Harry Parkes

Phase 10 Score Sheet

# Introduction

For ages I have wanted a simple way of scoring my games of Phase 10. I also want to teach myself how to write in JavaScript and improve my general programming skills. To fix my problem and learn JavaScript I will program a scoresheet website using HTML5, CSS3, and JS.

## Outcome Statement

I propose a digital solution to fix many of the problems with scoring card games. My solution will be an application which allows users to track their scores. This application should meet the following:

### Requirements

Reduce opportunity for arithmetic error (When compared to paper scoring)

Provide the ability to keep track of scores and player positioning over the course of a game.

Allow for scoring of Phase 10.

Protect the eyes of the app users.

### Specifications

To reduce opportunity for error, the program will sum scores itself, so the user doesn’t have to.

To provide ability to keep track of scores, At the end of every round, each person playing the game will enter the amount of points they scored. The scores of each round will be displayed in a table. My program will then sum each player’s scores and display their total score.

As to not hurt the user’s eyes when playing at night, the U.I. will be dark.

Link to Trello: <https://trello.com/b/PlhvUDR2>

## Implications

|  |  |
| --- | --- |
| **Implication name** | **Description of the implication** |
| Usability | Is the program user friendly and easy to access? |
| Functionality | Does that program function how it was intended? |

### Usability

My outcome will be affected by usability implications. This is because my outcome is intended to be used. Usability means that my program will have to be user friendly, feel consistent and be easy to access. If users do not understand how to use my scoresheet, they will not be able to properly use my program. To make sure that everyone is able to properly use my program it will match how someone would score their game on paper meaning they would already feel familiar with my interface before they have used it. I will ensure there is minimal information on the screen, and all information given is relevant.

### Functionality

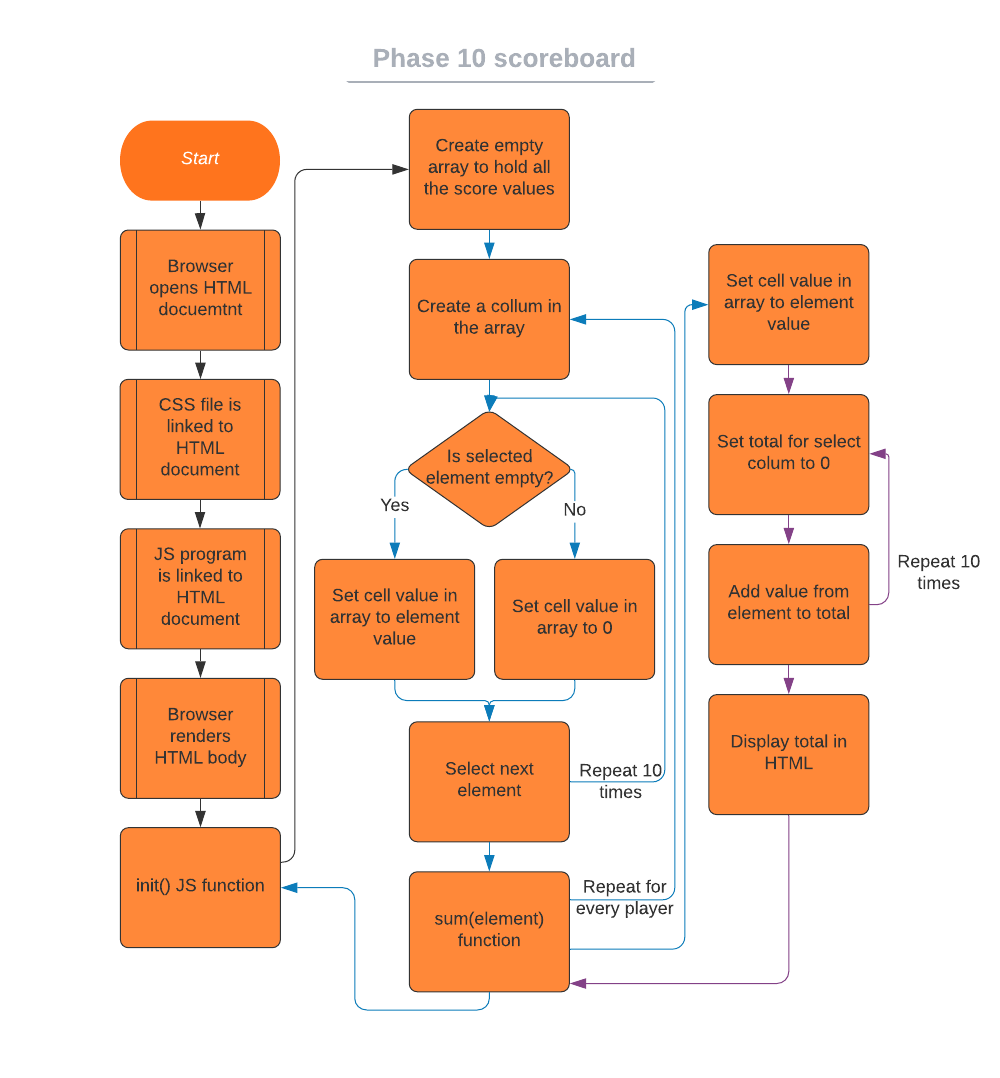
My outcome will also be affected by functionality implications. This is because my outcome is intended to be functional. Functionality means that my program will have to function how the user expects it to function. If my program is unpredictable it will not fulfil its function. I will have to make sure that my program achieves its purpose in a simple and functional way.

