Ant Scape

The game starts off with a HTML page that has 2 radio buttons for selecting the levels and the start button to start the gameplay. Once the start button is clicked the HTML div is hidden and the canvas appears with the start of the gameplay.

The animation is a time based where by it calculates how long the last frame took to load then sets the load time of the next frame to match it. This is done so the game icons move/animate at the same speed on various devices and at different frames per seconds. The *setTimeout* function is used to set the frames per second to 60fps. While the *setInterval* function is used to countdown from 60s every second once the game starts. The *setInterval* function is used to create new Ants between 1 to 3 seconds randomly. The game animates using the *requestAnimationFrame* function using the various webkits to allow smooth function of the animation on various browsers.

When the page is loaded initially the *startpage* function is called which shows the start page of the game with selectable levels. When the start-button is clicked, all the variables are reset and the *onload* function is called which draws the canvas with the top-bar holding the timer, score and pause buttons. The food are set on the canvas and the speed of the Ants are set depending on the level selected.

The *ifclicked* function determines the x and y coordinates of a mouse click. If the pause button is clicked or if a click happens within 30px radius of any Ants. The function then determines which color of Ant was within the radius and adds the score accordingly. The Ant is then removed from the game.

The *draw* function simply draws the top bar of the game which holds the score, timer and pause button. The *updateAnts* function cycles through the array of Ants and check which color of Ant it is then passes it to *updateAnt.*

The *updateAnt* function takes 3 parameters which are speed of the Ant, the image path of the specific ant and the index in the array. It creates the image of the ant on the canvas. Then the closest food to that specific ant is found and the position of the Ant is updated to move towards the closest food. The ant moves in a diagonal line to the food closest food item. The Ant image is rotated to match the direction of movement.

The *updatefood* function checks to see if any ants are within the perimeter of the food items and if so then the food items are removed and the remaining food items are redrawn. The *setFood* function simply sets the food x and y coordinates randomly on the canvas. The *ran* function determines a random number between the parameters min and max of the function.

The *getDistance* function gets the distance between the food and ant while the *gethighscore* sets the high score in the local storage. The *newAnt* function spawns a new ant at a random x coordinate at the top of the canvas. The *settimer* function decreases the time from 60s and signifies gameover.

During gameover the HTML page becomes visible with their score and two options to restart and exit the game.

Test Plan

The game was manually tested on 3 different browsers, Firefox, Chrome and Internet Explorer.

1) The game was tested on the 3 pages to check the animation between the start page, canvas and game over page. It was tested to see if screens switch without any delays and the overlook and feel was the same from page to page.
2) The game was run on level 1 to check for bug speed, bug color, and timer. The pause button was clicked to pause the game and resume the gameplay. The game was run with any mouse events to check if the bugs were moving towards the nearest food and when all the food was gone then the game ended.
3) The mouse event was checked to see the functionality of the ifclicked function. When the left mouse button is clicked then any Ants within 30px radius should fade. While the remaining Ants continued towards the food.
4) The game was tested to end when the timer reached zero to signify the end of the game.
5) The ant speeds were tested by changing the speed variables within javascript to check if the functionality was working correctly
6) The ant score was tested by changing the score of the different ants within javascript.
7) The food items were checked by increasing and decreasing their boundaries of when an Ant is within their boundaries then the found should disappear.
8) The canvas flexibility was checked by increasing and decreasing the screen size.
9) The game was given to 2 other individuals to test the over functionality and to check for any new errors during gameplay.
10) Various variables were changed using FireBug.