Location Tracker – Toolkit Guide

By Hyper Luminal Games LTD – www.hyperluminalgames.com

Getting Started:

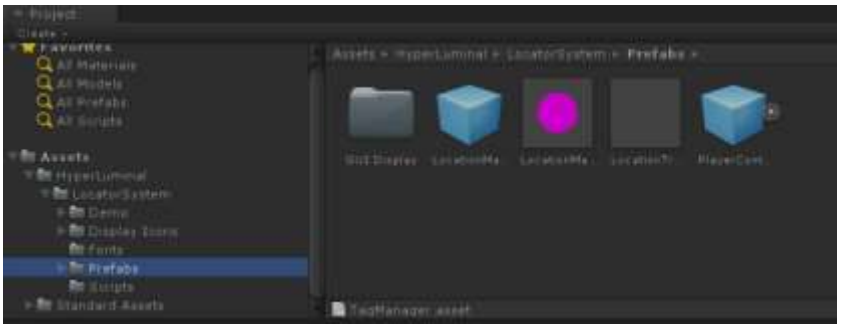
Setup Video Link: http://youtu.be/ViWtBQ2jsNY

Location Tracker by Hyper Luminal Games is a complete solution to marking, labelling and tracking locations (positions of key game objects, objectives, enemies, items etc.) within the Unity Engine. With minimal setup you can track hundreds of in-game locations instantly across single player or multiplayer split-screen cameras.

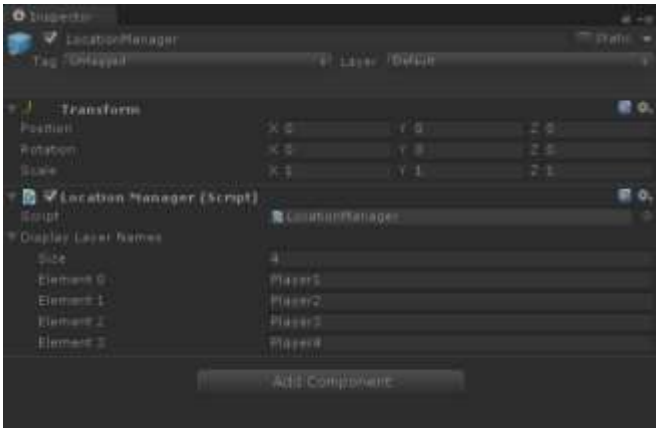
The setup for Location Tracker:

1. Download and import the "**Location Tracker Toolkit"** Asset Package from the Unity Asset Store, ensuring you have the **latest version**.

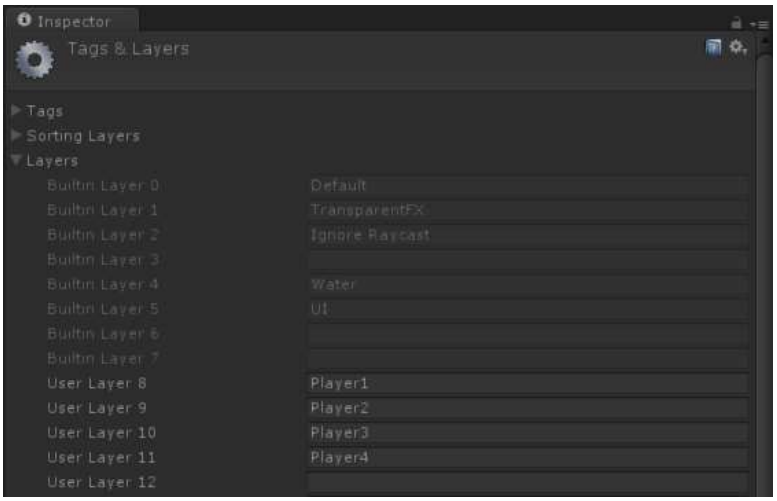
2. Once unpacked navigate to the "**Prefabs Folder"** within the assets folder structure.



