Tylah Molloy

Kaweb

Software Development Technician Project b – maze game

Contents

[Design 2](#_Toc7436340)

[Introduction 2](#_Toc7436341)

[UI Design 2](#_Toc7436342)

[Classes 3](#_Toc7436343)

[Configuration 3](#_Toc7436344)

[Maze Object 3](#_Toc7436345)

[Room Object 3](#_Toc7436346)

[Items Object 4](#_Toc7436347)

[Enemies Object 4](#_Toc7436348)

[Test 5](#_Toc7436349)

[Document 8](#_Toc7436350)

[Limitations & Improvements 9](#_Toc7436351)

[Command Line Interface 9](#_Toc7436352)

[Config 9](#_Toc7436353)

[Second text window 10](#_Toc7436354)

[ES6 JavaScript 10](#_Toc7436355)

[Attacking 10](#_Toc7436356)

[User guide 11](#_Toc7436357)

# Design

## Introduction

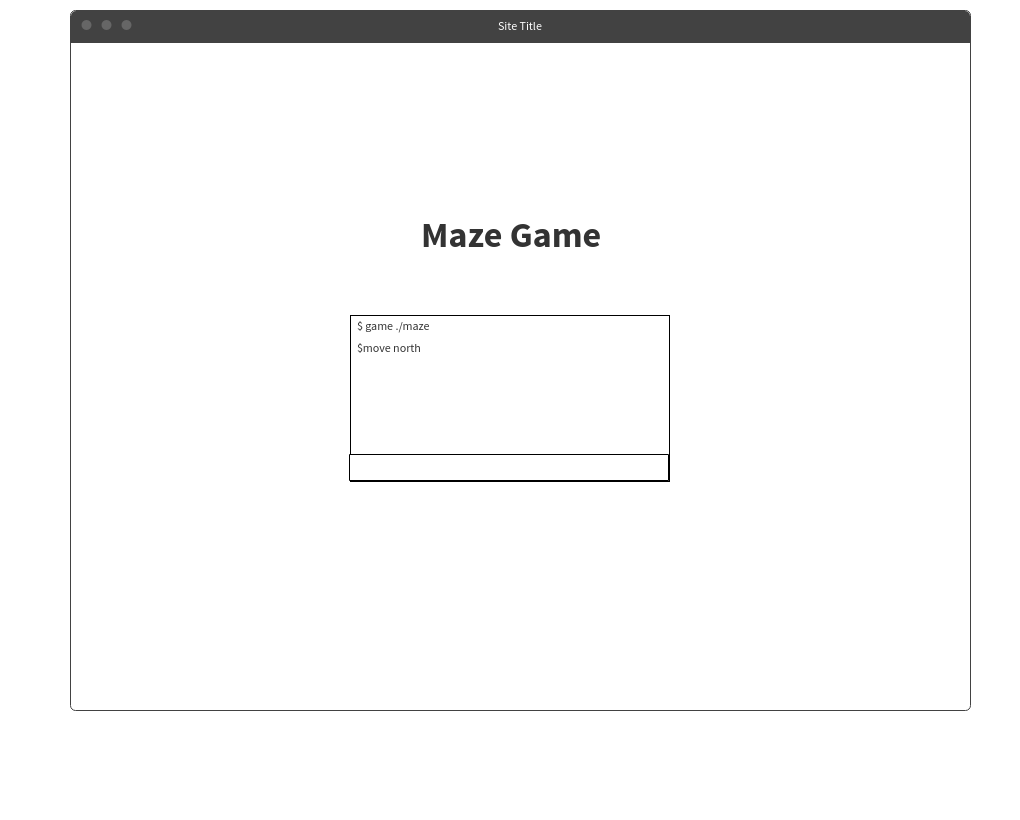
I am going to design the game in JavaScript and I’m going to use a library called three.js to help create graphics. Since I am using JavaScript the game will be playable via a web browser

I plan to get the game functional without graphics initially and make it fully playable via command line in browser.

