2207 SNAKE REBORN
FINAL ASSIGNMENT WRITE UP
NIKETA SHIRISH KINJAWADEKAR
A0113817J
18TH NOVEMBER 2015

For the code of my final project, I started writing the code on the flexbox layout code given during the game making class. Using that as a base and sticking to that layout for the app webpage, I built my game code on that. Knowing very well that the snake game is not an original idea, I decided to recreate it because that game always fascinated me as a child and I only got to play it on my old nokia phone when I was 12, so I decided I would recreate it as an app so that I would be able to bring that joy to big computer screens as well. Hence the name "snake reborn".

For the design of the webpage I used the basics from the first few classes, to design the divs I inserted images and manipulated the properties of the divs in css and changed the background colours as well. I installed a font on my computer called 'snake venom' and added it to the styling of the page for the header and the footer, however unfortunately, given that it is not a default font, it isn't shown on a computer that isn't mine. To give the app the design of an actual game page but simpler, I added an image of a highscore board on the right as well as added some properties of the game on the left of the main game canvas. Due to the lack of time, I was unable to make it into an active leader board so the aesthetic property would have to suffice.

Moving on to the actual game, I ripped and mashed code from various websites to get the code in running order. After facing multiple levels of frustration and confusion, I consulted numerous friends in order to get my game moving. Most of the code I used came from the website (w3schools.com) and some was hand coded with the regular guidance from friends. To start off I created a 2d canvas that would serve as the grid in which my "snake" would be moving, limited to moving, up, down, left and right using the arrow keys on a keyboard by creating event listeners for the keyboard arrows. Using the function Seefood I created an array of all the empty cells in the grid. Then using math.random (it choses a number between 0 to 1, multiplies it with the total number of empty cells in the grid to get a random index) sets the pebble at the randomly chosen position. Next it will initialize the grid and the snake, by default the snake will be placed in the upward direction, and the pebble will be set in the grid. Next I used the code to update the direction of the snake's movement,
(If the user presses left and snake is not moving right currently, change snake direction to left.
If the user presses right and snake is not moving left currently, change snake direction to right.
If the player presses up and snake is not moving down currently, change snake direction to up.
If the player presses down and snake is not moving up currently, change snake direction to down.
This is because the snake cannot change diection by 180 degrees.
If the snake is moving left, decrease x co-ordinate by 1.
If the snake is moving right, increase x co-ordinate by 1.
If the snake is moving up, increase y co-ordinate by 1.

If the snake is moving down, decrease y co-ordinate by 1.)
Next the conditions for game failure were set as, if the snake reaches the boundaries (eg: when x<0 or x>width-1 or y<0 or y> height-1) or the snake hits itself (haha). Given that the snake begins as a block just like the pebble, moving on in the game, if the snake eats the food, I made it such that the score increases by 1 unit and the new food cell gets placed randomly and a new cell gets added at the snake's head thus increasing the length of the snake I unit at a time as the game goes along.

As I had mentioned above, I realize this is not an original game idea but given that I really playing this game and as tedious as the process of making it was, I hope you enjoy playing the end result.