

Memo

To: Professor Pisano

From: Jessica Barry, Adiran Martinez, Quianna Mortimer, Sebastian Nevarez, Edwin Sun

Team: 25 - ARtour

Date: 02/27/20

Subject: ARtour 2nd Prototype Test Report

--------------------------------------------------------------------------------------------------------------------------------------------

**Required Materials**

**Hardware**

* Adafruit HUZZAH32 and Cable (x3)
* Apple Computer (x3)
* Lithium portable battery (x3)
* iPhone (x3)

**Software**

* C Code for Beacons
* Graphics
* iOS Applications
  + Front-End Application
  + AR Framework
  + Beacon Detectors
* Xcode

**Setup**

**Front-End Application**

An iPhone 6s was used to run and test the front-end application. Setup for the test began by pulling the latest version of the app from Github and installing it on the iPhone. The front-end app allows a user to select a tour, enter an access code to join the tour, enter the AR experience, and submit questions they have along with their email to receive answers from an ENG representative.

**AR Framework**

The AR framework displays the tour path from the ENG Undergrad Office to Photonics. Setup to test the AR framework began by pulling the latest update from Github and installing the app on an iPhone. The tester selected the Engineering tour, entered a valid access code, and clicked on the start button to enter the AR experience. Testing the AR framework followed the same procedures as testing for the front-end app.

**Beacons**

Three beacons were set up using HUZZAH32 boards and are coded in C. All beacons functioned as transmitters and the beacon detection app (coded in Swift) running on an iPhone was the receiver. The transmitters emitted information while the receiver detected the device and printed the device address - which composed of the UUID (Universally unique identification), the major, the minor - as well as the RSSI (Received Signal Strength Indicator) and proximity denoted as immediate, near, far, and unknown. Following prototyping, we will create a network of beacons for the desired tour areas, design convenient packaging, and focus on efficiently powering the boards.

**Measurements Taken**

**Front-End Application**

|  |  |  |
| --- | --- | --- |
| **Tested Functionalities** | **Description** | **Achieved** |
| Enter Access Code | If an incorrect code was entered, an alert appeared letting the user know the code was incorrect and the user was re-prompted. Entering the correct access code took the user to the tour start page. | 2/2 |
| Entering the AR Experience | When the user clicked the start button on the tour start page, the the AR experience began promptly and did not cause the application to crash. | 1/1 |
| AR Experience - Get Info | Within the AR experience when the user clicked on the get info button, a pop up appeared with information related to a tour stop. Each time the button was clicked a new stop with new information appeared. | 1/1 |
| Submitting Questions | When the user submitted a question, a new document with the question was created in the user’s subcollection in the Firestore database | 1/1 |
| Submitting Email Address | When the user submitted their email address, the user’s document in the Firestore database was updated. | 1/1 |
|  | **Total Possible** | 6 |
|  | **Total Achieved** | 6 |

**AR Framework**

|  |  |  |
| --- | --- | --- |
| **Location** | **Observations** | **User Experience** |
| Front of Ingalls | * Displayed the nodes correctly * Path displayed by nodes was straightforward * Application accurately tracked the user’s location and updated accordingly * Tracking was not obstructed by buildings | 5/5 - Was able to follow the tour path from Ingalls to Photonics |
| Front of Mechanical Engineering Department | * Displayed nodes * Path clipped through the walls * Not accurate - pathing required more direction changes (must turn left or right) | 4/5 - Although nodes were inaccurate, user could intuitively follow them to the destination |
| Front of Photonics | * Displayed nodes * Path did not accurately lead to entrance * iPhone location said users were in front of St Mary’s St. instead of the alleyway next to Photonics * Tracking was obstructed by nearby buildings * Application was not aware of the alleyway that most vehicles cannot enter | 2/5 - Nodes do not update frequently and do not realize the user has reached their destination |
| The entrance to Ingalls | * Displayed nodes * Path clipped through walls * Assumed user was outside the entrance to Ingalls * Mapkit did not provide information inside of buildings | 3/5 - Was close enough outside that iPhone could get GPS tracking |
| Inside Photonics | * Displayed nodes * Path was not accurate * Path clipped through walls * Application got the user’s location inside the building * Nodes did not update inside Photonics | 1/5 - Path did not update and frequently showed the wrong direction |

