Documentation

Language: **Swift** ⌄

MapKit / … / MapKit annotations / Decluttering a Map with MapKit Annotation Clustering

Sample Code

Decluttering a Map with MapKit Annotation Clustering

Enhance the readability of a map by replacing overlapping annotations with a clustering annotation view.

Download

iOS 18.0+ | iPadOS 18.0+ | Xcode 16.3+

Overview

TANDm is a fictional bike sharing app that uses annotation clustering to provide an uncluttered map. The app shows how MapKit automatically groups two or more annotations into a single annotation when spacing on the map doesn’t permit each annotation to be visible without overlapping. This enhances the readability of the map by replacing overlapping annotations with a clustering annotation view.

Annotation Clustering

To group annotations into a cluster, set the `clusteringIdentifier` property to the same value on each annotation view in the group. For example, to show overlapping unicycle annotations in a clustering annotation view, TANDm sets `clusteringIdentifier` on each instance of `[UnicycleAnnotationView]`[2] to `"unicycle"`.

```
override init(annotation: MKAnnotation?, reuseIdentifier: String?) {
    super.init(annotation: annotation, reuseIdentifier: reuseIdentifier)
    clusteringIdentifier = "unicycle"
}
```

Display Priority

To determine how an annotation view behaves when it overlaps another annotation view, set its `displayPriority` property. In the sample app, the map view is likely to hide the unicycle annotation if it overlaps with another annotation because the unicycle annotation view has a display priority of `defaultLow`, while the display priorities for bicycle and tricycle are set to `defaultHigh`. Here's an example of setting the display priority while preparing an instance of [`BicycleAnnotationView`][8] for reuse:

```
override func prepareForDisplay() {
    super.prepareForDisplay()
    displayPriority = .defaultHigh
    markerTintColor = UIColor.bicycleColor
    glyphImage = #imageLiteral(resourceName: "bicycle")
}
```

Custom Clustering Annotation Views

Customize the behavior and appearance of a clustering annotation view by subclassing `MKAnnotationView`; for instance, to display hints about the annotations within the cluster. TANDm, for example, uses the custom clustering annotation view [`ClusterAnnotationView`][7] to show the ratio between bicycles and tricycles at a location.

```
override func prepareForDisplay() {
    super.prepareForDisplay()

    if let cluster = annotation as? MKClusterAnnotation {
        let totalBikes = cluster.memberAnnotations.count

        if count(cycleType: .unicycle) > 0 {
            image = drawUnicycleCount(count: totalBikes)
        } else {
            let tricycleCount = count(cycleType: .tricycle)
            image = drawRatioBicycleToTricycle(tricycleCount, to: totalBikes)
        }

        if count(cycleType: .unicycle) > 0 {
            displayPriority = .defaultLow
        } else {
            displayPriority = .defaultHigh
        }
    }
}
```

See Also

Grouped annotations

`class MKClusterAnnotation`

An annotation that groups two or more distinct annotations into a single entity.