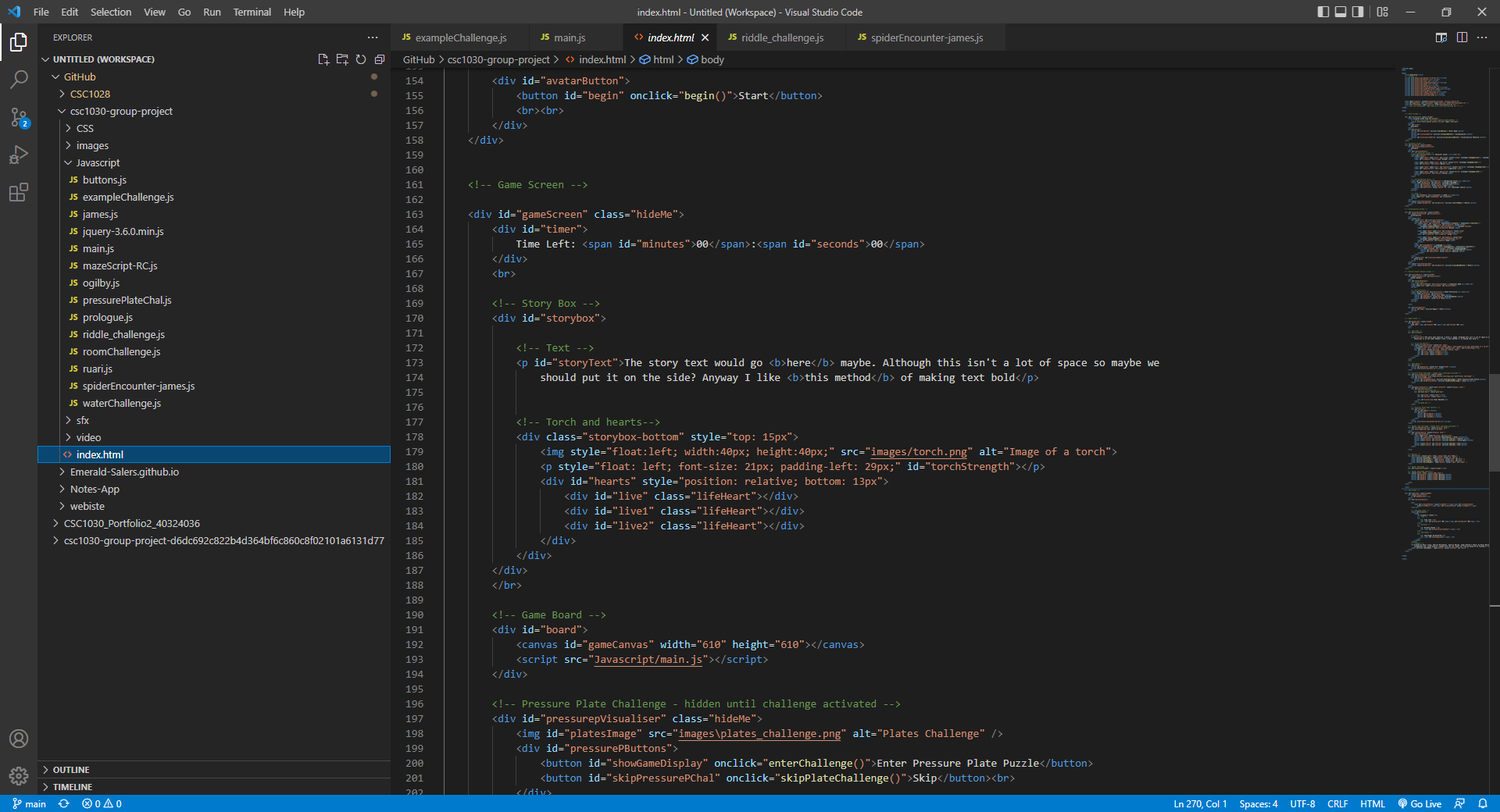
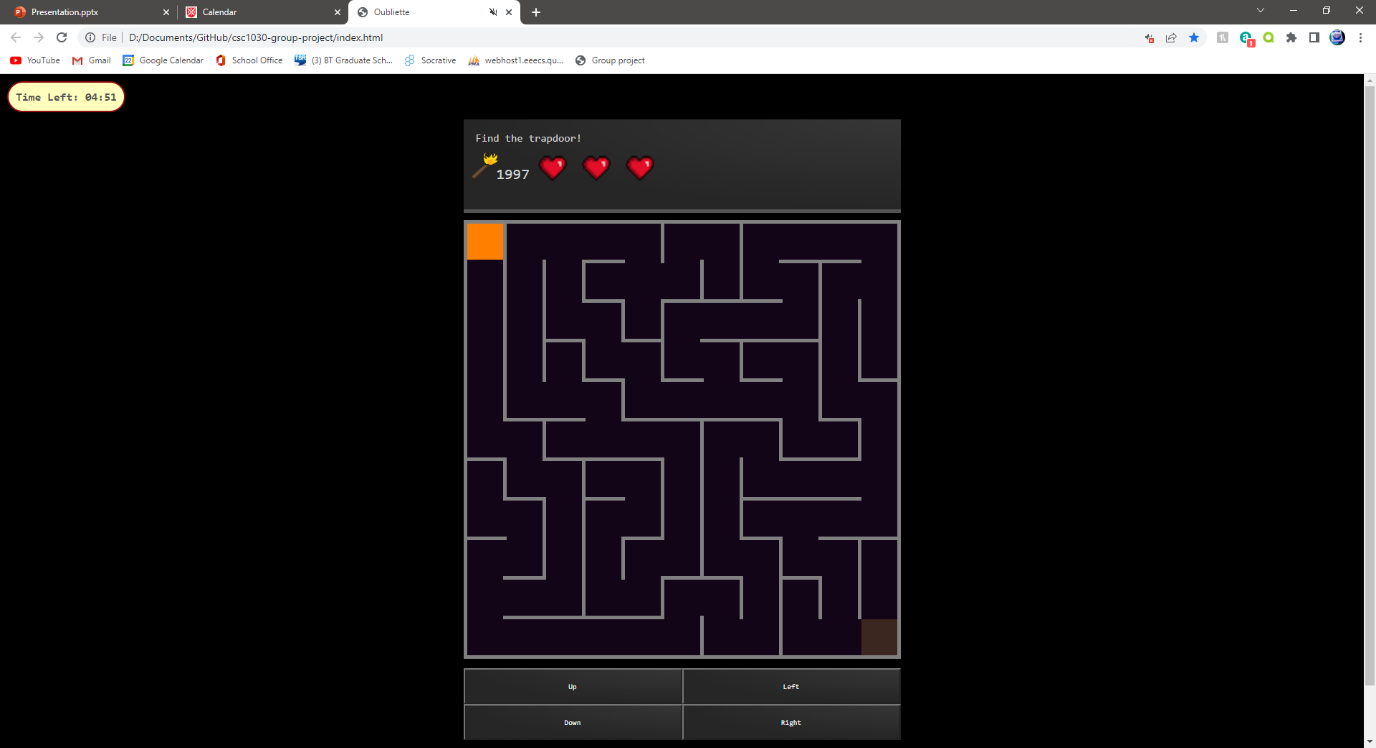
