
iBeacon

3.2 版本文档

简介

系统要求	3
------------	---

使用

设置	4
iOS 上的位置使用说明	4
安卓 6.0 或更高版本的权限	5
定义区域	5
Beacon 类	7
Telemetry 类	8
使用 BluetoothState	8
iBeaconReceiver 检查器	9
使用 iBeaconReceiver	9
iBeaconServer 检查器	10
使用 iBeaconServer	10

从旧版本升级

恢复旧区域值	11
代码更改	13

常见问题解答

您的插件是否支持公司的信标[在此填入名称]?	14
在哪里可以获得信标的 Region Name?	14
iOS 无法找到信标。为什么?	14
安卓无法找到我的信标。为什么?	14
为什么来自只有一个地区的信标出现，虽然附近也有其他信标?	15
我可以在背景中对信标做出回应吗?	15

简介

感谢您使用我们的 iBeacon 插件。

- ▶ 任何需知的重要信息将像这样做出标记。

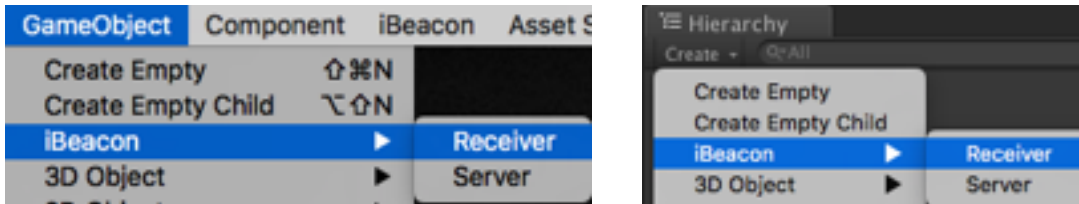
系统要求

- Unity 5
- iOS 8.0 或更高版本
 - iPhone 4s 或更新的设备
 - iPad（三代）或更新的设备
 - iPad mini 或更新的设备
 - iPod touch（5代）或更新的设备
- 安卓 4.3 或更高版本
 - ▶ 要在安卓中传输信标，您需要安装安卓 5.0 或更高版本及兼容的设备。您可以在这里查看部分设备列表 (<http://altbeacon.github.io/android-beacon-library/beacon-transmitter-devices.html>)，看您的设备是否支持。

使用

设置

如果要检测信标，请创建一个新的 *iBeaconReceiver* GameObject。



如果要传输信标，请创建一个新的 *iBeaconServer* GameObject。



在两种情形下，还将添加 *BluetoothState* 组件。

iOS 上的位置使用说明

在 iOS 上，需要用户的位置授权。当插件开始检测 iBeacons 时，用户将看到位置授权提醒。该提醒包含一个文本字段，说明应用程序要求使用用户位置的原因。

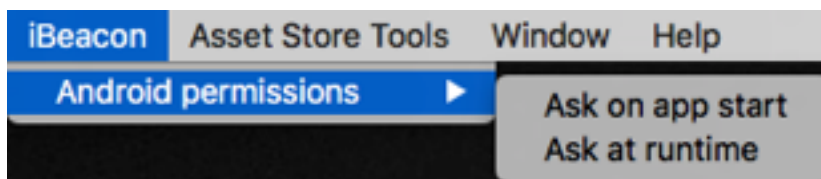
您必须在 Unity 的 iOS 播放器设置 *Location Usage Description* 中详细回应该文字。文字也会显示在设置应用程序的 *Privacy > Location Services* 下，用户可以随时允许或拒绝应用程序访问 iBeacon。

安卓 6.0 或更高版本的权限

从安卓6.0（API 级别 23）开始，应用程序必须持有 `ACCESS_COARSE_LOCATION` 或 `ACCESS_FINE_LOCATION` 权限，以扫描低功耗蓝牙设备，这些是危险的权限。此外，应用程序必须在应用程序运行时请求其需要的每个危险权限。

