



Memojis Under the Hood

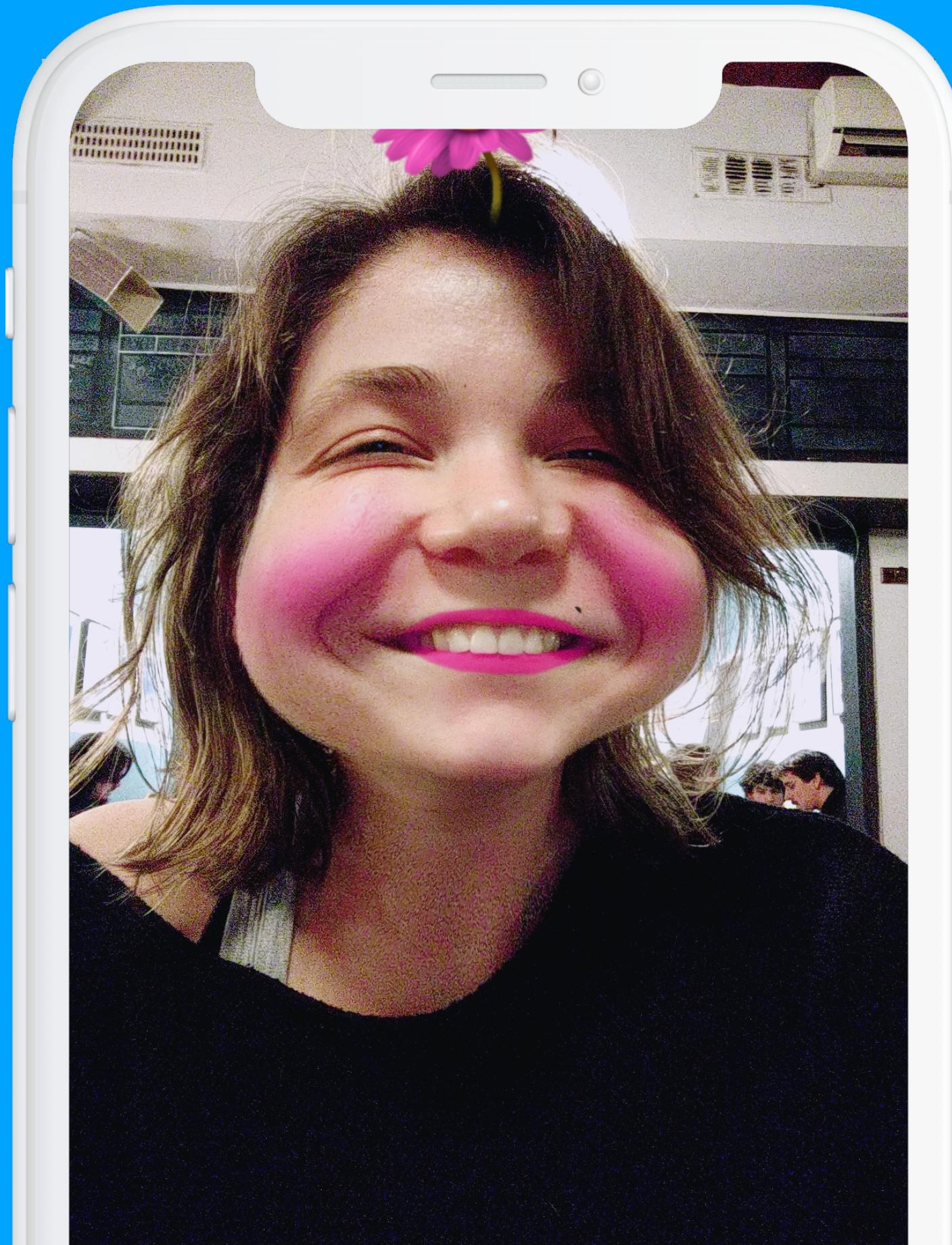
FaceTracking using ARKit



→ **Berta Devant**

→ **IOS Developer @Novoda**

→ **@bertadevant**



→ AR 101

→ ARKit Face Tracking

→ The Future ™

AR WHAT??

