Augmented Reality

for iOS

Henrique Velloso

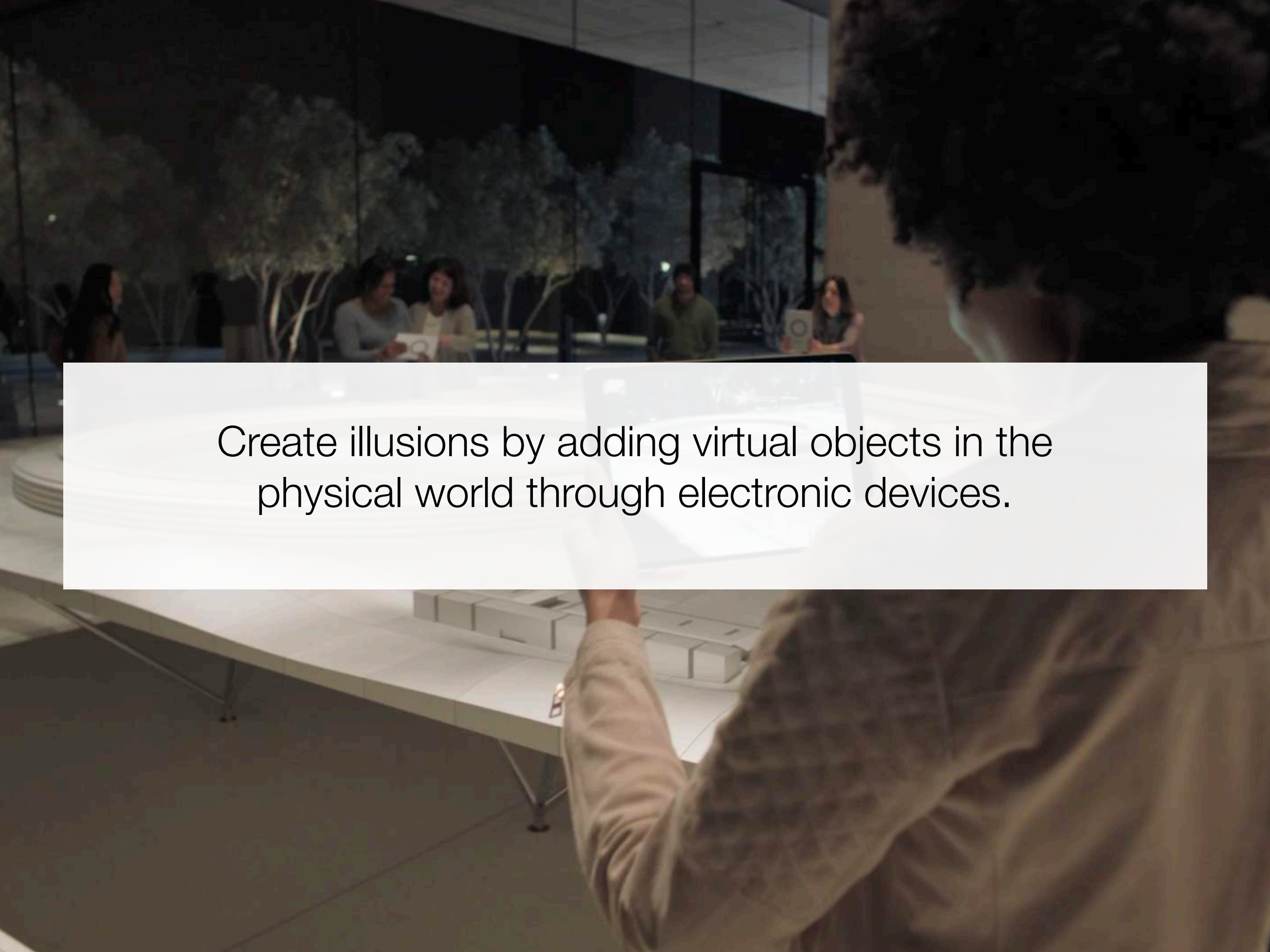
Developer iOS

- More than 10 years of experience in software development
- Since 2015 focused on iOS applications
- WWDC scholarship in 2015, 2017 and 2018
- Apple Developer Academy (BEPiD 2015)
- 9 Microsoft certifications in software development area

