Software Implementation Report

Iteration 2 Implementation Report

Document Control

Editor	Version	Date	Update
Alistair Jewers	**0.1**	**17/04/2015**	**Data Gathered**
Alistair Jewers	**1.0**	**02/06/2015**	**Document Formalised**

Table of Contents

Software Implementation Report

1.0 Introduction

Iteration 2 had several focuses:

- Development of the media handlers.
- Development of the sale code modules.
- Creation of the back end code to realise the LearnEasy application, including the run time data and renderer.
- Integration of code modules with the LearnEasy GUI.

One of the targets for iteration 2 was to be able to produce a working demo of the LearnEasy application (referred to as the 'Release 1 Demo' or just 'Release 1'). This required a model view view-model (MVVM) type system to be implemented in order to allow the application to utilise the data contained in the data structure created in Iteration 1 and render the pages of the lesson.

2.0 Target User Stories

- 'As a teacher I can include multimedia objects in the lessons I create'

This user story was satisfied by the creation of the media handlers, which allow media files identified in the XML to be displayed by the application.

- 'As a student I can view a lesson created by my teacher'
- 'As a student I can view all forms of multimedia that my teacher has included in the lesson'

These two user stories are again satisfied by the creation of the media handlers to display the media, as well as the creation of the renderer and runtime data classes to control the opening and displaying of lessons. Integration of the XML parsing functionality created in iteration 1 into the run time system was also key in allowing students to open lesson files.

- 'As a student I can interact with all suitable forms of media'

This user story mainly relates to the video and audio media handlers, but is also relevant to the answer box and multiple choice question handlers. In the case of video and audio, complex control was achieved through the use of the video.java and audio.java classes which are used within the handlers. Similarly the answer box and multiple choice handlers make use of extra classes to handle interacting with individual questions.

3.0 Features Implemented

Feature	Author(s)	Relevant Classes
Media Handlers		
Audio Handler	Alexander Cash & Calum Armstrong	AudioHandler.java, Audio.java
Video Handler	Alistair Jewers	VideoHandler.java, Video.java
Image Handler	Daniel Berhe & Jake Ransom	ImageHandler.java
Graphics Handler	Outsourced to WaveMedia	
Text Handler	Outsourced to WaveMedia	
Answer Box Handler	Daniel Berhe & Jake Ransom	AnswerBoxHandler.java, AnswerBox.java
Multiple Choice Handler	Emmanuel Olutayo & Penny Nicole	MultipleChoiceHandler.java, MultipleChoice.java, MChoiceRadio.java, MChoiceCheckBox.java
LearnEasy Application Back End		
Page Selection	Alistair Jewers	RunTimeData.java
Lesson Opening and Closing	Alistair Jewers	RunTimeData.java
Rendering Lessons	Alistair Jewers	Renderer.java, RenderUtil.java
LearnEasy Application GUI Code		
LearnEasy GUI	Lewis Thresh & Sam Hall	LearnEasyClient.java

Figure 1 – Features implemented in iteration 2

4.0 Timing

The software implementation portion of Iteration 2 took place between 16th February and 15th April 2015. Implementation ran to schedule allowing the planned release 1 demonstration to take place.

5.0 Review

The implementation portion of iteration 2 was successful overall. The core feature set of LearnEasy, including all media handlers, was implemented prior to, and demonstrated during, the release 1 demo. This involved integrating preview copies of the purchase modules in order to demonstrate the full range of media functionality. The implementation of a model-view-controller system was successful, with the data stored in the data

structure created in iteration 1 being correctly converted to on-screen components and media, and the interface controls being linked into the run time data fully, allowing control of the model.

6.0 Work Carried Forward

Considering the breadth of content in the iteration, bugs were to be expected. Therefore once again the fixing of outstanding bugs was carried forward to run in parallel with the third iteration. Similarly the process of refactoring and updating the XML components continued throughout this iteration and into the next.