SE20UARI101 PHANI SEKHAR REDDY
SE20UARI159 VEDA VYAS

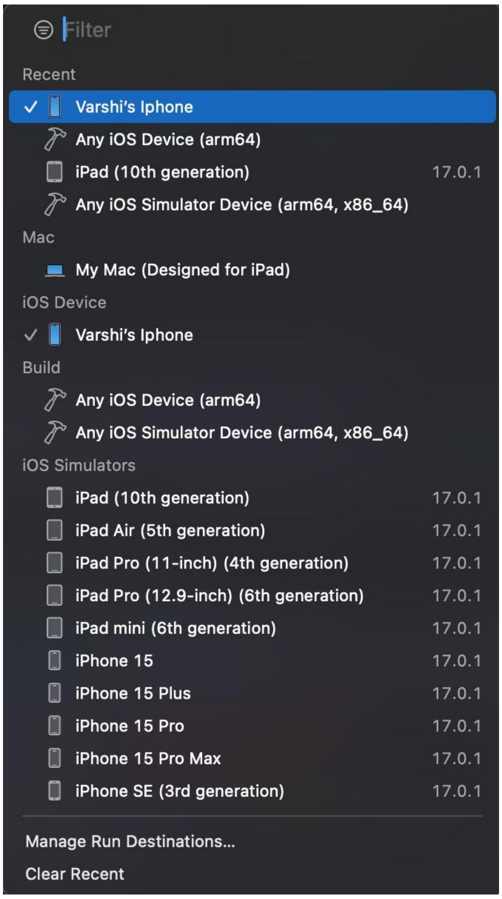
ROOM PLAN API USING AUGMENTED REALITY

INTRODUCTION

- Room Plan is powered by AR that uses the camera and LiDAR Scanner on iPhone and iPad to produce a 3D floor plan of space, including importantdetails like size and furniture types.
- These scans can also be the first step in architecture and interior design workflows tohelp streamline conceptual exploration and planning.
- Room Plan outputs in USD or USDZ file formats that include dimensions of each component recognized in the room, such as walls or cabinets, as well as the type of furniture detected. The dimensions and placement of each individual component can befurther adjusted when exported into various USDZ-compatible tools, such as Cinema 4D, Sharp3D, or AutoCAD.

IMPLEMENTATION OF THE PROJECT

1. **The thing is this app requires a Mac and iPhone logged in with the same Apple ID.**
2. First download the Xcode to your MAC. Make sure that the software version matches the Xcode version. We can also download the Xcode beta version if available to get a more interactive experience if either doesn't work.
3. First connect your mobile to Mac using cable and then Set the run destination to a devicerunning iOS 16 or later with a LiDAR Scanner.


Filter
Recent
✓ Varshi's Iphone
Any iOS Device (arm64)
iPad (10th generation) 17.0.1
Any iOS Simulator Device (arm64, x86_64)
Mac
My Mac (Designed for iPad)
iOS Device
✓ Varshi's Iphone
Build
Any iOS Device (arm64)
Any iOS Simulator Device (arm64, x86_64)
iOS Simulators
iPad (10th generation) 17.0.1
iPad Air (5th generation) 17.0.1
iPad Pro (11-inch) (4th generation) 17.0.1
iPad Pro (12.9-inch) (6th generation) 17.0.1
iPad mini (6th generation) 17.0.1
iPhone 15 17.0.1
iPhone 15 Plus 17.0.1
iPhone 15 Pro 17.0.1
iPhone 15 Pro Max 17.0.1
iPhone SE (3rd generation) 17.0.1
Manage Run Destinations...
Clear Recent

