COMPASS NAVIGATOR PRO

QUICK START GUIDE

Kronnect

Contents

Introduction

Thanks for purchasing!

Compass Navigator Pro is a GUI Scripting component for Unity useful in any kind of world exploration game. This is a compass bar type navigation helper, usually shown on top of the screen in many RPG games, showing destinations and point of interests (POIs) as well as text indications.

Compass Navigator Pro includes the following features:

- **Easy to use – drag & drop prefab** and you're set!
- Custom made customizable compass bar with **4 art styles (rounded, angled, celtic black and white**) with adjustable vertical position, width, alpha and fade in/out effects.
- **High resolution icons with customizable behaviour** (smooth scaling, visited vs non-visited icons, black and white variations).
- Add your own icons or **use the icons included in the asset: cave, city, dock, dungeon, forest, lighthouse, mine, monolith, palace, tower, generic**.
- Show **animated text under or on top of the compass bar when discovering new locations**.
- Show **optional title and other info for the centered POI in the bar.**
- **Can focus on one POI**, making its icon always visible in the compass bar.
- Can **show in-scene gizmos during playmode**, like active destination icon, which helps the player to get to the exact point.
- Can use **two icon variations per POI, to differentiate unexplored and explored locations**.
- **Custom Editor inspector**
- **Integrated Menu Items** to quickly create new POI game objects or attach a POI component to an existing game object.
- **Documented API** to exploit the asset functionality.

Quick Start

1. Once imported, Compass Navigator Pro can be added to your scene from the top menu **GameObject > UI > Compass Navigator Pro.**

   Alternatively, you can locate the CompassNavigatorPro prefab in Resources/Prefabs folder and add it to the hierarchy of your scene.

2. Select the new created Compass Navigator Pro game object and customize the behaviour and look & feel of the compass bar using the custom inspector.

HDRP Considerations

Important! In HDRP, please expand the CompassNavigatorPro in the hierarchy until you find the TopDownCamera. Set its Volume Mask setting to Nothing.

Inspector Settings

When you select the CompassNavigatorPro in the hierarchy, a custom inspector will be shown allowing you to customize it:

Please note that all of these parameters can also be controlled using C# (see API section).

Compass Bar Settings

In this section you can customize the general look & feel of the Compass Bar:

- **Camera**: the default main camera for compass orientation. The asset automatically picks the Main Camera but you can assign a different one.
- **Style**: choose a ready-to-use compass bar style. If you want to add your own sprite you can simply replace the existing sprites in the Resources/Sprites section. Read ”Adding your own sprites” section.
- **Vertical Position**: Specify a vertical position for the compass bar (0 = bottom, 1 = top).
- **Width of bar**: Specify a width (0..1 with respect to the screen width).
- **End Caps Width**: Specify a margin to limit the area where icons can be displayed (so they don’t overlap the end point art of the compass bar).
- **Alpha**: Set the transparency of the compass bar during playmode. Check “Visible in Edit Mode” if you want to show the compass bar irrespective of the alpha setting during Edit Mode.
- **Fade Duration**: Sepcify a fade duration in seconds. The compass bar appears/disappears smoothly and this parameter controls the duration of this effect.
- **Use 3D Distance**: by default, the distance to the POI is measured ignoring the Y-axis, like in a flat plane. Toggle this checkbox to use the distance in XYZ space instead.
- **World Mapping Mode**: determines the algorithm used to fit the POIs in front of the camera into the bar.
- **Same Altitude Difference**: this is a threshold in meters to determine if POI is above or below your position.
- **Show Cardinal Points**: if N/W/E/S will appear in the compass bar.
- **Show Distance (meters)**: if the current distance in meters to the POI is shown next to its name.
- **Idle Update Mode**: the contents of the compass bar and mini-map are always updated when the camera rotates or moves. But if the camera is idle, this property specifies the interval between POI changes check.

