**Dungeon Destroyer Final Document**

1: Overview of the project

Our project, Dungeon Destroyer, is a dungeon crawler game in which the objective is to defeat the dark mage and collect the treasure at the end of the dungeon. To do so, you have to traverse through multiple floors filled with snakes, goblins, ghosts, and spiders by choosing a class and either fighting or fleeing all of them. Only once you traverse through all of the floors and defeat the dark mage will you have access to the treasure. We accomplished creating an amazing project with map exploration, NPC combat, and more allowing us to achieve our goals. We did all of this using HTML for our menus and JavaScript for our game. Our final product is a playable, balanced, and enjoyable dungeon with 4 levels and a final boss.

Our team is comprised of:

Ryan Luckinbill - Team Leader, Master of Maps (Tilemaps and Collision)

Dakota Moore - Count of Character Creation (NPCs and Characters)

Alec Morris - Consultant of Combat (Combat)

Austin Konsor -Sultan of Saving (Saving)

Grant Duncan - Sergeant of Shurikens (Combat)

Alexander Schulz - Protector of Potions (Inventory and Potions)

James Eenhuis - Duke of Documentation (Documentation)

Overall, I would say we are around an intermediate to advanced level of programming proficiency but none of us had any experience with JavaScript or Phaser.io. In total, we implemented 25 story cards with many of them being difficult to complete. The commit statistics for our master branch are listed below:

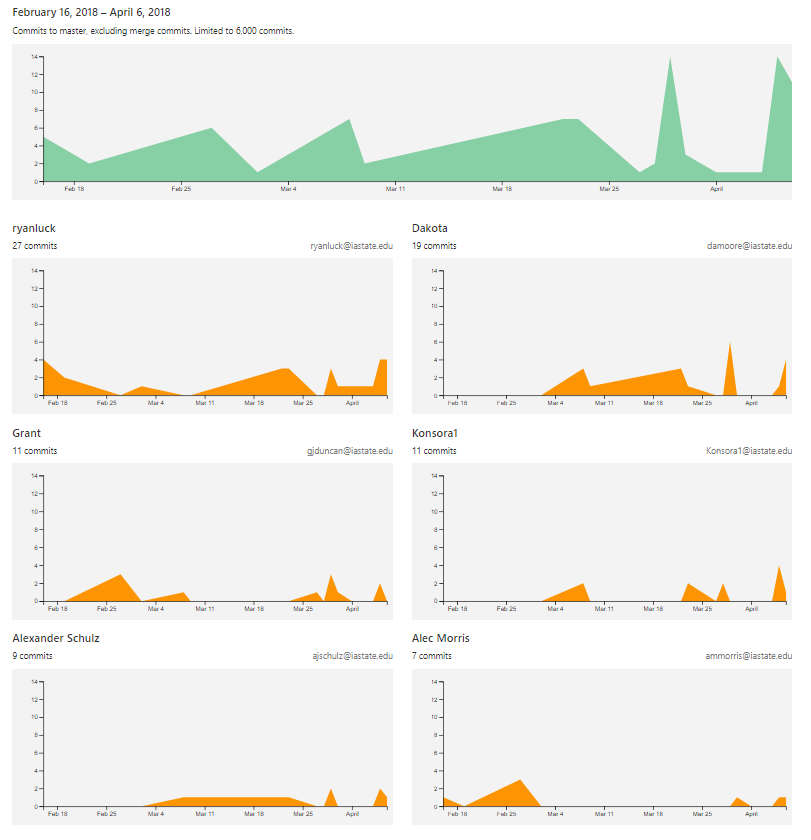


Figure: Number of Commits Made By Each Member

2: System Modeling and Design

2.1: Successful Major Story Cards

As a player I want to:

* Add movement to the character so that I can traverse the dungeon.
* View a main menu to select game options.
* Create a new dungeon when new game is pressed, and when a new floor is reached. Where is viewed as a TileMap where each sprite takes up a tile.
* Save the current game when a new floor is reached
* Continue from the floor you left on when ‘Continue’ is pressed. That includes player attributes, items and NPCs.
* Choose between 3 character classes to traverse the dungeon with.
* Spawn items throughout the map that I am able to pick up.
* Climb down stairs until final floor is reached
* View and use items in a player inventory.
* Spawn NPCs that attack me when I am near them.
* Attack NPCs (combat)
* End gates not passable until final boss dies.
* Weapon sprints “die” after their attack range.

