbl/1.0/>

Sputnik: <http://corp.sputnik.ru/maps>

AMap: <http://lbs.amap.com/home/terms/#English>

To use some features Online Maps, you need API Keys of some providers.

Google API Key is required to use: OnlineMapsFindAutocomplete, OnlineMapsFindPlaces, OnlineMapsFindPlaceDetails, OnlineMapsGetElevation.

You can create your Google API key in Google API Console:

<https://console.developers.google.com/>

Bing Maps API Key is required to use Elevations and Bing Maps Location API.

You can create your Bing Maps API Key here:

<http://msdn.microsoft.com/en-us/library/ff428642.aspx>

Description of fields

Online Maps Component

1 Online Maps (Script) Update Available Help

2 Source: Online

3 Provider: Google

4 Type: Satellite

5 Labels: ☒

6 Language: en
Use two-letter code such as: en

7 Latitude: 39.81504

8 Longitude: 8.727479

9 Zoom: 7

10 Target: Texture

11 Texture: [Map Preview]

12 Target: Tileset

13 Width (pixels): 1024

14 Height (pixels): 1024

15 Size (in scene): X 1024 Y 1024

16 Cache tiles to Resources

17 Save state

18 2D Markers

19 Add marker

20 Marker 1

21 Latitude: 39.81504

22 Longitude: 8.727479

23 Zooms (3-20): [Slider]

24 Rotation (0-1): 0

25 Label:

26 Align: Bottom

27 Texture: None (Texture2D)

28 Remove

29 Create texture

30 Width: 512

31 Height: 512

32 Create

33 Advanced

34 Redraw on play: ☒

35 Smart texture: ☒

36 Traffic: ☐

37 Empty color: [Color Picker]

38 Default tile: None (Texture2D)

39 Skin: DefaultSkin

40 Default marker: [Marker Preview]

41 Markers align: Bottom

42 Show marker tooltip: On Hover

43 Troubleshooting

44 Use this props only if you have a problem!!!

45 Software JPEG Decoder: ☐

1. Indicates that updates are available on the selected channel. Click to go to the built-in update.

2. **Source** – source of tiles (Online, Resources, Resources and Online).

3. **Provider** – provider of tiles.

4. **Type** – type of map.

5. **Label** – display the names of objects on the map.

6. **Language** – language names on the map.

7. **Latitude / Longitude** – coordinates the center point of map.

8. **Zoom** – the current zoom level.

9. **Target** – where will be drawn map (Texture or Tileset).

10. **Texture** – texture, which will draw maps. This field is available only in the «drawing in texture».

11. **Width / Height (pixels)** – size of tileset. This field works like texture size. More value - better map, but at a slower speed. Must be divisible by 512. Available only in «Tileset».

12. **Size (in scene)** – size of tileset in the scene. Available only in «Tileset».

13. **Cache tiles to Resources** – caches the current map tiles to «Resources». This button is available only in play mode.

14. **Save state** – saves the current state of map. This button is available only in play mode. After that you can choose what you want to save.

15. **2D Markers** – this section contains all the 2D markers, which are added to the map.

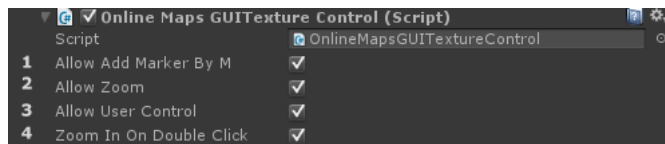
16. **Add marker** – creates a new marker.

17. Marker number.

18. **Latitude / Longitude** – coordinates of the marker.