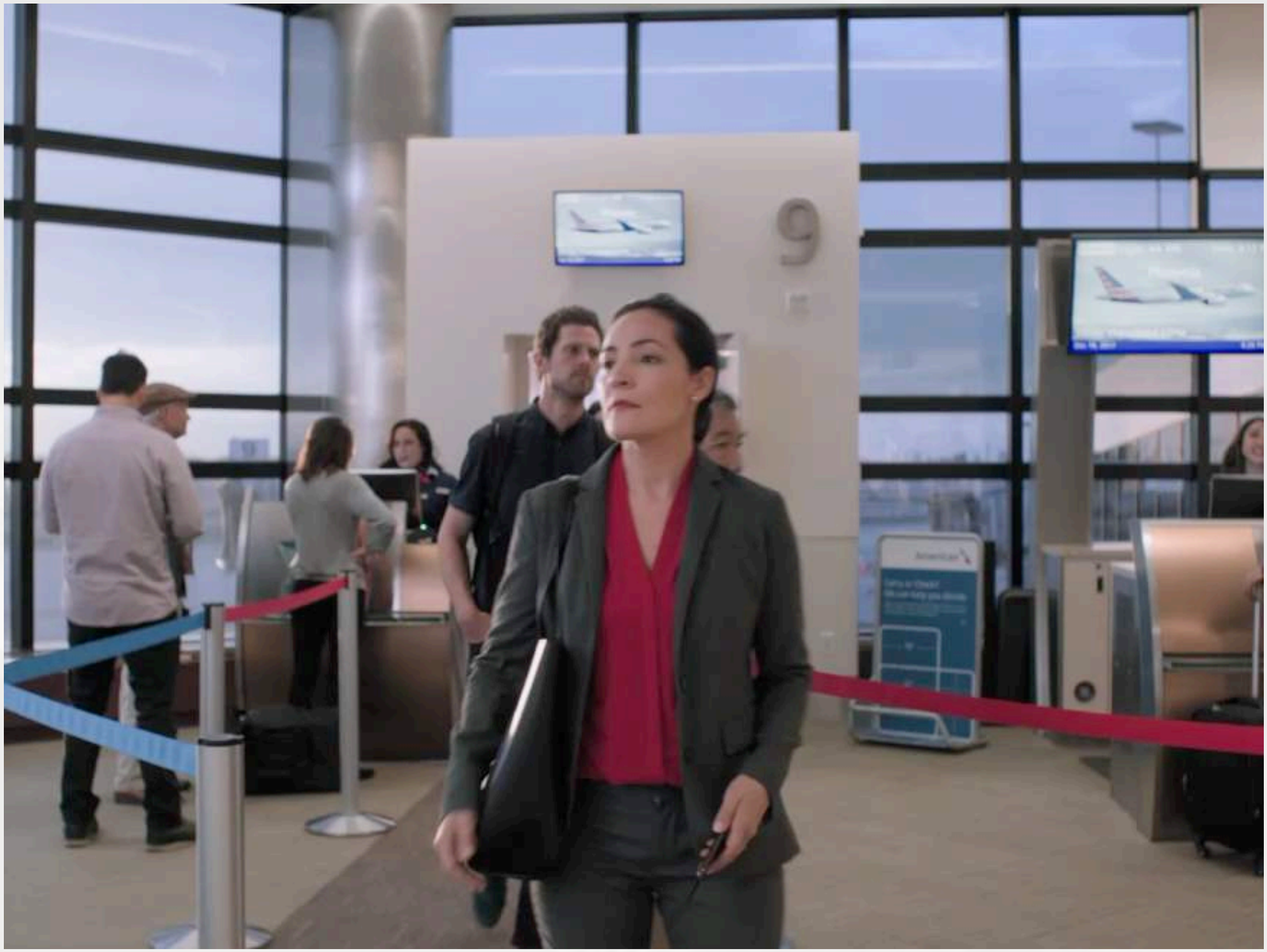
✦ velloso.io

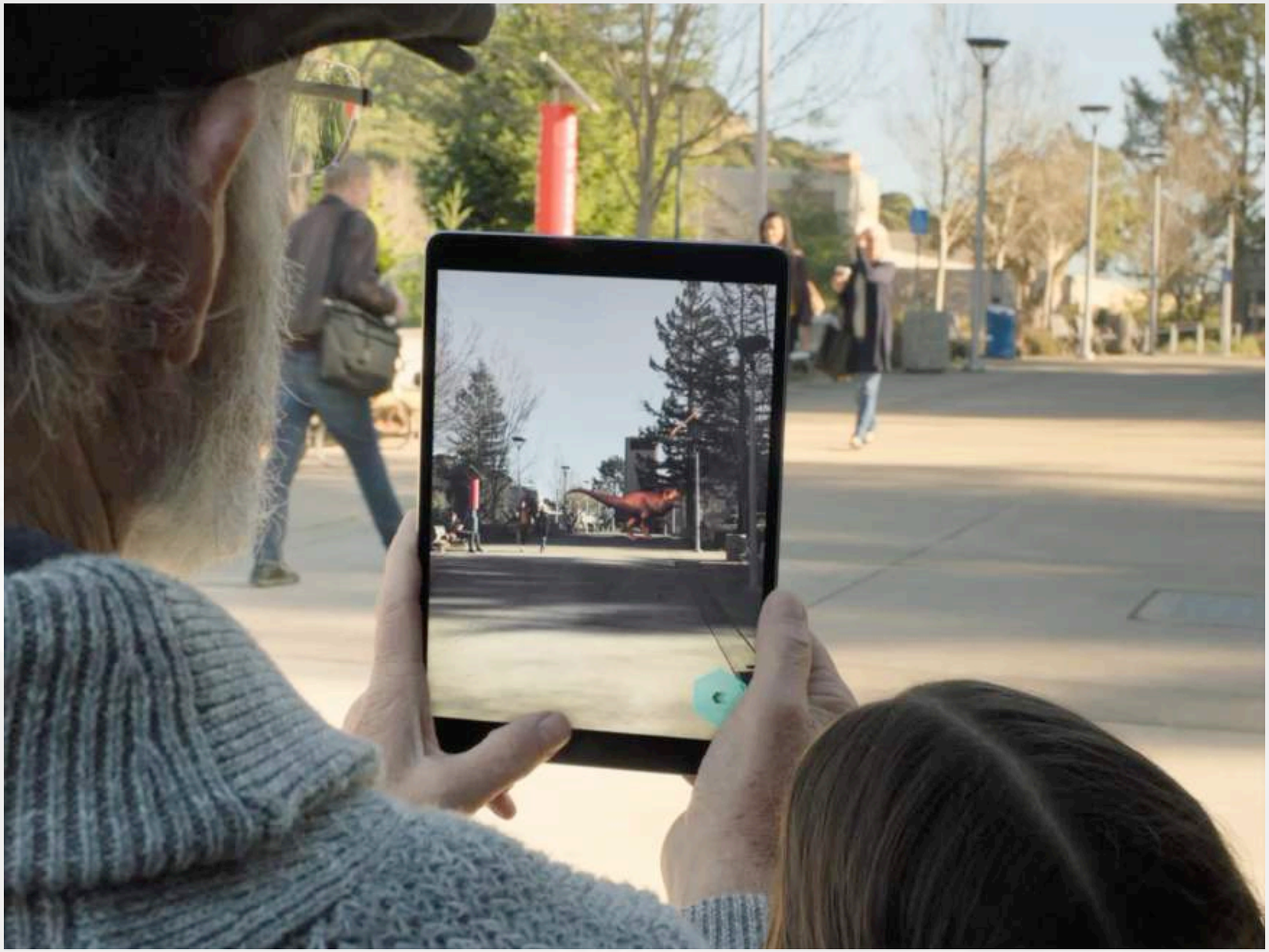
Augmented Reality

For iOS



Create illusions by adding virtual objects in the physical world through electronic devices.


9



Computer Vision

Triangulation

SLAM

Sensor Fusion

Scene Understanding

Camera Calibration

Visual-inertial Navigation

Optimal Correction

Augmented Reality

Feature Matching

Surface Estimation

Nonlinear Optimization

Camera Intrinsics

Light Estimation

Bundle Adjustment

Feature Detection

Apple ARKit

Mobile AR platform

High-level API

iOS (A9 and up)