2.2: High-level design

* Module Diagram

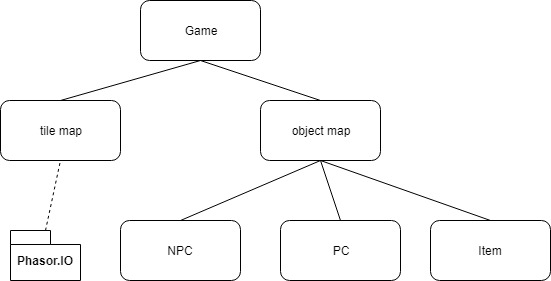


Figure: Module Diagram of Web Application

* Phasor.io was the only package used outside of our main game and that is mostly used for image processing so we had a gui and sprites to see on screen
* Image placement was contained in the tile map which would then call pahsor methods to load and display images.
* The game contains two maps one for images and animation and another to take care of backend work and object interactions
* Object map contains all the objects that interact in the game
* NPC is an object that acts like a player controlled by the computer to kill the PC
* PC is the player character or the players interaction object that is moved around the Game to do a variety of things
* Items is the last object that are consumed and moved by the PC in the game to buff the PC stats to have more favorable interactions with NPCs

2.3: Interaction Models

Use-case Diagram

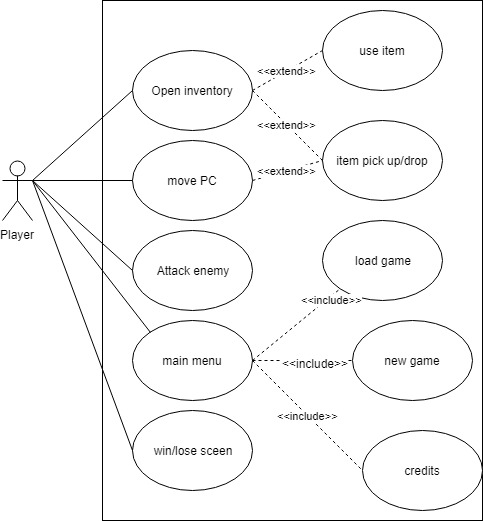


Figure: Use-Case Diagram

Activity diagram

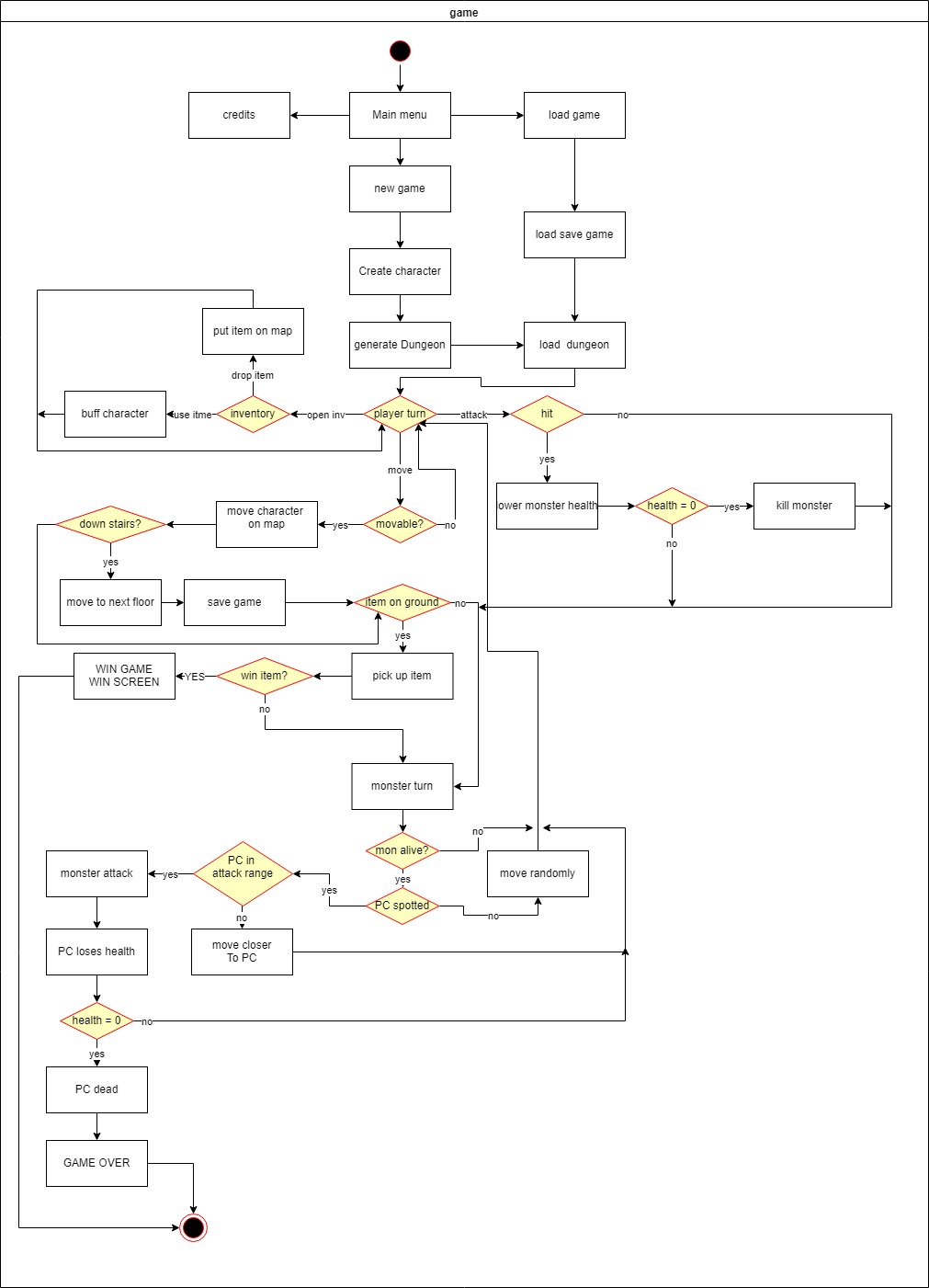


Figure: Activity Diagram

Sequence Diagram

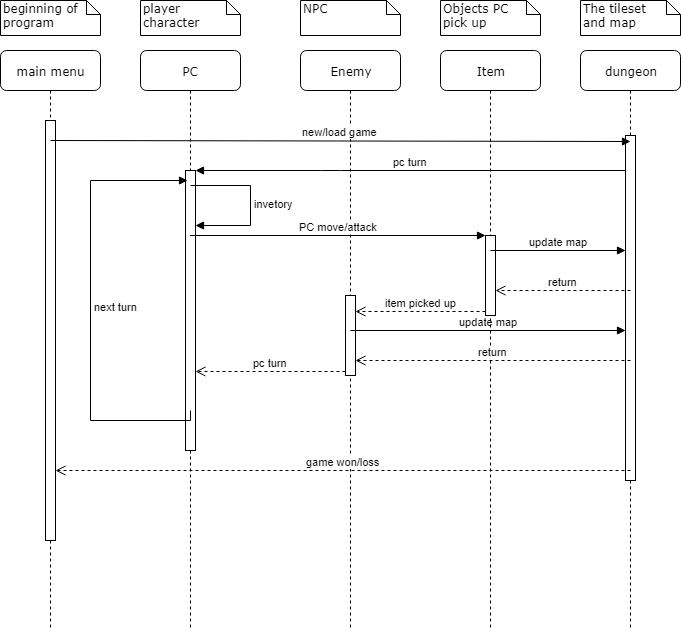
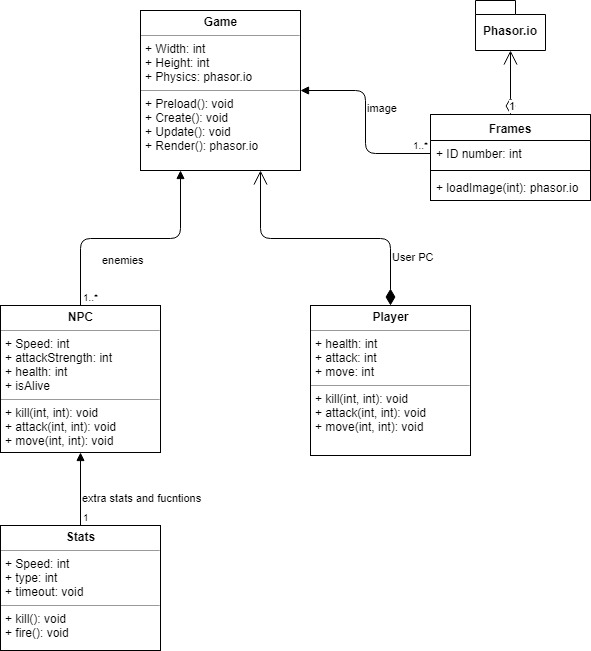


