Project Diary

# Weeks 1 & 2 (11/1/2016 – 24/1/2016)

* Prototype code was refactored into MVC using Alex Netkachov’s (2015) template and old (non-MVC) code achieved into “prototype” directory
* Included a custom event class that allows listeners to subscribe to an event and receive notifications with data when the events fire: other implementations of events involved sending notifications through the DOM, which I thought was messy and not true to MVC
* MVC required me to start using JavaScript objects/class, which use a prototype model
* Original DFD diagram had to be changed:
  + All rendering is now done in the view, as it regards the presentation of data and should have no effect on the representation of data in the model
  + Following on from above, a custom colour class was created, allowing me to define colours how I want, independently of the rendering framework, which should be isolated to the view

# Week 3 (25/1/2016 – 31/1/2016)

* Prototype is now completely refactored into an MVC design
* Began looking into testing, allowing me to start TDD and unit tests for existing code
* Configured WebStorm to use Karma to run QUnit tests (JetBrains, 2015), this took a day and was a lot of faff but nothing important to conclude
* Tests written for some existing code, mostly easy to test functions such as the utils and model
* Some functions are noted as being untestable, mostly due to being unable to simulate file inputs by loading a file from local file space: I intend to either modify them to become testable or to define manual tests that can be used instead
* Colour quantisation function in the model is the first to be written using TDD (ish!)

# Week 4 (1/2/2016 – 7/2/2016)

* Ran into confusion with the values in my LUT and during the conversion between 24-bit and 16-bit
* Resolved confusion: multiply 16-bit value to get 24-bit value
* I initially thought it was a factor of 16 (16 \* 16 = 256) and while this maps 0 to 0, and 1 to 16, 15 (16-bit max value) maps to 240
* I knew the 24-bit range would be divided into 16 ranges (256 / 16 = 16) but I believed these ranges would have a size of 16
* Instead they have a range of 17: 255 / 17 = 15 divisions, plus 1 from 0 to 17, totalling 16 divisions with a range of 17
* Confusion resolved, new code written, and tested