noveda

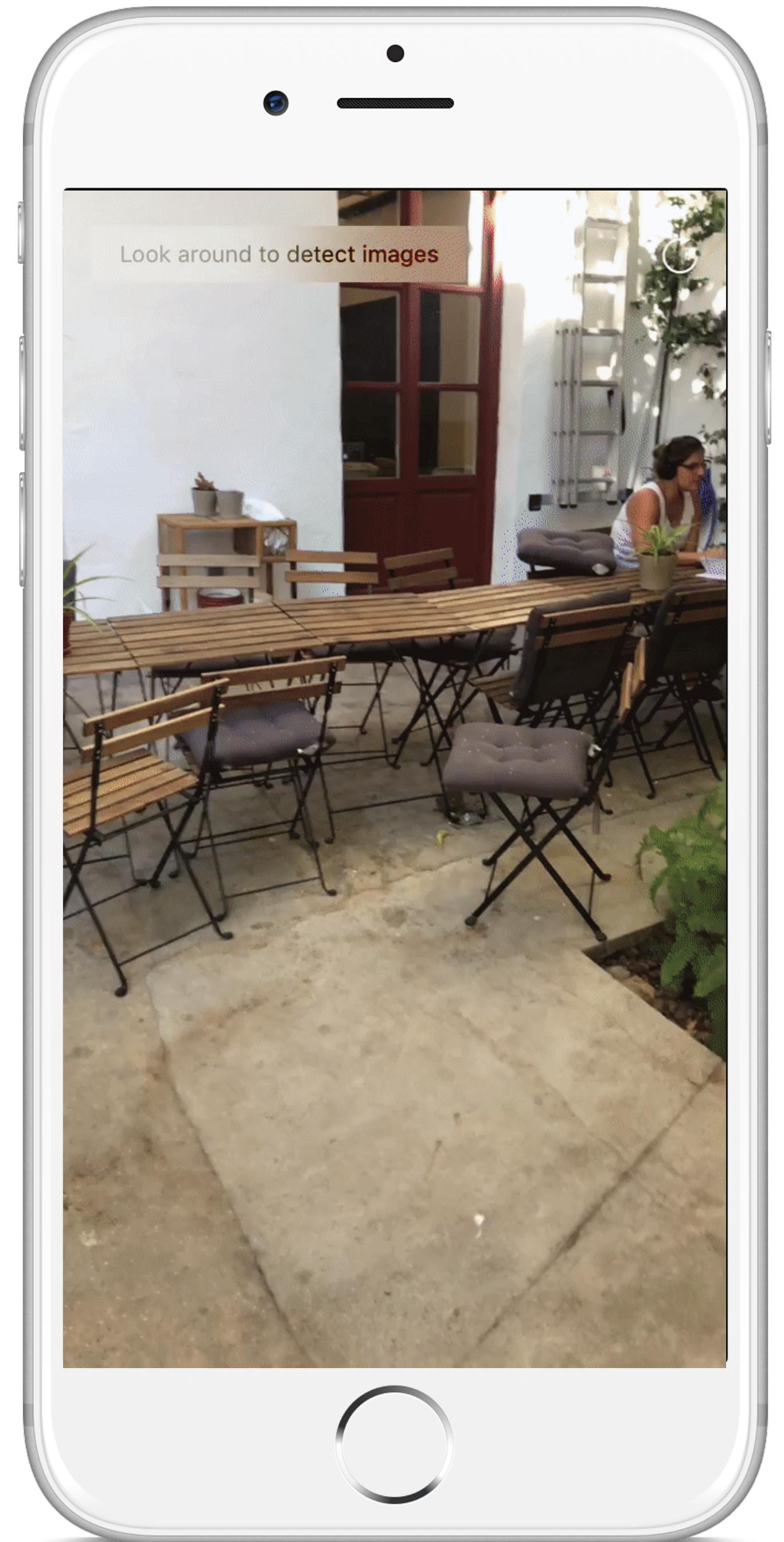
- **World Tracking & Scene Understanding**
- **Face Tracking**
- **Rendering**
- **Image Tracking**
- **3D object detection**
- **Persistence & Multi user**