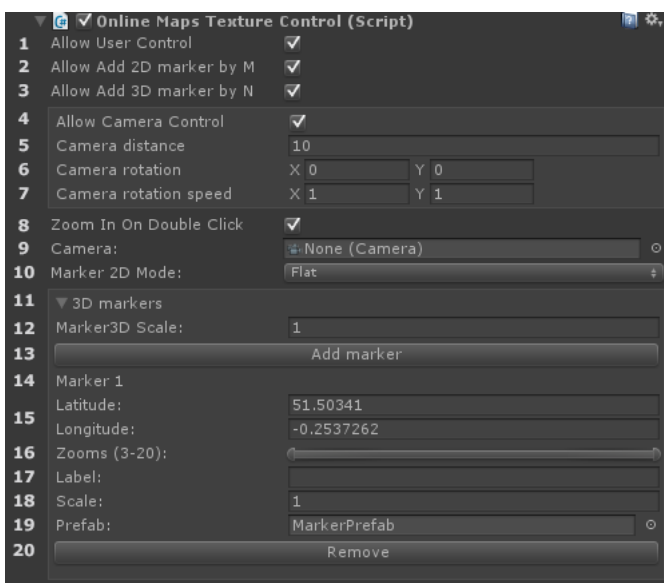
- 19. **Zooms** - zoom of the map, at which the marker is displayed.
- 20. **Rotation** – angle of rotation of the marker (0-1) in a clockwise direction.
- 21. **Label** – name (tooltip) of the marker.
- 22. **Align** – align of the marker.
- 23. **Texture** – texture of the marker.
- 24. Removes marker.
- 25. **Create texture** – this section allows you to create a new texture, which will draw maps. Available only in «**drawing in texture**».
- 26. **Width / Height** – size of the new texture.
- 27. Create texture.
- 28. **Advanced** – section, which contains additional settings.
- 29. **Redraw on play** – specifies whether the texture is redrawn immediately at startup scene, or only after changing the position or zoom. Available only in «**drawing in texture**».
- 30. **Traffic** – specifies whether to display the layer with information about traffic congestion.
- 31. **Smart texture** - creates a reduced virtual texture used when the user interacts with the map.
- 32. **Empty color** – color used to draw the tile until the texture tile is not loaded. If you specify a «**Default tile**», then this color is not used.
- 33. **Default tile** – texture used for the tile, until the current tile texture is not loaded.
- 34. **Skin** – style used for the tooltip marker.
- 35. **Default marker** – marker icon used by default.
- 36. **Markers align** – alignment marker icons used by default.
- 37. **Show marker tooltip** – when should show a tooltip (hover, pressing, always or never).
- 38. **Troubleshooting** – in this section contain settings that will help solve the problem. Use this section only if you have problems.
- 39. **Software JPEG Decoder** – to load JPEG will be used software decoder. Software decoder is much slower than the hardware.

2D Controls



1. **Allow Add Marker By M** – point to the desired location on the map and click on «M», to create a new marker.
2. **Allow Zoom** – specifies whether the user can change the zoom map.
3. **Allow User Control** – specifies whether the user can change the coordinates of map.
4. **Zoom In On Double Click** – specifies whether the map zoom in by double-clicking (double-tap on mobile devices).

3D Controls



1. **Allow User Control** – specifies whether the user can change the position of map.
2. **Allow Add 2D Marker By M** – point to the desired location on the map and click on «M», to create a new marker.
3. **Allow Add 3D Marker By N** – point to the desired location on the map and click on «N», to create a new 3D marker.
4. **Allow Camera Control** – specifies whether to rotate the camera around the map. To rotate the map, use the right mouse button (gesture Spread on mobile devices).
5. **Camera distance** – the distance from the camera to the center point of the map. Available only when «**Allow Camera Control**».

6. **Camera rotation** – camera rotation angles, relative to the center point of the map. Available only when «**Allow Camera Control**».

7. **Camera rotation speed** – the speed of rotation of the camera. Available only when «**Allow Camera Control**».

8. **Zoom In On Double Click** – specifies whether the map zoom in by double-clicking (double-tap on mobile devices).

9. **Camera** – camera which will interact with the Control. If the camera is not specified, will be used Main Camera.

10. **Marker 2D Mode** - display mode of markers (flat or Billboard).

