Creating an interactive maths learning environment, on a tablet, that can be used in classrooms

Conor Delahunty
B.A.I. Engineering
Final Year Project: April 2014
Supervisor: Brendan Tangney

School of Computer Science and Statistics
O’Reilly Institute, Trinity College, Dublin 2, Ireland
DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

______________________________________________  _______________________
Name                                      Date
Acknowledgements

I would like to thank all who have help me with the project at any stage. Most Notably Brendan Tangney, for his continuous help and willingness to put up with me and Aibhín Bray, without her the project would not have nearly been as successful as it was.
Abstract

Two years ago my old secondary school introduced an initiative for all newly joining first years, that they would now be using iPads in school for all classes and that all books and learning resources would now be solely on these tablets. The main reasons for this were to lighten the load of student’s bags, so that they wouldn’t have to be carrying a large amount of books to and from school, and also that they would now have access to all manner of online resources, tools and apps to assist in their subjects. Having always been extremely interested in tech, I was initially both excited and intrigued while at the same time incredibly jealous of them. That was until I managed to get a look at a neighbor’s iPad and all of the apps and books they had on it. I was shocked to learn that all the eBooks on their tablet were simply page for page copies of the written textbook, with no media, games or even links to external resources. This, to me, seemed like a complete waste of the capabilities of what is basically a handheld computer, using it only as something that is lighter than a physical textbook.

From this point I formed the idea in my head for a completely interactive and immersive textbook, choosing maths as the subject due to the fact that I was aware there was issues with maths education and also as it had been my favorite subject in school. From then on I began to start researching further into the area of maths education, specifically it being supplemented by technology and began to see more and more evidence for the need for reform in the education system.

This project therefore, is to develop an interactive, maths learning environment on a tablet. It seems clear that mobile is the future and where a lot of resources are heading, so to be able to create an application that could offer something over and above what students are currently getting from textbooks would be amazing. To do all this while at the same time encouraging students to engage with, and try and get them to see some genuine reasons to use and love maths seemed worth the effort. The app itself is made using PhoneGap, a hybrid
app development platform, and some native Android Java and has all been tested in a Samsung Nexus 10. It embodies 2 separate learning plans developed by Aibhín Bray and the CRITE department in Trinity.

At the end of the development process I was given the chance to evaluate the app by having 4 transition students of varying ability use and critique it to determine its functionality and if they felt that it provided enough structure to the learning plans and activities.
Table of Contents

Introduction ........................................................................................................................................ 2
The Problem ........................................................................................................................................ 3
  1.1 Stem Crisis .............................................................................................................................. 3
  1.2 Tech in Classrooms .................................................................................................................. 5
  1.3 The Book .................................................................................................................................. 8
Tablet Development Issues ................................................................................................................ 9
  2.1 Lack of Standardization ............................................................................................................ 9
  2.2 Hardware Capabilities ............................................................................................................. 10
  2.3 Interoperability of the Web and/or Webapps ......................................................................... 11
Approach/implementation ................................................................................................................ 12
  3.1 Architecture and Function ...................................................................................................... 12
  3.2 Set Up ..................................................................................................................................... 14
  3.3 Beneath the Glass ..................................................................................................................... 16
  3.4 Ux/Ui/Security ......................................................................................................................... 18
  3.5 MVP/Prototype/ Reflection on decisions ................................................................................. 19
Issues in Software ............................................................................................................................ 24
  4.1 Plugin Issues ........................................................................................................................... 24
  4.2 Tracker .................................................................................................................................... 26
  4.3 Barbie Tracker ......................................................................................................................... 27
  4.4 Other Important Issues ............................................................................................................ 28
Evaluation .......................................................................................................................................... 31
  5.1 How the Evaluation was carried out ....................................................................................... 31
  5.2 Issues with Evaluation ............................................................................................................. 33
  5.3 SUS Scores and Interpretation ............................................................................................... 36
Conclusion ......................................................................................................................................... 38
  6.1 Stem Crisis ............................................................................................................................. 39
  6.2 Technology in the classroom .................................................................................................... 39
  6.3 The book ................................................................................................................................. 40
  6.4 What Have I Learned? ............................................................................................................. 40
Bibliography ..................................................................................................................................... 41
Appendix A: System Usability Scale ................................................................................................. 43
Introduction

As it stands currently there is a crisis in the Science, Technology, Engineering and Mathematics (STEM) disciplines. We are told constantly that there is a need for more and more graduates with skills in these areas but there is still a lack of numbers and I believe that the teaching of maths in secondary school is a big factor in this. Currently there is still a large amount of rote learning off of problems, which is detrimental to the whole point of teaching maths, i.e. it takes away from applying a logical thought process to problem solving. Fostering engagement in the subject has always been an issue, I’m sure we’ve all heard the sentence, “But when am I going to use any of this in real life?” This I believe highlights the fact that the context of the maths these students are learning isn’t getting across by standard teaching means. The syllabus in place at present, project maths, is attempting to solve some of these current issues, however, I feel that for the most part it comes up short.

The use of technology in classrooms has long been touted as a possible solution to the shortcomings of traditional education. Even back at the dawn of the radio there were calls to broadcast the best teachers in the country’s lessons directly to classrooms. However, the teaching model as we know it really hasn’t changed much, 1 teacher lecturing 10 to 30 students in a chosen topic. So, why hasn’t technology changed the face of education? There are several reasons, firstly, the lack of familiarity most teachers have with technology. A majority has been teaching the same way for the past 10 to 30 years and to introduce technology now means time and effort for them as much as for the students. Secondly, funding in schools has always been an issue and unfortunately technology is often expensive and so many schools don’t see the benefit of spending a large portion of budgets on technology. Finally, although evidence is growing and there is a push across the world for facts and figures, there is no conclusive proof that technological solutions in the classroom actually improve learning. This may the biggest barrier of all as it influences all these other factors.
The evolution of the book has been a slow and incredibly long process. Up until the 15th century the only change in any sort of book was the medium on which it was printed. Then Johannes Gutenberg came along and pioneered the use of the printing press in Europe (SF Books). From then on innovation in the book industry gained little speed until the next notable change in the form of the Internet, with the integration of many forms of media across a sporadic network of information. Skip on to modern day and we see the latest progression in the form of eBooks and apps. I would argue, however, that the eBook as it is today hasn’t lived up to the potential it is capable of; it is rare, if ever seen at all, to integrate media and other technological features into these eBooks. They are simply static pdf or html files with no real interactive or media-rich capabilities, a real waste of potential.

So, with all of these problems before me I decided attempt to develop an interactive maths learning environment on a tablet. It should address the issues of lack of context and engagement in maths education through interactive activities, while being usable for students and teachers alike and above all else serve as proof of the possibilities that lie in the modern day textbook.

The Problem

1.1 Stem Crisis

A debate that has raged for years, are the Science, Technology, Engineering and Maths (STEM) subjects in a crisis is today more prevalent than ever. The debate is predominantly centered on maths education in secondary schools and the base it forms for college and other further study. Though there has been much more of a push in recent years for reforms of teaching, learning and engagement in the maths system I believe we are still quite a way off from having the syllabus where it needs to be.

An issue I believe to be central in the majority of problems students are having with maths is the lack of context associated with what they are learning. The pinnacle example of this being the question I’m sure many teachers are very
familiar with, “But when will I ever use this maths in real life?” The plain fact of the matter is that the majority of students are not making links between what they are learning in the classroom and the outside world. This has direct adverse effects on the enthusiasm of students in schools and their engagement with the syllabus on the whole. Following on from this, there seems to be a growing trend in the maths system whereby gaps develop in a student’s knowledge early on and they get left further and further behind as they attempt to build upon knowledge that simply isn’t there. Due to this they seem to assume they are not meant for maths and stop trying. This is solely a confidence problem. If we could simply boost the engagement and enjoyment of maths learning I believe we can change this fact altering the common belief that higher-level maths is only for more gifted children.