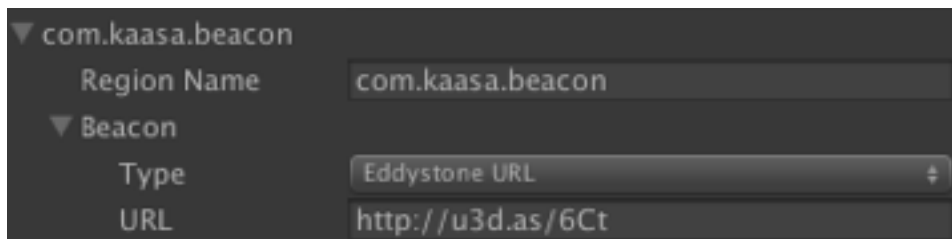
默认情况下，Unity 会在应用启动时请求清单中列出的所有危险权限。但如果安卓插件专为安卓 6.0 设计，则可以禁用此行为。

我们的 iBeacon 插件是为安卓 6.0 设计的。您可以通过 *iBeacon* 菜单中的 *Android permissions* 项设置这些行为。默认为 *Ask on app start*。



- ▶ 如果您有其他不是为安卓 6.0 开发的安卓插件，并且已设置 *Ask at runtime*，应用程序可能会崩溃或出现故障。因此，请在运行安卓 6.0 或更高版本的设备上测试您的应用。

定义区域



在检查器中的 *iBeaconRegion* 示例。

- *Region Name* 必须是区分区域的唯一标识符。它可以是任意的。此值不能为空。
- *Beacon* 是被关联信标的定义。有四种类型的信标：Any、iBeacon、Eddystone UID、Eddystone URL 和 Eddystone EID。

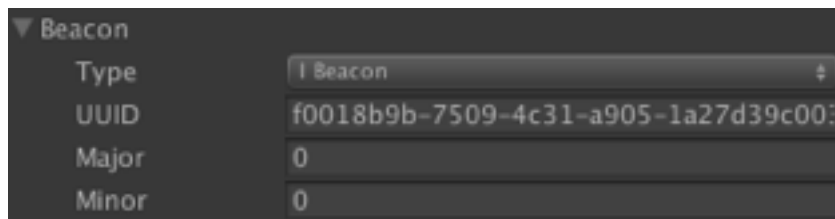
- *Any*



代表每个信标。

- ▶ 在 iOS 设备上它仅代表每个 Eddystone 信标。要检测 iBeacons, 它必须属于 *iBeacon* 的类型。

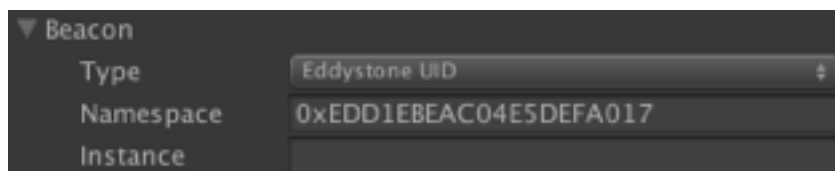
- *iBeacon*



代表 iBeacons 的子集。它有三个值：

- *UUID* 是目标信标的唯一 ID。此数值不能为空。
 - *Major* 是用于识别一个或多个信标的主要值。如果要忽略，请将此值设置为 0。
 - *Minor* 是用于识别特定信标的次要值。如果要忽略，请将此值设置为 0。
- ▶ *Minor* 将被忽略，如果 *Major* 是 0。

