

Online Maps

Version 3.9



Infinity Code

<https://infinity-code.com/>

Contents

Overview	5
Introduction	5
Several instances of the map in the scene	5
Updating project from Online Maps v1.x and 2.x	6
Controls	6
Modes of the map	6
Quick creation of map using Wizard	7
Manual creation of map	7
Limitations of WebGL	8
Interaction with the map	8
Saving state of map	8
Work on mobile devices	9
Using uGUI Image and SpriteRenderer	9
Display Tileset on UI	10
Scriptable Render Pipeline (URP, HDRP)	10
Licenses and API keys	11
Description of fields	12
Online Maps Component	12
2D Controls	13
3D Controls	14
Tileset Control	14
Buildings	15
Cache	16
Camera orbit	17
Elevation manager	17
Limits	18
Location Service	18
Marker manager	19
Marker 3D Manager	20
Tiles	21
Source of map tiles	21
Providers of tiles	21
Using «Provider - Custom»	22

Custom URL Wizard.....	22
Styled Google Maps	22
Using offline	23
Preparation of tiles using GMapCatcher	23
Preparation of tiles for offline use manually	23
Caching the current state of map	23
Other methods for offline usage.....	23
Markers	24
2D markers	24
3D markers	24
Animated markers.....	25
Cache	26
Tiles	26
Elevations	26
Custom data	26
Using the Location Service (GPS)	27
Elevations	27
Buildings	27
Third-party assets.....	28
Bolt	28
Curved UI.....	28
EasyTouch.....	28
Fingers - Touch Gestures for Unity.....	29
NGUI: Next-Gen UI	29
Playmaker.....	29
Real World Terrain	29
TouchScript.....	30
uPano	30
Third-party services.....	31
Google API.....	31
Open Street Map Overpass API.....	32
Other third-party web services	33
Online Maps API.....	34
Changes to source code of Online Maps.....	34

Updating versions.....	35
Troubleshooting.....	36
Known issues.....	36
About KML.....	38
Your problem is not listed.....	38
Support.....	38
Other Infinity Code assets.....	39
Huge Texture.....	39
Mesh to Terrain.....	40
Real World Terrain.....	41
Terrain Quality Manager.....	42
Tiny Terrain.....	43
Tree Tool.....	45
Ultimate Editor Enhancer.....	46
uPano.....	48
Final words.....	49
Special thanks.....	49
Links.....	49

Overview

Introduction

Online Maps - is a universal mapping solution that is suitable for all applications and games that use a map.

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

The remaining platforms have not been tested and their performance is not guaranteed.

Online Maps contains:

- **Online Maps Component** – it is core. This component is fully controlling 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 extend the functional of the map. For example: GPS (Location Service), building, cache.

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

Several instances of the map in the scene

Starting with Online Maps v3.6, you can have any number of active maps of any type at the same time.

But we ask you not to abuse this opportunity and use more than one instance of the map only in special (very rare) cases when this cannot be done using one instance of the map.

Using more than one map can dramatically affect performance and memory usage.

Important: You can have only one instance of Cache and Key Manager per scene.

Updating project from Online Maps v1.x and 2.x

Online Maps v3 has a lot of changes in API required for the further development and support of asset.

Before importing Online Maps v3, be sure to remove Online Maps v1.x or v2.x.

Import Online Maps v3.5 (deprecated APIs has been removed in Online Maps v3.6). You can request Online Maps v3.5 from support (support@infinity-code.com) indicating that you want to receive Online Maps v3.5 and enclosing your invoice number or order number.

After importing, you are likely to get several «Obsolete warnings».

Do not be scared. We are very careful about changes to API, and your project will work correctly.

Follow the warning messages to correct your scripts that use Online Maps API.

Then import the latest version of Online Maps over Online Maps v3.5.

Controls

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

Types of controls:

- **2D:** SpriteRenderer, UIImage, UIRawImage, NGUI.
- **3D:** Texture, Tileset.

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

Important: Map can have only one control.

To use NGUI Control, you must have NGUI in the project. This asset is not included in the package and you must purchase it in Unity Asset Store.

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 to 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 to texture. Supports 3D markers and elevation data. **Recommended for use on mobile devices and WebGL.**

Quick creation of map using Wizard

Select «Hierarchy / Create / Infinity Code / Online Maps / Map Wizard».

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.

For Drawing to Texture mode: In section «**Create texture**» select size and click «**Create**». Will create a new texture in the root of the project named «**OnlineMap**» and added to the source display. If a file with the same name already exists, it will be replaced.

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

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**».

Start 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.

Limitations of WebGL

For WebGL all requests to Bing Maps work through proxy on **infinity-code.com**. It is made because for WebGL server should return a response with the header:

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

Use of this header is not implemented in Bing Maps.

For all other platforms and services, requests go directly to services.

Currently, your proxy server does not have limitations on the number of requests. But with a strong increase in the load on our servers, we can add such limitations.

All requests to the server are cached. Currently cache life is four days, but it can be changed.

Important: for better performance and faster responses, it is best when you use proxy scripts located on your server. If you want to get a server-side scripts, please contact support. In this case, you will not depend on the work of our server.

Interaction with the map

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, Android and Windows Phone.

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 Online Maps Camera Orbit component.

Important: Online Maps does not use a rotation gesture. To rotate the camera around the map spread gesture is used. If you want to use the rotation gesture, use some gesture framework, which has integration with Online Maps (for example, EasyTouch or TouchScript).

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**).

Using uGUI Image 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 to GameObject.

Display Tileset on UI

Tileset is not a UI element and you cannot directly use this on Canvas.

To display Tileset to Canvas you need to do the following:

1. Create a new camera for the map, and set it in Tileset Control / Camera.
2. Place map GameObject on a new layer.
3. In Map Camera / Culling Mask, select only map layer, and in the main camera, disable map layer.
4. Map Camera / Clear Flags - Solid Color.
5. In some cases, you need to choose Map Camera / Rendering Path - Deferred.
6. Create a Render Texture, and set it in Map Camera / Target Texture.
7. On Canvas, create a Raw Image, and select Raw Image / Texture - Render Texture.
8. Add OnlineMapsRawImageTouchForwarder component and specify the references to the map, Raw Image and Render Texture.

OnlineMapsRawImageTouchForwarder converts the input from main Screen Space to Render Texture Screen Space.

Scriptable Render Pipeline (URP, HDRP)

Scriptable Render Pipeline is not compatible with the standard render pipeline, and when the scene starts you will see a pink rectangle.

Tileset Control can automatically detect this problem and fix this automatically.