A huge factor of the current problem with maths education is the fact that many students are learning off line by line how to solve maths problems. To a certain degree I can see a benefit to this, for example, at primary school the rote learning off of times table can allow students to quickly and efficiently solve problems. However, I feel this can only take place once a student has a proper understanding of the multiplication algorithm and how multiplying works. Rote learning shouldn’t take the place of base understanding only supplement, or build upon, it.

The department of education in Ireland has made a conscious effort to solve these problems. The revised maths syllabus, Project Maths, has several aims to fix the main issues of the previous syllabus. One of these is to try and provide more detailed context behind its questions in the exams by giving examples of how maths might be used to solve a real world problem. However, I feel this isn’t and has never been the problem when it comes to context. For years students have been given real world examples after they have learned the maths: if I have 5 apples and give Mary one how many do I have left? How to split a restaurant bill using fractions? Using equations of speed, distance and time to find when a train gets into a station? All of these are almost clichéd maths problems by now and while at times useful they don’t solve the contextual problem as they tell
students a real world application, almost as an afterthought. There is minimal exploration of problem solving techniques and creativity. There is a further issue of students with literacy problems struggling more with Project Maths that has also been raised by teachers and educators NCCA (2010). There is the concern that, rather than a universal fix for the problems of the maths syllabus the issues have just been shifted to a separate group of students. The creators of the course have been clever with respect to decreasing the amount of predictability of the exam and hence may have solved the issue with rote learning off and ‘playing the exam’. It remains to be seen whether Project Maths is successful in its goals but according to critics and supporters alike, we won’t know for definite until students who have taken the syllabus from their first year go on to complete their leaving cert, i.e. they have had full immersion in the Project Maths methodology.

1.2 Tech in Classrooms

The use of technology in classrooms has long been hinted at as being the savior of all education woes. However, with every new advancement in technology that seems poised to disrupt the classroom structure, we are yet to see a real change in how teachers educate their students. Although increasing steadily the use technology in classrooms is still quite low. From a student survey by the NFER only a mere 20% of students claim to use computers to solve maths problems, see figure 1.1. It should also be noted that this is the figure for schools that were chosen to pilot the Project Maths syllabus, who have been shown to have a higher use of technology than others. These other non phase-one school’s student’s claim only 12/13% use computers to solve maths problems. In order to find out why we first need to look at classroom structure itself and the type of teachers we have.
Figure 1.1: Percentages of students taking part in given activities.

I believe a large issue in the take up of technology lies with a sizeable proportion of teachers being unfamiliar with it and what they can do with the systems available to them. In a survey of teachers over 60% claim to use laptops or pcs at last once a week with only about 12% reporting that they never, or hardly ever use pc’s/laptops (NCCA, 2012). So why then is there such a discrepancy between student’s opinions on technology use versus the teachers? I believe this calls into question not only the fact there are quite a large amount of teachers not using technology in the classroom but also the efficiency of the use of this technology by these teachers. Like almost any profession there is a huge spread in the age, experience and type of teachers in the country. It is very easy to place the blame on older teachers who aren’t as familiar with pcs, the Internet and other tech but I think that with newer waves of young teachers taking up
roles in teaching there needs to be a serious look at how technology is used rather than just boosting the numbers of teachers using it.

In the same report cited above (NCCA, 2010) we also see correlations between schools who were initially chosen to trial Project Maths and a higher use of technology in classrooms. While I believe this doesn’t in itself represent an issue I think it highlights the fact that there may be a problem further down the line of creating wealth gaps between richer and poor. A big concern for the take-up rate of ICT in classrooms lies in the fact that many schools have limited budgets and may not be able to afford or supply their students with resource intensive technology. I think it is currently, and for the foreseeable future a big barrier for technology in classrooms. However, I think that as Project Maths develops we will see more grants and funding becoming available for technology in schools.

The previous paragraphs have dealt with the lack of technology in classrooms from more of an Irish standpoint. However, from a more global point of view perhaps the largest barrier of all to edTech is absolute conclusive proof that ICT improves the quality of the learning experience. What seems to have happened up to this point is technology has been put straight into schools, for example a whole computer room, but students get at most an hour a week to use them. While there is technology present in many schools its use isn’t thought out properly and so in effect it doesn’t solve anything. Things are beginning to change however. It is clear that the Internet and computers open up a huge selection of content to students but this alone doesn’t ensure better learning. Many of these resources are spread across the Internet and can be quite difficult to find. The quality of these resources also has to be quite high as engaging students is difficult but holding their attention can be even harder again. When used properly however, technology in classrooms gives students a chance to explore, at self-paced speeds, engage with the chosen topic and most importantly learn.
1.3 The Book

The printing revolution began in the late 1400s and within thirty or forty years was responsible for more than a thousand print shops and the printing of over 10 million books (SF Books). The reason for this was the invention, and widespread adoption, of the printing press, a machine that allowed you to arrange metal molds of letters, cover them in ink and transfer the text onto paper. The man credited with its invention was Johannes Gutenberg. Although similar inventions had been created in parts of Asia and Egypt it is said Gutenberg came up with his invention independently. Before the printing press, often hailed as one of the most important ideas in modern human history, all books were transcribed line by line, usually by monks. Thanks to the invention of the printing press books became widespread allowing ideas to flow across Europe and even across continents.

And so books and paper became the predominant way of sharing information and like before, nothing really changed for hundreds of years bar small incremental improvements. That is until the invention and adoption of the World Wide Web, which allows us to share information at an unprecedented level and rate. Though not a ‘book’ in the conventional sense the web allowed us to pass all kinds of information from point to point. Grown up alongside this has been the prevalence of multimedia. Though photographs had been round for over a hundred years they, like books, were still printed on paper. The web as it developed and got quicker started to support the widespread sharing of images, and subsequently video and audio. Today we need only look at the likes of Facebook, YouTube or Spotify to see the level of use that multimedia has gotten to.

Finally the most recent addition to sharing of information, utilities and services is the mobile app. Popularized by apple on their iOS platform in 2008. The number of apps downloaded in 2013 reached over 102 billion (Lunden, 2013). Around the same time as mobile apps were being introduced Amazon brought out their first eBook reader, The Kindle. Although the idea for an electronic book wasn’t particularly new, the kindle, alongside some other eBook
readers, caused the format to take off. However, even now, the lines between eBooks and apps haven’t crossed. eBooks still remain a collection of pdf or XHTML files, which don’t take full advantage of the capabilities of the resources they have, i.e. multimedia and the Internet. As it stands, especially with textbooks versions, eBooks are just text, occasionally with some images but nothing more.

**Tablet Development Issues**

2.1 Lack of Standardization

Like anything in the programming world there is never only one way to do something. Unfortunately for developers that, almost always, means trade-offs have to be made along the way. And of course tablet development is no different. At the high level scale there are several choices in OS: iOS, Android and Windows Phone being the most common. Choosing which of these to develop for is often a tricky task as you have to consider not only number of users and who will have access to your apps but also what array of hardware you will have to consider. This all adds to the difficulty of developing for mobile as there is no standardization across the different platforms and a programmer must gain expertise in at least 3 separate programming languages and environments in order to get their apps to the maximum amount of people:

- iOS: Objective C
- Android: Java
- Window Phone: C#

There are other solutions of course to make your app cross platform you could use something like PhoneGap or Titanium. These kinds of systems run web technologies like HTML and JavaScript inside an apps WebView. However, the issues with these are that because they are trying to work on all platforms they never seem to fit into the style of any particular device. They don’t feel like apps running on your phone but rather a website that’s been quickly retrofitted to fit your screen size. There are other emerging technologies in the space, such as Dart. Originally a Google project, Dart is looking to take on JavaScript in web functionality and due to the rise in mobile demand we can only assume they will
get better. For now however, a one-solution fits-all language still seems quite a way off, if it even ever appears at all.