POIs Settings

This section allows you to customize the behaviour and look of the POIs (Points of Interests) or icons shown in the compass bar:

- **Visible distance**: POIs farther than this parameter won't be shown in the compass bar.
- **Near distance**: this is a distance threshold where the icons will begin to grow as the player approaches them.
- **Visited distance**: the distance to the POI to be considered visited or explored. The icon shown in the compass bar will be chosen according to the IsVisited property of the POI.
- **Icon Size Range**: useful to customize the minimum and maximum icon sizes in the compass bar. As the player approaches the POIs, the icons will tend to grow.
- **Label Hot Zone**: a POI's label will be visible on the Compass Bar if its icon on bar is within certain distance from the center. This distance is defined by Label Hot Zone parameter.
- **Scale In Duration**: when a POI icon appears on the compass bar, this setting controls the duration for the scaling animation. Set this to zero to make the icon pop on the compass bar without any scale effect.
- **Gizmo Scale**: this is a scaling multiplier for the icon displayed in the scene (if the POI is marked with ShowGizmoInPlayMode property).
- **Visited Sound**: an optional audio clip to be played when this POI is visited for the first time.
- **Beacon Sound**: an optional audio clip to be played when a light beacon is activated for this POI.
- **Heartbeat Sound**: an optional audio clip to be played when a light beacon is activated for this POI. Each POI can have its own heartbeat clip.

Title and Text Settings

This section controls the look and behaviour of the title and text.

The title is shown over the compass bar when a POI is centered in it and has been visited.
The text is shown in animated way when POI is first discovered.

Mini-Map Settings

This section controls the look and behaviour of the mini-map. The mini-map is synchronized with the compass bar. Adding a POI to the compass bar will make it visible in the mini-map as well.

- **Follow**: select the gameobject that's below the mini-map center. Usually it's the main camera or player character.

- **Camera Mode**: the mode for the mini-map camera, either orthographic or perspective. An orthographic camera will have the same visible range (defined by Zoom Range property) while the perspective camera's area depends on the altitude (defined by Altitude property).

- **Layer Mask**: specifies which objects should be visible in the mini-map.

- **Style**: the graphical style of the mini-map. To provide your own textures, choose "Custom".

- **Alpha**: the transparency of the mini-map.

- **Border Texture / Mask**: the art used for the background of the map. The mask texture is used to clip any POI according to the border texture.

- **Image Resolution**: the resolution for the image shown in the mini-map defined by 2^value. For example, a value of 8 produces a mini-map texture of 256x256.

- **Icon Size**: the size for the icons shown on the mini-map.

- **Clamp Border**: clamped POIs will always be shown on the mini-map even if they're out of range. The clamp border property defines the minimum distance to the edge of the mini-map rectangle.

- **Snapshot Frequency**: determines when the mini-map camera is active. For better performance, choose Time Interval or Distance Interval (it updates only when camera moves a certain distance to update the mini-map background)

When in full-screen mode, the properties **World Size** and **World Center** determines the bounds of the entire map to be shown.

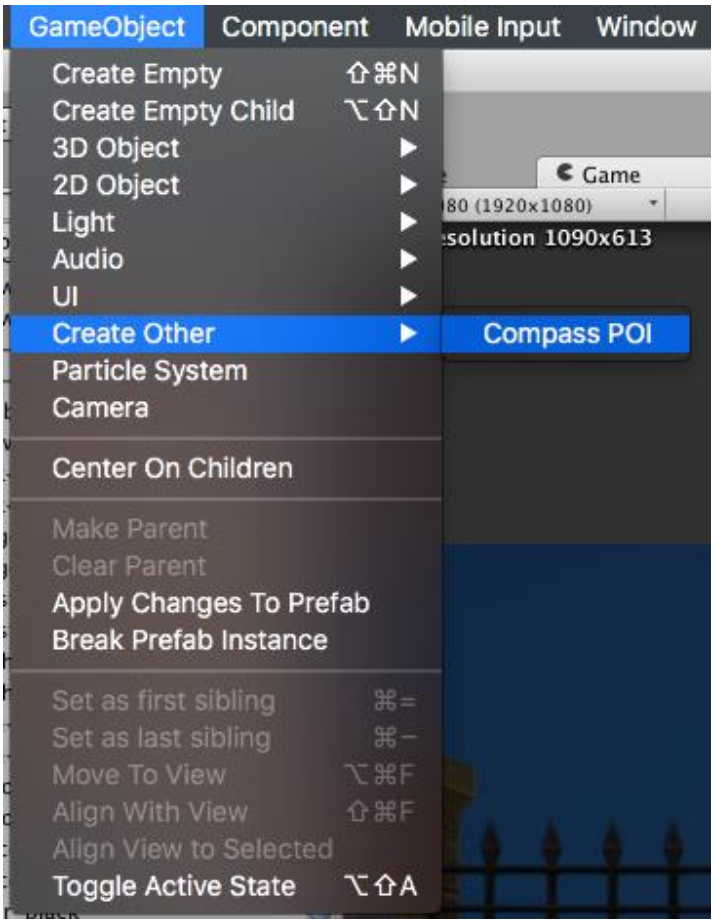
Adding POIs to the scene

A POI (Point of Interest), also known as destination or location, is just a CompassProPOI script added to any game object in the scene. You can create an empty game object and add this script to it, or attach the script to an existing game object in the scene.