All the words

- Feature points
- AR Anchors
- AR Scene View
- AR Session
- AR Tracking Configuration
- ARSCNViewDelegate



Show code or go home



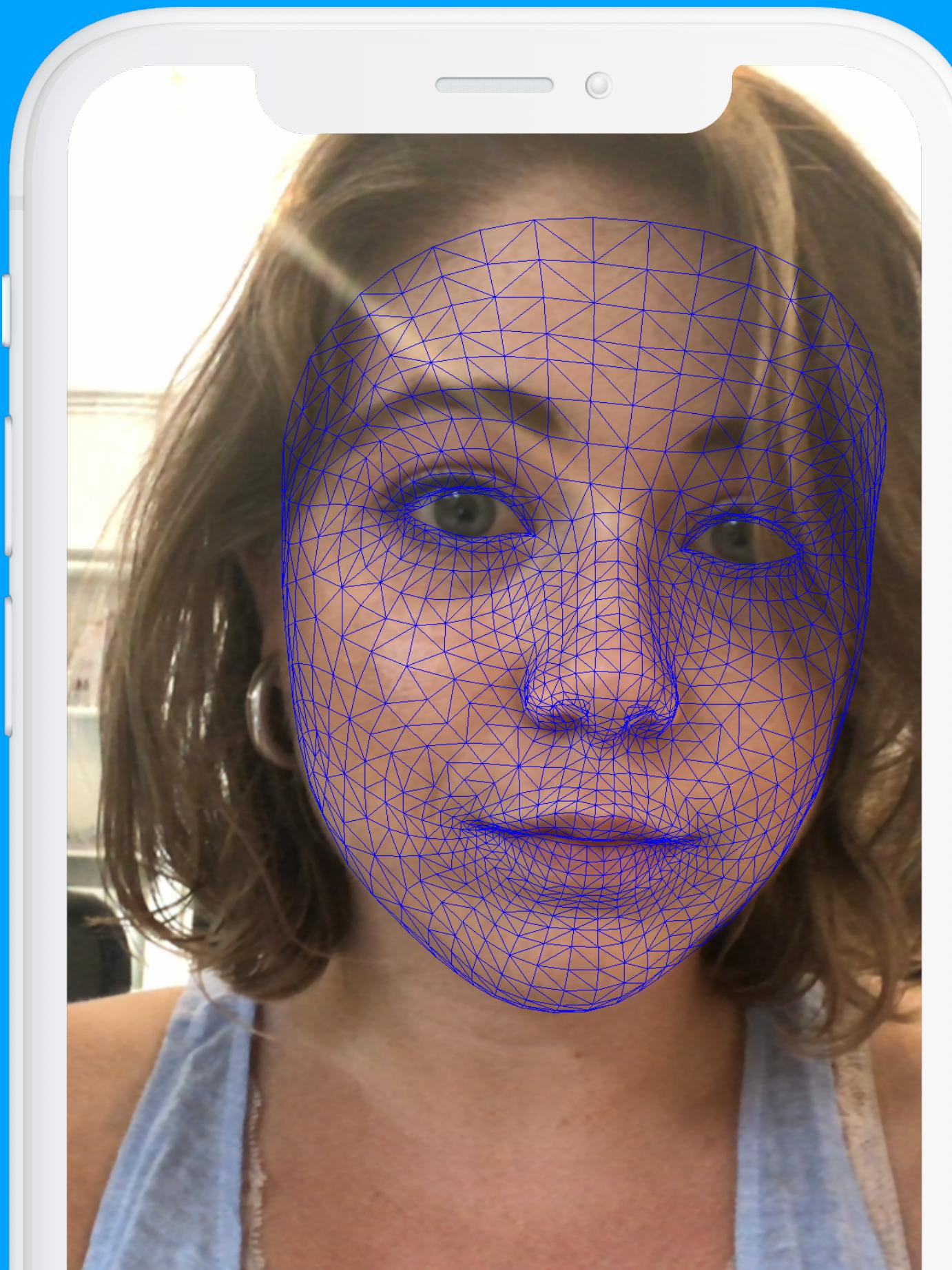
ARSCNViewDelegate

```
func renderer(_ renderer: SCNSceneRenderer, didAdd node: SCNNode, for anchor: ARAnchor) { }
func renderer(_ renderer: SCNSceneRenderer, didUpdate node: SCNNode, for anchor: ARAnchor) { }
```



```
guard ARFaceTrackingConfiguration.isSupported else { return }
```

```
let configuration = ARFaceTrackingConfiguration()
configuration.isLightEstimationEnabled = true
session.run(configuration, options: [.resetTracking, .removeExistingAnchors])
```



