**Snake algorithm**

We will start by creating the canvas element in our html with dimension 900x460.

Let’s create a function with an instruction inside canvas when when load with prompt user name and show instruction

**function showInst()**

Create button to start the game, save the game, resume a game, and the leaderboard of the game with HTML.

Make a Settings JavaScript file in which, using the getElementById method, we will get the canvas. Make a variable to get the content of the canvas and create some global variables to start:  
**Javascript**

**var GAME\_SPEED = 100;**

**var CANVAS\_BORDER\_COLOUR = 'black';**

**var CANVAS\_BACKGROUND\_COLOUR = "white";**

**var SNAKE\_COLOUR = 'lightgreen';**

**var SNAKE\_BORDER\_COLOUR = 'darkgreen';**

**var FOOD\_COLOUR = 'red';**

**var FOOD\_BORDER\_COLOUR = 'darkred';**

**// The user's score**

**let score = 0;**

**// When set to true the snake is changing direction**

**let changingDirection = false;**

**// Food x-coordinate**

**let foodX;**

**// Food y-coordinate**

**let foodY;**

**// Horizontal velocity**

**let dx = 10;**

**// Vertical velocity**

**let dy = 0;**

**// Get the canvas element**

**var gameCanvas = document.getElementById("gameCanvas");**

**// Return a two dimensional drawing context**

**var ctx = gameCanvas.getContext("2d");**

**// Start game**

**Change the background colour of the canvas to CANVAS\_BACKGROUND\_COLOUR and**

**draw a border around it**

Draw the snake’s body and the snake’s food. Remember that the snake is a chain of squares, therefore we will draw only one square which will be repeated inside the snake array.

**let snake = [**

**{x: 150, y: 150},**

**{x: 140, y: 150},**

**{x: 130, y: 150},**

**{x: 120, y: 150},**

**{x: 110, y: 150}**

**]**

\* Draw the food on the canvas

\*/

\*\*

/\*\*

\* Creates random set of coordinates for the snake food.

\*/

// Generate a random number the food x-coordinate

// Generate a random number for the food y-coordinate

// if the new food location is where the snake currently is, generate a new food location

\* Advances the snake by changing the x-coordinates of its parts

\* according to the horizontal velocity and the y-coordinates of its parts

\* according to the vertical veolocity

\*/

Draws a part of the snake on the canvas

@param { object } snakePart - The coordinates where the part should be drawn

Draws the snake on the canvas

// loop through the snake parts drawing each part on the canvas

// Set the colour of the snake part

// Set the border colour of the snake part

// Draw a "filled" rectangle to represent the snake part at the coordinates

// the part is located

// Draw a border around the snake part

/\*\*

\* Changes the vertical and horizontal velocity of the snake according to the

\* key that was pressed.

\* The direction cannot be switched to the opposite direction, to prevent the snake

\* from reversing

\* For example if the the direction is 'right' it cannot become 'left'

\* @param { object } event - The keydown event

\*/

/\*\*

// Create the new Snake's head

// Add the new head to the beginning of snake body

// Increase score

// Display score on screen

// Generate new food location

// Remove the last part of snake body

/\*\*

\* Returns true if the head of the snake touched another part of the game

\* or any of the walls

\*/

//Saves current gamestate in local storage

//declare variables for JSON object

// convert myUserObj to JSON and store it as key/value pair

//use the property UserName of myUserObj as key for storage so that a user can be queried

//Searches local storage for a username. restores the gameboard if the username exists

//read in top scores values from local storage and parse them to ints