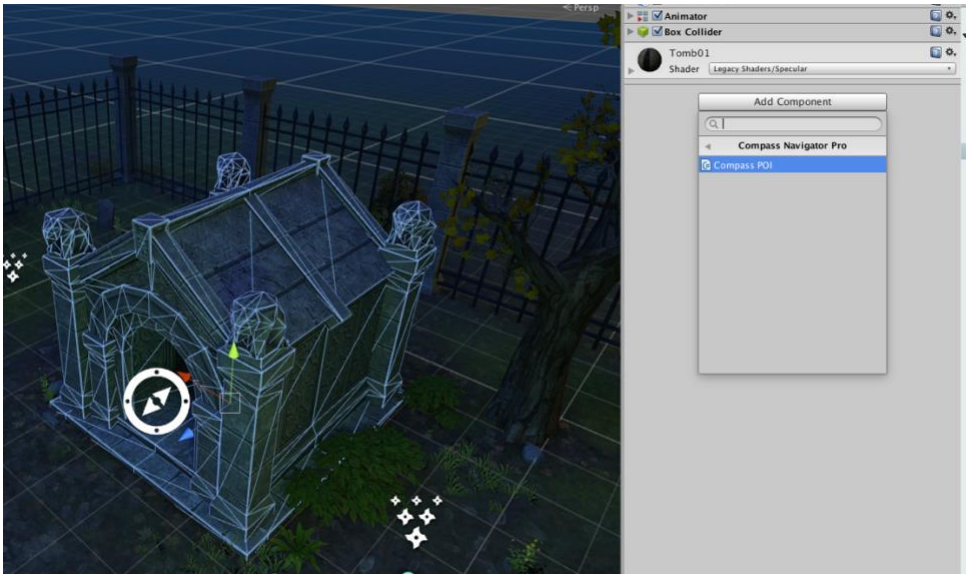
You can add any number of POIs to the scene, and you can do it using Unity Editor or scripting (C#).

Using Unity Editor to add POIs to the scene

To create a new game object as a POI, just select Game Object > Create Other > Compass POI:



You can also simply attach a Compass Pro POI script to an existing game object, and it will work as well:



Using C# to add POIs to the scene

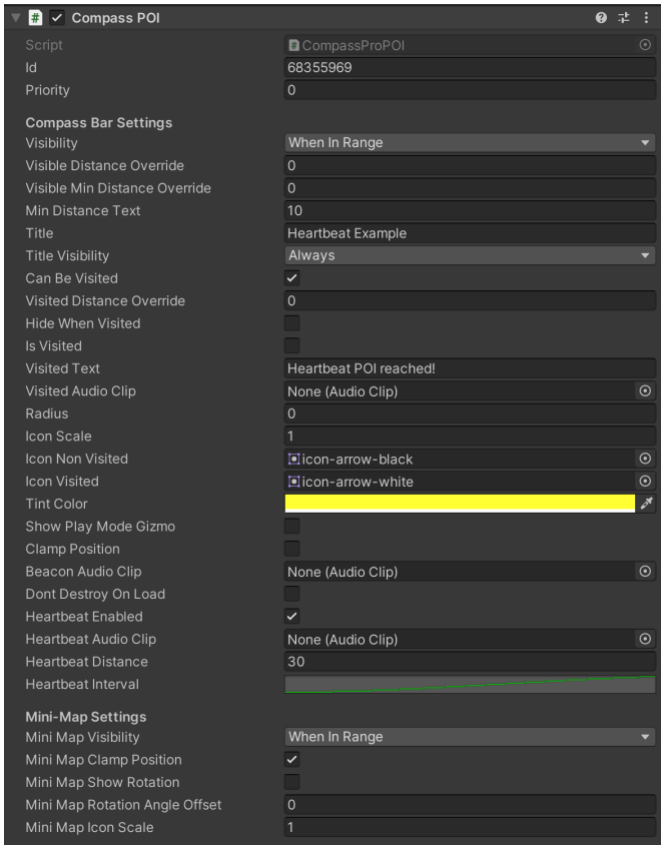
Simply attach the component CompassProPOI to any game object and populate it's properties.

```
using CompassNavigatorPro;
…
GameObject myGameObject = …
CompassProPOI poi = myGameObject.AddComponent<CompassProPOI>();
poi.title = "the title for this poi";
…
```

Get a look into CompassProPOI script for a list of available properties (same than shown in the Inspector).