To manually work around this problem, you need to add Online Maps PBR (PBR Bridge) component and specify the shaders for PBR in Online Maps Tileset Control / Materials & Shaders:

Tileset Shader - TilesetPBRShader.

Marker Shader - TilesetPBRMarkerShader.

Drawing Shader - TilesetPBRDrawingElement.

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.

Presets for all providers and map types are provided for testing purposes.

The legal way is to use Google Maps tiles in your applications:

<https://developers.google.com/maps/documentation/tile/>

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

Google API Key is required to use Google's web services.

You can create your Google API key in Google Console:

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

In the Google Console, enable APIs you want to use.

Note: For Google Places API you need to enable Google Places API Web Service.

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

You can create your Bing Maps API Key here:

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

Description of fields

This section contains the description of the fields of the main components of Online Maps.

Online Maps Component

This is the main component of the map and should be used in any case, no matter what result you want to achieve.



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

2. Help - documentation, API Reference and quick access to the built-in update system.

3. Source – source of tiles (Online, Resources, Resources and Online, Streaming Assets, Streaming Assets and Online).

4. (Only for WebGL) Use Proxy – Use a proxy server to bypass restrictions WebGL.

5. (Only for WebGL) Proxy – url of proxy server.

6. Provider – provider of tiles.

7. Type – type of map.

8. Label – display the names of objects on the map.

9. Language – language names on the map.

10. Latitude / Longitude – coordinates the center point of map.

11. Zoom – the current zoom level.

12. (Only for Target – Texture) Texture – texture, which will draw maps.

13. (Only for Target - Tilesset) 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 256.

14. (Only in playmode) Redraw - causes a forced redraw of the map.

15. (Only in playmode) Cache tiles to Resources – caches the current map tiles to «Resources».

16. (Only in playmode) Save state – saves the current state of map. After click you can choose what you want to save.

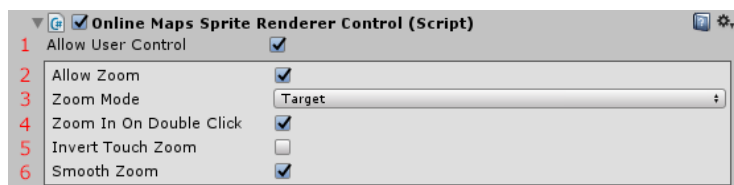
17. (Only for Target - Texture) Create texture – this section allows you to create a new texture, which will draw maps.

18. Filename – texture filename. If the file exists, it is overwritten.

19. **Width / Height** – size of the new texture.
20. Create texture.
21. **Advanced** – section, which contains additional settings.
22. **(Only for Target - Texture) Redraw on play** – specifies whether the texture is redrawn immediately at startup scene, or only after changing the position or zoom.
23. **Use Current Zoom Tiles** – by default (disabled) Online Maps uses parent tiles to improve the map display when changing zoom. You can prevent this by enabling this option.
24. **Traffic** – specifies whether to display the layer with information about traffic congestion.
25. **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.
26. **Default Tile Texture** – texture used for the tile, until the current tile texture is not loaded.
27. **Tooltip Background** – background texture of a tooltip.
28. **Show Marker Tooltip** – when should show a tooltip (hover, pressing, always or never).
29. **Troubleshooting** – in this section contain settings that will help solve the problem. Use this section only if you have problems.
30. **Software JPEG Decoder** – to load JPEG will be used software decoder. Software decoder is much slower than the hardware.
31. **Render In Thread** - calculation of map will be in a separate thread.
32. **Non Interact Under GUI** – prevents events under GUI elements.
33. **Stop Playing When Scripts Compile** – when Unity Editor compiles scripts, Online Maps stops PlayMode, to prevent crashes Unity Editor.

2D Controls

Displays a map on 2D sources such as uGUI, NGUI, SpriteRenderer.



zoom will change.

4. **Zoom In On Double Click** – specifies whether the map zoom in by double-clicking (double-tap on mobile devices).
5. **Invert Touch Zoom** – reverses direction of touch zoom.
6. **Smooth Zoom** - use a smooth touch zoom on mobile devices.

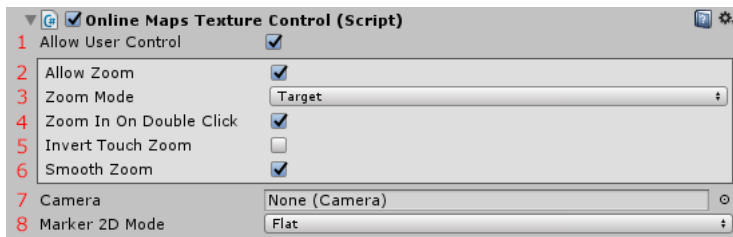
1. **Allow User Control** – specifies whether the user can change the coordinates of map.

2. **Allow Zoom** – specifies whether the user can change the zoom map.

3. **Zoom Mode** – the anchor point based on which the

3D Controls

Displays a map on a plane.



(flat or Billboard).

1-6. See the description of the 2D Control fields.

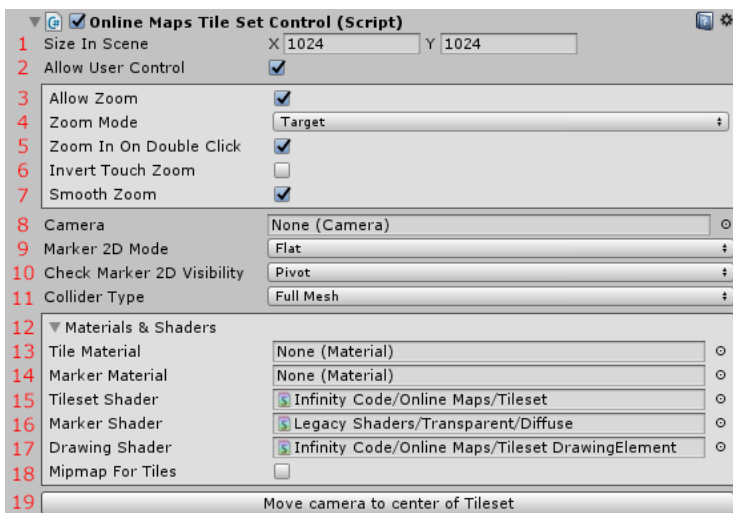
12. Camera – camera which will interact with the Control. If the camera is not specified, will be used Main Camera.

13. Marker 2D Mode - display mode of markers

Tileset Control

Displays the map as a dynamic mesh.

Highly recommended for use in any type of application, has much greater performance, requires less memory, and has many exclusive features.



1. Size In Scene - the size that the map will occupy in the scene.

2-7. See section 2D Control.