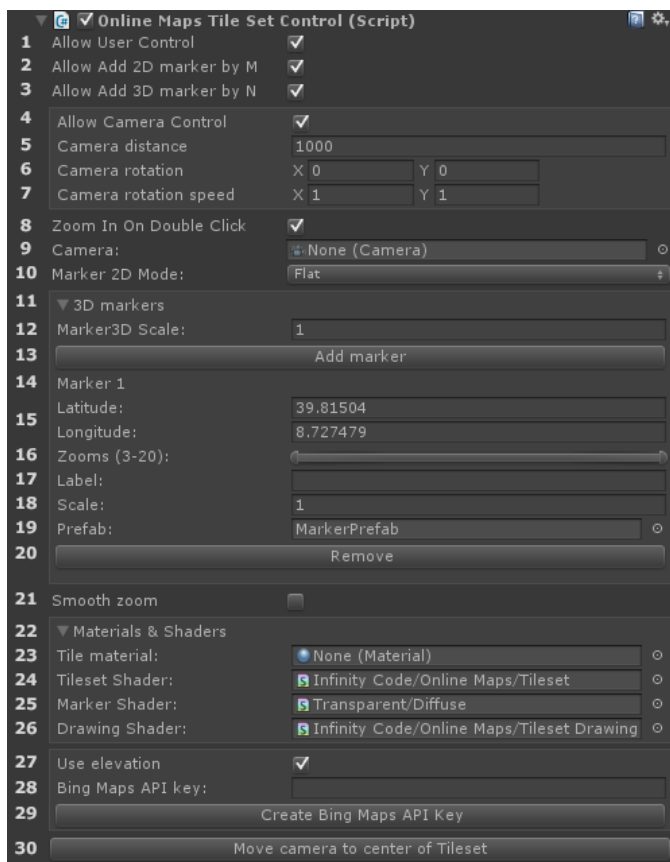
11. **3D Markers** – this section contains all the 3D markers.

12. **Marker3D Scale** – the default scale that will be used for new 3D markers.

13. **Add maker** – adds a new marker.

14. Marker number.
15. **Latitude / Longitude** – coordinates of the marker.
16. **Zooms** – zoom of the map, at which the marker is displayed.
17. **Label** – name (tooltip) of the marker.
18. **Scale** – scale of the marker.
19. **Prefab** – GameObject, which will be used for the marker. If prefab is not specified, will use the Box.
20. Removes marker.

Tileset Control



1-20. See 3D Control.

21. **Smooth zoom** – use a smooth touch zoom on mobile devices.

22. **Materials & Shaders** – in this section, you can set materials and shaders for Tileset.

23. **Tile material** – material that will be used for tiles.

24. **Tileset Shader** – shader that will be used to tile, if «**Tile material**» is not specified.

25. **Marker Shader** – shader that is used for 2D markers.

26. **Drawing Shader** – shader that will be used for Drawing Elements.

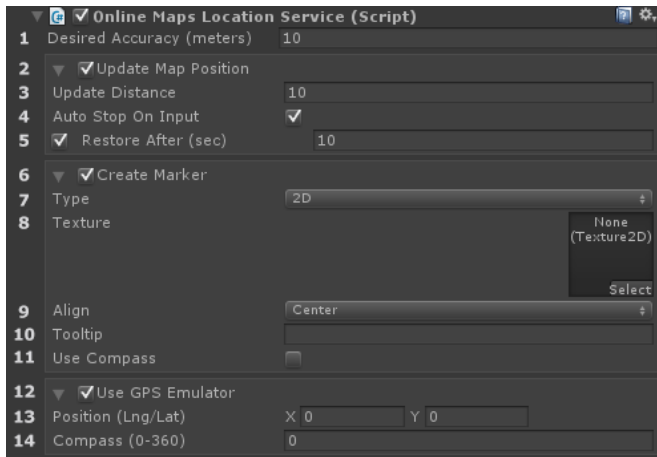
27. **Use Elevation** – specifies whether to use elevation maps.

28. **Bing Maps API Key** – key for Bing Maps API. Available only when «**Use Elevation**».

29. **Create Bing Maps API Key** – opens the page in a browser, where you can create your own API Key.

30. **Move camera to center of Tileset** – directs the camera at a central point on the map and a camera perpendicular to the map.

Location Service



1. Desired Accuracy (meters) - desired accuracy in meters. When using a large value (e.g., 500) typically do not require the use of GPS and saves battery power. Small values of 5-10 may be used to provide maximum accuracy.

2. Update Map Position - update the position map based on GPS data.

3. Update Distance - the minimum distance (in meters) at which the position will be updated. Higher values (eg, 500) means less overhead.

4. Auto Stop On Input – stop update position of the map after the user changes the position or zoom position.

5. Restore After (sec) – restore the auto update map position after the specified number of seconds of inactivity.

6. Create Marker – create a marker at the current GPS position.

7. Type – type of marker (2D or 3D).

8a. Texture – texture of 2D marker.

8b. Prefab – GameObject, which will be used for 3D marker.