Figure: Sequence Diagram

2.4: Data Organization



2.5: Design Rationale

There were a few issues trying to understand and implement the libraries of phasor in the begging and up to the were still learning new and interesting things of the library. JavaScript was new to us all to very few of us had little to no experience with it or even html causing a difficult learning curve. Though with enough time both of these difficulties were resolved through brute force. Lastly our overall design structure changed a few times but only in ways that just moved a few objects and methods around and have not caused a total code architectural overhaul.

3: System Implementation

3.1: Implementation Outline

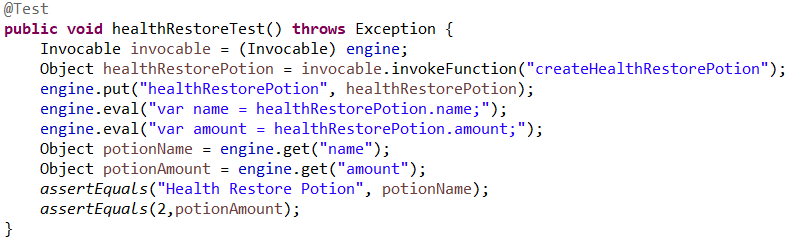
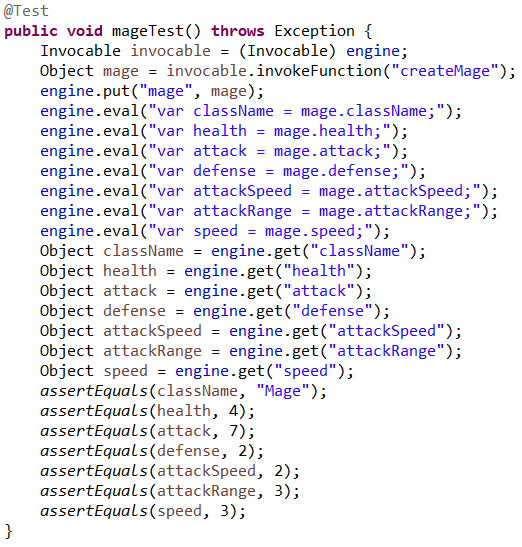
* Platform - PC through Node JS
* Front End - Phaser.io, HTML, CSS
* Back End - JavaScript, Node JS, Phaser.io
* Documentation tool - Draw.io

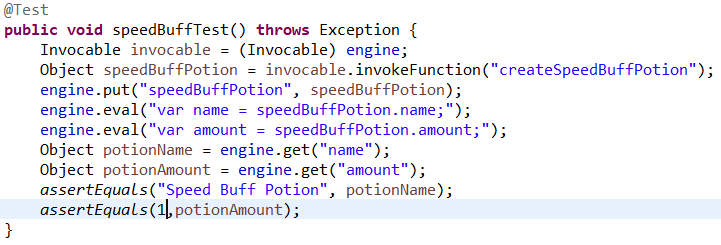
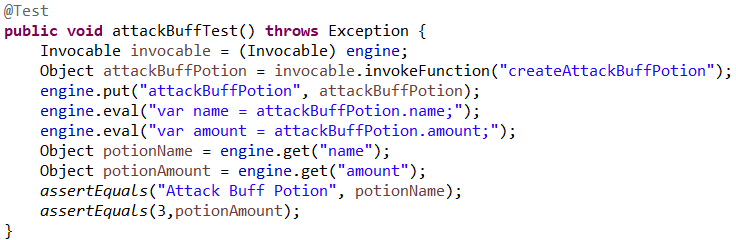
3.2: TESTING

JUnit Tests

To test our javaScript functions and objects we used an invocable script engine and input stream to pull the javascript document, then invoke a function or method that pulls it into the test file, adds it to the engine, then converts it into a java object to be tested with assertEquals.