Tracking

Scene Understanding

Rendering

Tracking

World tracking

Visual inertial odometry

No external setup

Tracking

Scene Understanding

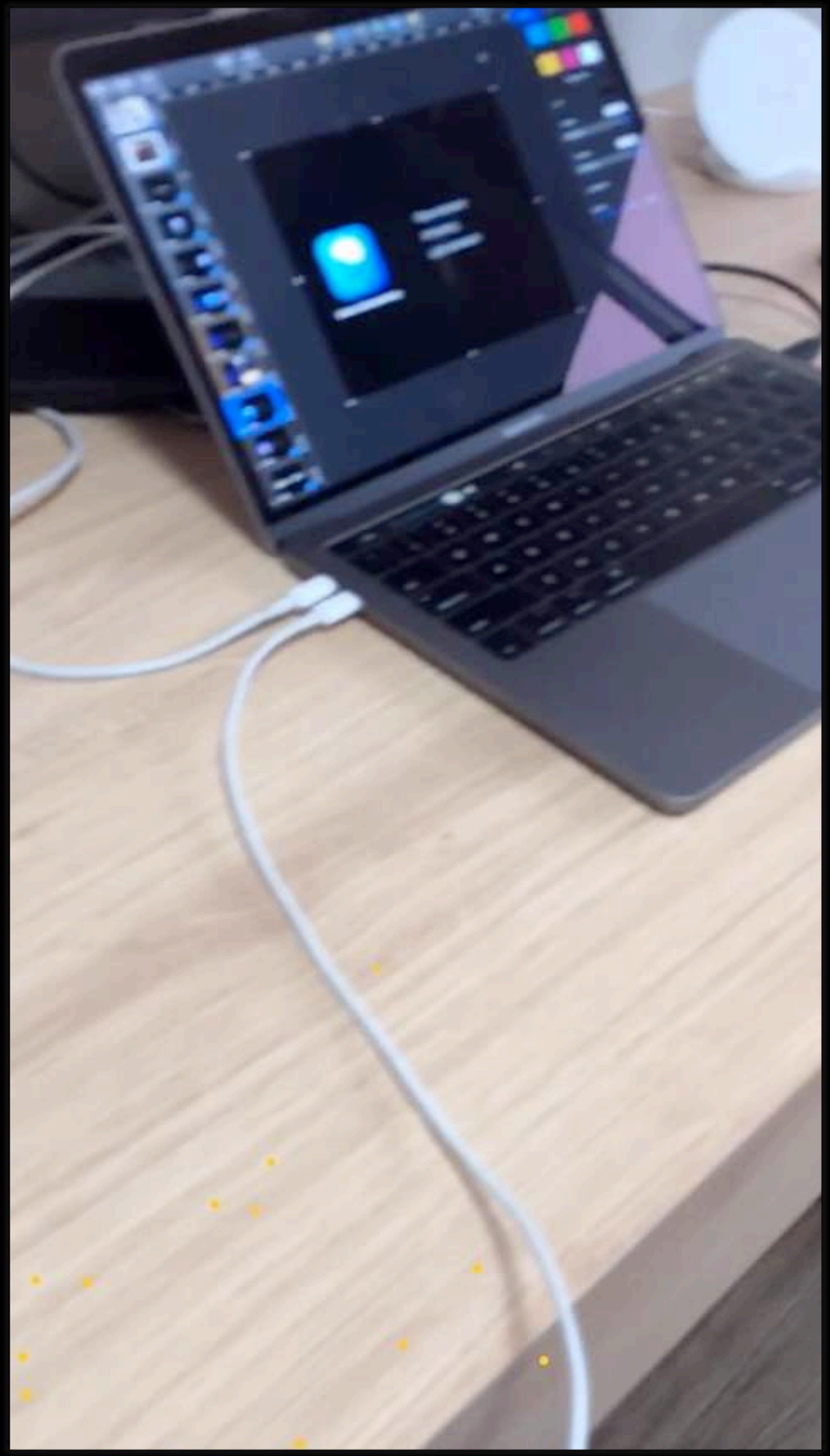
Plane detection

Hit-testing

Light estimation

Face detection

Scene Understanding



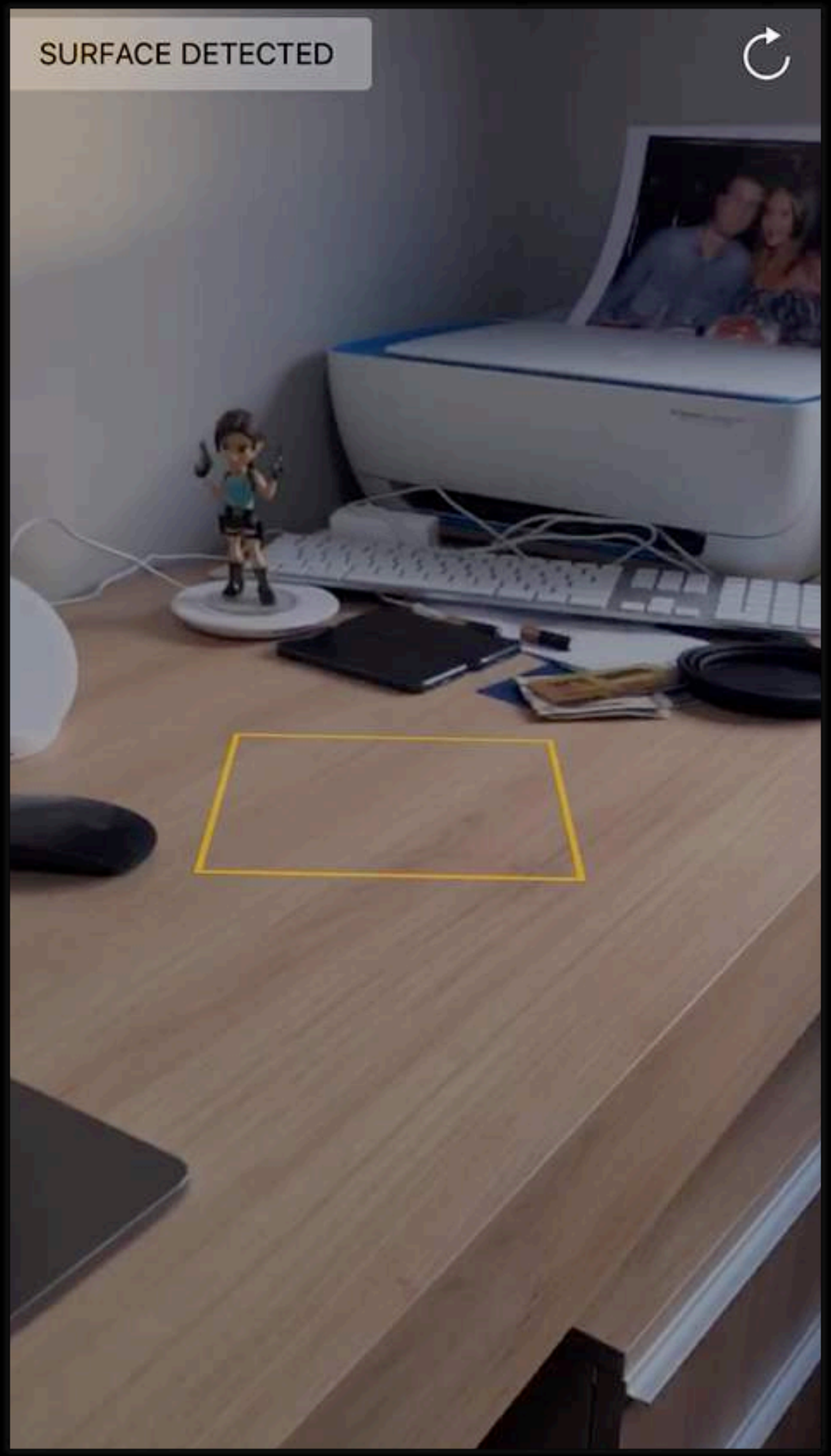
Rendering

Easy integration

AR views

Custom rendering

Rendering



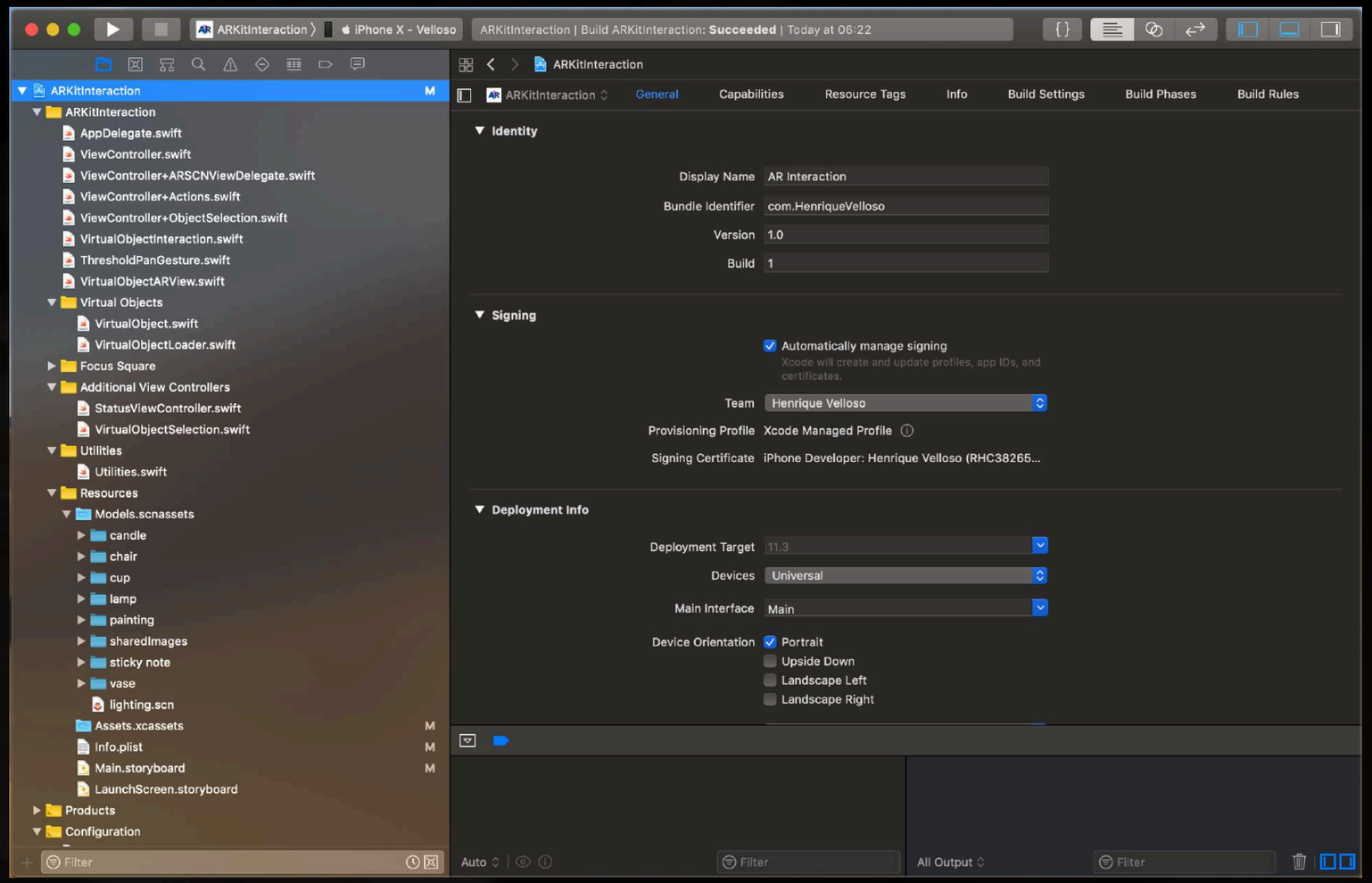

unity


UNREAL
ENGINE

Project


XCODE

Xcode 10


ARKitInteraction 〉 iPhone X - Velloso
ARKitInteraction | Build ARKitInteraction: Succeeded | Today at 06:22
ARKitInteraction
ARKitInteraction M
ARKitInteraction
AppDelegate.swift
ViewController.swift
ViewController+ARSCNViewDelegate.swift
ViewController+Actions.swift
ViewController+ObjectSelection.swift
VirtualObjectInteraction.swift
ThresholdPanGesture.swift
VirtualObjectARView.swift
Virtual Objects
VirtualObject.swift
VirtualObjectLoader.swift
Focus Square
Additional View Controllers
StatusViewController.swift
VirtualObjectSelection.swift
Utilities
Utilities.swift
Resources
Models.scnassets
candle
chair
cup
lamp
