Comparing Native and Cross-Platform Development Tablet Environments Based on an Application for Autism.

by

Laura Seletos

Advisor

Michael G. Branton

A senior research proposal submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in the Department of Mathematics and Computer Science in the College of Arts and Science at Stetson University Deland, Florida

Fall and Spring Term

2012-2013
Software development on tablet devices is very important. The leading tablet devices on the market are Google’s Android tablet and Apple’s iPad tablet. We will be comparing both native environment development and cross-platform development through the design and implementation of a tablet application for Autistic children. Autistic children show drastic improvement in school with the assistance of tablet applications. Utilizing both the iPad and Android tablet’s features like animations, touch sensors, button interaction and sound output all help to hold the attention of Autistic children.
# TABLE OF CONTENTS

1 Introduction ........................................................................................................... 5-6
2 Background ........................................................................................................... 6-9
   2.1 Multi-Platform Application Development ................................................. 6-7
      2.1.1 Multi-Platform Development Environment ..................................... 7-8
   2.2 Autism Background ...................................................................................... 8-9
3 iPad Autism Application ..................................................................................... 9-10
   3.1 Proloquo2Go .............................................................................................. 10
   3.2 Grace .......................................................................................................... 10
   3.3 iCommunicate ......................................................................................... 10-11
   3.4 AutismExpress ......................................................................................... 11
4 Android Autism Application ........................................................................... 11-12
   4.1 AutismSpeech DiegoSays ........................................................................... 12
   4.2 DTT Colors Full ....................................................................................... 12
   4.3 Alexicom AAC ......................................................................................... 12
5 Do2Learn’s Application User Interface ................................................................. 12-18
6 Proposal ........................................................................................................... 18-44
   6.1 Overview ................................................................................................... 18-20
   6.2 Conclusions ............................................................................................... 20-26
      6.2.1 Developmental Conclusion ............................................................... 20
      6.2.2 Work-to-Date .................................................................................... 20-23
      6.2.3 Choosing a Cross-Platform ............................................................... 23-26
   6.3 Code Challenges ....................................................................................... 26-35
6.3.1 Cross-Platform ..................................................26-28
6.3.2 Native Platform ..................................................29-35
6.4 Titanium Cross-Platform – Work So Far..........................35-37
6.5 Cross-Platform Conclusion .........................................37-39
6.6 Proposal for Stetson’s Review Board .........................40-41
6.7 Research Testing Conclusions ....................................42-44

7 Future Work ...............................................................45

8 References .................................................................46-51

9 Appendix .................................................................52-62
  9.1 Description of Statistical Methods ..............................52-53
  9.2 Informed Parental Consent for Research Involving Children......54-58
  9.3 Prototype Evaluation ..............................................59-60
  9.4 Script for Game Play and Interview ............................61-62
1 Introduction

Tablets have been around since at least 1985, when a company named Pencept joined with another, CIC, to create the first PC pen computer products that implemented handwriting recognition technology [19]. Apple then redefined the tablet industry with its release of the iPad in April of 2010 [20]. The Xoom, the first Android tablet released, hit the market on February 2011 [21]. Apple relayed that there are currently 700,000 applications available with 250,000 of them specifically designed for the iPad. In 2010, Android’s own application market place accumulated 1 billion user downloads of their applications, to 2012 where they are now up to about 25 billion application downloads. Similarly, Apple released a statement about 7 months prior when they had achieved their 25 billion applications downloads benchmark [22]. Autistic children show drastic progression when utilizing the educational applications offered on both the iPad and Android application markets.

Nearly one in every 110 American children is diagnosed with autism. This non-curable, complex neurobiological disorder comes with several lifelong symptoms [8]; however, with early intervention these symptoms can be either minimized or treated to the point of making them significantly more manageable later in life. Some of these symptoms include, but are not limited to, difficulties in social interaction, verbal and non-verbal communication problems, sensitivity of the five senses, and behavioral issues that include unresponsiveness. This list of symptoms is a vast collection of obstacles faced by those suffering from autism [2], but the use of technology can help. The utilization of tablets during the stages of early intervention is a new and supported method of help. The iPad, created by Apple, and the Android tablet, created by Google, are giving those with autism a voice. The tablets host a wide range of applications used to teach and aid those suffering from this neurobiological disorder [4].
We have redeveloped the Do2Learn application “What’s the Order” [6]. Do2Learn’s application is an autism education game that teaches linear and relative time. It has extensive options on its start menu for setting up different scenarios for individual learning experiences. This application employs videos, animations, voiceovers, a reward system, and game play assistance. We have redeveloped “What’s the Order” and ported it on to the Android and iPad tablets utilizing native application environments and a cross-platform framework, and we compare the results.

2 Background

2.1 Multi-Platform Application Development

With Today’s elevated standards of technology, companies are forced to utilize iOS, Android, Mobile and Web devices or else suffer a loss of competitiveness within the marketplace. It is plausible for a company to choose just one device and disregard development for the others but this path leads to the loss of accessibility among their clientele. A company could also choose to develop for multiple devices but this would require programming in different languages using different compilers for each of the devices. This method is costly and time consuming. A third avenue is for a company to develop their application on a multi-platform environment with a single code base. This would allow for one programming language to bring the application to all relevant devices without having to individually code them. For this research, we have used one code base to implement the application on both the iOS and Android tablets. Benefits like lower development and maintenance costs, increased productivity, quicker transition to application market, and reduced training or learning for the one development platform are all side effects realized by cross-platform development environments. While using a single code base is convenient and allows for rapid prototyping, can be web oriented, and offers shorter development time, negatives for
this method also exist. Complications like the increasing complexity, costs increases for added development features, bugs in software, flexibility limitations, and a large learning curve for implementing the new framework [18]. This method is still in its infancy stage struggling with several limitations, like increased complexity, costs, software bugs, flexibility limitations and learning curve, not seen by the other solutions and should not be seen as an error proof solution to multi-platform programming [14].

2.1.1 Multi-Platform Development Environment

Appcelerator’s Titanium Platform SDK allows programmers to code applications using a single JavaScript code base for developing iOS, Android and mobile web apps [15]. It offers support for iOS, Android and HTML5. Appcelerator also offers the Titanium Studio, which is an Eclipse-based IDE, integrated development environment, that can be used to build, test, package, and publish, desktop and web applications. This solution comes with four versions, the basic model available for free download and the premium models available for an unlisted fee. Even with the added benefits of the premium model, the basic model is fully functioning and has everything a developer needs just short of offering the support service found in the premium version.