→ **ARFaceAnchors**

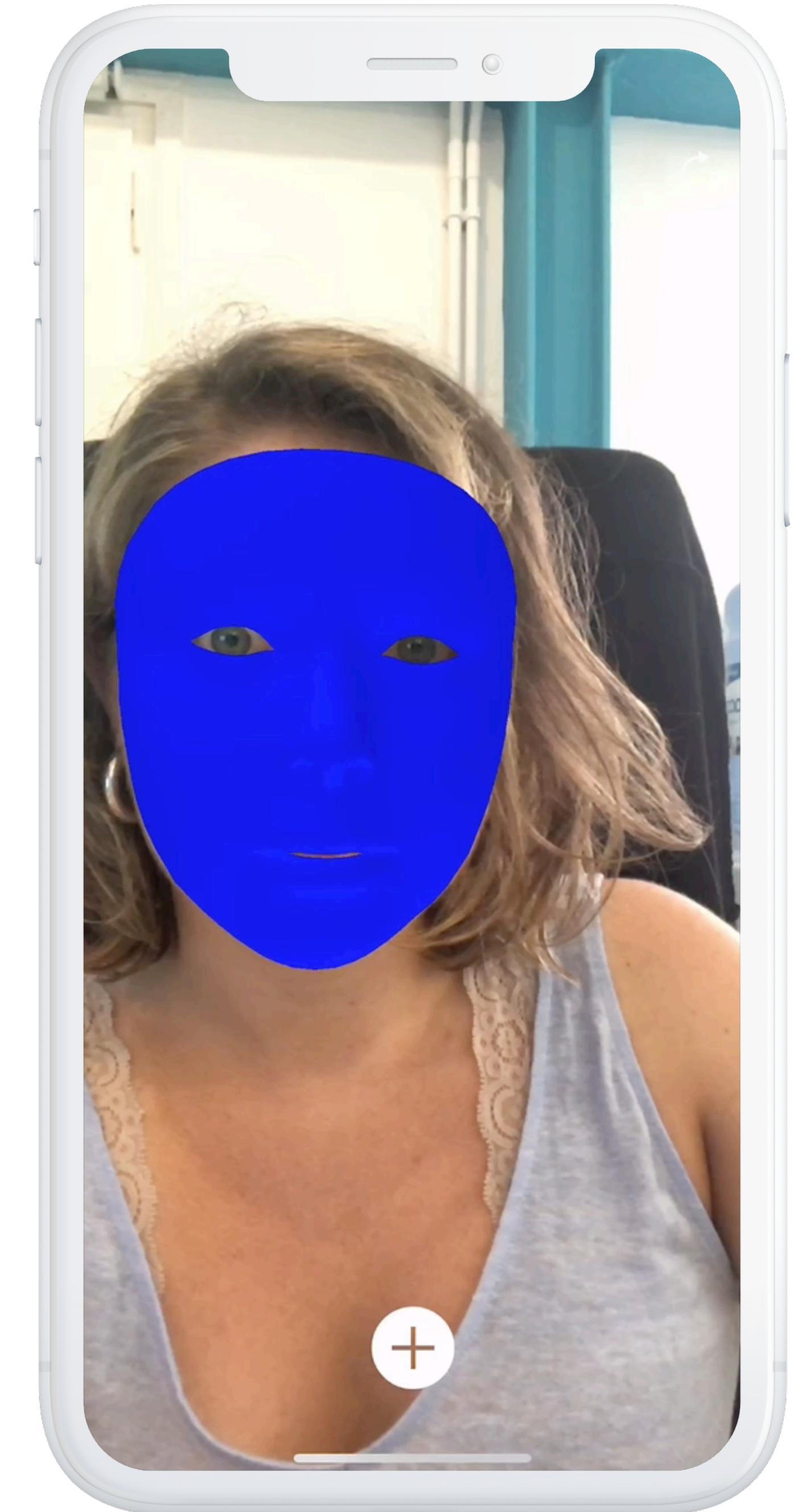
→ **ARFaceGeometry**

→ **ARSCNFaceGeometry**



ARFaceAnchor : ARAnchor