Customizing POIs

Select the game object with the CompassProPOI script attached and you can customize its public fields from the inspector (you can also edit them in code):



- **Id**: automatically assigned number to each POI. Used internally.
- **Priority**: POIs with higher priority render their icons on top of others in the compass bar and mini-map.
- **Visibility**: use this property to determine when the POI icon should appear in the Compass Bar. By default, POIs appear when they're in "Visible Range" but you can specify a POI will always be visible (ie. active quest) or always hidden until you determine it can appear in the compass.
- **Visible Distance Override**: a value of 0 assumes the general visible distance. Optionally assign a custom visible distance.
- **IsVisible**: this property is automatically set by the asset based on Distance property and determines if the icon is visible in the compass bar.
- **Title**: this is the name of the POI to be shown over the compass bar if IsVisited is true.
- **HideWhenVisited**: when enabled, the POI will automatically be hidden in the compass bar when visited.
- **canBeVisited**: this property specifies if the POI can be marked as visited when reached. Defaults to true.

- **IsVisited**: this property is also automatically set to true by the asset as the player moves over the scene and determines if the POI is explored or not (based on the VisitedDistance property). You can also set this property to true to ignore the visited/non-visited feature.
- **Visited Text**: this is the text shown in animated way when the POI is visited for the first time.
- **Visited Audio Clip**: optional audio clip to be played when this POI is visited for the first time. Note that a global default audio clip can be also specified in the Compass Navigator Pro script.
- **Radius**: this is a radius for the POI. Useful for areas or cities where it really has no an exact center.
- **Icon Non Visited**: the icon to be shown in the compass bar when IsVisited = false.
- **Icon Visited**: the icon shown in the compass bar when IsVisited = true.
- **Show Play Mode Gizmo**: mark this toggle to show the visited icon in the scene during playmode to mark exactly the location of this POI in the scene.
- **Clamp Position**: forces this POI icon to stay visible in the bar, even if it's behind the player (then it will show on the edges of the bar).
- **Beacon Audio Clip**: custom sound for this POI when the light beacon is shown.
- **Don't Destroy On Load**: enabling this option will preserve this POI between scene changes (also its visited state). Note that the POI will only be visible in the scene where it was first created.
- **Heartbeat Enabled:** enables heartbeat sound when camera is approaching this POI. The heartbeat sound will play at a variable rate based on distance.
- **Heartbeat Audio Clip**: optional audio clip for the heartbeat sound. If none set, it will use the clip specified in the Compass Bar property.
- **Heartbeat Distance**: distance from which the heartbeat starts playing.
- **Heartbeat Interval**: a curve defining the heartbeat interval rate. The X-axis specifies the distance to the POI from 0 to 1 (1 = heartbeat distance). The Y-axis specifies the interval in seconds between heartbeats.

Mini-Map Settings per POI:

- **Visibility**: defines when this POI is visible in the mini-map.
- **IsVisible**: is set to true when the POI is visible in the mini-map.
- **Clamp** Position: when set to true, the POI will always be visible in the mini-map even if it's out of range.

Methods of Compass Pro POI component:

- **GetCompassIconScreenRect**: returns the screen rectangle of the icon in the compass bar (if visible).
- **GetMiniMapIconScreenRect**: returns the screen rectangle of the mini-map icon if visible.

Light Beacons

You can call the function POIShowBeacon to quickly visualize a given POI in front of the player. Or call POIShowBeacon without passing any POI and all visible POI in the compass bar will illuminate briefly. Example:

All beacons use the same material, located in Resources/Materials/Beacon. If you want to change the color or any other material properties. feel free to edit this material.

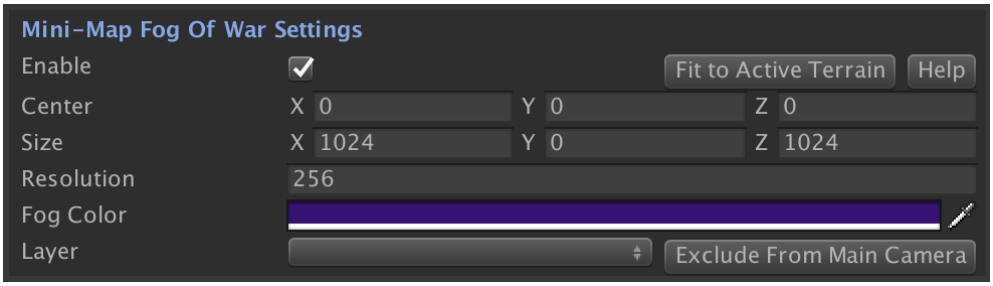
You can specify a sound effect when showing beacons in the CompassPro script or for each POI script (see previous sections).

