Tharane Thamodarar a1787411 - Lydia Castellucci a1822174 - Edward Gilchrist a1831741

Project Specification - Major Practical Introduction

Our project is a maths based, interactive quiz game which requires a player to create and/or log in to their account. It consists of three levels (easy, medium, hard) upon completion they receive a reward such as a gold star (in the form of a picture in the terminal). Each level is a combination of multiple choice and entering answers in the terminal.

Design Description - Assessment Concepts

Memory allocation from stack and the heap

- **Arrays:** We will use a dynamic array to add in a user's login information and to store the quiz questions.
- **Strings:** Usernames, passwords, quiz/question description and user input/output for the authentication system.
- **Objects:** Authentication, quiz, levels and reward.

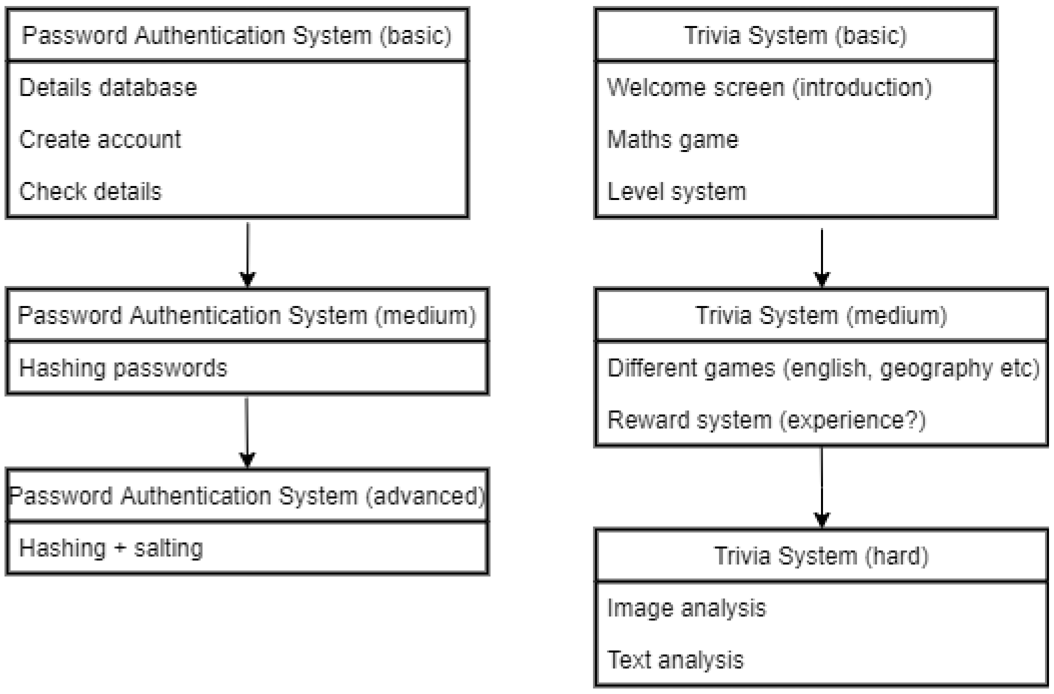
User Input and Output

- **I/O of different data types:** Command-line, the player will be presented with a prompt to log in or create an account. Then there will be a welcome page explaining how the quiz works. Following the welcome page, questions appropriate to level 1 will appear for them to answer either by entering a set of integers, doubles or floats and/or by choosing multiple choices (selecting between A,B,C).

Object-oriented programming and design

- **Inheritance:** The trivia system, there are different levels of questions within the quiz. Easy will be basic arithmetics in the single or double digits. Medium and hard will be scaled higher than easy but with similar types of questions from the easy level.
- **Abstract/Polymorphism:** The authentication system will have two pathways, one for login and another for creating an account. In this the user will be allowed to set or use their login details to access their quiz account. Which then stores their authentication details in dynamic memory, for future use.

Class Diagram - Class Descriptions


Password Authentication System (basic)
Details database
Create account
Check details
Password Authentication System (medium)
Hashing passwords
Password Authentication System (advanced)
Hashing + salting
Trivia System (basic)
Welcome screen (introduction)
Maths game
Level system
Trivia System (medium)
Different games (english, geography etc)
Reward system (experience?)
Trivia System (hard)
Image analysis
Text analysis

Authentication System

Abstract class. The authentication system works in the way that it compares and checks the user's information such as username and password from the records. The system has two behaviours which is logging in or creating new accounts. In the above class diagram there is a pathway for the initial plan and what can be added if time allows. The current aim is to have a system which has a database with users account details, which will then be checked to ensure the username and password given matches the initial information. If not there will be an option to create an account which will have certain security criterias the user has to meet when creating a username and password, at the end of the level they may choose to exit/log out or finish all levels.