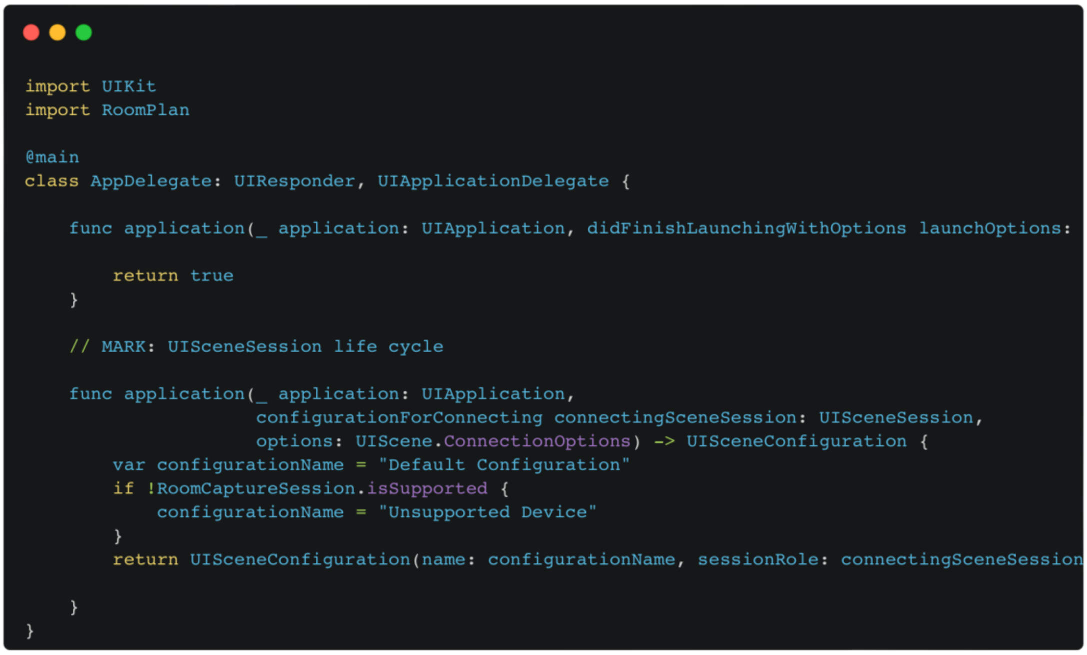
4. Go into Signing and capabilities and set up your team if you don’t have any add an account. Makesure to turn on developer mode in your mobile settings.

Then you are all set to run the application on your mobile.

DEVELOPMENT OF THE PROJECT

1. APP DELEGATE

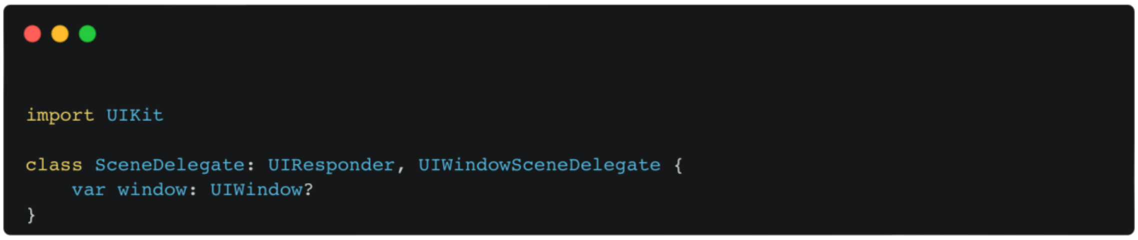
App delegate helps us to interact with the system to determine app behaviors to different events throughout the system.



```
import UIKit
import RoomPlan

@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:

        return true
    }

    // MARK: UISceneSession life cycle

    func application(_ application: UIApplication,
                     configurationForConnecting connectingSceneSession: UISceneSession,
                     options: UIScene.ConnectionOptions) -> UISceneConfiguration {
        var configurationName = "Default Configuration"
        if !RoomCaptureSession.isSupported {
            configurationName = "Unsupported Device"
        }
        return UISceneConfiguration(name: configurationName, sessionRole: connectingSceneSession

    }
}
```

2. SCENE DELEGATE

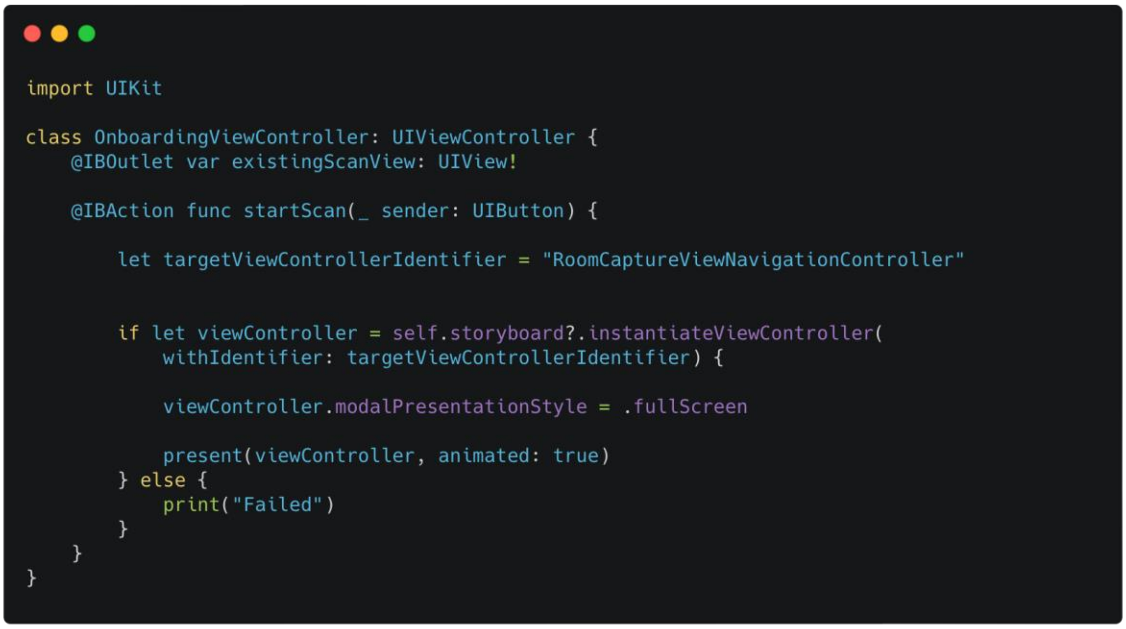
It helps us to show the application in the way we want, and we can also make changes to the code to change the visibility



```
import UIKit

class SceneDelegate: UIResponder, UIWindowSceneDelegate {
    var window: UIWindow?
}
```