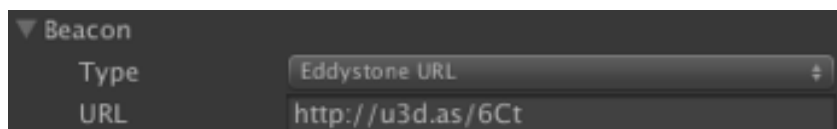
- *Eddystone UID*



这是 Eddystone-UID 数据包的子集的代表。它有两个值：

- *Namespace* 是一个信标 ID 命名空间。此值不能为空。
- *Instance* 是命名空间中的唯一 ID。如果你想忽略它，请留空。

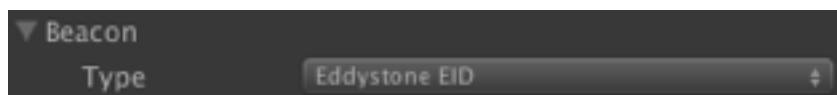
- *Eddystone URL*



这是特定 Eddystone-URL 包的表示。它有一个值：

- URL 是广播的 URL。此值不能为空。

- *Eddystone EID*



这是每个 Eddystone-EID 信标的表示。

由于改变和加密的短暂标识符，不可能将其设置为 Eddystone-EID 信标子集。

Beacon 类

- *BeaconType type* 是它所表示的信标的类型。
- *string UUID* 含义不同，取决于信标的类型：
 - Any: 是空字符串。
 - *iBeacon*: UUID
 - *Eddystone UID*: 命名空间
 - *Eddystone URL*: URL
 - *Eddystone EID*: 短暂标识符
- *int major* 是主要值。它是 0，如果它不是 *iBeacon*。
- *int minor* 是次要值。它是 0，如果它不是 *iBeacon*。
- *string instance* 是实例。如果它不是 *Eddystone UID*，它为空。
- *string regionName* 是信标所属区域的标识符。
- *int rssi* 是所测量的信标的 RSSI。如果不能确定，则为 127。
- *BeaconRange range* 是信标的粗略距离。

- *int strength* 是信标的校准的 tx 功率。如果不能确定，则为 *127*。
- *double accuracy* 是以米为单位的信标的距离。如果不能确定，则为 *-1*。
- *Telemetry telemetry* 包含信标的遥测数据。如果不发送，则为 *null*。
- *DateTime lastSeen* 是最后一次检测到信标。

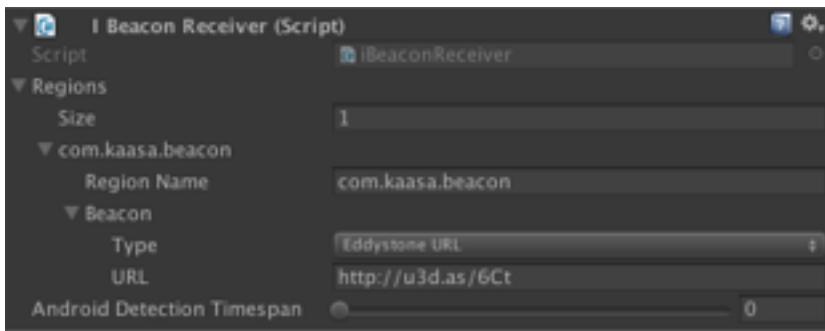
Telemetry 类

- *bool encrypted* 识别数据是否加密。
- *float voltage* 是以伏特为单位的当前电池电量。如果不支持或加密，则为 *0*。
- *float temperature* 是由信标感测的以摄氏度表示的温度。如果不支持或加密，则为 *-128*。
- *int frameCounter* 是从加电或重新启动以来由信标发射的所有类型的广告帧的运行计数。如果不支持或加密，则为 *0*。
- *TimeSpan uptime* 是信标上电或重新启动后的时间间隔。如果不支持或加密，则为 *null*。
- *byte[] encryptedData* 是信标的加密遥测数据。如果未加密，则为 *null*。
- *byte[] salt* 是来自广播的随机 salt。如果未加密，则为 *null*。
- *byte[] integrityCheck* 是消息的完整性检查。如果未加密，则为 *null*。

使用 BluetoothState

- *BluetoothLowEnergyState GetBluetoothLEStatus()* 返回设备上低功耗蓝牙的当前状态。
- *void EnableBluetooth()* 尝试启用蓝牙。如果设备不支持低功耗蓝牙，它将抛出 *iBeaconException*。
- *BluetoothStateChanged BluetoothStateChangedEvent* 每当设备上的蓝牙状态改变时将触发。