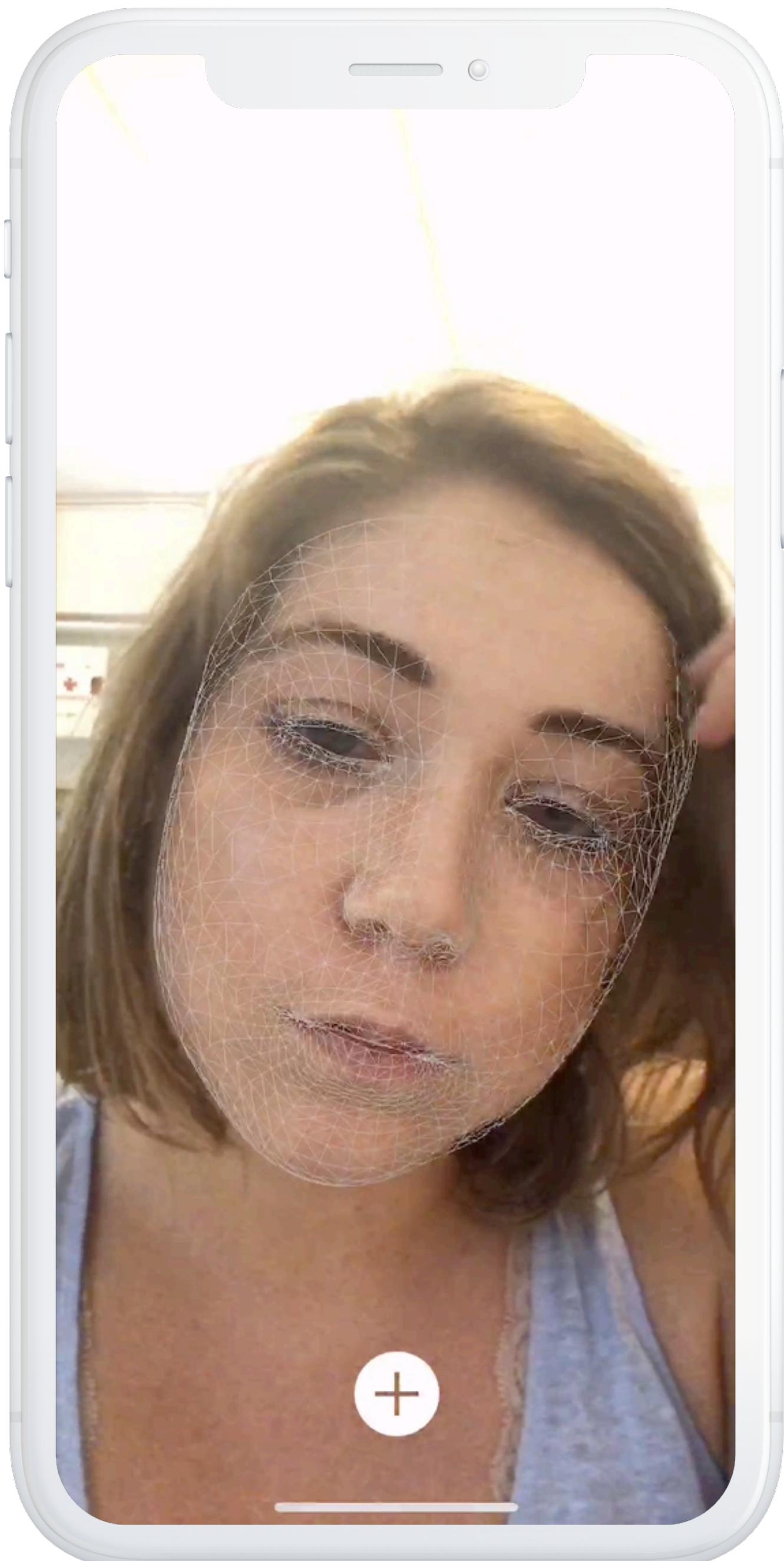
- Position and orientation relative to World
- Geometry
- 52 blendShape Movements
 - leftEye
 - rightEye
 - lookAtPoint





