Iteration Two Report

This iteration report covers the time between the hand in of the first Iteration on Friday Spring Week 10, to the final hand in Thursday Summer Week 8. It is intended as both a critical review of the product and the software practices used to produce it. It will contain:

* Discussion of change from JavaFX 2.2 to JavaFX8
* Review of Test Driven Development
* The user stories covered and how successfully they were integrated within eCook
* Analysis of how eCook meets the PWS (Project Wide Standards)
* A review of the contracted modules and integration of them within eCook
* A review of JIRA practices
* A review of version management using GitHub.
* Software managers review of the product

Changing Java Frame Work

Over the Easter break I made the decision to change the Java frame work again from JavaFX 2.2 to JavaFX8. This was done firstly to allow us to use new classes and methods that have been included within JavaFX8, which solved problems that would have been difficult to solve using JavaFX 2.2. JavaFX8 operates almost identically to JavaFX 2.2 and switching to it did not cause any significant problems other than having to download the latest Java JRE8. Ultimately this change should extend the working lifetime of eCook further as it now operates using the very latest version of Java.

Review of Test Driven Development (TDD)

Our QA manual states that where possible and appropriate we would use TDD to develop eCook. TDD was very successful for some aspects of project, particularly classes which involve large amounts of data manipulation in sequential programming context.

Problems however were encountered when trying to use TDD to test multithreaded code or testing GUI interfaces. The test frame we used (JUnit 4.0) is not intended to test multithreaded code or GUI elements, which resulted in difficulty when attempting to write tests for timed elements or GUI elements more complicated that using a button.

If attempting a another project I would encourage the use of a mocking framework which would allow us to more easily test the principles of the code using mock objects as opposed to trying to use actual class implementation but more importantly we should consider the objective of test driven development; that unit tests should be carried out in isolation of other code and that running tests of multiple threaded code breaks this premise.

User Stories

Iteration 2 covered sprints 3, 4, 5 which were detailed in the Sprint Plan document and the final sprint 6 which was intended to be only used for bug fixes and general polishing of the product. In the first Iteration report I stated that the deadline for sprint 5 was to be the end of Summer Week 3. This time scale was not kept to due to additional user stories that were added to sprints along with almost two weeks lost to the exams as nearly all group members devoted their time to revision. Adding these users stories and extending the sprint times has added significant value to the product.

Out of all of the user stories within the sprint plan only one was not attempted which was “Share links to recipes over social networking”. This was removed from the sprint plan due to other user stories which we developed while writing the program which addressed problems with eCook that had not been anticipated within the original design. The additional user stories were:

* Be able to pre fetch media which is available via HTTP in a recipe playlist
* Create an XML validator
* Scale slideshows to the computers resolution
* Produce the GUI

Pre fetching the media directly to the computer before displaying the slideshow is intended to speed up the performance of eCook when a recipe contains many instances of media content. Fetching the media can happen in parallel with using eCook and the software team felt this was an important feature to improve the users experience with eCook.

The XML validator was intended first to ensure that any XML playlist that is loaded into eCook would be compatible with it. Future versions of this feature are intended to be distributed independently of eCook to allow chefs to write their own recipes for distribution within the eCook online store.

Scaling slideshows is necessary as the positions of all content in XML playlists are defined in pixels. If the XML resolution does not match the screen resolution it is being used on, content may not appear correctly on the slide. This user story corrects these discrepancies.

During the design process we decided to let the GUI be driven by the features without any formal design, relying on only mock-ups of the GUI produced by Sam as a guide. As the project progressed it became clear that trying to have the GUI evolve out of the coding process without a formal direction was impractical and unlikely to produce the results we desired. Jon and I produced a new GUI design which was based on achieving a specific workflow within eCook leaving the details of the GUI content up to the programmer assigned to it. This method was also ineffective; the GUI was redesigned at least three times with constant tweaks meaning that at times that GUI was changing hourly.