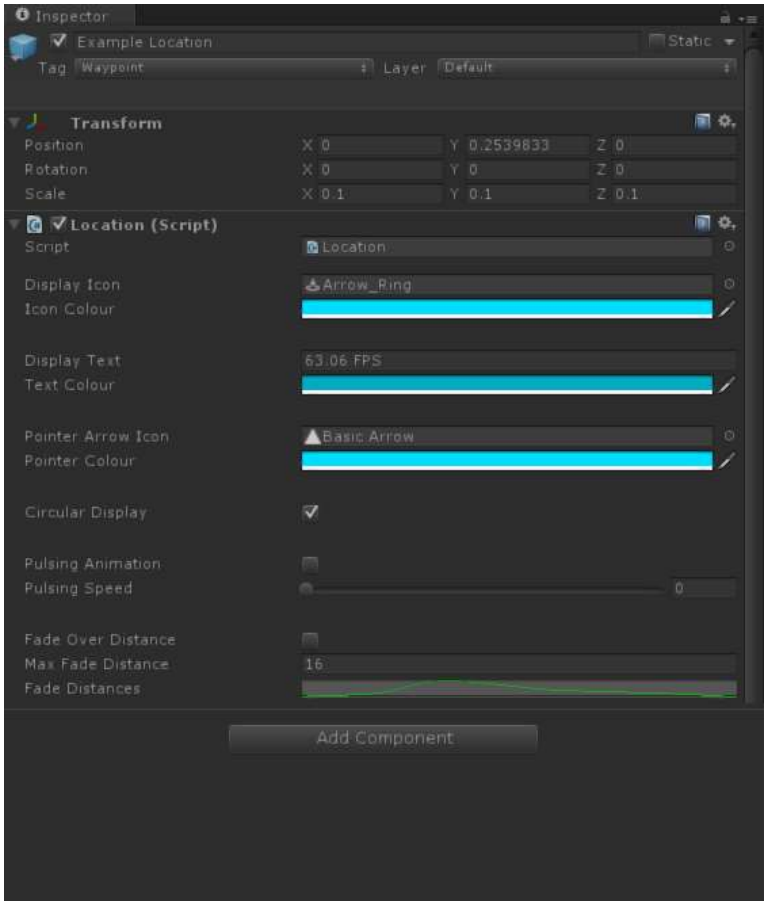
3. Drag the "**LocationManager.prefab**" Object into your scene. This object is responsible for keeping track of each of the games locations and should always be placed at position (0,0,0). On the location manager object there is a field called "**Display Layer Names**" please insure that there is one layer for each camera in the scene that you wish to track waypoints (1 player game = 1 layer, 4 player split screen = 4 layers).



4. Once you have set the layer names on the Location Manager Prefab, head to the "**Unity Layer Settings**" and add the same layer names as are on the Location Manager Prefab.



5. Now within the same folder drag the "**LocationMarker.prefab**" Object into your scene. This is a location that will be tracked by the system. Alternatively you can simply attach the "**Location.cs**" Script to any game object with your scene to begin tracking it. (Your game object will require either a "**Renderer**" or "**Collider**" component, any variation of these will suffice).

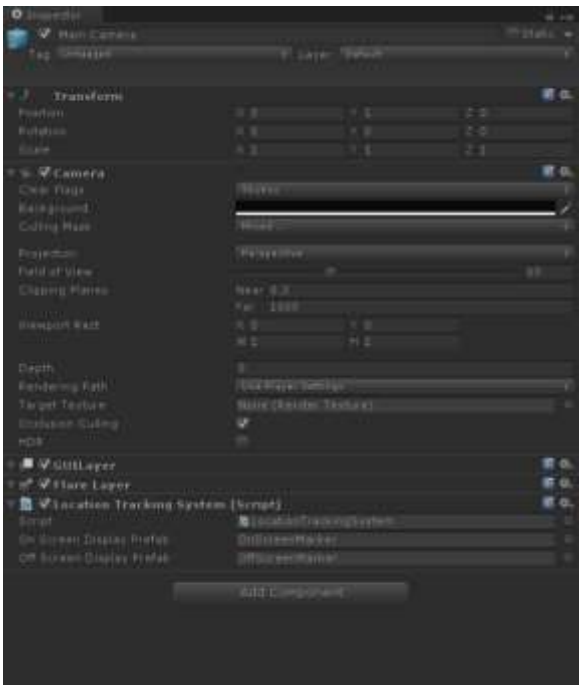


6. The next step is project dependent, either select the provide Unity Asset First Person controller or setup the Tracking System on your own camera:

    a. Drag the "**PlayerController.prefab**" Object into your scene. This is a first person camera example using the asset. Notice this game object has a camera component attached which includes a "**LocationTrackingSystem.cs**" Script, this script handles the display of the tracked locations and is the final component for this system.