2.2 Hardware Capabilities

Once decisions have been made on which platform, or platforms, to develop on there is the next crucial stage to consider: the hardware your app is going to be running on. In terms of iOS, though slowly growing, the number of different types of tablets is in fact quite small, with only around seven distinct models of iPad. On their website windows offer only slightly more with ten different tablets running windows 8. However, it should be noted that these are only windows 8 tablets there would be a much larger proportion in order to offer backwards portability and along side this, the ten that they advertise are made by seven different manufacturers. The problem gets much more complex for android however as there are countless different android devices on the market. According to OpenSignalMaps, (OpenSignal, 2012) there are over 3,900 different android device types made by nearly 600 separate manufacturers.

So we can see that when we step back from the different manufacturers and OS types there is a massive variability in tablet capabilities. This means there are certain considerations we must undertake when designing for tablets:

• Power: While some of the higher end devices might be able to crunch through programs at an acceptable rate we need to remember that some older tablets may not have the same processing capabilities.
• Battery: Always a concern for mobile devices. Is often directly related to power as the more intensive the processing the more battery the device uses up.
• Memory and storage: As with laptops and desktops the range in memory can be staggering however tablets are slightly more restricted usually with 512/1024 MB of RAM. With the exception of the iPad range lack of storage in older devices can be supplemented with external storage such as microSD cards so storage is in fact less of an issue.
• Screen size: With a multitude of different manufacturers building tablets for different reasons screen size varies wildly. For example iPad, used for
majority of household computing vs. iPad mini, which is designed more for casual web browsing and social media.

- Built in components: There is no way to guarantee that any given tablet will have the components. For example the Hannspad-SN1AT71B doesn’t even have a camera, something many consumers may perceive to be almost standard.

2.3 Interoperability of the Web and/or Webapps

There is often debate, on blogs, tech websites and newspapers, about whether a person could fully replace their home computer or laptop with a tablet. Sadly the answer, for now, is almost always now. There are many reasons for this, several of which are to do with factors outlined in the previous 2 sections, but I think one of the biggest reasons is the fact that with tablets you still can’t get the full web browsing experience.

Firstly, the mobile web; in order to combat the growing number of mobile users many developers have opted to design separate mobile versions of their websites, tailored for the smaller screens of smartphones. However as tablets are running mobile OSs they are often switched to these websites. This may be a small issue as users can click back to the desktop versions of the sites, however it is a key indicator that a tablet is considered as a different entity on the web than a laptop.

Secondly, following on from this there seems to be unclear discrepancies in what will run on tablet devices and what won’t. Where a simple download of software will work on a laptop or pc there is very often just no way to run applications on tablets if you have to download them, regardless of whether they are web apps or not. Put simply although huge strides have been made in recent years tablets are still not fully interoperable with the web and many web apps out there and until they are, tablets will be seen as lesser hardware in comparison to laptops and pcs.
Approach/implementation

3.1 Architecture and Function

We can see from the diagram, in figure 3.1, the way PhoneGap, and subsequently my app is structured. The app itself sits on top of a mobile OS, in our case Android. The OS links to the app through the usual APIs that also exist for native apps. For us specifically, we have the web app files in the form of: Index.html, styleSheet.css and main.js. Using the PhoneGap APIs we link to plugins such as media or the camera, but also to our custom plugin VideoPlayer.java. All of this is rendered in the WebView and appears on our screen a fully integrated application.

![PhoneGap Architecture](image)

Figure 3.1: PhoneGap Architecture

The entire point of the book for this project is simple; embed 2 leaning activities, provided for us by Aibhin Bray, into a tablet device. This is where we begin, with attempting to incorporate as many leaning exercises as possible into an app so that it all sits under one roof. The future goals for this app would be to eventually supplement these leaning activities with textbook like questions and maths problems, all of which would be relevant to the activity in question. Done this way we could allow for all of the benefits of the learning activities without students or teachers feeling like we have stepped off the track of the syllabus. We
could simultaneously give the students the freedom to explore and feel out the maths, while at the same time appreciating that, for the time being, they eventually will have exams to take and as such have to cover a set amount of material for the test.

The first of these activities is the catapult exercise comprising 5 chapters, which the students step through. The idea is that they record a video of a projectile in flight, analyze the trajectory by breaking it down to graphs and equations then see if this maths lives up to their real world observations. The beauty of this exercise is that the first thing it does is attack what I’ve called the context problem up to now. It does this by letting the students see the real world flight of the ball and how far it’s travelled and then they apply the maths afterwards, it’s a progressive process so it doesn’t seem to students like teachers are pulling equations out of nowhere. As all students are split into groups of only two or three, they all must participate in order to get the activity completed. This is especially important for students with lower levels of maths as not only are they challenged to learn and understand what they are doing but they are also given tasks, which they go on to complete. The activity works here at breaking down some of the negativity that maths is too hard and instead fosters confidence and self-belief for a student in their own abilities. In terms of the actual maths covered in this exercise, it covers: algebra, specifically quadratics and equations of the curve, trigonometry, in terms of the graph analysis, statistics, by way of the data collection and discussions over the graphs.

The second activity in the app is the Barbie exercise, which attempts to give a Barbie doll the most thrilling bungee jump without killing her. It works by recoding videos of a Barbie doll falling from a height while attached with rubber bands. For each video we string together more and more rubber bands and so her falls become longer and longer. The idea is we then take this information and graph it, generating a trend line so that the students can predict the best amount of rubber bands to use so that Barbie can get as close to the ground as possible, without hitting it and they then test this. Once again for this activity the strong emphasis on teamwork harbors creativity and confidence in students, as they
themselves are responsible for a good outcome to the activity. Also we see here again the fact that the real world observations come first then the maths is used to describe it, once again cementing context and purpose behind the activity. The activity equally covers a large amount of raw mathematics: statistics once again in the form of data collecting and representation, co-ordinate geometry of the line and idea how maths can be used to easily describe relations between different data sets.

3.2 Set Up

Once I had decided that a tablet app was the best approach for me there were several decisions to be made before I could begin coding up the book itself. Firstly was the decision of which operating system, computer languages and development environment to use.

iOS is apple's operating system for mobile devices. First released in 2007 it was developed in line with the iPhone. Since then it has also been extended for use with the iPad tablet range. For us, more importantly, the SDK was first released in 2008 and is updated usually with every release of the iOS, which is currently on its 7th version: iOS7. However to get access to some of the developer tools there is a subscription fee. Applications for iOS are written primarily in objective-C with the ability to use C/C++ for some parts. Not all submissions to iOS’s app store are accepted, as all apps must meet certain requirements and standards.

Android is Google’s open source operating system for mobile devices. Also released in 2007 Android has a multitude of manufacturers which use it’s software, some of the most successful being: Samsung, HTC and Sony. One of the most attractive parts of android for these manufacturers is the fact that they can tailor the OS to their liking, resulting in many different ‘flavours’ or versions of Android. All of the developer tools are free to download and anyone can submit an app to the Android Play store for others to download. One of the difficulties developing for Android the sheer multitude of device types, sizes and capabilities.
as outlined in the previous chapter. This issue is more difficult to overcome with Android than iOS.

There are other mobile operating systems in the computing world: Windows Phone and Blackberry’s OS. Both of these OS's were left out of the decision here due to the fact that they’re tablets are far less widespread than the other two. Subsequently the amount of developers and documentation on the Internet would be far less adding to a harder development process for myself without any real benefits.

