# Individual Report – Adam Logan

## Resizing Screen Elements

I have created two scripts which are called ‘stopAtFooter.js’ and ‘textBoxSizing.js’ and these scripts are used on all my web pages. These scripts allow the user to resize the web page with the user interface still being friendly. The ‘stopAtFooter.js’ script changes the ‘position’ property of the footer to prevent a bug in which the footer would not always be completely at the bottom of the web page at all times. This bug is described in more detail in a comment within the script and in my week 9 log. The second script ‘textBoxSizing.js’ just prevents the text box to appear over the main game. On both the colour pad and the basement this just stops the game from moving any further but on the fuse box it makes the text box disappear.

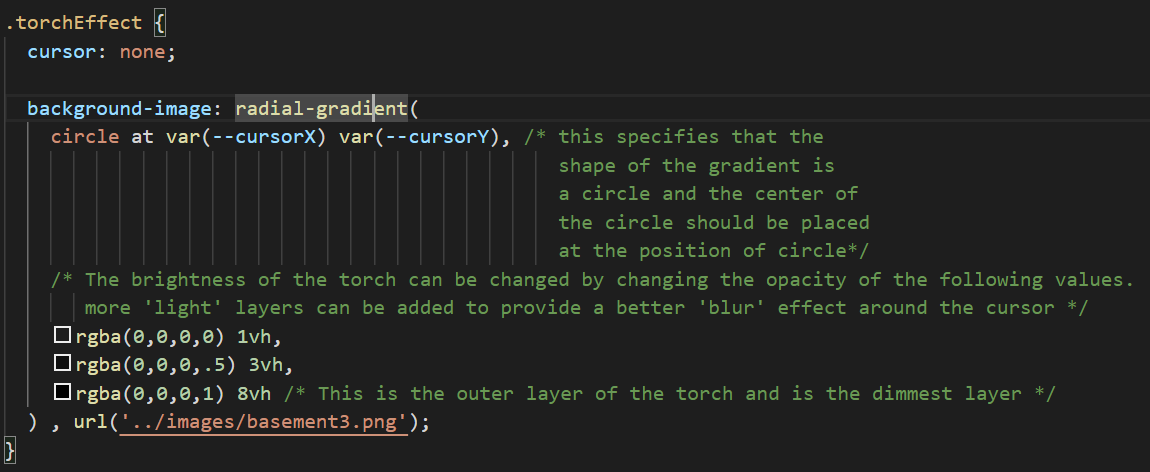
## Basement Web Page

Within the basement web page I have fully developed the torch effect, the positioning of both the coat button and the fuse box button, the individual basement timer (not to be confused with the group timer which was handled by different team members) and the switching on/off the lights within the basement using session storage. The basement timer counts the time in which the user spends on the basement web page and the fuse box web page and is displayed to the user at the end of the game within the statistics page.

In relation to the resizing of this web page I have created it where the image of the basement will stay a consistent aspect ratio and therefore the two buttons will always be over the same spot within the image. I also created the basement image by merging three images (basement image, coat stand image and the fuse box image) together.

The scripts that I developed for this web page are generalLayout.css, basement.css, torchEffect.js, basementScript.js, textBoxSizing.js and stopAtFooter.js.

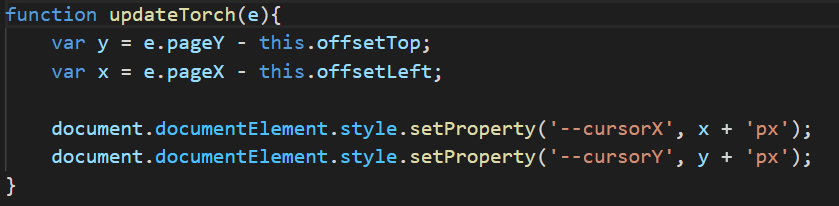
### Torch Effect CSS



This is code is only specific for the basement and is simply taken out for the fuse box.

### Torch Effect JavaScript

This is called every time the user has their mouse over the torch area:



These are the CSS custom properties that store the position of the cursor.