painting
sharedImages
sticky note
vase
lighting.scn
Assets.xcassets M
Info.plist M
Main.storyboard M
LaunchScreen.storyboard
Products
Configuration
Filter
ARKitInteraction
ARKitInteraction
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name AR Interaction
Bundle Identifier com.HenriqueVelloso
Version 1.0
Build 1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team Henrique Velloso
Provisioning Profile Xcode Managed Profile
Signing Certificate iPhone Developer: Henrique Velloso (RHC38265...
Deployment Info
Deployment Target 11.3
Devices Universal
Main Interface Main
Device Orientation Portrait
Upside Down
Landscape Left
Landscape Right
Auto
Filter
Filter
All Output
Filter

ARKitInteraction 〉 iPhone X - Velloso | ARKitInteraction | Build ARKitInteraction: **Succeeded** | Today at 06:22

ARKitInteraction M
- ARKitInteraction
  - AppDelegate.swift
  - ViewController.swift
  - ViewController+ARSCNViewDelegate.swift
  - ViewController+Actions.swift
  - ViewController+ObjectSelection.swift
  - VirtualObjectInteraction.swift
  - ThresholdPanGesture.swift
  - VirtualObjectARView.swift
  - Virtual Objects
    - VirtualObject.swift
    - VirtualObjectLoader.swift
  - Focus Square
  - Additional View Controllers
    - StatusViewController.swift
    - VirtualObjectSelection.swift
  - Utilities
    - Utilities.swift
  - Resources
    - Models.scnassets
      - candle
      - chair
      - cup
      - lamp
      - painting
      - sharedImages
      - sticky note
      - vase
      - lighting.scn
    - Assets.xcassets M
    - Info.plist M
    - Main.storyboard M
    - LaunchScreen.storyboard
  - Products
  - Configuration

Filter

ARKitInteraction

ARKitInteraction | General | Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

Identity

Display Name: AR Interaction
Bundle Identifier: com.HenriqueVelloso
Version: 1.0
Build: 1

Signing

Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Henrique Velloso
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iPhone Developer: Henrique Velloso (RHC38265...

Deployment Info

Deployment Target: 11.3
Devices: Universal
Main Interface: Main
Device Orientation: Portrait, Upside Down, Landscape Left, Landscape Right