Appcelerator, while giving a free solution in the form of developer tools and support services comes with unique “libraries, syntax, and development approaches” applicable only to its own environment [16]. This unique framework requires a lot of time to become proficient in utilizing its functionality, time that would be commanded by the “trial and error” life cycle.

Motorola’s RhoMobile Rhodes, a popular multi-platform development environment, offers an open source, single code base of a Ruby on Rails framework with a model-view-controller setup [9]. Their slogan of, “one codebase, every
smartphone,” offers insight into how their single codebase can encompass all of the most popular devices on today’s market without the costs and time restraints put on single platform development. RhoMobile, unlike Appcelerator, uses a programming language that allows programmers that already are familiar with Ruby to jump the learning curve of developing on this platform. This framework, joined with the use of the Git distributed version control system, DVCS, gives developers the tools needed in an effective and efficient manner and the support needed to secure the application’s code. This bundle is made more user friendly with its new installation executable file on Windows [16].

A different way of achieving cross-platform development that circumvents the limitations posed by a single code based framework, is to develop a web application using JavaScript, CSS, or HTML. This strategy altogether bypasses the native application development process capitalizing on the universality of web applications.

### 2.2 Autism Background

Autism is a spectrum of strengths, weaknesses, and intelligence levels. While these tablet applications aren’t a cure to Autism, it doesn’t disqualify the device as being a revolutionary medium for education in the classroom. Laura Holmquist quoted to Fox News that through the integrating of the iPad device into her son’s, Hudson Holmquist, life, they were able to get his violent hourly meltdowns under control and help increase the effectiveness of communication between them [5]. A Harvard Medical assistant professor and Autism expert, Dr. Martha Herbert, fully backs the implementation of the iPad for Autism. Dr. Herbert relates how the iPad gives control to Autistic individuals who can’t control or sift through the vast amounts of information being hurled at them daily [5].
Those suffering from Autism are constantly being alienated because of their differences. Having to use bulky communication devices just does that much more to draw unwanted attention to their daily struggles. By taking the tablets out of the classrooms and onto the streets, those with Autism can use the many functionalities for daily life, helping them to become more independent in everyday life. In addition, to meeting the social needs of Autistic individuals these tablets, while expensive for recreational use, is cheap in comparison to the bulky communication devices normally prescribed to help those with Autism [12].

3 iPad Autism Applications

There is a notable disparity between the amount of autism applications offered on the iPad verses the Android tablet. The iPad is overwhelmingly the tablet of choice among those looking to employ the benefits of tablet applications as a teaching medium. The specific iPad applications that are designed to help autistic symptoms are focused mostly on sensory stimuli, language and communication improvement, and the reduction of social deficiencies, functional shortages and stress levels. These categories cover a wide range of the symptoms that make autism a daily struggle.

Applications, like the iAssist Communicator for the iPad, allow Autistic individuals to touch corresponding pictures and words to help them communicate what they are thinking [3]. An app called iMean is essentially a large keyboard that displays texts and utilizes word prediction for Autistic people to communicate without being confined by a picture based word bank [3]. Autistic people tend to have trouble making eye contact, so the app Look in My Eyes: Steam Train uses the animation of a train to help teach eye contact [3]. For children in school dealing with autism the app School Skills has a database of expected and unexpected social skills pertaining to interactions one might encounter in different classrooms, the lunchroom, playing with friends as well as
interpreting feelings like anger, responsibility and disappointment. Some people suffering from autism have difficulty reading social cues like smiling, so the app Smile at Me uses a reward system to promote the learning of smiling cues [3]. The list of applications tailored to the needs of those with Autism is expanding exponentially. Through this avenue of technology those suffering from Autism now have an outlet that allows them to express themselves in areas they couldn’t before.

### 3.1 Proloquo2Go

Out of the top-rated applications for autism on the iPad, one called Proloquo2Go is rated first [1]. Proloquo2Go is a communication application with a wide spectrum of features tailored to provide solutions to the many difficulties these children face when communicating. This application has a text-to-speech option along with a library of up-to-date symbols that represent vocabulary. Proloquo2Go, though revolutionary and undoubtedly a helpful resource for those with autism, is priced at $189.99, making it a costly option for many people struggling to cover all of the other costs associated with autism.

### 3.2 Grace

The second top-rated application, Grace, is a more affordable communication solution [1]. Grace allows the user to build sentences by selecting images from its database that depict the subjects they wish to communicate. The unique feature of this application lies in its drive to build a relationship between the talker or typist and the listener or reader. The child with autism will communicate what they want through pictures and words that the recipient will understand, creating an open social interaction along with aided communication. This application is on the Apple market for $37.99.
3.3 iCommunicate

The iCommunicate application, ranked third among iPad autism applications, is utilized as a teaching aid more than just a communication aid [1]. iCommunicate allows users to create custom flash cards with their own images. These custom-designed flash cards can then be selected in different combinations to result in the desired sentence; it also allows the user to create custom audio input for communication and teaching in any language. Coming with a library of a 100-plus options to start with and user friendly operation, the iCommunicate application goes a step further in helping teach children with autism and is only $29.99 in the market.

3.4 AutismExpress

AutismExpress is an app designed to help struggling children learn how to express themselves emotionally [1]. It comes with a preloaded library of the most used and common emotions, which are depicted by smiley faces, frowning faces and other cartoon-like facial expressions. These depictions are used to help the children overcome their low functioning social and communication skills, allowing them to better understand and interact with those around them and to better understand and handle the emotions they experience. This application is not only effective but also free, allowing access to more people.

4 Android Autism Applications

Recently, the Android market has grown to be a more comparable size with the iPad’s autism application market. Top rated applications, like the AAC Speech Communicator application was developed for all ages of autism that are struggling with communication [7]. The application helps teach communication methods and skills,
focusing closely on those with reading difficulties. As a huge contrast to the top rated iPad app, the AAC Speech Communicator is free.

4.1 AutismSpeech DiegoSays

AutismSpeech DiegoSays, is the second highest ranked app in the list of top autism Android applications [7]. AutismSpeech is another communication assistor. This app utilizes pictures and voice functionalities to help give those with autism a voice.

4.2 DTT Colors Full

Autism/DTT Colors Full, the third most prominent amongst the android apps, was developed by Dr. Gary Brown [7]. Dr. Brown’s app uses the method of Discrete Trial Training, DTT, to teach the fundamentals of colors to autistic children. DTT works by breaking down a whole skill into its parts. The discrete trials break down a skill, teaching step-by-step the understanding and emulation of that skill, and then proceeds to slowly and methodically build the skill back up into its whole state [17].

