Matching Game Plan

Step 2:

* Part 1 - developing the web page content without javascript
* Part 2 - generating the left side images
* Part 3 - handling the right side images
* Part 4 - finishing the game

|  |  |  |
| --- | --- | --- |
| **Goals:** | **Dates/ Deadlines to Finish By** | **Benefits:** |
| 1. Finish the HTML & CSS by the end of class. | 11/17 | Start on Javascript ASAP, also to build the basis of the webpage. Alongside to create style rules that are applied to all the images. |
| **2.** Add Javascript that generates number of faces on only the left side. In the function the faces are created in a loop. | 11/22 | This sets the premise of the game, and lets us start working on the other side’s logic. Furthermore, the faces are created on a loop to determine the image source, position, and value. When the page is loaded the function generateFaces() runs it. |
| **3.** Extend the Javascript to handle the right side. From adding a variable. | 11/23  or  11/28 | A new variable is finally created for the right side, by copying the left side and adding it to the right side. |
| **4.** Extend Javascript to handle the events and game logic. Add an onclick event handler function to the last child of the left side, in which the player would click on, then delete all child nodes. | 12/06 | Helps construct a function that would be executed when the user clicks on the last child node. Also to make sure that the line event is only applied to the function once not on the other elements of the web pages. Also to make sure that if the user clicks on anything except the correct face, there would be another function to handle the situation |

Step 5: Possible Issues:

* Unexpected absences from group members
* Group Members being tardy
* Holiday absences
* Group Members might be getting sick due to temperatures dropping causing them to be absent
* West-mec internet might crash and servers could go down
* Syntax errors
* Logic errors

Requirement Analysis

1. The program will run on any browser, preferably Google Chrome, and any operating system including windows, Macintosh, and Linux
2. Any computer or smartphone
3. The end user will be playing a game in which he has to click on the extra smiley face on the left side. If anything except the correct face is clicked, a message is displayed saying that the game is over. If the correct face is clicked, all the currently displayed faces are deleted and a new set of faces is shown at random positions. Each time a new set of faces is shown there will be 5 more faces than before, on both the left and the right sides. There will always be one extra face shown on the left.

User wins the Matching Game

1. User launches the html document, and the game starts.
2. Five faces are shown on the left and four are shown on the right, but the left has an extra face.
3. The user identifies the face that does not match
4. User clicks on extra face.
5. The correct face is clicked and a message is displayed saying that the game is over.
6. With the correct face chosen, the current are deleted, and randomized.
7. 5 more faces are added onto both the left and the right side.
8. User continues until multiple amounts of faces are on the screen

User loses the Matching Game

1. User launches the html document, and the game starts.
2. Five faces are shown on the left and four are shown on the right, but the left has an extra face.
3. The User clicks on the wrong face.
4. The incorrect face is clicked and a message is displayed saying that the game is over.

Design Phase

1)Design Phase.png

A.

B.

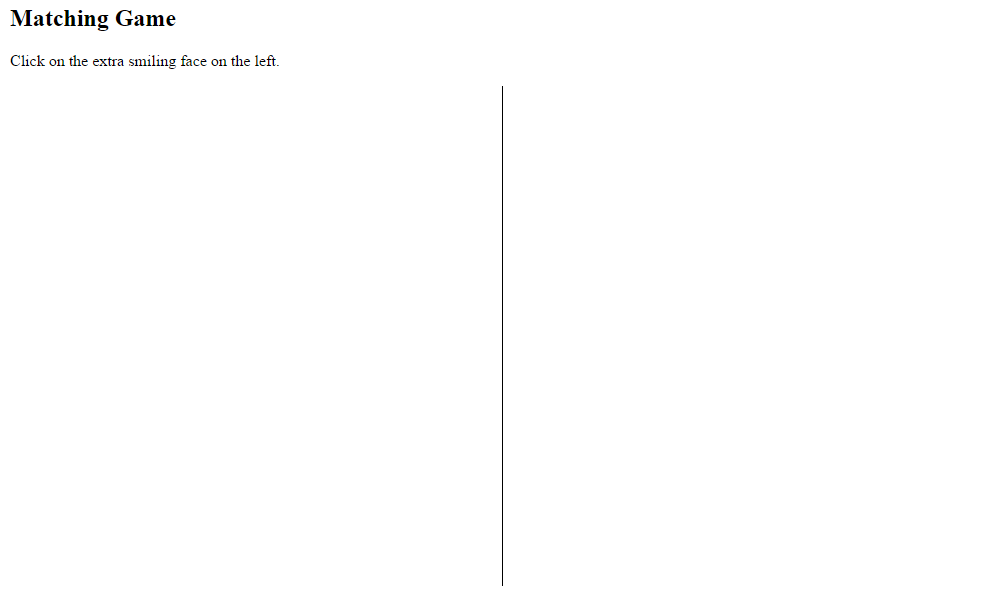
C.

