Name: Larry

Daily Work Log

Date/TIme	Things done	Things to do
Friday, June 3rd	- brainstormed possible project ideas - skimmed through all of the project resources	- Decide what type of project I’m going to make - choose a project resource to use
Monday, June 6th	- decided to make a game of some sort - looked through the tutorials of the game development project resources - skimmed through some of the beginner documentation of FXGL and LibGDX	- decide on the specifics of what type of game I’m going to create - choose one of the resources (game development frameworks) to create my game with (or figure out whether or not I can use both)
Tuesday, June 7th	- decided to make a 2d rpg reminiscent of the original NES Legend of Zelda, with the goal of creating a single dungeon players can traverse through - decided to use FXGL to create my game (it seemed to be the easiest to use for my project)	- read the documentation / github tutorials for FXGL and install it on my computer - that's it, figure out how to use the development framework to hopefully make a functional game
Wednesday, June 8th	- began brainstorming the specifics of my game - tried reading the documentation, got confused, will try again tomorrow	- try and understand the FXGL documentation (I swear this is harder to understand than legal jargon)
Thursday, June 9th	- Followed the first FXGL tutorial and created a base “game” with player (a blue square) movement and UI keeping track of the number of pixels moved	- Add sprite animation to the blue square - restrict movement so the player can only move horizontally or vertically at one time (no diagonal movement to match the original Zelda games)
Friday, June 10th	- Found a suitable spritesheet to hopefully use - Followed the tutorial and successfully imported images and sounds into the game	- The entire sprite sheet image represents the player, figure out how to make actual animation - figure out how to have the sound effects run when I want them to (right now the walking sound effects endlessly loops

		whenever the player walks, and never stops even though I tried to code it to stop)
Monday, June 13th	- Over the weekend, I figured out how to create sprite animations in FXGL (I had to watch 2 youtube videos and follow 3 different written guides, but it's done) - I had to crop the sprite sheet as FXGL automatically divides the spritesheets into equal sections with each section representing one frame, and that is the only way to do sprite animation - Had to make an AnimationComponent class in order to implement the animation, and the movement code had to be transferred over the AnimationComponent class for the animations to properly register	- The code is still incredibly buggy - as of right now: - you can walk diagonally when walking up or down but not when walking left or right - the player "moonwalks" when you press both right and left at the same time and then stop walking (which happens when you try to change direction and you momentarily hold down both direction keys) - Have to fix the bugs
Tuesday, June 14th	- Fixed the bugs that occurred during movement - Tried to add an overworld map to the game using Tiled map creator and an online spritesheet, but couldn't get it to load	- Figure out how to implement world maps from Tiled map (.tmx files)
Wednesday, June 15th	- Fixed the problem that prevented .tmx files from loading - there were 2 issues, both of which were silly: - I had to place the files in a folder called "levels"; the tutorial I was following was using an older version that placed them in a folder called "tmx" - I had to check a box labeled "embed in map" when importing the spritesheet png - this was only mentioned in a youtube video about coding a platformer	- Fix the player movement (right now the player keeps on moving after controls are released, and can move diagonally) - Add hazards and enemies that spawn with the level

	- Added collisions to the player and entities spawned from the txm level files - this took a long time to implement - I had to add a PhysicsComponent to the player and object, which meant that in order for the player to move I had to use PhysicsComponent.Velocity, which meant I had to rewrite the player movement code for the third time	
Thursday, June 16th	- Fixed the problem with player movement by declaring UserActions for movement in each of the cardinal directions (I was using a convenience method before, now I created instances of UserAction as that allows me to set velocity to 0 once the key is released) - Added playerHealth to the game which is currently represented by a number in the top left - Added hazards that the player takes damage upon coming into contact with	- The program should have a message pop up saying “you died” when playerHealth reaches zero and restart the game, but health just becomes negative. Fix this. - The player should be knocked back upon taking damage, but they instead stay inside the hazard and continue to take damage every frame. Fix this.
Friday, June 17th	- Fixed the problem with playerHealth so now when health reaches 0, a “you died” notification appears and the player dies and returns to the main menu like they should - Instead of staying in place and taking damage every frame, the player is now knocked away, just to a wrong location (sometimes further into the hazard, causing them to keep taking damage). Also, they should be stunned for a while after taking damage, but the stun is dysfunctional (they are stunned if they release the movement keys, but not if they keep holding it, and the animations for the player while they are stunned are broken) - Tried to follow an online tutorial to implement enemies, projectiles	- Fix the issue with knockback and stun upon taking damage - Fix the issue with sword and arrow entities spawning at (0,0)

	(arrow entities), and attacks (sword entity). Enemies with physicsComponents don't move unless given AI (which is not a priority), but the sword and arrows spawn at (0,0) instead of where they should no matter what I do.	
Monday, June 20th	- Over the weekend: - Fixed the issue with player knockback and stun when they take damage, so now the player is moved to the proper location and the player is unable to move/change their animations - Tried to add warp zones that would allow the player to move to different zones (ex. enter a door in the overworld to go into a house), but gave up after a couple hours - Tried for 4 hours to make it so that sword and arrow entities don't spawn at (0,0), but the documentation is non-existent, and my code seems no different from that of online tutorials who have working programs - Removed some redundancies in the code like code that was added to AnimationComponent when it still controlled player movement	- Accept that I won't be able to fix the spawn error - Try to fix the program crash that happens when I try to change the world scene (load the world from a different tiled map)
Tuesday, June 21st	- Tried to fix the warp zone error by spending many hours metaphorically banging my head against a wall trying to find what was wrong with my code, which included reading and trying to comprehend the uncommented FXGL demo programs - I think what I would have to do is create 2 new classes: a loading class that makes it so the game does not try to run as usual while a new level is in the process of being loaded, and a main loading class that tells	- Finish the commenting for the AnimationComponent class (and removing the redundant code / cleaning up the messy functional code) - Prepare for my presentation

	the game what coordinates and what map to load - I tried implemented that, but gave up after realizing it would probably take too much time	
Wednesday, June 22nd	- Uploaded the code to github after finishing the commenting for the AnimationComponent class - Finished the reflection	