# Planning

## Decomposition

My program consists of multiple key parts. These parts are the site’s content, which is written in HTML; the site’s styles, which are written in CSS; and the site’s logic, which is written in JS.

## Flow Chart



Note on my testing:

I use Visual Studio Code as my IDE of choice, so all my code snippets are in a dark theme (panda syntax extension). Also, for texting code I use the Firefox developer tools console. P.S. I have designed the CSS to look good in Firefox, opening in a chromium browser will not achieve the same results, for reasons out of my control.

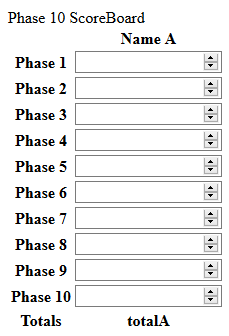
# Designing the components

## Component One – HTML

The first piece of my program is the content of the website, this is the text and table.



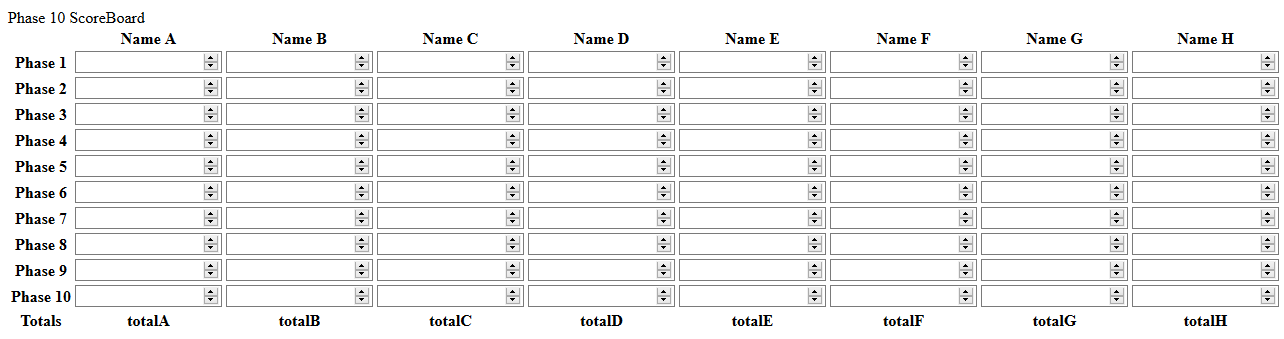
Here I have written a basic table with two columns, it looks like this



Each table data element (*<*td*></*td*>*) contains an input element (*<*input*>*) which allows the users to write in their scores. Each input element has an ID tag (id="cell\_A1"), a type (type="number"), a minimum value (min="0") and a step value (step="5"). The ID tag of an HTML element is a unique identifier which allows me to use the element in JS. The type attribute tells the browser that the input element is specifically for numbers. The minimum value means the number entered into the element may not be below this value. The step value means that any value entered into the element must be a multiple of the given value.

I have set these attributes as this is how the scoring works in Phase 10. All scores are numeric, cannot be negative (minimum value is 0), and are multiples of 5.