## UI Design

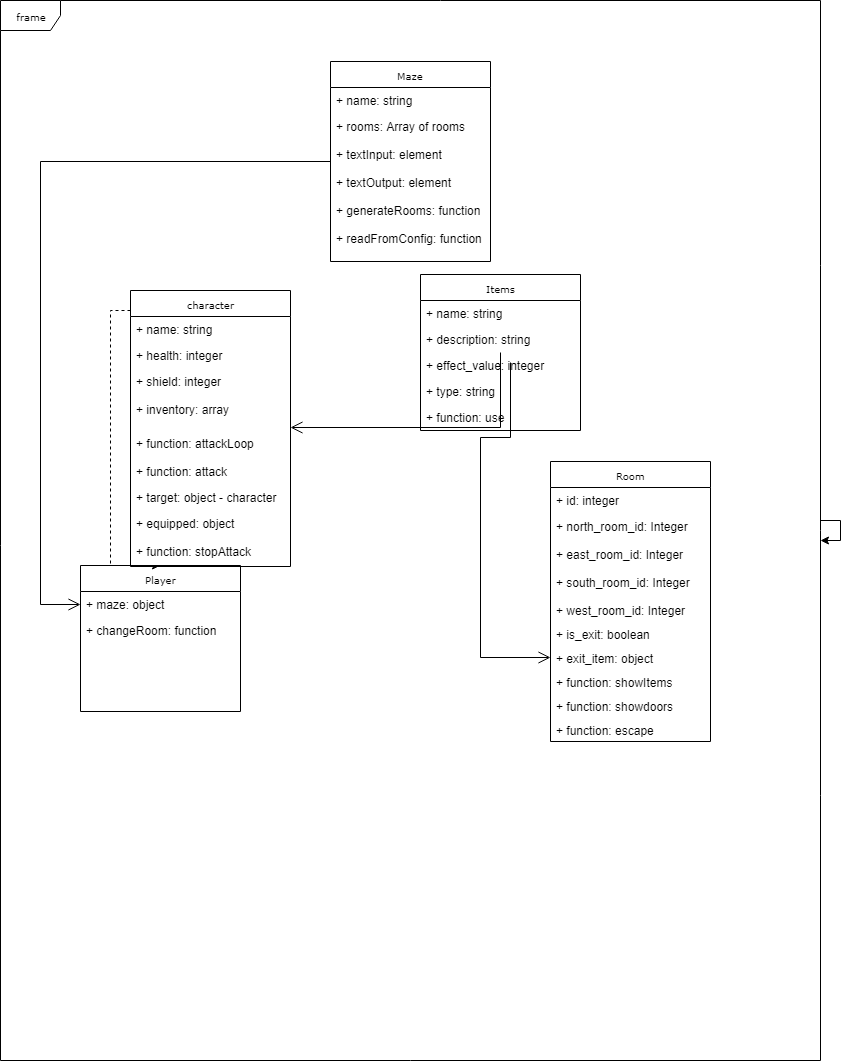
Since the game would be playable through a website I had to simulate a command line environment. I went for a minimalistic approach with the page simple being a scrollable div element where the commands and their output would be displayed as p elements.

Underneath the div element is an input field which will allow the user to enter commands to control the game. The element will also be designed to remove borders and have a parent div which wraps the text output and input to make it appear as one element like a real terminal.



