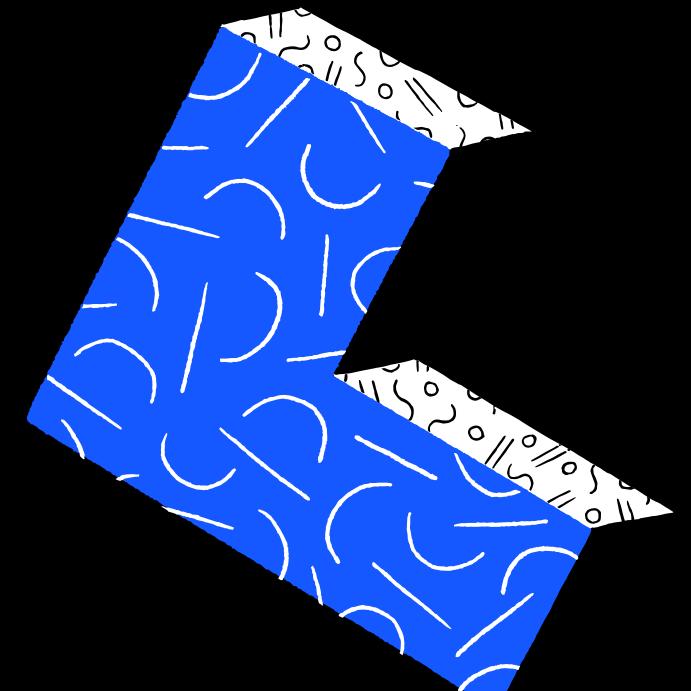
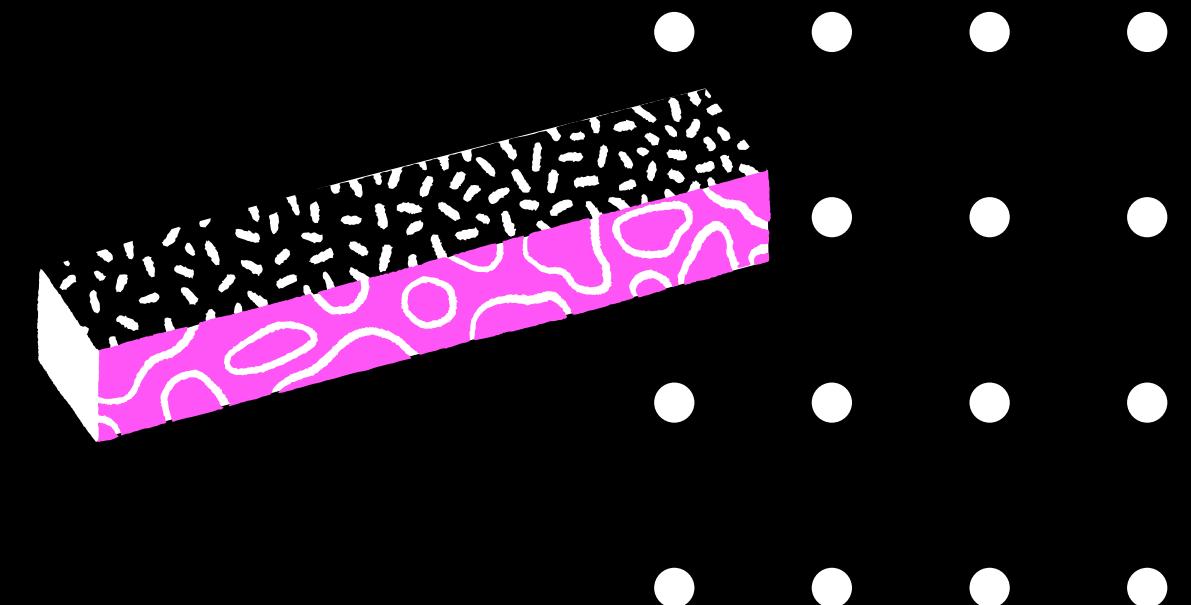
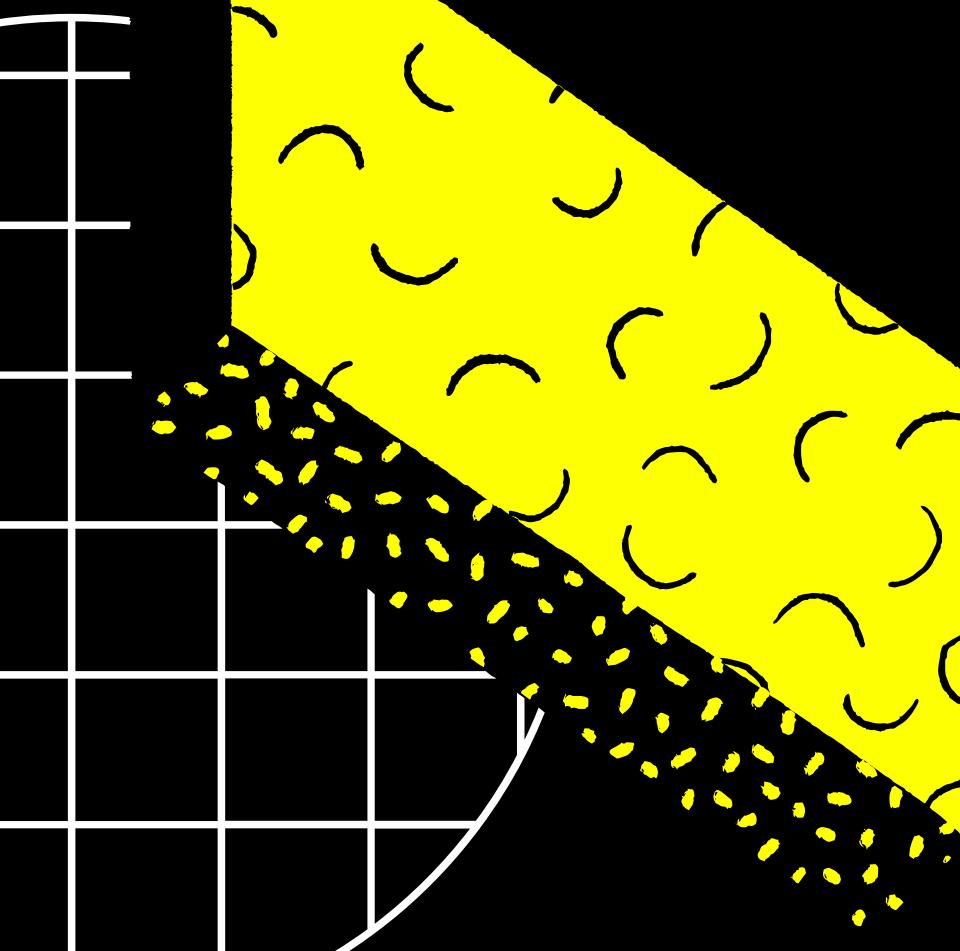