Trivia System

This will be an inheritance class. This system is the quiz game section, it has three levels easy, medium and hard. At the end of each level the user may choose to log out or continue and upon completion of the quiz they will receive a gold star. This class will have a scalar type method to increase difficulty in different levels. Will also incorporate a combination of multiple choice and typing answers in the terminal. Which will then compare the entered answer against testing and output a positive message if the answer was correct and a negative message if the answer was incorrect. If they answered incorrectly the system will simply ask another question.

Rewards System

At the end of the quiz if the player has successfully passed then they will be presented with a gold star in the terminal. It is a simple class which works with the trivia class.

User Interface

A user of this program is a player intending on putting their math skills to test. All their inputs and instructions will be done via the command-line. In this quiz-based game there are three components. The steps are authentication, quiz, and reward.

```
Welcome to Maths Trivia!

Please select from one of the following options:

1. Join as new Guest
2. Login as Current Guest.
3. Quit

Enter a number from the options above and press Enter:
```

and enter the number for the corresponding option such as '1' and it will take them to the quiz information and welcome page.

```
Welcome to Maths Trivia!
You will be given three levels and feel free to exit anytime by entering 3.
Each question will be basic arithmetic. If it is a multiple choice then choose from A, B or C. If it is a question in which you have to enter an answer, ensure to use numbers. For either type enter your answers in the terminal then press enter.

Enter 4. To begin, the quiz.
```

Once they enter 4. They will be presented with questions

Level 1

```
What is 10x10 =

a.      120
b.      100
c.      110

More questions………
```

Level 2

```
What is 100x10 =

    d.  1200
e.      1000
f.      1100

More questions………
```

Level 3

```
What is 1000x10 =

    g.  12000
h.      10000
i.      11000

More questions………
```

Once completed all questions

They will be directed to….

```
You have successfully completed this quiz. Please wait for your golden star to unlock the next category of quiz, which will be accessible in Study Prep Season.
```

Depending on what is achievable do a star or certificate with star borders.

Code Style

- Classes will begin with a capital letter. Class files (.h and .cpp)
- Testing files will be labelled as classname_test.cpp
- Function comments will be given in all files. Clear description will be provided in the header and testing files.
- Variables will have a relevant name to its use.
- All code is done locally then submitted to the remote GitHub Repository.
- Code will be indented to 2tab spaces after the initial starting column.

Testing Plan

All our tests will run through our makefile each time we compile our code, automatically and

Makefile:

```
file: securityDetails.h securityDetails.cpp createAcc.h createAcc.cpp logIn.h logIn.cpp Quiz.h Quiz.cpp mathsQuiz.h mathsQuiz.cpp main.cpp

	g++ securityDetails.cpp createAcc.cpp logIn.cpp Quiz.cpp mathsQuiz.cpp main.cpp -o quiz

	./quiz
```

OR

```
all: main

Quiz.o: Quiz.h Quiz.cpp

	g++ -c Quiz.cpp -o Quiz.o


MathsQuiz.o: MathsQuiz.h MathsQuiz.cpp

	g++ -c MathsQuiz.cpp -o MathsQuiz.o


main: Quiz.o MathsQuiz.o main.o
```

```
	g++ Quiz.o MathsQuiz.o main.o -o main


clean:

	rm -f *. o main
```

The test files will be named as Classname_test.cpp, the appropriate test files will be included in the corresponding makefile commands.

Unit Testing

Our unit tests will cover all public functions in our classes. Each class will have a corresponding test file like ClassName_test.cpp, which will include a main method that will exhaustively test each function with a set of inputs and test if it matches an expected output.

For example:

```
	TriviaSystem_test -input.txt:
	                      100
	                      1000



	TriviaSystem_test - output.txt:

	                     100
	                     1000
```

Schedule Plan

Stretch Goals - Meet at the 1pm to 3pm prac session on Fridays.

Week 8 - Preparing for assessment in Week 9

Check-list:
Develop project plan
Organise code sharing with group members.
Assign tasks to group members.

Break Week 1

Create SVN structure. - Tharane
Write a makefile. - Lydia

Break Week 2
Finalise testing strategy with make files and input/output files outlined in the testing plan.
Finalise basic skeleton of authentication system. - Edward

Week 9
Upload design document to svn. - Tharane
Present skeleton code, makefiles and design. - Tharane and Lydia
Pseudo code for trivia system. - Tharane and Lydia
Testing - Edward

Week 10
Math component quiz – Lydia
Another quiz type – Tharane
Updated authentication system – Edward
Testing for maths quiz – Lydia
update testing for authentication system – Edward
Combine Authentication and Maths trivia with Landing page - Tharane

Week 11
Finalise the project: Authentication system – welcome page - Math quiz - Another quiz – final certificate of pass or fail with stars.
Run everything from make file.
Consolidate all code onto University SVN and GitHub for access.

Token - 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