Fog of War

Fog of war is a mini-map feature useful to hide unexplored areas to the player. Since the mini-map can show distant areas, you can use the Mini-Map volumes to control what can be visible on the mini-map.

In the screenshot below you can see a blue box (fog of war) hiding an area.

To enable the Fog of War feature just enable it under the Mini-Map section below:

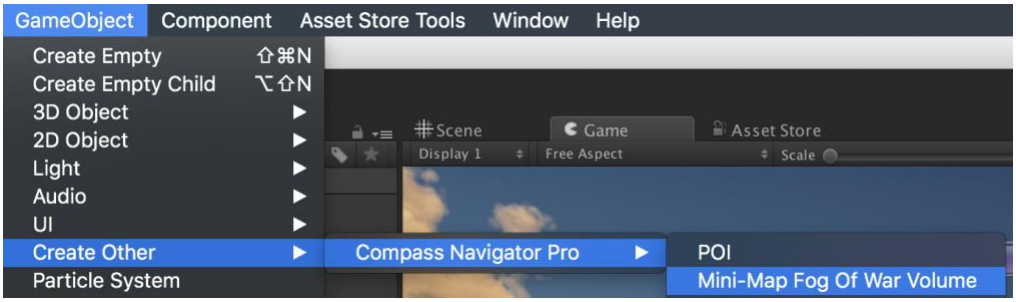


Parameters:

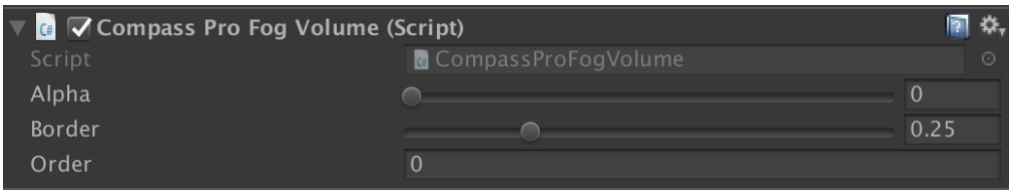
- **Center and Size**: represents the area of the world where the fog of war feature can be shown.
- **Resolution**: texture resolution for the fog of war effect.
- **Fog Color**: optional tint color for the fog.
- **Layer**: the fog of war layer is rendered as a big quad under the zenithal mini-map camera. In order to avoid it showing up in the main camera, make sure the Layer is culled from the Main Camera. You can also click “Exclude From Main Camera” to automatically exclude the layer of the fog of war object from the main camera culling mask.

How to add fog volumes:

You can add any number of fog areas to the scene by using the option:



It creates an empty gameobject with a Box Collider so you can resize it to match the desired area and a Compass Pro Fog Volume script attached with properties to customize the fog area appearance:



Use the transform position and scale to define the location and size of the fog area. And customize the fog area appearance using the script properties:

- **Alpha**: transparency of the fog (1=fully opaque fog).
- **Border**: width of the border for the fog of war volume.
- **Order**: fog volumes are rendered in the order defined by this property.

Changing the transform position or scale will trigger a fog of war update which can be expensive if you have many fog areas.

Another way to modify the fog of war at some position is to call the **SetFogOfWarAlpha** methods (please jump to API section for a full list of fog of war methods and examples on how to call Compass Pro methods).

API (using the compass bar with C#)

The asset includes some useful public methods and properties to customize via scripting.

First, you need to get a reference to the CompassPro script, using:

```
using CompassNavigatorPro;
…
CompassPro compass = CompassPro.instance;
```

Once you have a reference to the compass instance, you can access its properties and methods. Example: **compass.verticalPosition = 0.2f;**

