Outdoor AR Library can triggers events when users arrives at or leaves from certain locations or scenes. This tutorial shows how to write a Map View that sets up a location event manager and acts as a listener for location events. While the tutorial shows the feature in Map view, the same code can be applied to AR or List view components, as well.

### Setting up the Map View Activity

The first step in setting up our Map View Activity to act as a location event listener is to make it implement the OALocationEventListener interface. We then instantiate the location event manager in the `onCreate()` method of the activity. To do this we create a local variable, `locationManager`, of type OALocationEventManager, and use `getSensorManager()`, to pass the current sensor manager to its constructor.

Now that we have a location event manager we can pass it some scenes that we want to associate with location events. One way to do this is to retrieve the current list of scenes from the data manager, using `getDataManager().getSceneList()`, and then iterate over it and add each scene to the location event manager using `OALocationEventManager.addInterestedLocation()`. This method requires the scene to add as a parameter, and can also take a float value that acts as the range from the scene location that location events are triggered from. If the range is not specified then a default value is used. To specify a location to the manager that doesn’t have an associated scene we can use the same method but pass as parameters two doubles, representing latitude and longitude, instead of a scene.

Now we must implement the `onLocationEvent()` method that is required by our OALocationEventListener interface. This method takes an OALocationEvent as a parameter. Here is an example of what we might want to do in this method:

```java
@Override
public void onLocationEvent(OALocationEvent event) {
    String label;
    if (event.scene == null) {
```
label = "Location " + event.locationId;
} else {
    label = event.scene.getName();
}

if (event.type == OALocationEvent.EVENT_ARRIVED_AT) {
    Toast.makeText(getBaseContext(), "Entering vicinity of " + label,
    Toast.LENGTH_LONG).show();
} else if (event.type == OALocationEvent.EVENT_LEFT_FROM) {
    Toast.makeText(getBaseContext(), "Leaving vicinity of " + label,
    Toast.LENGTH_LONG).show();
}

This implementation simply takes the name of the scene involved with the OALocationEvent (or the location id if there is no scene associated) and toasts a message stating that the user is either entering or leaving the vicinity of the location depending on the type of the OALocationEvent. Once this method is done we now have a working Map View Activity that handles location events.

If you have further problems or questions, visit our website (http://www.hitlabnz.org/mobileAR) and post your problem on the Mobile AR Framework Support Forum.