Tylah Molloy

Kaweb

Software Development Technician Project b – maze game

Contents

[Design 2](#_Toc6909778)

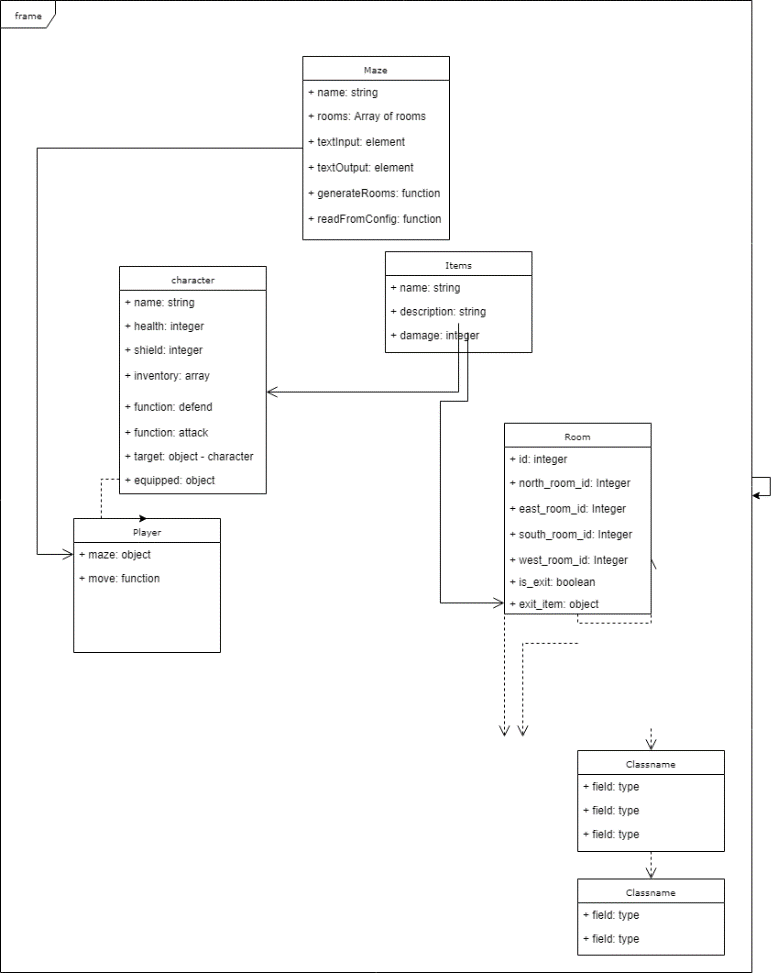
# Design

## 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. To get a command line working In a web browser I would need to simulate an event