9. Align – align of 2D marker.

10. Tooltip – tooltip of marker.

11. Use Compass – rotate marker on the value of the compass.

12. Use GPS Emulator – emulates the value of GPS and compass. Works only in Unity Editor.

13. Position – emulated GPS coordinates.

14. Compass – emulated value compass.

Tiles

Online Maps, like most mapping applications to create maps use tile system.

Online Maps supports only bitmap tiles, size - 256x256 pixels.

What is a tile system and how it works, you can read here:

<https://msdn.microsoft.com/en-us/library/bb259689.aspx>

<http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/>

Source of map tiles

Online maps can receive the tiles:

- **Online** - from provider site.
- **Resources** - from the folder «**Resources**».
- **Resources and Online** - if possible tiles will be loaded from the folder «**Resources**». If the tile is missing, it will be downloaded from provider site.

If you want to load the tiles from the folder «**Resources**», then place tile images in «**Resources / OnlineMapsTiles / {ZOOM} / {TileX} / {TileY} .png**».

Providers of tiles

Online Maps can work with any provider of tiles having a projection like in Google Maps.

It is most of the existing maps.

Supported: ArcGIS, Google Maps, Nokia Maps (here.com), MapQuest, Virtual Earth (Bing Maps), Open Street Map, Sputnik, AMap.

Not supported (have another projection): Baidu Maps, Yandex Maps.

Partially supported (need to manually use «Provider – Custom»): MapBox.

If you want to use a provider who is not listed, please contact us. We tested the desired provider.

Using «Provider - Custom»

To use «**Provider - Custom**» in the field «**URL**» specify a URL pattern for download tiles.

URL pattern supported tokens:

{zoom} – Zoom level.

{x} – Tile X.

{y} – Tile Y.

{quad} – Tile Quad Key. Uses in Virtual Earth (Bing Maps).

General instructions:

1. Open in a browser map on the site of provider, tiles from which you want to use.
2. Open a browser developer tools. Hotkey in most browsers - F12.
3. Find the address of tile.
4. The address replace the numbers that are responsible for the tile zoom, x, y, to tokens.
5. Insert the URL pattern into the field «URL».

Step by step instructions for using «Provider - Custom», you can read the sections «**Using MapBox**» and «**Styled Google Maps**».

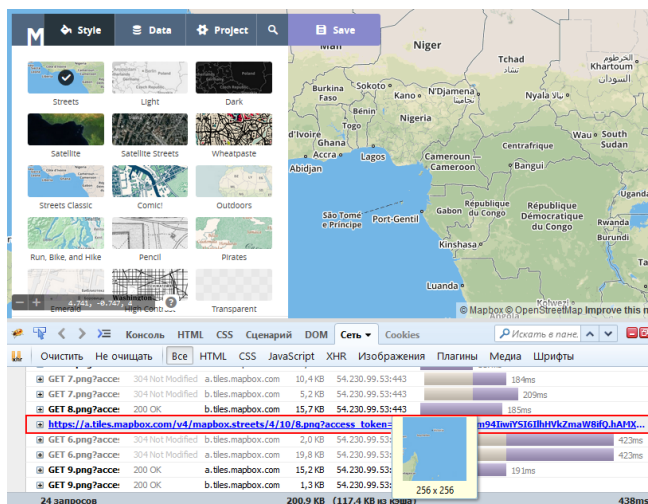
Using MapBox

Open MapBox map in your browser. For example, we will use the map mapbox.streets:

<https://www.mapbox.com/editor/?layers=mapbox.streets#style>

Open a browser developer tools. Hotkey in most browsers - F12.

Find an address of tile.



Copy the address.

https://a.tiles.mapbox.com/v4/mapbox.streets/4/10/8.png?access_token=XXX

In the address, replace the numbers that are responsible for tile zoom, x, y to tokens.

You should get URL pattern:

https://a.tiles.mapbox.com/v4/mapbox.streets/{zoom}/{x}/{y}.png?access_token=XXX

In Online Maps, select «**Provider - Custom**», and insert URL pattern.

