



PROG3B POE

Change Log

Application: Dewey Training
Application ID: DT1.20

Version: 1.20
Student Name: Karl Dicks
Student Number: 17667327
Course: BCAD3
Subject: PROG7312
Lecturer: Nirasha Ramckurran
Assignment: POE
Due Date: 04/11/2020

Contents

Introduction.....	2
Task 1 Improvements.....	5
Task 2 Improvements.....	6
Conclusion	7

Introduction

As part of our PROG7312 (3B) module, we were tasked with developing a Dewey Decimal training application. My application has been built in Windows Presentation Foundation (WPF) in .Net Core 3.1. There were a number of requirements that the application had to perform, which have been implemented in my Task 1, Task 2, and POE project, in accordance with our set question paper.

Task 1 consisted of two parts: research into gamification features to be implemented in the practical part of the assignment, and the practical part of the assignment, being the reordering of Dewey Decimal books on a virtual shelf.

The requirements of the practical part of Task 1 is described below:

1. On start-up, the application shall allow the user to choose between three tasks:
 - a. Replacing books.
 - b. Identifying areas.
 - c. Finding call numbers.
2. For this first task, only Replacing books will be implemented – disable the other two options for now.
3. When the user selects Replacing books, the application shall randomly generate ten different call numbers, and display them to the user.
4. The application shall allow the user to reorder the call numbers, and the application shall check whether the user got the ordering right.
5. Implement the gamification feature that you identified to motivate users to keep learning.

Technical requirements:

1. Make use of a list to store the generated call numbers.
2. Choose an appropriate sorting algorithm to sort the call numbers to check the order that the user put them in.

These criteria were met for my Task 1 submission, and I have received feedback for this task, as detailed in the following section.

Task 2 only had an implementation component to it, which had the following functions / criteria:

1. Enable the Identifying areas task.
2. When the user chooses the Identifying areas task, they should be presented with a user interface where they will match two columns: call number (top level only) and description.
3. A question in this context is defined as the whole matching set, including both columns.
4. The user shall be allowed to answer as many questions as they want to.
5. The questions should alternate between matching descriptions to call numbers and call numbers to descriptions.

6. Each question should have four randomly selected items in the first column, and seven possible answers (three of which are incorrect) in the second column.
7. Implement a gamification feature to motivate users to keep using the application. You may use the same one as before or choose to implement a different one.

Technical requirements:

1. Store the call numbers and their descriptions in a dictionary.

These criteria were met for my Task 2 submission, and I have received feedback for this task, as detailed in the following section.

For our POE submission, we were tasked with improving our Task 1 and 2 submission, along with completing theory and practical parts as follows:

In a Word document, create a multilevel list showing the call numbers, for example:

- 700 Arts & Recreation
 - 750 Painting
 - 751 Techniques, procedures, apparatus, equipment, materials, forms
 - 752 Color

This was completed for 160 unique entries, and has been submitted in my Portfolio of Evidence. The references for this information were included.

The practical component of this assignment has been detailed below:

2. Create a file containing the data that was gathered in the research part of this task in a format that can be read by your application.
3. Enable the Finding call numbers task.
4. When the user chooses Finding call numbers, the application must load the Dewey Decimal classification data into memory from the file created in step 1.
5. The quiz must work as follows:
 - a. For each question, randomly select a third level entry from the data, for example 752 Color. Display only the description, not the call number.
 - b. Display four top level options to the user to choose between, one of which must be the correct one and the other three randomly selected incorrect answers. For example:
 000 General Knowledge
 400 Language
 700 Arts & Recreation
 800 Literature
 - c. For the options, display both the call number and description. Display the options in numerical order by call number.
 - d. If the user selects the correct option, show them four options from the next level, until the most detailed level is reached.
 - e. If the user selects the wrong option anywhere along the way, indicate this and then ask the next question.

Technical requirements:

1. Make use of a tree to store the data in memory.

These criteria have been met, by implementing the complete solution for the POE submission, as well as the research document.

Certain aspects of my Task 1 and 2 were improved, and have been detailed in the next section, in accordance with feedback received for both Task 1 and Task 2.