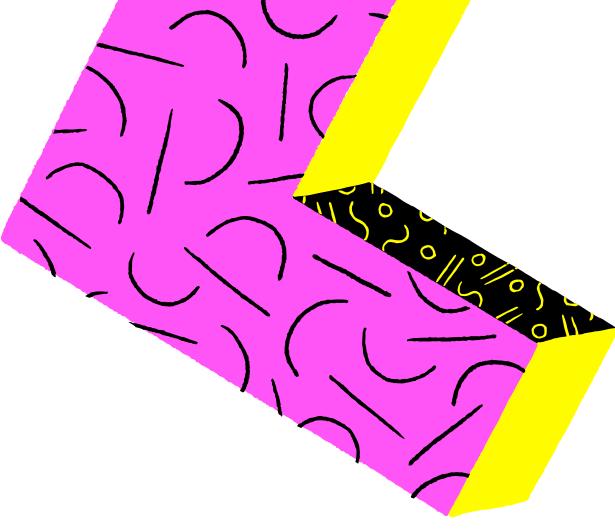


Borderless  
Engineering  
Conference

# AUGMENTED REALITY FACE FILTERS WITH IOS

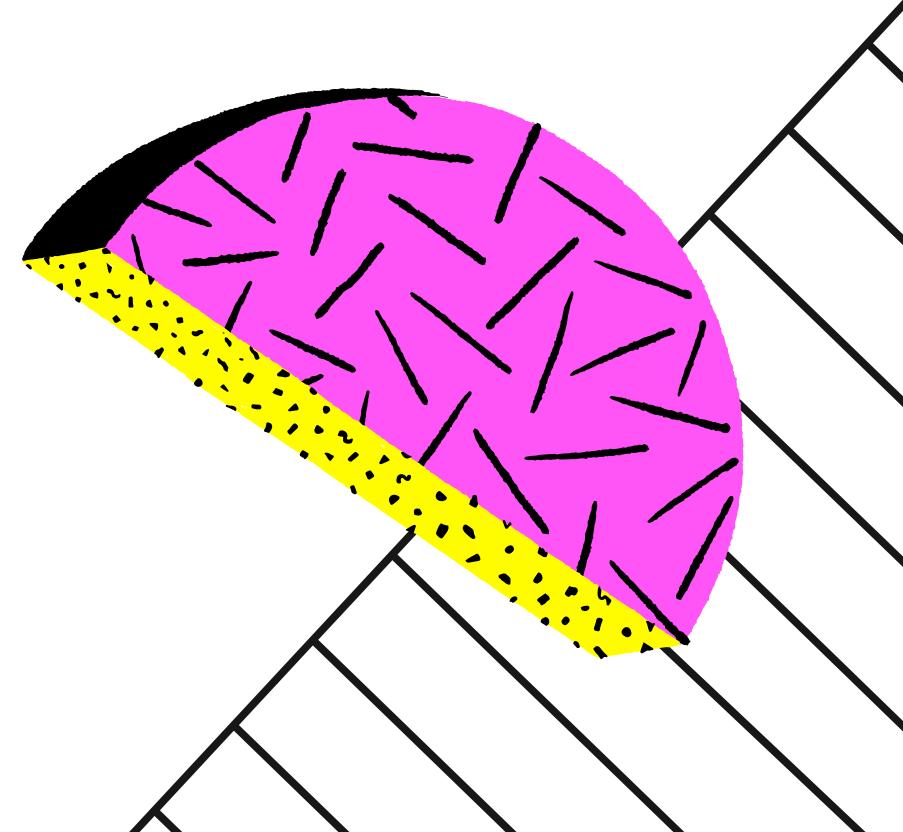
Roxana Jula





**”Build unparalleled augmented reality experiences for hundreds of millions of users on iOS and iPadOS, the biggest AR platforms in the world.”**

- Apple



**~ 650 million ARKit-compatible devices**

- arinsider.co (2019)



# Apple AR Glasses coming soon?

[Great roundup by MacRumours](#)

 **Jon Prosser**  @jon\_prosser · May 21

wwdc is gonna be crazy

302

522

8.8K



## Facebook Project Aria



Lauren Cason

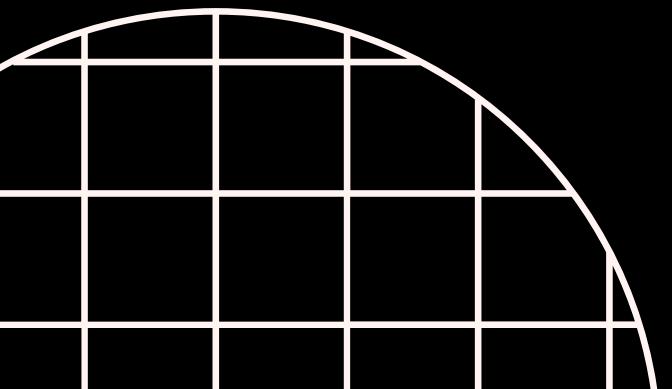
“

AR is an opportunity to create a new world. We may not like how we've done that with technology in the past, but we have an opportunity to create a new way.

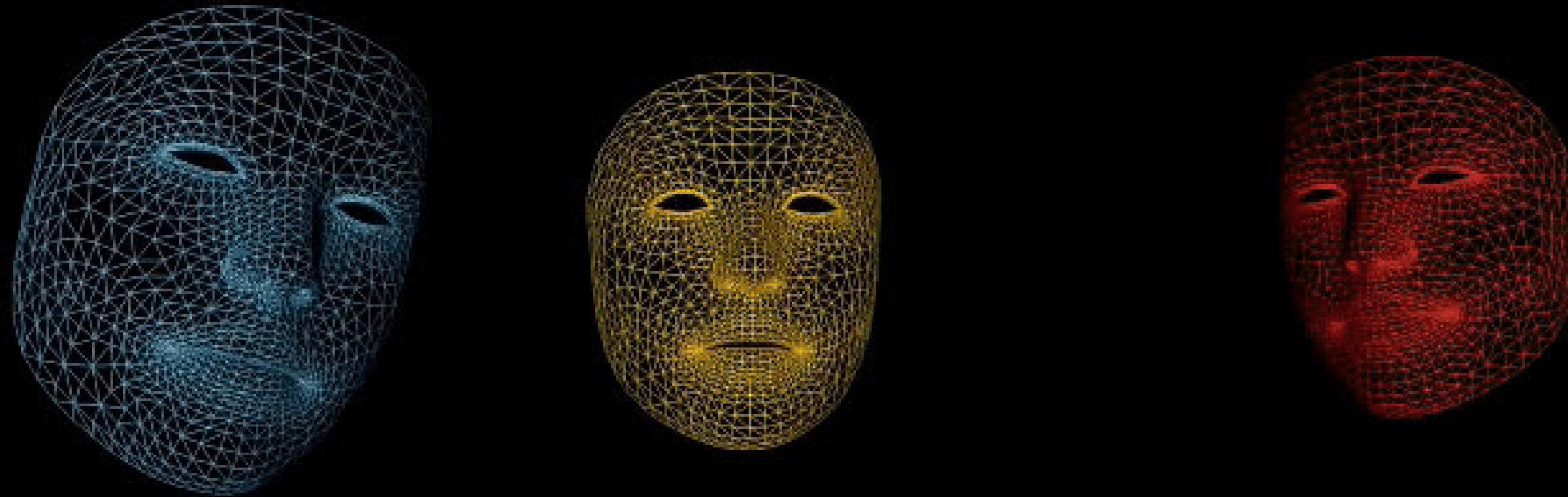


Image Source: Facebook

## Snapchat Spectacles



# Face tracking



- Detect faces in your front-facing camera
- Overlay virtual content and animate facial expressions in real-time
- Track up to 3 faces simultaneously

# Face tracking support

- Supported on any device with A12 Bionic chip and later

	TrueDepth camera	A12 or later
Face anchors	✓	✓
Face geometry	✓	✓
Blendshapes	✓	✓
Captured depth data	✓	✗