3. ONBOARDING VIEW CONTROLLER

It helps us to give the best graphic user interface experience to the user.It is the experience that helps to get started with the application.



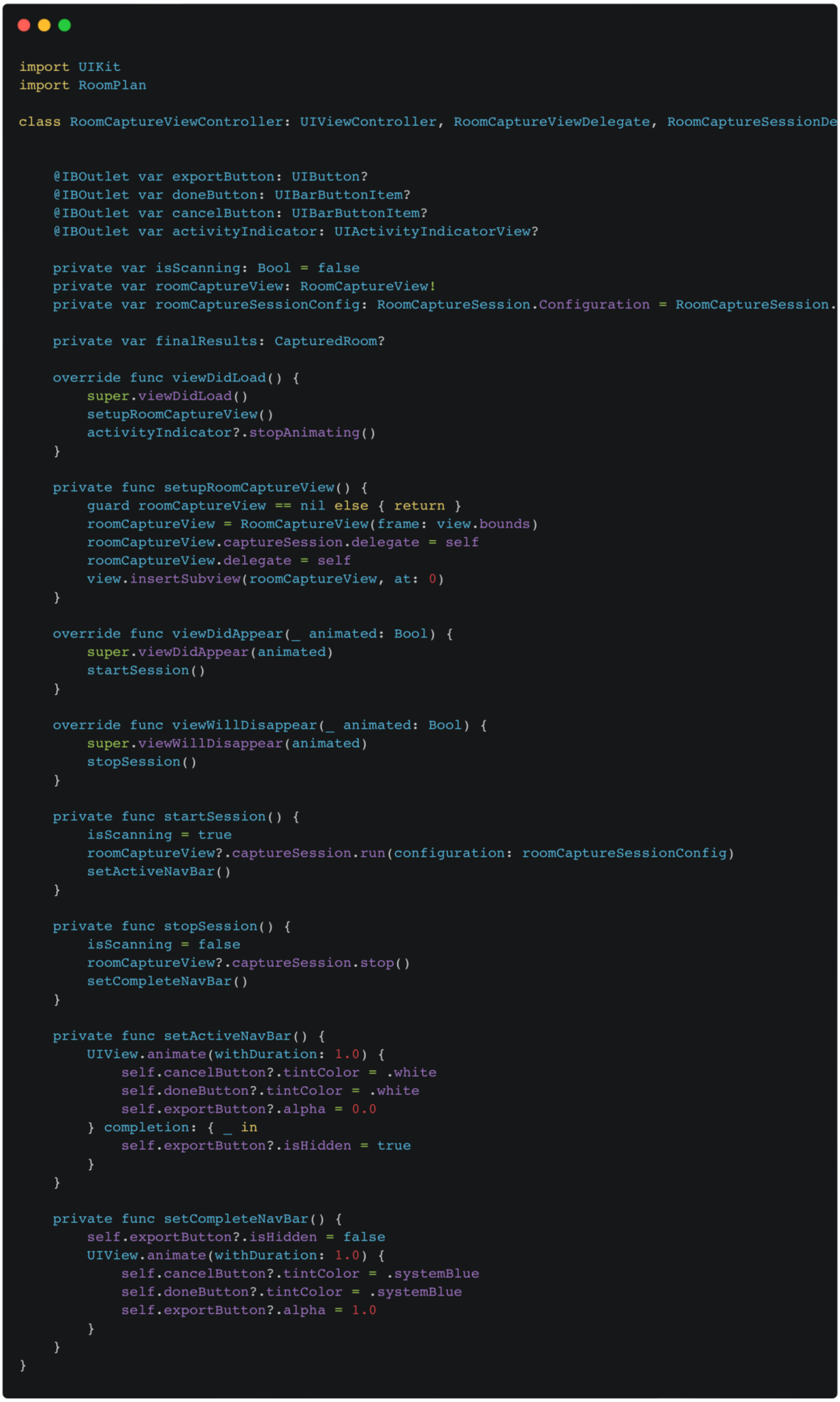
```
import UIKit

class OnboardingViewController: UIViewController {
    @IBOutlet var existingScanView: UIView!

    @IBAction func startScan(_ sender: UIButton) {

        let targetViewControllerIdentifier = "RoomCaptureViewNavigationController"


        if let viewController = self.storyboard?.instantiateViewController(
            withIdentifier: targetViewControllerIdentifier) {

            viewController.modalPresentationStyle = .fullScreen

            present(viewController, animated: true)
        } else {
            print("Failed")
        }
    }
}
```