Initially, my decision was to pick Android with a large developer community and easy to use, free tools it seemed like a natural choice. Also, the fact that I had some experience using Android was a key factor in the decision. However, after some discussion with others and a small amount of research I realized that there was a further option, PhoneGap. PhoneGap allows a developer to create apps that can work across all the different operating systems using web technologies such as HTML, CSS and JavaScript. It does this by running the code inside a device's WebView and making several API's available to use device hardware such as the camera or the accelerometer. This became the ideal solution for me as I could run the code on many different devices, keeping the functionality all while not deviating too far from the format that eBooks are usually in, HTML or XHTML.

So now that my decision had been made and the department had given me a tablet to develop on, a Samsung Nexus 10, I had to get a development environment set up. As I’d used it before and the PhoneGap documentation suggested it I decided to use Eclipse as my IDE. On the Android website it is possible to download the APK which contains eclipse and the Android SDK. In order for me to run HTML code on the tablet I had to create a folder called www in the assets folder of my project. Then in the onCreate() method of my MainActivity.java I simply had to write the line:

```java
super.loadUrl("file:///android_asset/www/index.html");
```
This passes control of the program to my index.html file by telling the native android WebView to run the HTML code in it. This will be expanded on in more detail in the next section.

Other notable details in the set up are; the file config.xml had to be added to res/xml/ in order to allow the app access to the hardware through the PhoneGap API as discussed earlier. And in accordance with this several changes had to be made in the AndroidManifest.xml to give the app permissions to use this hardware.

3.3 Beneath the Glass

Now that all of the set up has been done I’d like to discuss in a bit more detail the technologies I’m using and what role they play in the project itself. This section mostly just touches on the work that was done but to get an idea of the exact development timeline I have made an infographic placed in chapter 3.5.

Firstly, HTML or Hypertext Markup Language was created in 1995 and is used to create documents on the Web. It is used as the structure or backbone of a website. For the book I have 1 single HTML file: index.html. It is responsible for all of the elements that we see on the app (with the exception of tracker, outlined below). It works by allowing us to specify tags in which we house information, for example:

```html
<button onclick="webBrows();">Simulate</button>
```

This line allows us to create a button element that when clicked launches the webBrows() function in the JavaScript, the word simulate is also displayed on the button. Alongside this in the html we can call on external libraries, for example:

```html
<script type="text/javascript" charset="utf-8"
 src="javascript/cordova.js"></script>
```

This particular line of code is actually quite important for our app as it links in the JavaScript file that holds all of the PhoneGap functionality. My index.html file holds the information for each page in each chapter and is responsible for dividing what information is contained where. All of the text we see is held here along with all of the buttons and links. It should also be noted that this entire file was written by me.
Another web technology PhoneGap allows us to take advantage of is CSS or Cascading Style Sheets. Developed alongside HTML it allows us to specify the exact design of each element in a HTML document. An entire websites appearance and layout can be defined with just one CSS file. In this app the file is called styleSheet.css and it is used to define anything from the size and positioning of an element to its colour or font.

To complete our collection of Web Technologies we have JavaScript, a scripting language that allows us to inject some functionality into our static HTML. In this app the JavaScript provides the majority of the functionality. I have just one JS file in the project, main.js. Contained within this file is the functionality behind the HTML, such as; dealing with linking and button clicks. The entire algorithm for the paging is here too, which utilizes jQuery, a JavaScript UI library, to remove the class="current" from the page in view and set it in the requested page allowing it to be displayed by the HTML. Alongside this we have the PhoneGap API calls for the camera and a Web Browser window. Once again I have written all of this code.

Finally we come to Java which is the language predominantly used in native Android apps. Upon choosing PhoneGap, I had hoped that there wouldn’t be any need to write native code however, after some work I realized that some of the functionality that I needed could only be done using a Java plugin that I had to write myself. This is talked about in much more detail in chapter 4.1. This plugin was the tracker, the part of the program that divided up the frames in a video allowing us track the co-ordinates of the flight of a ball as time progressed.

All of these languages that were used here were a direct result of the decisions I had made previously, as detailed in 3.2. Personally I had some experience in Java, for Android applications, and minor experience in JavaScript, using Node.js for server side purposes. However, HTML and CSS I had never used before. It wasn’t detailed above but a factor in my decision to use PhoneGap was that I was very interested in learning more about these languages, as they are
extremely widespread in the computing world. This did mean of course that
there was a slightly sharper learning curve than if I had developed a solely
android app but I believe it was worth the effort.

3.4 Ux/UI/Security

I mentioned previously (section 3.2) that some CSS had been used to
format and set the layout, that was, unfortunately, the limit of work done on a
user interface. From the beginning of the project I was adamant that this app was
to be a prototype above all else. This means that there was minimal planning
done to the look and feel of the app and that functionality was more important
than all else. This means, unfortunately that this section will be rather short. The
colour scheme for the book I chose myself with the aid of a colour palette
website (http://www.colourlovers.com/palettes).

The same has to be said for the user experience, that ease of use of the
app has sadly taken a back seat to functionality. There was some thought put into
parts of the UX along the way though. There is an important concept in UX design
that if 2 things, for example 2 buttons, look the same then they should function
the same (Rose, 2013). On the back of this I have made a conscious effort to
standardize the use of certain HTML elements. For example: the pages. In the app
whenever we wish to click on onto a new page we have a piece of text saying
something like “Chapter 1:2”. In order to try and make the app more useable all
of these links to other pages are <a> tags in HTML. Using the CSS and the colour
scheme I mentioned in the previous paragraph I also assign every <a> tag the
RGB colour “#666666”. The idea here is that hopefully every time a user sees a
dark grey piece of text on a page, without having to read the actual words they
should have a fair idea that clicking here will change the page they are viewing.
While, admittedly this could have been implemented slightly better with a button
or an arrow it serves as an example of the line of thought necessary for good UX
design.

Also worth noting in terms of design is the implementation of the back
button. Due to the fact that I had to implement my own version of the paging
system the back button as it stood on it’s own would simply exit the app, as its usual function is to cycle back through previous Android Activities and we never switched from the initial Activity. In order to do this I created and array called previousPage. Every time a link was clicked to switch to a new page I would push the page value onto the array. I then simply wrote a small event listener so that every time the back button was pushed I’d remove class=”current” from the current page and assign to the previous page, then pop the current page off of the array.

3.5 MVP/Prototype/ Reflection on decisions

In the previous 2/3sections I’ve alluded to the fact that this project was developed primarily as a prototype or minimal viable product (MVP). This has lead to many of the decisions and trade-offs previously mentioned. Up to this point also I have only described what technologies I’ve used rather than how I’ve used them. So in order to show this I’ve made an infographic detailing the work done in each language to give an idea of the scale and exact contents of the project. See Figure 3.2.
<table>
<thead>
<tr>
<th>HTML</th>
<th>CSS</th>
<th>JS</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Hello World example</em></td>
<td><em>Set up of styleSheet.css</em></td>
<td><em>Main.js Initial function onDeviceReady()</em></td>
<td><em>MainActivity.java OnCreate()</em> hand over control to index.html</td>
</tr>
<tr>
<td><em>Linking of cordova.js and main.js so that PhoneGap and my own js can now run.</em></td>
<td><em>Sample function to test js functionality. Things like get phone imei etc.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>First few &lt;div&gt; One including image. Other including button for web browser.</em></td>
<td><em>functions: camera() takes and displays photos. webBrowse() launches web browser.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Chapter structure defined. jQuery libraries linked for use in js.</em></td>
<td><em>CSS to display only pages with class=&quot;current&quot; as defined by js.</em></td>
<td><em>Addition of jQuery library allow switching between pages implementation.</em></td>
<td></td>
</tr>
<tr>
<td><em>PhonGap API call to launch plugin VideoPlayer</em></td>
<td><em>VideoPlayer.java takes input from js and returns it’s own values.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Original colour scheme changed.</em></td>
<td><em>video() function passes videoPlayer a local video and processes callback, getFromGallery (source) function.</em></td>
<td><em>VideoPlayer.java used to play video in Android video player.</em></td>
<td><em>Dissects videos into frames.</em></td>
</tr>
<tr>
<td><em>All supplementary text for the activities added as &lt;p&gt; tags.</em></td>
<td><em>Takes information from plugin, recordXY(x, y) puts coordinates to array, equation() calculates quadratic.</em></td>
<td><em>playFrames() function creates new view containing video frames, buttons and calibration implementation.</em></td>
<td></td>
</tr>
<tr>
<td><em>plot() to display 3 separate graphs, questionTime() for multiple choice q’s in Barbie</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 3.2: Infographic to show individual tasks*

So now that we know exactly what went in to the project and all the considerations that were taken throughout the process what was the final
product? By the end of development we now have an app that embodies 2 full and separate learning activities. The app itself was developed in collaboration with the Centre for Research in I.T. in Education (CRITE). The content for the activities was provided by Aibhín Bray and alongside this Ger Sharpe and Brendan Tangney gave consultation and advice.

