Mobile Apps for Google Android

esp Group 6 | Summer 2015

Lab report

Paul Caldwell Christian Mejia, Jordan Sterling, Jerrod Tanner

2015

**Abstract:**

Our app was developed by the Android Studio application. It was titled “Sushi Samurai,” a mobile app game. It consisted of a single player, endless running game. The objective of the game was to collect sushi for the Samurai’s village while avoiding sharks. The actual game was created by java class files within Android Studio. Each java class file contained the different attributes that we desired the game to have. This included the Samurai’s endless running, the Samurai and shark actions, and activities such as the main menu and retry button. All of the goals set for the game were created by writing code in the java language. Our coding of the app was aided by Linus Granath, a YouTuber who created a series of videos that involved programming an endless running app. The code we used reflects Granath’s work; however, our code was altered in several ways during the course of viewing and learning from his videos. This resulted in modifications of the code, thus it is ours. The graphics used in our app was all original and created though the programs Adobe Photoshop and Piskel. Our app also doesn’t correctly implement a Processing code because not only did we not have the time to do so and find a solution the loss of the R.Java file, but our time constraints for creating the app was also limited. Although, when our app was near completion, we were only able to output an earlier version of the app because the R.Java file from our computers was rendered unrecognizable by our computers and was then removed forcefully. The R.Java file is an auto-generated file by Android Asset Packaging Tool that contains resource IDs for all the resources of res/ directory. Without the R.Java file, it is impossible to render anything within the Android Studio application and basic commands are outputted with errors. This problem was not resolved by the deadline. However, our code is still there, along with images, and other visuals that represent the final version of the app.

**Background:**

Our app was created with the use of the Android Studio application. Android Studio is an integrated development environment (IDE) that Google currently uses as its primary IDE for Android development. Previously, Google used an application known as Eclipse for their Android development needs. With Android Studio, anyone with access to the internet gains the ability to create applications to run on the various versions of the Android operating system. Also essential to the development of our app was the Processing program which is an open source programming language and IDE. Both of these programs build off of the Java programming language. The Processing program was to be used to implement a motion sensor that took advantage of the properties of the accelerometer in the device. The graphics of this app was created with the raster graphics editor Adobe Photoshop and a fee online sprite editor known as Piskel.