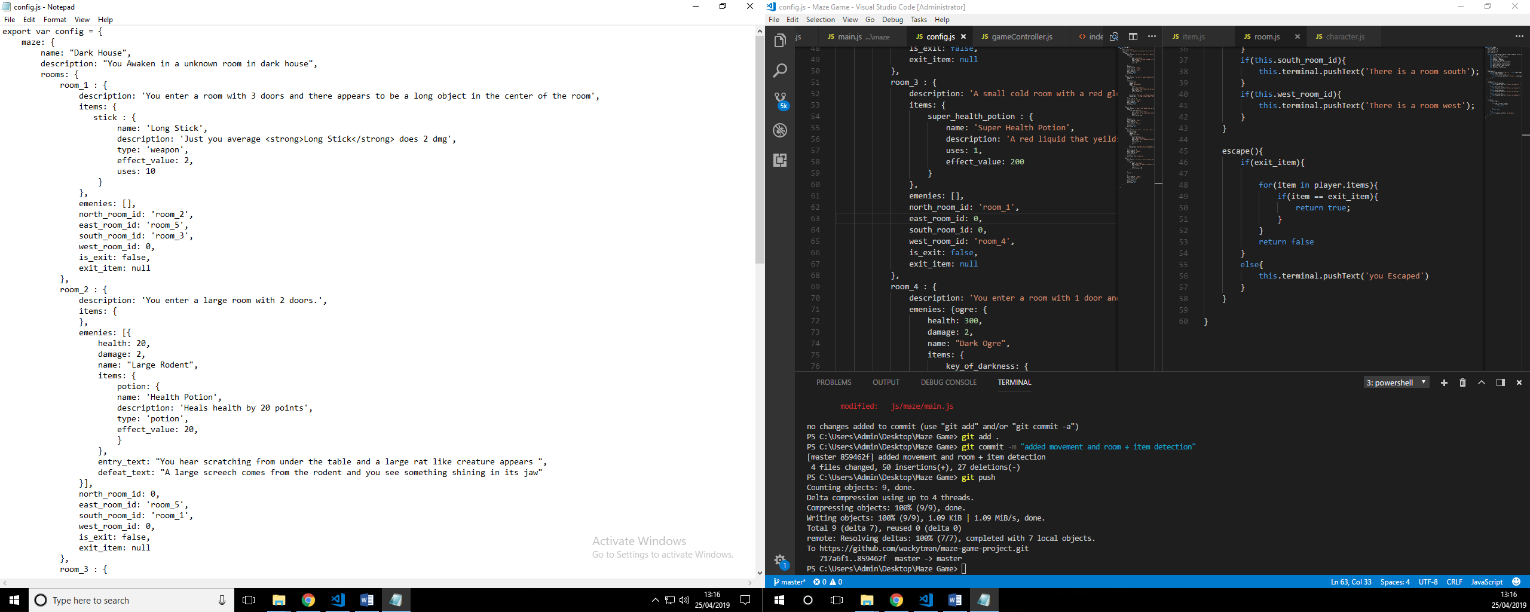
Within the text box I also play to show a user guide. The textbox will show text to help a new user get used to a command line interface.

## Classes



Here is a Class diagram that shows how each class is structured. The player class is an extension of the character class and allows you to control movement and interact with the maze object. This how I plan to structure my game.

## Configuration

The game configuration has been written as a javascript object. The reason I have done this since the code is very readable and friendly to a person who does not have much knowledge in javascript or programming itself.

The javscript object is imported into the script and is separate from the main code and when imported is read by the code as a normal object.

### Maze Object

The structure is set that a maze object which contains a name and description for the creator to recognise. The main game is then built from a object of multiple room configs. The reason I used a object is so the game can recognise a room via key instead of an index.

### Room Object

The room object is built of 9 properties:-

Description – The description is used as the text that is displayed to the player as they enter the room.

Items – The items object is similar to the maps room object which lets the writer to define a list of items within the room and what properties they have.

Emenies – The Emenies object again similar to the items object lets the writer define a list of emenies within the room and what properties they have.

[direction]\* Room ID – The key value of the room that the player can travel to by calling the ‘move [direction]\* command. If set to 0 then it is not possible for the player to move in that direction

\*direction being the north, east, south or west.

Is\_exit – Tells the player that this is the room in which they can escape the room.

exit\_item – The item that is required to exit the room. If set to false the player does not need an item to exit.

Wealth\_required – How much wealth the player requires to exit the level

### Items Object

The Item object is built of 5 properties:-

Name – The user friendly name of the item

Description – The items description to explain its use to the user

Type – The type of item it is can be potion, weapon, key or currency,

Effect Value – The integer value of the items strength or value.