# Face Tracking

with ARKit

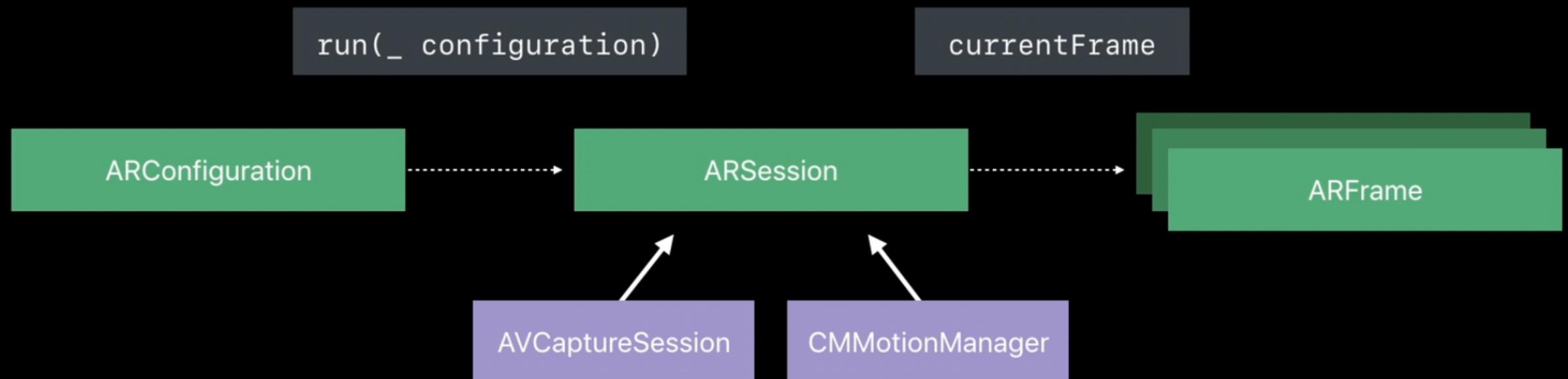


- Face detection and positional tracking in 6 Degrees of Freedom
- Facial expressions tracked in real-time with data on over 50 specific muscle movements available
- Access to the front facing camera color image as well as the front-depth image (if supported)
- ARKit uses your face as a light probe to estimate lighting conditions



# How it works?

- **ARSession** handles all processing done for ARKit
- For face filters you will use **ARFaceTrackingConfiguration**
- When you run a session, ARKit will configure the **AVCaptureSession** and **CMMotionManager** automatically
- After processing, results will be outputted as **ARFrames**
- Each **ARFrame** provides camera images, tracking data and anchor points



- **Blend shapes**, high level model of the current facial expression, a dictionary of named coefficients representing the pose of specific textures - eyelids, nose, jaw, etc -> floating point values from 0 to 1



## Blend Shapes

### ARBlendShapeLocation Coefficients

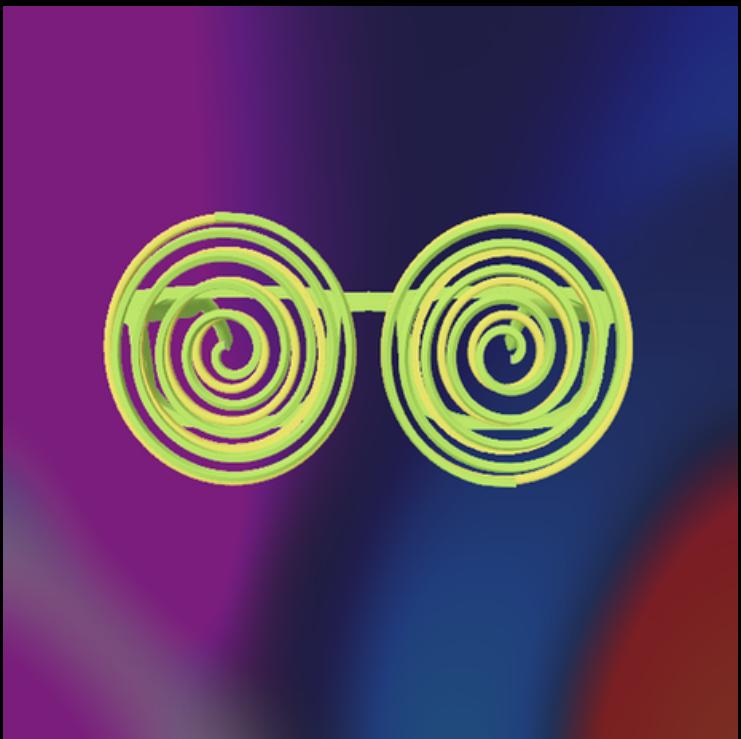
BrowDownLeft	EyeLookInRight	MouthClose	MouthRollLower
BrowDownRight	EyeLookOutLeft	MouthDimpleLeft	MouthRollUpper
BrowInnerUp	EyeLookOutRight	MouthDimpleRight	MouthShrugLower
BrowOuterUpLeft	EyeLookUpLeft	MouthFrownLeft	MouthShrugUpper
BrowOuterUpRight	EyeLookUpRight	MouthFrownRight	MouthSmileLeft
CheekPuff	EyeSquintLeft	MouthFunnel	MouthSmileRight
CheekSquintLeft	EyeSquintRight	MouthLeft	MouthStretchLeft
CheekSquintRight	EyeWideLeft	MouthLowerDownLeft	MouthStretchRight
EyeBlinkLeft	EyeWideRight	MouthLowerDownRight	MouthUpperUpLeft
EyeBlinkRight	JawForward	MouthPressLeft	MouthUpperUpRight
EyeLookDownLeft	JawLeft	MouthPressRight	NoseSneerLeft
EyeLookDownRight	JawOpen	MouthPucker	NoseSneerRight
EyeLookInLeft	JawRight	MouthRight	

# What we will build

**1/AR Quick Look**

**2/RealityKit**

**3/SceneKit**



# AR Quick Look & USDZ

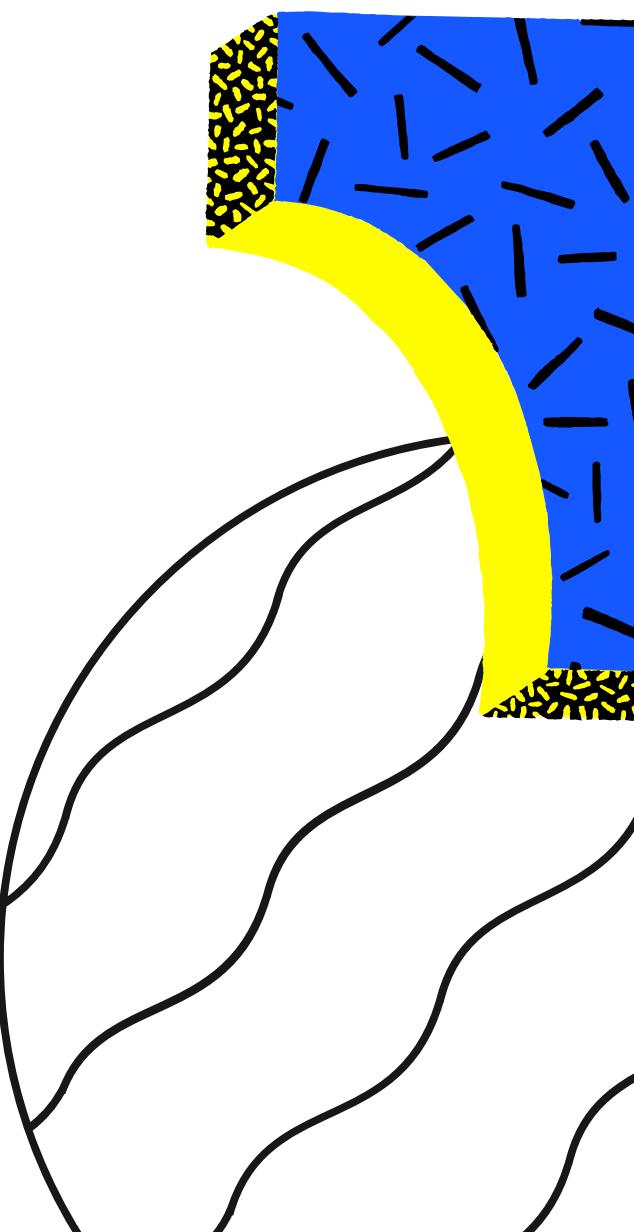
## AR QUICK LOOK

- AR Quick Look provides a very powerful augmented reality experience with user interactions like moving/scaling the object, people occlusion and sharing of the model supported “out of the box”.
- AR Quick Look supports 2 input formats: **.usdz** and **.reality**.



