Developer Documentation

The main.c file is where all our code has been developed. We have a content folder which holds all images and files that are included in our game.

The first 162 lines of the main.c file is mainly initialising all the variables that we use throughout the code. We have SDL variables that we had to initialise such as images and rectangles to contain those images for our game. All the images that we used are in the content folder next to the main.c file. We get these images by using the SDL library and create rectangles for them to be placed in before they can be displayed in the game.

The rest of the code is included in a while loop that uses a boolean to tell us whether the game is running or not. If ‘running’ is true then the game is running. Whilst the game is running we use boolean values to set which screen we want to display at that time in the game. For example, if ‘main_menu’ is true then we want to display the very first screen that the user sees, which is the main menu screen. Once the user presses the spacebar on the keyboard we set ‘main_menu’ to false as we are displaying another screen now. In each of these if statements, where we check to see which screen we are currently on, there is code that sets up all the features for that screen. These features include, waiting on a button press that will take you to a different screen (navigation), rectangles and images for objects displayed on the screen, and any animations that added to objects.

For the scorekeeping we use a text file, that we read and write to, which has scores already implemented that the user can try and beat. The text file has six lines with five lines being a name and a score value and the sixth line just has the value ‘x’. We use the ‘x’ value as a checkpoint to tell when the SDL library when to stop reading the file so that it can print out the five names and scores onto the leaderboard page. When the user beats a score on the leaderboard, their score gets added to the leaderboard in the appropriate place keeping just five scores. If the user does beat a score they can see their new score on the leaderboard screen (from the Game Over screen) with the name ‘You’. Also when you do beat a high score on the leaderboard it will show in the ‘Game Over’ screen that you have obtained a new high score with flashing text.

The game itself has two stages. Whilst playing the game the user will notice at the start that they have to avoid fireball looking objects or else they lose the game. Once 1000 points are reached, the goal is to avoid the walls that are moving in and out randomly. The stages alternate by 1000 points so once the user reaches 2000 points, they will go back to avoiding the fireball objects again and so on. The code for this can be found starting on line 610 where we are in the else statement meaning we no longer want any menu screens to be visible. In this code, you can also find the code for the opening animation here as well which displays instructions on the arrow keys to use and shows the player falling. The fireball’s coordinates are randomly generated so that we can’t predict where they will appear on the screen. The walls also work like this and change their direction based on a randomly generated value.