Uses – The amount of times the item can be used.

### Enemies Object

The enemies object is built of 6 properties :-

Health – The integer amount of enemies health

Damage – the integer value of damage done to the player

Name – The user friendly name of the enemy

Items – An items object that the enemy carries that the players can acquire

Entry Text – The text that is displayed when the player enter the room that the enemy resides in

Defeat Text – The text that is displayed when the player defeats the enemy

The configuration reader imports the JavaScript object and creates instances of the corresponding class through the data defined. All these objects reside within the class of maze which the player will have an instance to within the code. Function written in the player class we be able to interact with all other objects through the maze class.

# Test

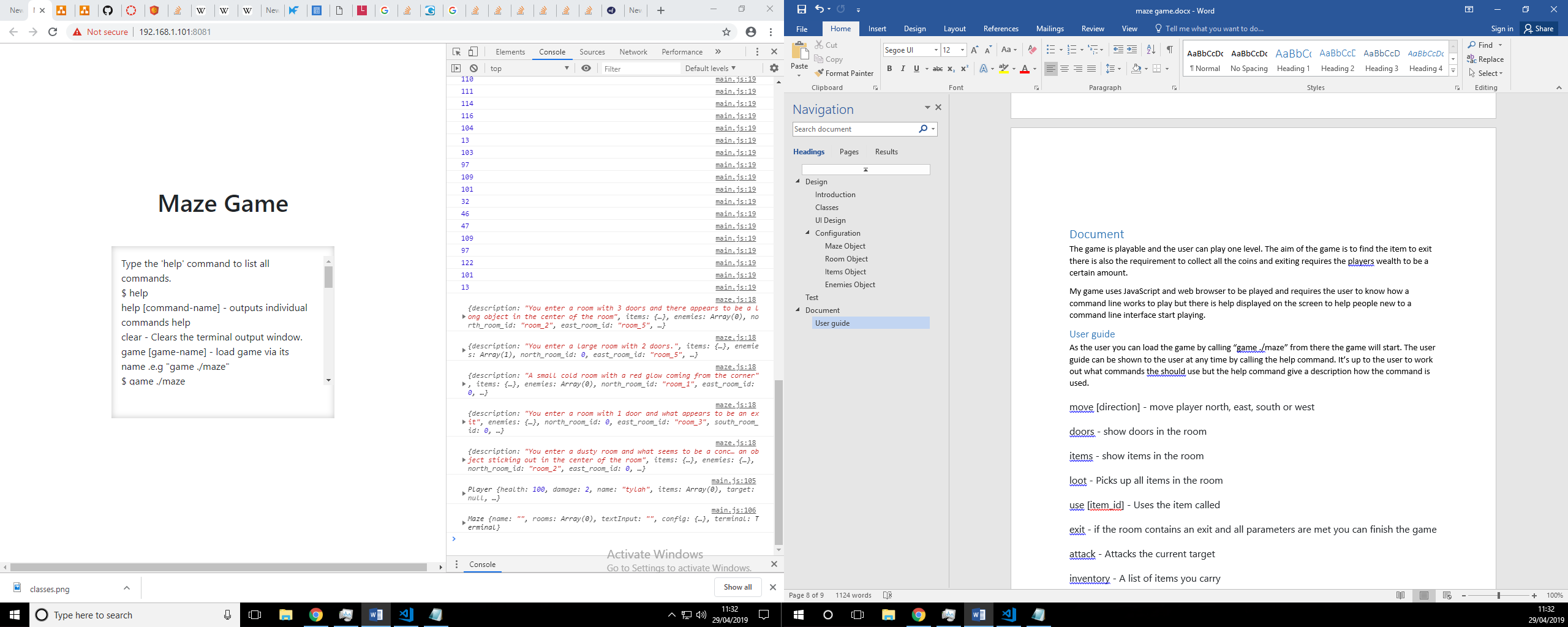
|  |  |  |  |
| --- | --- | --- | --- |
| Test | Expected Outcome | Actually Outcome | Comment |
| Does the “game ./maze” command boot game? | The game will boot and text will alert the user that game is booting. | The user is alerted the game is booting but it is outputted twice | To fix the output showing twice I removed the call from |
|  | | | |
| Does the “doors” command show the available passages? | The game will output what directions the player can move | The correct output is displayed | To improve I could possible output all exits and show which is passable and which is not |
|  | | | |
| Does the “Move [direction]” allow you to move into another room? | The game will change the player’s current room to the rooms that exist in the direction the player chooses to move. | The room entry description is displayed alerting the player they have entered a new room. | I could shorthand this command to up down left or right. |
|  | | | |
| Does entering a room trigger the enemy to attack? | The enemy attacks the player and cause damage and is displayed via the output window | The player is attack and its health is reduced |  |
|  | | | |
| Can you attack the enemy by calling the “attack” command? | The player should be able to attack the target and damage report should be outputted? | The damage report is outputted and the player can see how much damage they have done to what target | The user can see what damage they have done but can be confusing since all out put of enemy preparing to attack is shown also the user can copy and paste the command to attack very quickly. |
|  | | | |
| Can Be killed by the enemy? | Over time if the player is continuously attacked the health reaches 0 and the game should exit | The player is killed the text you have been defeated and the game beings its exit process |  |
|  | | | |
| Can the player loot items in the room? | The player should be able to pick up all the items in the room and displayed what items have been picked up | The items are picked up into inventory  And the output shows what items was picked up | The key is not specifically clear when inventory is outputted which may make it difficult to use items |
|  | | | |
| Are you able to use an items? | Items should be called by using “use [item\_id]” and the effects should be outputted to user | The item is used and the damage the player can emit has been increased |  |
|  | | | |
| Can the player exit the map and complete the game? | The game exit should be launched and you should be made aware you have escaped | The game is exited and the text “you Escaped” is shown |  |
|  | | | |