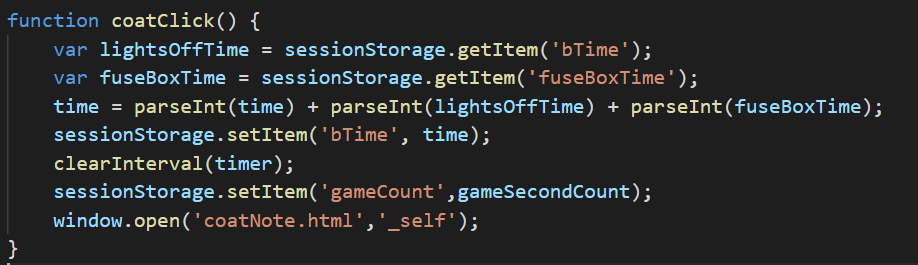
### Keeping Aspect Ratio of Basement Image

This is the range of the aspect ratio that is acceptable and that keeps the coat and fuse box buttons in the same place. The reason for the range is the image does flicker when the user has the aspect ratio just on the edge.



The code which adds the time spent in the basement (including the fuse box)

### Total Basement Time Code

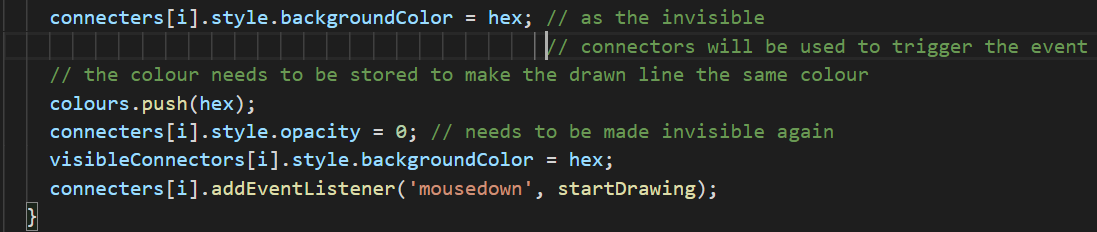
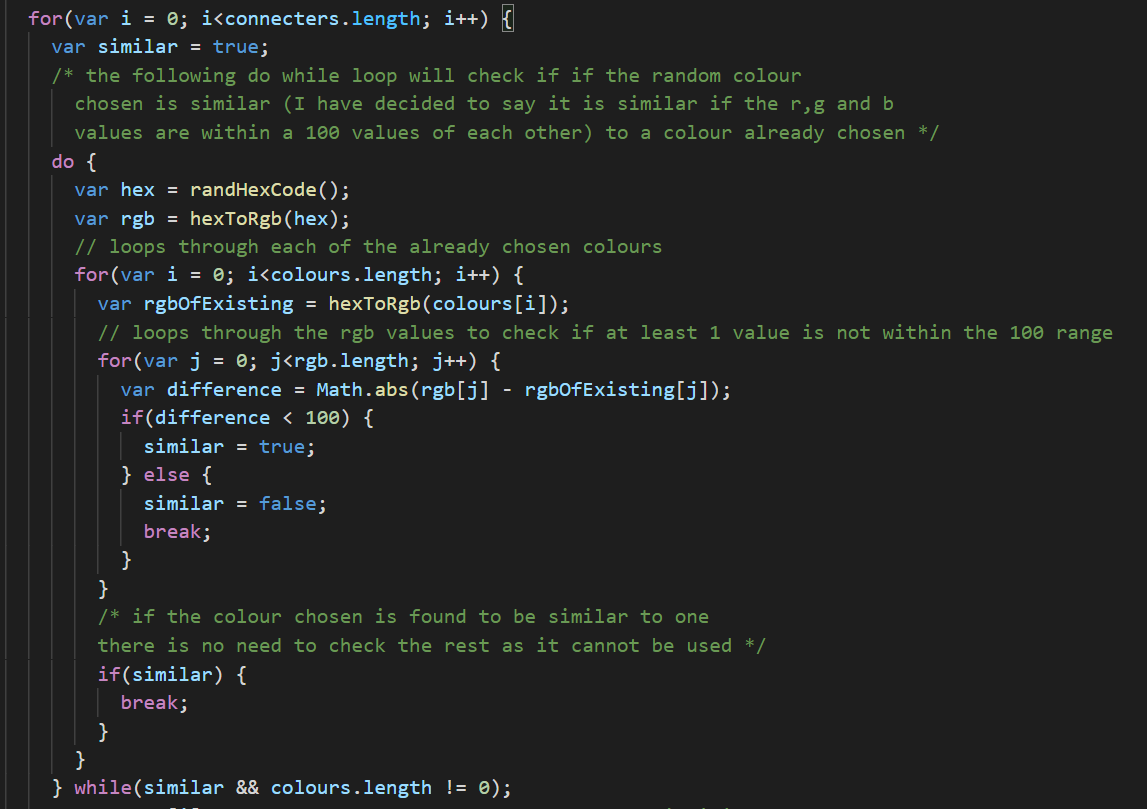


## Fuse Box Web Page

I also fully developed the wire game including the audio that is played when the wires are connected. The wire game is played by dragging from the connectors at the top to the corresponding coloured connector at the bottom. When the user lets go of the wire the wire will disappear and the user will need to start dragging the wire from the beginning. I also implemented the torch effect within this web page. I also fully developed the random colour generator for this web page and developed the code to make sure that the colours of the wires are not similar.

