README

Daniel Eryavec
g3olimar

Design Doc:

Game title: Tap-Tap Bug

The Screens:
There are 3 screens:
The Start screen contains the radio buttons for selecting difficulty, the button that starts the game, and a high-score indicator.
The Game screen contains the timer, a score counter, the pause button, the game canvas, and a quit button.
The Game Over screen contains a message telling the user the game is over, a score and high score indicator, and 2 buttons, one for retrying and one for quitting.
The screens have sidebars on wide screens with art I drew.

A NOTE ON MY ARRAYS:
Rather than use JSON objects I leaned towards using arrays whose indexes corresponded to arrtibutes. I realize in retrospect that using objects would have
a) made my code more readable and
b) encouraged me to code object functions
I will now explain the arrays that SHOULD have been objects here:

livingBugs is an array of arrays that represent the color and x and y coordinates of any living bug.
dyingBugs is an array of arrays that represent the color, x and y coordinates, and alpha of any dying bug.
foods is an array of arrays that represent the x and y coordinates and eaten-status of the five foods.
bugs is an array that contains the name of the color, level 1 speed, level 2 speed, score value, and left and right sprites of each color of bug.

FUNCTIONS:

window.onload
Attaches event handlers and initializes some variables

pauseHandler()
click event handler for the pause button. Pauses if unpaused, unpauses if paused.

pause()
pauses the game by clearing the interval

unpause()
unpauses the game by setting the interval

visibilities(a,b,c)
Start menu is visible iff a, Game view is visible iff b, Game Over screen is visible iff c.

start()
Start the game by switching to the game view and initializing all relevant variables. starts the first frame of the game and sets the interval. It uses whatever difficulty the user chose. The visibilities are changed so only the Game screen is visible.

shootBug()
If the game is not paused, kill all bugs withing a 30px radius of the cursor.

killBug(bugIndex)
remove the bug at livingBugs[bugIndex], add the score for bugs of that color to the score, and add the bug to the list of dying bugs.

within30px(bugX,bugY,x,y)
return true iff a 10px * 40px rectangle with the top-left corner at (bugX,bugY) is within 30px of (x, y).

game()
One frame of the game. If it is time to add a bug, add a bug. If it is time to change sprites, change sprites. Erase un-eaten food. Move the bugs to the closest food. Fade out dying bugs. End game isf all food is eaten or if time is up.

addABug()
Add a new bug at some point on the top of the screen.

moveABug(bugIndex)

Move the bug at livingBugs[bugIndex] closer to the nearest food at the speed for the current difficulty.

eatFood(x, y)
Set any food whose sprite overlaps with a bug whose top-left corner is at (x,y) to eaten.

gameover()
The visibilities are changed so only the Game Over screen is visible. High score is updated if necessary.

quit()
The visibilities are changed so only the Start screen is visible.