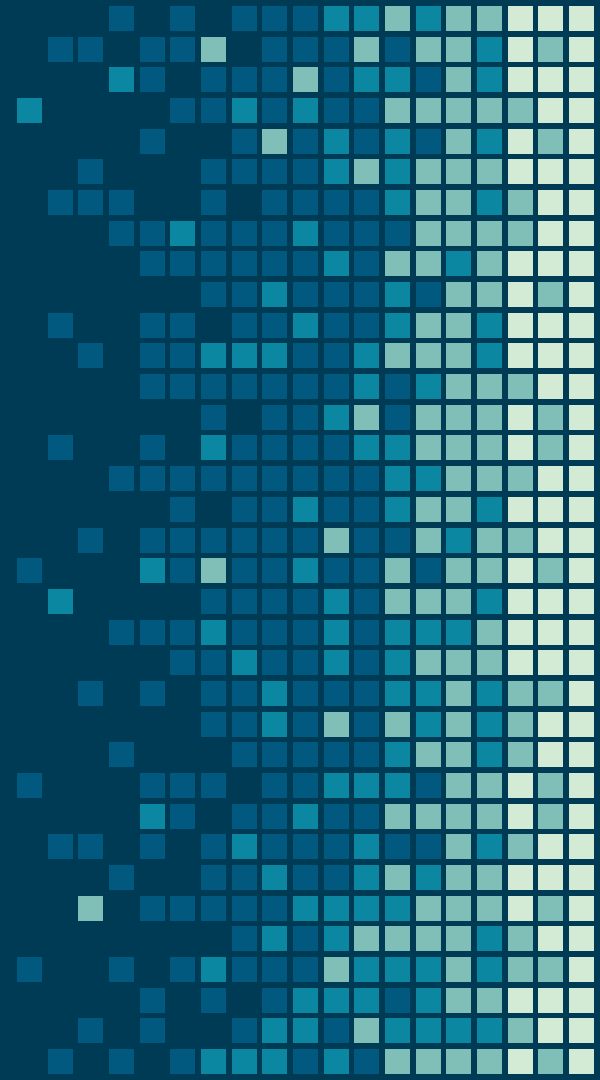


AR Home Builder

Kyle McFadden

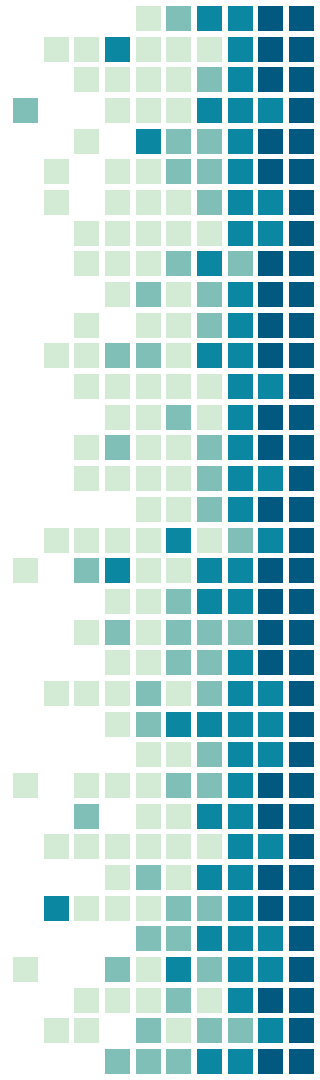


What is AR?



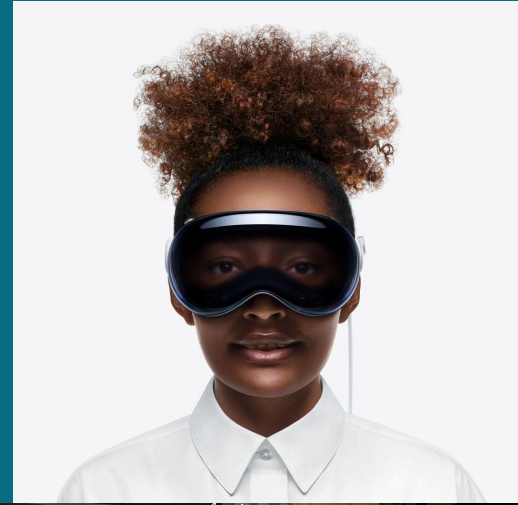
Project Objectives

- Users can choose from a selection of colors/textures for:
 - Walls/siding, doors, shutters/blinds, roofing, and stone
- Users can import other colors and textures (custom shapes)
- Virtual objects should have realistic lighting
- Users can move about while objects remain in place
- Users can take a snapshot and/or recording of the AR camera feed



