NIMBUS

Our project is based on the mathematical game of strategy called NIM where two players take turns to remove objects from distinct stacks and the player drawing the last object or simply clearing the last remaining stack is the winner.

Our project is a modified version of the game of NIM . In the game of NIMBUS a player is allowed to draw only one or two elements from any stack in turns. The player clearing a stack is rewarded a point and the player with the most points is regarded as the winner.

We have divided our project into two segments, Main Game and Leaderboard and GameLog.

Our game has 4 game modes. The first one is the traditional nim game and Under NIMBUS we have 3 different game modes which are regular NIMBUS, vs Computer and Special NIMBUS.

REGULAR NIM : As stated before

In case of NIMBUS: the rules are quite similar to that of the traditional nimbus but here a player is allowed to draw only one or two objects during his/her turn . A point is awarded to the player who clears a stack. And the player clearing the most number of stacks is the winner.

AI : vs Computer AI

SPECIAL : Similar to that of regular NIMBUS, but in this game mode each player is allowed to make a special move where he/ she can draw any number of objects from a stack once.

The coding approach for the different game modes are quite similar, so here I'll be explaining the case of Regular NIMBUS.

In the game of NIM a player might get an advantage if he makes the first move. So, in order to avoid this, we have used a TOSS function which will randomly choose which player will make the first move.

To initialize the stacks, we have randomized the number of stacks and the number of elements in each stack. To ensure that there's always a winner , the number of stacks will be odd for all game modes and the number of elements in each stack will be within 5 to 10.

In the game, at first , the name of two players will be taken as input. Then the TOSS function will be used to determine which player will make the first move.

Then we will implement each move using a loop . Inside the loop we have handled the case of Invalid moves and the loop will break when there are no nonempty stack left.

After the game is over, the results are linked with the Game Log And LeaderBoard Functions.

Game log is the record of the played games. It will include Game Mode, date, time, name of the two players, score and the winner. We have used two functions to implement the game log . One to create the game log and another to present it. Here we will be showing the function to add data to the Game log . After completing a game, the above mentioned information of the game is appended in a text file . For time and date we have used the preprocessor macros.Then using another function the Game Log is displayed as output.

And the LeaderBoard will show the top 5 players of a certain Game Mode. To implement this We have combined three functions. One function to add the names of the winners in a certain text file , another to read the names of the winners from the text file and create a rank list by sorting and the last one to display the LeaderBoard.

Firstly we have appended the names of the winners in a certain text file. After that we read the names as strings and stored them in a vector. Then the vector is passed to another function where a map is used to count the frequency and sort them. Finally the ranked LeaderBoard is displayed.

We also have a help section showing the rules of the game, finally the credits.