Work In Progress Report 4

Dino Dynamite LTD.

**Major Breakthroughs:**

Daniel: My DD\_HitDetectionHitPoints scratch has been my greatest breakthrough. In this scratch the dinosaur sprite has six different hit point sprites, each with their own unique size and location. I load the different textures into an array and pass them to the dinosaur sprite.

***for (int i = 0; i < 6; i++) {***

***txHitPoint[i] = new Texture("target"+i+".jpg");***

***}***

An array of vectors is also used to update the position of the hit points (***Vector2[] vHitPoints;***). In the main screen the **HitDetection** function loops through the array of hit points and prints to the debug window which parts of the dinosaur are colliding with the platforms.

David: My major breakthroughs were finding “Alferd” for unpacking sprite sheets and “codeandweb” for creating sprite sheets. I also was able to get the void Animate(Texture \_txDinoState) { to finally work with our sprites and sprite sheets. Finally with a nested if structure,. if (nAni == 0) { using a position variable like the array animation I was able to create a simplified animation class using this sprDino.Animate(txDinFor1); which allowed for the game not to take 5 minutes to load.

**Major Challenges:**

Daniel: The greatest challenge I’ve had this work in progress is the ***DD\_HitDetectionHitPoints*** scratch. I first attempted this scratch using 2D rectangles but the problem I had was getting the hit points to print to the screen. The compatibility between shaperenderer and the draw function was an issue, so later on I converted the rectangles to sprites. This made it much easier to draw to the windows, it works in the same way as the array of platform sprites. Resizing the sprites was also a huge issue. The issue was solved by loading in an array different textures.

David: My major challenges had to do with my poor photoshop skills which made creating sprites difficult and then with having multiple custom sprites even though the pngs are the same they do not evenly placed when using the texture packer which made my animation with an array difficult, fH = txSheet.getHeight() / 7;. Also since I don’t have a right-left animation the cycle with positioning doesn’t work properly. Also the animation class in the sprDino using an array was causing RAM issues, which caused me not to use the sprDino class at all.

**Scratches:**

Daniel:

DD\_HitDetectionHitPoints-The purpose of this scratch is to add hit points to the dinosaur. The hit points are an array of sprites within the dinosaur sprite itself. By adding hit points, we are hoping to make hit detection between the dinosaur sprite much smoother, by using the ***.overlaps()*** function on the individual hit points instead of the bounding rectangle of the entire dinosaur sprite. The basic function of this scratch is to identify which hit points are colliding with the platform sprites.

David:

DD\_AnimationSprites-This simply shows the sprite sheet I created with directional key movement and holds my 7 sprites I photoshopped.

DD\_AnimationArray- This loads the sprite sheet into the main game with an array of animations, it then draws the sprite based on the nPos variable but you have to eliminate the sprDino class in order to work properly. Also the scalability is horrible as it cuts the sprite as soon as it is not in the texture packer.

DD\_AnimationIndividual- This loads the 7 individual sprites as textures into the game and with nested if structures and the animation class it passes each texture through the main in sprDino and then back to render. The input processor then sets a nAni variable which changes the texture that is reassigned to sprDino.animation.

**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>

-Sprite Animation

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

-Font Test

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

-Scrolling background with velocity

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

**Peer Assessment:**

David: 100

Daniel: 100