## USDZ

- The technology behind it is called **USD (Universal Scene Description)**.
- It is a 3D file format developed by **Pixar** and it has a focus on speed, scalability and collaboration.
- USDZ is the distribution format for USD, which is a compact single file that is optimised for sharing.



# RealityKit vs. SceneKit

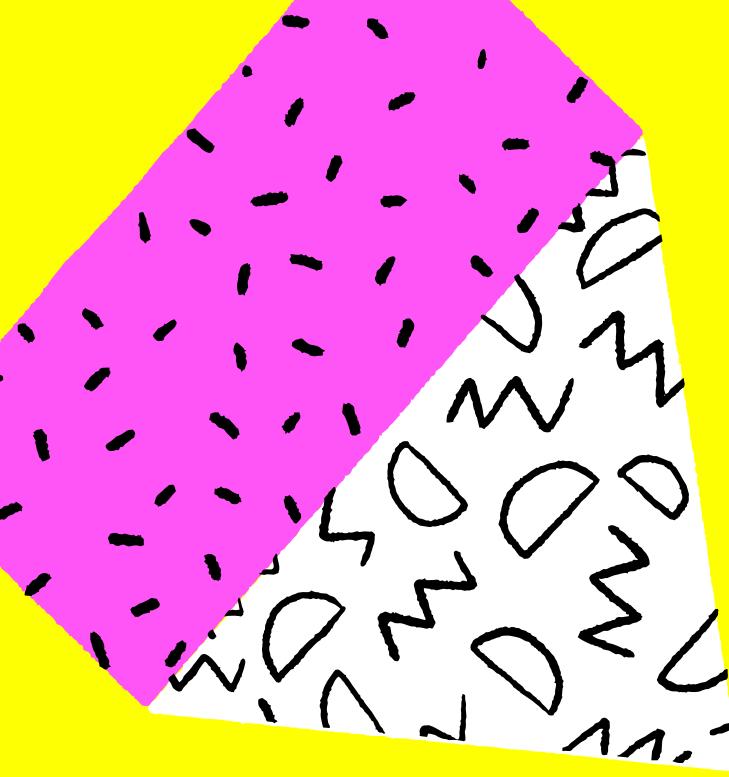


## SCENEKIT

- ARKit originally built on top of SceneKit (3D graphics framework developed for mobile games)
- SceneKit is quite neglected with no major recent updates
- *More resources available (projects, tutorials, StackOverflow/Apple Forums questions and answers)*

## REALITYKIT (NEWER)

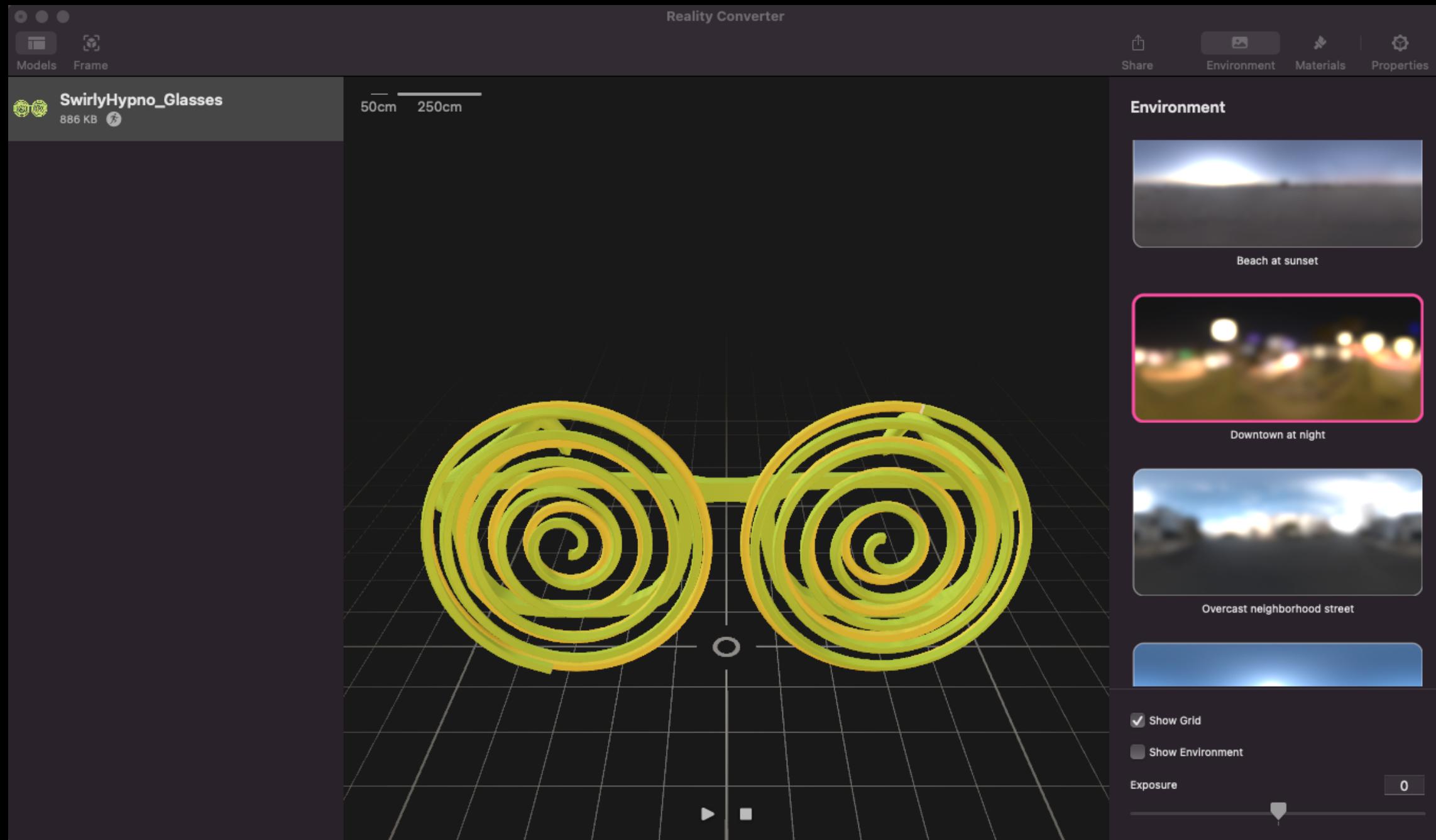
- High performance framework, built from the ground up for AR experiences with a more familiar and less intimidating API
- Reality Composer integration
- Video textures support
- Easier shared AR experiences
- *Less resources available (projects, tutorials, StackOverflow/Apple Forums questions and answers)*



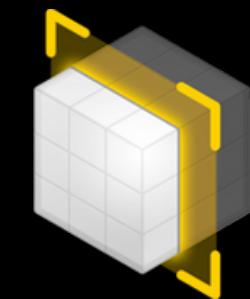
# **Great tools to know when developing iOS AR experiences**



# Reality Converter



- BETA 3 version
- Convert, view and customise USDZ 3D objects
- Drag and drop functionality
- Supports common file formats such as .obj, .gltf and .usd
- Preview the objects under different lighting and environment conditions
- USDZ Python based tools also available for generating, validating and inspecting USDZ files
- *Available on macOS*