Properties

- **cameraMain**: let you set the camera at runtime (useful if you change active camera).
- **degrees**: returns the current heading degrees of the compass.
- **style**: choose between the available compass bar graphic styles.
- **visibleDistance**: POIs beyond this distance won't be visible in the compass bar.
- **nearDistance:** distance threshold where the icons will start to grow.
- **visitedDistance**: distance at which the POI is considered visited. The radius of the POI is also used with this property.
- **gizmoScale**: scaling factor applied to in-scene icons.
- **alpha:** the transparency of the compass bar.
- **autoHide**: will hide the compass if no POIs are below visible distance. Compass will revert to visible when first POI gets nearer than visible distance.
- **fadeDuration**: duration of the fade in/out effect.
- **verticalPosition**: value (0 = bottom..1 = top) for the vertical position in the screen.
- **width**: value (0..1, 1 = screen width) for the width of the compass bar.
- **endsCapWidth**: size in pixels for the ending parts of the compass bar where you don't want to show icons. This property must be set for each compass bar style.
- **minIconSize** and**maxIconSize**: scaling factors for the icons in the compass bar.
- **textVerticalPosition**,**textScale**: controls the position and size of the text.
- **textRevealDuration**,**textRevealLetterDelay**,**textDuration**, **textFadeOutDuration**: controls the animation cycle for the text (reveal, duration of the text on the screen, and finally duration for the fade out).
- **titleVerticalPosition,titleScale**: controls the position and size of the POI's title shown over the center of the compass bar.
- **showDistance:** will show the distance in meters to the centered POI in the compass bar.
- **showCardinalPoints:** will show N, W, S, E in the compass bar.
- **use3Ddistance**: if enabled, the Y coordinate will be ignored when computing distance for icon scale. This property has no effect on the distance shown in the title (if showDistance is true).
- **sameAltitudeThreshold:** the difference in altitude between the POI and the main camera to show "Above" or "Below" as part of the title.
- **visitedDefaultAudioClip:** an optional audio clip to be played the first time a POI is visited. Note that you can specify a different audio clip in the POI script itself.

Mini-Map related properties

- **showMiniMap:** shows/hides the mini-map.
- **miniMapZoomLevel:** gets or sets the current mini-map zoom level (0-1) based on the min/max range defined in the Mini-Map properties.
- **miniMapZoomState**: gets or sets the current mini-map zoom state (false = normal size, true = full-screen size)
- **miniMapFullScreenSize**: the % of screen the mini-map fills when in full-screen mode. A value of 1 will make the mini-map fill the full screen.
- **miniMap*:** public properties shown in the inspector are also accessible through scripting.

Methods

- **Refresh:** forces a refresh of compass-bar and mini-map icons.

- **FadeIn(duration):** shows the compass bar with a smooth fade in effect.

- **FadeOut(duration):** hides the compass bar with a smooth fade out effect.

- **POIFocus(CompassProPOI poi):** makes a POI the principal POI. The icon will always be visible in the compass bar and a gizmo will be shown in the scene during playmode.

- **POIBlur():** cancels POIFocus effect.

- **POIShowBeacon(CompassProPOI poi, duration, horizontalScale, intensity, color):** activates a light beacon oriented to the sky which lights for a few seconds making easy to locate the distant POI in the scene from the player perspective.

- **POIShowBeacon(duration):** activates a light beacon for all non-visited POIs oriented to the sky which lights for a few seconds making easy to locate the distant POIs in the scene from the player perspective.

- **ShowAnimatedText(text):** triggers text appearing animation with a custom text.

- **MiniMapZoomIn** / **MiniMapZoomOut**: zooms in/out the minimap with optional speed parameter.

- **UpdateFogOfWar()**: resets and render the fog of war volumes in the mini-map.

- **ResetFogOfWar(alpha)**: fills entire scene with fog of war with given opacity.

- **SetFogOfWarAlpha** (Vector3 worldPosition, float radius, float fogNewAlpha, float border) Fills or clears an area around worldPosition. The radius is the distance in meters, fogNewAlpha specifies the value of transparency of the fog of war on this position (0-1) and border parameter specifies the smoothness of the edges (0-1).

- **SetFogOfWarAlpha** (Bounds bounds, float fogNewAlpha, float border)
  Same than previous method but updates within a given bounds.

- **SetFogOfWarAlpha** (List<Vector3> points, float fogNewAlpha, float border)
  Same than previous method but apply the fog of war along a path given by points. Can be used to simulate a route in the mini-map.

- **GetFogOfWar**(Vector3 position).
  Returns the transparency or level of fog of war at a given position.

- **fogOfWarTextureData**: gets or sets the contents of the internal fog of war texture. Can be used to persist fog of war state between sessions.

Events

- **OnPOIVisited:** triggered when a POI is visited the first time (the POI is passed as parameter).

- **OnPOIVisible:** triggered when a POI gets near than the visible distance (and appears in the compass bar).

- **OnPOIHide:** triggered when a POI gets farther than the visible distance (and disappears in the compass bar).

- **OnHeartbeat:** this event is triggered by the individual POI (it's defined in CompassProPOI class and not in CompassPro). Triggered when the POI plays a heartbeat sound.