8-9. See section 3D Control.

10. Check Marker 2D Visibility - how the visibility of the 2D marker on the map will be checked.

11. Collider Type - type of collider of the map.

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

13. Tile Material – material that will be used for tiles.

14. Marker Material - material that will be used for markers.

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

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

17. Drawing Shader – shader that will be used for Drawing Elements.

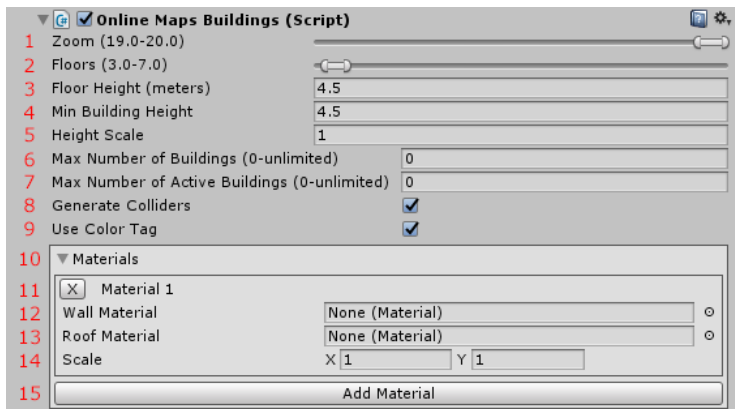
18. Mipmap For Tiles - whether mipmap will be used for tile textures.

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

Buildings

Displays 3D buildings on a map based on Open Street Map data.

Requires Tileset control.



1. Zoom - zoom of map in which the building will be shown.

2. Floors – number of floors of buildings.

3. Floor Height – floor height.

4. Min Building Height - the minimum height of the building.

5. Height Scale – scale of building.

6. Max Number of Buildings – the maximum number of buildings that will be used.

7. Max Number of Active Buildings – the maximum number of buildings that can be displayed on a map

simultaneously.

8. Generate Colliders - whether to generate colliders for buildings.

9. Use Color Tag - is it necessary to use a color tag from OSM.

10. Materials – Building materials. When generating building material is selected randomly.

11. Material number, and a button that allows you to remove the material.

12. Wall Material – wall material.

13 Roof Material – roof material.

14. Scale – scale of wall texture.

15. Add new building material.

Cache

Caches tiles in memory and the file system, and elevations and custom data in the file system.

1 Use File Cache ☒

2 Size (mb) 100

3 Unload (%) 30

4 Cache Location Persistent Data Path

5 Tile Path TileCache/{pid}/{mid}/{lbs}/{lng}/{zoom}/{x}/{y}

6 Available Tokens

7 Current Size (mb) 56.11

8 Clear

9 Use Memory Cache ☒

10 Size (mb) 10

11 Unload (%) 30

12 Current Size (mb) 0.48

13 Clear

14 Custom Cache

15 Size (mb) 100

16 Unload (%) 30

17 Current Size (mb) 0.40

18 Clear

1. Use File Cache - whether to use file cache. Not supported for WebGL.

2. Size - size of file cache.

3. Unload (%) - the percentage of the cache that will be unloaded.

4. Cache Location - where to store the cache.

5. Tile Path - a template for storing tiles in the cache.

6. Available Tokens - available tokens for the template.

7. Current Size - the current size of the file cache.

8. Clear - clear the file cache.

9. Use Memory Cache - whether to use the memory cache

10. Size - the size of the memory cache.

11. Unload (%) - the percentage of the cache that will be unloaded.

12. Current Size - the current size of the memory cache.

13. Clear - clear the memory cache.

14. Custom Cache - stores elevation data and custom data in the file system. Not supported for WebGL.

15. Size - the size of the custom cache.

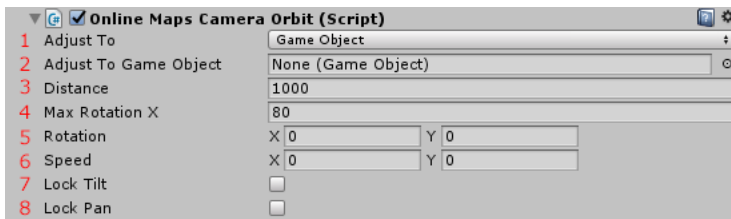
16. Unload (%) - the percentage of the cache that will be unloaded.

17. Current Size - the current size of the custom cache.

18. Clear - clear the custom cache.

Camera orbit

Rotates the camera around the center of the map. Requires Tilesset Control.



1. Adjust To - the place where the camera will look.

2. Adjust to GameObject - GameObject the camera will be looking at.

3. Distance - the distance from the point where the camera is looking to the camera.

4. Max Rotation X - the maximum tilt of the camera.

5. Rotation - the angle of rotation of the camera.

X - tilt. 0 - perpendicular to the map, 90 - parallel to the map.

Y - rotation (pan) of the camera clockwise.

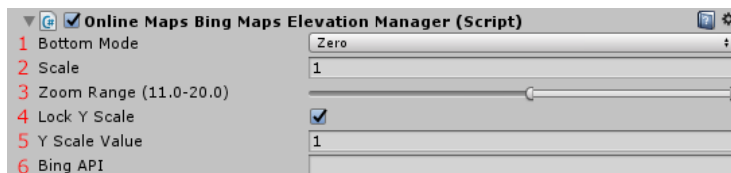
6. Speed - the rate of change of the value.

7. Lock Tilt - prevent tilt (X) change.

8. Lock Pan - prevent rotation(Y) change.

Elevation manager

Displays elevations on the map. Requires Tilesset Control.



1. Bottom Mode - the rule for determining the lower point of the map.

2. Scale - scale of heights.

3. Zoom Range - the zoom range of the map for

which elevations will be displayed.

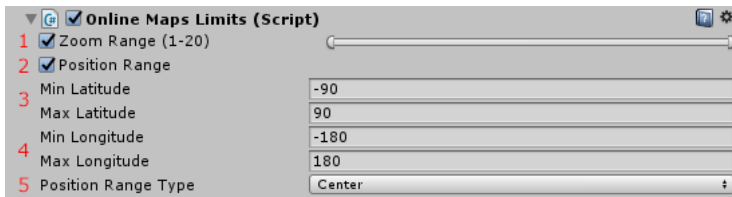
4. Lock Y Scale - use fixed yScale.

5. Y Scale Value - the value of yScale.

6. Bing API - Bing Maps API key.

Limits

Limits the range of locations and zoom for the map.



