User guide

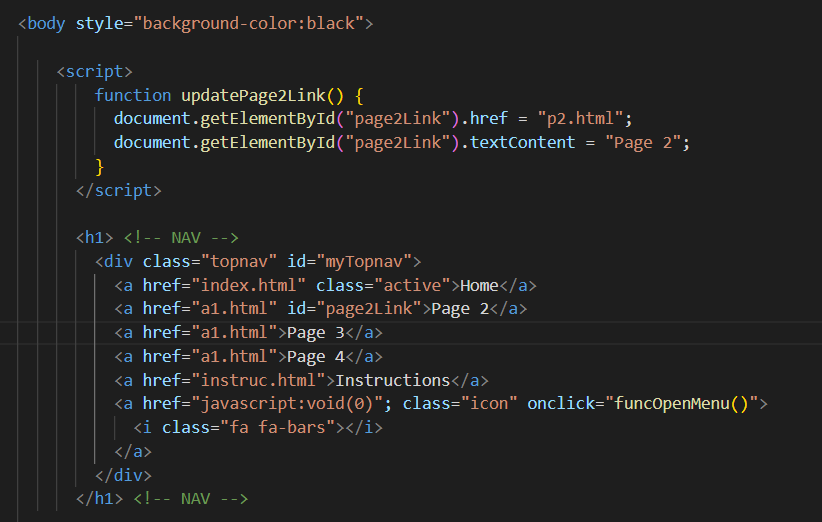
There are instructions/answers in the instructions tab of the website.

This project is a hidden game where you have to get through the different challenges to get to the end.

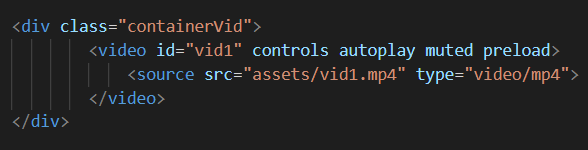
Index –



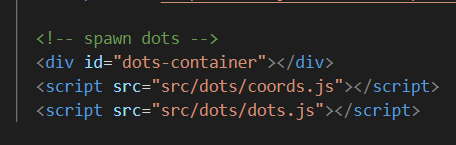
The head tag has all the links needed such, they are commented.



Here is the start of the body, function updatePage2Link is what locks and unlocks the page. The id for page 2 on the navigation is page2Link which is what gets changed. If you try to press page2 before completing page1 it will take you to a separate page that tells you to go back and try again. Once the game has been completed it will unlock page2.

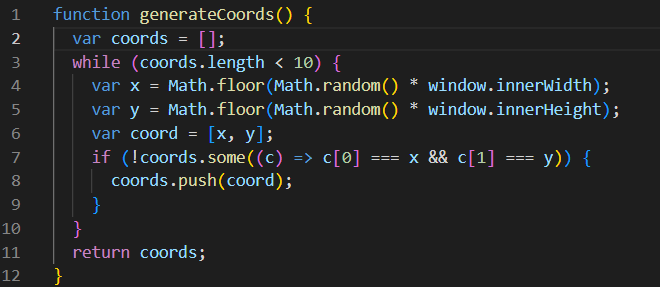


This plays a video, it has controls to play/pause the video, it automatically plays, it starts muted but can be unmuted, and it preloads which means that before the page loads up the video loads up first so there’s no bugs or lag.



This part of the code spawns the dots but its just calling the other JavaScript files

Coords.js –



This function generates an array of coordinates pairs an x and y.

Var cords creates an empty array, and the while loop makes 10x2 random values within the screen width and height. Math.floor is used to round the random numbers into full integers. The if statement makes sure that two coordinates are not the same.

Dots.js –



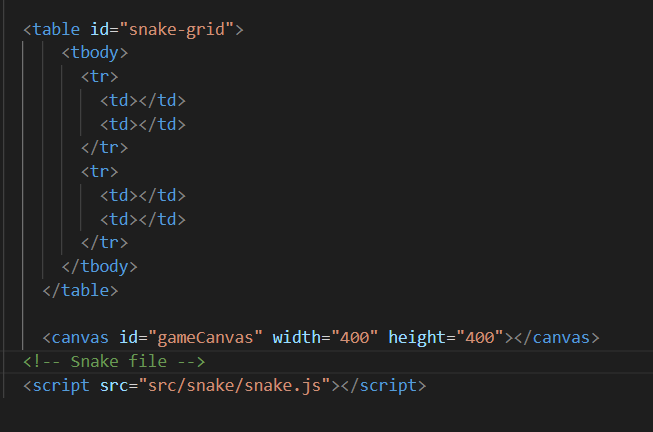
Line 2 creates a variable which calls the generateCoords function, line 5 is initialising the dot counter.