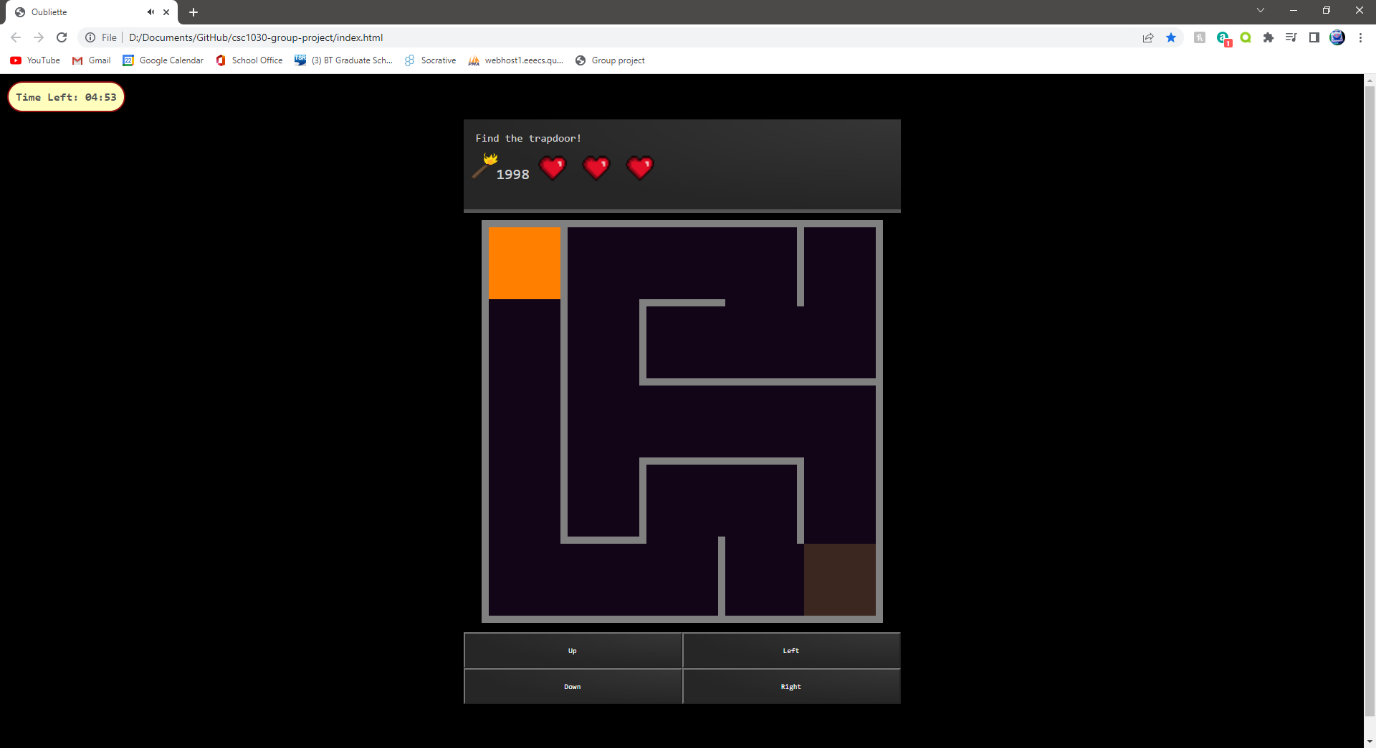
**Individual Summary Report**

