5: Appraisal

5.1: Meeting objectives

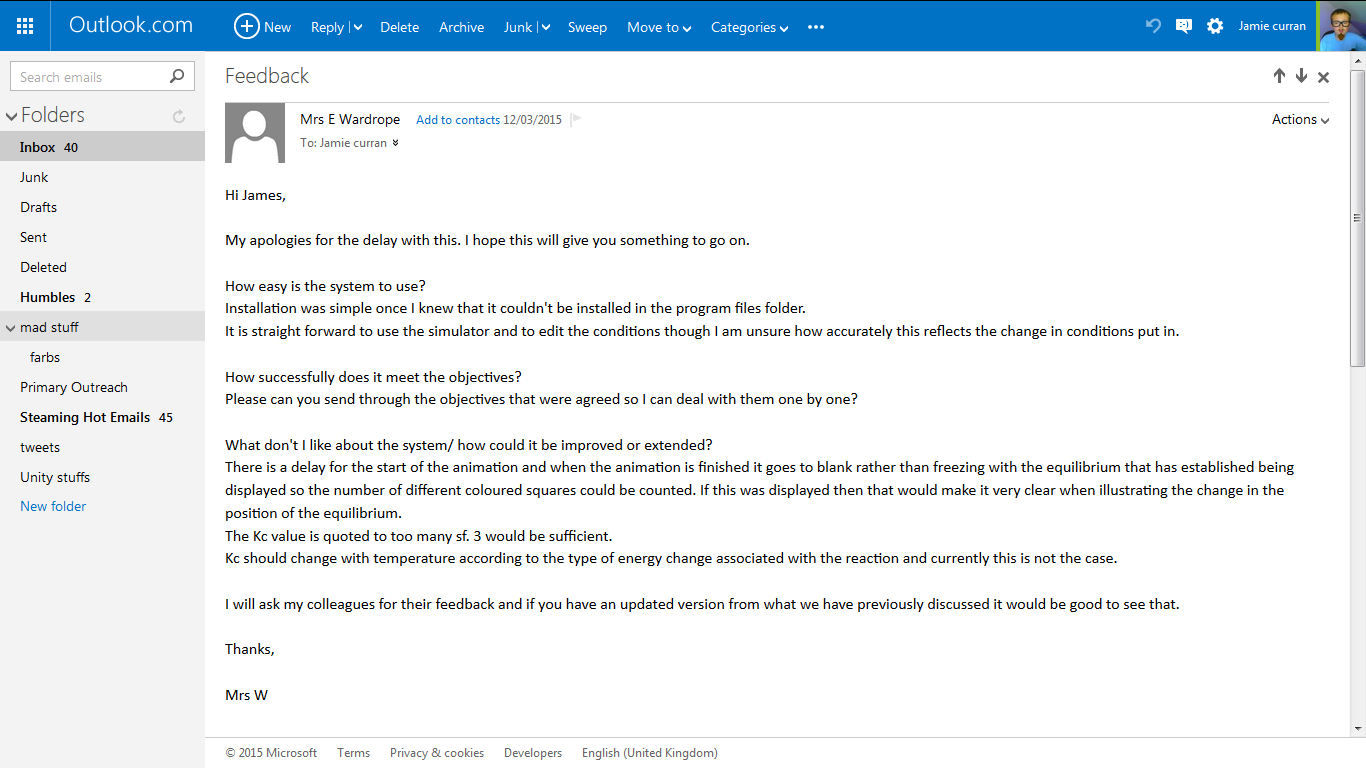
The program’s objectives are repeated here:

1. The program must be able to save and load files that it has created one at a time, within two seconds for each file. The program files must consist of a maximum of five separate reactions.
2. The program must be compatible with and run smoothly on the school’s computer systems. It must also have a similar-looking interface to the general-purpose software the school uses.
3. The program must be able to produce a simple 30-second animation, using a grid of coloured squares to represent a given reversible reaction, in colours selected by the user.
4. The program must be able to produce a graph of concentration of reactants and products against time for a given reversible reaction, which displays in colours selected by the user, progresses alongside the animation and is plotted at the same rate.
5. The program must be able to correctly calculate Kc for a given chemical reaction and output the steps involved in its calculation, as well as displaying whether Kc has increased, decreased or not changed when a condition is changed.
6. The user must be able to compare two reactions side-by-side in the same window.
7. The user must be able to select colours from a colour palette to represent the reactants and products in a particular reaction.
8. The user must be able to print all data related to the reactions being compared, in colour or black and white, on A4 paper.
9. The program must have options to toggle the visibility of descriptive labels in reaction windows at runtime. The labels must be hidden by default and are shown or hidden immediately by pressing a toggle button.
10. Users must be able to enter chemical formulae using normal, superscript and subscript text.
11. The program must allow users to attempt to balance the reaction equation for themselves, by entering stoichiometric ratios for each reacting species.
12. The program must be able to calculate the stoichiometric ratios of a given reversible reaction.
13. The program must be able to compare its calculated ratios against those the user entered, and immediately display whether or not the user is correct.
14. The program must be able to produce a graph of rate of the forward and backward reactions against time for a given reversible reaction, which displays in colours selected by the user, progresses alongside the animation and is plotted at the same rate.