I added the rest of the columns to create this

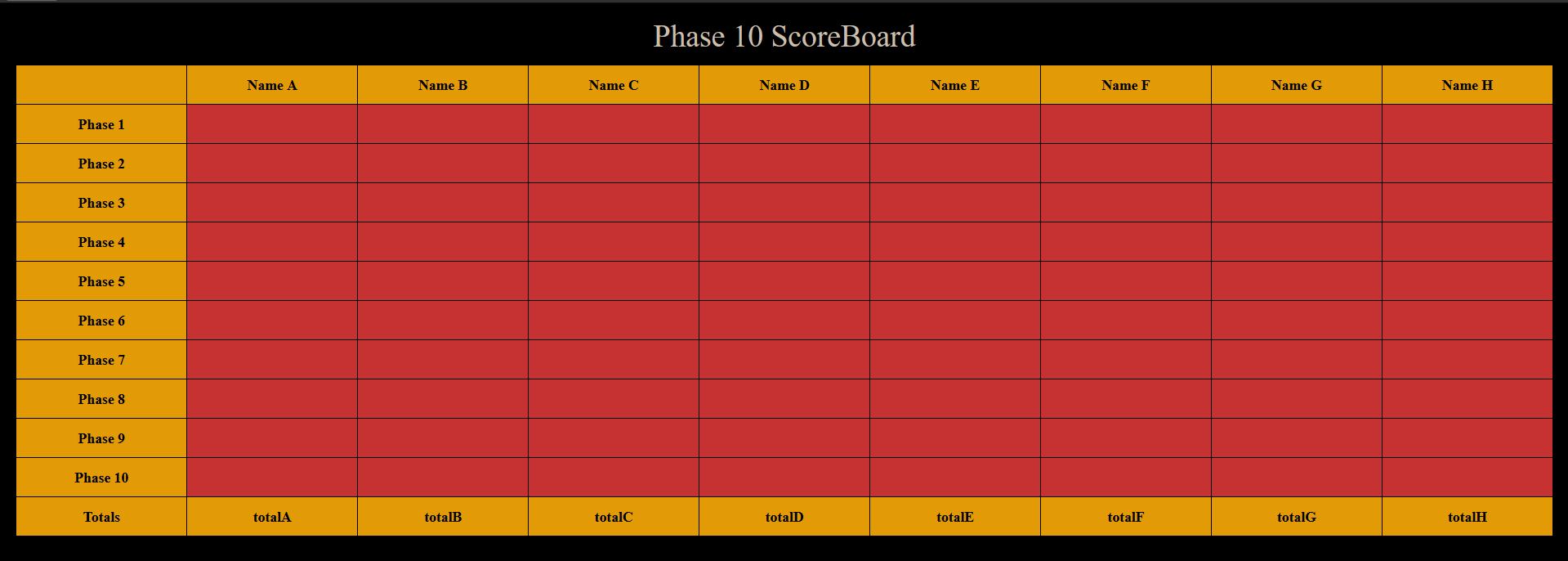


## Component Two – CSS

The second component is the styling. I need to be able to turn the above scoreboard into something that looks appealing and minimalistic. CSS, Cascading Style Sheet is the language used to do this. I wrote the following code to style my scoreboard.

For some reason, Word has hidden the bottom of this code block.

Now, rendering my page gives the following:



I have used yellow and red as they are colours from the Phase 10 branding. The black background is to minimise light emission from the screen to help save the user’s Eyes.

## Component Three – JS

The third part to my program is the logic that takes the values from the cells and sums them up. I have done this using JavaScript. My JavaScript program can be further split into two sections. These are the Initialisation, and the dynamic summation.

### Component Three A – Initialising

#### Version One



In the first version of my program, initialisation consisted of creating a multidimensional array to store score values in, creating an array to store column totals in and an array for iterating through the alphabet.

#### Version Two



In the second version of my program, I initialise by making an 8 by 10 array with every value set to 0.

#### Version Three



Before writing the third version I had realised that initialising all the table values to 0 made all the summed values also 0, even if some cells were filled upon initialisation. Instead of setting all values to 0, I only set the empty cells to zero and set all the cells with existing values to their existing value. Also, to do this I had to run the code after the HTML had been rendered so I placed the code in a function and called the function at the bottom of the HTML file.

### Component Three B - Summing

#### Version One



In the first version of my program I used a for loop inside of a for loop to iterate through the entire scoresheet.

The entire thing is repeated every 500ms to ensure scores are always up to date

#### Version Two



In the second version of my program I have removed the 500ms repeat and replaced it with an onchange call to a new sum(element) function in the HTML. The sum(element) function takes in a parameter which is the ID string of the HTML element which has changed. The function resets the total value of the changed column, adds the new value of the changed call into my table array, and then sums all the values in the changed column. Once all that is done it shows the updated total value in the HTML.