4.3 Alexicom AAC

Alexicom AAC for Android, is an app whose functionality allows the user to input their own images [7]. In addition, this app offers word prediction, text-to-speech, 1,200+ pre-packaged pages, over 7,000 images, 20 natural voices in over 7 languages, and a cloud back up service for between device sharing for a $40 per month fee.

5 Do2Learn’s Application User Interface

Do2Learn’s “What’s the Order?” has a start screen interface that allows the instructor to select options for customizing each individual game for their autistic students or children. The application focuses on linear and relative time and, within those, several subcategory options including basic and advanced play. In linear time,
once the player’s name has been entered, and the instructions, accessed by pressing the corresponding button located in the top right quadrant, has been read, the instructor can select as many actions as they want. Then they need to choose if they want the game to continue on repeat, if they want practice rounds before each game, how many rounds they want per game, if they want a reward screen to come up after the player is successful in the game, if they want instructional, conversational text to appear on the top of the screen during the game, and if they want a video of the actions selected to be played before each game round; see Figure 1. The Instructions button links to a fully detailed instructional page to help users navigate through the application; see Figure 2.
The relative time option, as well as the linear time option, offers basic and advanced playing modes, both with continued options available with the Advanced Options button; see Figure 3. The Advanced Options button takes you to a separate screen where you are given control over how you want the game dynamics set up; see Figure 4.
Once all options in the game’s start menu are selected and the instructor presses the Play Game button, the player is taken to the game screen. If the video option was selected, the player is immediately shown a video of someone completing one of the previously selected actions; see Figure 5.

Once the video runs, the player is directed back to the game screen where they are looking at an array of boxes and images for the action selected in the start menu and seen in the video. In Figure 6, the action ‘put on sock’ was selected under linear time. Here, the player is shown two images and asked to place them in order of what image happened first and then what image happened second. In Figure 7, the same ‘put on sock’ action was selected but under relative time. In a game with relative time selected, the player is given the middle image and asked to select what image comes earlier and which one comes later than the middle image.
If the player incorrectly guesses the images order, a voice clip plays telling them that is incorrect and to try again. The player is also given a help option, located on the
top right corner. Once selected, a white-gloved hand appears and points to the correct answer of the part the student is currently trying to solve; see Figure 8. It then disappears and allows the student to continue trying on his or her own. When they guess correctly they receive a point, which manifests as a bubble in the bottom panel of the screen. If the rewards screen was selected in the start menu, upon correctly finishing the game rounds, a short animation of a wizard with a ‘good job’ sign pops up; see Figure 9.
Once the player completes all rounds the instructor can select the Back button, located on the top left of the screen, and navigate to the main start menu again. From here, the instructor can select the View Report button, located in the top left corner of the main start menu, and see an overview of data collected from the student playing the game; see Figure 10.

![Performance Report](image)

6 Results

6.1 Overview

We have redeveloped an application, which was originally a web app from Do2learn.com, a site dedicated to helping those with Autism. Do2Learn’s application is called “What’s the Order” [6]. Our version focuses on integrating the benefits of a tablet with the proven autism games for teaching linear and relative time, concepts not easily absorbed by those struggling with autism. Research has established that the rods and cones found in the eyes of autistic children show irregularities, most likely due to
chemical imbalances and deficiencies, in comparison to their non-autistic counterparts. This difference manifests itself into color sensitivity. 85% of those tested reported seeing colors at a “greater intensity” than those tested without autism [13]. “What’s the Order’s” user interface displays neutral colors that are shown to calm and not agitate those with autism [11]. This allows the user interface to promote visual learning with no negative visual effects. The app’s set up screen allows for high customization, which gives teachers or parents the ability to focus on the best method of teaching and best kinds of stimulation for each individual person with autism. Research shows that using animation and touch features help to engage those with autism far better than alternative methods [10]. In addition, top rated applications on both the iPad and Android markets seem to lack realistic images. Instead they teach through use of cartoon-esque imagery. “What’s the Order” utilizes real life images to help not only teach the concepts but teach the concepts in a practical, real life related context. “What’s the Order” does not have the same disconnect that other top rated applications have between their cartoon-styled pictures and the real world corresponding equivalents. In order to help a wider spectrum of autistic learners, implementing realistic libraries of images within the applications teaching games would help these students better connect with the world around them.

This research takes the idea of “What’s the Order” and ports it onto both the Android and iPad tablets. The purpose of this application is to teach young children with moderate to severe autism how to tell time in both a relative and linear capacity. The ported, tablet application’s user interface will be loosely modeled after its web application predecessor but adds color neutrality, touch functionality, and layout size adjustments. The color difference is adopted due to research indicating that neutral, non-bright colors help autistic students focus better than bright reds and yellows that appear almost florescent to them [13]. By porting the concept over from a web
application to a tablet application, adding touch functionality becomes a necessity. The tablet's user interfaces run off of touch function whereas web application's run off of mouse gestures. Additionally, when developing for a web application, layout orientation and dimension constraints are more flexible than developing for a tablet interface that is constricted by the smaller screen resolution. This research produces functional prototypes for the Android and iPad tablets using two different development methodologies; individual prototype development for the separate Android and iPad environments, Eclipse and Xcode respectively, and a prototype utilizing the cross-platform framework of Motorola’s RhoMobile Rhodes.

6.2 Conclusions

6.2.1 Developmental Conclusion

After developing prototypes under Eclipse, the native Android environment, and Xcode, the native iPad environment, we found using the cross-platform development method to be the best solution. By developing a prototype through RhoMobile we were able to cut development time, and maintenance time, in half. RhoMobile is an Eclipse installation, which cuts down the learning curve of having to get familiar with a new program interface. RhoMobile uses one code base of Ruby on Rails to deploy an application to the iPad, Android and many more devices, which is unlike its alternative of programming for multi-device-compatible application in their native environments of the iPad tablet, which uses Objective-C, and the Android tablet, which uses Java.

6.2.2 Work-to-Date

Our work so far consists of three functional prototypes; one developed through the native Android environment only compatible with the Android tablet, see figure 11, one developed through the native iPad environment only compatible with the iPad
tablet, see figure 12, and one developed through RhoMobile with multi-device compatibility.

