Catani WIP 5

# Breakthroughs:

**Firebase:**

A big breakthrough in this WIP was firebase connection. Started by using default firebase realtime database code and setting up my database on the firebase site. After doing some research I realized you can not implement firebase code directly into Libgdx without a library to interface. So I found a library Gdx-Fireapp <https://github.com/mk-5/gdx-fireapp>. At first I couldn’t upload integers to the database because there is an issue with the library where you can only upload strings to the database, so I had to parse ints to strings to interact with the database. After doing I created the firebase scratch which is described in detail below.

**Updated Object Spawning:**

We decided that the game was too difficult with all the obstacles spawning at the same time. So I decided to completely redo the way we spawn our obstacles. We talked about this for a few minutes, and came to the decision that it would be best to change obstacles by adding and removing them from the SprObstacles array we have. This is how it works. When an obstacle goes off the screen, I clear the array using

obstacles.clear();

Then I choose a random integer. The amount of integers the program can choose from is decided by how many meters you have traveled.:

if (nScore < 15) {

nObstacle = MathUtils.random(0, 1);

} else if (nScore >= 15 && nScore < 30) {

nObstacle = MathUtils.random(0, 2);

}

Then I set the X coordinate of the “obstacle” to the worldwidtch + 300 before I add an obstacle to the “obstacle” array:

obstacle.setX(Constants.WORLDWIDTH + 200);

And then finally I add 1 obstacle to the array and redraw each one with their own personal redraw function

if (nObstacle == 0) {

obstacles.add(obsPitfall);

obsPitfall.reDraw();

} else if (nObstacle == 1) {

obstacles.add(obsTree);

obsTree.reDraw();

}

**SprChar class update:**

The SprChar class was really messy, I was using two sprite sheets and two different textures to draw the animation. I got rid of this, and merged the two sprite sheets into an image called “SprAniPlayer.png”. I set this texture as the super texture:

super(new Texture("SprAniPlayer.png"));

And then I redid my “PrepareAnimation” function to look like this:

tmp = TextureRegion.split(this.getTexture(), this.getTexture().getWidth() / columns, this.getTexture().getHeight() / rows);

txtregRunLeft = new TextureRegion[4];

txtRegRunRight = new TextureRegion[4];

txtRegRunRightInv = new TextureRegion[4];

for (int i = 2; i < rows; i++) {

for (int j = 0; j < columns; j++) {

txtregRunLeft[index++] = tmp[i][j];

}

}

index = 0;

for (int i = 1; i < rows - 1; i++) {

for (int j = 0; j < columns; j++) {

txtRegRunRight[index++] = tmp[i][j];

}

}

index = 0;

for (int i = 0; i < rows - 2; i++) {

for (int j = 0; j < columns; j++) {

txtRegRunRightInv[index++] = tmp[i][j];

}

}

AnmCreateAnimation = new Animation[3];

AnmCreateAnimation[0] = new Animation<TextureRegion>(0.10f, txtregRunLeft);

AnmCreateAnimation[1] = new Animation<TextureRegion>(0.10f, txtRegRunRight);

AnmCreateAnimation[2] = new Animation<TextureRegion>(0.10f, txtRegRunRightInv);

# Scratch Description: Firebase-Test

In this scratch it displays a number in the center of the screen , if you tap the left side of the screen it subtracts from the number and if you tap the right side of the screen it adds one to the number. This number is connected to the database, so if the numbers changes on one instance of the app it changes on all instances. So both Tim and I can manipulate the number on our phones as well as each other's. The code is relatively simple it reads from the database every from and updates the database value with the touchDown function. This is the function that updates the database value:

public void updateNumtouches(){

GdxFIRDatabase.instance().inReference("/Tapapp").readValue(Datacache.class, new DataCallback<Datacache>() {

@MapConversion(Datacache.class)

@Override

public void onData(Datacache data) {

Datacache dataRec = data;

nTouches = Integer.parseInt(data.getWord());

}

@Override

public void onError(Exception e) {

System.err.println(e);

}

});

}

# 

# Challenges:

**Changing the way I draw animations to the way Grondin did it:**

Mr. Grondin had a different way of doing the animations so I attempted to do it his way. The thing is it does not work if you have a sprite sheet that is not square. I had Mr. Grondin help me and he made something that I thought worked, However it did not for some reason. I ended up giving up on it and going back to update my old animatons.

# 

# Updated Release Schedule:

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| **1.0** | **Scrolling background and ability to flip gravity** |
| **2.0** | **Scrolling obstacles with hit detection** |
| **3.0** | **Add larger variety of obstacles such as pitfalls and spawn obstacles with increasing difficulty for the player as score progresses** |
| **3.5** | **Add coin with random spawn locations** |
| **4.0** | **Develop Main Menu and Gameover screen that transition into one another as well as Game screen** |
| **4.5** | **Add dynamic score in meters and speed up scrolling background as game progresses** |
| **5.0** | **Add complex obstacle spawning. Having only one obstacles spawn at one time.** |
| **6.0** | **Attempt cloud based score and possible multiplayer on android devices** |