Work In Progress Report 2

Dino Dynamite LTD.

**Major Breakthroughs:**

Daniel: During our second phase of work, we’ve moved on from the basics of LibGdx and are now working on specific aspects of our game. From previous scratches, I struggled with the organization of my code, not splitting sprites up by classes. A major breakthrough I had was my DD\_HitDetectionSplitUp scratch where I managed to get hit detection working, but more importantly I was able to organize the code better. I split up the different sprites into their respectable classes, with their own attributes. By splitting up theses sprites I’m able to manipulate with much less effort. I made a bounding rectangle for each sprite and used the rectangle overlaps function to detect collision **if (sprDino.getSprite().getBoundingRectangle().overlaps(sprPlatform.getSprite().getBoundingRectangle())) {**. I also implemented an array list of platforms to make the implementation of David & I’s scratches easier **arsprPlatform = new Array<SprPlatform>();**.

David: My major breakthroughs this work in progress were mostly related to the scrolling background. I also realized that my platforms had stopped updating it’s random Y every update do to an updated pull so I fixed this with this fY = (((Gdx.graphics.getHeight() - nDinoX) / 3) \* ((int) (Math.random() \* 3 + 0))); in the if structure of the update function. Also I was having issues with screen zoom when using a camera so I created an aspect ratio to compress everything fixed with this float aspectratio = (float) Gdx.graphics.getHeight() / Gdx.graphics.getWidth();. Finally I had an issue with adding velocity to the camera, and continuing to redraw the background, so in order to fix this I have a velocity for the background image that is drawn as a camera.

**Major Challenges:**

Daniel: Hit detection was something I struggled with for quite sometime. I struggled with rearranging my code into classes. All my code was very much a mess and poorly organized in the beginning. I didn’t plan ahead, and set myself up for a for a strenuous cleanup. Once I had the code nice and organized, getting hit detection to work felt like a piece of cake. I referenced your drop repository with the rain drops and the bucket, but I found it difficult to comprehend at first. It took me longer than expected, but I was able to implement an array list of platforms into the hit detection scratch. An issue I had was going through the isHitPlatform function where the function was working against itself, eliminating the hit detection from the partner platform. The biggest challenge I’ve had is getting the **DD\_HitDetectionPlatformLanding** scratch to work. I’m still continuing to work on this scratch, but the goal of this scratch is to have the dinosaur land on the platform. I detect the previous position of the dinosaur just before the dinosaur collides with a platform. The dinosaur keeps getting set back to the ground position and will not jump. I worked on it from 11:00pm to 1:00am last night and a github push issue resulted in the elimination of my hard work. I managed to do some effective debugging and I’m on track to getting this scratch to work very soon.

David: Most of my challenges this WIP were related to the scrolling background, this first was caused by setting the orthographic camera. This line camBack = new OrthographicCamera(fScreenWidth \* aspectratio, fScreenHei); was giving me problems relating to the size of the screen, it would zoom in so much that all sprites would not fit completely on screen anymore. Then I had issues with the scrolling not working as it would scroll to slowly and you could run off the screen this was fixed by locking you within bounds, if(fX + (sprImg.getWidth()\*2.4) >= Gdx.graphics.getWidth()){ fX = fXOrL;. Then I had an issue with the progress bar that would decline beyond the bar, and scroll out of the bar which was fixed with nested structures and a function call, if (fProgBar >= ((fScreenWidth / 3) \* 2)){.

**Modified Plans:**

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| **1.0 Movement** | **Sprite on screen with keyboard controlled sprite movement.** |
| **1.1 Gravity** | **User controlled sprite affected by gravity.** |
| **1.2 Platforms** | **Platforms will generate and spawn on right of screen. They will move from the right to left of screen.** |
| **1.3 Hit Detection** | **Have hit detection between a platform and the sprite.** |
| **1.4 Background** | **Have a scrolling background that moves from right to left.** |
| **1.5 Enemies** | **Have enemies randomly spawn within bounds and move from right to left.** |
| **1.6 Shooting** | **Have the ability to use the laser vision to shoot the enemies based on mouse input.** |
| **1.7 Killing** | **Hit detection between bullet and the enemies** |
| **1.8 Score** | **Have a progress bar for distance travelled and a score bar for enemies killed.** |
| **1.9 Difficulty** | **Have platforms and enemies progressively get more difficult to avoid and kill as you progress further and high scores stored in JSON file.** |
| **2.0 Animation** | **Animation of sprite and animation of enemies.** |
| **2.1 Menu** | **Have a menu page with working buttons that redirect you to new pages.** |
| **2.2 Final Touches** | **Integrate menu and make manual.** |
| **2.3 Echo** | **Final Release** |

**Sources:**

Daniel:

<https://github.com/Mrgfhci/Drop>

<https://docs.oracle.com/cd/E17802_01/j2se/javase/technologies/desktop/java3d/forDevelopers/j3dapi/javax/vecmath/Vector2d.html>

<http://2oip.sgrondin.ca/ss09/Shoot.html>

David:

- Orthographic camera with sprite left and right movement

<http://stackoverflow.com/questions/18553209/orthographic-camera-and-selecting-objects-with-raycast>

-Basic LibGdx foundations

<http://www.gamefromscratch.com/page/LibGDX-Video-Tutorial-Series.aspx>

-Screen foundations

<https://github.com/WeeGunny/OverLap2dScratch>

-Font Test

<https://github.com/WeeGunny/FontTest>

-Scrolling background with velocity

<https://github.com/LittleDrEvil/Sonic-DoggoGame>

**Peer Assessment:**

David: 100

Daniel: 100