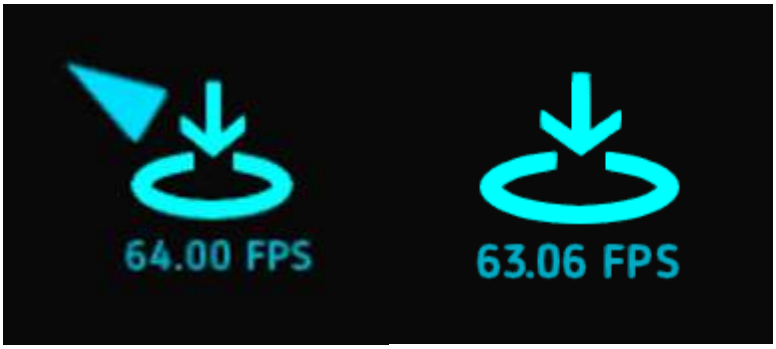
OR

    b. Attach "**LocationTrackingSystem.cs**" Script to your current in game camera. Ensure its inspector looks like:



    c. If it doesn't then hook up the game objects from the folder: **"Prefabs->GUI Display".**

7. Run the game in the editor and you should see an indicator pointing towards the location marker we placed in game.



Location Marker Customisation:

Location Markers are highly customisable and can be fine-tuned to suit your needs. Small tooltips can be read in game by hovering over the component field in the inspector.

Below is a brief overview of each of the customisation options:

Display Icon - Set your display icon from the Indicators folder or create your own, use the provided Photoshop files (.psd) as a reference if required.

Icon Colour – Sets an overlay colour for the Display Icon, making your icon's white as per the provided examples, enables in engine colour customisation.

Display Text - Set the display text to whatever you want to say or leave it blank to hide the text display.

Text Colour - Sets an overlay colour for the Display Text, making your icon's white as per the provided examples, enables in engine colour customisation.

Pointer Arrow Icon - Set the Arrow Display Icon from the Pointer Arrows folder or create your own, use the provided Photoshop files (.psd) as a reference if required.

Pointer Colour - Sets an overlay colour for the Arrow Icon, making your icon's white as per the provided examples, enables in engine colour customisation.

Circular Display - Circular display arranged the indicator in a compass like circle in the center of the screen. When false the display will default to snapping the indicators to the relevant edge of the screen instead.

Pulsing Animation – Pulsing will scale the indicator icon up and down on the screen, this is useful for indicating a warning or time limited location.

Pulsing Speed - Pulsing speed determines the rate that the icon scales.

Fade Over Distance - Fades the Indicator over distance using the Fade Distances graph and by referencing the maximum fade distance variable.

Max Fade Distance – This is the physical unity distance that the fading will reach the end of the fading graph (where the graphs X Axis = 1).

Fade Distances – Fade Distances is a simple graph option that allows smooth fading, where the Y Axis = Alpha of Inidcator and the X Axis = Distance from the target Location. Ensure you add lots of key frames (around 20) for a nice smooth fading transition.

Location Scripting:

Interfacing with Location Markers is very easy, since everything is setup via the Location script attached to your in game object. To interface with the Location and its Marker simply use the following method:

GameObject.Find("**Example Location**").GetComponent<Location>().**PARAM**

The red text "**Example Location**" in the above example is the name of your location game object. Please note that this method uses the GameObject.Find() functionality which is not ideal, for better performance hook up a reference to your location object in the inspector and simply use the GetComponent<>() Functionality on it. For regularly updating values it is best to cache both the game object and the Location component.

The red text "**PARAM"** in the above example is the parameter you wish to customise. This parameter can be any of the customisation parameters details in this documents "Location Marker Customisation" section.

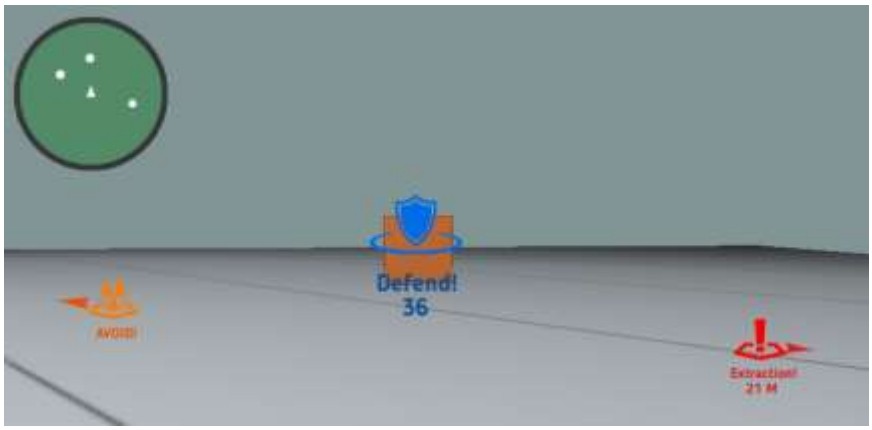
Remember to include the "**Using HyperLuminalGames;**" Namespace at the top of your script file.