The scripts that I developed for this web page are generalLayout.css, fuseBoxStyle.css, torchEffect.js, fuseBoxScript.js, rgbAndHexFunc.js, textBoxSizing.js and stopAtFooter.js.

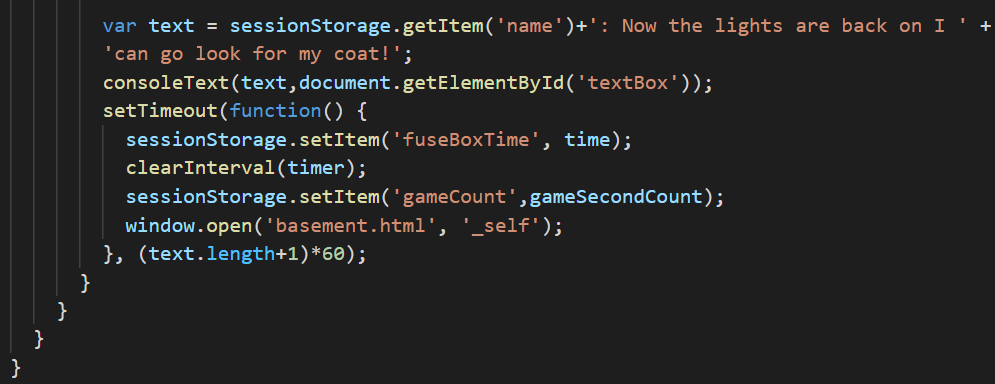
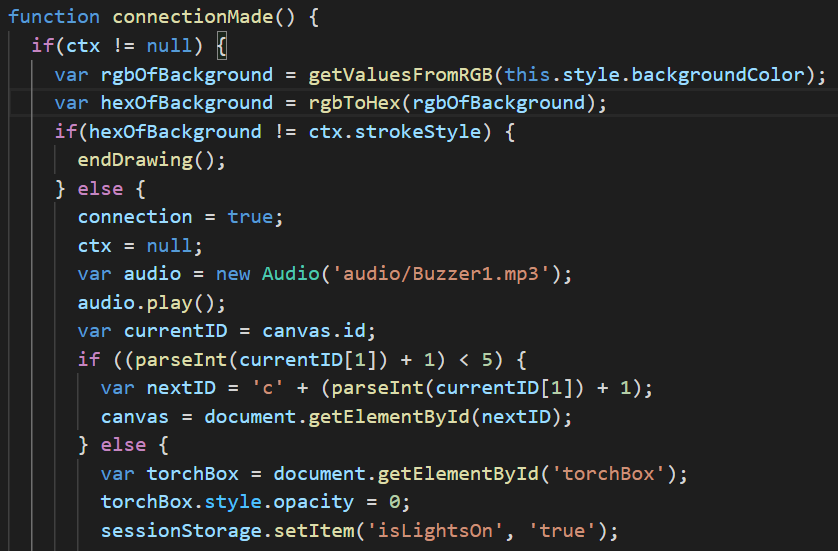
### The code to check if a similar colour has been used



This loop checks the red, green, and blue values of the chosen colour to see if it has a difference of a 100 with the colours that have already been selected.

This continuously generates a new random colour until it generates a colour which is not similar to any previous colour.