# Document

The game is playable and the user can play one level. The aim of the game is to find the item to exit there is also the requirement to collect all the coins and exiting requires the players wealth to be a certain amount.

The game requires a http server to be playable since it uses some of the latest ES6 JavaScript features like importing functions from scripts.

My game uses JavaScript and the web browser Chrome to be played and requires the user to know how a command line works to play but there is help displayed on the screen to help people new to a command line interface start playing.

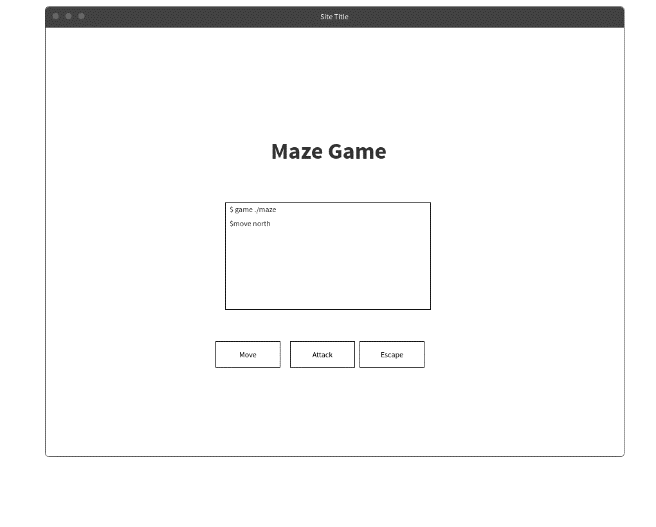


The game provides a retro feel since it utilises a command line. Being a 5 room map still is a challenge

