CS-150 2023

2nd Assignment
10/03/2023 - 24/03/2023

Part 1: Player [25%]

Implement the class **player.** Class player will need to have the following variables

- Name: the username of the player
- Alive: if he is still alive in the game
- Role: his role, gangster, doctor, citizen
- CurrentVotes: the number of times other players have voted this player in the current trial

For all the above variables implement setters and getters functions.
Also, implement a function vote(), which will compute the process of voting for that particular player.

Finally, implement two constructors with the following signatures

- Player(string Name , bool isAlive, string Role)
- Player()

Note: the above functions and variables is the necessary functionality that we ask, you can expand this class as much as you need to better improve your implementation.

Part 2: Rounds [15%]

Implement the class **Round** that will store the information of the current round.
Particularly it must include:

- Which player left due to the gangster (empty if that player was saved from the doctor).
- The number of the current round (1st, 2nd etc...)
- Which player left due to the trial (empty if no player left)

At the end of each round, you will have to store the above information in a new instance of the class round.

Note: the above functions and variables is the necessary functionality that we ask, you can expand this class as much as you need to better improve your implementation.

Part 3: Player & Rounds Usage [25%]

Improve the implementation of assignment 1 using these new classses.

Some exmaples,

- Wherever you are printing player IDs, print also their usernames
- Instead of using vector<int> with the ID of each player to hold the current players use vector<Player>
- Use the function vote for the voting of each player.

The above are examples. The scope of this part is for you to identify as many application of the above classes and to use them as much as possible to improve the initial implementation.

Part 4: Multiple Files [10%]

Implement classes Player and Round in separate .cpp files. You can have a global header file with all the declarations or multiple header files each corresponding to a different .cpp. Finally, you will need to have three .cpps files and as many .h as needed. *{see deliverables}*

Part 5: Input [15%]

Your program will to be initialized from a file. At the beginning of the game, you will to open a file with the name "Players.txt", in which at each row there will be the username of each player (player1, IamnotTheGangster, Winner99 etc...), and their role, separated by space.

Note: you can assume that this fille will always have the correct format and not have duplicate usernames. In addition, there will always be exactly one doctor and one gangster.

Part 6: Output [10%]

At the end of the game you will have to create a new file "TownOfSalem_output.txt" in which you will write the information that you saved on each round in the instances of the class round.

Deliverable

Submit all the .h files that you used, **three** .cpp **files** (example names: Player.cpp, Round.cpp, Assignment2.cpp) and the CMakeLists.txt that produces the final executable. Add all the above files to .zip archive with name Assignment2_AMXXXX.zip where XXXX is your student id.