

# Unity Gimbal Plugin

The Unity Gimbal Plugin makes it easy to include Gimbal events in your Unity projects for iOS and Android.

## Requirements

- Unity 5.3 (untested on previous versions)
- Android Version 4.4.3 or higher
- iOS
  - Using Xcode 7.3 or higher
  - Targeting iOS 8.0 or higher
  - Using iOS device with Bluetooth 4.0

## Getting Started

1. Create a new Unity Project (or open an existing project)
2. Within Unity, switch to iOS or Android platform
3. Open and import UnityGimbal.unityPackage into your Unity project
4. Open the example scene /gimbalsdk/Gimbal/Example.unity
5. Set the example scene as the launch scene (build settings)
6. Select the GimbalPlugin object
7. If you haven't already, sign up for a developer account with Gimbal and register your beacons and your add bundle id. See: [Gimbal Developers](#)
8. Paste your iOS and/or Android Api Keys from the Gimbal dashboard into the corresponding fields on the GimbalPlugin behavior
9. Ensure the bundle id of your project matches the bundle id set in the Gimbal dashboard for the Api Key
10. Optionally set Beacon Manager and/or Place Manager to start automatically when the scene starts
11. Build and run the project **on a device** (the beacons will not work in the simulator)
12. Toggle the managers on if needed
13. Beacon and place info will be logged in the UI of the example scene

## Adding Gimbal Event to Your Own Classes

The GimbalBehavior class publishes the following events:

```
public event BeaconSightingHandler BeaconSighted
public event BeginVisitHandler BeginVisit
public event EndVisitHandler EndVisit
```

You can easily add and remove event handlers using standard C# techniques. There are two easy ways to get a reference to the GimbalBehavior class:

- create a public property and then drag the GimbalPlugin object onto it
- use GetComponent< GimbalBehavior >()

```
public GimbalBehavior gimbalBehavior;

void Start() {
    gimbalBehavior.BeaconSighted += new
    GimbalBehavior.BeaconSightingHandler(BeaconSightingFound);
}

...

private void BeaconSightingFound(BeaconSighting sighting) {
    // do things in response to beacon sightings...
}
```

## Working with the Gimbal Models

Each of the following models mirrors the fields provided by the native Gimbal SDK.

- BeaconSighting
- Beacon
- Visit
- Place

You can read more about the Gimbal SDK on the [Gimbal Developer Site](#).