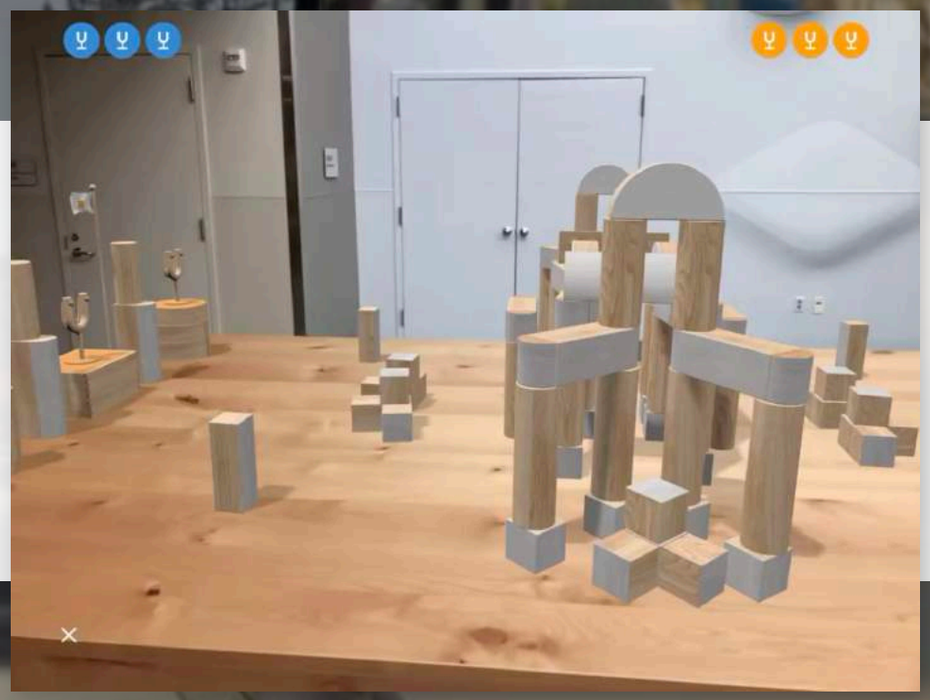
Auto | Filter | All Output | Filter

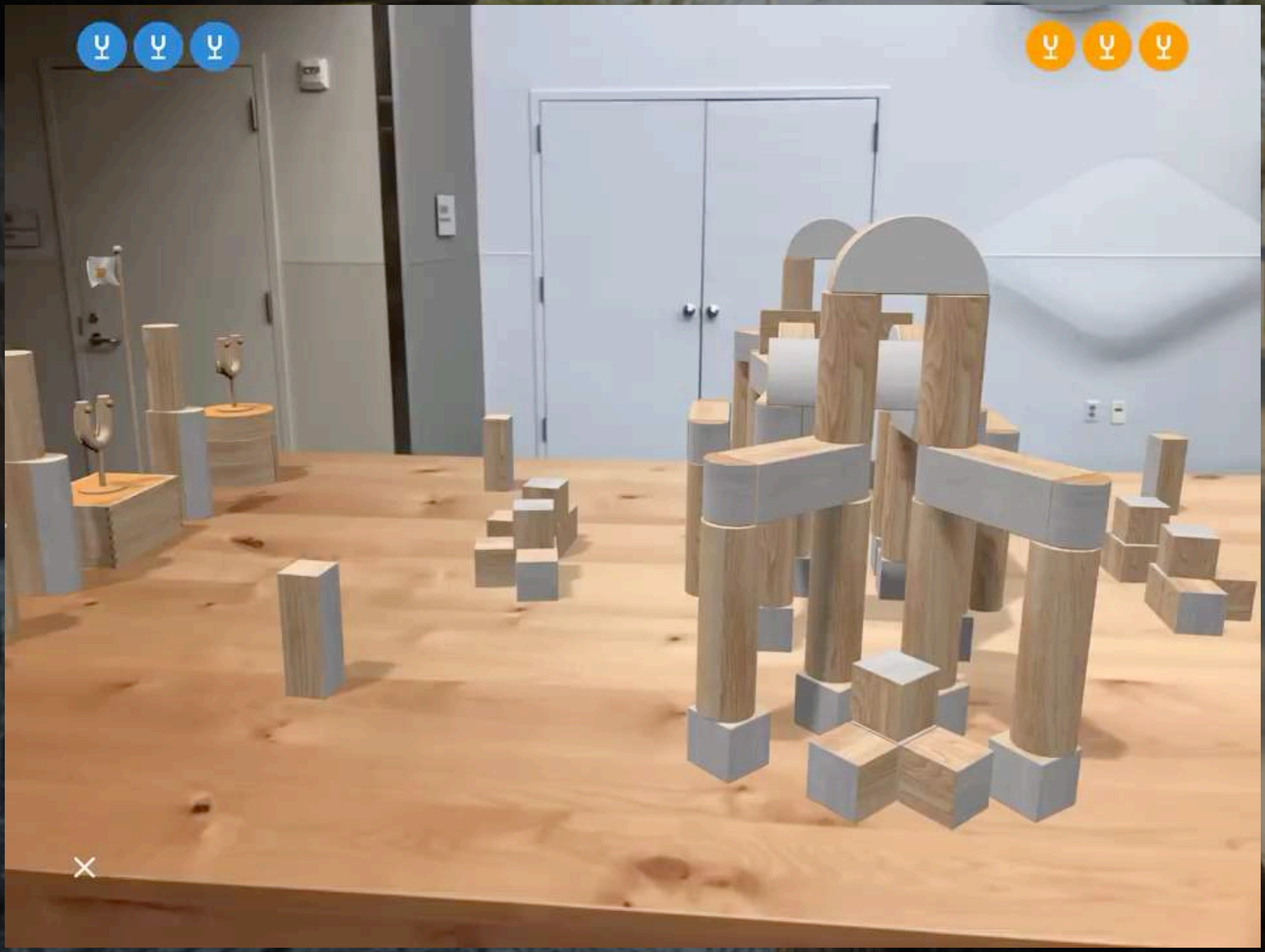
```swift
override func viewDidLoad() {
    super.viewDidLoad()

    // Set the view's delegate
    sceneView.delegate = self

    // Show statistics such as fps and timing information
    sceneView.showsStatistics = true

    // Create a new scene
    let scene = SCNScene(named: "art.scnassets/ship.scn")!

    // Set the scene to the view
    sceneView.scene = scene


    let tapGesture = UITapGestureRecognizer(target: self, action:
        #selector(ViewController.handleTap(gestureRecoginze:)))
    view.addGestureRecognizer(tapGesture)
}
```

