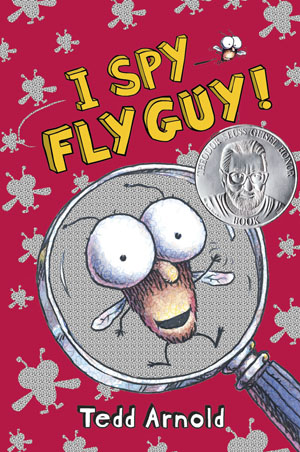
Goal: Make a simple guessing game

- finding Fly guy

Statement:

If you had a sibling aged around 1st grader, one of the book the school recommend to read is “Fly guy” – and with some reasons, kids love the story. The story is about the boy who adopted a (smart) Fly as a pet. One of the main adventure between two is to find the Fly Guy, because it is not easy to spot him and it is very easy to lose him. Based on the (random number) guessing game and Alien game in Chapter3, I want you to make a simple game to find the Fly Guy. You will be given a scene (your choice of background or solid color) and your pet fly can be randomly located on the background - you will guess the coordinate of the Fly by entering x and y coordinate in the input box at the bottom. When you press “examine” button, you can see if your guessing is correct or not, by highlighted color of “search beams” that turns its color green when it touches the fly. You can see if your guess on y coordinate were correct, if the Y-beam turns green. You can adjust the width of the search beam to make it easy to find him. You are given a limited time of chances (ex. 10 times, but you will choose the number) to guess and if you used all the chance, the fly will change the location. When your guessing coordinate (x, y) is inside of the the zone where fly is sitting, you will show x- and y- highlighted green breams (and maybe celebrate by blinking or rotating the beams) and show the magnified Fly Guy.



Step0: Building the workflow

1) Introduction

“A boy had a pet fly. He named him Fly Guy. Fly Guy was the smartest pet in the world. He could say the boy’s name “Buzz”. They always love to play a hide and seek. Could you help Buzz to find Fly Guy?”

2) How to play

“You will enter x and y coordinate in the input box below; and click “examine” button. If you found the fly on that coordinate (or nearby), the search beam (x- or y-) will turn green, otherwise the beams will stay gray.”

3) End of the game

If you found the Fly Guy, the x- and y- turn green and you can see the bigger image of Fly Guy (optional: and see the message, “you found it!”)

Step1: set <html> elements for JavaScript

* In <body>, show the introductory sentences containing the blank for the participant’s name

<div id=”into”> [some text] </div>

* ask the participant’s name (p.103)

<input id="xcoord" type="number" placeholder="x" />

<input id=”ycoord” type=”number” placeholder=”y”/>

* provide a button to remember the name in a variable (p. 93)

<button id=”button01” > Examine </button>

* provide a background image or color
* provide a place you may want to show the tiny Fly Guy <div id=”playground”> </div>
* (default position of search beams) you can either create by drawing (canvas, which will need some extra javascripts) or use image file. Either way you feel it comfortable should work.
* Complete the css code for the element you created for presentation (“id”s and buttons; width, height, and basic color settings)

Step2: Start JavaScript

* After </body> before </html>, add <script> to write JavaScript codes
* Make a variable to find an element; assign the element to a variable (p.89)

If you want to find the element whose id is “button01”,

var myButton01= document.querySelector(“#button01”);

if you want to find the element whose tag is <body>,

var myBody = document.querySelector(“body”);

\* Be sure to make the id, the class, or the element you used in the querySelector accessible. Doublecheck if you had those selectors in <script> in the <html>

* Add event listener to the button for the input, to listen the click event (p.93-7)

Button01.addEventListener (“click”, clickHandler, false);

* Define function (ex. clickHandler)

function clickHandler () {

DO\_SOMETHINGS…

}

Step3: make variables for the random position of the Fly Guy (p.134-138)

Step4: make variables for the input x and input y from the users; validate the new input numbers, the numbers should not negative or too large than screen size (p.172)

Step5: when the button “examine” clicked, run a function that compares the position of the fly and the user’s inputX and inputY

Step6:

* write a code to check if either user’s input x or input y were inside of the zone that contains Fly Guy’s image (p.167-p.170, or Chapter8: Collision detection, we will cover that topic in detail later)
* if the position were overlapped in any axis, show the inverted color of the image (green) on that axis’s search beam, otherwise show the original color of the search beam

filter: invert(100%);

-webkit-filter: invert(100%)

Step6: Apply CSS and improve overall look&feel and usability

-----------------------------------------------------------------------------------------------------------------------

Lab03: Fly Guy

Rubrics:

(3 points) Create a random position of Fly Guy and show it

ex. whenever I refresh the html, I expect to see a new positon of Fly Guy

(5 points) get user’s input (x, y) and compare it with the position of Fly Guy

if it were the coordinate in Fly Guy, it will show any message “you got it” or show a graphical interface (changing the color of search beam, or a magnified image of Fly Guy)

if it were not overlapped with any part of Fly Guy, it will reset the input (and count up/down, such as “try it again, 5 remaining” or)

(2 points) show a proper graphical interface and messages (such as introduction, end of game)

(extra 5 points) nice animation of search beam (ease option, inverting)