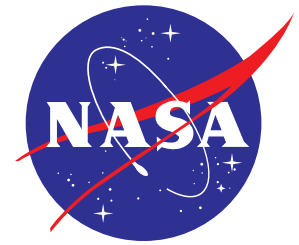




College of Engineering



CS CAPSTONE TECHNOLOGY REVIEW

NOVEMBER 21, 2017

ISS BAROMETER APP

PREPARED FOR

NASA

DON PETTIT

Signature

Date

PREPARED BY

GROUP20

AUTHORED BY

CADE RAICHART

CADE RAICHART

Signature

Date

DANIEL KATO

Signature

Date

NATHAN SHEPHERD

Signature

Date

Abstract

This document will review three major components, Graph interface, Barometer implementation and UI library, of the nine our group is doing. The paper will review three technologies for each component, it will then display the chosen technology and why it was chosen for this project.

CONTENTS

1	Introduction	2
2	Graph Interface	2
2.1	Option 1: ScrollableGraphView	2
2.2	Option 2: Charts	2
2.3	Option 3:Swift Charts	2
2.4	Decision	3
3	UI library	3
3.1	Option 1: Cocoa controls	3
3.2	Option 2: CRPageViewController	3
3.3	Option 3: FlatUIKit	3
3.4	Decision	3
4	Barometer Implementation	3
4.1	Option 1: iPad Air 2	3
4.2	Option 2: Extrnal Barometer-arduino	4
4.3	Option 3: iPhone 6	4
4.4	Decision	4
	References	5

1 INTRODUCTION

The ISS Barometer is an IOS app created for the astronauts going and on the International Space Station(ISS). This app will be used in emergency situations to find atmospherical pressure leaks, using built in technology of the iPad Air 2s barometer. This project has been split into 9 pieces, my role in this project is to create the Graph page using the barometer and helping with UI design.

We want our graph to display dt/dp as a slope, with our graph constantly updating and growing. It needs to be able to zoom and pan the entire graph, as well as be easily visible in low light. The y-axis will need to be able to grow indefinitely while the app is running, the astronauts may be using the app to collect atmospheric data for hours, we need to be able to display this.

Our UI needs to be easy to program, easy to change, and readily by our employer. It needs to be able to move from one page to the next and have buttons. The client has told us to using the iPad Air 2 as our barometer implementation. I have reasearch some other options in case the employer wants them but since there is already iPads aboard the ISS i do not see them changing this do to cost of sending physical things to the station.

2 GRAPH INTERFACE

Our app will be using running graphs, graphs that continually display as the device records data, that can zoom and pan this ability is required from the client. A swift library will be chosen to accomplishes that task. ' The graph must also be legible and easy to use so that there is no time wasted, this is an emergency app we need it to be simple. Our client has asked for the graph but have not give us any set libraries to use, so we get to choose the best library.

2.1 Option 1: ScrollableGraphView

Scrollable Graph View features the animations for the graph, relocation of values while in use, changing the size of the range while scrolling, smooth scrolling, handles large amount of points and has customizable for looks. [1] I picked this technology because of its ability to scroll, change range while working and its workload ability. This technology is better than the others i chose because the others don't have as fluid and easily displayable graphs. As well as this library has the most support to help you program a graph for the iPad.

2.2 Option 2: Charts

Charts features eight charts, x and y scaling, side scrolling, combining charts, highlights, saving chart to camera roll, color template, customizable legends and Animations. [2] I picked this technology because of its scrolling ability and how easy it is to read the data. Unlike the other two, this Library has a very clear and easily customizable labels as well as the fact that it is fastest and easiest to change and upgrade from within the code. These three reason the advantages it holds against the other two libraries.

2.3 Option 3:Swift Charts

Swift Charts features 7 types of charts, multiple labels per value, Creation of arbitrary markers, Modular architecture, Chars can be combined, legends, Zooming and panning, lockable x and y axis, extensible axis values and label generators. [3] I chose this library for its ability to zoom, panning and customize almost everything, including the zoom ability. This one is superior to the other two because of its ability to zoom, which our employer has demanded that we must have.

2.4 Decision

The app will be using Swift Charts as our graph interface. I chose this because of its ability to zoom and pan. Zooming is a required feature that the client has asked us to include, Swift Charts is the only one that offers the zoom capability with all the same features as the other libraries.