![Figure 11](image1.png)

![Figure 12](image2.png)

All three user interfaces feature the same options and information from Do2Learn’s original web application. The prototypes error check to make sure the
necessary information on the main menu screen has been input or selected before the user can start the game. In game mode, the prototypes feature placeholder images to be later filled in with Do2Learn's image and video library; see figure 13. While the user plays the game a logic method checks to make sure that the move is correct and that the image repositions itself to the image placeholder location the user just touched, see figure 14. After the selected rounds have been played and completed, the score integer is incremented and sent to the report screen along with username and rounds played, see figure 15.
6.2.3 Choosing a Cross-Platform

Of the main cross-platform solutions available in today’s market for developers, my research focused on the three that consistently were ranked in the top 4. Those main three were PhoneGap, RhoMobile, and Titanium Appcelerator; comparing the level of developmental difficulty associated with each environment narrowed these three solutions to one. The platform PhoneGap presented several difficulties in its command line styled project compilation and in its lack of learning resources.
RhoMobile, like PhoneGap, also seemed to lack sufficient learning resources in addition to utilizing a Ruby-based framework that poses a large learning curve to those not fluent in the language.
Titanium Appcelerator had the least amount of difficulty, in terms of development for this application.

(Titanium Appcelerator’s User Interface)

Titanium utilizes typical, and more widely known, web development languages such as Python, PHP, Ruby, HTML and JavaScript. The availability of such a multitude of languages allows for developers to bypass the learning curve of RhoMobile and program in whichever language they are most proficient in. Titanium’s graphical user interface is very clean and simplistic and allows for developers to quickly grasp the architecture of the environment. Titanium also surpasses PhoneGap and RhoMobile in its surplus of online resources, direct from the company itself.
These resources include online tutorials, both in plain text and through video, discussion forums with responses from Titanium’s developers, and source code developed to demonstrate the different functionalities available through Titanium. In this paper, Titanium Appcelerator is the cross-platform solution that is being compared to the native platform environments of Xcode and Eclipse [25].

6.3 Code Challenges

6.3.1 Cross-Platform

The selection of which cross-platform environment to use in the comparison to native platform development methodology was done through a comparative analysis of time logs, learning curve, and usability. There was no preexisting experience with any of the three platforms so for each cross-platform several beginning tutorials were followed through to completion. These basic applications helped determine how long the learning
curve of each specific platform would most reasonably be if further development happened, how long it took from setup all the way through to tutorial completion, and how difficult were the problems encountered during application development for each platform.

First, RhoMobile, by Motorola, was examined due to its top placement on review sites. RhoMobile setup and configurations took a total of about an hour.

There were very few tutorials found to help make learning RhoMobile more easily accessible. Even motorola.com has very few training tools for the platform much further than the basics. This made the learning curve more a product of trial and error, which was found to be very time consuming. The Internet does provide resources, but they were hard to make applicable to what the Autism application would require for development. For example, the sample code and tutorials found posed a challenge when trying to duplicate their functionality in the form of video functionality, button behavior and the UI look and feel. Ruby on Rails, the programming language used to develop in RhoMobile, was more challenging to pick up in comparison to its web counterpart languages like HTML, CSS, and JavaScript.

Following RhoMobile in the comparative analysis was PhoneGap. PhoneGap, created by Nitobi and later bought out by Adobe, was chosen next for its repeat appearance on cross-platform review sites as a top three candidate for mobile application developers [26]. PhoneGap’s setup and configurations took a total of about 40 minutes. One of the incentives for using PhoneGap is its more widely known web programming languages like HTML, CSS and JavaScript. PhoneGap.com also provides users with a documentation library full of tutorials ranging from beginner to advanced.

Issues encountered with PhoneGap development started with command line set up and project creation. This was time consuming but not difficult, as PhoneGap’s documentation section guides users through the process.
Finally, Titanium Appcelerator was put through the comparative analysis. Titanium’s setup and configurations took a total of about an hour. Appcelerator.com offers users a large bank of documentation for learning and getting acquainted with the UI of Titanium, as well as getting it set up. What pushed Titanium’s user help base above RhoMobile and PhoneGap is the Developer Center that offers videos and fully functioning sample applications for learning best coding practices of the framework [27]. Though videos and sample code for the other platforms are available upon scouring YouTube.com and GitHub, Titanium has several solid code samples conveniently located on its own site in addition to other Internet hosted supplements [33].

In lieu of sampling each platform, Titanium Appcelerator came out on top for its minimal set up time, its easy-to-pickup web development languages, and its large accumulation of documentation.

Once Titanium was selected over RhoMobile and PhoneGap as the cross-platform environment to be compared to the native platforms, prototype development started. During development more code roadblocks were uncovered. “What’s the Order” utilizes radio buttons on its home screen during the game’s setup. iOS and Android on the Titanium platform does not recognize radio buttons as a default component. This required the creation of separate code bases to make a radio button work around for both the Android and iPad tablets. Aside from iOS and Android not supporting default radio buttons, no real detrimental issues were found pertaining to the framework itself. This issue, though, shed light on the capabilities of Titanium to support separate code bases for the iPad and Android Tablets. This is helpful because now Titanium allows mobile application developers to use common class modules in their code bases for iOS and Android all within the same language, unlike native platform development’s separate code bases.
6.3.2 Native Platform

The native platforms, Xcode and Eclipse, were selected in part for their popularity in the development community, as well as for their lack of competition within the native platform environment industry.

Xcode, on the Mac, is the most popular and widely used standard for developing native iOS applications as well as being one of the few options available for developers in the market. No prior experience with Xcode or Objective C was utilized before the start of this research. To develop on Xcode users simply have to go to Apple’s Developer section of Apple.com and download the newest version of the software [28]. Set up time required minimal effort and time. When starting development, one exceedingly large challenge posed by Apple is its requirement of a iOS Developer License [29]. The iOS Developer Program costs $99 a year for a developer.

iOS Developer Program
The fastest path from code to customer.

With this license you are given access to all of Apple’s developer resources and training materials in addition to being able to test applications on the iOS device itself. The huge disadvantage of not having this License is the restriction placed on deploying your application in the native environment for testing and the loss of access to all of Apple’s training resources. Without the license a developer would only be able to test applications on iOS simulators and have to learn Xcode development with their own resources.
In addition to the challenges and large, in comparison to the free development options offered by Android, $99 price tag of becoming an official iOS developer, Xcode’s development language, Objective C, was more challenging to pick up than its competition’s web languages. This added a lot of time to development in contrast to the cross-platform.

Another challenge presented by native iPad development was tablet orientation. When wanting to rotate a tablet application to a landscape orientation a developer had to rework an existing method in the .m class or else the view will rotate to landscape but the content within the view will remain in the default portrait orientation. Previously, developers achieved this effect by editing the Simulated Metrics attributes; now developers must do both in order to achieve a proper landscape rotation.