An example script provided with the project, this can be used to turn Location Markers on and off during game run-time.

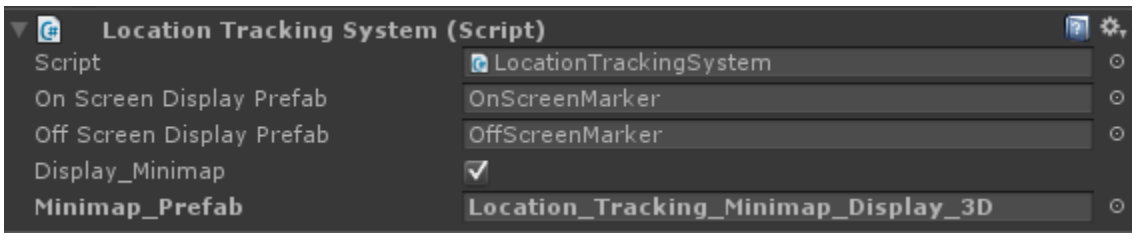
```
using UnityEngine;
using System.Collections;
using HyperLuminalGames;

public class ToggleLocationMarker : MonoBehaviour
{
    private float time = 0.0f;
    public float ToggleTimer = 2.5f;
    public Location LocationScript;

    // Update is called once per frame.
    void Update ()
    {
        // track the amount of time it has been in one state.
        time += Time.deltaTime;

        // change state after a certain time period.
        if(time > ToggleTimer)
        {
            // toggle the location marker on/off.
            LocationScript.enabled = !LocationScript.enabled;

            // reset the timer.
            time = 0.0f;
        }
    }
}
```

Mini-map / Radar Location Marker Display:

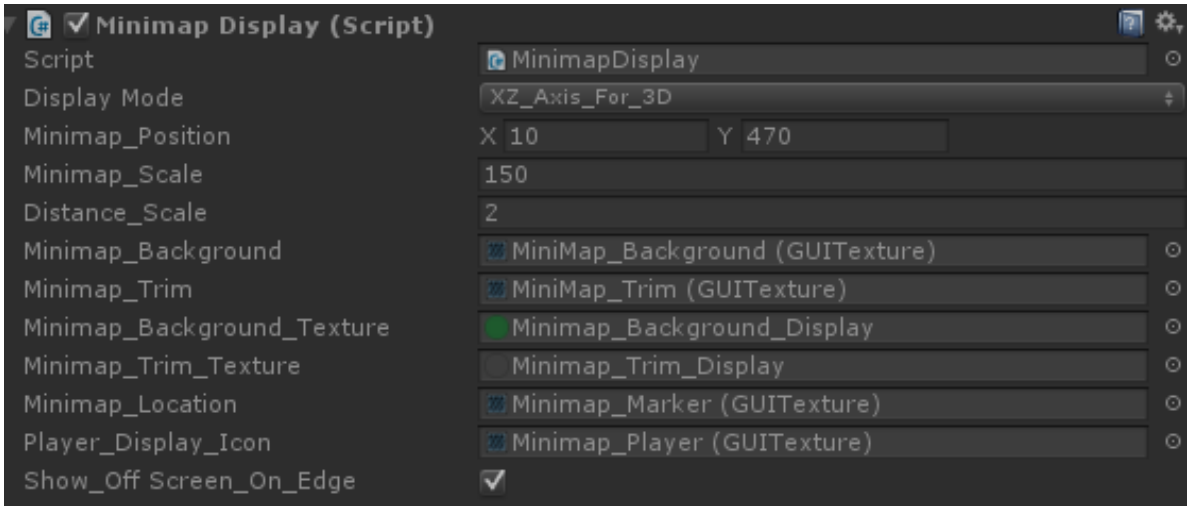
Version 1.2 has brought a brand new feature to the Location Tracking UI Toolkit! A dynamic one click Mini-map display that supports 2D, 3D and Split-screen cameras.



