Burrell, David
300209541
SWEN225
Project Individual Report

Individual Contribution:
I worked on the Application package of the game, mainly focusing on the front end of the program, running the game loop, switching states, buttons and UI features. As well as integrations of other packages into the running of the application. A few of the most important parts I worked on were:

Getting the renderer to show within the application window and performing correctly with user interaction.

Getting teleporters integrated and implemented.

Updating the map when the player interacts with an interactable item.

Detailed Appraisal:
1. Main responsibility:

The application package is responsible for handling the front end, creating the window and any subsequent dialog boxes and confirmation windows of the program. As well as integrating other packages to combine into a full game.
Retrieving and computing user input was also necessary and implemented using simple switch statements and method calls/Boolean toggles.
2. Component Integration:

My package had to work closely with both the renderer package and the Gameworld package. With the renderer I had to implement the render methods for refreshing the camera location and the screen that can be seen from the current camera location into the game loop using AnimationTimer, I also need to initialise and correctly use all the objects from renderer.
On top of that I had to make sure the correct version of the map and player were ready by the time the renderer needed them requiring my integration with the Gameworld to be solid. With the Gameworld package, I had to make sure we were correctly updating the world at the correct times, such as during player interaction. A lot of my work with the Gameworld package was with event handling of either the on screen Buttons during the menus, or with direct user input.
With handling the user input I went ahead and made the proper methods within some of the Gameworld classes and I went and built the framework of our interaction checks for certain objects, as I needed to have these checks return certain parameters to my package to be able to display the item pickups on the players UI.
This leads to my interaction with the persistence package, as making sure the Gameworld is in order is needed for saving and loading, other than that the pass off to persistence was mainly through two methods, save game and load game, in which I created the dialogs and set the event handler to the persistence package, another nice feature I implemented here was the drop down menu for selecting previous saves and having the last used save name preloaded into the textField when bringing up the load and save functions.
3. Quality design aspects:

A lot of my work with the menus and dialogs makes use of lambdas for event handling, as with the many Buttons I was using would create some awful looking code blocks. Using switch statements when I could to keep the code cleaner than using too many if statements. I am happy with my main menu background feature; however, the implementation isn't the best. As well as trying to avoid using too many duplicate code blocks when and writing methods to use in multiple places.
I Eventually cleaned up my Main class as it was being overly complicated and hard to find certain features and methods when I needed to, so I implemented some utility classes to hold methods that were similar in nature.
4. Design aspects in need of improvement:

Due to unforeseen functionality problems many of our potential features were neglected so that we could get the game in a state we were happy to submit. I was unable to finish a nice-looking HUD for the player and went with a simple functional display. I also was trying to get a message to display for finishing each level.
Obviously, I would have liked to go back and clean up a lot of my code that I threw in during the last couple of days work, the render method that my AnimationTimer calls had a few overly long method call chains that could've been cut down.
As for my work outside of application I would've liked to clean up and properly implement the teleporters and have object interaction been a far more elegant process.
And figuring out the problem with the resetting the game, amongst other strange functionality.
5. Alternative approaches:

Many alternative ideas were brainstormed, potential top down 2dish games like pokemon, and minit, to a traditional point and click adventure as the project handout images showed. As time went on most alternative approaches came as ideas to short cut or bypass problems we were having in figuring out how to render things and calculate other issues.
As such during the final week we made many alternative approaches as a way of actually getting a game with some functionality and then possibly working in the more complex goals we were aiming for, given time.

onjaim@ecs.vuw.ac.nz 5

colemamich3@ecs.vuw.ac.nz 1

gandhishiv@ecs.vuw.ac.nz 5

Tom.Clark@ecs.vuw.ac.nz 5