(Native iPad attributes editor for orientation rotation – Simulated Metrics attributes)
This coding challenge, among other smaller ones, was solved through extensive online research. The large amount of support offered by Apple, with the membership of a Developer License, helped overall to mitigate this platform’s developing challenges.
While the tutorials and training resources make it easier for a developer to create the backend of applications, Xcode’s drag and drop style UI allow for extremely easy creation of mobile application’s UI frontends with instantaneous display results. Essentially, Xcode lets the developer design a GUI for the application and, based off of their visual layout, generates the view code automatically. This is a huge advantage to developing GUIs in native environments verses cross-platform environments that do not offer any view generation for the developer.

Eclipse, version Juno, was selected for its popularity in the development community and to decrease production time with prior Eclipse development experience. In order to use the Eclipse IDE for Android development the user needs to install the Juno IDE, or whatever version of preference, from the Eclipse website [31]. Eclipse now offers a mobile developer version on their site that includes “a Java IDE, C language support, a Git client, XML Editor and Mylyn” [32]. Once a platform version is selected and installed, Eclipse has to be configured with the Android Development Tool, or ADT, plugin. Then the user must download and set the path directory in Eclipse to the latest SDK and NDK. Once configuration is completed, users need to set up the Android Emulator based on what iOS device the application requires.

Once everything was set up, development with Java, the language used by Eclipse, helped shorten development time due to previous experience with the language. Learning the UI of the Eclipse platform, with the ADT plugin, took minimal effort in light of previous development work with the platform. Like Xcode, the Eclipse platform, configured for Android development, also uses drag and drop style functionality in its UI for developers. This frontend UI allows for developers to easily create application GUIs with minimal code writing as Eclipse generates the view code based off of the graphical representation created by the developer.