3 UI LIBRARY

3.1 Option 1: Cocoa controls

Cocoa controls library features input bars, search displays, calendar displays, ability to put pictures on the app, file indexes, map displays, rating displays, card swiping, small graph displays, easy to read data displays, pop ups, easy buttons and many many more displays. [4] I chose this UI Library because of how much you can do with it, there are just so many different displays, view and inputs that can be customizable in so many ways. It is the largest of the libraries from the other technologies I choose for the UI library, that what gives it an advantage compared to the other two.

3.2 Option 2: CRPageViewController

CRPageViewController Library features the ability to switch from page to page easily and easy to customize how those pages are displayed (size, margins, height,etc.).[5] I chose this library for its easy to use page swiper, easy on the go customization and the how the library looks. This library is superior to the other two because of its on the go customization and professional look.

3.3 Option 3: FlatUIKit

FlatUIKit is a swift library made by Grouper and it features buttons, text fields, segment controls, switches, alert Views, sliders, bar button items, navigation bars, table cells, popovers, and customizable colors, fonts and icons.[6] I chose this Library for its easy to read displayable items, we don't need fancy gadgets for our app just straightforward easy to use and read. This is superior to the other two technologies because it's straightforward to code and easy to manipulate/customize .

3.4 Decision

I chose to use Cocoa controls for is massive library and larger features list. The UI library will not consistent of just one though, no library for swift contains all the perfect functions that we will need. Cocoa controls will have most of them and be used more than the FlatUIKit and CRPageViewController.

4 BAROMETER IMPLEMENTATION

4.1 Option 1: iPad Air 2

The iPad Air 2 has an on board barometer that is easy to use and already has libraries to use it. We choose this because the employer has asked us to use these, He asked us to use these because the ISS already has iPads on board. We would be able to upload the iPad app without ever having to take a new product up to the space station, thus making it the most superior of these technologies. There are many technologies that could be used in place of the iPad but all of them would require a trip to the space station, which takes time and money they do not want. If every crew member has a iPhone we might be able to use those ones.

4.2 Option 2: Extnal Barometer-arduino

Arduino has made a easily programmable barometer for cheap purchase. Arduino has been considered one of the easiest languages to learn and boards to use. I choose this technology for it huge community and how cheap the barometer is. It is superior to the other two technologies because of its support, customization, cheapness and easy to program. This was not chosen by the employer but if needing a cheap barometer that will get the job done, use arduino.

4.3 Option 3: iPhone 6

The iPhone 6 comes with a barometer on board. This would be used the same was as the iPad and programmed using the same language. The difference between the two is the ios is older in the iPhone 6, It has a smaller display, and we don't know if they have them aboard the ISS. I chose this because it would keep us from having to learn something different then the language we have chosen and since the NASA is using iPads they might also use apple's iPhones. This technology would be superior to the other two because of how small the iPhone is (easy to carry around), it isn't heavy, no wires and can be carried in a pocket. This could replace the iPad if they didn't have to make new trips to the ISS.

4.4 Decision

We will be using the iPad for our barometer implementation. This was chosen by our client, but i would also choose it with their situation. There is already iPads aboard the ISS, so no need for a new mission just to send a new app up to the station. The iPad has a bigger screen that will allow easy viewing and not cramp the graph space.

REFERENCES

- [1] (2017) Scrollablegraphview. Philackm. [Online]. Available: <https://github.com/philackm/ScrollableGraphView#features>
- [2] D. Gindi. (2017) Charts. [Online]. Available: <https://github.com/danielgindi/Charts>
- [3] I. Schutz. (2017) Swiftcharts. [Online]. Available: <https://github.com/i-schuetz/SwiftCharts>
- [4] (2016) Cocoa controls. cleveroad. [Online]. Available: www.cocoacontrols.com/controls?language=2-swift&platform_id=ios
- [5] (2016) CrpageviewController. cleveroad. [Online]. Available: <https://github.com/Cleveroad/CRPageViewController>
- [6] M. Waxman. (2016) Flatuikit. Grouper. [Online]. Available: <https://github.com/Grouper/FlatUIKit>