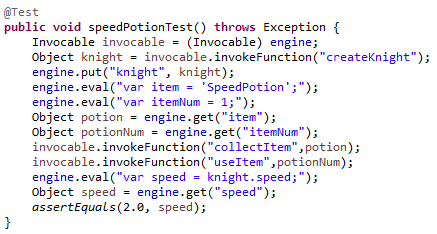


Figure: Example Test Cases

3.3: UI Description with Screenshots

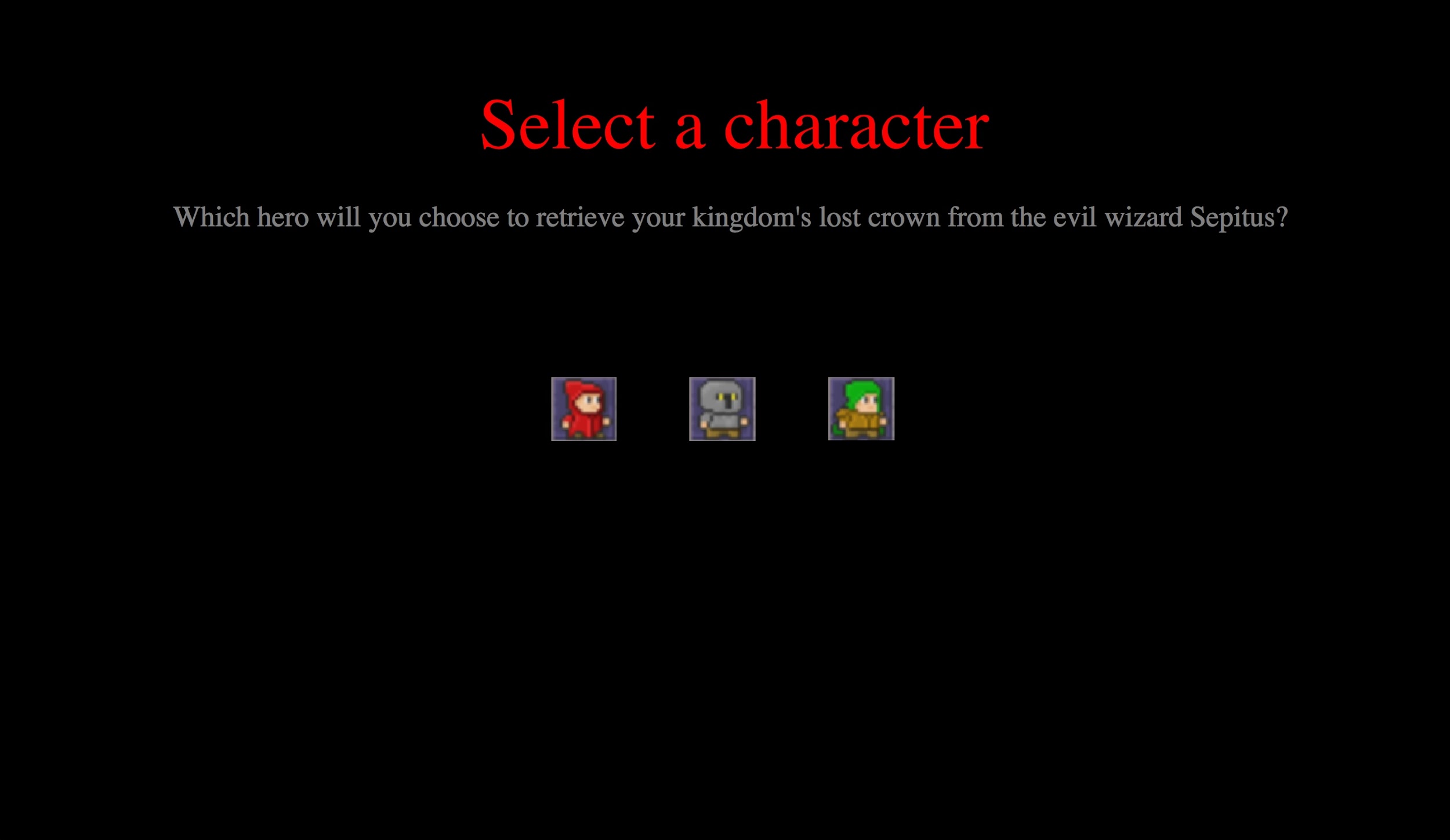
The UI is navigated through HTML with each page styled in CSS. You begin the game at the main menu (index.html), with options:

