Ruari Craig – Individual Summary Report – Group 30

One of the main components of our game I developed was the ‘Game Landing’ section – Start Screen, Accessibility Screen, and the mandatory customisation screen (Character Details Screen). For the ‘Game Landing’ section, I was the main person responsible for writing the HTML, JavaScript, and CSS. The only section of the game landing section I didn’t write was the Customisation screen as James had some good ideas for it.

A screenshot of a computer

Description automatically generated with medium confidenceA picture containing text, candle

Description automatically generatedFor the start screen the most important thing for me was getting a good-looking background that fit the theme of the game. I found the flickering candle video online and thought it was perfect for the game as it directly related to a main trait of our game – the torch. I thought that addition as well as the minimalist start screen would help increase the users experience by increasing immersion.

Figure : Start Screen - displaying candle background video and associated buttons

Figure : Start Screen - HTML code, I wrote corresponding CSS to make this look like Figure 1

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedThe Accessibility Screen has 2 accessibility features: change text size and language. In choosing what features to include I researched multiple mainstream games and what accessibility features they use and the most common were: text size, subtitles, colour blindness and language. I made the decision to include text size (1x size, 1.5 size, 2x size) for those that are hindered visually, and I chose language (English, Spanish) because it’s ignorant to assume all users who play our game are English. I chose not to include colour blindness because in my research I found that the 3 main types of colour blindness (red blind/green blind) wouldn’t affect the users experience of the game as no matter what blindness they have the colours between the player, floor, walls, end goal should still be distinctive. I chose not to use subtitles because it wasn’t necessary due to no speech in the game. The accessibility JavaScript implementation was an interesting task to write because to my knowledge there was no real way to scale up all font CSS. So, my work around it was to create 2 new stylesheets that could be swapped to on an onchange event and individually scale up their font sizes by 1.5 or 2 times. Disclaimer: for the language feature nobody on our team spoke another language so we used google translate so the Spanish may not be 100% accurate.

Figure 4: Accessibility Screen - Language selected as Spanish so now the whole game is playable for those that read Spanish

Figure 3: Accessibility Screen - Large/1.5x multiplier selected for text size of the game

Text

Description automatically generatedText

Description automatically generated

Figure 6: Accessibility Screen - stylesheetHuge is active stylesheet when Huge is selected and stylesheetLarge is active stylesheet when Large is selected

Figure 5: Accessibility Screen - CSS I wrote to make radio form elements look like how they do in Figure 3

For the main game section, I was involved in the development of creating the 5 minute timer for our game, creating a decreasing torch function for medium and hard versions of our game and creating 2 challenges – ‘room’ and ‘pressure-plate’ challenges. I wrote all the JS, CSS and HTML for them, the only assistance I got for these was de-bugging and Charlie designed the images for visual aids for the challenge.

Text

Description automatically generatedGraphical user interface, application

Description automatically generatedI developed the room challenge in an attempt to hammer home the ‘text-adventure’ theme of our game. The challenge involved the user being trapped in a room and using the text and button aids the user inputs text to escape the game. My JavaScript for this challenge was very long-winded using lots of if statements to reflect the user’s choice. It wasn’t particularly technically complex as it mainly involved repetition of changing ‘innerHTML’ based on user input. I believe the challenge enhances the user experience as it adds a new environment to the game rather than always being in the maze.

Figure 8: Room Challenge - Text adventure elements of Room challenge

Figure 7: Room Challenge - JS to chnage the innerHTML of buttons if Part 1 of the room challenge is complete

My second challenge – the pressure plate challenge – is definitely one of the most technical aspects of the game. It involves the having to escape another maze within the maze game. Due to my limited knowledge of JavaScript, I didn’t know how to design a maze from scratch, so I followed a maze tutorial online. So, although my maze JavaScript file isn’t 100% from my own knowledge, I still believe it is one of the most technically complex segments due to the tutorial’s code complexity and the fact that I had to alter many parts of the code to make it fit the challenge/game theme I envisioned. I had to change the layout of the maze, size of the maze, add buttons, make the movements in the maze interact with the story dialogue and had to make the character reset Text

Description automatically generatedText

Description automatically generatedA screenshot of a game

Description automatically generated with low confidenceA screenshot of a game

Description automatically generated with low confidencewhenever it stepped on a ‘dummy’ pressure-plate.

Figure 12: Pressure Plate - updateToStart method updates the DOM to display the character back at the start when a dummy plate has been stepped on

Figure 11: Pressure Plate - version of challenge in actual game where you don't know which pressure plates are valid which are dummies

Figure 10: Pressure Plate - in-game challenge with wall tiles CSS as skyblue to show layout of maze

Figure 9: Pressure Plate - JS code showing 2d array filled in withs 0s and 1s to display layout of maze (0s floor, 1s walls)