ARFaceGeometry

- 3D description of the Face
- Init(blendShapes)
- Topology of mesh
 - Vertices
 - textureCoordinates
 - triangleIndices



Show code or go home



```
guard let device = sceneView.device,  
    let maskGeometry = ARSCNFaceGeometry(device: device) else {  
    print("failed getting geometry")  
    return  
}  
  
let faceMesh = SCNNode(geometry: maskGeometry)
```

Show code or go home



```
class FaceMesh: SCNNode, VirtualFaceContent {

    init(geometry: ARSCNFaceGeometry) {
        geometry.firstMaterial?.fillMode = .lines

        super.init()
        self.geometry = geometry
    }

    func update(withFaceAnchor anchor: ARFaceAnchor) {
        guard let faceGeometry = geometry as? ARSCNFaceGeometry else {
            return
        }
        faceGeometry.update(from: anchor.geometry)
    }
}
```

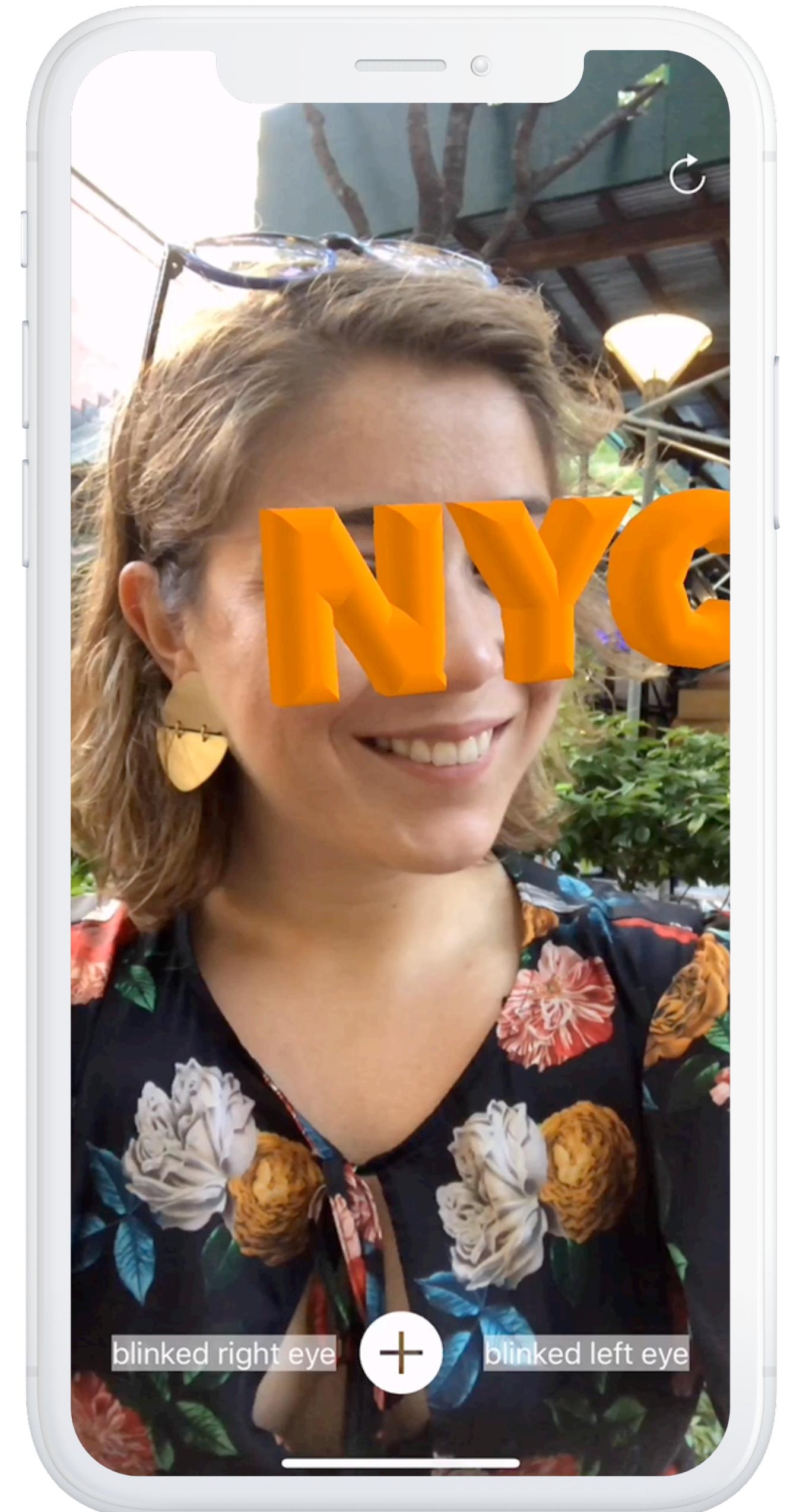


Show me!!!



ARSCNFaceGeometry: SCNGeometry

- SceneKit representation of face topology
“What you see on the screen”
- Update(from: ARFaceGeometry)
Updates movements and expressions only
- init?(device: MTLDevice)
Need Device