D.

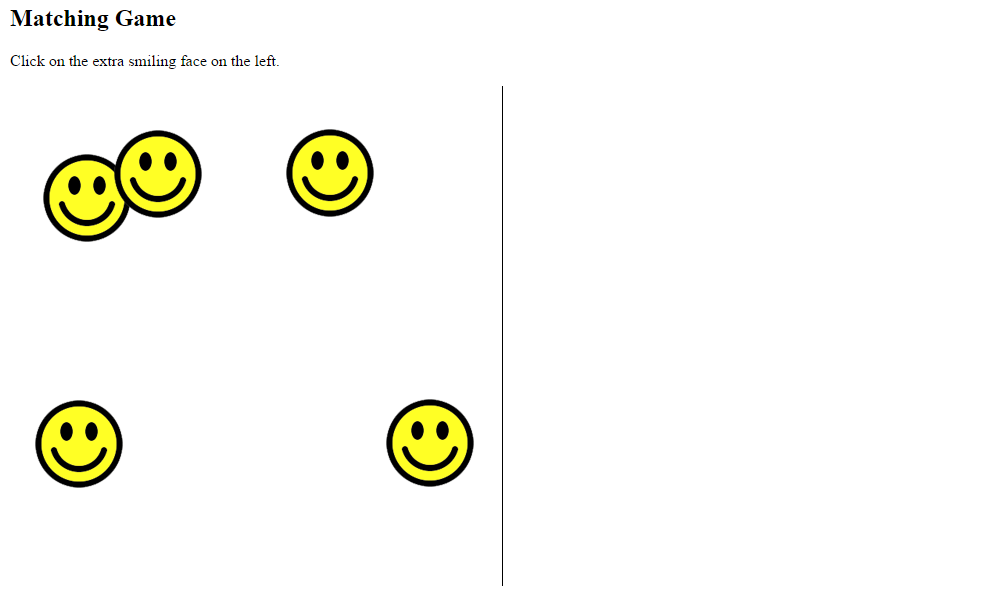
E.

2)