3. The range of valid values of latitude.

4. The range of valid values of longitude.

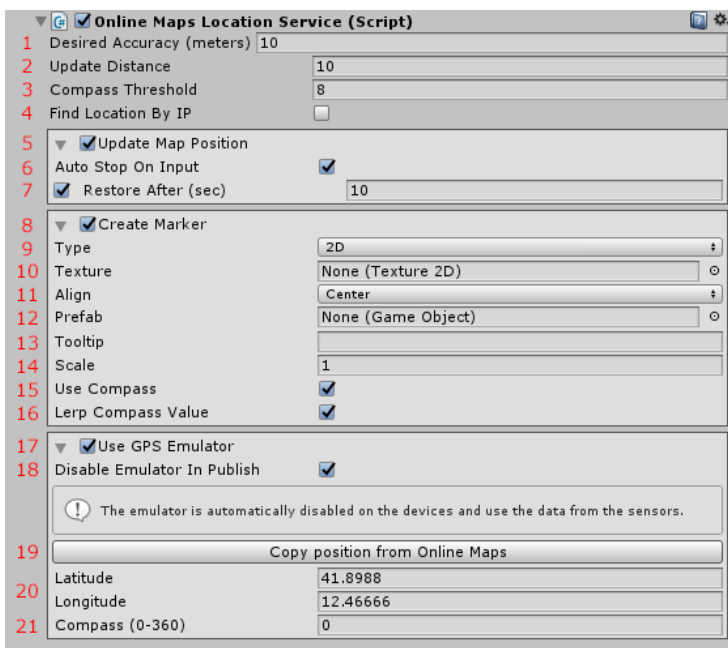
5. **Position Range Type** - the rule for checking the position values (by the central point or along the boundaries).

1. **Zoom Range** - whether to limit the zoom map, and the range of valid values.

2. **Position Range** - whether to limit the position of the map.

Location Service

Provides interaction with the GPS and the compass of the device, and emulates the values of these sensors for Unity Editor.



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

3. **Compass Threshold** - compass threshold (degrees) to prevent shaking.

4. **Find Location By IP** - looking for the user's location by IP address. It has a very low accuracy.

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

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

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

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

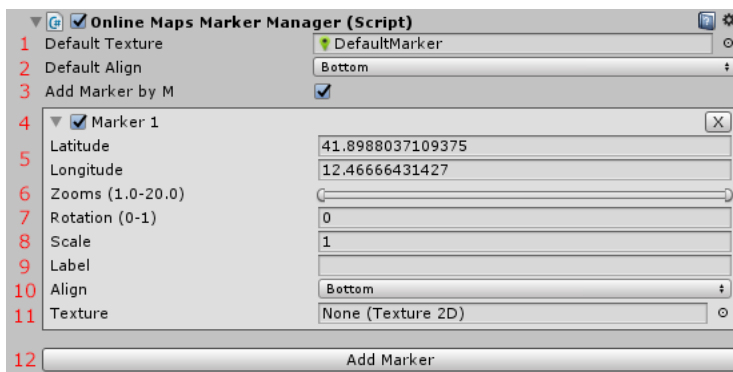
9. **Type** – type of marker (2D or 3D).

10. **Texture** – texture of 2D marker.

11. **Align** – align of 2D marker.
12. **Prefab** – GameObject, which will be used for 3D marker.
13. **Tooltip** – tooltip of marker.
14. **Scale** - marker scale.
15. **Use Compass** – rotate marker on the value of the compass.
16. **Lerp Compass Value** - smooths out changes in compass values to avoid jitter on mobile devices.
17. **Use GPS Emulator** – emulates the value of GPS and compass. Works only in Unity Editor.
18. **Disable Emulator In Publish** - disables the emulator when a project is published.
19. **Copy position from Online Maps** - copies the current position from Online Maps.
20. **Latitude / Longitude** - emulated GPS coordinates.
21. **Compass** - emulated compass value.

Marker manager

2D markers that are present on the map.



1. **Default Texture** - texture that will be used if marker texture is not specified.
2. **Default Align** - alignment that will be used for new markers.
3. **Add Marker By M** - create a new marker under the cursor in the playmode when you press M.
4. The index of the marker and the button to delete the marker.

5. **Latitude / Longitude** - latitude and longitude of the marker.
6. **Zooms** - map zoom, at which the marker will be displayed.
7. **Rotation** - rotation of the marker relative to the alignment point in a clockwise direction.
8. **Scale** - marker scale.
9. **Label** - the text that will be shown when you hover on the marker.
10. **Align** - alignment of the marker.
11. **Texture** - texture marker.
12. **Add Marker** - creates a new marker in the center of the map.

Marker 3D Manager

3D markers that are present on the map.

▼ ☒ Online Maps Marker 3D Manager (Script)

1 Default Prefab None (Game Object)

2 Default Scale 1

3 Add Marker3D by N ☒

4 ▼ ☒ Marker 1

5 Latitude 41.8988037109375

6 Longitude 12.46666431427

7 Zooms (1.0-20.0)

8 Scale 1

9 Rotation Y 0

10 Label

10 Prefab None (Game Object)

11 Add Marker

1. Default Prefab - prefab, which will be used if the prefab marker is not specified.

2. Default Scale - the scale that will be used for new markers.

3. Add Marker3D By N - create a new 3D marker under the cursor in the playmode when you press N.

4. The index of the marker and the button to delete

the marker.

5. Latitude / Longitude - latitude and longitude of the marker.

6. Zooms - map zoom, at which the marker will be displayed.

7. Scale - marker scale.

8. Rotation Y - clockwise rotation of the marker by the vertical axis.

9. Label - the text that will be shown when you hover on the marker.

10. Prefab - prefab marker.

11. Add Marker - creates a new marker in the center of the map.

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 of tiles 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.
- **Streaming Assets** – from the folder «**StreamingAssets**».
- **Streaming Assets and Online** – if possible tiles will be loaded from the folder «**StreamingAssets**». 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 or StreamingAssets] / OnlineMapsTiles / {ZOOM} / {TileX} / {TileY} .png** ».

Important: in the path pattern for tiles in «Resources», you **do not need** to specify the file extension, and in «StreamingAssets» **you need** to specify the file extension.

Providers of tiles

Online Maps can work with any tile provider that has a projection of WGS 84 Spherical Mercator (Web Mercator, EPSG: 3857) or WGS 84 Ellipsoid Mercator (EPSG: 4326), and returns bitmap PNG or JPG.

Currently, it is most of the existing maps.

Important: Vector tiles are not supported. We are working on it and will try to add support for vector tiles in future versions.