The main coding obstacle presented by native Android development was creating a video player that was facilitated by a sleep thread to handle automatic view
switching. The original attempt at solving this problem revolved around implementing a video player within the MainActivity.java file to be activated upon button click. This method posed several problems in development that resulted in the need for a new solution. Instead of developing within the main class this alternative solution worked by creating a class dedicated solely to the video player and Intent, or view, switching.

```xml
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="1280dp"
    android:layout_height="800dp"
    android:background="#666666"
    tools:ignore="DeprecationWarning">

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:layout_below="@id/textView"
        android:layout_gravity="center"/>

</FrameLayout>

(VideoView for Block.mp4 – blockvideo.xml)
This allowed for an easier call of running a clip within the main and the game classes. Instead of revalidating the code for the video player multiple times through all of the classes now a developer only needs to call a single Intent activity to switch views. Once that Intent activity is called, the user’s screen will switch to the BlockViedo.java class. This view will play the duration of the video and then switch automatically, based on a timer, to the next view designated by the developer.
There are a good number of tutorials and training videos for Android development within Eclipse that proved to be very helpful in cutting down development time and create an effective prototype of the native Android version of the autism application.

6.4 Titanium Cross-Platform – Work So Far

Through development with Titanium several roadblocks became apparent. Some components of the graphical user interface, or GUI, were only functional on specific devices. For instance, when implementing radio buttons for both the iPad and Android tablets, the iPad Simulator would not display them while the Android Emulator would. Further troubleshooting by running the application on the respective physical tablets showed the same outcome. This prompted further research of how to circumvent GUI components that were only available for these specific devices. The solution was to split the code base into three sections. This solution is along the same lines as the native platform environments, but there are still two main differences between the two developmental methodologies. As previously discussed, native platform development requires two programs, Xcode and Eclipse, two code bases, two different programming
languages, Objective C and Java, and two different applications. This native platform methodology used in the cross-platform development environment as a solution overcomes the original shortcoming of most cross-platform development—the loss of the native environment feel. By doing this, the cross-platform development, in comparison to the native platform development, still has one program, Titanium, one programming language, JavaScript, and one application but now, in addition, requires maintenance of two code bases. This was a necessary sacrifice in order to obtain the native look and feel on both the Android and iPad tablets.

Of the three divisions, the first section was made to handle the code base for Android. The second section was used to maintain a code base for the iPad, which enabled development of custom radio buttons. Since the radio buttons through Titanium are not functional for the iPad, but are for the Android, and the custom radio button class was not compatible on the Android, having these separate code bases allowed for the native Titanium solution to be used for Android and the custom button class solution to be used for the iPad. The third division was used for handling devices not supported for this research. This means that if someone downloaded this application on a Windows Mobile, Blackberry or a Symbian device the application would alert the user that their operating system does not support this application. A Window’s tablet was also not supported based on its lack of popularity in comparison to Android and Apple tablets and because Titanium, at this point in time, does not support it. Titanium released a press release, January of this year, stating that they plan to remedy this shortcoming by updating their software to include support of Windows 8 for both Windows RT and for the Windows Phone platforms. This new release is expected to become available to developers by the “second half of this year” [33]. This was done for several reasons, mainly for the purpose of like comparison for this research. This research focuses on the comparison of native application development for the iPad and
Android tablets and their cross-platform development equivalency. Second, these other devices are configured only for smaller, phone-specific, dimensions and not a larger, tablet-specific, dimensions which is a necessity in the development of the “What’s the Order” application. Third, the students, who are the target audience of the autism application being developed to compare native and cross-platform development environments, would have major difficulty in using this application on a small phone screen with all of the touch focused navigation and gameplay.

6.5 Cross-Platform Conclusion

This paper gives a detailed comparative analysis between mobile application development in native platform and cross-platform development environments through the development of an autism application-teaching tool. Xcode and Eclipse were the native environments for iPad and Android tablets, respectively. Objective C was utilized in the native application development of the iPad within the Xcode environment and Java was utilized in the native application development of the Android Tablet within the Eclipse environment. The cross-platform environment was narrowed down to the top three rated platforms, RhoMobile, PhoneGap and Titanium. Based off of a comparative analysis for each of the top three cross-platform environments- ease of use, efficiency, and learning curve, Titanium came out ahead in each category. Once one platform was selected amongst the top three, there was a breakdown of pros and cons to development in the native environments, Xcode and Eclipse, and the top cross-platform environment, Titanium. This research compared the time logs for development in both platforms. Native application development, with two separate code bases in two separate languages, took approximately 140+ hours, verses cross-platform application development, with two code bases in the same language, was approximately 70+ hours. This time gap was realized by the learning curve and completely separate code bases
maintained in the native environment development cycle. Production of the iPad section with in the native environment started 3 months earlier than its Android counterpart or Cross-Platform competition to try to circumvent the process of learning a new language, Objective C, and a new development environment, Xcode. This development took well over half of the native environment development time. The native Android time log was relatively short due to familiarity with the language, Java, and the development environment, Eclipse. Without experience in either languages or either development environments, it is practical to assume the learning curve would at least add double the amount of development time. This challenge is minimized in the cross-platform environment. Titanium allows for standard web development languages that are more widely known by programmers such as HTML, CSS, JavaScript, PHP, Python and Ruby on Rails. Not only is the production time in the cross-platform environment significantly faster, but Titanium also allows for customization of device-specific user interfaces, UIs. These separately coded UIs for the iPad and Android tablets allow for a previously noted shortcoming of the cross-platform development to be crossed off the list and added to its going account of advantages over native environment development. Common criticisms of cross-platform development over native development involve issues surrounding the differences in behavior of the cross-platform’s application programming interface, or API, its stability and its memory management. During this research there were no noticeable negative API behaviors that could not be worked around, no noticeable stability issues, and no significant memory management compromises. “What’s The Order” is a relatively small-scale application game in comparison to large-scale corporate directed applications with a high need for of functionality and scalability. For this research, the cross-platform development methodology far out-paced the native platform development methodology in the comparative analysis. In terms of large-scale application development with high levels of device-specific functionality, the better solution for development may lie in following
the native platform development methodology. In the case of a developer who is knowledgeable in Objective C, Java, and JavaScript and their respective platforms, Xcode, Eclipse, and Titanium, their development method of choice should be Cross-Platform application development. This conclusion, all else being equal, is based on the shorter amount of development time needed for the Cross-Platform verses the Native Platform. This is true unless developing on an enterprise level among a large programming team segmented by Android and iPad developers. In this situation, Native application development might be a more stable development option albeit a more expensive and time consuming one. Cross-Platform development is still in its adolescent years; every day new solutions and better APIs are being created to help make cross-platform development the pinnacle of mobile application development.
6.6 Proposal for Stetson’s Review Board

This research’s comparative analysis was part developmental and part environmental. Developmental analysis was done through the comparison of time logs, ease of use and application functionality. Environmental analysis was done after the four prototypes were developed. The second part of this research involved testing the autism application, “What’s The Order,” in several schools with the help of numerous students with autism. In order to run an environmental analysis test on humans, we submitted a full proposal that was reviewed and approved by the Institutional Review Board at Stetson University, which is a committee whose task it is to make sure that research participants are protected from harm. The proposal consisted of a copy of the project’s research paper, a script of what exactly would be said during testing, how testing evaluation will be conducted, a formal parental consent form that outlines the project and gives parents extensive details into the research, and a description of statistical methods that will be used. Once all mandatory documents were submitted to the Stetson review board, the next stage consisted of securing permission form the different locations to be used as testing sites, which, in the context of this research,
