Flazz Game

James Blankenship and Vrushank Mali

Abstract

We are writing about the group project for CSCI 352. The project will be about a short flash card game. It will also give information about a topic.You will be able to make quizzes and add them to a database for storage. You will be able to later edit the quiz.

1. Introduction

The project will be a simple flash quiz game. The group will be trying to make an application that will work as a quiz that will shuffle through questions. The project will have a decent amount of questions and when you choose an answer it gives you information about the topic of the question.The questions will come from random topics and . The target for the project is people wanting to take quiz or find out information about a topic. The target audience will get information about topics when using the program.The target audience will be younger people so it will have likely simpler questions.The ability to add quizzes will like broaden the audience age diversity.

1.1. Background

There are not currently any terms that need to be explained. The reason for this idea is we wanted something information based and after running through some ideas decided on this on.The reason for deciding on this game was that it is a very broad program concept.For example the quizzes could have a lot of diversity in the question.You could have a quiz on history and another on music. The game will be a general knowledge game. The game will have a timer as you go through the questions. The quiz

1.2. Impacts

There are not a lot of impacts for the project besides the grade for the project. If there was a impact it would be the learning that would come out of it.The impacts could be that people learn new information.

1.3. Challenges

Classes would be them main problem getting them all connected and dealing with the database.Certain functions such as a timer might be a difficult thing to figure out. The easier part of the project will likely be getting the overall design of what we want down.If we get to the stretch goals use links to give information might be a challenge.

2. Scope

The scope of this project will be having a program that has multiple hard coded questions that will tell you if you got the question right. It will also tell you information about the topic in question.The stretch goals would likely be adding links, but we do not fully know how that works.The other stretch goal will be adding more questions.

2.1. Requirements

2.1.1. Functional.

- User needs to choose difficulty.
- User needs to have the ability to answer questions, and get information on topic.
- User needs to be able to make their own quiz.
- User needs to be able to store their quiz in a data base.
- User needs to be able to keep track of how many questions they got right.
- User needs to be able to edit the quiz they have added, or remove it.

2.1.2. Non-Functional.

- Users quiz must be stored in the data base.
- Users should be able to see the interface and interact with it.
- The quiz should stay in the database until removed.

2.2. Use Cases

Here are some brief examples of how the game should work.

Use Case Number: 1
Use Case Name: Player starts a new game
Description: A player starts the game in easy difficulty. Player will click on ”Easy” button. This will load the easy level game page with first question and options for answer on the screen.

1) Player loads the game which will load the start menu on the screen.
2) Player will left-click on the ”Easy” button.
3) The easy difficulty game will load on the screen with first question and the options for answer on the screen with a timer.
4) Player will choose one of the options for the answer by left-clicking on the button.
5) If the selected answer is correct, it will turn the answer button ”green” and will show a dialogue box with more information about the topic which was in the question.

Termination Outcome: If the timer ends for first question, it will skip the first question and will show second question.

Use Case Number: 2
Use Case Name: Pause
Description: If the player wants to pause the game, player will click on the pause menu and it will pause the timer and will show the pause menu.

1) Player will left-click on the ”Pause” button, which will open the pause page on screen.
2) Clicking pause button will pause the timer and will show the pause menu.
3) The pause menu shows ”Resume”, ”Restart”, ”Volume”, and ”Exit” options for the player to choose.

Use Case Number: 3
Use Case Name: Restart
Description: If the player wants to restart the game, player will click on the restart button and it will restart the whole game with same difficulty and reset score and time.

1) After game getting paused by the player, the player left-clicks on the ”Restart” button.
2) This will restart the whole game with the same difficulty level, reset score and time.
3) The player continues to play the game again.

You will then need to continue to flesh out all use cases you have identified for your project.

2.3. Interface Mockups

These are screen shots we made as interface mockups.

