Create a C++ Console application bases on C++ Classes/Objects, OOP, Exception Handing, IO, … to complete below exercises.

1. Business needs

* Create a C++ console game Caro (10x10).

// Example Main Menu screen and Play with BOT screen:

\* ------MAIN MENU-------\*

Press number to choice:

1. Play with Other Player

2. Play with BOT

3. Replay

4. Player's Infomation

5. Guide

6. Exit

\*------Play with BOT-------\*

Press number to choice :

1. Easy Mode

2. Normal Mode

3. Hard Mode

4. Back to MAIN MENU

* Example how to play:

// Console:

=> Player 1 = X, Player 2 = O:

0 1 2 3 4 5 6 7 8 9

+---+---+---+---+---+---+---+---+---+---+

0 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

1 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

2 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

3 | | | X | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

4 | | | | | | O | | | | |

+---+---+---+---+---+---+---+---+---+---+

5 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

6 | | | | | O | | | X | | |

+---+---+---+---+---+---+---+---+---+---+

7 | | | X | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

8 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

9 | | | | | | | | | | |

+---+---+---+---+---+---+---+---+---+---+

Player 2's turn: 3 3

// End Console

Player 2 enters 3 3, saves the moves and prints the new checker status as image

1. Requirements

R1:

* Allow users to enter moves by command lines.
* Show current status of the board.
* Check Win condition and report result.
* Continues until one of two players win or draw.
* Winning condition: when a player has 4 moves in a row horizontally, vertically or diagonally, he wins.

R2: Create bot for game with 3 levels: easy, normal, hard (referece minimax algorithm).

R3:

* Create function replay games, watch replay. Players can review the game history, see each move by controlling the previous/next move
* Create file (.txt, .ini) to manage player, save information of each player: name, number of wins, number of losses, number of draws.
* Create function search player by name, must print out the player information and when making a player must find someone with "level" equivalent to that person, i.e. the person with the most odds of winning or losing to that person.

1. Working requirements

* Working environment: Visual Studio, Code Blocks, Eclipse, Net Beans.
* Delivery: Source code (\*.h, \*.cpp).

1. Technical Requirements

* Use Object-Oriented programming style.
* Apply the default C++ naming coding convention.
* Remember to comment each class, method, and attribute (if needed).
* The application should be on the console.
* Use I/O Streams for working with files.

1. Stored Data

All data information will be saved to text file (if required).