List of known providers:

Supported: AMap, ArcGIS (ESRI), CartoDB, Google Maps, Hydda, Mapbox, MapQuest, Mapy.CZ, Nokia Maps (here.com), Open Map Surfer, Open Street Map, Open Topo Map, Open Weather Map, Stemen, Sputnik, Thunderforest, TianDiTu, Virtual Earth (Bing Maps), Yandex Maps.

Not supported (have another projection): Baidu Maps.

This is not a complete list, and will be updated in the future.

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 from your source.

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 section «**Styled Google Maps**».

Custom URL Wizard

For Google Maps and Mapbox you can use «Custom URL Wizard». To open «Custom URL Wizard», click the button the right of «Custom Provider URL». Enter URL of the tile that you got in step 3, and click «Apply» button.

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: <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:
<https://maps.googleapis.com/maps/vt?pb=!1m5!1m4...>
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:

<https://maps.googleapis.com/maps/vt?pb=!1m5!1m4!1i{zoom}!2i{x}!3i{y}!2m3!...>

8. Make sure URL ends at «!4e0» (inclusive). If you have characters after this, delete them.

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

Using offline

Preparation of tiles using GMapCatcher

To prepare the tiles convenient to use GMapCatcher.

<https://github.com/heldersepu/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**» or «**StreamingAssets / OnlineMapsTiles**», in the format required for use.

Preparation of tiles for offline use manually

Place your tiles in the format «**[Resources or StreamingAssets] / 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

2D markers

To work with markers, use Online Maps Marker Manager component.

Click «Add Marker» to create a new marker in the center of the map.

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.

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

3D markers

3D markers are a GameObject in a scene automatically positioned on a map based on geographic coordinates.

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

To work with markers, use Online Maps Marker 3D Manager component.

3D markers can be created from any prefab in Assets.

Important: 3D markers containing UI elements will not be positioned correctly in the Canvas. If you want to use UI elements as markers, use the uGUI Custom Marker Engine.

<http://infinity-code.com/atlas/online-maps/pages/uGUICustomMarkerEngineExample.html>

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

3D markers can have three types of sizes:

Scene - The marker will always have the same size in the scene for all zoom.

Real World - The marker will have a size equal to the specified scale for the zoom - 20. For each previous zoom, the size will be halved.

Meters - The marker will have a size in meters equal to the specified scale. **Important:** The size of the markers is calculated as 1 Unity World Point of the original prefab equal to 1 meter. This means that if

your prefab had a size of X (Y, Z) equal to 0.5, it would have a size of X (Y, Z) equal to half the specified size in meters.

If half 3D marker are below the map, do the following:

1. Create a new empty GameObject.
2. Create an instance of your marker in the scene inside the new GameObject.
3. Move the internal marker up so that the position (0, 0, 0) is at the bottom of the marker.
4. Create a prefab from GameObject created in step 1.
5. Remove the external prefab from the scene.
6. Use the new prefab for the marker.

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.

Cache

Tiles

Online Maps has built-in system of caching tiles that allows you to greatly reduce the number of requests to tile servers and speed up the application.

To use caching, add «Online Maps Cache» component.

Caching works in two modes: memory cache and file cache. You can use both modes together or only one of the modes.

Important: For WebGL, the file cache is not supported because there is no access to the file system on this platform.

Elevations

Elevation data caching is implemented for tiled elevation managers: Bing Maps Tiled Elevation Manager, Mapbox Elevation Manager.

Bing Maps Elevation Manager do not support caching.

Data is stored in a custom cache.

Important: For WebGL, the elevation cache is not supported because there is no access to the file system on this platform.

Custom data

Using the Online Maps API, you can cache any user data.

Data is stored in the file system.

To add data to the cache, use `OnlineMapsCache.Add`.

To get data from the cache, use `OnlineMapsCache.Get`.

Important: For WebGL, the custom cache is not supported because there is no access to the file system on this platform.

Using the Location Service (GPS)

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

Online Maps can find the location by IP (by default enabled). Find location by IP has a very low accuracy (in most cases up to the city), but can be very useful when the GPS is not available.

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.

Elevations

Online Maps supports elevation data from Bing Maps and Mapbox.

To create a map with elevations you must use Tileset Control.

Add Online Maps {SOURCE OF DATA} {Tiled?} Elevation Manager component, and specify your API key (if required).

Tiled Elevation Manager supports elevation data caching: Bing Maps Tiled Elevation Manager, Mapbox Elevation Manager.

Important: Public Windows App or Public Windows Phone App have the 50.000 transaction within 24 hours. With the other chooses there's only 125.000 transactions within a year and the key will expire if exceeding it.

Buildings

Generation of buildings is only available using Tileset Control.

Add Online Maps Buildings component and specify materials for walls and roofs.

Buildings are created dynamically based on Open Street Map data.

Start the scene and when the zoom maps are in the specified range, buildings will be shown.

Important: Do not use too much range. A map can have hundreds and thousands of buildings, which will adversely affect the performance of the application.

Third-party assets

Online Maps has integration with many third-party assets.

See below for instructions on how to enable and use integration with each asset.

To disable integration, delete the corresponding key in «Edit / Project Setting / Player / Other Settings / Scripting Define Symbols».

Bolt

<https://assetstore.unity.com/packages/tools/visual-scripting/bolt-87491>

Bolt can use the variables and methods of Online Maps without any integration.

But the problem is that Bolt does not yet support System.Action. This means that you can not use events of map, markers, etc.

To work around this, import «Online Maps Bolt Integration Kit».

Before importing «Online Maps Bolt Integration Kit», make sure Bolt is present in your project.

Select «GameObject / Infinity Code / Online Maps / Packages / Bolt Integration Kit».

Video how to use the Online Maps events in Bolt:

<http://forum.infinity-code.com/viewtopic.php?id=820>

Curved UI

<https://assetstore.unity.com/packages/tools/gui/curved-ui-vr-ready-solution-to-bend-warp-your-canvas-53258>

Online Maps UIImage Control and Online Maps UIRawImage Control support work with Curved UI.

Import Curved UI in the project and in the inspector of control, click «Enable Curved UI».

EasyTouch

<https://assetstore.unity.com/packages/tools/input-management/easy-touch-5-touchscreen-virtual-controls-3322>

EasyTouch allows you to add Online Maps support for standard gestures, such as the rotation gesture.

Add OnlineMapsEasyTouchConnector and click «Enable EasyTouch».

Fingers - Touch Gestures for Unity

<https://assetstore.unity.com/packages/tools/input-management/fingers-touch-gestures-for-unity-41076>

This asset allows you to add Online Maps support for standard gestures, such as the rotation gesture.

