**Work in Progress Report**

Major developments/breakthroughs(reference specific code please):

One of the major issues that we ran into was getting multiple classes to all render. The solution to this problem was to have the android launcher initialise a main class and then has the main class makes instances of the MainCharacter class and The MakeButtons Class.

This is the Android launcher and instead of trying to initialize each class a Main class is Initialized

public class AndroidLauncher extends AndroidApplication {

@Override

protected void onCreate (Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

AndroidApplicationConfiguration config = new AndroidApplicationConfiguration();

initialize(new Main(), config);

}

}

This is the main class that makes instances of the Classes

public class Main implements ApplicationListener {

MainCharacter mainChracter;

MakeButtons makeButtons;

@Override

public void create() {

makeButtons = new MakeButtons();

mainChracter = new MainCharacter();

mainChracter.create();

makeButtons.create();

}

@Override

public void render() {

mainChracter.render();

makeButtons.render();

}

Major Challenges/setbacks( reference specific code please):

-Initially we had problems connecting to the internet on our laptops, we have found that we cannot use Google Chrome on our laptop so we must use Firefox instead.

-Another Major Challenge is putting the projects onto GitHub, Weihan helped us with this.

-The main challenge was getting multiple classes to render properly. It turns out the way I was trying to do it was overriding one Initalize with another.

public class AndroidLauncher extends AndroidApplication {

@Override

protected void onCreate (Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

AndroidApplicationConfiguration config = new AndroidApplicationConfiguration();

initialize(new MainCharacter());

initialize(new MakeButtons()); //This was Overriding the first the previous initialize

}

}

Any modifications to your specifications/release schedule:

revised release schedule.

Basically put beta and gamma into alpha this will likely allow for more features in the future.

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| **Alpha** | **Have Animated Character on screen and Buttons that control the direction of a walking animation\*** |
| **Beta** | **Make the arrow buttons on screen move the chracter(Remove)\*** |
| **Gamma** | **Make the character rotate according to the direction of motion(Remove)\*** |
| **Delta** | **Make a base enemy that will follow the character after the character comes in close proximity to it** |
| **Epsilon** | **Make a “Nest” That will spawn The enemies when the character is close and at a set rate untill the “Nest” is destroid** |
| **Zeta** | **Make the first map/level with just walls and floor** |
| **Eta** | **Add potions that the character can store one at a time and use when they choose** |
| **Theta** | **Add keys that the character can hold multiples of** |
| **Iota** | **Add door that will be opened by the chorisponding key** |
| **Kappa** | **Add the boss at the end of the level** |
| **Lambda** | **Add the ability to save the state of the game** |
| **Mu** | **Add music** |
| **Nu** | **Add a start menu with Start, Load, New, and Controls** |
| **Xi** | **Add options to the start menu for music and difficulty** |
| **Omicron** | **Check points that you can return to when you die** |
| **Pi** |  |
| **Rho** |  |
| **Sigma** |  |
| **Tau** |  |

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

Describe the generic concept you needed to test out:

Scratch1 Testing animation.

Scratch2 Testing Buttons.

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

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

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

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

In the testing animation Link it shows how to take a sprite sheet and split it into an animation by looping through a 2d array that holds the rows and columns of the sprite sheet.

In the Testing buttons link it shows how to make a TextButton and attach an image to it using .pack files with .png images as well as so how to have a button show a different image when pressed.

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

The main issue was not entirely understanding the structure of Android programming and how the AndroidLauncher class works. I thought that Initializing a class was making an instance of it however that was not the case. The way around this was to Initialize a main class that makes instances of each class.