iBeaconReceiver 检查器

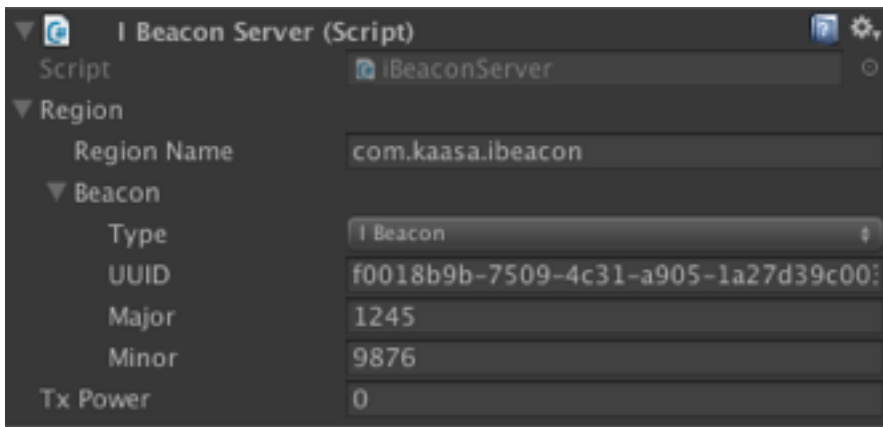


- *Regions* 是它将检测的所有 *iBeaconRegion* 的数组。
- 在安卓设备上，最后测量的 RSSI 值被平均以计算距离。 *Android Detection Timespan* 是选择这些值所需的时间间隔。

使用 iBeaconReceiver

- *void Scan()* 尝试开始检测信标。如果出现问题，它将抛出 *iBeaconException*。
- *void Stop()* 停止检测信标。
- *void Restart()* 应用更改并尝试重新启动检测信标。如果出现问题，它将抛出 *iBeaconException*。
- *BeaconRangeChanged BeaconRangeChangedEvent* 将在每次检测到信标时被触发。

iBeaconServer 检查器



- *Region* 是将被发送的 *iBeaconRegion*。
 - 在这种情况下，每个输入字段必须用不同于默认值的值填充。
 - iOS 设备只能作为 *iBeacon* 传输。
 - 不是每个能够检测信标的安卓设备都可以作为信标进行传输。有关详细信息，请参阅[系统要求](#)。
- *Tx Power* 是将被作为信号强度广播的值。指定 0 以使用设备的默认值。

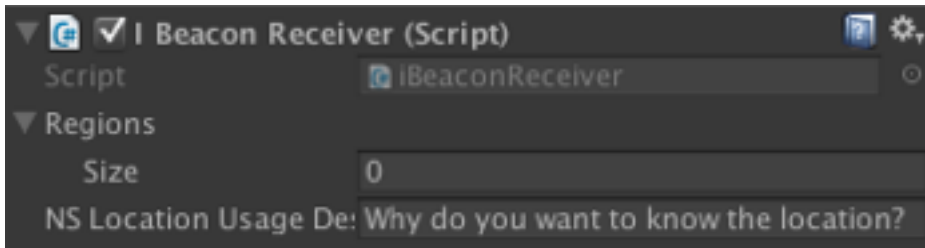
使用 iBeaconServer

- 如果设备可以作为信标进行传输，则返回 *bool checkTransmissionSupported()*。
- *void Transmit()* 尝试开始作为信标进行发送。如果出现问题，它将抛出 *iBeaconException*。
- *void StopTransmit()* 停止传输。
- *void Restart()* 应用更改并尝试重新启动作为信标的传输。如果出现问题，它将抛出 *iBeaconException*。