We will begin with the catapult activity as outlined in section 3.1. The students take the tablet outside to do this filming in a nearby screen or open area. We then click onto the next chapter, which encases to analysis of the video itself. This page contains a small amount of explanation text and 2 buttons. The first, ‘Gallery’, allows us to pick a video from our tablets gallery. Following this we click the ‘Frame Analysis’ button that launches the Java plugin that we often refer to as ‘tracker’. This allows us to step through the video frame by frame, each time placing a cursor over the projectile’s position. The program records all of these co-ordinates. This part of the app also calibrates the video so as to map the screen co-ordinates to real world ones. It does this by asking the user to mark a known horizontal and a known vertical distance then converts all subsequent screen co-ordinates to real world ones. Once this part has been done we click the finish button, that window closes and we can click onto the next the chapter. In this chapter we have some text asking some questions of the students and there is a button, which will display 3 separate graphs: horizontal distance travelled vs. vertical distance travelled, horizontal distance travelled vs. time and vertical distance travelled vs. time. Before they click this button to see the graphs however, this is a place in the activity where students pause for a while to discuss what they think the graphs themselves will look like. Once this discussion has taken place the students can view the graphs to see if they were correct then click onto the next chapter. Here, now that they know what the graphs look like again, the students pause to discuss what sort of equations would be suitable for the graphs. They are then given the equation of the line for the horizontal distance travelled vs. vertical distance travelled graph, which has been calculated in the backend by the graph using quadratic regression (Had2Know, 2010). The students now break from the app to an external application, GeoGebra (Hohenwarter, M., Hohenwarter, J., Kreis, Y., & Lavicza, Z.
2008), which is a graphing tool. Here the students can get the angle at which the projectile was launched and find out the horizontal distance travelled, according to the maths. Finally the students return to the app and click to the final chapter. They then click the ‘Simulate’ button that launches an external, flash, projectile simulation (Phet, 2009). Using what they have calculated up to now the students can see if the maths and what they have observed in the real world match up and hopefully it will.

This next section of the app, the Barbie Activity consists of just 3 chapters. Firstly as with the catapult, the initial chapter is just a small amount of text describing how to set up the activity and progress through it. The next chapter we have some small amount of text offering further instruction. There are also 2 buttons, the first of which we use to select which video we wish to use from the tablets gallery. The second button is the video analysis button, which launches the Java plugin as before however this is a slightly different implementation than the catapult version. Here we firstly calibrate, this calibration will be the same across all further videos in this section so there is no need to do it multiple times. We then mark the initial launch point of the Barbie doll, use the slier to scrub through the frames until the Barbie is at its maximum distance, open the dropdown to record how many rubber bands were used in this video and click finished. We repeat this step for each Barbie video recorded. Once this step is done the students can click onto the next chapter. This chapter is our final one for this activity. It starts with a multiple choice question which asks what kind of graph do the students think would be most suitable for the information they have, as before this would act as a point for the students to pause and discuss the question. Once they select the right answer the graph of distance travelled vs. number of bands appears, along with a trend line. From this the students can then predict the maximum amount of bands needed and test this out to see if that maths matches up with the real world.
**Figure 3.3: Catapult activity**

**Figure 3.4: Barbie Activity**
Issues in Software

4.1 Plugin Issues

Discussed earlier (section 3.2) was the discussion to use PhoneGap in order to be able to develop a quick and easy hybrid app that might work across platforms. Initially this seemed like a sensible and clever idea. However it was only after actually starting writing code for the app that I realized this solution was not as simple as it had first seemed. The problem first arose when I was testing to see if I could simply add all the different types of media I wanted in the book. Adding photos, text and linking to webpages was actually relatively straightforward as this was just either simple HTML or basic use of the PhoneGap API. However once I attempted to embed video into the HTML it wouldn’t display. After a large amount of investigation I discovered that video playback was not possible solely through the HTML and JavaScript, as I couldn’t access the Android video player. Attempts to solve the issue included:

- Trying to use the <video> tag however this is only supported in HTML5 which PhoneGap does not yet take full advantage of HTML5 functionality.
- Trying to use the <iframe> tag in the HTML, however once again PhoneGap didn’t support it.
- Trying to find or create an API call that would access the video Player from the JavaScript.
- Using the working API call to open a webpage. This actually worked, as I was able to play YouTube video in the webview. Unfortunately this would have only been a partial fix as I would have been able to supply content from the web to the video but I wouldn’t have been able to analyse or dissect it for the activity, which was the initial point of having the video.

So these were all of the initial attempts to fix the issue but they all came up short. In the end I came across a piece of PhoneGap documentation (PhoneGap) that explained a way to access extra functionality of a device that would be otherwise unavailable to a WebApp. It described how to write a JavaScript interface that would call some native code, which might access this excess
functionality. It works by the following function call (taken directly from my main.js file):

```javascript
cordova.exec(callback,
    function(err) {callback('Nothing to echo.');},
    "VideoPlayer",
    "playFrames",
    [str]);
```

Where:

- `callback` is the success callback, a separate JavaScript function.
- `function(err)` is the error callback, which in this case just passes back the string 'nothing to echo'.
- "VideoPlayer" is the Java class name for this plugin.
- "playFrames" is known as the action. It is simply a string that specifies for the plugin what method we wish to call.
- `[str]` is an array holding JavaScript Object Notation (JSON) objects. The code uses this to pass information between the plugin and the JavaScript.

All of this simply equates to the calling of a native method `execute()` in the plugin, the rest is just information on where this method is and what to do with it once we find it. All of this was done in accordance with the documentation but for some reason wouldn’t work, i.e. the execute() method in the java wasn’t being called. Initially I checked that everything was written correctly, logging outputs in the JavaScript to ensure everything was working as I had expected it to. However it wasn’t until a day later and a lot of inspection that I discovered I needed to put this line into my config.xml file:

```xml
<plugin name="VideoPlayer"
    value="com.phonegap.plugins.video.VideoPlayer"/>
```

This line of code tells the project that I am using the class `VideoPlayer` as a plugin and is what the PhoneGap API uses to run the execute Method. From this
point on I was now able to write native Android code, which allowed me greater access to my devices capabilities.

4.2 Tracker

Now that I have described the plugin implementation I’d like to discuss what it has been used for in this app. Initially I’d written it simply to utilize the Android video player that the PhoneGap API didn’t have access to. However, once I began trying to assemble the learning plans I came across a rather large issue; the original learning plan uses an application called tracker (Brown, 2005), which is an open source video analysis tool. I had planned to embed this straight into my app but despite the project being open source it was far too large and complicated to add to an Android app. The next option was to use what they called their ‘Web Start Tracker’, which meant running the tracker software over the web with out having to install it. The problem with this was in order to even start the program I had to download it first and then subsequently run it but this would not work on the tablet. I looked through the Google Play Store and elsewhere on the Internet to see if there was software like it but didn’t find anything suitable for the activity. So unfortunately I was left with only one real option and that was to try and develop a simple version of the tracker software myself.