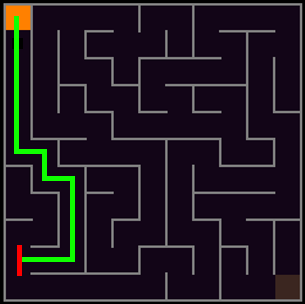


One of the main html elements I created for the game was the game board itself (a canvas which is later populated and edited by the javascript in order to display different images as appropriate throughout the duration of the game. As well as the hearts and torch ui at the bottom of the story box section (another element which is updated as the game goes along) which displays how many lives the user has left and the strength of the users torch at any given time.

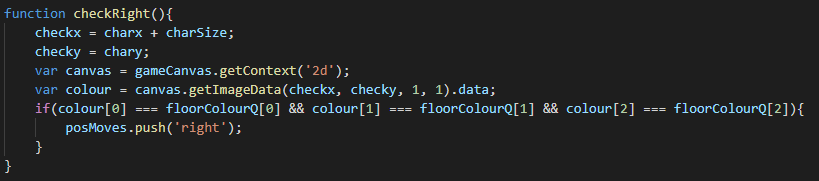


The main Javascript feature and also the most technically complex feature I implemented in the game was the random generation of a maze. I used a recursive back tracking function in order to implement this. It greatly enhances the quality of user experience as a new maze can be played each time the user loads up the game.

