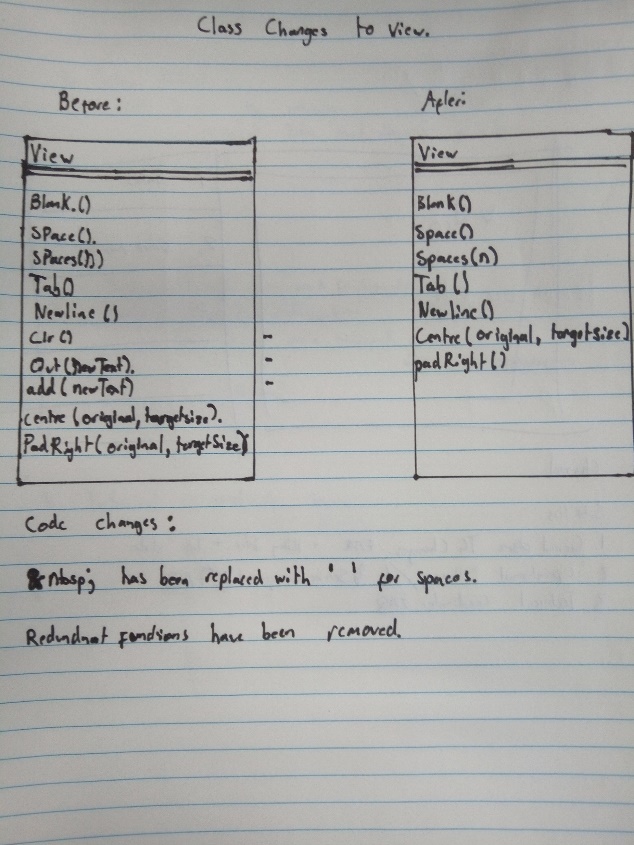
# Iteration 2

## Goal:

Optimise, add readability and eliminate redundancy in code

## Class Diagram of Comparison of View



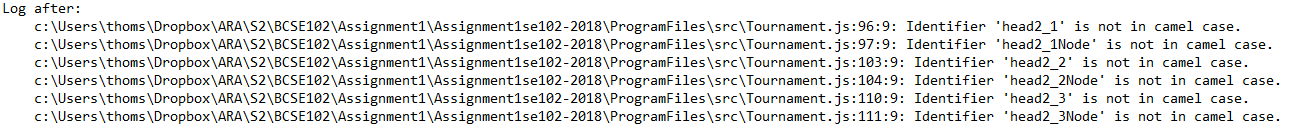
## Tasks:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Description | Products | Time est: | Time Taken |
| Planning | * Complete Worksheet and Planning sheet | * Worksheet | 1 hours | 1.5 hours |
| Analysis: | * Build Class Diagram of existing System * Look in code sheets and identify errors for fixing | * Class Diagram | 1hour  2 hours | 30 mins  2 hours |
| Designing: | * Create pseudocode for dunction we have to fix * Complete Complex algorithm sheet | * pseudocode * Algo worksheet | 30 mins  30 mins | 15 mins  30 Mins |
| Building | * Duplicate function removal * Data Generation and Storing fixing * Adaption of View function for html formatting   + Testing this will involve creating a CSS Sheet * Fixing Standard JS * Debugging * Commenting | * Working assignment (Hopefully!). * Some Styling with Div Classes being formed | 15 mins  30 mins  1 hour  15mins  15mins  30 mins | 10 mins  30 mins  20 mins  (30 mins)  20 mins  30 mins  60 mins |
| Testing | * Check all Data is displayed through comparison * Run through standard Js. | * Tested Web page | 15 mins  10 mins | 30 mins  10 mins |

## Standard JS Log

I fixed all the gripes and errors with previous code that StandardJS noted. These exceptions listed in the log below are technicalities that the system can’t pick up, not the fact that it isn’t in camel case.

Our new StandardJS log:



## Pseudocode for adjustment to sport.addMatch()

addMatch () {

when = new date (DATE HERE)

pool = create a new pool by using addPool()

teamA = create a team using addTeam()

teamB = create a team using addTeam()

add teamA to the pool using thePool.addTeam(teamA)

addd teamB to the pool using thePool.addTeam(teamB)

call the match class: newMatch = new Match(MATCH HERE)

+ add the match to pool array using thePool.allMyMatches.push(newMatch)

add the match to sport array using this.allMyMatches.push(newMatch)

The 2nd to last line of code was missing in the supplied model answer. This addition and restructure added all the matches to the pools as well, which fixed data dependency errors for outputting some results in the Tabulated data of tournament.getResults()

## Review:

### Mistakes:

I took the charge into manipulation of code without observing what the code did to its full extent.

For example, my trial and error removal and fixing of the pool matches took a lot longer than necessary by poor analysis of the data generation structure. I should have looked where and when the data generation was occurring, and then analysed the best place it could have gone, instead of taking random stabs at code to provoke it to work.

This Charge also meant that I never noticed some of the redundancy availability in the code until the end of this iteration. During my commenting session, I noticed (near) identical files with what looks like a test version of adding pool results. I removed this file, but it has highlighted my lack of attention to detail against the whole project.

### What I Will do Different Next Iteration

Taking More Time in the analysis section of the Process may be a more beneficial part of the process, especially for code maintenance sections like this. Also, reducing the amount of trial and error might be a good thing. I think more regular GitHub commits might be a better idea. There was only a few commits for this branch merge.