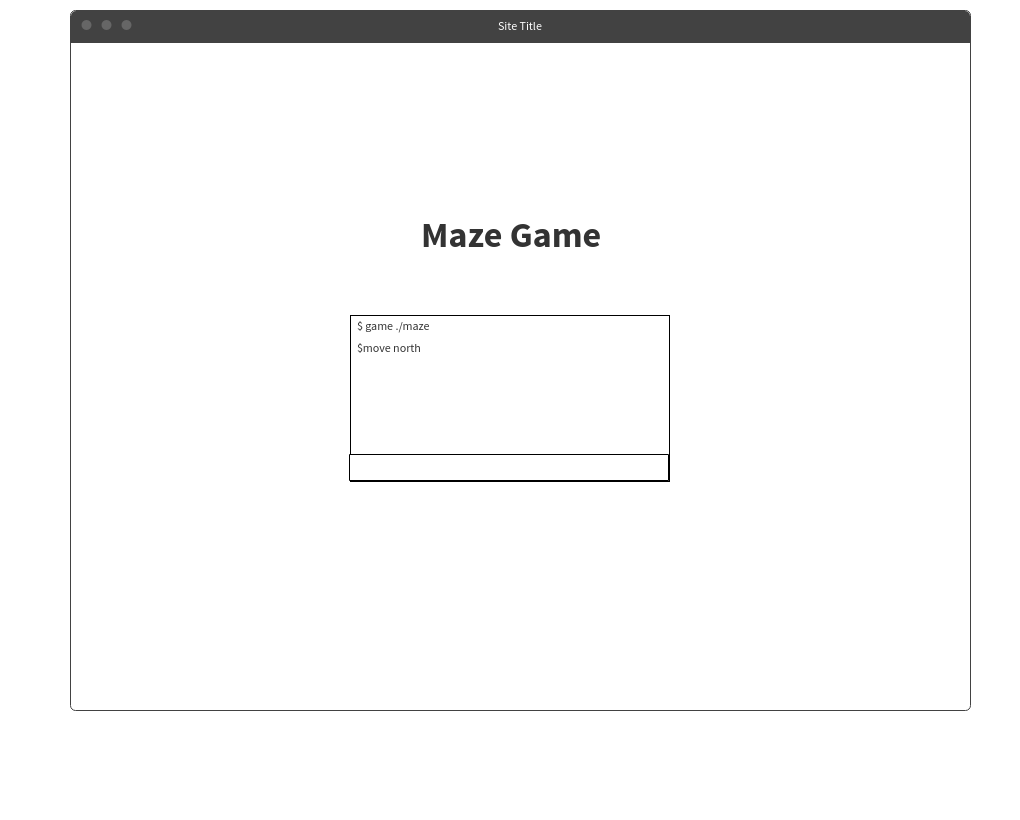
## Classes



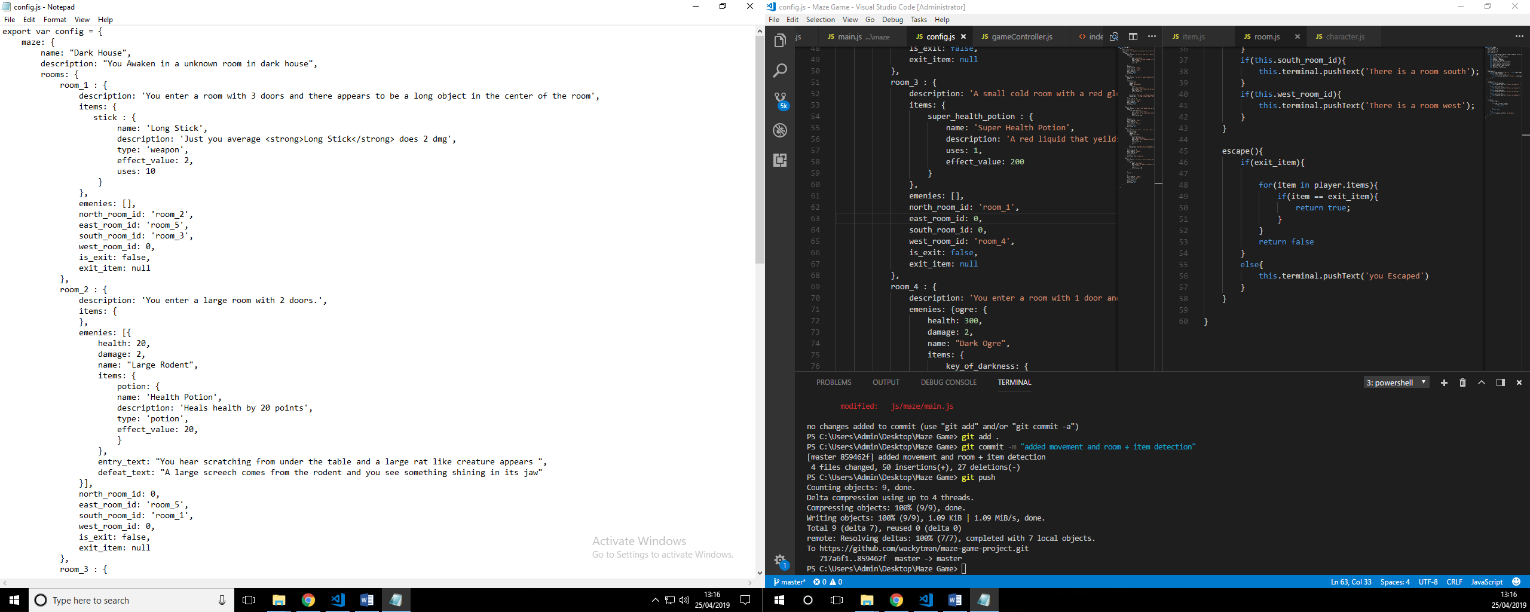
## UI Design

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.



## 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.

### 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.