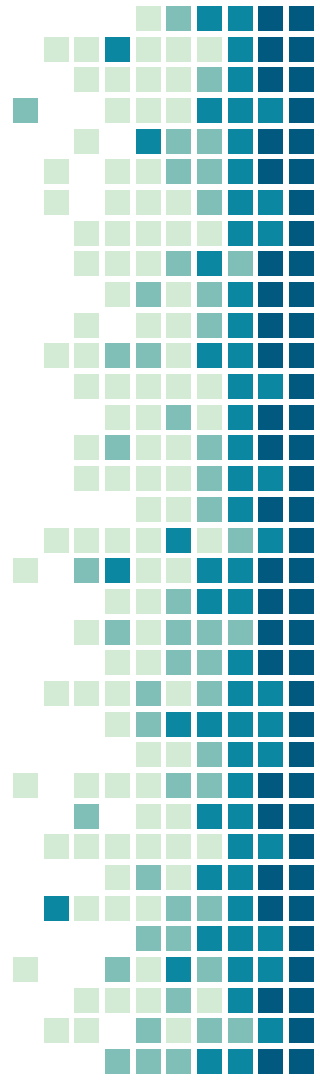
Early Beginnings

- The project was planned to be using the Apple Vision
- Had to pivot due to physical and personal reasons



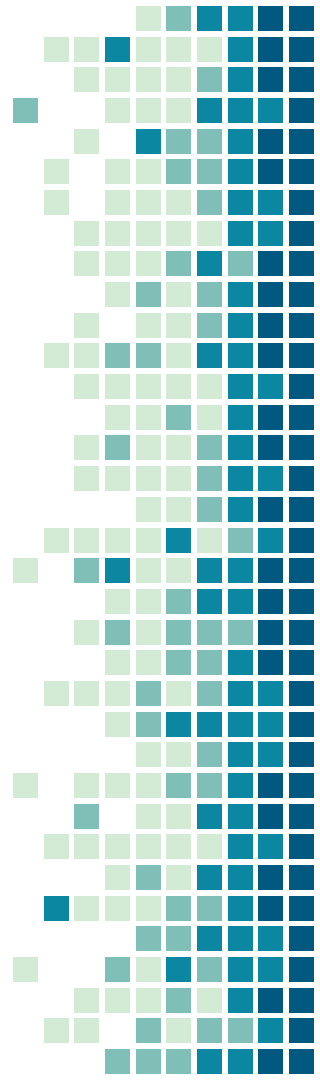
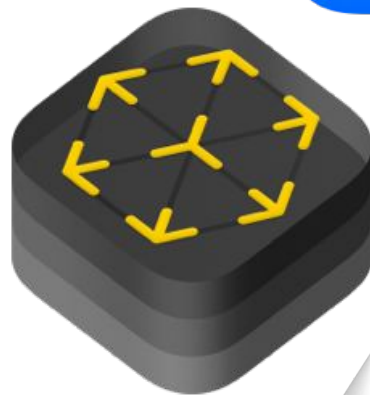
My Goals for the Project

- Ease of Use - To create an app that is easy to navigate and intuitive to use
- Customizability - Giving the user control over look and size/shape of placed objects
- Simplicity - Keep it simple, do not over bloat with features
- Good Aesthetic - Use a consistent design theme throughout the whole app