Add OnlineMapsFingersTouchGesturesConnector and click «Enable Fingers - Touch Gestures».

NGUI: Next-Gen UI

<https://assetstore.unity.com/packages/tools/gui/ngui-next-gen-ui-2413>

Add the desired control on the GameObject with the map. Enable support by pressing «**Enable NGUI**».

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

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

Playmaker

<https://assetstore.unity.com/packages/tools/visual-scripting/playmaker-368>

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

Select «**GameObject / 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**».

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.

Real World Terrain

<https://assetstore.unity.com/packages/tools/terrain/real-world-terrain-8752>

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.

TouchScript

<https://assetstore.unity.com/packages/tools/input-management/touchscript-7394>

TouchScript allows you to add Online Maps support for standard gestures, such as the rotation gesture.

Add OnlineMapsTouchScriptConnector and click «Enable TouchScript».

uPano

<https://assetstore.unity.com/packages/tools/integration/upano-126396>

Allows you to add support for Google Street View to the map.

Add OnlineMapsPanoConnector and click «Enable uPano».

If necessary, specify prefabs for directions and transitions of opening and closing panoramas.

Third-party services

Google API

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

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

Google Geocode - 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 - 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 – provides detailed information about the object by place id, or reference.

Google Place Photo – is a read-only API that allows you to add high quality photographic content to your application. The Place Photo service gives you access to the millions of photos stored in the Places and Google+ Local database.

Google Places Autocomplete - 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.

Google Roads API - allows you to map GPS coordinates to the geometry of the road, and to determine the speed limit along those road segments.

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 API web services requires Google API Key. You can get them in Google API Console:

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

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 web services

AMap Search - China service for the POI search.

Bing Maps Elevation - provides elevation data for locations, polylines or region in the world.

Bing Maps Location - 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.**

HERE Routing - calculates routes between two or more locations, and provides additional information about the route associated with traffic updates in real time.

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 Directions - 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.

Open Route Service Geocoding - The ORS Geocoding service accesses global databases and transforms a description of a location, such as a place name, street address or postal code, into a normalized description of the location with a point geometry. Additionally ORS Geocoding offers a reverse geocoding service which does exactly the opposite. It transforms the coordinate of a point into the description of the next enclosing object which surrounds the given coordinate. To obtain more relevant results you may also set a radius tolerance around the requested coordinates. Within the ORS API you are able to distinguish between geocoding and reverse geocoding depending on your use case.

QQ Search - China service for the POI search.

What 3 Words - provides a precise and incredibly simple way to talk about location. We have divided the world into a grid of 3m x 3m squares and assigned each one a unique 3 word address.

Online Maps API

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

Order of the values in the coordinates:

- Vector2, Vector3: X - longitude, Y - latitude;
- Parameters of methods: first - longitude, then - latitude.

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

Important: do not modify the scripts in «Examples (API Usage)» folder. You will lose your changes when you update Online Maps. Make a copy of the script with a new file name and class name.

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.

Important: Always make a backup before updating assets.

Click «**GameObject / Infinity Code / Online Maps / Check Updates**» or «**Online Maps / Help / Check Updates**», to open the window checking for updates. Enter your Invoice Number или Order 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://assetstore.unity.com/orders>

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.

Important: Do not be afraid to Online Maps Beta. Online Maps developed linearly, and all changes (bug fixes, new features, etc.) first published in the beta section. After 1-2 weeks after the release of a new stable version of Unity, we apply all changes to Online Maps Stable.

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.

How to work around the problem:

1. `OnlineMapsDrawingElement.checkMapBoundaries = false`.

This mode works correctly only if the lines do not intersect.

2. Use «Draw to Texture» mode.
3. (Not recommended) Use Drawing as Overlay.

Markers and other objects are incorrectly positioned on the map

Most likely you use the map size to equal $N * 256$.

This is a very complex and deep problem. We are working on a solution.

Currently, the solution is to use the map size equal to $N * 256$.

I created a marker or another object on the map, but I do not see it

Please make sure you do not mix up the latitude and longitude.

If you use `Vector2`: X - Longitude, Y - latitude.

If you use a method that takes a double: first - longitude, then - latitude.

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.

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.

How to make shadows on a map

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.

Playmaker does not save FSMObject (markers). Debug shows «None»

Actually FSMObjects saved, and you can use it. Just Playmaker does not show it.

This is a bug in Playmaker, and we hope that sometime this will be fixed.

You enabled support of asset, that does not have in the project, or remove asset without disabling it in Online Maps

Import required asset, or delete the relevant key in «Edit / Project Setting / Player / Other Settings / Scripting Define Symbols».

White map on iOS 12:

Try this solution:

<https://forum.infinity-code.com/viewtopic.php?pid=8409#p8409>

If this works for you, please send confirmation to support in the way that is convenient for you.

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) or use forum (<http://forum.infinity-code.com>). We will try to fix all errors and release an update as soon as possible.

Support

We provide support by email (support@infinity-code.com) in English and Russian languages, or using forum (<http://forum.infinity-code.com>) in English language.

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

Please, 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.

Other Infinity Code assets

Huge Texture



<https://assetstore.unity.com/packages/tools/input-management/huge-texture-163576>

Huge Texture allows you to import and use textures larger than 8192x8192px.

How it works:

When importing a texture, Huge Texture splits the texture into pages and saves it as a Texture Array. Texture Array is combined on the shader side, which does not create extra draw calls and has almost no effect on performance.

Requirements:

- Not all platforms support Texture Arrays.

The list of supported platforms is here:

<https://docs.unity3d.com/Manual/SL-TextureArrays.html>

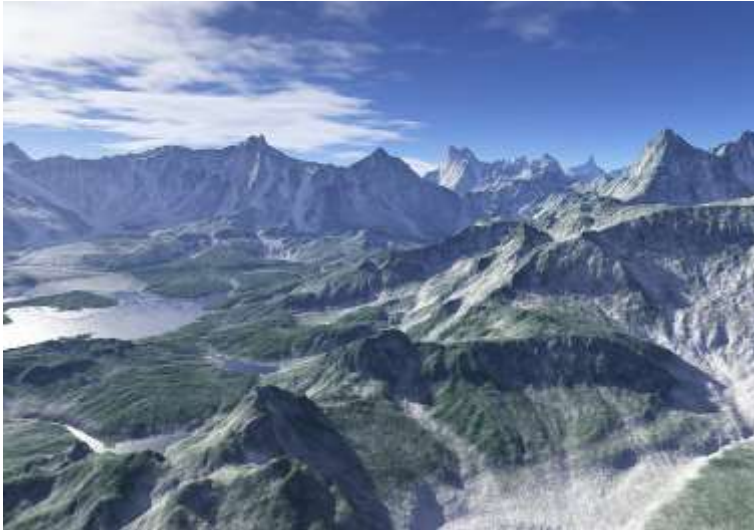
- The field must accept Texture (not Texture2D), and the component must accept custom material or shader.