- **OnPOIVisibleInMiniMap:** triggered when a POI appears in the mini-map.

- **OnPOIMiniMapIconMouseEnter:** triggered when mouse enters an icon in the mini-map (as it enters the icon rectangle).

- **OnPOIMiniMapIconMouseExit:** triggered when mouse exits an icon in the mini-map.

- **OnPOIMiniMapIconMouseDown:** triggered when mouse button is pressed on an icon in the mini-map.

- **OnPOIMiniMapIconMouseUp:** triggered when mouse button is released from an icon in the mini-map.

- **OnPOIMiniMapIconMouseClick:** triggered when mouse button clicks on an icon in the mini-map.

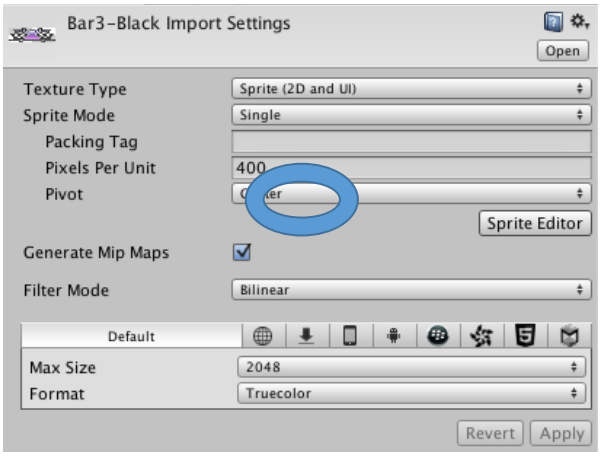
- **OnMiniMapChangeFullScreenState:** triggered when mini-map full screen mode is changed.

Adding your own art

You can add your own graphics for compass bar and icons.
Make sure you import them as Sprites and set the correct Pixels Per Unit setting in the import options.

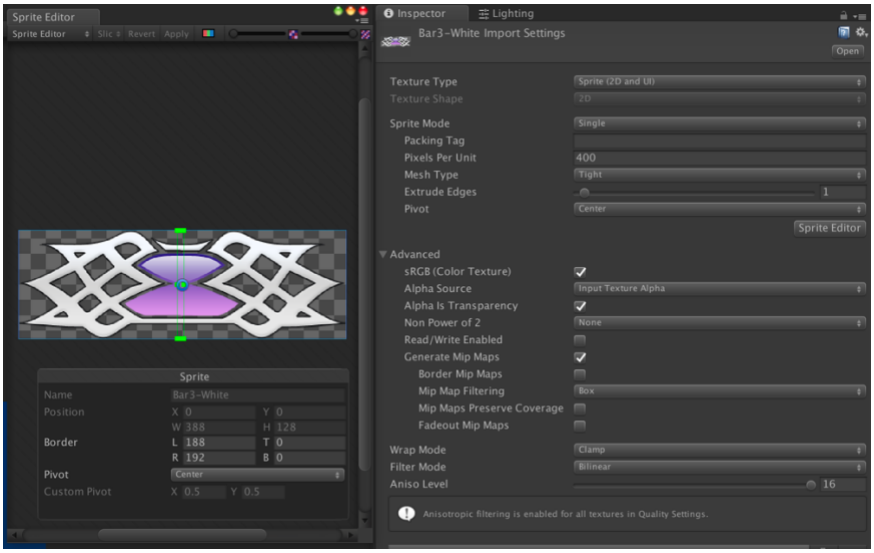
The Compass Bar is expected to be 32 pixels height, but you can add higher resolution sprites, so they will look better with different screen resolutions (HDPI or retina displays) for example.
For instance, if your compass bar sprite is 128 pixels height, then set Pixels Per Unit to 400 (as 128 is 4x32). Just divide the height of your sprite by 32 and multiply by 100.

This trick is used with Bar3-Black and Bar3-White sprites (the Celtic style bars):



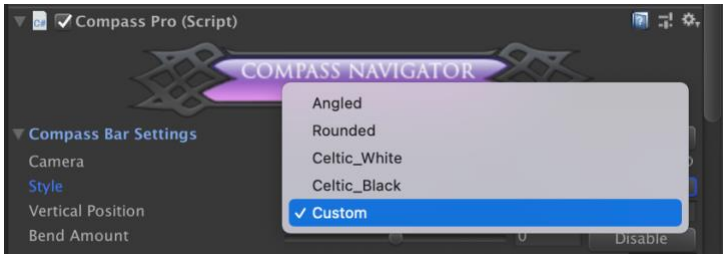
As per the icons, the asset expects 128x128 icons. If you supply bigger resolution icons, then adjust the Pixels Per Unit accordingly (for a 256x256 icon, set Pixel Per Unit to 200).

