CIS 153 Final Project

200 Points

Distributed: March 27, 2023

Due: April 23, 2023

Upload Instructions:

1. You need to submit the source code for all of the programs in the homework assignment (these files are the ones with the .cs extension). The easiest way to make sure that you submit all of the required files is to zip the entire visual studio project folder. All source files need to be clearly labeled (Final\_ProgramTitle)
2. You need to submit a word file that states how to use the program and what the program does. This should also tell me any issues that you ran into and things that you were unable to complete.
3. These two files need to be zipped into a folder that is clearly labeled and uploaded onto Brightspace (CIS153\_Final\_Group1).

Coding Rules:

1. Source code needs to be properly formatted. This means that all brackets line up properly, indentation is consistent, comments are included, comment header included, and variables have meaningful names.
2. Comment header must be at the top of your program and must state your name, the date, and a description of the program
3. Comments are used at your discretion but having no comments is not acceptable.

**Program 01 Connect Four**

You are required to write a connect four program as part of a group. This program will have at least 4 different forms (possibly more). The first form is the welcome form that will display 4 different actions, single player, two player, statistics and exit. Single player will open a new form that allows the user to play against the computer. At a minimum you need to create a rules based AI. Do not overthink this! A rules based AI can perform the following rules: if I can take a win take a win, if I can block you from winning block you, otherwise follow some basic strategy. Two player will open a new form that allows the player to play against another person. Statistics will allow the player to see their record against the AI. Exit will exit the program. Please note that I want an exit button on each form that closes the entire program not just a single form. This can be accomplished by using the form exit button in the top right if you would like. You can make sure this is happening by running your program in visual studio. If you click the exit button but it is still running you will see the red stop button in visual studio. When the game board is up we should see a label at the bottom telling the user who’s turn it is as well as an exit button (if you choose). We should also see the connect four grid and any moves that have been made. If the user hovers their mouse over the column they want to play it will display what the move will look like, the move will not take place until the user clicks in that column. Once a move is made we should check for a winner. If there is a winner or a draw I want you to display a new form. This form will say who won, show the players statistics against the computer, have an action to play again, have an action to review the game, and have an action to exit. Play again will close the current form stating who has won and open a new game board form. Review the game will allow them to see who won and how they won.

**Additional Functionality:**

* A rules based AI
  + Make an AI that the user can play against.
  + The AI must have 3 rules. If you want to research different algorithms for your AI to follow, that is perfectly fine but it is not required. (Minimax algorithm would be the easiest to implement. However the sample size for this algorithm is too large so you would have to optimize). Please do not just copy someone else’s AI algorithm for this.
  + Having your AI randomly pick an open cell to play will result in no credit for the AI portion of the points.
* Player Statistics
  + The winner form will show the players statistics against the computer. The welcome form will have an action that will allow the user to see statistics on a new form.
  + The statistics should show how many times the player has won, how many times the AI has won, and how many times it was a tie. It should also show the total number of times the game has been played and win percentages for both the AI and the player.
  + This needs to be persistent data. Therefore, when the game is closed and ran again the player can see their stats from all previous runs. This can be accomplished by using a simple text file. You do not have to make this work for multiple players.
* Classes
  + You must include a cell class and a board class. You must have private members and public functions. The functionality and design of these classes is determined by you. However, I will be grading you on you implementation.

**Notes:**

* There should only ever be one form open at a time unless the user chose to review their previous game board.
* I want all forms to be opened in the same location. I do not want them to open randomly on the screen. (hint: use .top and .left to tell a form where to load)
* Once the game is over if the user chooses to review the board they can no longer interact with that specific game board.
* Getting the close and open for different forms to work properly will be a challenge. (hint: using the keyword this, using hide, using show, using Application.Exit(), and a form close function will help)

**Point Breakdown**

Listed below is the functionality needed to complete the assignment. Getting all of the functionality will result in full credit. Getting partial functionality complete will result in partial credit given at my discretion.

|  |  |
| --- | --- |
| **Functionality** | **Points** |
| Main menu options | 10 |
| Exit completely closes the application | 10 |
| Winning conditions | 20 |
| Game over form options | 10 |
| Hovering functionality | 10 |
| Restart game functionality | 10 |
| All forms load at same position | 10 |
| Only one form open at a time | 10 |
| Review game functionality | 20 |
| AI functionality | 20 |
| Player statistics | 20 |
| Cell class | 15 |
| Board class | 15 |
| Creative GUI | 10 |
| User experience | 10 |