Journal

**Name:** Abdullah Bin Asad

# September 5 – 8

* Helped setup android studio for Kieran and Brennon
* Dealt with issues regarding the placement of the SDK and the versions of windows (32bit vs. 64bit)
* Understood the basic concept of MVC (separate the presentation from the functionality)
* Thinking about doing Firebase
* Awaited further instructions

# September 11 – 15

* Decided to make an app that rolls 2 dice and flips a coin
* Learned about tab layouts and Fragments
* Researched and read about Tab layouts
* Managed to create tabs in my screen
* Those tabs were then populated with fragments (parts of activity)
* These fragments needed to be managed with a FragmentPagerAdapter
* The FragmentPagerAdapter deals with how the fragments are stored in memory
* Each fragment requires a layout and needs to be stored in a container (viewpager)
* The viewpager takes care of switching to the correct layout on correct screen
* Learned how to load images into the app and change their source later
* Learned about the uses of ID for each view in the layout files

Sources:

<https://www.youtube.com/watch?v=UqtsyhASW74> (tabs + fragments)

<https://developer.android.com/training/implementing-navigation/lateral.html> (tabs)

<https://www.youtube.com/watch?v=00LLd7qr9sA> (tabs + fragments)

# September 18 – 22

* Finished the MVC application – uploaded it to GitHub
* Found a tutorial series creating an Instagram clone + the backend is all firebase
* Haven’t had a chance to journalize much (have been following the series)
* I did a lot of work over the weekend
* The app has multiple activities, a “my profile” page and a lot of layouts and widgets
* Got started with Firebase, the user can now login or signup.
* I have also got the hang of the common xml tags such as ID, width, height etc. and setting “onClickListeners” for different widgets
* This series has a lot to offer and I will continue to follow it

Tutorial Series:

<https://www.youtube.com/watch?v=qpJRgr6HzAw&list=PLgCYzUzKIBE9XqkckEJJA0I1wVKbUAOdv>

* Uploaded everything up until Tutorial Series part 24.
* From now on, will commit after every video
* Helps with documentation
* Will get started on commenting everything in the code.
* All firebase dependencies must have the same version at the end

# September 23 – October 10

* Continuing with the app on the weekend
* Firebase likes to store numbers as longs
* The tutorial attaches a user model to the firebase database and the database accepts all the attributes of the model as parameters and their values as the parameter’s values. Not sure how it works but it’s neat.
* Able to add user info to the database
* Talked to Mr. Grondin about learning angular, js, node and mongoDB
* Maybe considering electron for building a cross platform app
* Will do research and document everything to present later
* End Goal: make a web based application (possibly a game)
* The user can sign out in the app now.
* Will work on the progress report this weekend

# October 11 – October 30

* Hoping to finish the tutorial series between now and the next report
* Learned about interfaces and dialogs in android studio
* Can update email and username in the database
* Took a break from the project and did a bit of research into creating a game
* I want my game to be able to run on the web as well as desktop (mobile is a plus) and use firebase for multiplayer functionality
* I considered compiling LibGDX projects to HTML but firebase cannot connect to the html module inside LibGDX
* Then I tried making something with pure JavaScript but it lacked a good framework for developing games
* I found Phaser which is a framework for developing games for the web using JavaScript
* I made a quick game using JavaScript and Phaser with basic HTML
* To make it run on desktop, I found a git repository that sets a basic project with electron configured and all you have to do is worry about writing the game
* A bonus to this approach is that the game uses TypeScript which deals with everything that’s wrong with JavaScript
* It allows for declaring variable types, interfaces, classes functions and defining scopes for variables.
* All the typescript code then gets trans compiled down to pure JavaScript that can be embedded into any html page
* I was able to re-create the simple game made in JavaScript and got it to work as a native desktop application using electron
* Spent several days learning the basics of phaser and typescript while figuring out the optimal setup for development
* At the same time, thought about the game mechanics for a game
* Implemented states and basic mouse input in the game
* Setup Mr. Grondin’s computer so that he can run my code as well
* Wil work on creating a setup guide today
* I spent the past 1 and a half week figuring out how to work with firebase and typescript
* I learned lots of other things along the way
* The next two days will be dedicated to documenting all of my discoveries and finishing up the final spec for the game

# October 31 – November 13

* Finished documenting all my scratches (struggles) for connecting to firebase
* Created a scratch with a tank moving on the screen
* Able to open tiled maps in phaser
* For my game, I have to create a huge map (slow process)
* Got the bullets to fire properly
* After reading through phaser’s documentation, I found that you can attach an onFire listener to the weapon and then add properties to a bullet as it is fired.
* Really neat stuff and I love it
* Will start on version 1.0 of the game where the program connects to firebase
* Figured out a way for a player to find new players and the new players to find previous players
* Really excited
* Right now, it’s a quick and dirty of doing it but it works.
* Spent the day cleaning up how I deal with passing info between classes
* Implemented the onDisconnect() function to remove all the info about the player on disconnect
* Players are now able to see each other shoot bullets
* I accomplished this by pushing the number of bullets a player has to firebase and each time that changes, the local instances of other players on the client side will fire bullets as well
* These bullets also hit the map properly
* Finally got the tanks to shoot each other and synchronize that event across all other players.
* Tested the game on the entire class, huge success!
* In the next version, I will implement all the less important parts such as choosing a name, health bars, reset on death and possibly a leaderboard

# October 14 – December 4

* Finished version 2.0
* Made lots of progress (bullets are synchronized across all players)
* All players see each other and die appropriately (when the player leaves)
* Will work on polishing the game up in the next version
* Will work on a report for Version 2.0
* Made a lot of progress with Version 3.0
* Implemented a name screen and displaying it with the player
* Implemented a health bar using yet another plugin
* Will work on creating a mobile controller for the game
* Still have to iron out the specs for Gr. 8 night
* I have wasted too much time trying to synchronize info when players are off their screen
* I am going to leave this feature for later and come back to it if I have the time
* Will move on to creating a mobile controller for the game
* Tried using PeerJS but it didn’t work out
* The technology is cool, but it is not supported on all browsers and it is hard to find examples
* Started version 4.0 and have a server running which keeps track of all the players and their positions
* The client pushes their position to the server and receives everyone else’s position
* I like this approach, but I still have some questions to ask Mr. Grondin

# December 5 – December 20

* Able to detect previous and new players through the server
* Able to send info (position, rotation, name) through the server to clients
* Tried several methods of synchronizing the bullets for each player
* In the end, the most effective method is just to tell everyone else that a player shot a bullet and take care of hit detection locally
* This makes it very easy to cheat but right now that isn’t the biggest concern for me.

# January 8 - January 20

* Moved all the collision detection from within each object and centralized it in the GameState.ts file
* Finished the status report for December
* The next bit will be just adding small features here and there and completing the game