The mini-map will track all in game locations converting their 3D position to a 2D display. The white arrow in the center represents the player and the arrow points in the direction the player is currently facing. The Mini-map setup is literally one click on your Location Tracking System script. Ensure you tick the "**Display_Minimap**" Option and have a relevant mini-map prefab selected in the "**Minimap_Prefab**" slot as shown below:



There are three pre-made prefabs packaged with the release of version 1.2 – 2D, 3D and Split-screen mini-maps. There prefabs can be used straight away for their relevant display type. The Mini-map is however highly customisable just like the Location Markers. Below outlines the customisation options for the Mini-map.



Mini-map Display Customisation:

Display Mode – this should generally be set to the type of game world you are currently using (2D or 3D). The exception to this is what axis you wish distance to be measured on. For example if you 2D world uses the X and Z axis rather than the X and Y axis then you should choose the 3D option instead for accurate measurement.

Minimap_Position – this is the screen position for your mini-map, this will allow you to position the mini-map anywhere on screen. It can also be updated during gameplay for animated mini-map effects.

Minimap_Scale – this is the screen size for your mini-map, this will allow you to scale the mini-map to fit your game style. As the mini-map is currently a circle only one single radius size is used. It can also be updated during gameplay for animated mini-map effects.

Distance_Scale – this value will determine how close or far your locations appear on the mini-map. This value is completely game dependant and you will be required to change it to suit your game style. Generally a value between 0.5f and 5.0f is normal.

Minimap_Background – this is the background texture object for the mini-map you should not need to change this.

Minimap_Trim – this is the trim texture object for the mini-map you should not need to change this.

Minimap_Background_Texture – this is the background texture for the mini-map you can change this to any texture suitable for your game style.

Minimap_Trim_Texture – this is the trim texture for the mini-map you can change this to any texture suitable for your game style.

Minimap_Location – this is the location marker object for the mini-map you can change this to any texture suitable for your game style.

Player_Display_Icon – this is the player object for the mini-map you can change this to any texture suitable for your game style.

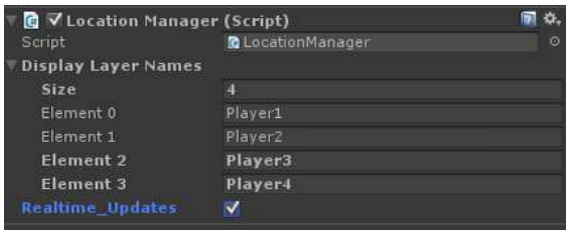
Show_Off_Screen_On_Edge – this customisation option determines if the location markers sit at the edge of the screen or disappear when they are out of the maps display range.

WARNING! Your Location Script now has a new variable to be able to use the mini-map – "**Minimap_Display_Icon**" this icon must be set to enable customised visuals on the Mini-map display.

V1.2 Features and Changes:

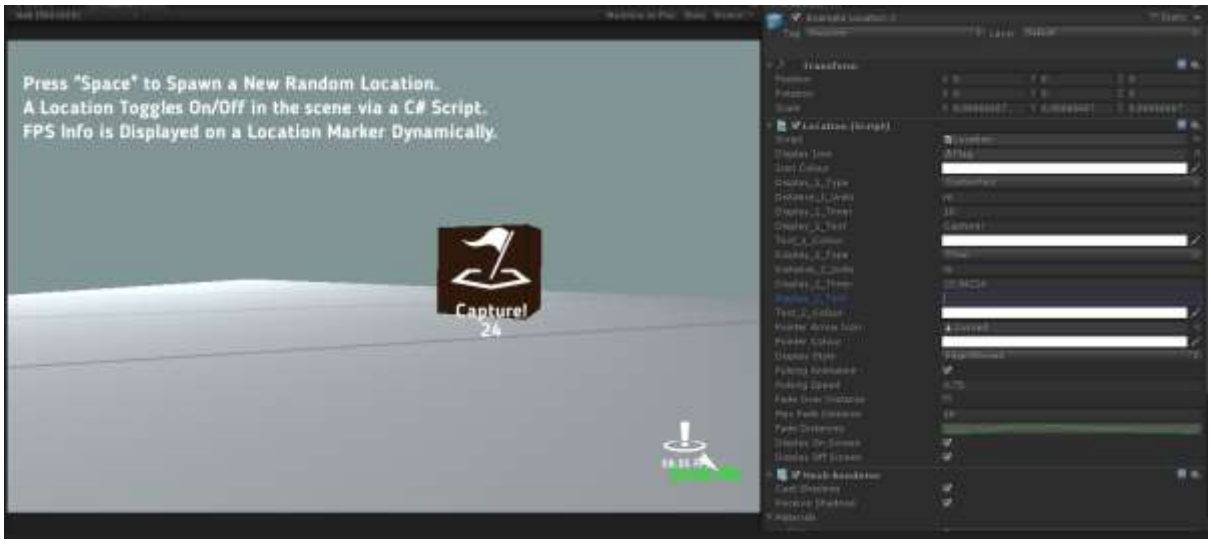
Version 1.2 brings a host of code optimisations and additional flexibility for display options as well as the brand new mini-map/radar display feature for Locations. Additional display icons have been added to the pack and a new set of 6 pre-made Location prefabs have been included utilising all of the latest customisable content for instant use in your games.