A waterfall method of GUI design would have been more efficient, after agreeing the workflow we should have had new designs for the GUI drawn up and agreed to them before writing any more GUI code and a direct user story to design and produce the GUI should always be a priority on any future project.

Received Contracted Handlers

Both of the handlers we received (the Audio and Graphics Handlers) were fully working handlers in their own right but did not meet our design aim of having entirely encapsulated handlers which required a single constructor to display content. This later became less important however as we were later forced to break the encapsulation of our handlers in order to meet all PWS requirements

All of the handlers were refactored however to avoid the significant amount of repeated code which was present in the design of all of the handlers by using abstract classes and inheritance to provide all of the methods common to the handlers.

Project Wide Standards Review

Meeting the PWS has always been a priority within eCook. As with any standard the PWS process was rife with disagreements and I definitely do not consider it to be an optimum solution as I believe it tries to emulate the function of Microsoft PowerPoint too closely. The PWS that was agreed however does provide all of the functionality required for a functioning slideshow containing the five different types of media so in that regard I believe the PWS process was a success.

Once the PWS had been agreed I wanted to keep any changes to an absolute minimum. The only change we requested as a group was to define that the slides display order must be given consecutively in the XML playlist as this was not explicitly defined prior to our suggestion.

Changes to the PWS throughout the project have required the content handlers to be redesigned on at least two occasions but the result is a product that can accept a playlist containing all of the elements and attributes defined for a PWS playlist.

The only exception to this is the YEnd value for the Text Handler. Our implementation of the Text Handler uses the Text Flow class from JavaFX 8 which will resize itself to be larger than its defined size (which is defined by the XStart Xend YStart and YEnd values) if the input text string is larger than the box. This effectively renders the YEnd value irrelevant as the Text Flow will ignore it if the string is too large.

I believe that this is a better implementation however as I believe that the user would want all of the text that has been entered into the text string to be displayed and would not want it chopped off because the size of the box containing the text is too small.

Review of JIRA Practices

JIRA was intended to streamline the tracking of work on the project and to remove the need to provide separate documentation for each of the classes produced. The company’s use of JIRA was defined in the QA manual and all programmers were expected to conform to these standards.

Our use of JIRA for documenting the code was successful; all the user stories that were produced had a JIRA ticket assigned to them which contains a description of how it was implemented and the DTP (Deterministic Test Procedure) result. JIRA was also very useful for determining the work that had been completed at the end of each sprint and which user stories needed further work.

What was less successful was tracking the work done in real time; the main selling point of JIRA is that the board is a reflection of the work completed or being worked on. This only works however if JIRA is updated regularly and for the majority of the project this did not happen. Towards the end of the project (during iteration 6 especially) updates were made to JIRA much more regularly and the data on JIRA reflected the work done more accurately. I believe that using JIRA was a good experience for the group, there are many software companies that use it and similar technologies to track their workflow which meets the aim of the SWEng project to “gain experience in the procedures… through applying established industry practices”.

Review of GitHub Practices

GitHub desktop was the primary method of accessing eCook, which is widely used for group software projects. While it is a useful tool our groups experience with it (including my own) has caused a wide variety of problems and errors throughout the project due to incorrectly committing files.

Attempts were repeatedly made by myself, the development manager Jon and the management team to urge everyone not to commit files to GitHub which would cause problems, such as the .classpath files which were intended to be relatively so that eCook could be compiled on any computer but were repeatedly set by everyone in the group at some point to their own machines class path which would propagate to the other group member’s computers and break eCook.

This problem is mitigated through the use of branching within GitHub, which requires each programmer to take a branch of the code, produce some work and then merge their branch back into the master branch. However as using GitHub in a project of this scope was a new experience for almost all the members we decided not to use this feature as the merging of branches together can become very complicated and we feared that the problems caused by this would be more significant than simply committing the wrong files.

The group’s confidence and knowledge of using GitHub has increased substantially throughout the weeks of the project and were it to continue further the software team would begin to encourage the use of branches and merging to group members which would hopefully mitigate the majority of the problems that were encountered while using GitHub.

Review of eCook