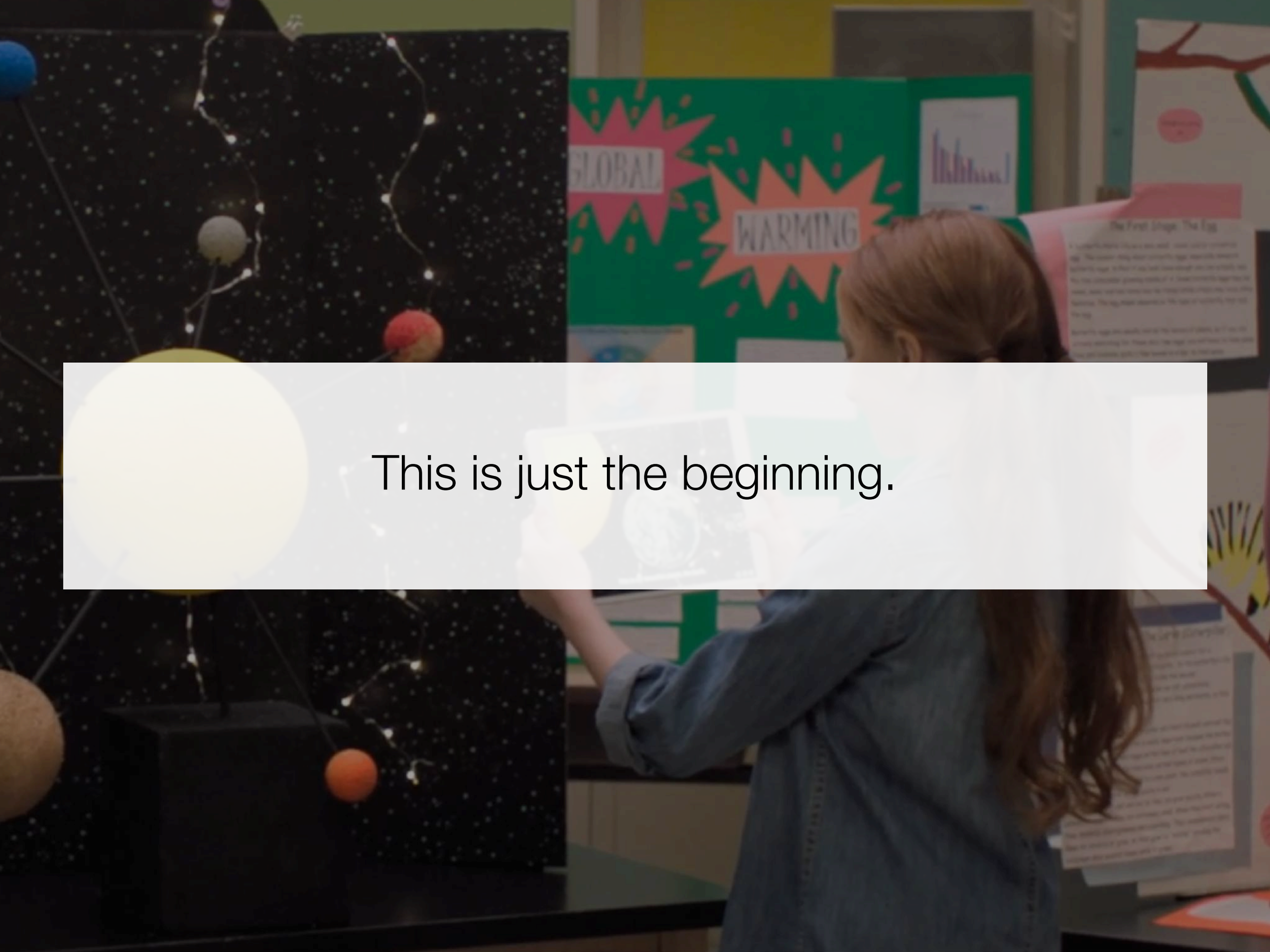
```swift
@objc
func handleTap(gestureRecoginze: UITapGestureRecognizer) {

    guard let currentFrame = sceneView.session.currentFrame else {
        return
    }

    let imagePlane = SCNPlane(width: sceneView.bounds.width / 6000,
                              height: sceneView.bounds.height / 6000)
    imagePlane.firstMaterial?.diffuse.contents = sceneView.snapshot()
    imagePlane.firstMaterial?.lightingModel = .constant

    let planeNode = SCNNode(geometry: imagePlane)
    let rootNode = sceneView.scene.rootNode
    rootNode.addChildNode(planeNode)

    var translation = matrix_identity_float4x4
    translation.columns.3.z = -0.1
    planeNode.simdTransform = matrix_multiply(currentFrame.camera.transform, translation)

}
```

Demo

Plus content







This is just the beginning.