The for loop is used for using the generateCoords to place the dots on the page. A div element is created for each dot. Line 10 is so that dots can be styled in css.

The event listener looks for a click on the dot, then line 17 removes the dot. Line 18 adds 1 to the dot counter. The if statement on line 19 saying if 10 dots have been pressed then to change the updatePage2Link variable which unlocks page2.

Line 27 displays the dots on the page.

Page2.html –

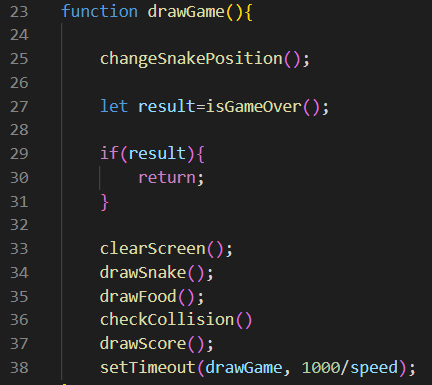


This snippet of code created a grid for the snake game, then a canvas is created for the snake game and the JavaScript file is called.

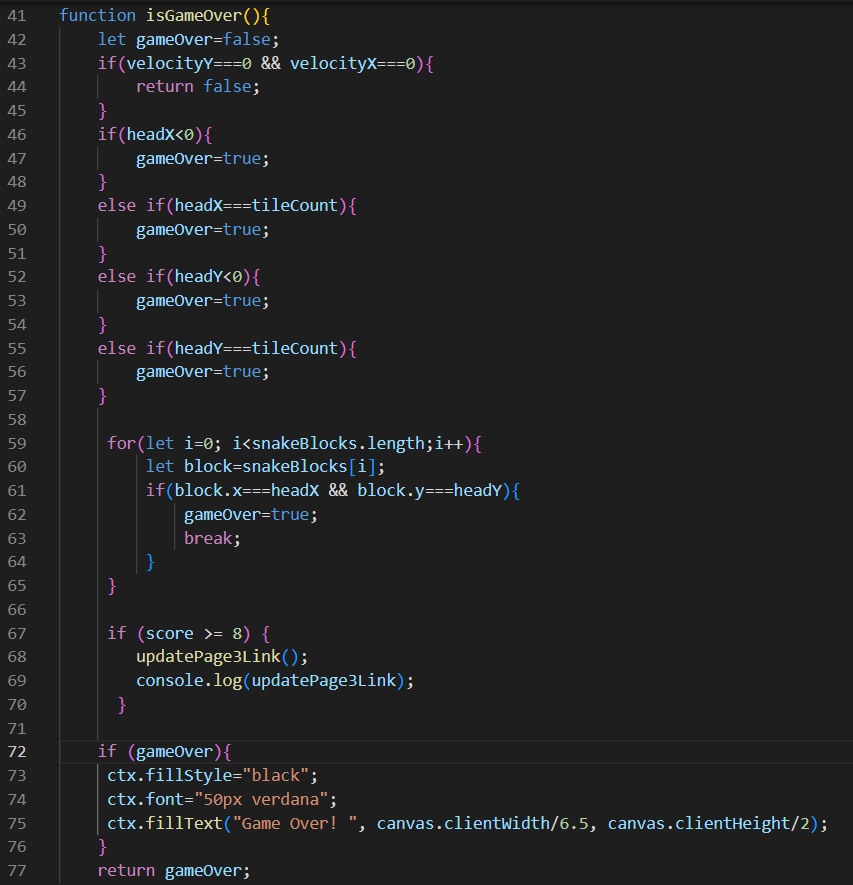
Snake.js –



Lines 1-21 sets up all the variables needed, this includes speed, food location, score, snake head, snake length, array of how many blocks the snake has.



This function is the main game loop, it has the game over logic and sets FPS to 7.



