# Summary

We are going to upgrade our game graphically and mechanically.

* The game will support any number of Fighters on any number of teams
* The game will move in “rounds”, in which each Fighter will have a “turn”
* Each fighter will be highlighted on the page and their controls activated when it is their turn
* The fighter’s attack ability will attack a random Fighter NOT on their team and alive, and end the turn.
* Once all fighters have taken a turn, the round resets
* If a Fighter “dies”, they are greyed out and cannot be attacked or take turns.
* Activity in the game should be output to page
* See layout below:

|  |  |
| --- | --- |
| Team 0 | Team 1 |
| Output | |

# Data Structure

Organize the data of the game to accommodate future upgrades to the app.

1. Create a number to track ‘all fighters’
2. Create a list of names for Fighters
3. Create a ‘Fighter’ class that all fighter objects will be made from
   1. In the constructor:
      1. The constructor arguments are: id, name, team
      2. Has the properties: id, hp, hp\_max, atk, spd, name, ready, team
      3. name is set to the name argument
      4. team is set to the team argument
      5. id is set to the ‘all fighters’ tracking number
      6. ready is set to true
      7. Pick default numbers for all remaining properties
   2. Create an “attack” method that takes a ‘target’ argument
      1. For each Fighter in the Fighter list:
         1. If their hp is greater than 0, reset their img’s src property to the appropriate idle image.
      2. Create a temporary counter variable
      3. Using a do..while loop, increase the counter variable by one and randomly pick another Fighter until you’ve found one whose hp is greater than 0 and is not on this Fighter’s team
         1. If the counter variable is greater than the length of the Fighter list, tell the user in the output that there are no targets, break the loop, and return
      4. Decrease the target’s attack by a damage formula of your choice that incorporates this fighter’s atk property.
         1. If the target’s hp is less than 0, make it 0 and add the “dead” class to their div box (see below) and set the Fighter’s target img src property to the appropriate image
      5. Change this Fighter’s img src property to the appropriate attacking image
      6. Change this Fighter’s target’s img arc property to the appropriate damage image
      7. Change the target’s hp as displayed in their stats box
      8. Append text updates about the attack to the output div (see below)
      9. Remove “active” class from this Fighter’s div (see below)
      10. Change this Fighter’s ready property to false
      11. Increase the game turn counter by 1 (see below)
      12. Call the Game Loop function (see below)

# Game Setup

Start by setting up all the “backend” stuff to handle the game.

1. Create an empty list of Fighters
2. Create a constant number storing the number of teams and set its value to 2
3. Use a normal for loop (using ‘i’ as the iterator) to push two fighters into the Fighters list
   1. Each Fighter will randomly pick a name from the Fighter name list
   2. Each fighter’s team will be the remainder of the iterator divided by the number of teams
      1. ( i % numberOfTeams )
   3. Increase the ‘all fighters’ tracking number by one
4. Set a number to track the game round
5. Set a number to track the game turn
6. Call the Game Loop function (see below)

# Page Setup

Draw the bulk of the page using Javascript

1. Use a for loop to create a number of “team” div boxes equal to the number of teams
   1. The id of the div box will be “teamX”, where “X” is the number of the team (0 or 1)
   2. The class of the div box should be “teamBox”
   3. Each team should be a column in a row of teams
2. For each Fighter in the Fighter list (for..of):
   1. Get this Fighter’s index in the Fighter list ( using indexOf() )
   2. Create a Fighter div box with the id “FighterX” where “X” is that Fighter’s id property
      1. Give it the “fighterBox” class
      2. Create an img element with the class “fighterGFX” and append it to this div
         1. Set the img src property to the correct image
            1. ‘crashr\_idle.png’ for team 0, ‘saml\_idle.png’ for team 1
      3. Create a div box with the class “statsBox” and append it to this div
         1. Put this fighter’s atk and hp properties here
      4. Create a div box with the class “controlsBox” and append it to this div
         1. Add a disabled “Attack” button to this box
            1. The onclick property should be this Fighter’s attack method
            2. (You can find this Fighter in the Fighter list by using its index you got earlier)
3. Create a final div box with the id “output”

# Game Loop

1. In a function:
   1. If the game turn counter is greater than or equal to the length of the Fighter list:
      1. Reset the game turn counter to 0
      2. Increase the game round counter by 1
      3. For each Fighter in the fighter list:
         1. If their hp is greater than 0, reset their ready property to true
   2. Sort the Fighter list by their “spd” property (highest to lowest)
   3. For every Fighter in the sorted list
      1. If the fighter whose index in the sorted Fighter list is the current game turn counter, their ready property is true, and has more than 0 HP:
         1. Add the “active” class to their box (find by using their id property, not index)
         2. Activate their Attack button
         3. Tell the user it’s the Fighter’s name’s turn in the output box
      2. Otherwise, increase the game turn counter by 1 and call the Game Loop again

# Design Notes

* You can getElements inside a specific element by using the appropriate getElement method with the element you want to search inside:  
  // Get the box you want to search  
  let fighterBox = document.getElementById('fighter' + index);  
  // Retrieve a list of img tags inside that box by class  
  let fighterIMG = fighterBox.getElementsByClassName('fighterImage');  
  // Retrieve a list of img tags inside that box by tag  
  let fighterIMG = fighterBox.getElementsByTagName('img');
* Remember there are many different types of for loops that all function slightly differently. The biggest key difference is are you looping through the elements themselves, or the indexes of those elements
* Indexes and id’s are often confused. Remember that the HTML elements will have “ids” and “classes”, and Fighters have an “id” property that has nothing to do with HTML.
* For loops and functions can greatly decrease the total amount of code you must use.