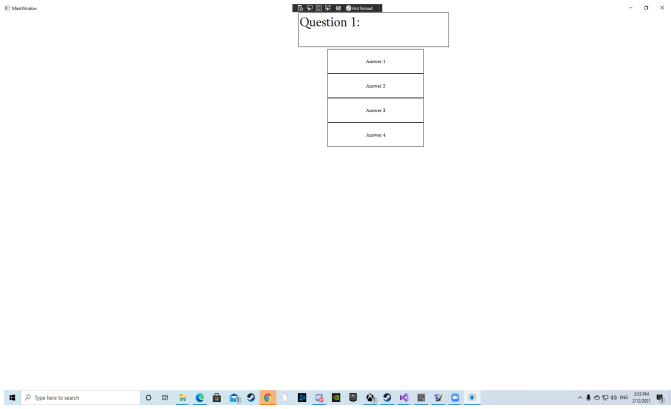


Figure 1. Interface 1

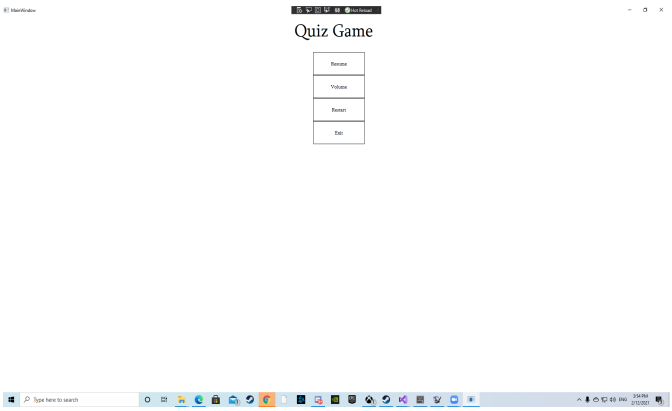


Figure 2.

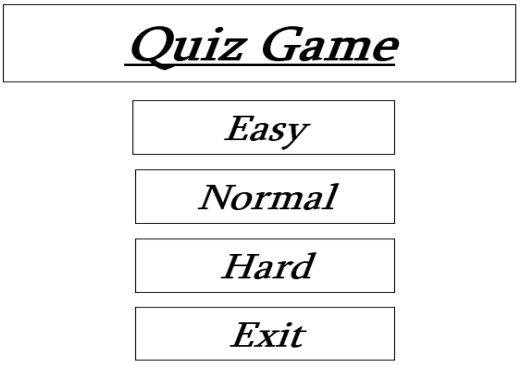


Figure 3. start menu

3. Project Timeline

- Project Proposal Draft 02/01/2021: We got the basics of the project with topic and general idea. Also some of the features it will have.
- Project Proposal Update 02/12/2021: We updated the non functional and functional aspects. We also talked about use cases and requirements. Project Interface presentation 2/16/2021: We talked briefly about the overview of the project and looked at the interface mock ups. Dr. Guerin gave us feedback and some ideas for the project.
- Project Update 4 3/19/2021: We have started on the structure and project time line.We have also started a UML following the project on the project structure section.
- Interfaces and Elements 04/01/2021: We will have most of the interfaces and elements ready.
- Database 04/06/2021: The database will be ready for use.
- Finished 04/20/2021: The product is finished and ready to be demoed.

