WORLD MAP
GLOBE EDITION LITE

Developer's Reference Guide

RELEASE NOTES
V.4.0 / 13.OCT.2015

Introduction

Thank you for purchasing!

World Political Map – Globe Edition Lite is a commercial asset for Unity 5.1.1 and above that allows to:

1. Visualize the frontiers of 177 countries and the location 1249 most important cities in the world.
2. Colorize and also highlight the regions of countries and provinces/states as mouse hovers them.
3. Automatically draw country labels.
4. Add markers and line animations to the globe.
5. Fly to a chosen country, city, province or location. It will make the globe rotate until the destination is reached.
6. Imaginary lines: draw custom latitude, longitude and cursor lines.
7. Lots of customization options: colors, labels, frontiers, provinces, cities, Earth (2 styles included)…
8. Works on Android and iOS.
9. Comprehensive API and easy setup.
10. Dedicated and responsive support forum.

You can use this asset to represent or allow the user choose a location in your game/application, in mission briefings, reports, statistical or educational software, etc.

QuickStart and Demo Scene

1. Import the asset into your project or create an empty project.
2. Open scene “GlobeDemo” in Demo folder.
3. Run and experiment with the demonstration.

The Demo scene contains a WorldMapGlobe instance (the prefab) and a Demo gameobject which has a Demo script attached which you can browse to understand how to use some of the properties of the asset from code (C#).

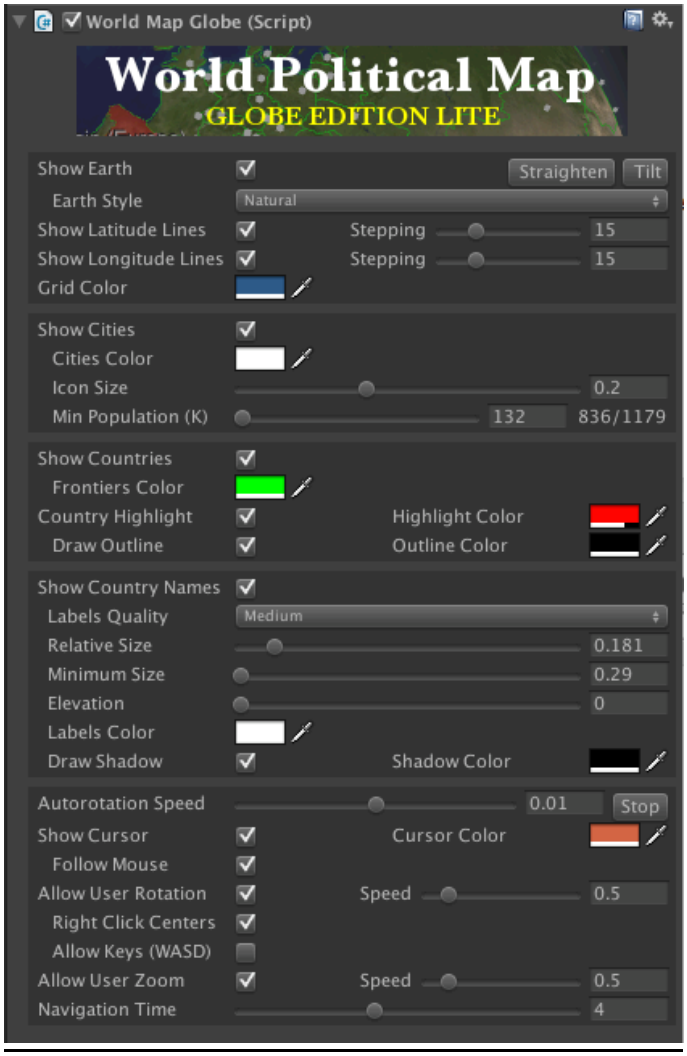
Support & Contact info

We hope you find the asset easy and fun to use.
Feel free to contact us for any enquiry.