Styled Google Maps

To use Styled Google Maps:

1. Configure the desired style map using any online service that allows you to customize the style of the map. For Example:
<http://gmaps-samples-v3.googlecode.com/svn/trunk/styledmaps/wizard/index.html>
or
<https://snazzymaps.com/>
2. Open the browser developer tools. Hotkey in most browsers - F12.
3. Find an address request to download any tile. It will look like:
<http://mt0.googleapis.com/vt?pb=!1m4!1m3...>
4. Open this URL and make sure that the image has a size of 256x256.
5. Copy this address.
6. In Online Maps Inspector select «Provider - Custom», and paste the address in «URL».
7. In the «URL» replace «!1iX!2iX!3iX», to «! 1i{zoom}!2i{x}!3i{y}».
It should look something like this:
<http://mt0.googleapis.com/vt?pb=!1m4!1m3!1i{zoom}!2i{x}!3i{y}!2m3!...>

That's all. You are using Styled Google Maps.

Using offline

Preparation of tiles using GMapCatcher

To prepare the tiles convenient to use GMapCatcher.

With GMapCatcher, download the desired area of the map.

In Online Maps click «**Import From GMapCatcher**».

All downloaded tiles are placed in the folder «**Resources / OnlineMapsTiles**», in the format required for use.

Preparation of tiles for offline use manually

Place your tiles in the format «**Resources / OnlineMapsTiles / {ZOOM} / {TileX} / {TileY}.png**».

Caching the current state of map

Click «**Cache tiles to Resources**», to cache the current state of map in the folder Resources.

Select «**Source - Resources and Online**».

Cached tiles will be loaded from the Resources folder, to speed up launch scene.

Other methods for offline usage

Using Online Maps API, you can intercept a request to download the tile, and load it from any source, for example: AssetBundle or file system.

Example of use: CustomDownloadTileExample.cs

Markers

Working with markers

In section «**Markers**» you can see all the markers.

If you specify «**Label**», then when you hover over the icon in the play mode, will see tooltip.

To add markers in play mode:

Place your cursor in the place where you want to add a marker and press the «**M**». In that place there will be created a new marker, with the default settings.

You can drag the marker on the map, hold down the «**Left Control**».

After making changes in the play mode, press «**Save state**».

Important: Marker texture should be formatted **ARGB32**, and enabled «**Read / Write Enabled**». These parameters can be found in the Import Settings textures.

Working with 3D markers

Important: 3D markers are only supported on Texture Control or Tileset Control.

Working with 3D markers similar to working with 2D markers. You can create a 3D marker on map in the specified location by pressing «**N**».

All created 3D markers can be found in the component «**OnlineMapsTextureControl**» or «**OnlineMapsTilesetControl**».

Animated markers

Unfortunately, Online Maps does not have a built-in ability to work with animated markers.

But you have a few ways to create animated markers:

1. Use the 3D markers. For example: make a flat 3D marker, and change marker texture using script.
2. Create your own marker system, such uGUI. It's easy. Examples of creating own system of markers NGUI and uGUI in «Examples (API Usage)». If you need assistance in creating your own marker system, please contact us.

Using the Location Service (GPS)

For mobile devices available determination of the location by GPS. To enable the determination of the location, add a script «**OnlineMapsLocationService.cs**» or select «**Component / Infinity Code / Online Maps / Plugins / Location Service**».

To create a marker at the current position switch «**Create Marker**», select the type of marker and specify its parameters. It will automatically update the position when changing locations. If you want marker indicates the direction then enable «**Use Compass**».

To enable GPS Emulator enable «**Use GPS Emulator**». Specify the location value and compass value.

Important: GPS Emulator only works in Unity Editor. When publishing to any platform it will not be used.

Elevation data

To gets elevation data uses Bing Maps Elevation API.

To make a map with the real-world elevation data use Tileset Control.

Enable «**Use Elevation**» and enter your Bing Maps API key. If you do not have the key, press the «**Create Bing Maps API key**».

Generation of buildings

Generation of buildings is only available using Tileset Control.

To enable the generation of buildings add «**OnlineMapsBuildings.cs**» to map GameObject, or select «**Component / Infinity Code / Online Maps / Plugins / Buildings**».

Buildings are created dynamically based on Open Street Maps data.

In «**Online Maps Buildings Inspector**» select the required materials for walls and roofs.

Third-party assets and services

Using with NGUI, DF-GUI and iGUI

Add the desired control on the GameObject with the map. Enable support by pressing «**Enable NGUI (DF-GUI or iGUI)**».

For NGUI: To become an interactive map, add Collider to GameObject (**NGUI \ Attach \ Collider**).