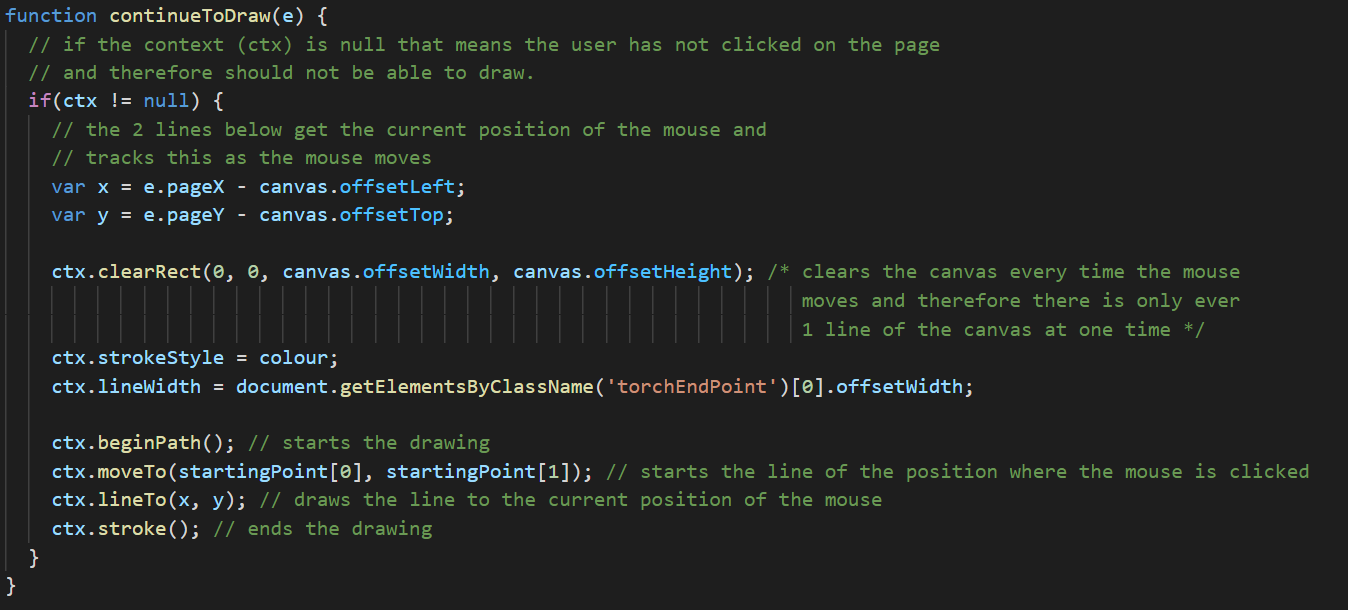
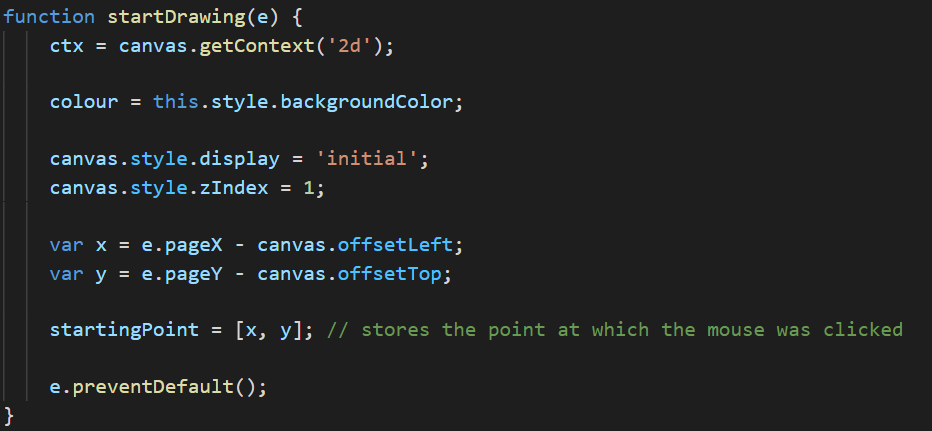
### Code to Check for Connection



To check if all the wires have connected I checked the ID of the previous canvas and if it ends in a value less than 5 then connections still need to be made. This happens as the ID of the canvas is names as ‘c[number of connections made]’.

This is the code to play the play the sound effect.