# Reality Composer

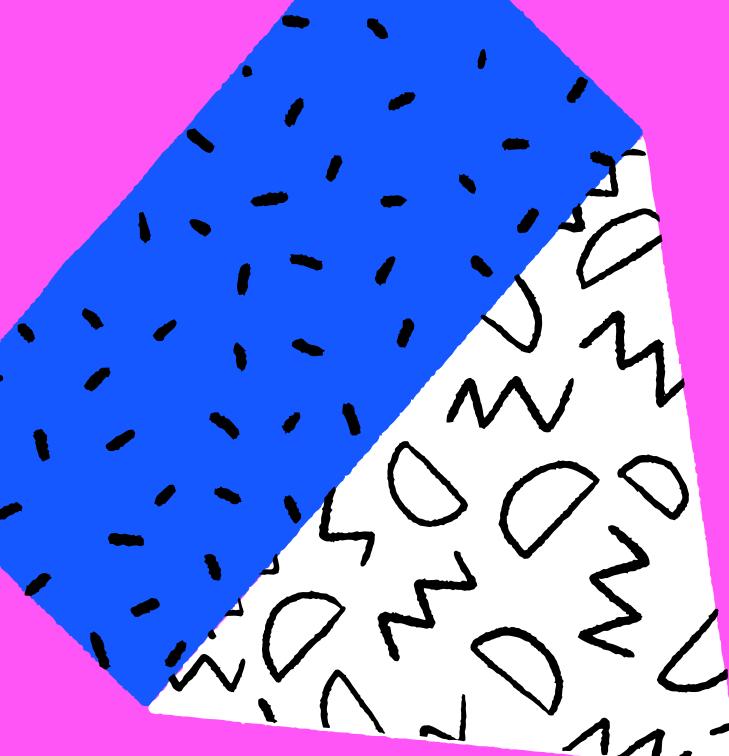
- Prototype and produce content for AR experiences that you can integrate in Xcode or export to AR Quick Look
- Great for creators with no prior 3D experience
- No code needed
- Built-in AR Library
- Live linking
- Seamless tools
- Animation and spatial audio
- Record and play
- [Available on the App Store](#)
- Available on iOS, iPadOS, macOS





# Intro to Apple's Augmented Reality Tools: Reality Converter and Reality Composer

iOS CONF SG 2021

A large blue triangle with black dashed horizontal lines is positioned in the top-left corner. Below it, a white triangle contains black wavy and zigzag patterns.

# Let's build some filters!

A large yellow triangle with black wavy and zigzag patterns is positioned in the bottom-right corner. A smaller black triangle with white wavy and zigzag patterns is attached to its top edge.

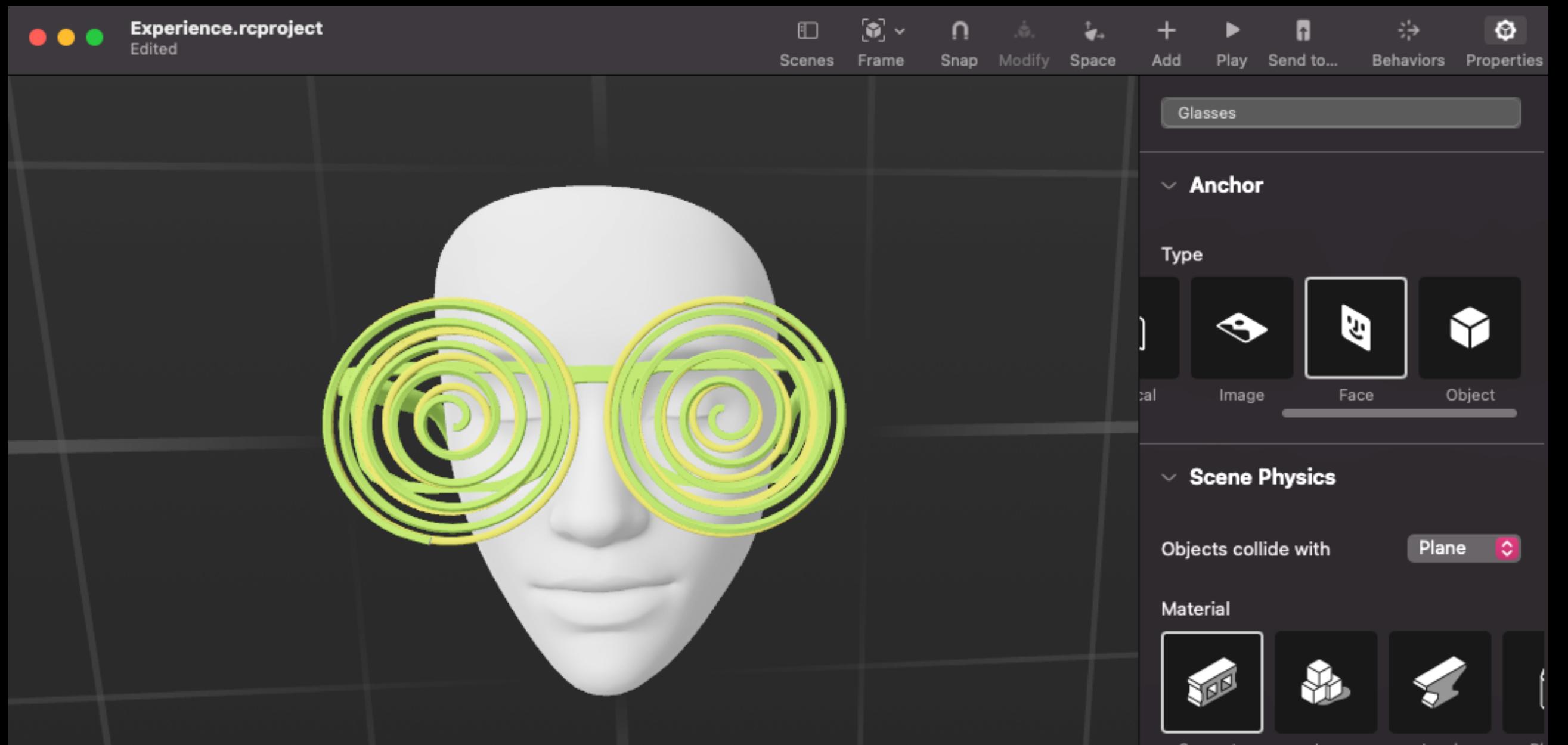
○ ○ ○ ○

○ ○ ○ ○

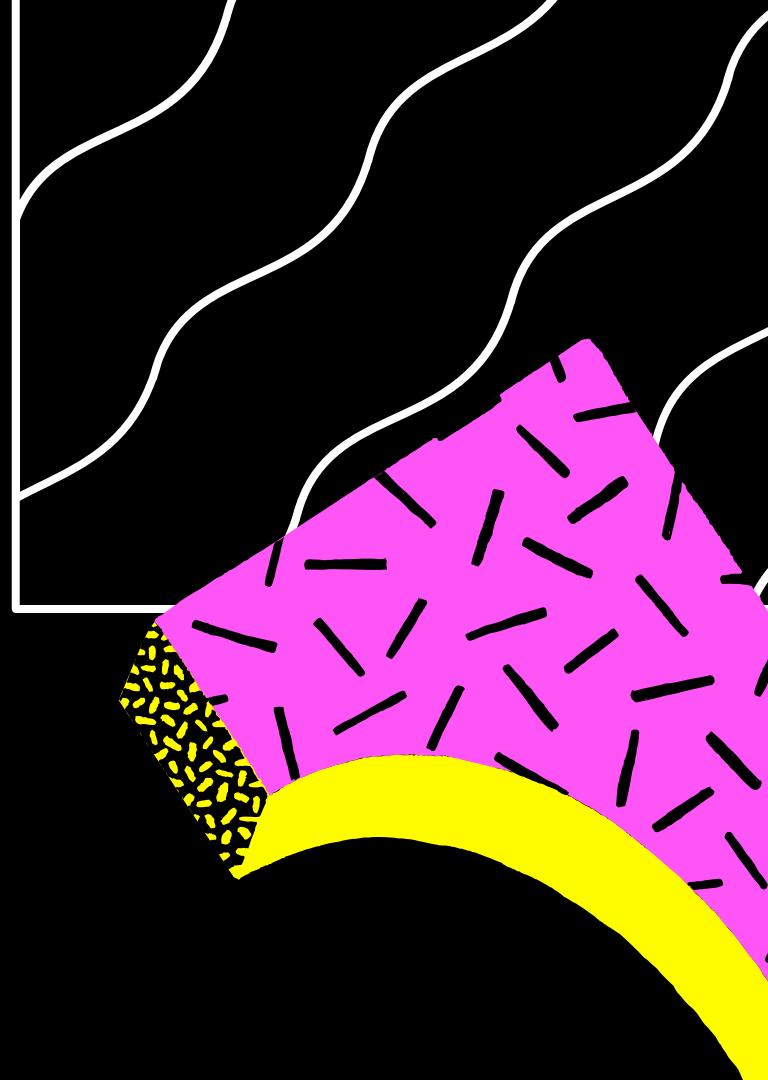
○ ○ ○ ○

# 1/AR Quick Look

and Reality Composer - Demo



Example model: [SwirlyHypno\\_Glasses](#) by [SirLiquorice](#) ([Sketchfab.com](#))