4. ROOM CAPTURE VIEW CONTROLLER

A view that enables the user to scan their room with the device’s camera.

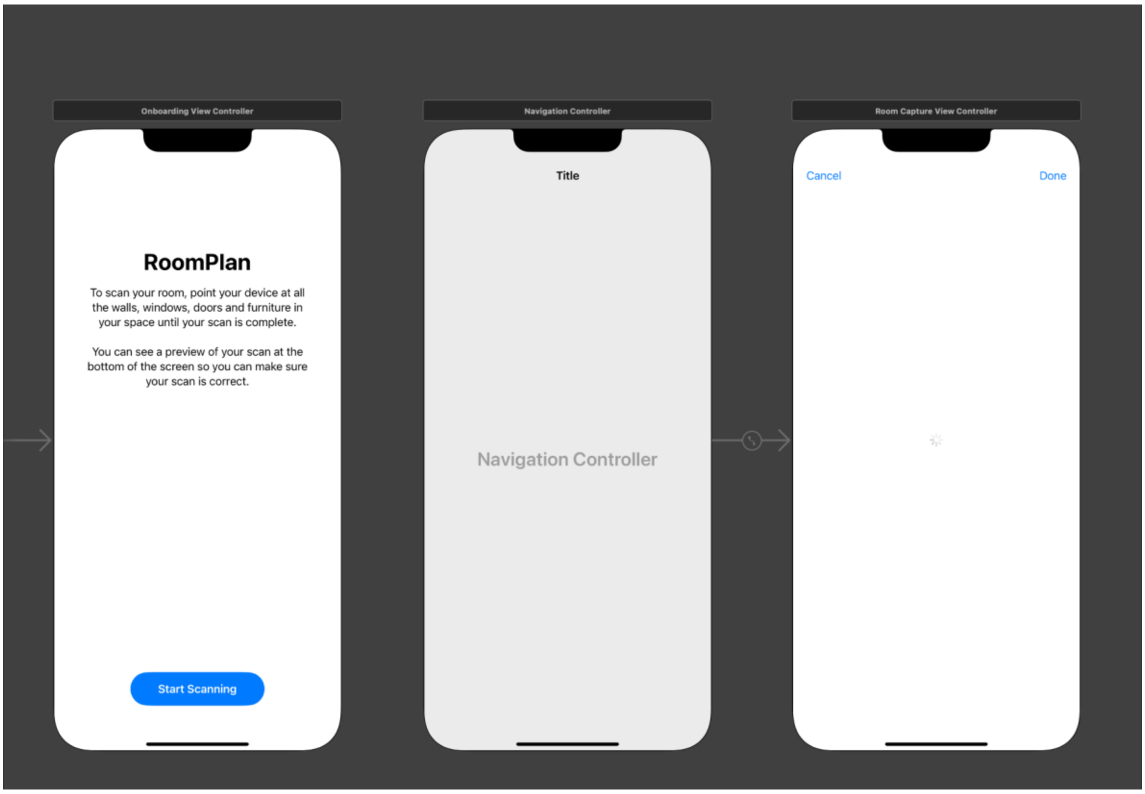
This class gives your app a view that oversees the scanning process from beginning to end, including:

