Campus Data For Geo-Location Based Augmented Reality

A Project Report

Presented to

The Faculty of the Department of Computer Science

California State University, Bakersfield

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

in

Computer Science

By

Cesar Aleman

Spencer Austin

Andrew Burt

Rodrigo Ortiz

December 2019

ABSTRACT

Campus Data For Geo-Location Based Augmented Reality

By

Cesar Aldair Aleman

Spencer Ryan Austin
Andrew Burt
Rodrigo Alejandro Ortiz

The lack of a reliable method of navigation was the motivation for this project. The goal of this application is to be able to locate *and* identify buildings across our campus. The method that we will use to deliver this data is the user’s smartphone camera. The camera will be pointed toward the building of interest and it will generate a box over the building effectively identifying it and it will also be able to display for relevant data for that building. This will be merely the building blocks to show what it is capable of.

ACKNOWLEDGMENTS

A thank you must be given to the origin of our idea, Richard Nelson, ITS Support Services Director. He inspired us to tackle this issue when he shared ideas he had for the ITS department.

Special thanks to Dr. Chengwei Lei for supporting our idea and always providing feedback.

Table of Contents

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

CSUB is a very small campus, yet finding specific locations can seem like an impossible task. The layout of CSUB can become confusing especially since identifiers on the buildings are not always clear and visible for students. It is clear that this will continue to be an issue because it has not been formally addressed. Around campus there are maps that students may walk up to locate a building. This map does not contain the common “You are here” maker for the student to orient themselves in regard to their current location on campus. Furthermore, at the beginning of each semester, campus police and other volunteers walk the campus helping students locate their classes. This is far too old fashioned. Students should not have to rely on there being a volunteer to help them locate their classes. This solution is also very temporary. When students need to find an instructor’s particular office mid-semester, those volunteers will no longer be there. Using state of the art AR technology, this is the issue our group will try to resolve this semester.

Outdoor mapping [6] allows a student to open up their “go to” mobile mapping application, such as Google Maps to navigate to a desired location on campus. Although this technology is great in a general sense, when it comes to distinct mapping of CSUB, where a student is trying to find the name of a building, or a specific landmark on campus, not a street address, the technology isn’t exact. This issue isn’t exclusive to universities; enclosed areas such as parks and zoos aren't provided guidance towards distinct buildings and locations. Additionally, the information outdoor mapping gives is limited to general directions and does not give classroom or instructor information.

Geo-based AR is another possible solution. AR is a technology that falls under the Artificial Intelligence umbrella and is used to insert graphically made objects into the real-world using a medium; Wikitude is a good example of this. Wikitude's origin had a similar idea to our own when it was created. It is based on image recognition of landmarks. With Wikitude, the user would point their phone to a landmark and relevant information of the land mark would be displayed. Wikitude uses image recognition to be able to identify these structures. Image recognition is achieved by using neural networks that are set up exclusively to process images. This network works to learn the subject of the image and thus make a distinction among images [2].

Another great example of geo-based AR is the application called PeakVisor. PeakVisor is an application that allows the user to point their phones around them and they will see nearby mountains/peaks. PeakVisor is an example of an Augmented Reality (AR) enabled application. The medium in the case of PeakVisor is a cell phone camera. The user points the camera around themselves and the app will outline mountain peaks that are nearby. The app does this by using the user's location as well as the sensors that come with the phone such as the barometer, gyroscope, and etc. In this case, there is no need for image recognition needed because the application is able to find where it is itself, and based on its location, it can then find the mountains around it. It is a clever use of the sensors that our modern smartphones are equipped with. This application as has a 3D map function that allows the user to see the terrain. It also has a virtual compass in the lower portion of the app.

MapBox is another type of Geo-Based AR. Mapbox is a navigation building tool that also allows for implementation of AR technology. Their main focus is navigation. A demonstration on the Mapbox website shows an example of a form of heads-up display. The user points their phone in front of them and the application will give guided directions. It uses the phone camera to give a path of where to go. It also displays information such as, time of arrival. The app also shows the areas surrounding the user but the main subject of the application is giving directions.

