BERKE ÖZDEMİR / 2016400246

İBRAHİM ÖZGÜRCAN ÖZTAŞ / 2016400198

CMPE 230 PROJECT 3

May 15, 2019

Overview

This projects helps us to create, manage and distribute applications using QT Programming Framework. In this project concept, we are requested to create a mini-game, which is Find The Pair. We used to play this game in our childhood. Now, we coded it, and it gives us a great joy.

Goals

1. **Understanding the framework:** We are requested to use QT Framework, so our first objective is to understand and be able to manipulate the framework as our wish.
2. **Understanding the target:** In second goal, we are requested to understand and compartmentalize the issue. We divided it to separate parts to handle each task well-worked and efficiently. Thus, we unite our tasks into the former issue.
3. **Implementing the tasks:** We firstly implement the user interface of the project, that allows us to have a clearer insight of the main structure. Then, we added functionalities to the objects we implemented in user interface. Furthermore, that brings us to handle bugs. In the end, we've coded the whole project from scratch.

Specifications

In the project, we used three classes, with their header files and ui implementations.

1. InitialScreen
2. MainWindow
3. AfterScreen

After using three classes, we've linked them as in order such as;

InitialScreen -----> MainWindow ----> AfterScreen

The game board resides in MainWindow, the others are developed for improvements of overall usage.

Milestones

1. Implementing InitialScreen

   In InitialScreen, we've two buttons that has common signal clicked() and separate slots for each. The Start button has a slot that links the mainwindow instance to the initialscreen instance. After pressing that button, the initialscreen hides and the mainwindow appears on the screen. The second button, exit, does the quit operation from the game itself.

2. Implementing MainWindow

   In MainWindow, whole game structure resides. The class fields are constructed to handle several operations in the functionality. All necessary information is collected via those class fields. Then, the slots are coming into the operation. On the game board, there is 25 buttons, 24 of them holds cards and 1 of them is reset button. The cards are closed at the very beginning, thus we gave the user no hint about the card layout. After pressing a button, it is turned on and can't be clicked once more. After that, user presses another button, which is a card, and it turns on as well If those two cards are same, the user gets 1 pair. If not, then cards get turned off and user gets no pair.

In the end, user has tried to find a pair, thus tries increases by 1, whether it is a success of failure. After 12 pairs found, the afterscreen appears and tells the user that the game is finished.

At some point, user may restart the game via reset button, thus the button resets all the variables that holds pairs, tries and remainingcards. Then shuffles the cards and links the button again. However, if user finishes the game successfully, then user should restart the game again, due to lack of play again button.

3. Implementing AfterScreen

After finishing the game, user will face a third screen that shows the information about game ending. There is one button in the window that enables user to exit the game. User may quit via pressing the X on the upper side of the program window, or by exit button if finishes the game.