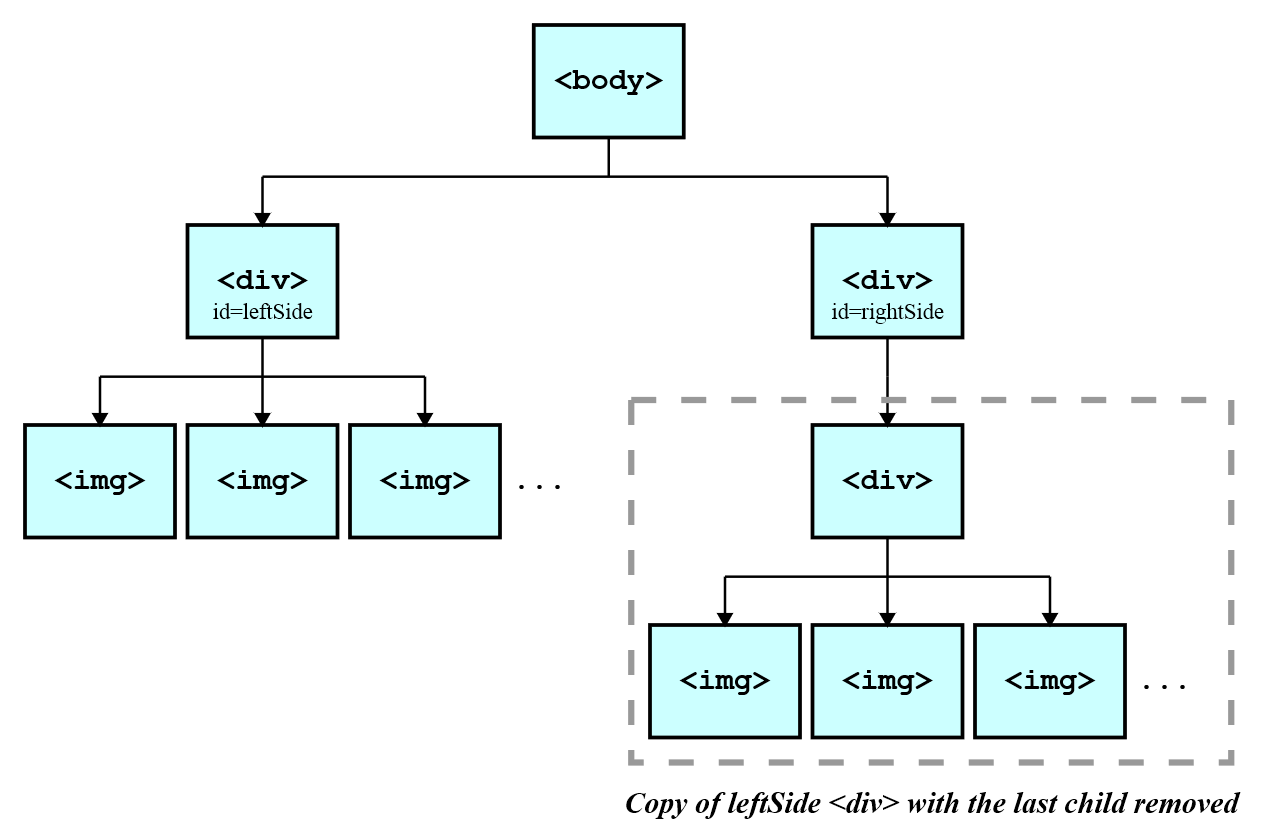
**Part 1**



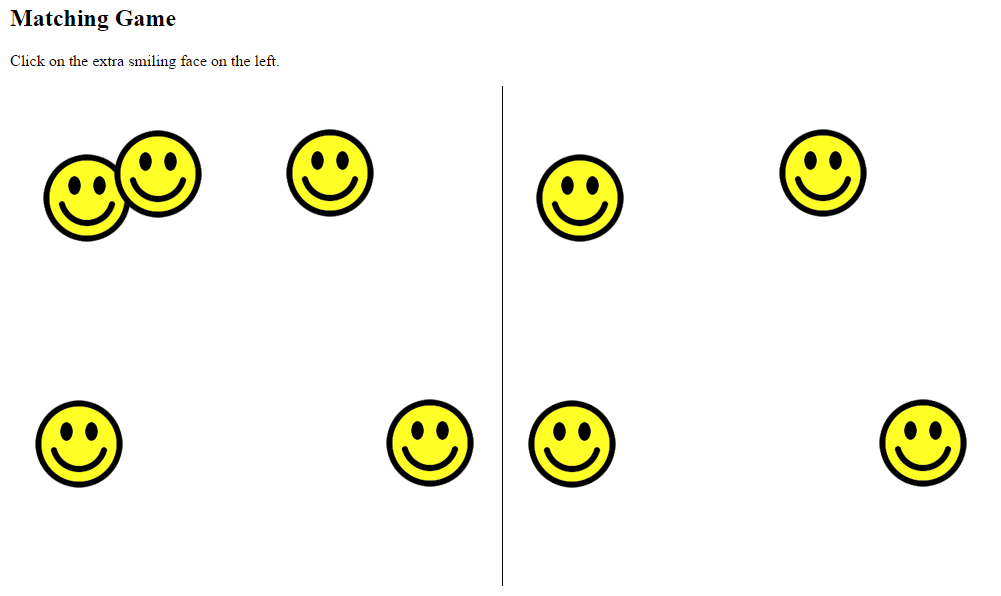
**Part 2**



**Part 3**



**Part 4**



3)Flowchart.png