PRD Version One

Authors

Cole Margerum cole.margerum@gmail.com (Group lead)

Michael Amalfitano amalfitano@ucsb.edu (Scribe)

Isaac Zinman zinisaac@gmail.com

Jake Guida jlguida@ucsb.edu

Artem Jivotovski artem.jivotov@gmail.com

Team

'); DROP TABLE TEAMS;--

Project Title

Free Real Estate

Introduction

Our project will allow property managers to avoid the hassle of buying furniture and other decor by staging a property virtually. Currently, property managers have to hire people to move in furniture, decorations, and appliances. This takes a considerable amount of time and money, and it prevents the property manager from showing anyone the property until it is completed. Our augmented reality application will allow the user to virtually place furniture, walls, and decor in an empty room for prospective tenants to view. In addition to making the property manager's job cheaper and easier, prospective renters will be able to view properties decorated in their preferred style. Our application gives the customer the power of choosing how the interior looks while they view the property, in real time. Moving into a new property is a big commitment -- with our app, customers will be able to feel more certain of their investment.

Products that offer similar features are Ikea Place and ColorSmart by BEHR. Ikea Place allows the user to place furniture in their home, and ColorSmart allows the user to change the color of the walls. What makes our application so innovative is its ability to combine these features into one app, as well as providing additional functionality. Our app will differ from

these because it will have persistent objects, which the other two do not offer, and it will allow the user to customize the floor.

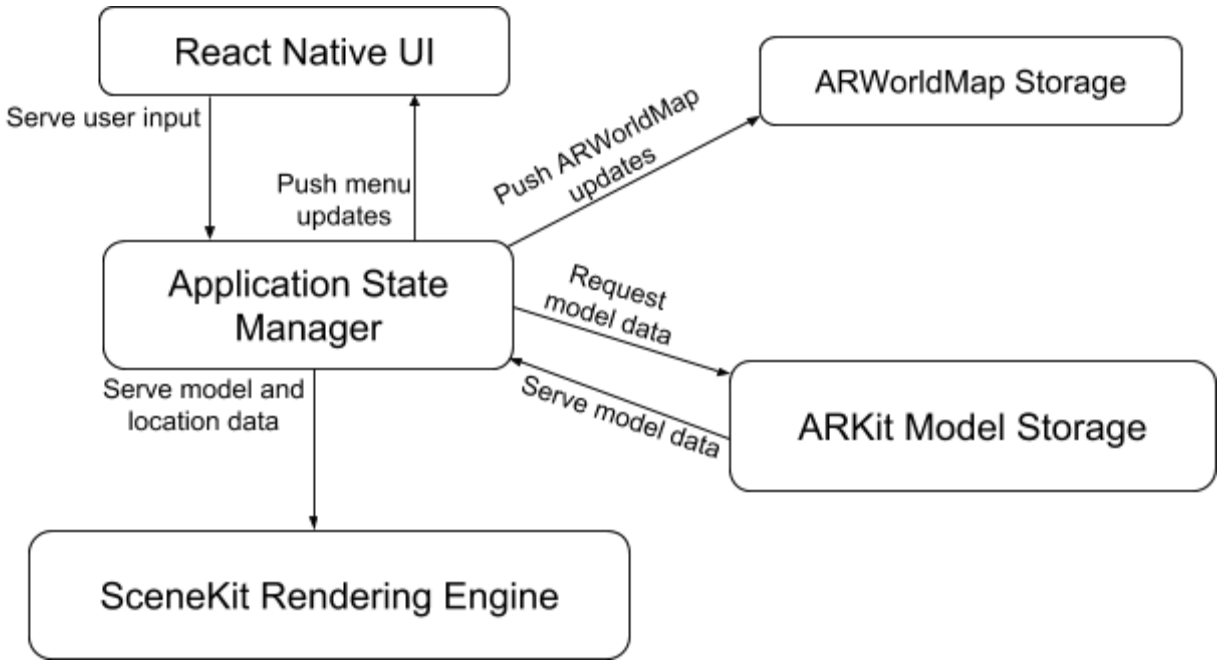
Although there are current augmented reality applications on the market that allow users to place furniture, none of them are designed for property managers, nor do they feature persistent AR settings. Our biggest innovation will be the introduction of AR furniture presets designed by the property manager, which will be visible to users once they start touring the property. In order to introduce such functionality, we will be leveraging the capabilities of the latest advance in AR technology, Apple's ARKit 2. The API features persistent ARWorldMaps, which allow the state of the virtual space to be saved and loaded between sessions.