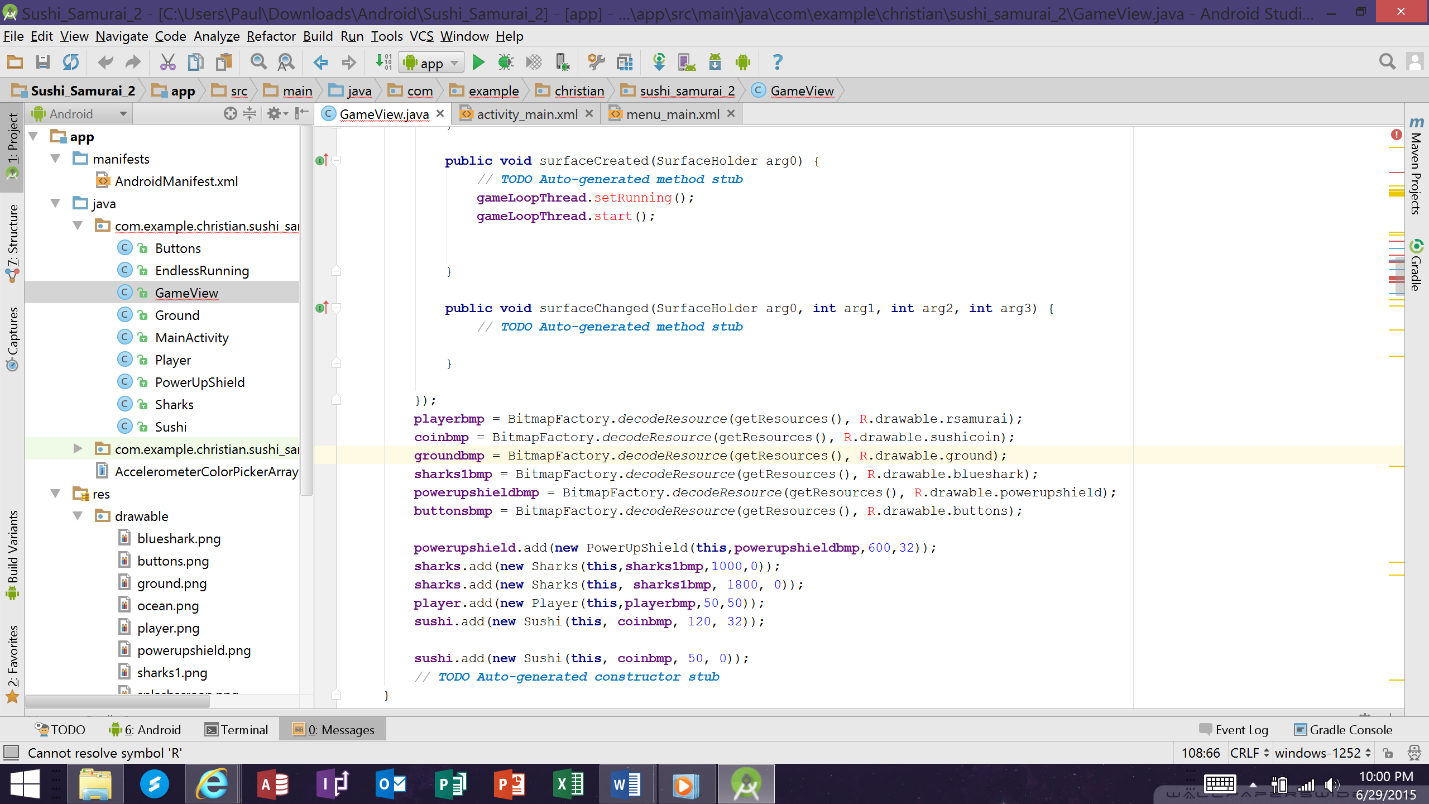
**Methods:**

We were given a week's worth of time to create our app. There were four members in our group. Two members were designated as graphic designers and were responsible for creating all of the graphic elements of our app. The other two members were responsible for creating the Android java-based code that actually manifests the app. As the code was developed, the graphic materials were inserted into the code as drawables. The members of this single group then worked accordingly and in union to develop our app.

**Results:**

As a result of coding several methods in java code, our app was almost fully functional. Our app does not incorporate any Processing due to the time constraints that limited our app development. Due to an error with the R.Java file in both of the programmers computers, our code was not able to implement and update an earlier version of our app. However, our code explains what the app’s function should have been. When the app is first called, it would open up and contain a main menu that displayed buttons. Within the actual game, our player was a Samurai, on a pier, with an ocean background. The user would have to collect sushi along the way, which the app would record the amount of sushi collected. Also, sharks were implemented in the game as the obstacle. The sharks would pop up from the ocean, and the Samurai would have to avoid it by jumping. The jumping was caused by the user tapping the screen. At any instance that the user was hit by a shark, they would be taken to a screen with their score and a retry button. Sushi Samurai is an endless running game, so the user was able to play as long as they wanted until they lost. Even though our code was unable to output the correct version of our game, the code explains the functions.

Figure 1:



\*The following graphics reveal the detrimental effects of a missing R.Java file.