For NGUI: If you use a mode Tileset, it must be located outside the object at Hierarchy UIRoot.

Using uGUI and SpriteRenderer

To work with the map through uGUI Image and Sprite Renderer, you select the texture map in «Project» window. In the Import Settings texture you specify «**Sprite mode - Single**». After that, add this texture to «**Image / Source Image**» or «**Sprite Renderer / Sprite**».

If «**Sprite Mode - Single**», automatically reverts to «**Sprite Mode - None**», follow these steps:

- Texture Type – Sprite;
- Apply;
- Texture Type – Advanced;
- Read / Write Enabled – On;
- Apply.

For SpriteRenderer: To become an interactive map, add BoxCollider on GameObject.

Working with Playmaker

Before importing «**Online Maps Playmaker Integration Kit**» make sure Playmaker is present in your project. Supported Playmaker v1.7.7 and higher.

Select «**Component / Infinity Code / Online Maps / Packages / Playmaker Integration Kit**», to import actions, to work with Online Maps.

New actions are grouped into sections, whose name starts with «**Online Maps**».

These actions contained about 95% general features of Online Maps. This is sufficient for most applications.

Unfortunately, the implementation of some specific tasks cannot create action, so you need to use the Online Maps API.

Learn how you can use Playmaker official documentation:

<https://hutonggames.fogbugz.com/default.asp?W1>

Important: Each time you update Online Maps reimport «**Playmaker Integration Kit**» again.

Integration with Real World Terrain

You can see on the map the position, based on the result of the Real World Terrain.

Add «**Component / Infinity Code / Online Maps / Plugins / Real World Terrain Connector**», to object generated by Real World Terrain (near Real World Terrain Container), and select connector mode.

Google API

Online Maps is able to work with some Google API services:

Google Direction API - is a service that calculates directions between locations. You can search for directions for several modes of transportation, include transit, driving, walking or cycling. Directions may specify origins, destinations and waypoints either as text strings (e.g. "Chicago, IL" or "Darwin, NT, Australia") or as latitude/longitude coordinates. The Directions API can return multi-part directions using a series of waypoints.

Google Elevation API – provides elevation data for all locations on the surface of the earth, including depth locations on the ocean floor.

Google Geocode API - converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map.

Google Places API - allows you to query for place information on a variety of categories, such as: establishments, prominent points of interest, geographic locations, and more. You can search for places either by proximity or a text string. A Place Search returns a list of places along with summary information about each place.

Google Place Details API – provides detailed information about the object by place id, or reference.

Google Places Autocomplete API - returns place predictions. The request specifies a textual search string and optional geographic bounds. The service can be used to provide autocomplete functionality for text-based geographic searches, by returning places such as businesses, addresses and points of interest as a user types.

If you need to support anything else that services Google, please contact us. We will try to add their support in Online Maps.

Important: To use Google Places API and Google Place Autocomplete API requires Google API Key. You can get them in Google API Console:

<https://console.developers.google.com/>

Working with Open Street Map Overpass API

Using Open Street Map Overpass API, you can get information from Open Street Maps, for example: buildings, roads, rivers.

Full list of objects:

http://wiki.openstreetmap.org/wiki/Map_Features

Documentation of Overpass API:

http://wiki.openstreetmap.org/wiki/Overpass_API/Language_Guide

Use «**OnlineMapsOSMAPIQuery.Find** (Overpass QL request)», to query the Overpass API.

You can test your queries using:

<http://overpass-turbo.eu/>

Other third-party API

Bing Maps Location API - converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map. **Required Bing Maps API Key.**

Open Street Map Nominatim - converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map.

Open Route Service - is a service that calculates directions between locations. You can search for directions for several modes. Directions may specify origins, destinations and waypoints either as text strings (e.g. "Chicago, IL" or "Darwin, NT, Australia") or as latitude/longitude coordinates. The Directions API can return multi-part directions using a series of waypoints.

Management of C# code

Detailed information about all the available classes, methods, and variables can be found in API Reference.

Examples of using Online Maps API you can find in the folder «**Examples (API usage)**».

Atlas of Examples is located in the «**Documentation**».

Important: unpack API Reference and Atlas of Examples outside of your project.

If you have an example that you would like us to have published in «**Examples (API usage)**», please send it to us. We will publish it with authorship attribution and conservation copyrights.