CHAPTER TWO: OUR SOLUTION

Our solution is to use a combination of all of the above referenced technologies. Using the Wikitude SDK as our base we intend to create a mobile application that will allow users to point their phone camera at buildings around our campus, and they would be given the building name. Selecting a building name would present other relevant information to the user such as instructors whose offices are in these particular buildings, what classrooms are located in the building, which department(s) the building houses such as Math, Biology, etc. Furthermore, by inputting a instructor's name or classroom into the mobile application, the student will be given navigation instructions using outdoor mapping techniques.

When considering this idea, we wondered if it fell in line with Virtual Reality (VR) or Augmented Reality. However, through some research we were able to find a distinct difference among the two concepts. VR technology immerses the user completely into the desired environment. It cuts all elements of the real-world and places the user in another. AR takes the real world and adds elements to it that would normally not appear [5], such as the compass that appears in the PeakVisor application. We understood that our goal was not to immerse the user into another setting, but rather assist them with the setting they already find themselves in by adding relevant information to their environment.

CHPATER THREE: IMPLEMENTATION

Point of Interest Object Creation

The backbone of this project became Wikitude. Once we became familiar with it we began to gather data for the JSON file we would use to populate the users screen with information. The POIs, or Points of Interests, would become these:

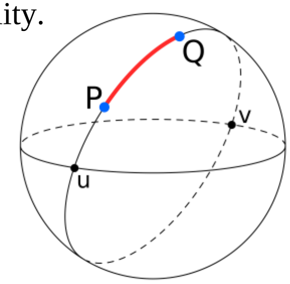
{"id":"20","longitude": "-119.104056","latitude":"35.348926","description":"Science 3","name":"SCI-III"},

Figure 1

Each location of campus would need to have its own object in order for it to populate with its unique properties. Once we were able to gather all this data, we then moved onto how to actually get the data to appear for the user. The main page that the users sees once the app is launched and the phone camera is used is a simple yet robust HTML page. The page interacts with various JavaScript functions to allow for the population of the points. All these points are stored in https://cs.csub.edu/~caleman/SenSem/testdata.

Our main JS file creates a couple of variables to store basic information such as the locations of various resources that we use for the app to work. But the main object to consider is the World object. Here is where the 'world' that is seen through the screen is created. Within the object we can obtain the users location to be able to begin populating the points that are relevant to the user. Each of the properties of the object are stored within the POI object variable. These properties were shown above in figure 1. Now something to note here is that altitude plays a huge role in this. Originally altitude was not being accounted for but after reading some Wikitude documentation we were able to use altitude to fix an issue where all the points were either too low or too high. The way we dealt with the issue was to use the user's altitude and dynamically change how high or low an object sat depending on the proximity of the user to the object.

The proximity to the user also took some time to figure out as we were dealing with an object in a three-dimensional space where we have latitude, longitude, and altitude. After several hours of research, we were able to use the following formula in Figure 2 to find the distance between user and object that would later prove to be essential for our range functionality.



$$D = 3963.0 * \arccos\left[(\sin(lat1) * \sin(lat2)) + \cos(lat1) * \cos(lat2) * \cos(long2 - long1)\right]$$

Figure 2

Point of Interest Selection

Once the user has found a POI that they would like to see more information of, they are able to select the POI to get information about that building. Once the object is selected, a window will slide from the right or left, depending on the location of the object and it will display some information that relates to the object that has been selected as can be seen in Figure 3. In this case, Science 3 was selected, and the panel came to show some of the relevant information about Science 3. Our original intention was to be able to show events, meeting, etc that would be happening in each building. However, that proved to be a challenge as we would need to find a way to incorporate software that is being used by CSUB to populate such information. Unfortunately, we began to speak to CSUB about our project in the last half of the development cycle. One thing we were able to implement was the floorplans. These floorplans were our solution to indoor navigation without having to spend too many resources solely focusing on indoor navigation. These floorplans show a static map of where each room is within a building.