从旧版本升级

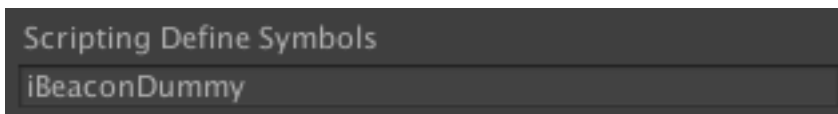
恢复旧区域值

升级后，旧区域值将不可见：

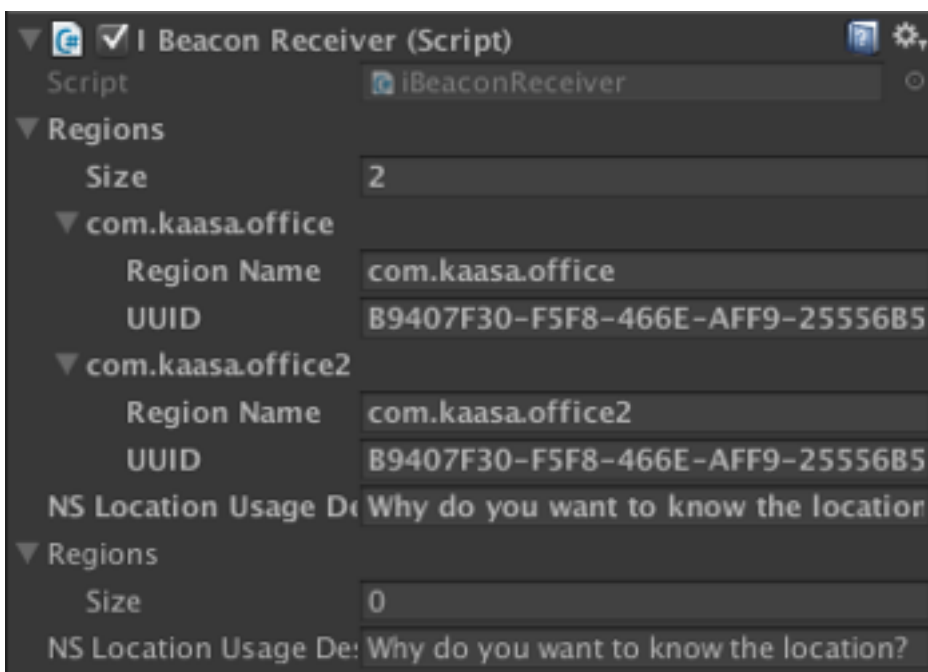


如果要重复使用它们，则必须执行以下步骤：

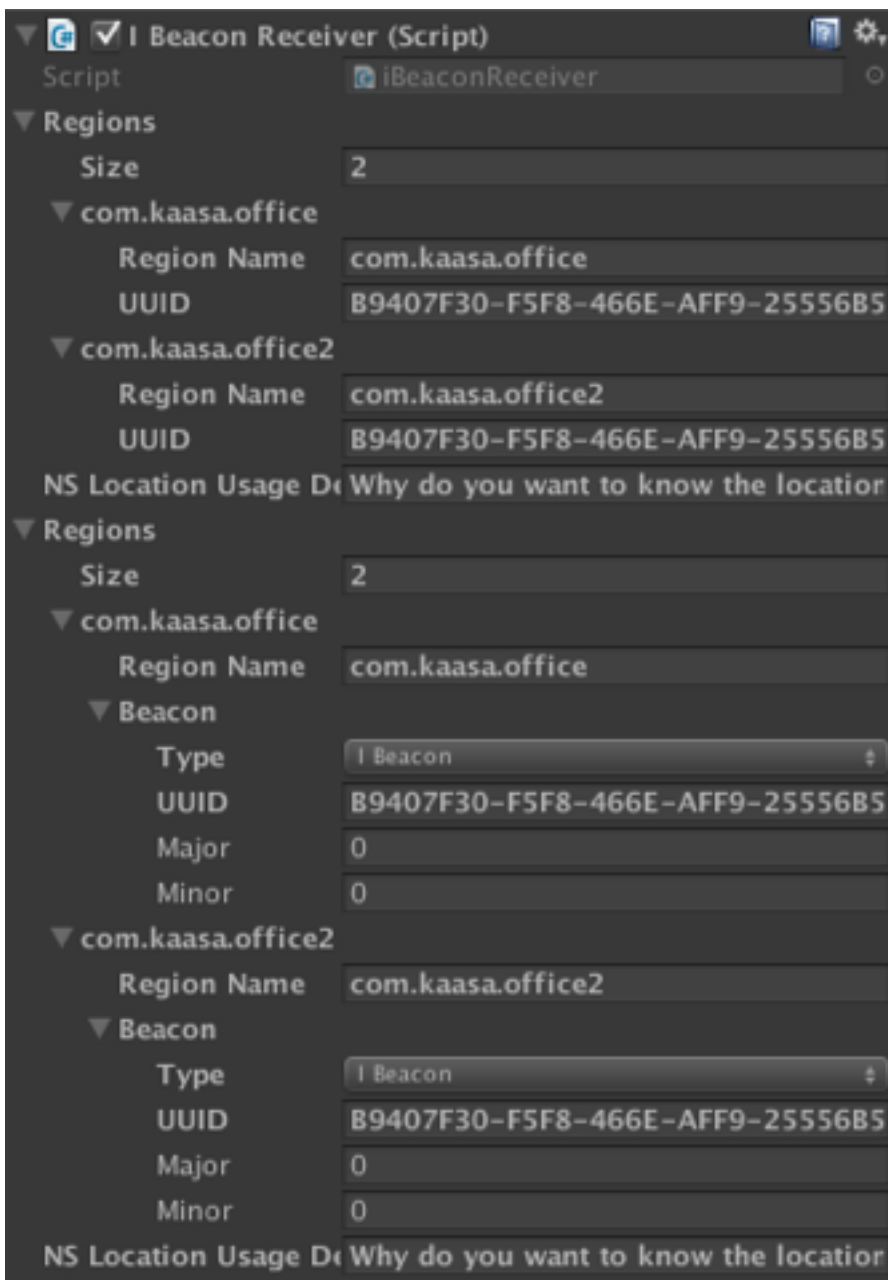
1. 转到 *Player Settings*。
2. 将 *iBeaconDummy* 添加到 *Scripting Define Symbols*。