were two elementary schools, one high school and one autism support group. The review board was very detailed during the proposal process and required a large amount of additional details about the research, potential statistical methodology, and permission from test sites. The review board’s response to the proposal was that once explicit permission from all of the testing sites was given, then the board would officially approve testing. This process of creating all of the formal paperwork, submitting the proposal, and then resubmitting the proposal with edits of the board’s added requirements, took about 3 to 4 weeks’ time. Formal contact with each of the testing sites was reestablished. This contact involved submitting to them all of the official paperwork created during the review board’s proposal process (Included in Appendices 9.1-9.4) and giving them several copies of the formal parental consent form. The teachers were asked to provide the consent form to all parents to be filled out. The principal of J.W. Mitchell High School, Jim Michaels, and the assistant principals of Longleaf, Debora Wichmanowski, and Starke Elementary Schools, Shannon Hay, were also contacted to receive explicit testing permission. Longleaf required a background check in order to allow testing on their campus. The school directed us to a background check company that would satisfy their requirements. They only accepted money orders, purchased a Publix, for $85. They required a social security card, driver’s license, finger printing, and additional paperwork to be filled out. Approval was said to take 24/48 hours but took around 72 hours, not including the weekend. J.W. Mitchell High School and Longleaf Elementary School required additional approval from the District. All of the previous paperwork was re-submitted to the Pasco Country School District, specifically Dr. Peggy Jones, and we held communication with her and her office over email and phone. Their response to the initial proposal submitted was that they needed more paperwork to be filled out. Once their additional forms were filled out, which held a lot of redundant information that could be found on the previous paperwork, and resubmitted, it took a little over a week to receive confirmation. For
Starke Elementary School, the assistant principal, Shannon Hay, gave permission but required additional approval from their District school board as well. We were directed to contact a Dr. Colwell to start the district’s ‘request to conduct research’ process. I received feedback from Dr. Colwell and was redirected to contact a B. Lockman but when paperwork was submitted again no response was received and it was increasingly difficult to get in contact. Once approvals from the District and principals/assistant principals were received the teachers, Jack Gilbert and Danielle Grier from J.W. Mitchell High School and Janice Whittaker from Longleaf, an attempt was made to schedule time slots for testing with the students. Testing time slots were also stalled due to FCAT testing. Unfortunately, time ran out to see this process to completion but a case study was conducted with a child from an autism support group.

6.7 Research Testing Conclusions

A test case was conducted involving a 9 year-old high functioning boy with autism. The prototype testing took place in his own home and was facilitated by his parent and the researcher. The testing was delayed due to the student’s aggravated behavioral state. He was unfocused and disinterested at the beginning of testing. After coaxing from his mother and a demonstration of how the application prototype worked, he was willing to participate in the study. With the help of his parents, the game was explained and what he needed to do in order to ‘win’. Beforehand the tablets were randomly assigned a letter that represented their order of use. The Cross-Platform Android was labeled as ‘A’, the Native iPad as ‘B’, the Cross-Platform iPad as ‘C’, and finally the Native Android was labeled ‘D’. A demonstration was performed of how to play the game on the Cross-Platform Android prototype and he seemed to pick it up quickly. We proceeded to allow him to play one round of a linear progression game of building blocks for each of the four tablets.
He struggled on tablets ‘A’ and ‘B’, the Cross-Platform Android and the Native iPad respectively, but began scoring higher and completing the game at a quicker pace with tablets ‘C’ and ‘D’, Cross-Platform iPad and Native Android. This increase in ease can be attributed to the learning-effect. I noticed that while the student enjoyed touching, interacting and watching the reward animation on the tablet he was hard to keep engaged from tablet to tablet and spent little time paying attention to the beginning instructional video.
Once the student had completed the 1 round linear time game for each tablet, he was asked evaluation questions. When asked questions like ‘which tablet was the easiest’ and ‘which tablet was the most fun’ he was noncommittal and answered that ‘all’ of the tablets were. When asked ‘which tablet was the hardest to use’ he couldn’t answer but when asked if the smaller images made it harder he agreed. When asked which tablet he would like to continue playing on he referred physically to tablet ‘D’, the Native Android, and the tablet he played last. The tablet with the highest score was tablet ‘D’, the Native Android, and the prototype with the lowest score was tablet ‘A’, the Cross-Platform Android. Finally, additional comments were recorded pertaining to the student’s behavior and focus. The main push behind using tablets in autism education is to mitigate the primary challenges faced by teachers of students with autism. These include keeping them focused, engaged and dissolving behavioral issues. During the study the student remained in a highly agitated state and was difficult to get focused. His attention level towards the tablets, on a scale from 1 to 5, was recorded as a 1. He didn’t ask questions about the testing or study but did ask if he could go do other activities each time he finished playing a round. His behavioral issues before the start of the study were high and consisted of screaming, hitting, throwing things, and hiding. When testing started he seemed to calm down considerably but was really aggressive when touching and interacting with the tablets. Over all, it was inconclusive if he could tell a difference, or develop a preference, for any of the four prototypes.
7 Future Work

Continued development would include fulfilling school testing, running a chi-squared test on the results, and the bringing the Cross-Platform tablet application to completion.
8 References


<http://mashable.com/2012/02/16/cross-platform-app-design-pros-cons/>.


<http://www.appcelerator.com/platform/>.


<http://www.trainsignal.com/blog/ipad-history>.


9 Appendix

9.1 Description of Statistical Methods

Experimental trials will be constructed as follows:

The number of ways of obtaining an ordered subset of 4 from a set of 4 tablet prototypes.

**Permutations Calculations:**

\[ P(n, r) = \frac{n!}{(n - r)!} \]

\[ P(4, 4) = \frac{4!}{(4 - 4)!} = 24 \]

**Permutations Testing Groups:**

\[ a = \text{Native Android} \quad b = \text{Native iPad} \quad c = \text{Cross Android} \quad d = \text{Cross iPad} \]

\[ a,b,c,d \]
\[ a,b,d,c \]
\[ a,c,b,d \]
\[ a,c,d,b \]
\[ a,d,b,c \]
\[ a,d,c,b \]
\[ b,a,c,d \]
\[ b,a,d,c \]
\[ b,c,a,d \]
\[ b,c,d,a \]
\[ b,d,a,c \]
\[ b,d,c,a \]
\[ c,a,b,d \]
\[ c,a,d,b \]
\[ c,b,a,d \]
\[ c,b,d,a \]
\[ c,d,a,b \]
\[ c,d,b,a \]
\[ d,a,b,c \]
d,a,c,b
d,b,a,c
d,b,c,a
d,c,a,b
d,c,b,a

This will allow for experimental trials to be conducted in all possible orderings, to try and control for learning effects.

Once the trials are complete, a Chi-square goodness-of-fit test will be used with the null hypothesis that the child is equally likely to choose one of the 4 tablets.
Informed Parental Consent for Research Involving Children

[Informed Consent Form for Parents of Adolescent Girl and Boy Students Participating in the Research Titled, “Comparing Native and Cross-Platform Development Tablet Environments Based on an Application for Autism”]

[Laura Seletos]
[Stetson University]
[Dr. Michael Branton]
[“Comparing Native and Cross-Platform Development Tablet Environments Based on an Application for Autism”]

This Informed Consent Form has two parts:
- Information Sheet (to share information about the study with you)
- Certificate of Consent (for signatures if you agree that your child may participate)
- Attached on back is the Prototype Evaluation sheet with all of the questions or information that will be recorded during post-game play interview

You will be given a copy of the full Informed Consent Form

Part I: Information Sheet
Introduction

Hello, I am Laura Seletos and I am a Computer Information Systems Major at Stetson University. I am inviting you to have your child participate in research of the comparison between native and cross-platform tablet development through the development of an autism learning application. Essentially I am testing and comparing the development methods used to produce a tablet application. This comparison is done through the development of Do2Learn.com’s web application, “What’s the Order?” through the two different development methods.
My research would essentially entail working with your child and asking questions pertaining to tablet accessibility. They will be playing a pre-tested and approved autism prototype game designed to teach linear and relative time. *(Linear teaches by showing the student a video of an event and then asking them to place three images into their correct, linear order. Relative teaches by showing the student a video of an event, after the video, the middle image in the sequence of events is given to them in the middle answer slot and they are asked to place the image of the event that happens before the middle event into the first slot and the image that depicts the last event sequence into the last slot.)* You are more then welcomed to contact me for more information and you can take time to reflect on whether or not you want your child to participate in this non-invasive tablet comparison study.

I assure you that if you do not understand parts of my study, I will be more then happy to take the time to talk with you and explain. Whenever researchers study children, we talk to the parents and ask them for their permission. After you have heard more about the study, and if you agree, then the next thing I will do is ask your daughter/son for their agreement as well. Both of you have to agree independently before I can begin. You are also more than welcome to be present during game play observation. You do not have to decide today whether or not you agree to have your child participate in this research.