This is how well I feel the objectives were met:

1. The program saves and loads files in roughly half a second; much less than the two second target in the objective. Each file is also capable of storing up to five reactions, so this objective has been fully met.
2. I believe this objective has been satisfied because its interface is comparable to the GUIs of other software the school uses. The 32-bit build of the program works on the school computers without freezing or failing to respond, and there is a minimum of crashing so I believe that this objective is fully met.
3. This objective is not completely satisfied: the animation works, reflects the reaction data and uses the colours that the user has selected (see objective 7); however the duration of the animation varies depending on the computer used. This is because the animation is currently based on screen updates. The animation duration will stop varying if the time variable increments based on a standard milliseconds-based timer instead.
4. The program is able to produce graphs of concentration for any given reaction, and the graphs change according to catalyst data. Also, as seen in Video 6 the graph plots at the same time as the animation. However, temperature does not affect the graph at all. The graph not being complete means that the objective is not completely satisfied.
5. The program completely fulfils this objective; however the values given for concentration and Kc are much more precise than students are expected to be. Students only require 3 significant figures in their answers.
6. Reactions display side-by-side and update simultaneously. Users might want the two animations to start at the same time when a “Start animation” button is pressed but this wasn’t specified in the objective, so the objective can be considered fully met.
7. As shown in Testing, figures 4.2.18 and 4.2.19, pressing the “Set Reactant Colour” or “Set Product Colour” buttons open “Select Color” dialogs which set their respective colours, so the program completely fulfils this objective.
8. The user is able to print the left and right reaction screens and graphs. However this does not include the concentrations of each reactant and product as they are not included in the reaction screens, so the end user may not consider this objective met.
9. Toggling the “Show data” checkbox successfully hides text related to Kc as discussed with the end user, so this objective is fulfilled.
10. This objective was not solved ideally – it is more intuitive to use buttons to determine whether text is superscript or subscript as in most word processors. However, the PyQt QLineEdit classes used for user input only support plain text, so use of the “\_” and “^” characters was implemented instead.
11. The program, fulfils this objective; however, not completely: the values the program compares the user input to are not calculated by the program (see 12).
12. The program does not fulfil this objective as an algorithm could not be devised in time. This should be a priority in future development. Instead, the user must enter the stoichiometric ratios themselves.
13. The program completely fulfils this objective, except for the fact that the ratios are not calculated; they are provided by the user (see part 12).
14. The program is able to produce graphs of reaction rate for any given reaction, and the graphs change according to catalyst data. Also, as seen in Video 6 the graph plots at the same time as the animation. However, temperature does not affect the graph at all. The graph not being complete means that the objective is not completely satisfied.

5.2: User feedback

The end user made several comments on the program, after their first viewing since the prototype (section 2.9), which are shown below.

5.3: Feedback analysis

In future versions it may be useful to add a message to the installer, saying that program file folders should not be used (the program does not function in this type of folder because it does not have permission to read or write files there).

The animation algorithm needs to be tweaked to stop at the end rather than resetting. The code for the “Start animation” button should also be revised to ensure there is no delay between the button press and the animation starting.

A rounding system should be created to round Kc values to 3 significant figures. This can be done via Python’s Decimal module: to round a number x, the operation Decimal(x) + Decimal(0) can be used.

5.4: Possible extensions

* An algorithm still needs to be devised and implemented for ratio calculation. This could be done after further analysis of the balancing process, but a major part of it is essentially trial and error, which doesn’t lend itself well to algorithms.
* The formula regular expression should be modified to support full stops. For complex molecules it is common to add “.xH2O” as some reversible reactions are to do with water molecules becoming bonded to another molecule. Adding “(\.[2-9]?H\_2O)?” to the regex would allow this to happen.
* This regex should also be modified to allow square brackets, which can surround the more complex molecules. However these can be placed in a few different places in the formula, and the inclusion of the opening square bracket means the closing bracket must also be included. Adding support for normal brackets caused the regex to almost double in size, and adding square brackets would have a similar effect, making the regex very long and unwieldy, but it is possible using a similar pattern to the normal brackets.
* The animation code needs to be adjusted to include temperature. This can be done by adjusting the graph and animation the same way as catalysts do, but having a different effect depending on the Endothermic boolean.