- A camera stream that users can view in AR to see their room.
- Graphic overlays that appear in real time on top of physical structures in the room to show scanning progress.
- If the framework requires a specific type of device movement or perspective to finish the capture, the user instructions will clarify how to place the device.
- When the app deems that the current scan is finished, the view presents a scaled-down version of the scanned room for the user to approve.



```swift
import UIKit
import RoomPlan

class RoomCaptureViewController: UIViewController, RoomCaptureViewDelegate, RoomCaptureSessionDe


    @IBOutlet var exportButton: UIButton?
    @IBOutlet var doneButton: UIBarButtonItem?
    @IBOutlet var cancelButton: UIBarButtonItem?
    @IBOutlet var activityIndicator: UIActivityIndicatorView?

    private var isScanning: Bool = false
    private var roomCaptureView: RoomCaptureView!
    private var roomCaptureSessionConfig: RoomCaptureSession.Configuration = RoomCaptureSession.

    private var finalResults: CapturedRoom?

    override func viewDidLoad() {
        super.viewDidLoad()
        setupRoomCaptureView()
        activityIndicator?.stopAnimating()
    }

    private func setupRoomCaptureView() {
        guard roomCaptureView == nil else { return }
        roomCaptureView = RoomCaptureView(frame: view.bounds)
        roomCaptureView.captureSession.delegate = self
        roomCaptureView.delegate = self
        view.insertSubview(roomCaptureView, at: 0)
    }

    override func viewDidAppear(_ animated: Bool) {
        super.viewDidAppear(animated)
        startSession()
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        stopSession()
    }

    private func startSession() {
        isScanning = true
        roomCaptureView?.captureSession.run(configuration: roomCaptureSessionConfig)
        setActiveNavBar()
    }

    private func stopSession() {
        isScanning = false
        roomCaptureView?.captureSession.stop()
        setCompleteNavBar()
    }

    private func setActiveNavBar() {
        UIView.animate(withDuration: 1.0) {
            self.cancelButton?.tintColor = .white
            self.doneButton?.tintColor = .white
            self.exportButton?.alpha = 0.0
        } completion: { _ in
            self.exportButton?.isHidden = true
        }
    }

    private func setCompleteNavBar() {
        self.exportButton?.isHidden = false
        UIView.animate(withDuration: 1.0) {
            self.cancelButton?.tintColor = .systemBlue
            self.doneButton?.tintColor = .systemBlue
            self.exportButton?.alpha = 1.0
        }
    }
}
```

MAIN STORY BOARD

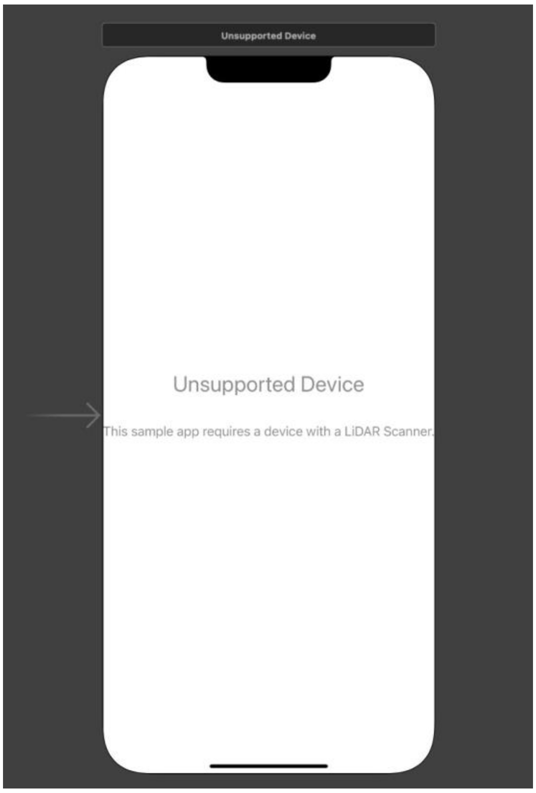


RoomPlan

To scan your room, point your device at all the walls, windows, doors and furniture in your space until your scan is complete.

You can see a preview of your scan at the bottom of the screen so you can make sure your scan is correct.

UNSUPPORTED DEVICE

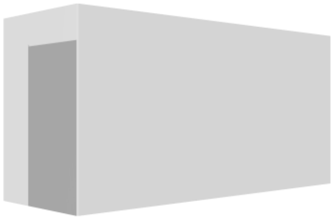


Unsupported Device

This sample app requires a device with a LiDAR Scanner.

OUTCOME

Using this application, we scanned the Study room of library & Library entrance.



We just attached the outcome obtained i.e. USDZ file. We actually got the 3D view object and AR view. I am attaching the files for more 3D visualization.

- This application is useful for Interior designing.
- We can edit the USDZ file using REALITY COMPOSER PRO, whereas that can be used for Architecture workflow.

You can launch Reality Composer Pro directly by choosing Xcode>Open developer tool>Reality Composer pro