In order to ensure a smooth and consistent user experience, we will develop the frontend utilizing React Native. This language will enable us to quickly prototype and deploy UI components in order to develop a minimalist but powerful interface. Additionally, React Native features robust APIs for interacting with native Swift code, which will ensure that the apps performance is not bottlenecked by calls to the backend components. The project's backend will be written in Swift to allow for efficient interfacing with ARKit and iOS.

Our initial goal is to create a minimum viable product that can handle simple AR functionality. This simply includes the ability to place furniture and small decorative items in a space. Once we have that completed, we will move on to being able to change the style and color of the floor, as well as the color of the walls. After making these additions, we will work on our stretch goals like allowing the property manager to create staging presets with ease and being able to quickly swap between them. We hope to achieve such functionality through a robust and intuitive UI, focused on providing a simple but dynamic user experience. In addition to this, we plan on optimizing the storage and loading of 3D models in order to ensure that transitions between presets are seamless.

Our non-technical goals are to cultivate a dynamic and efficient work environment. We aim to split work up evenly among all of our team members, and support each other when needed. We also want all of our team members to be excited about what they are working on; this includes the tech stack, the product, and individual components in development. The biggest goal for us as a team is to have a presentable project by the end of winter quarter.

System Architecture Overview - High Level Diagram



Functional & Non-Functional Requirements

Use Cases

1. Allow camera access to application
    a. Actors: User 1
    b. Precondition: User launches our app from their home screen for the first time.
    c. Flow of Events - Basic Path
        i. System asks the user for permission to use the iPhone's rear facing camera
        ii. User grants the app permission to use the iPhone's rear facing camera
    d. Flow of Events - Alternative Path
        i. User does not allow the app to use the iPhone's rear facing camera. The app displays a message that in order to use the app the user must go to Settings and allow camera access.
    e. Postcondition: User can see a live camera feed on their display.
2. Identify flat surfaces

a. Actors: User 1
b. Precondition: **uses** case 1 basic path
c. Flow of Events - Basic Path
    i. System renders the live camera feed on the display
    ii. System uses ARKit and SceneKit to identify flat horizontal and vertical planes in the scene
    iii. System overlays a yellow colored plane above flat objects it recognizes on the display
d. Flow of Events - Alternative Path
    i. User has not yet allowed the application access to the device's camera. An error is shown on the display
e. Postcondition: User can see all flat surfaces recognized in the scene.

3. Select a piece of furniture to place in scene from a menu
    a. Actors: User 1
    b. Precondition: **uses** case 1 basic path, **uses** case 2 basic path
    c. Flow of Events - Basic Path
        i. System has a menu button (+) in the bottom center of the display
        ii. User taps menu button
        iii. System pulls list of 3D objects from ARKit model storage
        iv. System displays list of 3D objects in a menu window overlaying the live camera render and surface detection
        v. User selects a 3D model option from the menu by tapping it
    d. Flow of Events - Alternative Path
        i. System is unable to recognize flat surfaces in the scene and the menu items are grayed out
    e. Postcondition: User successfully selects one 3D object and the menu disappears, revealing the live camera render and recognized flat surfaces
4. Place the piece of furniture selected
    a. Actors: User 1

b. Precondition: **uses** case 3 basic path
c. Flow of Events - Basic Path
    i. System allows user to drag the selected 3D object around the scene
    ii. System retains position of 3D object (placed on a recognized flat surface) once user lifts their finger from the display
d. Flow of Events - Alternative Path
    i. System does not allow user to place an object because the surface the user chooses has not been recognized by ARKit as a flat surface.
    ii. The object disappears from the screen and a message is displayed "Unable to place object. Please choose an item from the menu and place it on a flat surface."
e. Postcondition: System retains position of the 3D object as the user moves their device