Show code or go home



```
● ● ●

class Overlay: SCNNode {

    let occlusionNode: SCNNode

    init(geometry: ARSCNFaceGeometry) {

        geometry.firstMaterial!.colorBufferWriteMask = []
        occlusionNode = SCNNode(geometry: geometry)
        occlusionNode.renderingOrder = -1

        super.init()

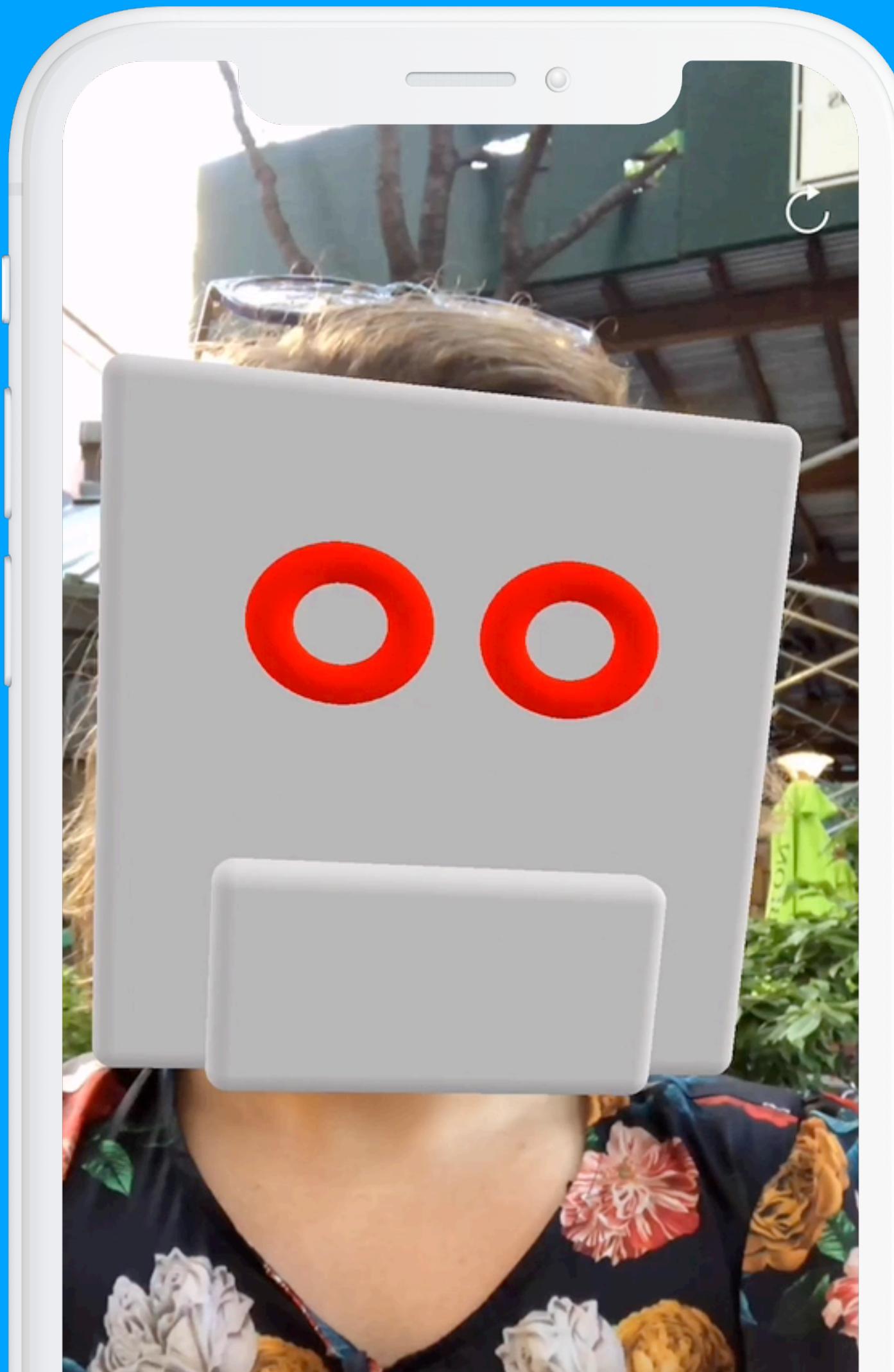
        addChildNode(occlusionNode)

        // Add 3D content.
        guard let overlayContent = loadedContentForAsset(named: "overlayModelNYC") else {
            return
        }
        addChildNode(overlayContent)
    }

    func update(withFaceAnchor anchor: ARFaceAnchor) {
        let faceGeometry = occlusionNode.geometry as! ARSCNFaceGeometry
        faceGeometry.update(from: anchor.geometry)
    }
}
```



Show me!!!



- **Face Alterations**
- **Face Identification**
- **Security**
- **Occlusion Geometry**
- **Accessibility**
- **User Interaction**



Thanks !

<https://github.com/novoda/ios-demos>



@bertadevant