**Purpose**

My study is designed to help me determine which one of the four autism game prototypes is the most effective and efficient. With the help of your child, I will be able to get an inside perspective to help me understand which of the prototypes is designed to your child’s preference. Your child will be playing the same time application game on four devices, two iPad and two Android tablets, and, once finished, will be asked a series of questions. The questions are used to determine and evaluate the ease of use of the different prototypes, the use of effective interaction, and the preference for which one was most appealing to your child. Once I have collected this data, I will continue the development, to completion, on the winning prototype’s application environment. This winning prototype is determined on which prototype received the highest accumulative scores from all comparison testing.

**Type of Research Intervention**

The intervention will include a verbal questionnaire focused on questions about game play and the different prototypes. *(Attached on back for your review)*

**Selection of Participants**

I want to work with children diagnosed with autism because the application I am developing has been designed specifically to help those struggling with reading and understanding the concepts of relative and linear time.
1. **Example of question to elucidate understanding:** Which prototype was easier? Which was hardest to use/understand? Which one would you like to continue playing with?

   (Full document attached on back for your review)

**Voluntary Participation**
You do not have to agree that your daughter/son can participate in my study. You can choose to say no. I know that the decision can be difficult when it involves your children. You can ask as many questions as you like and I will take as much time as needed to answer them. You don't have to decide today. You can think about it and tell me what you decide later.

**Procedure**

1) **Your Daughter/Son will be taken aside for a one on one study:**
Your daughter/son will be given four tablets, two iPads and two Androids, with a single, different prototype of the same game loaded on each one. Your daughter/son will then be asked to play each of these prototypes that will be given to them in a randomly selected order (to account for the learning error). Once your child has completed playing one round of the game on each tablet they will be given a verbal questionnaire.

2) **Verbal Questionnaire:**
Your daughter/son will then participate in a verbal questionnaire administered by me, Laura Seletos. I will record their answers and their game scores from each prototype and then help them to rejoin the rest of their class.

**Duration**
We are asking your child to participate in a verbal questionnaire, which will take about 5 to 10 minutes of her/his time. This will be done during school hours with teacher consent. Altogether, with the game play, we are asking for about 30 to 45 minutes of your child's time.

**Benefits**
There will be no immediate and direct benefit to your child or to you, but your child's participation is likely to help me better develop an application that is designed to help children overcome some of the challenges associated with autism, in addition to making autism applications more readily available to families on both the iPad and Android tablets.

**Confidentiality:**
All of your child’s answers, scores and feedback will not be recorded and all personal information will be kept completely anonymous. The only information that will be
recorded is the raw data from their game play and their opinions on the different prototypes.

**Right to refuse or withdraw**
You may choose not to have your child participate in this study. Your child does not have to take part in this research if she/he does not wish to do so. Your child may stop participating in the game play or verbal questionnaire at any time that you or they wish to.

**Who to Contact**
If you have any questions you may ask them now or later, even after the study has started. If you wish to ask questions later, you may contact me at the following: Laura Seletos, Lseletos@stetson.edu, cell 727.271.3205

This proposal has been reviewed and approved by the Institutional Review Board at Stetson University, which is a committee whose task it is to make sure that research participants are protected from harm. If you wish to find out more about the IRB, contact Dr. Bob Boozer at IRBHP@stetson.edu

**PART II: Certificate of Consent**

**Certificate of Consent**

I have been asked to give consent for my daughter/son to participate in this research study which will involve her/him completing game play on four different tablets and one questionnaire.

I have read the foregoing information, or it has been read to me. I have had the opportunity to ask questions about it and any questions that I have asked have been answered to my satisfaction. I consent voluntarily for my child to participate as a participant in this study.

Print Name of Parent or Guardian __________________

Signature of Parent of Guardian___________________

Date _________________

If illiterate
A literate witness must sign (if possible, this person should be selected by the participant and should have no connection to the research team). Participants who are illiterate should include their thumbprint as well.

I have witnessed the accurate reading of the consent form to the parent of the potential participant, and the individual has had the opportunity to ask questions. I confirm that the individual has given consent freely.
Print name of witness_____________________ AND Thumb print of participant

Signature of witness ______________________
Date ______________________
   Day/month/year

Statement by the researcher/person taking consent
I have accurately given out the information sheet to the parent of the potential participant, and to the best of my ability made sure that the person understands that the following will be done:

1. Their child will participate in playing one round of an autism time game application on four different tablets
2. Post-game play, their child will be given a verbal questionnaire

I confirm that the parent was given an opportunity to ask questions about the study, and all the questions asked by him/her have been answered correctly and to the best of my ability. I confirm that the individual has not been coerced into giving consent, and the consent has been given freely and voluntarily.

A copy of this Informed Consent Form has been provided to the parent or guardian of the participant ____

Print Name of Researcher/person taking the consent__________________________

An Informed Assent Form will ____ OR will not ____ be completed.
9.3 Prototype Evaluation
(Given verbally to the kids testing the prototypes)

Have cutout numbers taped on each prototype so that they can be referenced during post-play evaluation.

Equipment: Two Androids and two iPads

1.) Native environment application - Android Prototype
2.) Native environment application - iPad Prototype
3.) Cross-Platform environment application - Android Prototype
4.) Cross-Platform environment application - iPad Prototype

2. Point to the tablet that was easier?

1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

3. Point to the tablet that was more fun?

1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

4. Point to the tablet that was hardest to use/understand?

1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

5. Point to the tablet that you would like to continue playing with?

1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

Additional comments from users:
Researcher:

- Record the prototype with the highest recorded game score?
  1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

- Prototype with the lowest recorded game score?
  1-(N/A)  2-(N/I)  3-(C/A)  4-(C/I)

Additional comments from researcher:

1.) How attentive is the player

(Not at all) (Neutral) (Very)

1 2 3 4 5

2.) Did the player have any questions throughout the process?

   Yes      No

   a. If yes what kind of questions/what was unclear?

3.) Did the player have any behavioral issues during game play?

   Yes      No

   a. If yes what kind of behavioral issue?
9.4 Script for Game Play and Interview:

(Place all four tablets, 2 iPads and 2 Androids, in front of them. Place corresponding cutout labeled ‘A’, ‘B’, ‘C’, and ‘D’ next to the correct tablet. This order is determined by permutation so that all 24 possibilities are accounted for. These letters are also used for each prototype so that they can be referenced during post-play evaluation.)

(Pre-set up: All prototypes’ game menus will be set up before time to the game play screen)

(Interview and game-play will take place in student’s normal classroom but away from their normal seat. They will be directed to a table with two chairs, one for them and one for me, in the back of the classroom but not out of the room.)

Hello, my name is Laura Seletos. Today I wanted to work with you on a research project for my college. I was wondering if you would like to play a time game on these tablets and give me feedback on them afterwards. The game I would like for you to play is a time game called “What’s the Order.” Would you like to play on the tablet?

(If the student gives affirmation then continue. If not, then I will thank them for their time and lead them back to their seat.)

Awesome, then let’s start. The game is played by clicking on the images in front of you in time order.

(I will step through a round of the game and talk them through how to win the game. I will get them involved and get them to touch the images with me on the first round. Once they get the hang of it reset the tablet prototype and let them play on their own).

(Once I have guided them through all four prototypes and tablets, I will ask them the following and record their responses)

I am going to ask you some questions about the different tablets you just played with.

6. Point to the tablet that was the easier.

7. Point to the tablet that was more fun.

8. Point to the tablet that was the hardest to use/understand.

9. Point to the tablet that you would like to continue playing with.

(Once all of the questions have been answered I will thank them again for help and tell them that I hope they had fun. Then I will lead them back to their seat to rejoin their class. Next I will go back to the testing table in the back of the classroom and record all
of the scores they got on each game. I will restart the games, prepare a new order for the tablets and go to the next student on the list to continue the cycle.