Changes to source code of Online Maps

We do not recommend you to make changes to source code of Online Maps, because they will be lost when you update version.

We tried to make API as open for use and extension. Most problems can be solved using API or extend the class of the corresponding Control.

If for your application you must make changes to source code of Online Maps, please send us the modified code. We will try to apply your code into the mainline project, or give you the best solution.

Updating versions

In the Unity Asset Store, we send only the stable versions. Renewal period is several months.

Online Maps has built-in update system, using which you can update Online Maps to the latest version and get early access to all versions and updates.

Click «**Component / Infinity Code / Online Maps / Check Updates**», to open the window checking for updates. Enter your Invoice Number, select a channel of updates and click «**Check New Versions**».

You can find out your Invoice Number in order confirmation, or visit the account transaction page in Unity Asset Store:

<https://www.assetstore.unity3d.com/en/#!/account/transactions>

If from the current version was released more than 10 updates in the selected channel then will only show the last 10 updates.

If updates are available, you can read the list of changes and download the update.

If you have any problems with installing the update, then:

1. Open an empty scene.
2. Delete the folder «**Infinity Code / Online Maps**».
3. Import the new version of Online Maps to the project.

If you want to rollback to the previous version of Online Maps, select the channel «**Stable Previous**».

Using this channel, you can get 10 previous stable versions.

Online Maps automatically checks for updates every 24 hours. If a new version is available, you will see a red button in Online Maps Inspector. When clicked, will open check for updates window.

Automatic check for updates does not require Invoice Number, and works on the selected channel updates. If you do not select a channel update, it will be checked only stable versions.

Troubleshooting

Known issues

Drawing API - polygon has no fill in «**Tileset**». This problem is related to the very difficult polygon triangulation. Most likely, this problem will never be solved. If you really need to be able to draw a filled polygon, then use «**drawing in texture**».

Publishing for Windows Phone, Windows Store - we cannot add the ability to publish to these platforms, because of incompatibility «**System.XML**». Online Maps uses XML to work with Google API. If you know a simple way to read XML for Windows Phone and the Windows Store, please contact us.

Do not work Google Relief and Google Terrain maps on iOS - Sometimes Unity + XCode using broken JPEG decoder. Direct solution to this problem is not present.

There are two workarounds to solve this problem:

1. Use a styled Google map. How to make a styled map, see «**Styled Google Maps**». The trick is that the styled map is returned PNG, even if the style is empty.
2. Enable «**Troubleshooting / Software JPEG Decoder**». Use this option only if you could not use a styled map because software decoder works slowly.

If «**Sprite Mode - Single**», automatically reverts to «**Sprite Mode - None**», follow these steps:

- Texture Type – Sprite;
- Apply;
- Texture Type – Advanced;
- Read / Write Enabled – On;
- Apply.

Tileset not work correctly with Skybox, or has display problems:

Online Maps Tileset Control Inspector / Materials & Shaders / Tileset Shader - Tileset Cutout Shader.

Error «CS0576: Namespace `global :: 'contains a definition with same name as alias `XXXX'»:

In your project has class with the name specified in the description of the error.

It is bad practice to have in the global namespace a class named as Unity classes or standard classes. You will have a problem with all assets, uses this class.

If you created this class, rename it.

If this is part of another asset, then contact the developer and ask them to fix the problem.

About KML

Online Maps does not support KML, and most likely will never support it.

KML is a very comprehensive format that has many features that are not directly related to the maps.

Add partial support for KML we believe to be wrong.

If you need some data from KML, you can write a script to read it and use it in the map.

Your problem is not listed

Try to update to the latest beta version, using built-in update. Perhaps we have fixed this problem.

If problem persists, please write us about your problem (support@infinity-code.com). We will try to fix all errors and release an update as soon as possible.

Support

If you have something does not work, you find a bug, or you have a suggestion, please contact us.

In email specify your version of Unity, OS, and the current version of Online Maps.

We strive to answer all emails in the customer support within 24 hours.

Links

Product page: <http://infinity-code.com/en/products/online-maps>

API Reference: <http://infinity-code.com/docs/api/online-maps/>

Atlas of Examples: <http://infinity-code.com/atlas/online-maps/>

Support: support@infinity-code.com

Forum: <http://forum.infinity-code.com>

Videos: <https://vimeo.com/channels/onlinemaps>