Kronnect Games
Email: contact@kronnect.me
Kronnect Support Forum: http://www.kronnect.me
Unity Forum Thread: http://forum.unity3d.com/threads/released-world-political-map-globe-edition.343245/

How to use the asset in your project

Drag the prefab “WorldMapGlobe” from “Resources/Prefabs” folder to your scene.
Select the GameObject created to show custom properties:



Custom Editor Properties

Show Earth: shows/hide the Earth. You can for example hide the Earth and show only frontiers giving a look of futuristic UI.

You may want to not hide the Earth, but instead use the Solid style, which will hide the Earth, but will prevent the geographic elements and lines to be seen when they’re on the back of the sphere.

Earth Style: changes current texture applied on the Sphere of the prefab.

Show Latitude/Longitude Lines: will activate/deactivate the layers of the grid. The stepping options allow you to specify the separation in degrees between lines (for longitude is the number of lines).

Grid Color: modifies the color of the material of the grid (latitude and longitude lines).

Show Cities: activate/deactivate the layer of cities.

- **Min Population (K)**: allows to filter cities from current catalog based on population (K = in thousands). When you move the slider to the right/left you will see the number of cities drawn below. Setting this to 0 (zero) will make all cities in the catalog visible.

- **Show Frontiers:** show/hide all frontiers. It applies to all countries, however you can colorize individual countries using the API.

- **Frontiers Color:** will change the color of the material used for all frontiers.

- **Country Highlight Enabled**: when activated, the countries will be highlighted when mouse hovers them. Current active country can be determined using *countryHighlighted* property.

- **Country Highlight Color**: fill color for the highlighted country. Color of the country will revert back to the colorized color if used.

- **Draw Outline** and **Outline Color**: draws a colored border around the colorized or highlighted country.

- **Show Country Names**: when enabled, country labels will be drawn and blended with the Earth map. This feature uses RenderTexture and has the following options:

    - **Labels Quality**: controls the size of the RenderTexture, thus affecting to the resolution of the labels shown in the map. Low quality uses a texture of 2048x1024, Medium 4096x2048 and High 8192x4096.
    - **Relative Size**: controls the amount of "fitness" for the labels. A high value will make labels grow to fill the country area.
    - **Minimum Size**: specifies the minimum size for all labels. This value should be let low, so smaller areas with many countries don't overlap.
    - **Labels and shadow color**: they affect the Font material color and alpha value used for both labels and shadows. If you need to change individual label, you can get a reference to the TextMesh component of each label with Country.labelGameObject field.

- **Show Cursor:** will display a cross centered on mouse cursor. Current location of cursor can be obtained with *cursorLocation* property when *mouseOver* property is true.

- **Navigation Time**: time in seconds for the fly to commands. Set it to zero to instant movements.

- **Autorotation Speed:** will make the Earth continuously rotate around its axis. Set it to zero to disable autorotation.

- **Allow User Rotation:** whether the user can rotate the Earth with the mouse. You can implement your own interactions setting this to false and modifying the rotation / position fields of the gameObject transform.

- **Allow User Zoom:** wether the user can zoom in/out the Earth with the mouse wheel.

- **Zoom Speed:** multiplying factor to the zoom in/out caused by the mouse Wheel (Allow User Zoom must be set to true for this setting to have any effect).

Choose Reset option from the gear icon to revert values to factory defaults.

Programming Guide (API)

You can instantiate the prefab “WorldMapGlobe“ or add it to the scene from the Editor. Once the prefab is in the scene, you can access the functionality from code through the static instance property:

```
WorldMapGlobe map;

void Start () {
map = WorldMapGlobe.instance;
        ...
}
```

(Note that you can have more WorldMapGlobe instances in the same scene. In this case, the instance property will returns the same object. To use the API on a specific instance, you can get the WorldMapGlobe component of the GameObject).