* New Game: Takes you to character\_select.html to start a brand new game with the ability to select between 3 playable characters..
* Continue Game: Takes you straight to game.html to the place you left off.
* Tutorial: Navigates to tutorial.html to show how to play the game.
* Credits: Goes to credits.html which displays each team member and their duties.

After completion of the game, you will be taken to youlose.html or youwin.html depending on if you won or lost. With options to begin the game again.



index.html



character\_select.html



game.html



youlose.html

4: Individual Task Distribution and Role Description​

Ryan Luckinbill:

I was the team leader for the project and mainly worked on the sprites used for the map, the tilemap design and display, the Map2D array of which stores the data of each tile and is the basis for the whole game, the different tile types, and collision. I also created the tutorial, and winning and losing html pages and updated the index page to make it more user friendly. I have committed 26 times to master out of the 82 total, and have had the most commits for every branch. Like all of my team, I had no experience with Phaser, Javascript, and Node.js but learned a lot on this project about all of these and project management. This was my first project working with a library like Phaser and one of my big takeaways from this project is how to work with a big library. Coming in to this project, I would have considered myself a newbie with Javascript but have done a lot of work with Java and other programing languages. I also knew a decent amount about HTML and refined my skills. This project taught me a lot about the sprints we would do every week and especially project management. Getting to know my team and how they liked to work provided an extremely valuable resource over the course of the project and helped me lead my team to do well on this project.

Dakota Moore:

I was primarily a programmer that worked on the player, NPC (including creation and movement), potion objects (including creation and placement) and sprites. I also worked on initially researching developing the player and tileMap but moved on to item and NPC creation later in the project. I worked on the following trello card topics: New dungeon on New Game pressed; Researching implementation of character; Traversing the dungeon as a player; Implementing player inventory and item creation; Detecting items on a 2D map, save and load the 2D maps with items, sprites and stats; Spawning items on the map; and having NPCs spawn and attack when near a player. I have commited 15 times to master, 15 commits to Current\_3\_8, 2 commits to charactercreation, and 1 commit to experimental-tile map, totalling 33 commits. I had no experience with the Phaser.io library, JavaScript, and very little experience with HTML, the only experience with HTML being personal side projects that did very little. I would not describe myself generally as a newbie because I would consider myself proficient in Java, C, C++, and Python and am in my fourth year of Software Engineering, but coming into this project I definitely was. I learned a lot about JavaScript and the Phaser.io framework and could easily do another project using these. I also learned how to test JavaScript files using Java JUnit tests and that is one of the most useful things I learned this semester.

Alec Morris:

I initially started by creating the menu (which has since been updated multiple times by me and others), and ultimately ended up doing a lot of work on both combat and combat animations, so I guess you could say I was mostly a programmer and a consultant to others. All in all, I had 11 commits, 7 of which to master. Grant Duncan and I utilized extreme programming, and often went back and forth between driving and consulting week-to-week, so many of each of our commits belonged to both of our work. Before this project, I had seldom used Javascript and HTML effectively, so I feel as though I gained a lot throughout the course and project. I feel as though I am now proficient in using Javascript, HTML, and CSS. I would consider myself as a “newbie” more or less before this project, and now am much more comfortable. I feel like learning the game library used, Phaser.io, will also be beneficial in my future endeavors.

