C++ Programming Mini-Project – 5

Old Maid

Practical Work

By using Microsoft Visual Studio 2010, write a C++ object-oriented program to implement the game Old Maid (潛烏龜). There are 4 players in the game and one of them should be a human player. A deck of cards are evenly distributed to the 4 players with one player having 1 card less (which will lead to one "odd" card). The "odd" card is kept unknown to all players until the end of the game. At the beginning of the game, the players discharge all the pairs they have in their hands. When there are no more pairs in each player's hand, each player draws one card from another player one by one. After revealing the drawn card, if it forms a matching pair with a card in his/her hand, the player may discharge the pair; if not, the player has to keep the card in his/her hand. When there is no card in a player's hand, he/she is out of the game. The game ends when one single player is left with one card in hand. That player then loses the game. The "odd" card must then be revealed to everyone, which will form a matching pair with the last card the loser has in the game.

One of the following 5 rules has to be implemented by you. The rule number should be determined by the remainder obtained from dividing your team number by 5. For example, if your team number is 6, 6%5 = 1; you should follow rule 1) of this project.

0) A pair of joker cards is included and the "odd" card must be one of the jokers.
1) A player can draw again from the same player if a matching pair is found.
2) When a matching pair is found, the player whose card was drawn cannot draw card in this round.
3) The drawing direction reverses when a matching pair is discharged.
4) At the beginning of each game, the first person who gets rid of all cards in the last game can give away 3 cards to any player(s).

Reference:

- http://en.wikipedia.org/wiki/Old_Maid

1. Your program needs to be designed with a graphical user interface (GUI).

2. While you are responsible for the final design of the program, it is required that the code governing the rules of the game should be implemented with a separate static library developed by native C++ and linked into the managed C++ application. The managed C++ code you write should mainly be responsible for the GUI.

3. Before a user starts to play, he needs to login the system first. A file is then created to store the result of this user after playing the game, e.g. how many times the player loses and wins the game. If this user has played this game before, his/her record should be retrieved and shown on the screen. After he/she has played the game, his/her record should then be modified and stored.

4. Should you want to get a credit, you should design the game with some advanced features as follows.
   - You can add a timer to the game so that the amount of time that the player uses to finish the game will also be recorded down.
   - You can add different levels of difficulties by restricting the amount of time the player can do the pairing. If a player cannot do the pairing within the given time, he has to do the pairing during the next turn instead of drawing a card from another player.
   - You can show some animation for the card movement.

5. Should you want to get a distinction, you are free to add two more features to the game that will make it to be more interesting. You may seek the opinion of your subject lecturer/tutor before you

do so. In principle, it must be some new features that require additional efforts but not a repetition of work.

Report

Your report should include:

1. **Abstract**: An abstract of less than 200 words that summarizes the objectives and achievement of your project.

1. **Introduction**: A detailed description of the objectives and requirements of the project, and a brief description of the methodology.

2. **Methodology**: It contains
   - How your group divides the work among the group members (very important, to be used as the basis for assessment)
   - The schedule of implementing the project
   - The structure of the program developed, including
     - The specifications of the classes defined, and the public/private member functions/variables inside - explain as far as possible why your group makes such choices
     - The flow of execution. (It is good to include a flow chart to illustrate it.)
   - What problems your group encounters, and how your group tackles the problems
   - Testing of your program, which shows
     - How you validate your program, i.e. confirm that the solution is correct.

3. **Results**
   - Include the results of executing your program captured from the screen.

4. **Conclusion and future development**
   - Summarize the experience gained in the mini-project
   - Indicate how your program can be extended and improved if more time is allowed.

In addition, a user manual of your program should be attached to the report as an Appendix to illustrate the usage of your program. It may include, for example, the screen shots of the GUI you developed when running the program.

The report should be in PDF format. It is NOT required to include the complete source code in the report. Rather, you should copy the folder(s) containing all your project files into a CD, which also stores the report. (See the General Description below.)

General Description

1. Each group should comprise 2 students. Students need to obtain prior approval from the subject lecturer if they want to form a group with fewer or more group members.

2. After finishing the project, each group should hand in one CD that contains the following:
   - The soft copy of the report (including the Appendix of the user manual)
   - All folder(s) and files of your project
   - A readme file (`readme.txt`) that tells us how to build the project and, if any, other important (IT) requirements for running the program.

3. You have to make a presentation to demonstrate and explain the details of your work. In the presentation, each group member should declare his/her responsibility in the project. Each member will be individually assessed based on the declared responsibility and the result obtained.

4. The documentation for your mini-project is a very important part. The ability of writing good comments will also be an important factor to the final assessment of your mini-project.

5. It is compulsory to use a word processing tool to write your report. The font size must not be bigger than 12 or smaller than 10. Print your report in 1.5 lines spacing on both sides of a page. Including all diagrams and tables, if any, the length of the report should not be shorter than 15 pages.