The first step to get the tracker software was to break up the video into separate frames and be able to cycle through them. The breaking up of the video itself was actually quite simple I just needed to call the function:

```java
metaRetriver.getFrameAtTime(t,
    MediaMetadataRetriever.OPTION_CLOSEST);
```

This line as it says lets us get the frame at a certain time with the first parameter being the requested time and the second is just an option to do with how close to the time we want the frame or if there is some leeway. With this in place we just need to increment the time variable ‘t’ and re-call the function to advance a frame (in the book we increment ‘t’ by 1/3 of a second each time we update). The real problem for this step however, was accessing the gallery to get one of the videos recorded and then getting this information to the plugin. The solution
actually lay in an API call. In PhoneGap there’s a function called
camera.getPicture(), which allows us take a picture with our devices camera.
After a lot of investigation I found out that it is possible to use this same function
to access the devices photo album by passing the following as an optional 3rd
parameter to the getPicture() function:

destinationType: destinationType.FILE_URI,
sourceType: Camera.PictureSourceType.SAVEDPHOTOALBUM,
mediaType: Camera.MediaType.VIDEO.

These statements allow us to now open the android gallery and select a video’s
file URI, which we store as a variable. The following step was to pass this video
onto the plugin and although the URI passed through seemed to be the exact
same as in the JavaScript the video’s first frame was not showing. The problem
here was that the Java needed a full path to access the video but was only getting
the URI. In order to combat this I have a function, which converts a URI to a
String containing the path, yet the video was still not displaying. What had to be
done was the following:

```
url = url.replace("\\", "");
url = url.replace("file://", "");
```

The problem was the path contained these characters that were throwing the
function. Once these were in place the video could open and we could cycle
through the frames with ease. Now that we could cycle through the frames I
simply had to record the position of a cursor on each frame and pass that back to
the JavaScript to be recorded.

4.3 Barbie Tracker

The previous paragraph outlined how the tracker part of the app was
created, initially this was solely for the catapult activity but once it became
apparent that the software used for the Barbie activity was also not compatible
with the tablet the tracker functionality had to be retrofitted to accommodate
them both. My logic was that the best way to tackle the problem was to abstract
the entire playFrames function into the smallest pieces I could so that I could
reuse most of the code without parts becoming redundant. This, however, was
much more difficult than initially anticipated. The main reason for the difficulty
was the fact that the majority of code had to be run on Android’s UI thread to be
displayed. In order to do this I had to place most of the code inside a runnable function and tell it specifically to run on the UI thread which places certain limitations on the function. In the end I left most of the set-up code from the playFrames function and transferred the runnable to a new function, catapultRunnable() subsequently creating a barbieRunnable() for the UI elements in the Barbie activity. While this was good for my program itself it is also considered good practice as it makes code more maintainable and readable.

The final addition to the tracker development was the calibration I had to implement so that the co-ordinates the students mark for the ball correspond to their real world distances. This was much more of a maths problem than an issue in software so the fix in this case was actually relatively simple:

```java
float proportion = (float) (screenWidth / Math.sqrt(Math.pow((double)(caliZeroX - caliHorizX), 2.0) + Math.pow((double)(caliZeroY - caliHorizY), 2.0)));
realWorldWidth = proportion*15;
```

These 2 lines of code generate a variable, ‘proportion’, which as the name suggests is the proportion of the screen width that is taken up by our known horizontal length. In our case the known horizontal length is 15 meters therefore once we multiply the proportion by 15 we know the real world width contained in our video. For example: if the known horizontal length spans 1/3 of our screen’s width (proportion = 1/(1/3) = 3) then the width of the shot is 45 meters (15 * 3).

### 4.4 Other Important Issues

As much as I’d like to say the previous two sections entirely covered all the problems that I encountered in the project, they sadly did not. An issue that seemed to affect the project from beginning to end was actually something I was aware of from the offset, that tablets are still quite a way off being fully interoperable with the web and with web apps. Despite being initially positive that I could embed existing applications into the app I wasn’t actually prepared for how difficult this would be. One key example of this was in the projectile
simulation, seen in the catapult activity. The simulation itself ran on Flash which as far as I was aware Android supported. However, after it became apparent that the simulation wasn’t running I found out that Flash is no longer supported on Android. I believe this is due to HTML5 taking over as the predominant technology on mobile devices. There was, in the end, a simple fix: download the last version of Adobe Flash made for Android from the list of Archived Flash builds. However, this fix, while it solved the problem, is no better than a band aid for the time being as once Android releases its next big software update there will be no Flash for it. Also it would require any user to download the archived Flash versions as well, which is not scalable.

Another issue that was difficult to overcome was the use of a spreadsheet, and its graphing capabilities, to record and analyze the data from the Barbie activity. Initially I tried to pass information to a Google Doc Spreadsheet using Google’s API. Although the code isn’t actually included in the project I did manage to get the data to pass through to a spreadsheet. However, the mobile version of Google Docs is actually much more basic than their browser version itself and so I was caught in a catch-22 situation: The mobile version wasn’t equipped with graphing capabilities and the browser version was unusable with a touchscreen interface. Naturally I searched the web for other spreadsheet applications but no others fit the purpose I was looking for. The reason we wanted to use a spreadsheet in the first place was to allow the students to experiment with different graphs to find the correct one to use and upon finding the correct one experiment with lines of best fit. Therefore in the end I decided to implement my own graphs, using the graphing library JQ Plot, and try to imitate the learning experiences that the students would’ve gotten from using the spreadsheets. This was done by hiding the graphs until the students had answered a multiple-choice question on which type of graph would be they thought the best. This would also be supplemented by a discussion in the classroom. Then, once the correct answer is selected, the graph is displayed along with the trend line.
The only other notable issue to mention is the algorithm used to fit an equation to the trajectory of the ball. All of this was done in the JavaScript every time a co-ordinate of the ball was passed back from the plugin it was passed to the recordXY(x, y) function, which totals up all the following values: x, y, x², x³, x⁴, x²y, xy and n, where x is the horizontal displacement, y is the vertical displacement and n is the number of points. Once all points have been taken we then call the function equation(), which uses the quadratic regression formula to estimate an equation for the parabola using matrices as defined in the picture below in figure 4.1.

\[
\begin{bmatrix}
\sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\
\sum x_i^3 & \sum x_i^2 & \sum x_i \\
\sum x_i^2 & \sum x_i & n
\end{bmatrix}
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
= 
\begin{bmatrix}
\sum x_i^2 y_i \\
\sum x_i y_i \\
\sum y_i
\end{bmatrix}
\]

Figure 4.1: Quadratic Regression Matrices

There were many other problems faced throughout the project, the ones above were just the longer ones that took more time to fix. The other include:

- Plotting of graphs: mentioned above, the solution was to use the JQPlot external library, which was actually intuitive and easy use after some initial research.
- Paging and back button: Had to go completely with my own implementation of switching back and forth through different parts of the HTML. Problem was solved by firstly loading the entire html up front, this is a small optimization for speed, then switching class="current" between <div> tags on a link press. The CSS then uses this current value to only display certain parts of the html, the current page.
- Integrating with web apps has been mentioned above, however there was also the issue of integrating other native apps into the project. This was actually solved for the MobiMaths app, which no longer appears in the project, however I haven’t been able to get it to work for GeoGebra
due to access permissions and not having their apps namespace. The initial solution was writing a plugin, called IntentPlug.java that fired an intent to launch an Android Activity in the external app.