Don't forget to edit the compass bar sprite and select the middle area that will be enlarged to fill the horizontal area of the compass while keeping the ends cap undistorted:

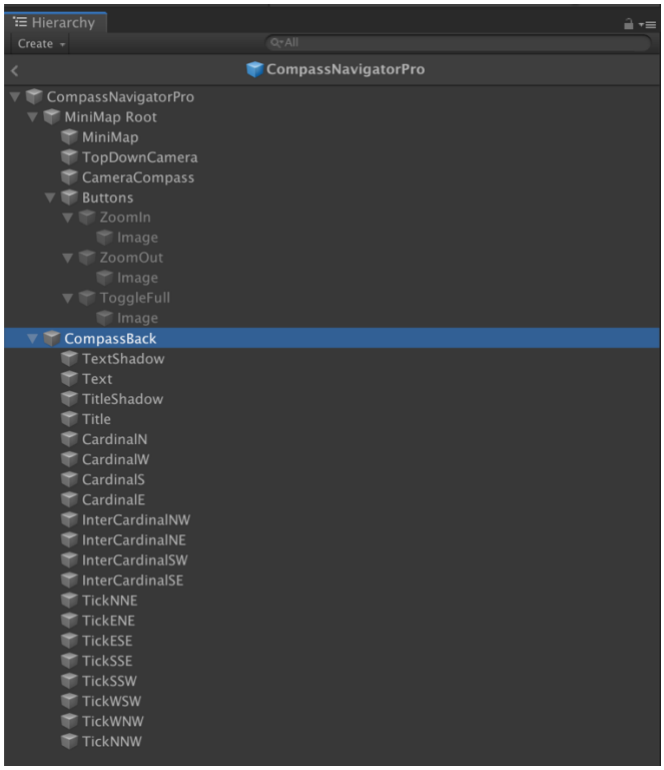


Replacing the Compass Bar elements

The asset includes 4 styles plus a “Custom” option. To use your own compass bar background sprite, make sure you select “Custom” as Style in the Compass Pro inspector. Then you can replace the sprite used in the “CompassBack” gameobject of the CompassNavigatorPro prefab or in your scene directly if you already added the compass to the scene:



Then you can edit the elements of the CompassNavigatorPro prefab and replace the sprites with your own (check previous section for hints about sprite resolutions):



For example, the “CompassBack” holds the compass bar background sprite. Feel free to replace that sprite with your own (as long as you have set “Custom” as the style in the inspector first). You can also customize the rest of the compass elements like CardinalN, CardinalW, CardinalS, interCardinalNW, TickNNE, etc.

Important: do not change the order or names of the different elements in the prefab. That can break the asset!

Frequent Asked Questions (FAQ)

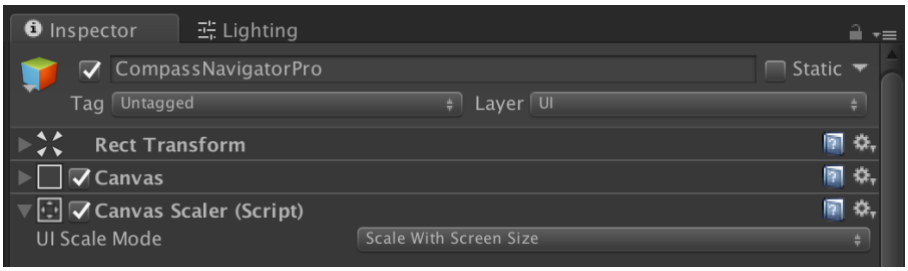
1) When I load another scene and come back to the original scene, the POIs state is not preserved, ie. their visited status is cleared. How can I preserve POIs between scene changes?

Just tick the DontDestroyOnLoad property of the POIs you want their state to be preserved between scene changes. Enabling this option will have the following effects:

- The POI gameobject won't be removed when the other scene is loaded.
- The POI won't be visible in the new scene though, but it will be activated again (visible in the compass bar) when you load the original scene back.

2) The compass shows too small on mobile devices. How to scale it up?

The compass asset is an Unity UI element so it respects the Canvas Scaler options. Try using "Constant Physical Size" or "Scale With Screen Size" to get the appropriate scale on any device.



Support

Please visit kronnect.com for questions, support and more info.

On kronnect.com you'll find latest beta updates for this asset and many others!