**Software Implementation and Testing Document**

**For**

**Group 18**

04/28/2019

Version 2.0

**Authors**:

Matthew Wix

Ryan Kenney

Michael Tesfaye

Riley Garrison

# Programming Languages

HTML, CSS, JavaScript

# Platforms, APIs, Databases, and other technologies used

* Firebase – user authentication
* GIMP – image manipulation for creating maps and editing spritesheets

# Execution-based Functional Testing

* **Player movement –** tested by using all arrow keys to see if the player could move in all directions, checked for movement when multiple keys are pressed and held at the same time, ensured that movement stopped when attempting to move outside of the boundaries of the map.
* **Attacking enemies –** tested by attacking an enemy from all different directions to make sure hits were registered.
* **Enemy damage calculation –** tested by attacking enemies and watching their health bar decrease after every hit, as well as make sure the enemy was flagged dead after its health has reached 0.
* **Player takes damage –** tested by approaching the enemy from different directions to make sure the player took damage.
* **Player dies –** tested by letting the enemy attack the player until they were flagged as dead and then navigating every screen/menu option to ensure everything functioned properly.
* **Logging in –** tested by logging in with correct credentials, using an invalid password, using an invalid username, and having no internet connection.
* **Signing up –** tested by signing up with correct credentials, using a username that already exists, using a password with too short of a length, and having no internet connection.
* **Logging out** – tested by logging out from every different screen/menu option.
* **Pausing the game –** tested by opening up the menu and ensuring the menu could be navigated and its resume play and exit game function worked properly.
* **Handle keyboard input –** tested by pressing every key that is used by the game in different order at random times to make sure each key press was detected and functioned as it was supposed to.
* **Traveling between maps –** tested by having the player cross the boundary that would take them to the next map and ensuring the player arrived in the map the were supposed to for every map that was available.
* **Choosing a map location to travel to –** tested by creating new users to make sure only map locations that had been visited before showed up in the list of options. Selected all map options available and ensured they took the player to the intended destination.
* **Dialogue with NPCs –** tested by pressing the appropriate key to interact with the NPC on every NPC that was available and making sure each one had a unique, fully functioning dialogue box.
* **Buying items from NPCs –** tested by buying every item in the NPC’s inventory and ensuring the correct amount of gold was taken from the player’s inventory as well as having the correct amount of items placed into their inventory. Also tested the case where the player attempted to buy items without having any money.
* **Using items from the player’s inventory –** tested by selecting every item the player had in their inventory and either equipping it if it were a weapon or consuming it if it were a consumable item (i.e. a potion). Ensured the corresponding effect took place and the item was removed from the player’s inventory after use.
* **Saving the game –** tested by opening the save menu in every map and saving the game. Tested saving multiple times in a row and checking the save file screen from the main menu to ensure the player’s information had been updated. Also exited the game and reentered it multiple times on every map to make sure save information was retained.

# Execution-based Non-Functional Testing

* The Chrome, Firefox, Opera, and Microsoft Edge browsers were tested with each test case to ensure that the game functioned properly on all of them.
* The game was tested on multiple different devices to ensure the graphics looked the same on different screen sizes.
* The game was tested with no internet connection to see if error handlers worked properly and how having no internet connection affected the game.
* The game was tested for smooth, consistent 60 FPS through hours of gameplay and it was verified that it was able to handle large amounts of enemies and tiles on screen at once even on computers with relatively low internet speeds (~1Mb/s)
* Repeated saves and loads were performed to ensure there was no large latency from connecting to the database and updating the game state
* Repeated reads/writes to the external database were performed to ensure it was consistent and did not crash under any circumstances

# Non-Execution-based testing

Non-execution-based testing was performed by having each team member look over the code for any potential bugs or logical inconsistencies. After critiquing the code, changes and alterations were made accordingly to ensure the code was sensible and would function properly under all test cases for this iteration.