# 1/AR Quick Look

and Reality Composer - Testing it out

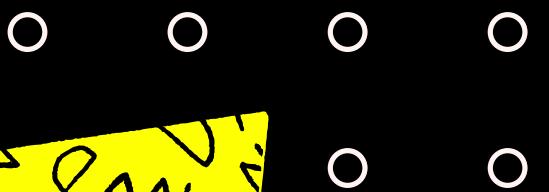
- Airdrop the USDZ model exported from Reality Composer to your device and save it to Files. Opening it from your iOS device will start the AR Quick Look experience.



# 1/AR Quick Look

and Reality Composer -Where to go from here?

- Share the experience as is. Built-in apps, such as Messages, Mail and Notes use Quick Look to display USDZ files of virtual objects in 3D or AR on iPhone and iPad.
- Integrate the USDZ file into an Xcode project
  - How to make an augmented reality decorating experience app with AR Quick Look - Section: Making the app
- Embed the USDZ file into your website
  - Easy web augmented reality with AR Quick Look



# 2/RealityKit

and Reality Composer project - Demo

```
guard ARFaceTrackingConfiguration.isSupported else { return }

let configuration = ARFaceTrackingConfiguration()
if #available(iOS 13.0, *) {
    configuration.maximumNumberOfTrackedFaces =
        ARFaceTrackingConfiguration.supportedNumberOfTrackedFaces
}
configuration.isLightEstimationEnabled = true
arView.session.run(configuration)

// Load the "Glasses" scene from the "Experience" Reality File
let glassesAnchor = try! Experience.loadGlasses()

// Add the box anchor to the scene
arView.scene.anchors.append(glassesAnchor)
```

# 2/RealityKit

and Reality Composer project - Where to go from here?

- [Face Tracking with RealityKit Course \(raywenderlich.com\) with SwiftUI](#)
- [Apple Augmented Reality by Tutorials \(raywenderlich.com\)](#)
- [Apple's RealityKit Documentation and Examples](#)
- [Reality School Youtube channel](#)

# 3/SceneKit

## Demo

```
extension ViewController: ARSCNViewDelegate {
    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
        guard let sceneView = renderer as? ARSCNView,
              anchor is ARFaceAnchor,
              let device = sceneView.device else {
            return nil
        }

        let faceGeometry = ARSCNFaceGeometry(device: device)!
        let material = faceGeometry.firstMaterial!

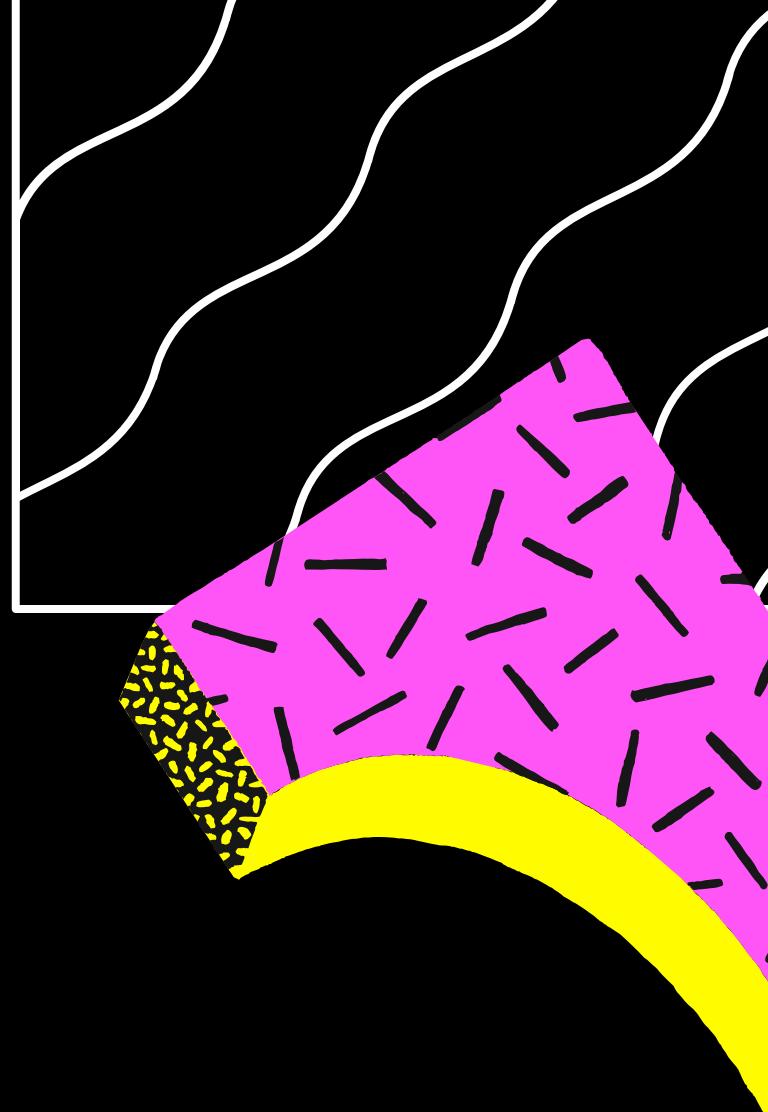
        // material.fillMode = .lines

        material.diffuse.contents = ...
        material.lightingModel = .physicallyBased

        contentNode = SCNNode(geometry: faceGeometry)
        return contentNode
    }

    /// - Tag: ARFaceGeometryUpdate
    func renderer(_ renderer: SCNSceneRenderer, didUpdate node: SCNNode, for anchor: ARAnchor) {
        guard let faceGeometry = node.geometry as? ARSCNFaceGeometry,
              let faceAnchor = anchor as? ARFaceAnchor
        else { return }

        faceGeometry.update(from: faceAnchor.geometry)
    }
}
```