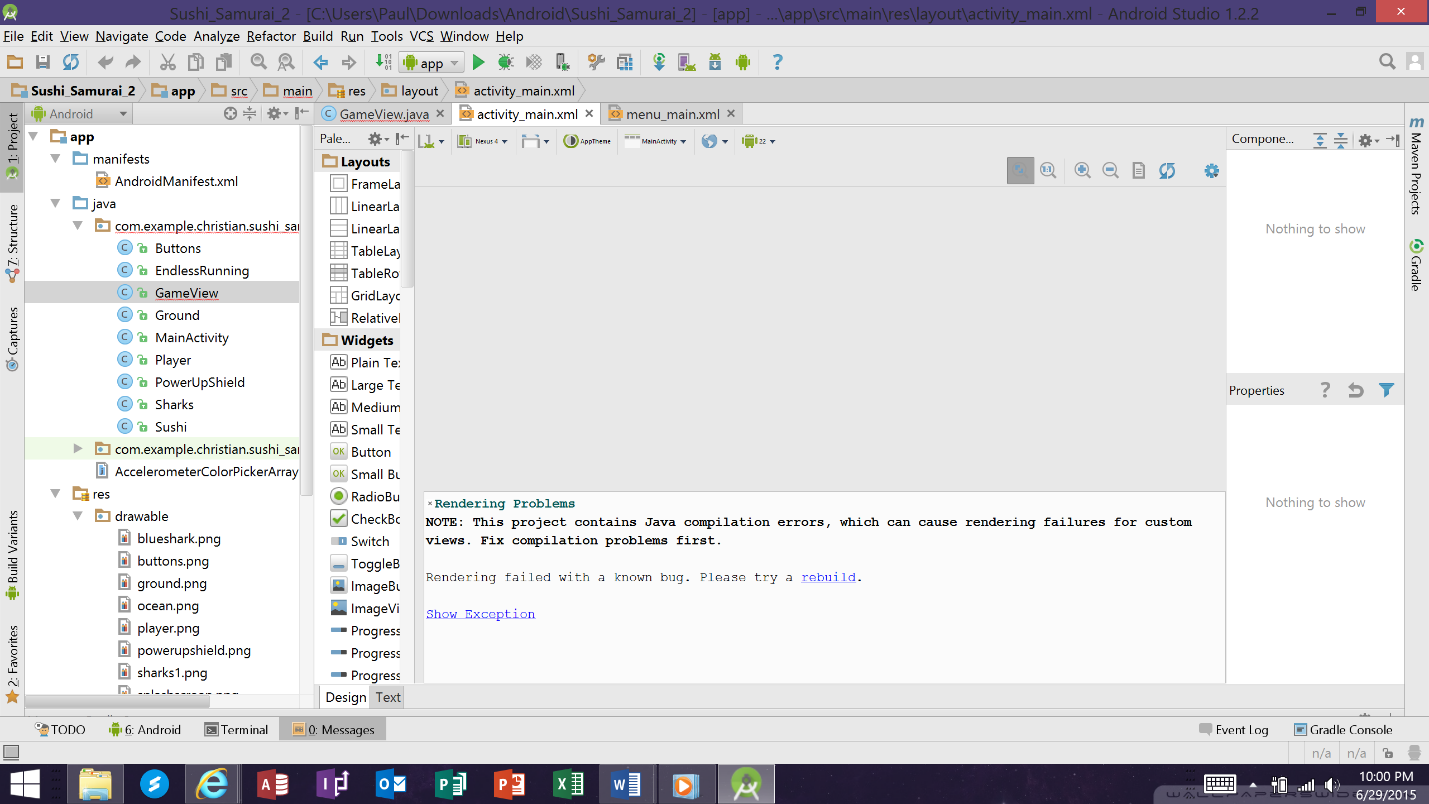


Figure 2:

**Discussion:**

Had we been given the appropriate amount of time to create our app, it most certainly would have been a subliminal experience; however, we were not benedicted with the basic, yet essential luxury of time. With no previous coding experience, we were able to make a functional, endless-running game. Two weeks would have been the appropriate amount of time to truly complete such a daunting task. Many ideas for our app had to scrapped because of our time constraint. For example, a slashing feature was going to be implemented into our game but was cut out due to our lack of time. A sensor was not integrated with our app because our instructors did not teach us how to implement into Android Studio through Processing. If we had the capabilities to incorporate Processing correctly, a color picker array would have been created to work in union with the accelerometer. Depending on the way the Android device was held, the background would have changed color. The only Processing instructions that we received were in terms of graphics and creating an image which wasn’t necessary since the tools used by the graphic students were Adobe Photoshop and Piskel. Our progress was also hampered by the disappearance of the R.Java as already mentioned but the code was still written. Though there are methods to fix the R.Java file problem, such solutions would have required reinstalling Android Studio and we didn’t have the time for such actions.

**Conclusions:**

By using the Android Studio app and working as a team, we were able to create a working mobile app. Our app was a game called “Sushi Samurai,” which was inspired solely off our imagination. The graphics teammates created the Samurai, the sushi, the ocean background, and sharks. This was combined with the coding teammates that implemented the graphics parts and the endless running. As a result, “Sushi Samurai” was accomplished. Although our actual app is not fully functional due to an error, our code speaks for itself.

**References:**

The development of this app was not possible without resources. A main guide was the YouTuber Linus Granath who exemplified how to program an endless running app. He is credited for teaching and explaining the methods and function needed to create an app. His code was not directly copied, it was typed by the java and processing members of the group who altered the code to suit the creation of “Sushi Samurai.” Two books called “Android for Programmers: An App-Driven Approach, “ by Harvey Deitel, Abbey Deitel, and Paul Deitel, and “Rapid Android Development,” by Daniel Sauter, were used to code certain attributes of the app.

The following is a link to our app at GitHub:

<https://github.com/ESP2015/ESP6>