## Limitations & Improvements

### Command Line Interface

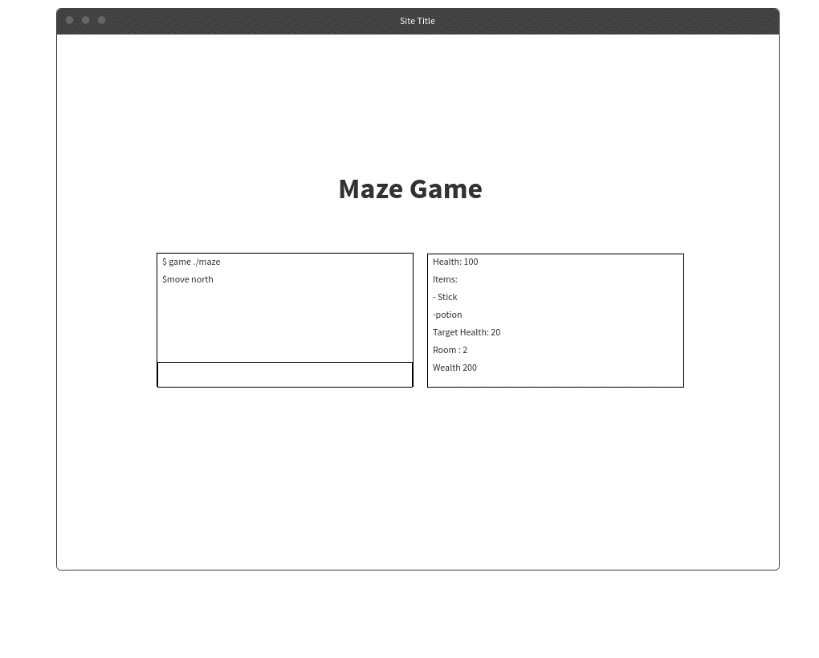
My current implementation of the maze game has a few limitations. Firstly the game requires the user to have some basic knowledge of a command line interface. At first glance they can be very confusing, which is not very user friendly. To improve this I could have provided a graphical user interface of the game with buttons.

Here is a brief wireframe how I could of implemented a Graphical User Interface. This could help the game look more appealing to other users who are not familiar with a command line interface. Behind the scenes the command would not need to change much either. The buttons can be hooked up to the same functions that the commands use.

### Config

The config allows you to define all the rooms and items in the map. You have to manually type the configuration and it can be a long task to write since you have to define every room, item, and enemy. To improve this I could potentially have a script that would randomly generate some rooms and allow you to tweak some parts. It will speed up the task of writing a config and I could implement it in way that the player can play randomly generated mazes.

### Second text window

An improvement I could make to the game is providing a second text window. The second text window could be used to constantly show player statistics such as health, inventory, target health, current room and wealth.

The wireframe example shows how I could implement the second text area. You can see that the player would not need to use the inventory command or wealth command.

I also think it will help the user feel more engaged as they would be able to see the health or target health more dynamically.

### ES6 JavaScript

Since I am using some of the latest JavaScript feature it means the game will not work in every browser. This becomes a problem since there is still users using browser like Edge and Internet explorer. Since I am making this for a company that may have users using these browsers and will not be able to play the game. The will reduce the target audience so less people will be able to play the game

### Attacking

When the player attacks an enemy they do so by calling the “attack” command. The attack command can be exploited by copying and pasting it in to the input box. This can allow they player to kill an enemy very quickly.

To prevent the attack command from being exploited I can suggest an improvement to the combat system. I can use a randomly generated pin that is shown to the user that changes every time the player attacks. The pin can stop the player from copy pasting the command since the pin code will not match

## User guide

As the user you can load the game by calling “game ./maze” from there the game will start. The user guide can be shown to the user at any time by calling the help command. It’s up to the user to work out what commands the should use but the help command give a description how the command is used.

Here is a list of all the commands a user can use please a description on what they can do:

move [direction] - move player north, east, south or west

doors - show doors in the room

items - show items in the room

loot - Picks up all items in the room

use [item\_id] - Uses the item called

exit - if the room contains an exit and all parameters are met you can finish the game

attack - Attacks the current target

inventory - A list of items you carry

wealth - list players current wealth

game [game-name] - load game via its name. The user can also exit the game and restart by calling “game exit”

help [command-name] - outputs individual commands help