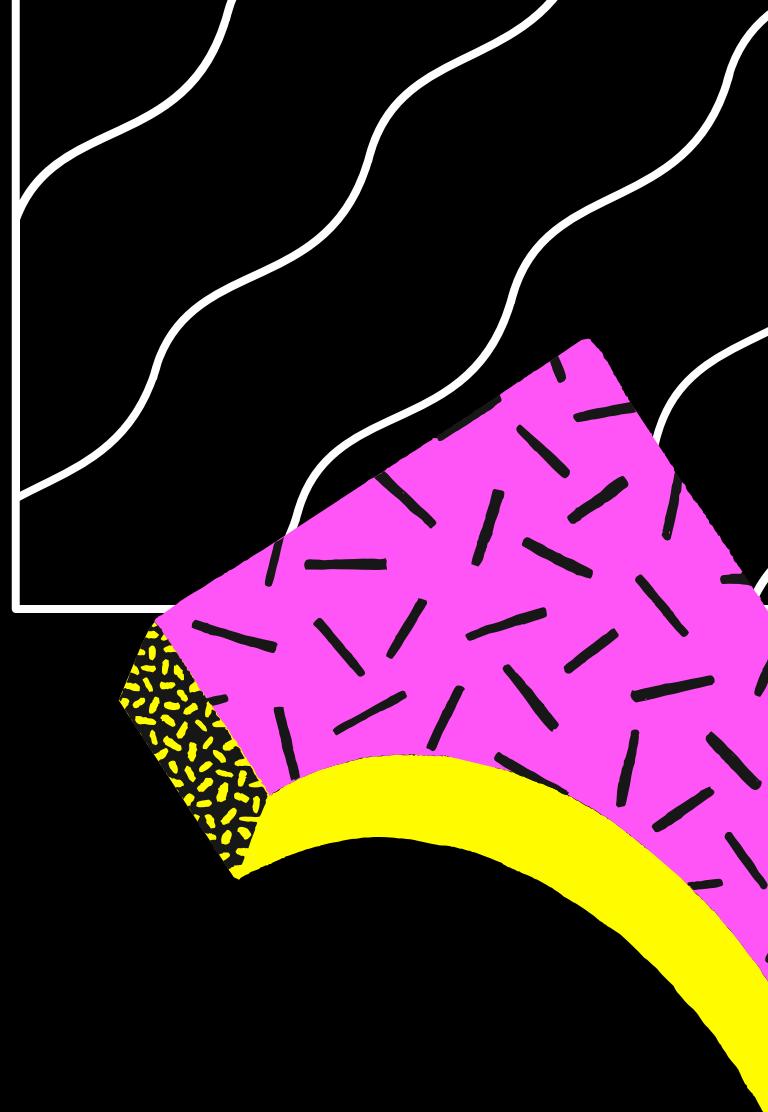


# 3/SceneKit

Testing it out



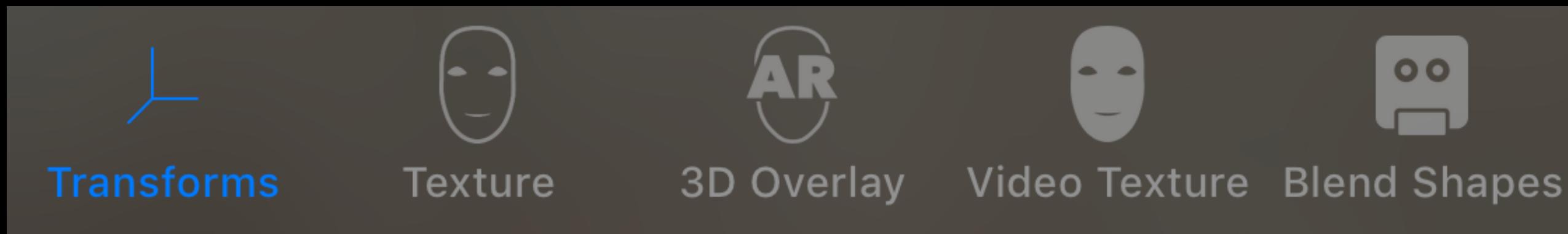
[Butterflies Assets from Freepik](#)



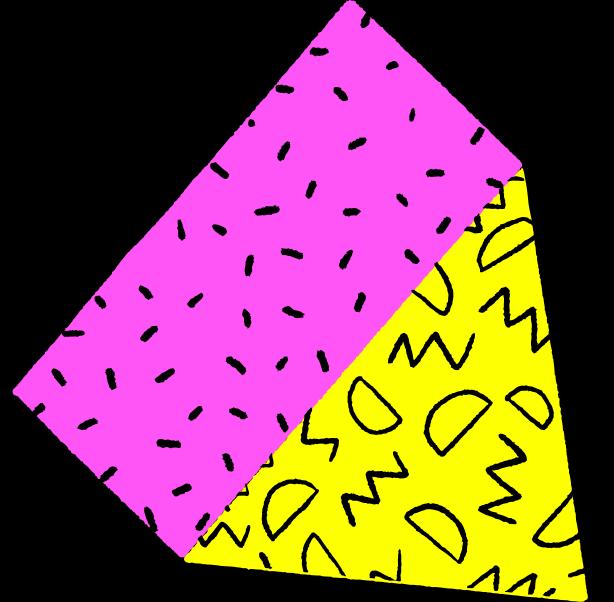
# 3/SceneKit

Where to go from here?

- [AR Face Tracking Tutorial for iOS: Getting Started \(raywenderlich.com\)](#)
- [Tracking and Visualizing Faces Sample code by Apple](#)
- [Apple's SceneKit Documentation and Examples](#)



# 3 bonus tips for your AR face filter experiences

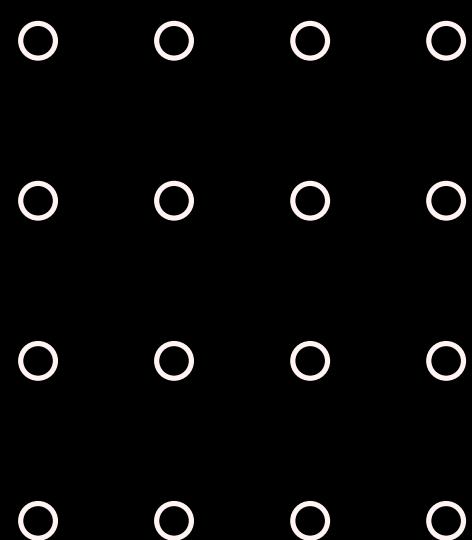


**1/Disable the idle timer** - *to prevent auto screen dimming as your users might not interact with the device for some time [viewDidAppear]*

```
UIApplication.shared.isIdleTimerDisabled = true
```