For example, Huge Texture can be used in Mesh Renderer, Raw Image, etc.

Features:

- The maximum size of the texture up to 16384x16384px;
- Standard Render Pipeline, Universal Render Pipeline (URP, LWRP), High Definition Render Pipeline (HDRP);
- Compressed and uncompressed formats;
- Does not produce extra draw calls. You will have as many draw calls as you would with a regular texture;
- API for working with huge textures, as with regular Texture2D;
- Built-in update system.

Mesh to Terrain



<https://assetstore.unity.com/packages/tools/terrain/mesh-to-terrain-7271>

Mesh to Terrain is a tool for easily and quickly converting a 3D terrain model created in 3ds Max, Terragen or any other editor to Unity Terrains.

Mesh to Terrain can convert textures to SplatPrototypes (Terrain Layers), generate terrain from several models and split the model into several terrains.

Features:

- Unlimited number of models and terrains used in the component;
- Terrains are created in the same place where the meshes;
- Convert the textures of models to textures of terrain;
- Seamless result, when working with multiple terrains;
- Manually adding models or automatic detection of all models in the layer;
- No need to manually add the components of physics;
- Works with [Relief Terrain Pack](#) (optional);
- Built-in update system.

Real World Terrain



<https://assetstore.unity.com/packages/tools/terrain/real-world-terrain-8752>

Real World Terrain is a tool for automatically creating high-quality terrains, meshes, Gaia stamps and RAW files based on real-world data with global coverage.

Incredibly fast and easy to use, and allows you to create high-quality terrains in a couple of clicks. In addition, Real World Terrain can create buildings, roads, trees, grass, and rivers based on Open Street Map data.

Real World Terrain is incredibly powerful and flexible. It has a powerful Editor API to automate the generation of terrains, and Runtime API positioning objects by coordinates, etc.

Real World Terrain has integration with the best assets of the Asset Store, which gives almost unlimited possibilities in the generation of terrains.

Features:

- Elevation Heightmaps:
 - Bing Maps resolution of max 10 meter per pixel;
 - Mapbox;
 - SRTM v4.1 resolution of 90 meters per pixel;
 - SRTM30 resolution of 30 meters per pixel.
- Texture providers: ArcGIS, DigitalGlobe, Map Quest, Mapbox, Mapy.CZ, Nokia Maps (here.com), Virtual Earth (Bing Maps), Open Street Map + ability to download tiles from custom url;
- Satellite images resolution of max 0.25 meter per pixel;
- Can create: Unity Terrains, Meshes, [Gaia](#) stamps, RAW files;
- Can create objects based on Open Street Map:
 - Editable roads for [EasyRoads 3D v3](#) and [Road Architect](#);
 - Editable buildings for [BuildR2](#) (editable) or built-in building engine;
 - Rivers;
 - Trees;
 - Grass.
- A tool to select area directly on Google Maps;

- A lot of extra tools for working with coordinates, objects and postprocessing;
- Unlimited number of generated terrains;
- Integration (optional): [BuildR2](#), [EasyRoads 3D v3](#), [Gaia](#), [Online Maps](#), [Playmaker](#), [Relief Terrain Pack](#), [Road Architect](#), [uContext](#), [Vegetation Studio](#), [Vegetation Studio Pro](#), [VolumeGrass](#), [WorldStreamer](#).

Terrain Quality Manager



<https://assetstore.unity.com/packages/tools/terrain/terrain-quality-manager-28949>

Terrain Quality Manager allows you to change the resolution of Heightmap, Detailmap, Alphamap and Basemap, **without losing data**.

If you need to increase the quality of terrain, or optimize terrain for better performance, with this tool you can do it.

Features:

- Allows you to change resolution of Heightmap, Detailmap, Alphamap, Basemap and Resolution Per Patch, without losing data;
- Works for single terrain, all terrains in scene, all terrains in project;
- Allows you to scale up and scale down terrain maps;
- Seamless result, when working with multiple terrains;
- Very easy to use.

Tiny Terrain



<https://assetstore.unity.com/packages/tools/terrain/tiny-terrain-236464?aid=1100liByC>

Tiny Terrain is a lossy Terrain Data compression technology for Unity Terrain Engine, which allows you to reduce the size of the data that Terrain takes in the project many times and significantly reduce the size of your application.

You pay a little CPU time and additional memory usage, but save tens and hundreds of megabytes for each Terrain in the project.

Even though the compression is lossy, in most cases you won't see any visual difference at all.

A very high compression ratio is achieved due to the author's data preparation algorithms and then compression using Brotli.

Even if your build or AssetBundle is already compressed using LZ4, LZ4HC or LZMA, Tiny Terrain will be able to reduce its size. Of course the difference in size won't be as huge as with the uncompressed data, but it's still pretty big.

The asset has highly optimized code using direct memory access, which allows for fast compression and decompression, and minimizes additional memory usage. On average, one Terrain is loaded in about 1 second.

Tiny Terrain is not a 3rd party Terrain Engine, and you can use Unity Terrain API and any terrain tools you like, such as Real World Terrain, Gaia, Vegetation Studio, etc.

Using Tiny Terrain is very easy, and anyone can do it, even if it is their first day using Unity.

Just add Tiny Terrain Loader component to Terrain and click Compress.

If you are an advanced user or programmer, Tiny Terrain has events and API that will help you take full control of the loading process.

Features:

- High compression ratio from 3:1 to 30:1 with no visible difference, and up to 100:1 with visible quality loss:
 - Heightmap. Compression from 2:1 to 10:1;
 - Alphamap. Compression from 5:1 to 50:1;
 - Detailmap. Compression from 5:1 to 20:1 lossless. Note: Unfortunately, I don't have exact information on how Unity Terrain Engine stores detailmaps, so compression ratio is based on my observations;
 - Trees. Compression 3:1;
 - Holes. Compression from 1000:1 to 10000:1 lossless;
- Minimal visual differences. In most cases, you won't see visual difference at all;
- Does not affect rendering and can be used with any Render Pipeline: Built-in RP, URP, HDRP, custom RP;
- Fast compression and decompression;
- Synchronous, asynchronous and multi-threaded decompression;
- Terrain API and third-party tools for Terrain can be used;
- Third party terrain shaders like RTP and GTS can be used;
- API for compression and decompression;
- Built-in update system.

