Game: Candy Counter

Creator: Koren Lam

I had designed a simple puzzle game which involves identifying the number of candies within 10 seconds. The game generates a random amount of candies depending on the difficulty level chosen. Games that test an individual's short term memory skills are proven to increase long term memory skills if training partakes everyday. What really inspired me to create this game was a memory game I often played when I was young called “Potato Counter” from <http://www.neopets.com/> website. Although Potato Counter uses static motion and a timer to indicate the score I had decided to create a version that is dynamic and timed.

I had decided to incorporate 2 assignments and create a game altogether using both user feedback and an array of candies. My final project uses the SVG layout content from previous assignments (Game with a moving object and array with dots) however I had customized many browser tools for aesthetics and better accessibility of the game. When designing the game I knew that the game would involve a timer function and the uses of arrays while creating interaction with the user. I had decided to execute my knowledge in arrays, loops (for/while), set timing events and user interaction. By merging these two assignments it displays animation and interaction to the player.

I have organized my code using proper directories and files. The javascript page is organized by variable definition, action event listeners and declared functions.

The instruction button is used to guide the player if they do not understand how the game works. The list of levels is known as the “userDifficulty” variable and determines the amount of candies that will be displayed depending on the level selected. The console.log helps the user specify which level he/she has chosen. An array was created using math.random function along with the variable “sizeJellyArray” to generate a random number between 1 and 15 depending on the difficulty selected. In addition, the “speed” variable is used to determine the rate of speed at which the candies are moving. If the player chooses ‘Easy’ then the speed will be set the highest and the number of candies will be low. If the player selects ‘difficult’ then this will trigger the fastest speed per second while the numbers of candies are increased. A “timer” function is called to end the game within 10 seconds and a prompt box will ask how many candies the player has counted. If correct, the player will be prompted that he/she is correct or else a window will state that he/she is incorrect and will have to play again by selecting a new difficulty. Lastly, the end game button is a “premature” function where the player has the choice to end during gameplay.

Some challenges I had encountered were some animation elements that could not be initialized, for example at the start of the game I was unable to hide the “sizeJellyArray” even when using Raphael’s tutorial (<http://cancerbero.mbarreneche.com/raphaeltut/#sec-animation> )“.toFront/.toBack” element functions. Therefore the aesthetic of the game could have been improved.