### The Code to Draw the Wires

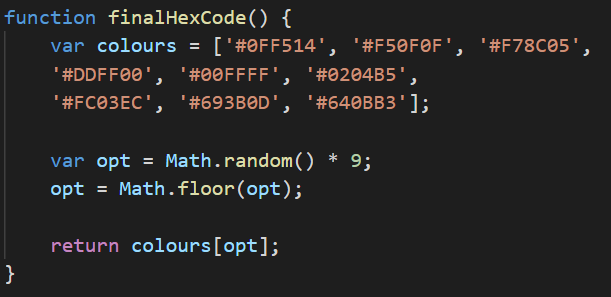


## Colour Pad

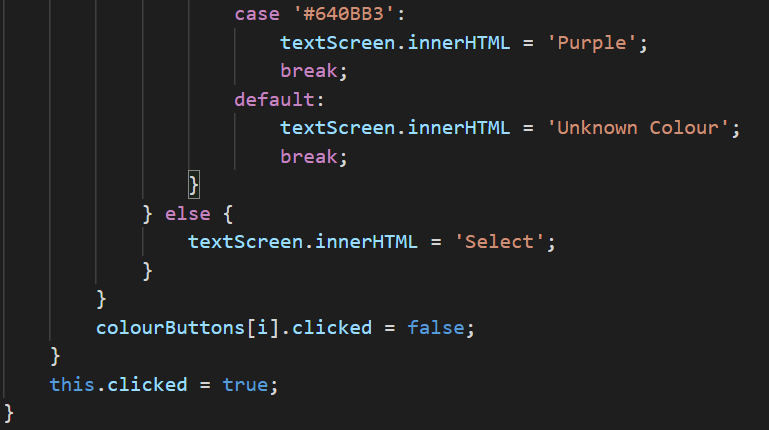
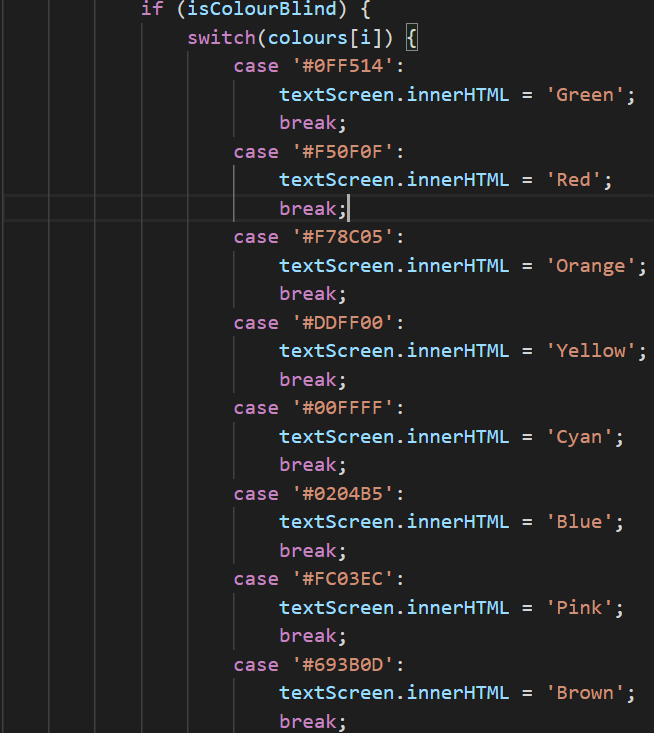
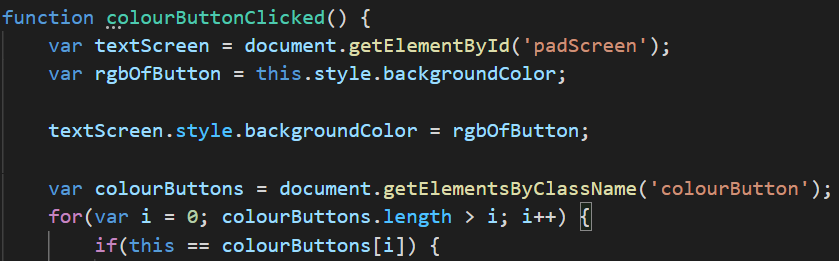
For the colour pad I fully developed the accessibility option within this web page which is the colour-blind option. I also developed the colour pad itself, the individual timer for this page and the function to randomly pick one of these colours which is passed through the web pages that require the hex code.

The scripts that I developed for this web page are generalLayout.css, colourPadStyle.css, colourPadScript.js, rgbAndHexFunc.js, textBoxSizing.js and stopAtFooter.js.

### Generating Final Answer Code



### Colour Blind Code



### Colour Pad Screen Changing



This green button has been pressed and the colour pad screen has changed to the same colour as the button.

### Colour Blind Button



The colour blind button has been pressed and then the green button has been pressed. As you can see the colour pad screen has turned green and the message on the colour pad screen has changed to ‘GREEN’ to tell the user the colour that they have selected.

## Success End Screens

I fully developed both the success end screens (not including the escape video) and I implemented the code to decide which success screen to display using session storage which can be seen below.

### The code which decides the end screen