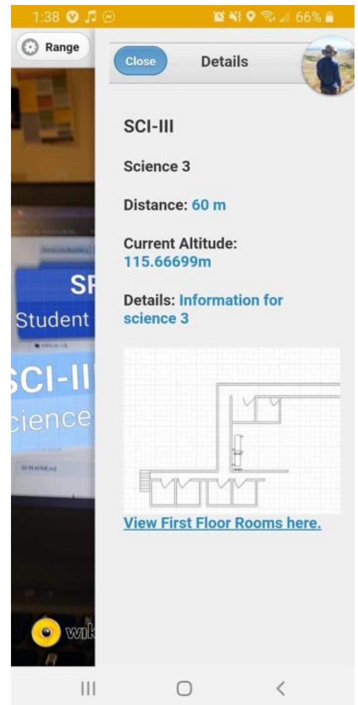


Figure 3

Point of Interest Range Functionality

We also were able to implement some important functionality in the way of adjusting the range of objects that appear for the user. We thought about not having this function because we would only want to display points that are relevant to the user and not every single, once the app has loaded in. However, it is likely that a user may need to find a location that they are not near and thus the range function would be very useful. The range function can be seen in Figure 4.

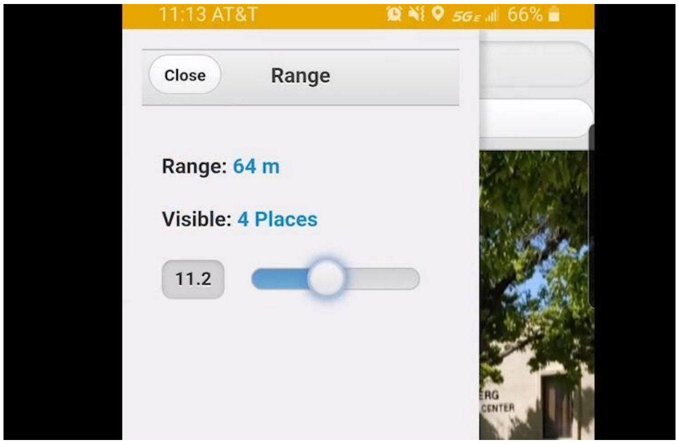


Figure 4

This range function is also dynamic. This means that if the user is very far away from any point of interest, the max range will the furthest point to the user. It will still show points despite not being near campus. This can be useful for a passenger in a car or bus using the app to be able to see what building is what.

