**Work in Progress Report**

Major developments/breakthroughs(reference specific code please):

One of our major developments was finally getting our Collision Detection and camera scrolling system to work after a few weeks of work. We changed up how we were controlling our camera in order to accomplish this. We used a combination of clamping our Orthographic Camera and clamping the player to prevent the camera and the player from being able to move past the limits of the game map. We then imported the collision detection system using object layers from a previous scratch to control collision detection objects within the map, and everything worked well, so our plan now is to integrate this into the main game.

Another one of our breakthroughs was getting the game to save the player’s x and y-coordinates for the next time the game is run, using the Preferences feature of LibGdx.

Another one of our major breakthroughs was getting the hero’s health to decrease random amounts on a set timer.

Major Challenges/setbacks( reference specific code please):

One of the major setbacks we overcame with the issue of the camera and the player being able to move past the limits of the Tiled Map. We attempted to use object layers within Tiled to prevent the player from being able to move any further, but with our previous camera movement system the game prevented the player from moving 1 screen over, even if there was no map edge there. After a lot of research and trial and error, we changed our Orthographic Camera to follow the player’s movement, and we discovered that we could clamp the camera and the player’s movement to prevent both from moving off of the map.

Another issue we ran into was right now we are saving dialog in the prefs folder, which is hindsight was not the best idea. This is because the text was modified on our computer, but when someone else pulls it they don’t have the dialog text on their computer, and we can’t move the prefs folder into the project. We plan on working on getting a xml file in the project with the dialog first thing next week.

Any modifications to your specifications/release schedule:

We have worked on the battle screen scratch, completing the majority of our original goals for the battle screen like a “you lose” screen, an attack button that depletes health, and the hero’s health decreasing after you attack the enemy. This was all done before integrating our collision detection and a basic battle screen into the main project, which originally was supposed to be done before the battle screen specifics. We’ve also added a release to incorporate the element of saving the game.

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

Describe the generic concept you needed to test out:

We created a few scratch programs to test out a few generic concepts:

1. Tiled Camera Scratch: Our goal with this scratch was to develop an effectively camera scrolling system, that was limited to moving within the game map.
2. Gamesave Scratch: With this scratch, we wanted to create a program that saved player progress, in this case, the x and y coordinates.
3. Battle Screen Scratch - This time we focused on having the health bar change colours when you get lower and lower on health. As well, we worked on when you click the attack button a black box pops up that should have some dialog on it that disappears after a set time (right now it’s 2 second), and then the hero’s health decreases a random amount.
4. Overlap2D Scratch: Just testing the Overlap2D system, it seems like a very good way to handle assets and creating maps fast, however it seems more well suited to side scrollers and platformers, so we might not end up implementing it into our rpg. It maybe useful for setting up UI but it we will continue to use tiled and the default way of reading in assets.

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

1. Tiled Camera Scratch: We used the following sources:

http://www.gamefromscratch.com/post/2014/04/16/LibGDX-Tutorial-11-Tiled-Maps-Part-1-Simple-Orthogonal-Maps.aspx

https://github.com/libgdx/libgdx/wiki/Orthographic-camera

http://gamedev.stackexchange.com/questions/57325/how-to-get-width-and-height-of-tiledmap-in-the-latest-version-of-libgdx

<http://gamedev.stackexchange.com/questions/74926/libgdx-keep-camera-within-bounds-of-tiledmap>

1. Gamesave Scratch: We used the following sources:

https://github.com/libgdx/libgdx/wiki/Preferences http://stackoverflow.com/questions/18607689/how-to-save-game-state-preferences-in-android-using-libgdx

1. Battle Screen Scratch:

* -Prefs for the dialog:
  + Ashleigh’s save scratch
* -Timer for the decreasing health:
  + http://atsiitech.blogspot.ca/2013/09/adding-15-second-timer-to-your-games.html

1. Overlap2D Scratch: <https://www.youtube.com/watch?v=bhvHm2sM0qo>

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

-Battle Screen - Learned how to use a timer in libgdx, how to schedule events, and how to make things disappear. As well, we also learned about what not to do for reading the dialog into the project, and therefore we now know for when we try and use dialog on the map later on.

Tiled Camera Scratch: The code allows the player to move around a generic map, but the player is not allowed to intersect any obstacles or leave the game map. We figured out that sometimes it is easier to use a simpler solution, instead of using complicated collision detection with several rectangles, we instead just limited x and Y values. We also learned about how to clamp the camera and the player coordinates so that the player was not able to move past the limits of the game map.

GameSave Scratch: The code allows a player sprite to move around the screen, saving its x and y-coordinates to an xml file on the computer. When the player closes and reloads the game, it starts the player sprite at those same x and y-coordinates. This scratch taught us what Preferences are and how they work, so that we can retain information about our game for future reference.

Overlap 2D Scratch: We learned a lot about Overlap2D and how to use it, even though we will not be using it for our game at this moment in time, though we may find a use for it later on.

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

We have not yet integrated these scratches into our major project, but this is our first priority for next week. Due to the fact that these scratches are so integral to our main game, it was important to get them fully functional before we integrated them. However the main problem we will have while integrating these scratch is that the map based scratches are all in one file splitting them correctly into multiple files will be difficult.