iBeacon Unity plugin Documentation

iBeacon

Functionality

The iBeacon acts as the server component for an iBeacon network. An app running the server will advertise an iBeacon UUID, region, major, minor and range to iBeacon receivers in an area.

How to use

Attach the IBeacon script to a gameobject in your scene or drag the “IBeaconServer” Prefab into the scene. You will find it under “Resources/Prefabs/”. The IBeacon script exposes 5 variables which you can configure according to your needs.

- Generate: clicking this will generate a valid UUID in the UUID field
- UUID: you can enter any valid UUID here or generate a random one by clicking on Generate
- Identifier: this is the region you will broadcast, e.g. com.example.myregion
- Major: the major number could acts as a hint for the location of the beacon. A company with more than one structure may for example identify each building by a different Major
- Minor: the minor number will act as a further hint where exactly you are. Said company may for example identify each room inside said buildings by a different Minor

An example would be a company that has 2 branches, one in New York and one in Chicago. The storage room in the second building in Chicago may look like this:

UUID: 9346e6ca-93c4-495d-81d5-efa64a005a8d
Identifier: com.company.chicago
Major: 2
Minor: 3

While the storage room in the first building in New York could be:

UUID: 9346e6ca-93c4-495d-81d5-efa64a005a8d
Identifier: com.company.newyork
Major: 1
Minor: 3

I hope you get the idea, you can edit these values to your needs, just make sure the receiver app you build recognizes them.

IBeacon exposes the following methods:

public static void Init() -> Inits the plugin, call this before anything else
public static void Transmit() -> Starts advertising
public static void StopTransmit() -> Stops advertising

IBeaconReceiver

Functionality

The IBeaconReceiver acts as a receiver for the IBeacon you lernt to set up above.

How to use

Create a gameobject in your scene, call it "IBeaconReceiver" and attach the IBeaconReceiver script. You can also drag the IBeaconReceiver Prefab into your scene that you will find under "Resources/Prefabs/". The IBeaconReceiver exposes two variables:

- UUID: here you can enter the UUID that your iBeacon hardware is configured to. If you set up your hardware with an app and the above iBeaconServer example just copy the string from the UUID field of the IBeacon
- Region: here you can enter the region that your iBeacon hardware is configured to. If you set up your hardware with an app and the above iBeaconServer example just copy the string from the Identifier field of the IBeacon

Initilaisation:

Just call IBeaconReceiver.Init() to initialise the plugin. IBeaconReceiver fires an event with a List of Beacons that you can attach to. For an example have a look into the IBeaconExampleScene.

IBeaconReceiver exposes the following methods:

public static void Init -> Inits the plugin, call this before everything else
public static void Stop -> Stops scanning for beacons
public static void Scan -> Starts Scanning for beacons, you don't have to call this if you didn't stop the scan before, init will call this for you

IBeaconReceiver fires the following events:

BeaconRangeChangedEvent -> delivers a list of all beacons in range
BeaconArrivedEvent -> fires when a new beacon is in range
BeaconOutOfRangeEvent -> fires when a beacon leaves the range

Android Usage

The android usage is essentially the same as the IOs one except that you will need to adjust your AndroidManifest.xml to have access to Bluetooth. An ExampleManifest.xml comes with this plugin and is located under Assets > Plugins > Android. If you don't have any other Android plugins that you use you can safely rename it to AndroidManifest.xml, if you do just copy these 2 lines to your own AndroidManifest.xml

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```