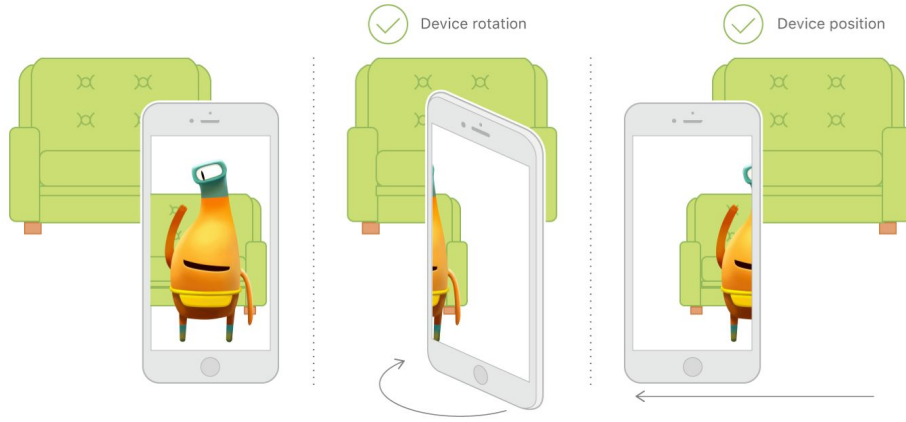


Components/Architecture

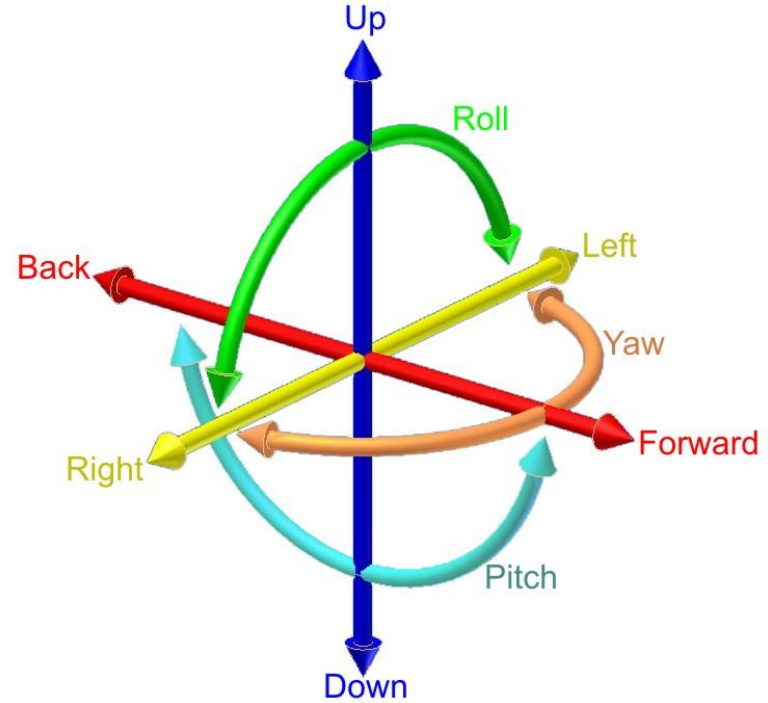
- Coding languages: Swift
- Libraries used:
 - Swift UI (and a bit of UIKit)
 - AR Kit
 - Reality Kit



How Does it All Work?



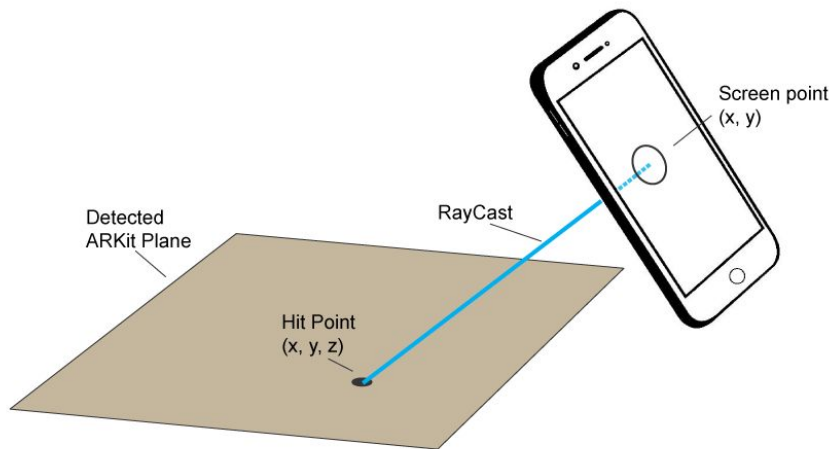
An illustration from Apple's dev website on AR World Tracking. Shows how virtual objects remain in position while the user moves about with the camera.