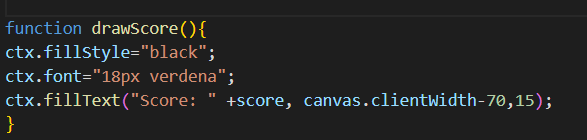
The isGameOver function has a Boolean value of whether or not the game is over.

Line 43 checks if the game has started, line 46 turns gameOver to true if the snake hits the left wall, line 49 if it hits the right wall, line 52 if it hits the top and line 55 if it hits the bottom.

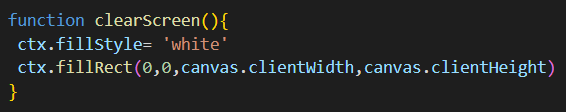
Line 59 check to see if the snake turns into itself.

Line 67 checks if score is 8 or more and if so, it updates the Page3 link to unlock the page.

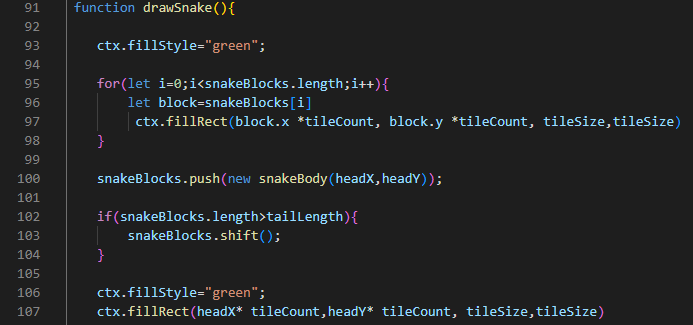
Line 72 is an if statement that says if the game is over fill the canvas in black and display Game Over in the middle of the canvas.



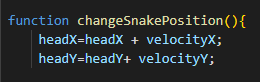
This function draws the score on the top right of the canvas



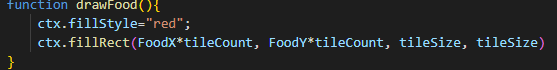
This function clears the screen.



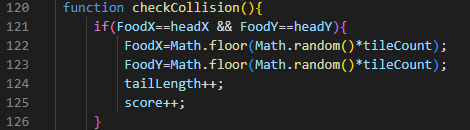
Line 95 draws the snake body. Line 100 adds a block to the snake body. Line 102 removes a block if the snakeBlocks is larger than tailLength.



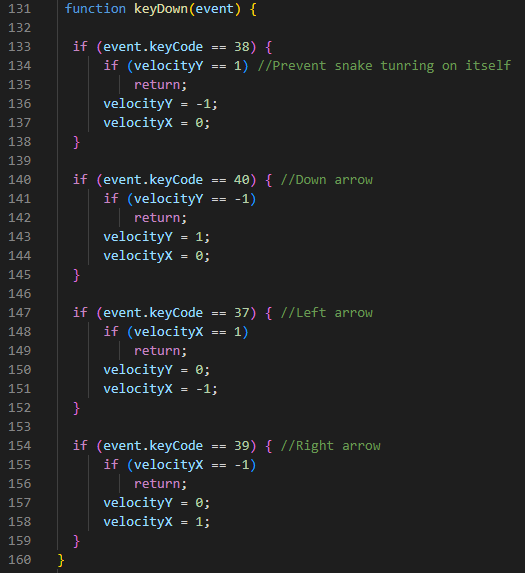
This function updates the snake head position based on velocity.



Draw food puts the food on the canvas, food x and y are multiplied by the tileCount to determine where they land on the canvas.

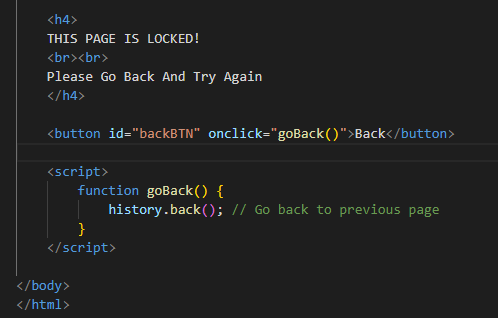


The CheckCollision function detects if the snakes head position intersects the foods position, then on line 122 and 123 its generating new coordinates for the new food block, line 124 adds a block to the tail and line 125 increases the score by 1.



This function handles user input with arrow keys and prevent the snake from turning on itself.

A1 –



This is the alternate page which shows the page is locked, it is red with a warning, it also has a go back button which takes the user to the last page they were on.