Public Properties

Country related

map.countries: return a List<Country> of all countries records.

map.countryHighlighted: returns the Country object for the country under the mouse cursor (or null).

map.countryHighlightedIndex: returns the index of country under the mouse cursor (or null if none).

map.countryRegionHIghlightedIndex: returns the index of the region of currently highlighted country.

map.countryLastClickedIndex: returns the index of last country clicked.

map.enableCountryHighlight: set it to true to allow countries to be highlighted when mouse pass over them.

map.fillColor: color for the highlight of countries.

map.showCountryNames: enables/disables country labeling on the map.

map.showOutline: draws a border around countries highlightes or colored.

map.outlineColor: color of the outline.

map.showFrontiers: show/hide country frontiers. Same than inspector property.

map.frontiersDetail: detail level for frontiers. Specify the frontiers catalog to be used.

map.frontiersColor: color for all frontiers.

Cities related

map.cities: return a List<City> of all cities records.

map.cityHighlighted: returns the city under the mouse cursor (or null if none).

map.showCities: show/hide all cities. Same than inspector property.

map.minPopulation: the mínimum population amount for a city to appear on the map (in thousands). Set to zero to show all cities in the current catalog. Range: 0 .. 17000.

map.cityCatalog: the currently city catalog used.

Earth related

map.showEarth: show/hide the planet Earth. Same than inspector property.

map.earthStyle: the currently texture used in the Earth.

map.autoRotationSpeed: the speed of the automatic/continuous rotation of the Earth.

map.showLatitudeLines: draw latitude lines.

map.latitudeStepping: separation in degrees between each latitude line.

map.showLongitudeLines: draw longitude lines.

map.longitudeStepping: number of longitude lines.

map.gridColor: color of latitude and longitude lines.

User interaction

map.mouseIsOver: returns true if mouse has entered the Earth's sphere collider.

map.navigationTime: time in seconds to fly to the destination (see FlyTo methods).

map.allowUserRotation/map.allowUserZoom: enables/disables user interaction with the map.

map.mouseWheelSensibility: multiplying factor for the zoom in/out functionality.

map.showCursor: enables the cursor over the map.

map.cursorFollowMouse: makes the cursor follow the map.

map.cursorLocation: current location of cursor in local coordinates (by default the sphere is size (1,1,1) so x/y/z can be in (-0.5,0.5) interval. Can be set and the cursor will move to that coordinate.

Labels related

map.countryLabelsSize: this is the relative size for labels. Controls how much the label can grow to fit the country area.

map.countryLabelsAbsoluteMinimumSize: minimum absolute size for all labels.

map.labelsQuality: specify the quality of the label rendering (Low, Medium, High).

map.showLabelsShadow: toggles label shadowing on/off.

map.countryLabelsColor: color for the country labels. Supports alpha.

map.countryLabelsShadowColor: color for the shadow of country labels. Also supports alpha.

Public Methods

map.GetCountryIndex(name): returns the index of the country in the collection.

map.FlyToCity(name): start navigation at *navigationTime* speed to specified city. The list of city names can be obtained through the cities property.

map.FlyToCity(index): same but specifying the city index in the cities list.

map.FlyToCountry(name): start navigation at *navigationTime* speed to specified country. The list of country names can be obtained through the cities property.

map.FlyToCountry(index): same but specifying the country index in the countries list.

map. FlyToLocation (x, y, z): same but specifying the location in local Unity spherical coordinates.

map.ToggleCountrySurface(name, visible, color): colorize one country with color provided or hide its surface (if visible = false).

map.ToggleCountrySurface(index, visible, color): same but passing the index of the country instead of the name.

map.ToggleCountryMainRegionSurface(index, visible, color): colorize and apply an optional texture to the main region of a country.

map.ToggleCountryRegionSurface(countryIndex, regionIndex, visible, color): same but only affects one single region of the country (not province/state but geographic region).

map.HideCountrySurface(countryIndex): un-colorize / hide specified country.

map.HideCountryRegionSurface(countryIndex): un-colorize / hide specified region of a country (not province/state but geographic region).

map.HideCountrySurfaces: un-colorize / hide all colorized countries (cancels ToggleCountrySurface).

map.ToggleContinentSurface(name, visible, color): colorize all countries belonging to speficied continent with color provided or hide its surface (if visible = false).

map.HideContinentSurface(name): un-colorize / hide specified continent.

map.Tilt & StraightenGlobe: returns the globe to default rotations.