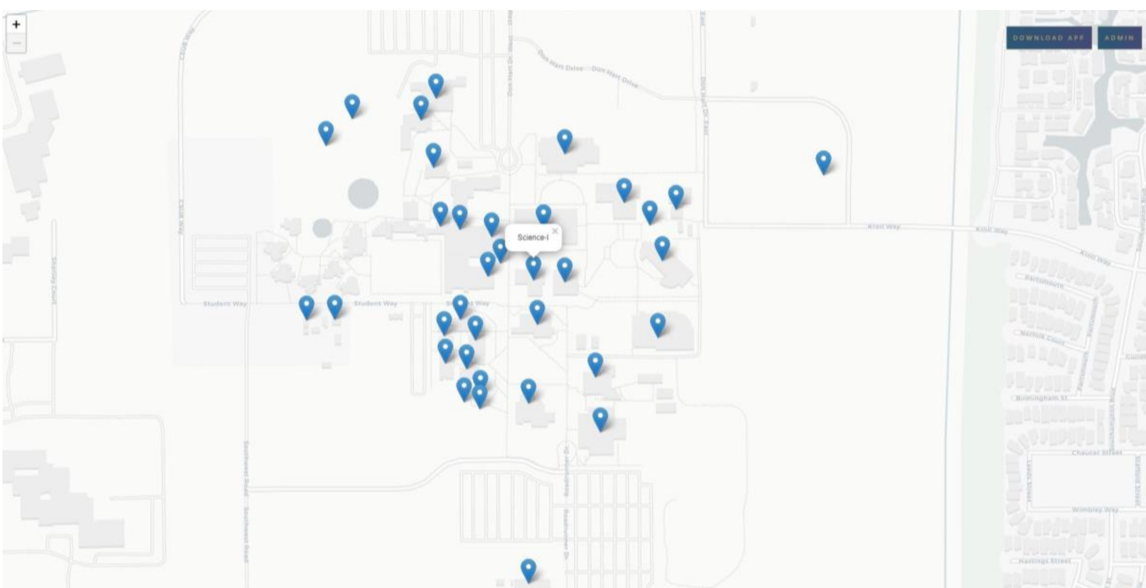
CHAPTER FOUR: DATA MANAGEMENT

For the data management portion of the project we initially created a simple JSON file that contained the information for each point on campus. The format of these points is show in Figure 1. The test data that we used under development is located at https://cs.csub.edu/~caleman/SenSem/testdata. But we took it a step further. This was a hard decision to make because these points do not move often, if at all. The only way the data set would be changed would be if there were to be the addition or demolition of a building. As can be seen in Figure 5 below.

1	FABLAB	FAB-LAB	35.348053	-119.104688	Fabrication Lab
2	WSL	Walter Stern Library	35.351424	-119.103242	Walter Stern Library
3	DDH	Dorothy Donahoe Hall	35.350372	-119.103609	Dorothy Donahoe Hall
4	RC	Runner Cafe	35.350743	-119.102228	Runner Cafe
5	SU	Student Union	35.349926	-119.101571	Student Union
6	SRC	Student Recreation Center	35.3488513	-119.1016427	Student Recreation Center
7	SHE	Student Housing East	35.3511242	-119.0988055	Student Housing East
8	IC	Icardo Center	35.347532	-119.102633	Icardo Center
9	SHS	Student Health Services	35.347927	-119.103862	Student Health Services
10	PE	Physical Education Building	35.348295	-119.102714	Physical Education
11	ENGRCOM	ENGR-COMP	35.347852	-119.104701	Engineering Complex

Here the manager of the data may edit, add, and remove a point which would represent a POI in the app. Changes made here would take a refresh to appear in the app. This has been tested and it is possible to pull he data set from this console.

In addition to this data set, there is also a map that goes along with it. It is a very basic bird's eye view of CSUB with labeled points. This can be seen in Figure 6 down below.



This is how we handled our data and were able to stay away from incorporating a database that may have weighed down the app. This way we use a simple file that can be edited, and changes made to it are instantly recognized in the app.

CONCLUSION AND CLOSING REMARKS

This project was rather linear for us. We all formally worked together at the CSUB Helpdesk so our lines of communication were in place at project inception. From day one of this undertaking, we knew what our project was going to be and worked toward making our concept reality. From that starting point, we began to research, making significant progress each week. The resulting application of all of our work can be of great use, not only to CSUB, but in a "real world" setting as well. It would be a simple matter to add in as many points as you would like and change the app for City, County or even recreational use. However, we also want to give back to a University which has done so much for us. So, instead of gaining monetarily from this application, we plan on donating our application to CSUB. As such, we set up a meeting with Rich Nelson, the director of Information Technology Services at CSUB. After presenting our application to him he indicated he would like to incorporate it into the CSUB digital ecosystem. However, other people than himself, need to sign off on it. To accomplish this integration, we are currently in process of setting up a second meeting with the heads of each department within Information Technology at CSUB. As soon as we present our app, our app will begin the implementation phase. However, we found ourselves in the midst of a global pandemic, thus our progress in this regard, has been placed on hold as the resources of CSUB ITS have shifted to adapting to our current circumstances.

Regardless of where our application ultimately ends up being used at, we stand by our application and our proud of our product. Working as a team, our learning curve was ramped up and we have learned how to communicate and manage many different spokes of a project. We would also like to thank our advisor on this project, Dr. Chengwei Lei, whose guidance helped our concept become a reality.

