LocationComponent

This guide will demonstrate how to utilize the LocationComponent to represent the user's current location.

1. When implementing the LocationComponent, the application should request location permissions.

   Declare the need for foreground location in the `AndroidManifest.xml` file.

   For more information, please refer to the Android Developer Documentation.

```
<manifest ... >
  <!-- Always include this permission -->
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <!-- Include only if your app benefits from precise location access. -
->
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

2. Create a new activity named `BasicLocationPulsingCircleActivity`:

- This Activity should implement the `OnMapReadyCallback` interface. The `onMapReady()` method is triggered when the map is ready to be used.
- Add a variable `permissionsManager` to manage permissions.
- Add a variable `locationComponent` to manage user location.
- At the end of the `onCreate()` method, call `checkPermissions()` to ensure that the application can access the user's location.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/location/BasicLocationPulsingCircleActiv
ity.kt:top}}
```

3. In the `checkPermissions()` method, the PermissionManager is used to request location permissions at runtime and handle the callbacks for permission granting or rejection.

   Additionally, you should pass the results of `Activity.onRequestPermissionResult()` to it.

   If the permissions are granted, call `mapView.getMapAsync(this)` to register the activity as a listener for onMapReady event.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/location/BasicLocationPulsingCircleActiv
ity.kt:permission}}
```

4. In the `onMapReady()` method, first set the style and then handle the user's location using the LocationComponent.

   To configure the LocationComponent, developers should use LocationComponentOptions.

   In this demonstration, we create an instance of this class.

   In this method:

   - Use the annotation `@SuppressLint("MissingPermission")` to suppress warnings related to missing location access permissions.
   - In `setStyle(),` you can utilize other public and token-free styles like demotiles instead of the predefined styles.
   - For the builder of LocationComponentOptions, use `pulseEnabled(true)` to enable the pulse animation, which enhances awareness of the user's location.
   - Use method `buildLocationComponentActivationOptions()` to set LocationComponentActivationOptions, then activate `locatinoComponent` with it.
   - To apply options, make sure you call `activateLocationComponent()` of `locationComponent`. You can also set `locationComponent`'s various properties like `isLocationComponentEnabled` , `cameraMode` , etc...
   - `CameraMode.TRACKING`[^1] means that when the user's location is updated, the camera will reposition accordingly.
   - `locationComponent!!.forceLocationUpdate(lastLocation)` updates the the user's last known location.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/location/BasicLocationPulsingCircleActiv
ity.kt:onMapReady}}
```

5. LocationComponentActivationOptions is used to hold the style, LocationComponentOptions and other locating behaviors.
   - It can also be used to configure how to obtain the current location, such as LocationEngine and intervals.
   - In this demonstration, it sets 750ms as the fastest interval for location updates, providing high accuracy location results (but with higher power consumption).
   - For more information, please visit the documentation page.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/location/BasicLocationPulsingCircleActiv
ity.kt:LocationComponentActivationOptions}}
```

6. For further customization, you can also utilize the `foregroundTintColor()` and `pulseColor()` methods on the LocationComponentOptions builder:

```
val locationComponentOptions =

LocationComponentOptions.builder(this@BasicLocationPulsingCircleActivity
)
        .pulseEnabled(true)
        .pulseColor(Color.RED)             // Set color of pulse
        .foregroundTintColor(Color.BLACK)  // Set color of user location
        .build()
```

7. Here is the final results with different color configurations. For the complete content of this demo, please refer to the source code of the Test APP [^2].

   result

[^1]: A variety of camera modes determine how the camera will track the user location.
They provide the right context to your users at the correct time.
[^2]: In Test APP, it also uses menu items to manage user location icon.