**Beacons**

|  |  |  |
| --- | --- | --- |
| **Tested Functionalities** | **Description** | **Achieved** |
| Beacon-to-iPhone Connection | iPhone could detect all three beacons and correctly display the UUID, majors, and minors | 3/3 |
| Beacon Proximity Coloring | iPhone detected all beacons, determined which beacon is closest and changed the background to the beacon’s assigned color - red bcn 7, blue bcn 9, green bcn 8 | 2.5/3 |
|  | **Total Possible** | 6 |
|  | **Total Achieved** | 5.5 |

Stagnant Readings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Distance(m):/ Readings: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 62 | 68 | 71 | 69 | 74 | 69 | 74 |
| 2 | 61 | 69 | 73 | 68 | 75 | 67 | 76 |
| 3 | 60 | 70 | 75 | 67 | 74 | 69 | 79 |
| 4 | 62 | 69 | 76 | 68 | 73 | 68 | 78 |
| 5 | 61 | 68 | 77 | 67 | 72 | 69 | 76 |
| 6 | 62 | 69 | 78 | 68 | 71 | 70 | 75 |
| 7 | 61 | 66 | 77 | 67 | 70 | 69 | 74 |
| 8 | 60 | 68 | 76 | 71 | 71 | 71 | 73 |
| 9 | 59 | 69 | 76 | 70 | 72 | 70 | 74 |
| 10 | 60 | 70 | 75 | 71 | 71 | 71 | 77 |
| Avg: | **60.8** | **68.6** | **75.4** | **68.6** | **72.3** | **69.3** | **75.6** |

Constant Walking Readings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trails: | 1 | 2 | 3 | 4 |
| Speed: | 0.848 m/s | 0.986 m/s | 0.583 m/s | 0.885 m/s |
|  | 89 | 85 | 88 | 82 |
|  | 87 | 86 | 84 | 86 |
|  | 85 | 85 | 83 | 85 |
|  | 84 | 82 | 84 | 87 |
|  | 83 | 83 | 86 | 85 |
|  | 82 | 84 | 85 | 83 |
|  | 75 | 80 | 84 | 82 |
|  | 74 | 79 | 85 | 79 |
|  | 75 | 75 | 83 | 78 |
|  | 77 | 76 | 85 | 74 |
|  | 74 | 72 | 84 | 76 |
|  | 71 | 69 | 82 | 77 |
|  | 72 | 67 | 83 | 75 |
|  | 63 |  | 80 | 72 |
|  |  |  | 76 | 69 |
|  |  |  | 72 |  |
|  |  |  | 73 |  |
|  |  |  | 76 |  |
|  |  |  | 75 |  |
|  |  |  | 76 |  |
|  |  |  | 77 |  |
|  |  |  | 76 |  |
|  |  |  | 71 |  |
|  |  |  | 70 |  |
|  |  |  | 67 |  |
|  |  |  | 60 |  |

The constant walking testing was conducted in a 16 meter hallway with a subject holding the phone and walking towards the beacon (located at bottom of the wall).

**Conclusions**

**Front-End Application**

From the data collected during prototype testing for the front-end app, we can conclude that keeping the user interface simple with only a few buttons allows the user to navigate effortlessly back and forth through the app. Furthermore, using alerts to let users know when features are unavailable allows the user to only use the app the way it was intended and avoids unforeseen bugs. The front-end app is currently at a stable version that meets all of the requirements outlined by our clients. Moving forward, our team can begin integrating both the AR framework and beacon detectors with the front-end app to make one ARtour application. With one app, our team can begin testing our design along the actual tour route and creating content to add to the AR experience.

**AR Framework**

During testing, we found that although the user can intuitively follow the path left by spherical nodes, the user experience began to suffer when the nodes did not update and accurately change based on the user’s location. Starting from areas in front of Ingalls, the ME Department, and Photonics, the accuracy of the path suffered as we approached areas that are more walled and required more turns. For example, walking down Cummington had no issues until we turned onto Babbitt St - the spheres that represent the nodes could not be found, did not change, nor clipped through the walls. One suggestion we received in order to make the experience more user-friendly, we should detect the user’s orientation and display a message or an arrow to orientate them correctly to start the tour.

**Beacons**

Following the phone to beacon communication testing, we found some outliers in the readings, specifically in the stagnant readings trial, but an overall improvement over the former beacon to beacon testing done in the last prototype testing. The constant walking trial had more stable readings and less strange outliers as seen in the table above. The proximity color app suffered from a delay when registering which beacon is closest to the phone and changing the screen color accordingly.

Moving forward, our next steps for indoor navigation are to create NFC devices for target locations, as well as design a beacon network using coordinate graph method. This entails establishing a point or beacon of origin, plotting the surrounding beacons in relation to it, and using the RSSI readings to find the user’s position.