REFERENCES

1 - https://www.wikitude.com/documentation/
2 - https://www.techopedia.com/definition/33499/image-recognition
3- https://peakvisor.com/
4- https://www.mapbox.com/augmented-reality/
5- https://www.fi.edu/difference-between-ar-vr-and-mr

APPENDIX

Roles and Responsibilities

The work for our project is assigned as follows:

Cesar Aleman

- Create base SDK which allows for Augmented Reality capability within Android application.
- Map GPS coordinates on campus
- Data Entry – Classrooms & offices within buildings
- Implement Radar within app
- Testing and De-Bugging

Spencer Austin

- Obtain Education license for the Wikitude SDK
- Create base SDK which allows for Augmented Reality capability within Android application
- Configure Android application to use the license provided by Wikitude
- Set up Git Hub for use in Android studio
- Data Entry – Classrooms & offices within buildings
- Map GPS coordinates on campus
- Implement Radar within app
- Testing and de-bugging

Andrew Burt

- Data entry and coordinate management web application to manage the JSON data for our project
- Map GPS coordinates on campus
- Configure Git Hub for use with Android Studio
- Data Entry – Classrooms & offices within buildings
- Implement Radar within App
- Testing and debugging

Rodrigo Ortiz

- Graphical User Interface Design
- Map GPS coordinates on campus
- Data Entry – Classrooms & offices within buildings
- Implement Radar within app
- Testing and debugging

Timeline

Figure B1

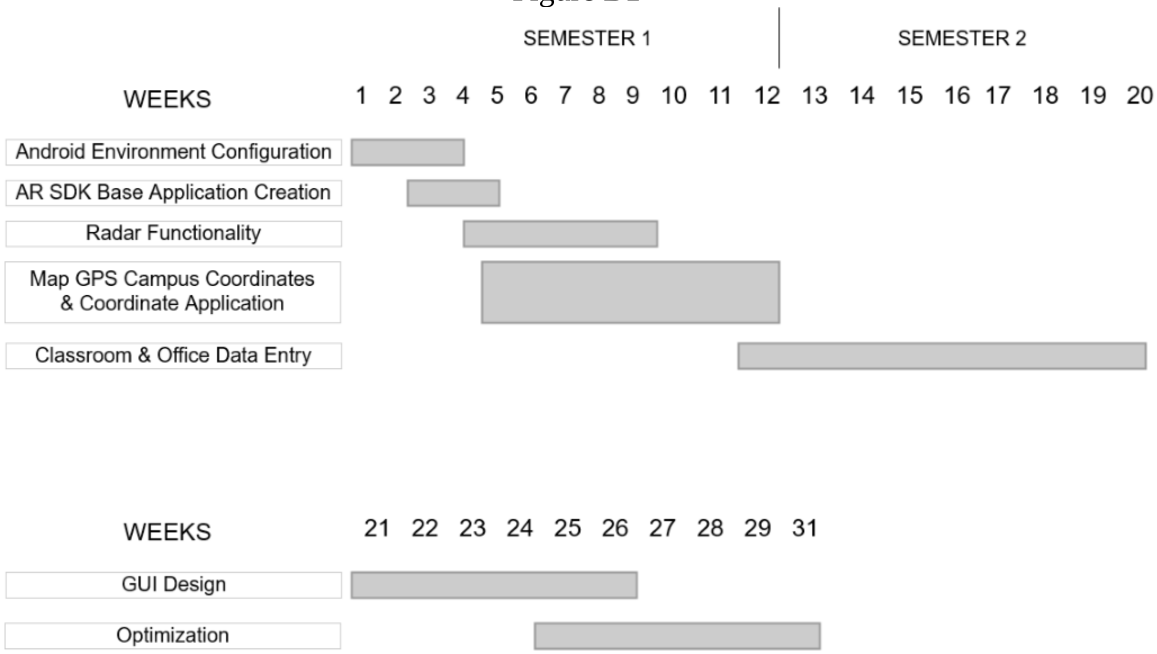


Figure B1 Software development timeline showcasing priorities and the length of tasks