4. Project Structure

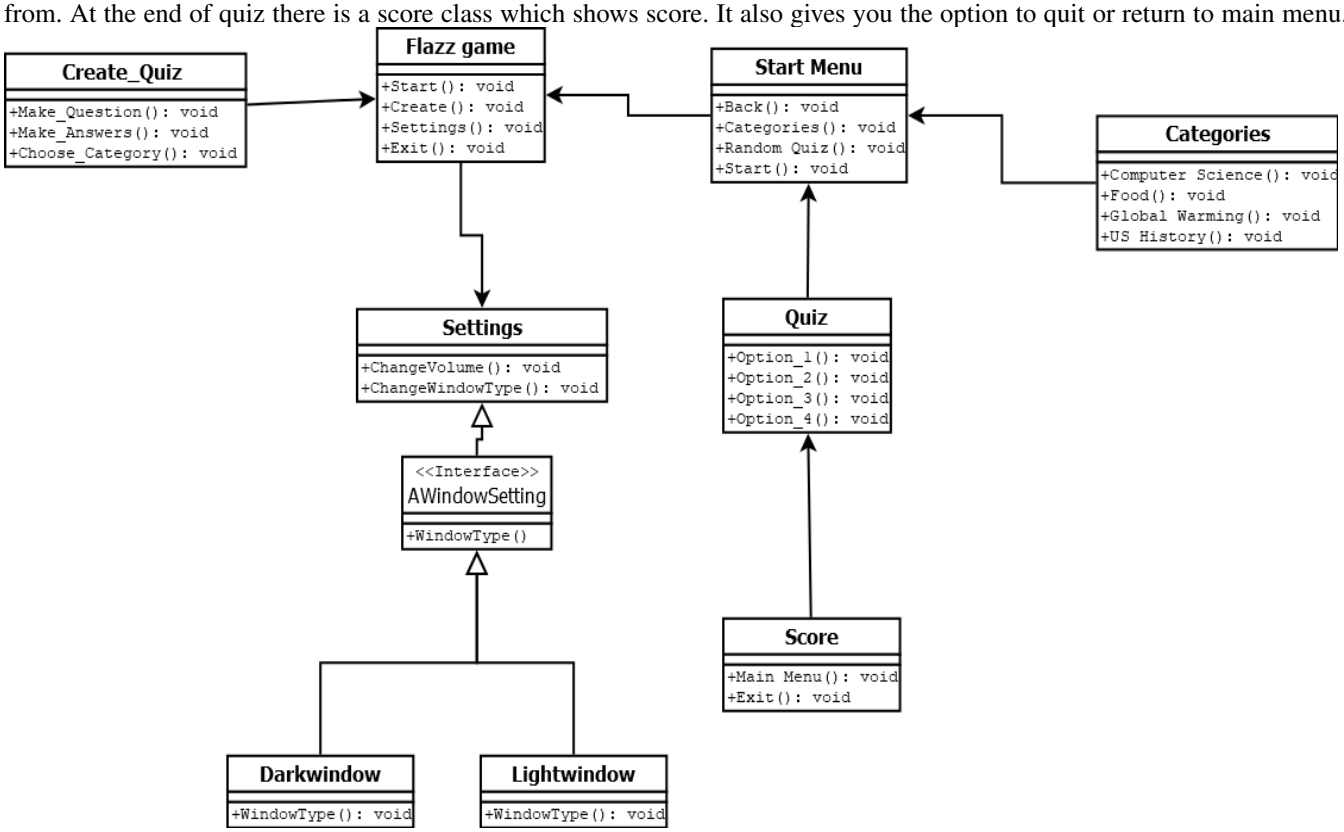
The project will have default question that will be loaded into a data base. The database is set up to have categorizes organized under an ID.It would be connected to a questions table.The table would have a topic ID,and the actual question.The answers table would have a question ID to know which questions to go to.It would also have a answer section to hold the answers,and a correct field to see if they are correct.The Quiz bridge would organize the questions together for a quiz table.The quiz table would just have the name and the ID which would connect to the quiz ID in the bridge.

The GUI is set up at the Main menu then allows you to go to settings,start menu,make quiz and quit. The start menu allows you to choose what type of quiz you would like and then allows you to start the game. Then each of the pages has option to go back to the main menu.The settings tab has the options to allows you to have a light mode, dark mode, and a mute button. The quiz interface has a text box that has a question on it,and buttons that have answers on it.

Users will be able to add questions at a later date There will be about 4 categories to choose from to make the quiz on.There There will be options to change how the quizzes look to get an example of abstract factory. The main menu will have options to start, settings, create, and exit.The start menu will have categories and difficulties to choose from.There is also a random quiz option.

4.1. UML Outline

The UML starts in the Flazz Game class. If then leads to the settings class which has an option to change how the window looks.which leads to a interface for those two classes.The start menu class has options for choosing categories, going back to the main menu, making a random quiz,and starting a quiz.The categories class gives you a list of categories to choose from. At the end of quiz there is a score class which shows score. It also gives you the option to quit or return to main menu.



4.2. Design Patterns Used

One of the patterns that is going to be used is the abstract factory method. It will be a light and dark mode for the windows. It will be found in the settings menu in the program. The other design pattern will be the composite pattern. It will deal with the buttons and how they work on the quiz since clicking one will affect all of them in some way.

5. Results

The project has most of the GUI done the database is almost done. All that needs to be done is allow the user to make a quiz and hook up the data base to the quiz section of the project. Most of the questions for the quizzes were added to the database. We lowered the amount of categorizes and removed difficulty.

5.1. Future Work

The next thing we are doing with the project is finishing up the database, and adding the logic so it will display the quiz.The program will need to add the ability for a user to add a your own quiz.Then add a random quiz function if we have the time.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.