3. 重新载入场景而不保存。
在虚拟模式下将出现错误日志。您将看到您的旧值。



4. 将旧值复制到新字段



5. 从 *Scripting Define Symbols* 中删除 *iBeaconDummy*。

好了。

代码更改

- *Init()* 已弃用。您可以删除它或使用 *Restart()*。
- *BluetoothStateChangedEvent* 和 *EnableBluetooth()* 从 *iBeaconReceiver* 移至 *BluetoothState*。
- *iBeaconReceiver.CheckBluetoothLEStatus()* 已删除。您可以通过 *BluetoothState.GetBluetoothLEStatus()* 获取当前的低功耗蓝牙状态。
- *BeaconRangeChanged(List<Beacon> beacons)* 被 *BeaconRangeChanged(Beacon[] beacons)* 替代。
- *iBeaconReceiver.NSLocationUsageDescription* 被删除。您可以在 Unity 的 *iOS Player Settings (Location Usage Description)* 中设置描述。

常见问题解答

您的插件是否支持公司的信标[在此填入名称]?

我们的插件支持根据 iBeacon 或 Eddystone 协议广播的每个信标。不支持在专有协议中广播的数据。

在哪里可以获得信标的 *Region Name*?

Region Name 是您必须为自己指定的唯一值。这不是从信标获得的值。

iOS 无法找到信标。为什么?

这些是常见原因:

- 设备不符合系统要求。
- 蓝牙已禁用。
- 未设定位置使用描述。您必须在 iOS 播放器设置中指定它。
- 有权访问该位置的权限已被拒绝。您可以在设备的“隐私”>“定位服务”下更改设置。
- 信标广播为 iBeacons，并且 *BeaconType* 被设置为 *Any*。为了检测 iBeacons，它需要是 *iBeacon* 类型。

安卓无法找到我的信标。为什么?

这些是常见原因:

- 设备不符合系统要求。
- 蓝牙已禁用。
- 安卓 6 及以上版本: 拒绝访问该位置的权限。您可以在设备的设置中更改它。

为什么来自只有一个地区的信标出现，虽然附近也有其他信标？

检查是否为您的区域定义了唯一且不同的 *Region Names*。

我可以在背景中对信标做出回应吗？

因为 Unity 在背景中不会工作，因此你编写的代码，失去信标焦点时会停止运行。