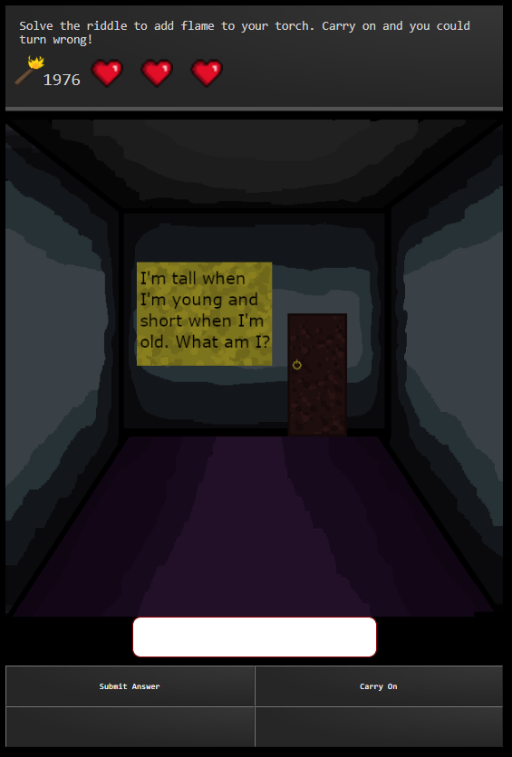
As part of this automatic maze generation I also implemented the ability to change the size of the maze and thus the difficulty. This is done by having all of the variables that draw and generate the maze being multiplied by a single variable. Thus linking all of the variables to the difficulty variable.



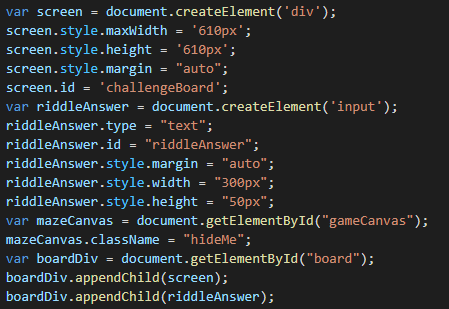
Another element which I implemented is the automatic movement. When a direction is chosen (in this case down) the character will continue to move automatically when there is only one choice other than the way you came from. Once more than one choice becomes available (indicated by the red lines) the character stops and awaits your input. This greatly reduces the number of times the user has to press a button on the screen. Making the game much less effort and hence much more enjoyable to play.



In order to do this there is a function for each direction which checks if there is a wall in that direction from the user. Each of these checks are called and if exactly two are returned as true (the way you came from and the way you are travelling) then the character will continue to travel forward. If any number other than two return as true (the character is at a junction or a dead end) then the character stops moving and awaits user input.

One of the challenges which I created was the challenge in which the user is asked a riddle. I created a lot of the art for most of the challenges including this one which I was happy with. I chose a riddle which fits the theme of the game well as the answer is a candle.

This challenge really empthasises the text adventure element of the game. The challenge is described concisely and the user is left to input any text they like to see if it matches the correct answer. If they cannot get the answer and feel they are wasting too much time on the riddle they can skip it, however there are consequences. Such as torch strength decreasing, making it much harder for them to escape the maze.



In order to create my challenges I decided to create a new div within the javascript rather than have one pre existing in the html. I assigned the div an id which allowed me to edit it throughout the challenge as can be seen above. This kept the index file much cleaner and easier to read as it didn’t have several challenges in it as well as the body of the page.

After each of my challenges are complete the challenge automatically removes the div from the page in order to prevent it from disrupting anything else on the page. This makes this method of writing the challenge easier when working in a group as it makes the code for each challenge much easier to remain self contained within it’s own file.