**Evaluation**

**5.1 How the Evaluation was carried out**

The entire evaluation was carried out in conjunction with CRITE alongside Aibhín Bray. The actual evaluation itself took place in oriel house, which is the base for Trinity's Bridge21 scheme with four transition year students taking part. Due to time constraints we decided that trialing just the catapult activity would be sufficient for evaluation purposes. So for this section I'll discuss the evaluation itself and what happened. Then in the next section I'll highlight how, given more time I would address any of the issues or problems found in the evaluation.

The activity started with us dividing the students into two groups of two and running some quick brainteaser games like how to cut a piece of paper so that you can step through it and how many seconds have you been alive? Following this we gave the students the brief for what they had to do, namely use a catapult to fire goods into the courtyard of a prison where a friend of yours is locked up. We then laid out the equipment for the activity and let the students choose what they wanted to use, in this case it was simply the catapults and a ball. Merrion Square Park was the closest open space to us so it seemed like a natural choice to film the trajectory of the ball. As was the approach throughout we as the instructors would occasionally give hints as to the best way to approach certain parts of the activity. The filming of the catapult is a good example of the type of hints we were giving. For example: at one point we asked the students how far back they felt they needed to stand in order to get a shot, this was done to try and get them to consider the fact that the entire flight of the ball needed to be shown in the video. Both teams took turns shooting the ball across the green and made sure to aim for the target at 15 meters. Once both teams were satisfied with their videos we went back to the room and fired up the
app itself. Guiding them through I asked them to click through to the initial chapter in the catapult activity and read through the initial set-up text. It was here that we identified the initial problem. The students made the point that some of the language was too difficult and that they weren’t familiar with terms like projectile, trajectory and parabola. The next step was to launch the projectile simulation, this also brought up an issue but one that was known before hand. The issue in question was, as mentioned earlier, the fact that the simulation was running on Flash, which was not meant for the current version of Android on my tablet. So the actual simulation, despite being functional, was very buggy and slow. As a direct result of this the students found it quite difficult and frustrating to use.

After this we took a short break and once they returned we got into the actual activity itself. They all clicked through to the next chapter and began analyzing their chosen video. There were some problems here, once again though; all of these were known issues. Firstly was the fact that they had to click through a large number of frames until just before the ball was launched from the catapult which was time consuming and honestly boring. The second was the calibration implementation, the students were able to do with me stepping them through it but said they would’ve found it quite difficult to navigate. Once they understood the method however there was no more problems with the tracker functionality. We clicked through to the next chapter and paused to discuss what the graphs that would look like before showing them on the app. On the next chapter we paused once again to discuss equations of a line, quadratic equations and how they were affected with manipulation of their coefficients. All this part actually went quite well and the discussions were interesting, especially with Aibhín introducing the concepts of correlation and causality in graphs and statistics.

The next step was to copy the quadratic equation across to GeoGebra for graph analysis to get the angle of the launch and the horizontal distance travelled. All of this went according to plan as we worked through the angle calculations and the maths behind it. However upon further inspection we
realized that the quadratic on the graph was actually wrong as it was saying the distance travelled by the projectile was over 100 meters. Initially I assumed something had gone wrong with the calibration step but it should be noted that when I ran the activity again while the students were at lunch it did actually produce an almost correct graph, once the c value in the quadratic had 15 subtracted from it then graph matched up with our real world observations. In order to give the students a fully rounded learning experience and as the simulation was buggy on the tablet we decided to finish out the last step on a laptop. Having recomputed quadratics for the flight of the ball, we ended the activity to see who had most accurately mapped the flight of ball by running the simulation with the student’s values. Though it was close the winning team were delighted to have gotten the upper hand on their friends, actually cheering for maths, which made it all worth it. From there the one last step was to hand out the system usability scale questionnaire that will give us the data for section 5.3.

5.2 Issues with Evaluation

The first problem we encountered in the previous section was the complexity of some of the words used in the description of the activity and set up required to get started. A quick fix of this would be to give a simpler word beside it, for example: “to video-capture the trajectory (flight) of the object”. This would be beneficial as not only would it cut down on confusion but also might add to the learning experience as the students may learn some new terminology. However, I think an even better solution to this issue would be to replace the majority of the text with diagrams and pictures, possibly even with a link to a video that runs through the activity. I think in this way we could accommodate a far greater number of students in using the app. Facilitating auditory, visual and, once we begin the activity, kinesthetic learners.

The next issue we faced in the evaluation was the using of the projectile simulation. To an extent this issue was known of before we ran the evaluation. Currently there is no real fix to running the buggy Flash more efficiently. A possible work around would be to use the simulation on a laptop or computer;
this however, goes against the entire purpose of this project. For now though the only real solution is to either wait for or develop my own HTML5 simulation, or possibly a native simulation written in a plugin. For the scope of the project, neither of these solutions was plausible, as they both would have taken up large portions of time to develop simply to just add some extra usability into the activity, which, as discussed in section 3.4, was not a priority for the project.

On returning from their break the student’s next issue was: having to click through many frames in order to get to the part of the video where the projectile was actually being launched. While frustrating for the students there was no fix for this problem that would not make the system even more complicated still. Like the simulation problem, this was an issue I had been aware of before the evaluation was carried out. Were I to continue on with the project I would attempt to solve the problem one of two ways. Firstly, try and find some Android app, which would allow us to edit the video. This way we could cut out the initial redundant part of the video with the advantage of adding to the learning experience of the activity as many of the students may not have seen or used video editing software. However, this would once again be external software meaning that the students would have to leave the app in order to use it, which goes against the idea of an all inclusive app. The second possible solution would be to develop a hybrid version of the tracker combining parts from both the catapultRunnable() and the barbieRunnable(). What I mean by this is that we could use the slider, seen in the Barbie tracker, to scrub through the frames until we reached the part of the video where the projectile was being launched. From here we could then resort back to the previous method of recording the trajectory of the projectile.

The final big issue brought up in the evaluation was something that I had not come across when running the activity myself during the testing, namely the fact that the proportions on the graph were off by a factor of 4 or 5 times. As mentioned in the previous section when I re-ran the activity myself I was able to get a correct graph. Though I am still unsure as to what exactly lead to the students getting the wrong equations of the quadratic I have several theories.
Initially I assumed something had gone wrong with the quadratic regression formula in the backend (the function equation()). However, this had been working correctly from when it was first written and had not been changed since, I have subsequently tested this function and can confirm it still executes correctly. My next thought was that there had been a copy and paste issue in taking the equation from the app to GeoGebra, however this was rechecked and had been carried out correctly. The only real explanation was that something had gone wrong with the calibration step. However, since I tested it at the time of the evaluation and have also run it again more recently to be sure, I can confirm that the calibrate function itself and the maths it contains is correct. What I have concluded is that the errors we saw in the evaluation are a combination of several smaller bugs:

- In order to let the students all gain proper benefit of the activity we had to split them up into teams, this meant a second tablet. I had some difficulty finding a spare one to use and so only managed to get the second tablet the night before the evaluation. Therefore there wasn't a chance to carry out a full test of the activity on this tablet. What I hadn't taken into consideration was that the proportions of the screen was different and so some of the calculations if the (x,y) co-ordinates of the projectile were incorrect. A fix for this is to simply remove any 'magic numbers', which are numbers only specific to a certain device or system, from the code. This step has been carried out.

- Originally before the evaluation there were some issues with the numbers being outputted for the co-ordinates. Although I was under the impression that it had been fixed the system seemed to be adding a value 15 to each y co-ordinate. This had to do with how I had been calculating the height of the video frame. This has also now been fixed.

- The final issue that I believe to have been a factor in the calculation error was actually the UI. At the time of calibration the students had said that they had found it difficult to use. While I can't be sure I think due to lack of an intuitive calibration implementation the students may not have actually calibrated the system correctly. As it stands I believe it is plausible that one of the students may have clicked re-calibrate but not
actually carry it out. Resulting in the misrepresentation of all future values of x and y. An ideal solution, which I had looked at but was taking up too much time, was calibrating by drawing lines for the known horizontal and vertical distances. Keeping these lines visible for the remainder of the tracker use would have cleared up this problem, as students would be sure of their calibration.