Austin Konsor

I was mostly a programmer for this project. Throughout our game I have worked on saving so we are have the ability to “continue game”. All saving is done in localStorage and saved in strings. So it will save character, health, attack, speed, floor level, and inventory. This is done by auto saving every time the character goes down a floor. So each time you press continue you will start at the beginning of the floor you were just at with the same attributes and items that you had once you got there originally. I had also worked on transferring data between screens, for example, when you select a mage, it will give you the correct sprite and attributes. I had 13 total commits, which I had been working on saving myself. Prior to this project, I had no experience in Phaser or JavaScript, and had very little experience in HTML. I’d say I am fairly high experienced in all three now, and hope they will help me a great amount in my future.

Grant Duncan

I started this project by helping with the menu by making it open new pages when buttons were clicked. For the remainder of the project, I worked on combat and combat animations. I helped to make directional attack animations when the user clicks the mouse and dealt with NPC damage and deaths. Alec Morris and I worked hand-in-hand throughout this project, utilizing extreme programming. My role in the project was as a programmer and consultant. I had 12 commits, not counting merges or reverts, but most of Alec Morris and I’s commits were shared between us. Before this project, I had zero experience with HTML, JavaScript, CSS, or Phaser, and by the end, I can say that I have learned a modest amount about HTML, a lot about Javascript, and a lot about Phaser. I think that learning Javascript will be the most helpful for me in the future.

Alexander Schulz:

My first job was constructing the player selection screen, which appears between the main menu and the game screen when launching a new game. Afterwards, I constructed and implemented the player inventory object, their items, and it’s visual implementation. As such, I worked on various story cards, including detecting objects on the 2D map, implementing player inventory and item creation, the creation of the character object, as well as creating a class selection screen. For gitlab contributions, I committed 8 times to the master branch, another 7 to the current\_3\_8 branch, 2 to experimental\_tile map, and 1 to charactercreation; this brings to a total of 18 commits. Before this project, I had experience with several other programming projects and related work. In the previous summer, I had worked in the LESEM labs at Iowa State University, developing a web-based educational software tool called PEWI. Developing this suite allowed me to gain a better understanding of JavaScript, CSS, and HTML. Thus, going into this project, I was not a newbie. Going into this project, I was already familiar with Java, C, C++, MATLAB, Python, and other languages, as well as with the Git repository system. This is also due to the fact that I am a senior in Aerospace Engineering. Despite this, I did learn a few new techniques and ideas, such as the Phaser.IO framework and related libraries.

James Eenhuis:

Most of my jobs that contained to this project was mostly dedicated to the documentations and small changes to overall design and bug fixing. I composed and created all reports and diagrams for this project even the ones on this document even if they seem a bit messy but that was the work we generated trying to work with phasor and javascript. My bug fixing was not on trello since it never was a story card to do but has been needed here and there. My pushes to git are few due to my focus on documentation and design. Mostly I just pulled and analyzed the code to make the reports and communicate between members how code interacted. I had no idea what java script was or even how html worked or what script languages were and worked. I now know a little bit of how java script is a function langue mixed with a script language but I still need better understanding with what a script language is and how it even works before I think I can work proficiently with it. I am completely new to script languages and servers which is seen from my 2 weeks trying to get my git and node.js server working just to even try to code and begin to understand javascript (I still don’t know why I need a server to run script languages Though I believe is has something to do with compiling on the fly? Still need to learn more of this in the summer). But I have gained extensive knowledge on how proper and efficient design techniques work that will be useful in the future when I create more projects of my own in my own time or in the workforce.

5. Summary

We feel like we were overall very successful with our final project relative to what we first imagined. Initially we were very unsure with what we wanted in our finished game, but knew we just wanted a basic game where you would traverse a dungeon fighting monsters. In the end, that’s exactly what we have. We have 4 levels, even with a couple bosses, compared to our initial thoughts of just a couple floors and no bosses, and have monsters, combat, and even items to use. There are a few things that we would’ve done differently had we had the extra time, like implementing different special abilities for the monsters, more levels, better balancing, and just an overall more polished end game, but we are happy with what we accomplished with the time permitted. Going forward after this, some team members plan on implementing more into the game to hopefully create a longer and more fulfilling game, including the things that we had to leave out for time.