Journal: Deep Jose Jash

Today we started to develop our initial specifications. The process of coming up with ideas was efficient as we have a group of three, ideas are coming from all directions. We decided on creating a game that implements many aspects of physics such a gravity and air resistance. First day of programming, we decided to install eclipse IDE. Many problems we faced were SDK and JRE updates, which consumed a lot of our time. Then we learned about the environment called Andriod Studio by Java **IntelliJ**. Besides the fact that this environment is in BETA, the stability and the quality of the work environment proved to work better for our needs. We started to use LibGDX which was Gradle based. The reason we chose the GDX library is due to is specified use in games programming. We started from the basics, learn to WAlK before you can RUN. Digging through countless tutorial videos and websites, we picked the ones that fit your game plan. Starting will simples plumbing like drawing shapes and such we moved up the ladder in terms of complexity and difficulty, such as touch input and animations. Then Mr. Grondin gave us orders to share our work using the GitHub. This is a very quick and efficient way to share our code within our group.

After updating all our build tools and sdk we got the basic program to work on a nexus 5 via USB. Setting up github in android studio was a pain because you have to define the remote repository but there is no option in the IDE. We had to edit the git files manually in notepad.

Being 3-4 weeks into the project, we are now beginning to see the overall look of game. We have implemented some new features such as orthographic camera and Box2D. We also set up the GIT so we can share our projects in between our team . Our challenges were mainly dealing with SDK and gradle version conflicts. We managed to pull,fetch and commit changes. We progressed on the idea of branching all of our related code is under one repository (Flightless Fury). We managed to add physics to the action listener.

This way we can building our game on button control. We are now working on angling our image with button control. We had difficulties as we were rotating the wrong texture/ sprite. This week we shift our approach to put everything together. This initial process started out with the creating of side scrolling.

Compared to what our group did last year, we discovered a new way to do things. We are now using a texture wrapper, which wraps the around the existing image to prevent spectator lines. The next approach was to configure all the physics aspects of our game. But during this process, Deep had a error when running GenyMotion external emulator. His device would not startup, but thankfully the solution was found online. The problems were piled on by one. This time we had difficulties with running the project. The ADB was not responding, this was caused by an abnormality in the run configuration. This was a common problem, with a simple solution. We simply had to go to CMD and type the command, kill ABD. More is explained on the third W.I.P.

This week was based on the architecture of program. We split the code into multiple classes, which makes the actual program easier to debug. We are also having difficulties with File I/O. Mr.Grondin showed Deep Aaron and Grimes code with used FileIO. Based on that, the focus is integrate that into our game. The next step is to add accelerometer. This is a built in functionality, which LibGDX has an API for. We now have a fuel bar which is based on an inbuilt timer. Power-ups and points are placed in the game, and negative attributes such as enemy birds are also present. Distance, points, fuel, speed are dynamically set and updated at every instance. The next focus is working on our presentation. We met up every lunch to work on your presentation, we also set up a specific time to meet up online.