5.3 SUS Scores and Interpretation

While on initial viewing it seems like there were quite a lot of issues with the evaluation itself there are several things to be noted. Firstly all of the problems we had, with the exception of the quadratic calculation error, were actually bugs that I had been aware of from before the test. Alongside this there is a fix for each of them, some of which have already been implemented in my program. I think it’s also fair to say that the root of these errors is the lack of UI/UX planning. While from the point of view of evaluating the app as a software project, I appreciate that this doesn’t negate the fact that there were issues but I just felt it was necessary to include.

As mentioned in 5.1, at the end of the evaluation we asked the students to fill out a questionnaire on their thoughts about using the app for the learning activity. The questionnaire used here was the System Usability Scale (SUS), see Appendix A. Created in 1986 it has become the industry standard for testing usability of any system. It works by asking 10 questions, all of which are answered on a scale of 1 to 5, where 1 is strongly disagree, 5 is strongly agree and 3 is neutral:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

In order to calculate the results: all questions with an even number are given the value of 5 minus the users response. All odd numbered questions are calculated by taking 1 away from the users response. The reason this is done is to avoid answering bias towards either positive or negative responses. We then add these up and multiply by 2.5 to put the results on a scale of 1 - 100 as opposed to 1 – 40. I feel it should also be noted that while the scores are marked out of 100 there are not percentages. As can be seen in the graph below, figure 5.1.

![SUS Score graph](image)

**Figure 5.1: System Usability Scale Scoring**

Alongside these 10 SUS questions 4 more extra questions were asked:

11. I liked using these apps for learning mathematics.
12. Using these apps for learning mathematics is worth the extra effort.
13. Mathematics is more interesting when using these apps.
14. The apps helped me to better understand the mathematics.
Interpreting the scores once we have them can be tricky but I will outline my results. As this was a small test group (only 4 people) results may not be as accurate as if we had run this evaluation multiple times. However, when the totals of the 10 questions were calculated and subsequently added up there were 2 scores of 70, one of 67.5 and one of 62.5. Taking the mean of these 4 values we get a total SUS score of 67.5. According to the literature, and the graph above, the average SUS score is 68, which would mean that our score for the evaluation is almost exactly average in terms of usability. I think that this is actually quite a startling figure, especially when you consider that fact that ease of use was not a major concern for the project. Were this a consumer product being readied for market it would be very likely that there would have to be some redesign of parts of the system, however, for a prototype project I believe this is an excellent score. It seems that, although the students found some parts confusing, they responded positively to the functionality of the system. I feel these SUS numbers, especially when backed up with the four extra questions, also clearly represent a hunger for these types of learning plans and activities, as they provide insight into how much the students benefitted, engaged and enjoyed the activity.

From my point of view, in software development, there are few things that give me as much excitement as the idea of taking something from purely being a idea all the way through to making it a tangible end product and putting it in someone’s hands. From that standpoint I thoroughly enjoyed the evaluation process itself. Although the fear of everything going wrong and breaking did dull this excitement slightly, I felt like the students we ran the activity with were excellent in terms of their engagement and enjoyment of the activity. For me it really did remind me of how important this line of work can be.

**Conclusion**

Although, like any software project, this app has annoyed and irritated me at times, I have genuinely enjoyed to process and the progression from the forming of an idea all the way through to it’s actual creation. The only question that remains here is possibly the most important of all, what has been achieved?
6.1 Stem Crisis

From the beginning we've discussed the main issues that continue to plague the maths syllabus. The first of these mentioned was the lack of context. With this app students participating in the learning activity get to actively model the real world. The question of context never occurs as, simply put, it is at the center of these learning plans. The app addresses the issue of rote learning by never making it an option, instead posing discussion between a class of students where, regardless of level, they all participate. The platform is set for this app to be fully integrated with the Project Maths Syllabus as it is, supplementing the syllabus' ideals of exploration of concepts and context driven mathematics. For the record, I'm not claiming that were this in schools it would fix the entire STEM problem, but I firmly believe the problems it addresses are the fundamentals that need to change in maths education.

6.2 Technology in the classroom

Love it or hate it, mobile technology is, and will continue to grow in the foreseeable future. As mobile devices continue to saturate the consumer market, proficiency in the use of them will only increase. The beauty of a tablet solution, in relation to the issues posed in the introduction, is that when designed correctly they are actually much easier and more intuitive to use. With better solutions we can envisage more and more teachers integrating technology with the classroom. The app itself by virtue of being a hybrid WebApp will actually look the same across different platforms once its functionality is expanded. This gives an originally unforeseen advantage to teachers in that it will look the same to all students, regardless of device, facilitating easier instruction. The barriers technology in classrooms face are mostly generational and some will pass with time. The rest however, can only be changed by continued research in order to prove its benefit. Projects like this, and the feedback we get from them, all form stepping stones towards a greater understanding of the role technology can play in education.
6.3 The book

In its current state, this project does appear to be more application than book. The reason for this is solely to do with the fact that MVP was prioritized above all else, i.e. it had to work before anything else could be done. However I hope that it isn’t hard to see that with the simple addition of some text we could add full question and problem sets to the activities. Other functionality could be added to allow a teacher to specify questions to be added, the possibilities are endless in this regard. To phrase it correctly, all of the components have been made, manipulated and tailored to create a MathsBook 3.0.

6.4 What Have I Learned?

A tricky question for anyone, what have I personally have learned from the entire project? Having never written such a long and formal paper I actually found this entire process difficult but rewarding. As alluded to before I love the idea of having an idea and then, with some work, seeing it before you. From a technical point of view I have gained a huge understanding of HTML, CSS and JavaScript, all of which I had minimal experience of. In projects normally I tend to shy away from development decisions like what technology or system to use, so I liked the fact that I was pushed to make a decision as it was my project and my responsibility, from this I’ve learned to trust some of my intuition and that even if a bad choice has been made, it is always fixable. I’ve always had the impression that I am, at least, a good problem solver and having done the evaluation and the project in general I have gained a lot of confidence in my abilities. Finally, and most importantly, while doing the evaluation and seeing the creativity and excitement it created in the students I re-learned the reason I wanted to do this project in the first place: To try and pass on the marvel, wonder and beauty of mathematics.
Bibliography


(Rose, 2013) http://www.freshconsulting.com/homepage-ui-design/. UI design

(Had2Know, 2010) http://www.had2know.com/academics/quadratic-regression-calculator.html

Hohenwarter, M., Hohenwarter, J., Kreis, Y., & Lavicza, Z. (2008). Teach-
ing and learning calculus with free dynamic mathematics software GeoGebra. In 11th International Congress on Mathematical Education. Monterrey, Nuevo Leon, Mexico.

(PhoneGap) [http://docs.phonegap.com/en/2.7.0/guide_plugin-development_index.md.html] Phonegap documentation

(Brown, 2005) [https://www.cabrillo.edu/~dbrown/tracker/], Tracker Software.

Appendix A: System Usability Scale

Name:
Gender:

<table>
<thead>
<tr>
<th>Strongly</th>
<th>Strongly disagree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I think that I would like to use this app frequently</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. I found the app unnecessarily complex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. I thought the app was easy to use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. I think that I would need the support of a technical person to be able to use this app</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. I found the various functions in this app were well integrated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. I thought there was too much inconsistency in this app</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. I would imagine that most people would learn to use this app very quickly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. I found the app very cumbersome to use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. I felt very confident using the app</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. I needed to learn a lot of things before I could get going with this app</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>I liked using these apps for learning mathematics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Using these apps for learning mathematics is worth the extra effort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Mathematics is more interesting when using these apps.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>The apps helped me to better understand the mathematics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>