Platforms:

The asset was tested on PC, Mac, Android, iOS, WebGL.

The rest of the platforms have not been tested, but the asset should also work well on them.

Tree Tool



<https://assetstore.unity.com/packages/tools/terrain/tree-tool-221137>

Tree Tool allows you to manage each tree individually.

Using this tool, you can perform the following actions on trees:

- Move (with and without snapping to terrain);
- Scale;
- Change color;
- Change prefab;
- Extract;
- Inject;
- Duplicate;
- Delete.

In addition, the tool can perform the following mass actions:

- Clear;
- Extract;
- Inject.

Tree Tool is compatible with all assets that create trees for Terrain.

Ultimate Editor Enhancer



<https://assetstore.unity.com/packages/tools/utilities/ultimate-editor-enhancer-141831>

Take your productivity in Unity Editor to the next level with **Ultimate Editor Enhancer** - the ultimate productivity toolkit! This powerful tool simplifies your workflow, improves the built-in behaviors of the editor, and adds new features that will take your editing experience to the next level. With Ultimate Editor Enhancer, you can easily enhance the Scene View, Inspector, Hierarchy, Project and the editor itself. It also offers a range of tools for quick object navigation, creation, and manipulation. Upgrade your editing game and streamline your workflow with Ultimate Editor Enhancer!

Key Features:

- Extremely easy to use. Even a beginner can learn how to use it in a couple of minutes. Many features do not need to be learned at all. They are where they should be in the editor out of the box, but for some reason they are missing.
- Suitable for everyone. It doesn't matter what role you have in the project and at what stage your project is. Ultimate Editor Enhancer helps you right now and saves you hours of chores every day.
- Huge number of tools. Over 100 useful tools for everyone in one asset.
- Seamless integration into the editor workflow. Many features are integrated into the editor so that you instantly forget that it was not in the editor out of the box.
- Excellent performance. Ultimate Editor Enhancer is very heavily optimized and has almost no effect on the speed of the editor.
- Fully customizable. If some feature does not fit your workflow, you can adjust or disable it.
- Superbly documented. The asset has full documentation for each feature, and an illustrated Getting Started.
- Frequent update. We are constantly improving the asset and release new versions every 4-6 weeks.

List of tools:

- **Standalone Tools:** Autosave, Bookmarks, Context Menu, Distance Tool, Editor Icon Browser, Smart Search, Toolbar;
- **GameObject Tools:** Add Component by Shortcut, Collections, Custom Pivot Rotation Tool, Drop To Floor, Duplicate Tool, Group / Ungroup, Missing Script Fixer, Rename + Mass Rename, Pivot Tool, Rotate by Shortcut, Replace, Select Prefab Instances, Selection History;
- **Hierarchy Tools:** Backgrounds, Best Icons, Display Errors and Exceptions, Enable/Disable GameObject, Filter by Type, Headers, Highlighter, Solo Visibility, Tree;
- **Inspector Tools:** Component To JSON, Detect Bounds of Box Collider, Drag from Object Fields, Empty Inspector, Hide Empty Help Button, Hide Preset Button, Inspector Bar, Long Text Fields, Nested Editor, Notes, Quick Add Events, Object Field Selector, Play Clips in Animator, Saving Component at Runtime, Search By Serialized Fields, Transform in World Space;
- **Navigation Tools:** Fast Zoom In/Out, Frame Selected Bounds, Jump To Point, Quick Preview, View Gallery;
- **Scene View Tools:** Custom Tool Switcher, Highlighter, Move To Point, Object Placer, Object Toolbar, Quick Access Bar, Selection Bounds, Select As One, Tool Values, Terrain Editor Brush Size, WAILA + Smart Selection;
- **Project Tools:** Create Folder Button, Create Folder by Shortcut, Create Material Button, Create Script Button, Folder Icons;
- **Runtime Tools:** Stepper, TimeScale Manager;
- **Windows Tools:** Favorite Windows, Maximize Active Window, Recent Windows, Switcher;
- **Improve Built-in Behaviours:** Change Numeric Field value using arrows, Drag and Drop to Canvas, Enum Field Search, Frame All Points of Curve Editor, List of Values in Curve Editor;
- Built-in Update System.

uPano



<https://assetstore.unity.com/packages/tools/integration/upano-126396>

uPano (Unity Panoramic Framework) is a universal solution for displaying dynamic and static panoramas, and creating virtual tours.

uPano is very easy to learn and use, and is great for people who do not have programming experience. Visual Tour Maker lets you create virtual tours in minutes.

In most usage scenarios, you can make interactive panoramas without creating your own scripts. If you have some very specific purpose, uPano has a powerful and easy-to-use API that will allow you to implement any behavior.

Most of the existing types of panoramas are supported: spherical, cylindrical, cubic panoramas on single or six images, cubemap.

Supports Unity 2017.4 LTS and higher.

Supported platforms: Standalone, iOS, Android (including Google VR), Universal Windows Platform, WebGL.

Other platforms have not been tested, but most likely uPano will work well.

Additional features:

- Can display textures larger than 8192px using [Huge Texture](#);
- Visual Tour Maker;
- Interactive elements (Hot Spots, Directions) + Visual Editor;
- Lots of built-in actions for HotSpots and Directions;
- Compound transitions;
- Video panoramas;
- Control with: mouse, touch, keyboard, gyro, compass, UI buttons, UI compass, UI joystick;
- Plugins: Auto Rotate, Downloader, Limits, Multi Camera, Orthographic Camera, Timed Gaze;
- (optional) Integration with [Online Maps](#), [Playmaker](#);
- Built-in update system.

Final words

We sincerely hope that you will enjoy using Online Maps.

If you have any questions or problems, please contact us.

We will try to help you as quickly as possible.

Please do not forget to leave your review in Unity Asset Store.

It is very important for us to have feedback from users to make our assets better.

For other users, it is also very important to make a right understanding of this asset.

Special thanks

Cameron Wise-Maas, Geoff Mortimer, Andrea Leganza, Pino De Francesco, Ging, Paul Iluhin, Quidam, mstrohackerHG, Magic73, Seto, Brian Chasalow.

And to all users who sent us ideas and suggestions, bug reports and corrections.

Many thanks to all of you for helping to make the asset better.

Links

Product page: <https://infinity-code.com/assets/online-maps>

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

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

Support: support@infinity-code.com

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

Twitter: <https://twitter.com/InfinityCodeCom>

YouTube: <https://www.youtube.com/playlist?list=PL2QU1uhBMew--JuETNLzZtKUY1BKG4Dib>

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