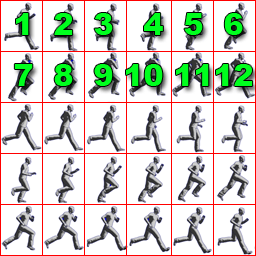
**Work in Progress Report**

**By: Deep Raithatha, Jose Rivas, Jash Pandya**

Major developments/breakthroughs(reference specific code please):

**Animations**

GitHub URL: <https://github.com/JoseRivas/LibgdxAnimation>

Tutorial Link: <https://github.com/libgdx/libgdx/wiki/2D-Animation>

**private static final int FRAME\_COLS = 6;**

**private static final int FRAME\_ROWS = 5;**

**for** **(int** i **=** 0**;** i **<** FRAME\_ROWS**;** i**++)** **{ This for loop, cycles through the columns and rows of image**   
 **for** **(int** j **=** 0**;** j **<** FRAME\_COLS**;** j**++)** **{**  
 walkFrames**[**index**++]** **=** tmp**[**i**][**j**];  
 }**  
 **}**  
 walkAnimation **=** **new** **Animation(**0.025f**,** walkFrames**);** *// #11*  
 spriteBatch **=** **new** **SpriteBatch();** *// #12*  
 stateTime **=** 0f**;** *// #13*

#11:

* The first parameter tells the animation, how much time is allocated for each frame. (in seconds)
* The more frames an animation has, the smaller the time it will be and the smoother it will look. (BUT…. It consumes more memory.)

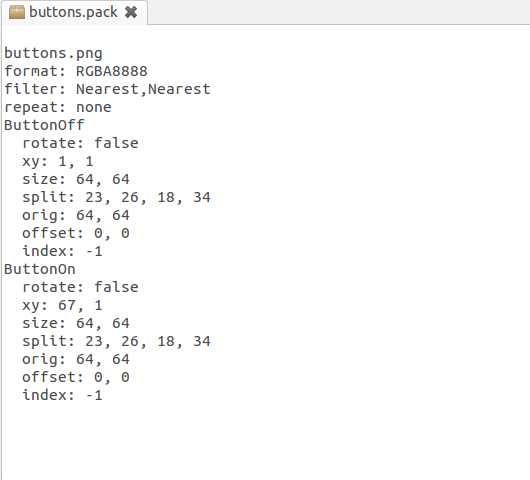
**Buttons** Creation is an aspect of our game to control the angle of flight of the penguin

Since we are using LibGDX, the way to create buttons is different from XML. We learned how to create and use buttons from the following site:

Paste URL: <http://stackoverflow.com/questions/21488311/libgdx-how-to-create-a-button>

GitHub URL: <https://github.com/JoseRivas/LibgdxButtons>

buttonAtlas = new TextureAtlas(Gdx.files.internal("buttons/buttons.pack"));

This lines references the pack of images such as button pressed and buttons released. An additional program needed to be installed called “Texture Packer”. It creates a pack file (.txt) 

Physics - Gravity

GitHub URL: <https://github.com/JoseRivas/LibGdxPhysics>

Tutorial URL: <http://www.gamefromscratch.com/post/2014/08/27/LibGDX-Tutorial-13-Physics-with-Box2D-Part-1-A-Basic-Physics-Simulations.aspx>

Adding gravity was quite difficult. We had an image moving at a constant velocity but, adding gravity took quite some time. We took about 3-4 days until we figured out how to add gravity.

Multitouch - GitHub URL: <https://github.com/JoseRivas/LibgdxMultiTouch>

for(int i = 0; i < 5; i++){  
 if(touches.get(i).touched)  
 message += "Finger:" + Integer.toString(i) + "touch at:" +  
 Float.toString(touches.get(i).touchX) +  
 "," +  
 Float.toString(touches.get(i).touchY) +  
 "\n";  
   
 }

Major Challenges/setbacks( reference specific code please):

* JRE and Updates
* Some of us didn’t have our SDK updated, we had to update them.
* SDK API’s were not synchronized (Needed 5.0 )
* Jash’s and Jose’s Android Studio suddenly got uninstalled. our

Eclipse Vs Android Studio

* We had to move everything from Eclipse to Android Studio.
* We had to move from Eclipse to Android Studio when Jash was absent so, we had to catch him up the next day.
* Jash’s and Jose’s Android Studio was suddenly uninstalled. We think that the problem was somewhere in Appdata, it’s still a mystery today of how it was deleted.

Button Creation - <https://github.com/JoseRivas/LibgdxButtons>

* Texture Packer download external
* buttons.pack
  + understanding the concept / logic behind textures
  + We have figured out how to add images in as buttons.
  + We know how to add a button, move the button around, and add action listener.
  + But we don’t have the button as an image and action listener together.
* We are still trying to figure out what the following code does.

Gdx**.**gl**.**glClearColor**(**1**,** 1**,** 1**,** 1**);**

**Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);**

Animation: GIT HUB URL: <https://github.com/JoseRivas/LibgdxAnimation>

* There weren't any problems in animation.

Any modifications to your specifications/release schedule:

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| **0.1** | **Add 3 buttons:**   1. **Up (angle change )** 2. **Down (angle change)** 3. **Launch** |
| **0.2** | **Add sidescroll motion(Background moving)** |
| **0.3** | **Add Edge Panning for up and down motion(Background moves only when the penguins reaches the edge)** |
| **1.0** | **Make the buttons change the penguin launch angle** |
| **1.1** | **Add physics Part 1: Gravity** |
| **1.15** | **Add physics Part 2: Air Resistance** |

**Description of your scratch/test program:**

Describe the generic concept you needed to test out:

The ability to add physics in our game, which required that function as we are simulating flight.

Source any web site/book that helped you with that concept:

<http://www.gamefromscratch.com/post/2014/08/27/LibGDX-Tutorial-13-Physics-with-Box2D-Part-1-A-Basic-Physics-Simulations.aspx>

Describe the code and the lesson that you learned from it:

The android environment for physics creates a virtual world based on vectors. Rendering based on time since last render, updates the render, so the image can move down the screen.

Describe any challenges that you enjoyed in integrating this scratch code into your major project:

* Useage of Box2D was needed, which was an very extension, but not overwhelming.
* We really enjoyed adding gravity to the penguin moving in constant velocity. It was quite a challenge but it’s fun to work with code that is difficult to conduct.