Task 1 Improvements

The following feedback was received for my Task 1 submission:

Dicks, Karl			Comments
	Max	Student	
Research: List of at least five gamification features.	5	5	<p>A well presented program. Well researched. Nice choice of gamification features. Concepts are explained well in the research doc. Readme file added and done well. Demo done via youtube link. Demo presented well with a voice over. Timer added. Good use of visualisation for the ordering of books. Program is fun nice use of colors and themes. A very good knowledge of programming concepts displayed. nice use of the DAL. Improve the conclusion and in text refs. Add a few more gamification concepts in theory that you think may also work.</p>
Research: Introduction and conclusion.	5	3	
Research: Motivation for choice of gamification feature.	10	8	
App Functionality: Allow the user to choose which task to	10	10	
App Functionality: Display ten randomly generated call	10	10	
App Functionality: User can change the order of the call	10	10	
App Functionality: App checks whether the user got the ordering correct.	10	10	
App Functionality: A gamification feature is implemented.	10	9	
App Logic: A list is used to store the call numbers.	5	5	
App Logic: An appropriate sorting algorithm is used to sort the call numbers.	10	10	
Coding Standards: Code is well structured and documented.	5	4	
Documentation: Readme file provides enough information to run the app.	5	4	
Other marks: App is easy to use	5	4	
Other Marks: Advanced features not covered in class (Bonus Marks).	5	0	
Total	100	92	
Penalty		0	
Percentage		92	

I have made the following changes to my Task 1 assignment:

- Introduction and conclusion have been improved in my Task 1 assignment.
- Motivation for choice of gamification feature has been expanded upon.
- Additional gamification features have been included in my Task 1 research.
- Gamification feature implementation has been improved by making it more clear to the user (background included for the timer).
- Coding standards improved by including additional comments.
- Application ease of use improved.
- Readme updated.
- Grammar corrected.
- Spelling mistakes corrected.
- In text references updated, for additional information included.
- Reference list updated.

The above changes have been made to my Task 1 assignment, in order to update its content and improve it based on lecturer feedback.

Task 2 Improvements

The following changes have been made to my Task 2 assignment, in order to improve on the original design, and functionality for the POE submission:

The following feedback was received for my Task 2 submission:

Dicks, Karl				
	Max	Student	Comments	
App Functionality: User can choose the new Identifying areas task.	5	5	<p>Very well put together program well researched and presented logins done although not required. Colours and themes set nicely. Nice dragging feature to order the books. Some menu options seem repeated. Readme file done well. Demo video done via YouTube. Link provided in readme file. Timing and scoring added. Identify the areas done well. Questions are randomised. Matching of columns work well. Program score as well as times as you. Program also gives the correct answers if you don't know them - nicely done. Scoreboard added. Readme doesn't have anything about the latest feature. Nice use of DAL & doubly linked lists. Changelog?</p>	
App Functionality: User is presented with a randomly generated match-the-columns question, with more answers than questions.	20	15		
App Functionality: The questions should alternate between descriptions to call numbers and call numbers to descriptions.	10	10		
App Functionality: User can complete the match the columns question.	10	10		
App Functionality: App checks whether the selected answers are correct	10	10		
App Functionality: App allows the user to keep practising.	10	10		
App Functionality: Gamification feature implemented.	10	10		
App Logic: Data stored in a dictionary.	10	10		
Coding Standards: Code is well structured and documented.	5	5		
Documentation: Readme	5	3		
Other marks: App is easy to use.	5	4		
Other Marks: Advanced features not covered in class (Bonus Mark)	5	0		
Total	100	92		
Referencing penalty		0		
Final mark		92		

- Readme updated.
- Application ease of use improved (timer easier to see).
- Coding standards improved by including additional comments and breaking the application into more methods.
- Changelog included in the final POE.

The above changes have been made to my Task 2 assignment, in order to update its design and functionality based on lecturer feedback.

Conclusion

In conclusion, the above-mentioned changes have been implemented in my Task 1 and Task 2 assignments, in order to improve their quality. Additional design improvements have been included in the practical submission, in order to improve its design, and make it look more professional and attractive for its users.

Debugging has been carried out, in order to address application bugs and issues which were not obvious in Task 2. These issues have since been resolved, and changes have been made for the final POE submission.

In addition to the above, the POE has included additional functionality as set out by the question paper, and the research has been completed.