**2/Hide the home indicator**

```
override var prefersHomeIndicatorAutoHidden: Bool {  
    return true  
}
```



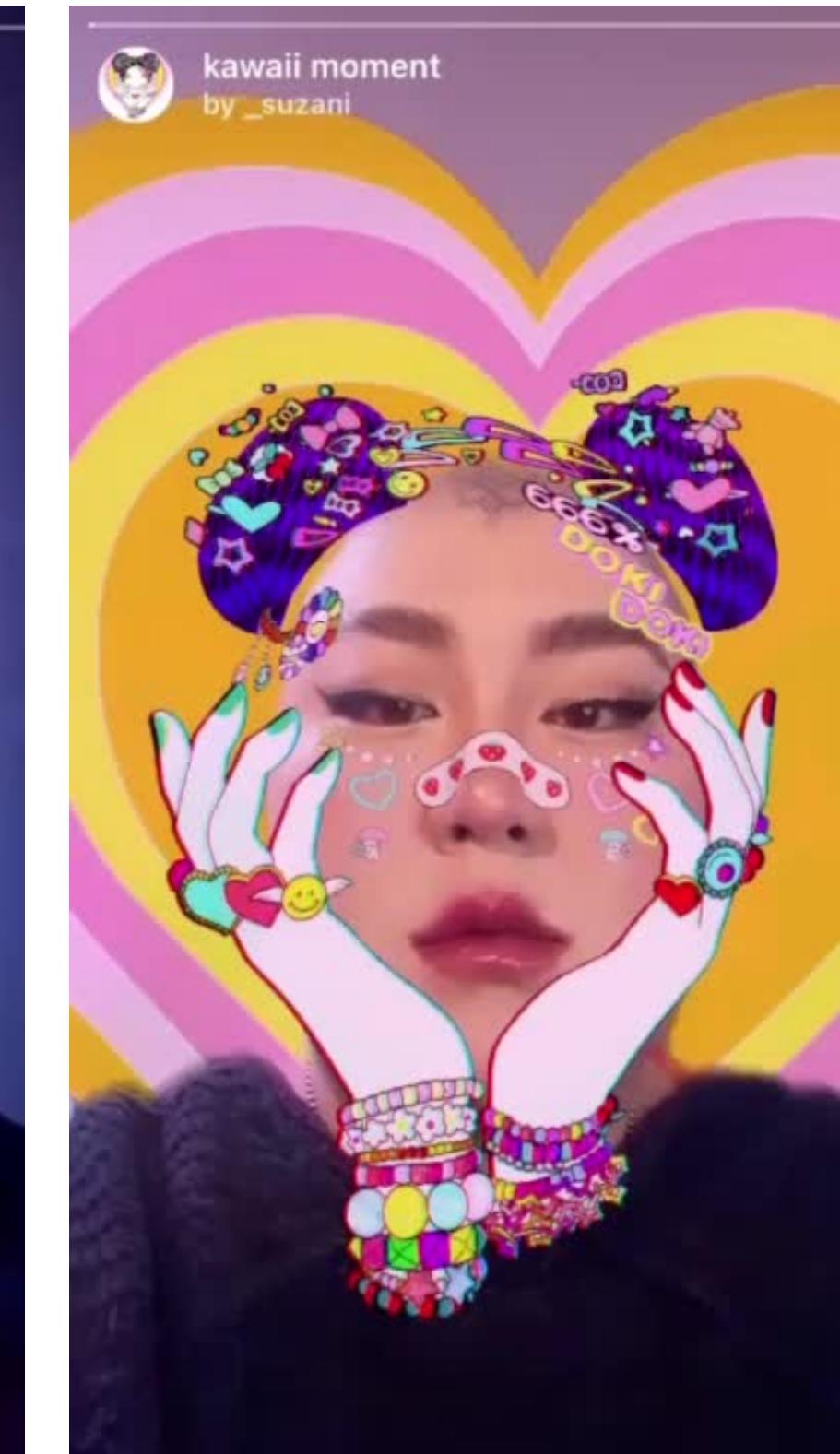
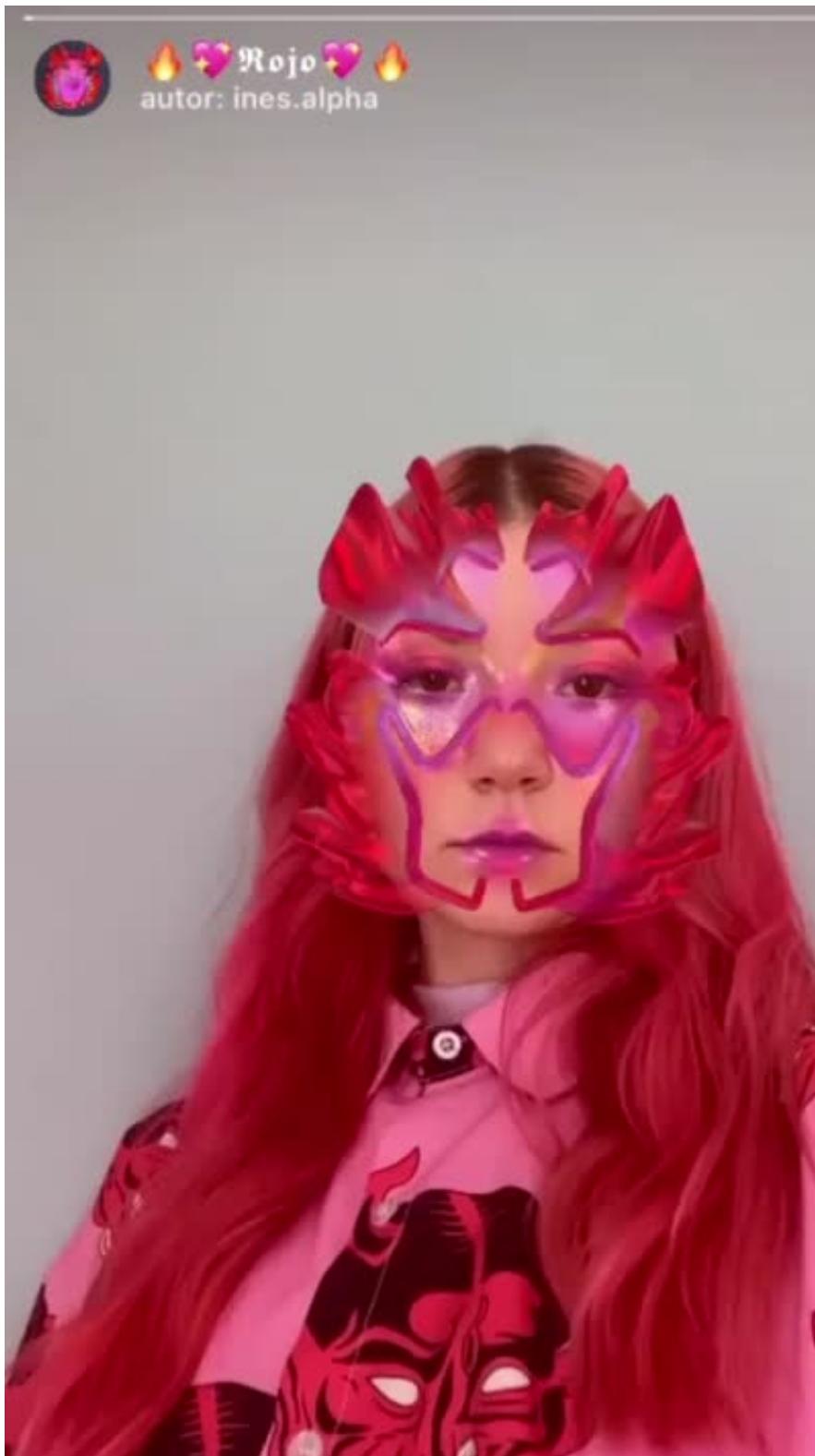
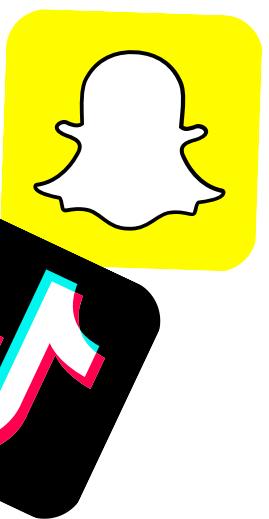
**3/Hide the status bar**

```
override var prefersStatusBarHidden: Bool {  
    return true  
}
```

# Projects available on Github



# Get inspired for your iOS AR face filters



## Community Speakers



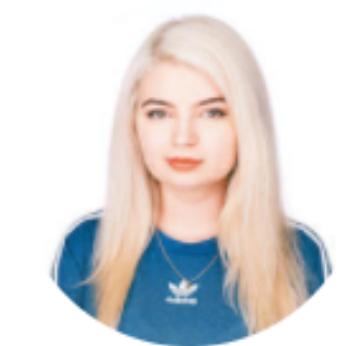
**Erica Sadun**  
Swift Author



**Daniel Steinberg**  
Author, Trainer, Consultant



**Marc Aupont**  
iOS Engineer @ Lickability



**Roxana Jula**  
Mobile Developer at  
Monstarlab



**Donny Wals**  
iOS developer, Author,  
Speaker



**Neem Serra**  
Mobile Developer @ Stitch Fix



**Craig Clayton**  
Senior iOS engineer @ Fan  
Reach



**James Dempsey**  
Developer, Trainer, Speaker,  
Musician



**Malin Sundberg**  
iOS & macOS developer



**Charlie Chapman**  
Lead iOS Engineer at Stitch Fix



try! Swift



**try! Swift World** @tryswiftworld · May 14

We're excited to announce try! Swift DUB DUB, sponsored by [@DuckDuckGo](#) on June 11th at 1pm EST. Discuss the new announcements w/ other [#iOSDevs](#) around the world, hear community lightning talks ⚡, and see amazing professional art performances 🎨💻

[tryswift.co/dub-dub](http://tryswift.co/dub-dub)

...



# THANK YOU!

ANY QUESTIONS?

You can find me at  
**@coderox** (Instagram)  
**@roxanajula** (Twitter)

