UNBOXING

FELIX WEHNERT & ANGELO CAMMALLERI

mapbox
UNBOXING MAPBOX

TABLE OF CONTENTS

▸ What does Mapbox do?
▸ Styling with Mapbox Studio
▸ Turn-by-turn
▸ Camera
▸ Annotations
▸ Polylines
WHAT DOES MAPBOX DO?

- Beautiful, powerful and accessible map framework
- Powered by openstreetmaps.com
- Cross-platform support with native (partly open source) frameworks
- Pay-as-you-go pricing model
STYLING WITH MAPBOX STUDIO

- Style nearly **everything**
- Style custom data that will appear on the map
- Adjust color, font, sizes, blur, icons, text alignment and much more
- Very detailed categories to style
STYLING WITH MAPBOX STUDIO

- Export style url from Mapbox studio
- Using your custom style in iOS:

```swift
let styleURL = URL(string: "mapbox://styles/icemanhh82/cjg6amys40m1s2rppvjsq8giy")!

let mapView = MGLMapView(frame: mapContainer.frame, styleURL: styleURL)
```
STYLING WITH MAPBOX STUDIO
TURN-BY-TURN NAVIGATION

- Prebuilt Turn-by-turn navigation
- Voice guidance
- Different UI for day and night
- Ready to use
- Custom styling support (limited)
- It is possible to create an own Turn-by-turn view from scratch
CAMERA

- Simple camera manipulation
- Create a camera
- Fly to it

```swift
extension MGLMapView {
    func flyToUser(trackingMode: MGLUserTrackingMode = .followWithCourse) {
        if let location = self.userLocation {
            let camera = MGLMapCamera(lookingAtCenter: location.coordinate,
                                        fromDistance: 200,
                                        pitch: 60,
                                        heading: 0)

            self.fly(to: camera, withDuration: 3) {
                self.userTrackingMode = trackingMode
            }
        }
    }
}
```
UNBOXING MAPBOX

CAMERA

Havana’s Forgotten Treasures
UNBOXING MAPBOX

ANNOTATIONS

- Completely customizable
- Annotate a map with custom UIViews
- Annotate custom or Mapbox data
- Various types of annotations to support even large datasets
- Clustering included
Annotations

- Custom user location annotation
- Switching shape and color
- Rotation by driving direction
It’s easy to display polylines on a MGLMapView:

```swift
// First creating a MGLShapeSource to hold our polyline
let polylineSource = MGLShapeSource(identifier: "polyline",
    shape: nil,
    options: nil)

// Adding it to the mapView in MGLMapViewDelegate
func mapView(_ mapView: MGLMapView, didFinishLoading style: MGLStyle) {
    style.addSource(polylineSource)
    style.addLayer(polylineSource.porscheStyle)
}

// var mutableCoordinates: [CLLocationCoordinate2D] from somewhere
// If we want to display the coordinates we need to:
let polyline = MGLPolylineFeature(coordinates: &mutableCoordinates,
    count: UInt(mutableCoordinates.count))

polylineSource.shape = polyline
```
UNBOXING MAPBOX

POLYLINES

- Customizable color
- Independently configureable zoom levels
- Joining of multiple lines
- Style the end points of lines
UNBOXING MAPBOX

SOURCES

- Custom user location annotation
- Interactive Pioneers
- Gallery | Mapbox
PIZZA

SPONSORED BY INTERACTIVE PIONEERS