Testing Documentation

Dark Ascent

Turkish Van

Stephen Aldred

Noelle Alston

Peter Gatira

Bhoj Raj Pandey

CMSC 447: Software Engineering I

Dr. Nick Allgood

# Test Plan

## In Development

While developing code for the project, functions were introduced incrementally and tested each time one was finished to ensure that it was working properly before developing/implementing any more code. When tying systems together, like the main menu and the map, we ensured that each part would work seamlessly with one another, working out any bugs and edge cases with transitions between them before combining more pieces. Some helper functions that return values for Python have a test.py file to run to test if they work, but most were just tested manually by playing the game.

## Finished Product

To test the finished product, we had each of our team members play through the game and report any bugs that they found to the person who developed each part with the console log at the time, in which case the person who coded it would go through the error code and ensure that the bug was fixed. They would then test it thoroughly before adding it back to the main project.

# Test Cases

## Login & Registration

* + Ensuring that the spaces are not allowed in username or password fields.
    - Test passed: true
  + Ensuring that the function checks if the password and password confirmation match.
    - Test passed: true
  + Ensuring that the function checks if the username already exists in the database.
    - Test passed: true
  + Ensuring that the initial game items and base game moves are assigned to a new user from the ‘item.db’ to the user\_info.db.
    - Test passed: true
  + Ensuring that upon successful registration, the user is redirected to the login page.
    - Test passed: true
  + Ensuring that any database errors during registration are handled gracefully, possibly returning a user-friendly error message.
    - Test passed: true

## Main Menu

* Ensuring that when “/render-menu” is posted, it redirects to the ‘main\_menu’ endpoint
  + Test passed: true
* Ensuring that when a new player accesses the ‘main\_menu’, the newPlayerFlag is set to 1.
  + Test passed: true
* Ensuring a returning player setup correctly sets ‘newPlayerFlag’ to 0 in ‘main\_menu’.
  + Test passed: true
* Ensuring “/load-game” redirects to ‘main\_menu’ with an alert if no moves are available.
  + Test passed: true
* Ensuring “/load-game” redirects to ‘/start-map’ when moves are available.
  + Test passed: true
* Ensuring “/new-game” successfully resets game data and redirects to ‘/start-map’.
  + Test passed: true

## Leaderboard

* Ensuring that the database query correctly retrieves usernames, player kills, and player deaths from the ‘login’ table.
  + Test passed: true
* Ensuring that the KDR is calculated correctly when deaths are zero and kills are greater than zero.
  + Test passed: true
* Ensuring that the KDR is calculated as zero when both kills and deaths are zero.
  + Test passed: true
* Ensuring that the KDR is correctly calculated as the ratio of kills to deaths when both are non-zero.
  + Test passed: true
* Ensuring that the function handles SQL errors gracefully, returning a user-default value.
  + Test passed: true
* Ensuring that the leaderboard ranks and displays the users’s username and the KDR in descending order.
  + Test passed: true

## Map

* Generates all the health, gold, items, and moves that the user has with no issues
  + Test passed: true
* When clicking the “Leave” button, the user is redirected back to the Main Menu
  + Test passed: true
* When clicking the “Combat” button, the user is redirected back to Combat
  + Test passed: true
* When clicking the “Leave” button, the user is redirected back to the Shop
  + Test passed: true

## Combat

* Ensuring that when “Attack” is clicked, it loads the user’s moves that they have
  + Test passed: true
* Ensuring that when “Items” is clicked, it loads the user’s items that they have
  + Test passed: true
* Ensuring that when “Items” is clicked, and they have an equipped item, it doesn’t include it in the list that is presented to the player
  + Test passed: true
* Ensuring that when an move is clicked, the move is used in combat and the turns of each enemy occur
  + Test passed: true
* Ensuring that when an move is clicked, and it has a cooldown, the cooldown starts
  + Test passed: true
* Ensuring that when a move is clicked, and it has an active cooldown, it cannot be used by the player:
  + Test passed: true
* Ensuring that when a move’s cooldown is no longer active, the player can use that move again:
  + Test passed: true
* Ensuring that when an item is clicked, the item is used and decremented in the user’s inventory.
  + Test Passed: true
* Ensuring that when a menu option is clicked, it is indicated with a color change.
  + Test passed: true
* Ensuring that when the user has confirmed their action for the turn, they are unable to make any other inputs
  + Test passed: true
* Ensuring that when a turn ends the log stays there and the player can make their next turn
  + Test passed: true

## Shop

* Ensuring that when user hits ‘Leave’ button, it brings them back to the map
  + Test passed: true
* Ensuring that when a user hits “Confirm” after selecting an item and they have enough money, it adds it to their inventory and updates their currency
  + Test passed: true
* Ensuring that when a user hits “Confirm” and they haven’t selected anything, it prevents them from buying something. Currency will stay the same.
  + Test passed: true
* Ensuring that when a user hits “Confirm” and they don’t have enough gp, it will not decrease their currency and not add it to their inventory.
  + Test passed: true