Representation of the six-degrees of freedom

Core Essentials of World Tracking

- Plane Detection
 - Detects flat surfaces on which the user can place down virtual objects
- Raycasting
 - Converts a point on the 2D screen and maps it to a coordinate on the 3D plane




App Layout


Width: Color: 

Height: Metallic: ☐


Depth: 0.02

0  0.1

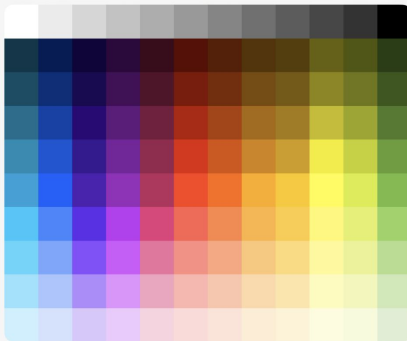
Roughness: 0.0








0  1


[Update Shape](#)

Select color: 


☐ Grid ☐ Spectrum ☐ Sliders




      

Color:  Metallic: ☐

Depth: 0.02

0  0.1

Roughness: 0.0

0  1



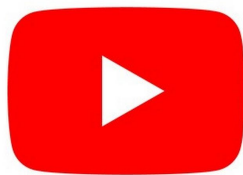
^ This will appear by doing an action which will be shown in the recorded demonstration

App Demonstration



Research/Learning Process

- Swift documentation from Apple
- YouTube
- Stack Overflow
- Hacking W/ Swift
- And various other blogs



HACKING
WITH SWIFT

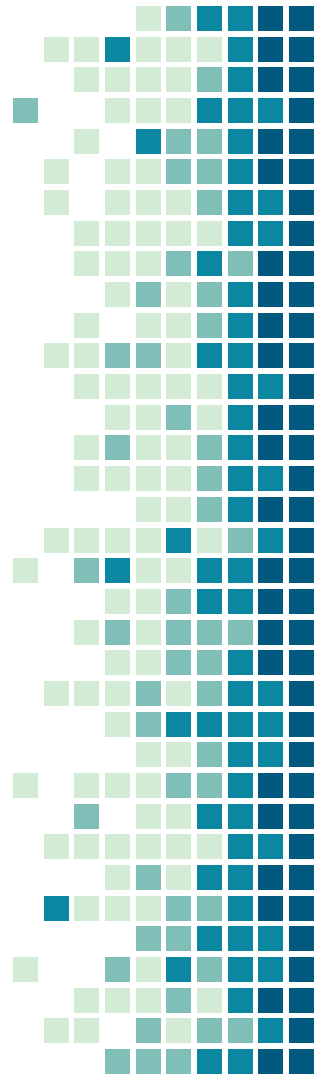
Challenges & Exceptions

- Placing objects on slanted surfaces
- Texture application
- Can not customize blinds or shutters
- Leaving gaps in the planes to prevent them from covering up windows

Further Optimizations

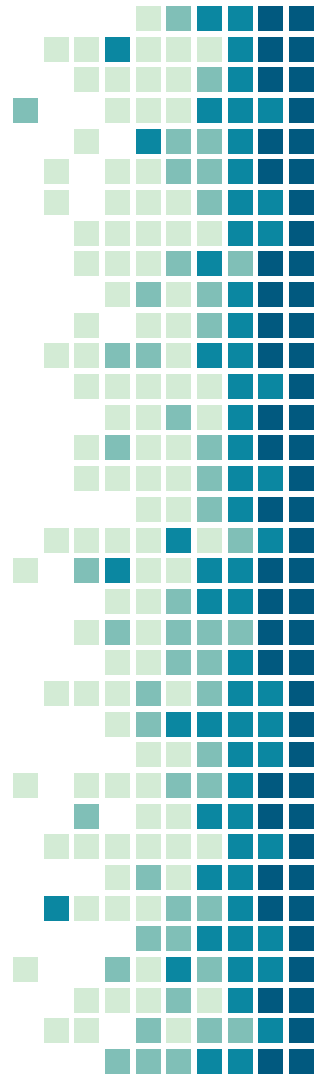


- Adding some QoL features
 - Make it easier to find detected planes
- Reducing the overall power usage
 - This app can eat through battery very quickly
- Adding the option to manually size objects
- Allow for adding opacity to the objects



The Project's Future

- Tips for a future CSCI student if they expand upon this topic:
 - Familiarize with how Swift UI / Storyboards work
 - Learn how to use event handlers efficiently
 - Look more into object detection and how it could be potentially used to detect certain parts of a house



Final Questions?