5. Walk around a piece of furniture and view it from different angles
    a. Actors: User 1
    b. Precondition: **uses** case 4 basic path
    c. Flow of Events - Basic Path
        i. System retains position of the placed 3D object as the user walks around the object, always pointing their device in the direction of the object
        ii. System renders and displays the 3D object from different positions as the user physically walks around it (front, sides, back, top, etc.)
    d. Flow of Events - Alternative Path
        i. System loses the orientation of the scene and the 3D object disappears
    e. Postcondition: System retains position of the 3D object as the user moves their device
6. Select a placed piece of furniture to alter its position
    a. Actors: User 1
    b. Precondition: **uses** case 4 basic path
    c. Flow of Events - Basic Path

i. User taps and holds a previously placed 3D object
ii. System allows user to drag the selected 3D object around the scene
iii. System retains position of 3D object (placed on a recognized flat surface) once user lifts their finger from the display

d. Flow of Events - Alternative Path
i. System does not allow user to move the object because the new surface the user chooses has not been recognized by ARKit as a flat surface.
ii. The object disappears from the screen and a message is displayed “Unable to place object. Please choose an item from the menu and place it on a flat surface.”

e. Postcondition: System retains position of the 3D object as the user moves their device

7. Select a placed piece of furniture to remove it
a. Actors: User 1
b. Precondition: **uses** case 4 basic path
c. Flow of Events - Basic Path
i. User taps and releases a previously placed 3D object
ii. System displays a small delete button with a trash can next to the recently tapped object in the scene
iii. System removes the 3D object from the scene if the user taps the delete button

d. Flow of Events - Alternative Path
i. User taps and releases a previously placed 3D object
ii. System displays a small delete button with a trash can next to the recently tapped object in the scene
iii. System does not remove the 3D object from the scene if the user taps anywhere on the screen besides the delete button

e. Postcondition: The selected/deleted 3D object is removed from the scene and is no longer visible to the user

8. User hosts their scene so it can be viewed on multiple devices
    a. Actors: User 1
    b. Precondition: **uses** case 1 basic path
    c. Flow of Events - Basic Path
        i. System displays a menu button in the top corner of the display
        ii. System lists two options when user taps on menu button: “Host multi-user session” & “Join multi-user session”
        iii. System displays the message “Users on your local wifi network will now be able to share in your AR session” when the user selects “Host multi-user experience”
    d. Flow of Events - Alternative Path
        i. System displays error message if device is not connected to wifi
    e. Postcondition: A multi-user scene is created
9. User joins a hosted scene
    a. Actors: User 2
    b. Precondition: **uses** case 1 basic path, **uses** case 8 basic path
    c. Flow of Events - Basic Path
        i. System displays a menu button in the top corner of the display
        ii. System lists two options when user taps on menu button: “Host multi-user session” & “Join multi-user session”
        iii. System displays list of available local AR sessions to join when the user selects “Join multi-user experience”
        iv. System connects to User 1’s session
        v. System duplicates any objects User 1 has placed in the scene and positions them accurately
    d. Flow of Events - Alternative Path
        i. System displays error message if there are no local AR sessions being hosted

e. Postcondition: A multi-user scene is joined. Each user can place 3D objects and the other opposite user will be able to see the changes in real time on their own device.

10. View the same scene from multiple devices
    a. Actors: User 1, User 2
    b. Precondition: **uses** case 8 basic path, **uses** case 9 basic path
    c. Flow of Events - Basic Path
        i. System generates and renders the same scene on two devices
        ii. System correctly positions 3D objects on both devices so they appear to be in the same place in the room
    d. Flow of Events - Alternative Path
        i. System displays error message if there are no local AR sessions being hosted or if User 1's wifi connection is lost
    e. Postcondition: Both Users view the same SceneKit.

Prototyping Code, Tests, & Metrics

https://github.com/izinman/droptableteams/tree/Apple_demos

https://trello.com/b/Cj62CfFf/freerealestate

Appendix

Technologies employed:

- Apple's Swift APIs:
    - ARKit 2
        - API for identifying flat surfaces, placing and viewing 3D virtual objects, and creating persistent scenes between sessions.
    - SceneKit
        - API that keeps state of the 3D objects, used in conjunction with ARKit 2.
- React Native

- Framework we will use to develop most of the UI components like menus and buttons.