The addition of the new toggle button “Real-Time Updates” that can be found on your scenes “Location Manager” object as seen below:



This option should be “unchecked” when your scene has a fixed maximum number of locations to be tracked. It will greatly improve performance for location tracking but will remove the ability to spawn in additional locations during gameplay. It is also possible to toggle the “Realtime_Updates” feature during gameplay to enable the collection and display of additional Locations when they are spawned.

A new display option has also been added to the Toolkit. This allows two text displays to be included with each Location. This allows the features from version 1.1 to be used together. For example “custom text” + “Distance” or “Timer” + “Distance” displays as shown below.



Mini-map / Radar:

For full details regarding the mini-map/radar display for Location tracking please read the section named Mini-map in this document.

V1.1 Features and Changes:

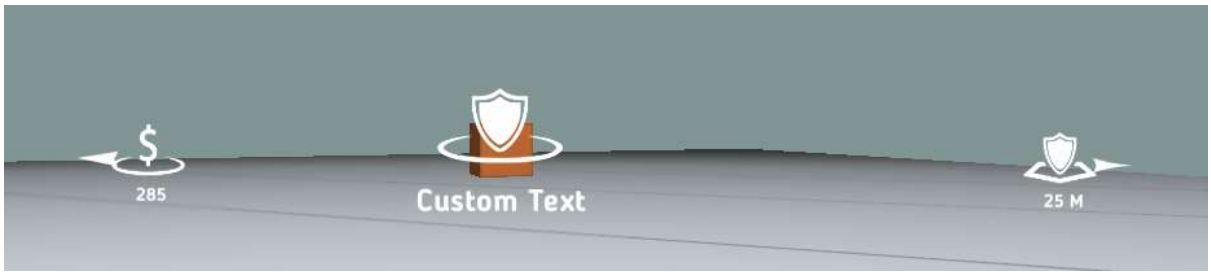
Version 1.1 brings **Unity 4.3+** Support enabling the assets use in even more games! The only missing features are Inspector Formatting.

Added a new display option “Mini_Compass”. This displays the locations in the bottom left hand corner of the screen. This is best for screen space limited games with few Locations.

A distance measurement between the camera and the Location has been added. The distance is measured in Unity Units and can have a customised units suffix added to the display such as “M” for meters. The distance can be displayed on screen instead of a text label.

Timer functionality which counts down from a set value and then disables the Location display once the time expires has also been added, this timer can be displayed on screen instead of a text label.

An example of the now three different display styles is shown below (Custom Text, Timer and Distance Measurement):



The last new feature for version 1.1 is the ability to toggle On/Off screen displays to allow either On, Off or Both Indicators to be displayed for each Location. This setting can be found on the “Location Tracking System” object. This is useful for hiding the Location markers if a Location is now perfectly visible instead of overlapping it with a UI display as shown in the image above.

FAQ:

Why can't I see my Location Marker?

- Make sure the "Display Icon" field has a texture
- Make sure the "Display Icon Colour" Alpha setting is not 0.
- Disable the "Fade over Distance" setting. (If this fixes it then adjust your Maximum Distance Setting and Alpha Graph until it is visible.)
- Ensure no errors are being displayed in the console log. If they are follow their instruction to correct your issue.

Can I use this with NGUI or 2DToolkit?

- This asset was only designed for use with the Unity UI system and currently does not support any other UI rendering plugin. Although it may be possible to add this there is currently no support for its development.

Can this asset be used in both 2D and 3D games?

- Yes, Locations can be marked in any setting where a Unity camera is being used.

Will this asset be updated to include new features in the future?

- Yes, this asset has already been updated with a host of features. This asset has the potential to grow into a much larger product. Depending on feature demand, different areas of the asset will be targeted for expansion.