Virtual Cricket Game Application

Hi, Welcome Back! Glad to see that we have made it so far in this training i.e. till the last module. This training has tested you on all fronts and has made sure that you become a good programmer. Continuing with the same spirit, enthusiasm, and excitement - we have one final project for you which is basically the next level of the **Gully Cricket** application that you have already developed in module 3. In this module, you have to create a **Virtual Cricket Game** application called **"CRIC-IN"**.

Now, how to create it? What all details are required? Don't worry, we have covered all the details for you. Below this, you will find all the necessary details required to code this project.

All the Best!

Problem Statement

Build a cricket game application using the C++ programming language.

Required Output

Please check the **game-output.pdf** file provided with this problem statement. It's not necessary to mimic the exact same output. You can use your own creativity and make your app output better than the given required output.

Project Requirements

1. There should be two teams: TeamA and TeamB
    a. Each team will have 4 players.
    b. The players for each team will be selected by the user from the given pool of 11 players. You can assign the names to those 11 players yourself.
    c. The sequence in which the players are selected for each team will decide the batting and bowling sequence for that team. For example, if Team A has Virat, Rohit, Dhawan, and Rahul; Virat will be the first player to bat or ball.
2. There should be two innings.
    a. Each innings will be of 6 balls.

    b. In each innings, only one bowler from the bowling team will bowl all the 6 deliveries and at a time one batsman from the batting team will bat until he is declared out.
    c. The first player from the bowling team will be always selected to bowl first and the first player from the batting team will be always selected to bat first.
    d. When a batsman is OUT, the next batsman from the batting team in sequence will start batting.
3. There should be a toss functionality.
    a. The team who wins the toss will decide to either bat or bowl first.
4. In each delivery, possible runs can be scored from 0 to 6.
5. OUT criteria: If a batsman scores 0 runs he will be declared OUT and the next batsman in sequence will start batting.
6. Match End criteria
    a. If runs scored by TeamA is greater than the opponent TeamB, then TeamA will win the game or vice-versa.
    b. If runs scored by TeamA and TeamB are the same then the match will draw.
7. Improve user experience by adding functions to display the pool of all the 11 players, display players of each team after team selection, display game scorecard after each delivery and display the match summary when the match ends.

Rough Algorithm with Approach

1. Create three classes namely Player, Team, and Game.
    a. Create 3 header files (.h) and the corresponding source files (.cpp) for the three classes.
    b. Declare Player class attributes to store information such as name, unique id, runs scored, balls played, balls bowled, runs given, and wickets taken in the match.
    c. Declare Team class attributes to store information such as team name, total runs scored by the team, total wickets lost, total balls bowled, and list of team players.
    d. Declare Game class attributes to store information such as team A and team B details, players per team, maximum deliveries allowed in each innings, names of all the 11 given players, and a variable to check which innings is going on.
    e. Additionally, in the Game class, declare two pointers of type class Team to point to the batting team and bowling team in each innings and two pointers of type class Player to point to the current batsman and the bowler.

2. Welcome users and show a few basic game instructions.
3. Display all the 11 players with the array index so that the user can select the players using that index for each team.
4. Select 4 players for each team one by one in an alternative way.
   a. Write a utility function to take **only** integer user input.
   b. Write a validation function to allow a player to be added to a team only once.
5. Display selected players from each team.
6. Add toss functionality.
   a. Allow the toss winning team to choose either to bat or bowl first.
   b. Initialize the batting team and the bowling team.
7. Start the first innings.
   a. Initialize the batsman and the bowler.
8. Start playing the first innings.
   a. After each delivery, update batsman score, bowler score, and both teams score.
   b. If in a delivery runs scored is 0 then the batsman will be OUT and the next player in sequence from the batting team will start batting.
   c. If in a delivery runs scored is 1 to 6, just update data and continue the innings.
   d. After each delivery, validate the innings score to check if the innings is over.
9. After each delivery, show the game scorecard.
10. When the first innings ends, start the second innings.
    a. Initialize the batting team and the bowling team.
    b. Initialize the batsman and the bowler.
11. Start playing the second innings.
    a. Reuse the same code written to play first innings.
    b. After each delivery, validate both innings’ scores to decide the winner.
12. Display match summary in the end.

Additional Points

1. Use escape characters such as **‘\t’**, **‘\n’** wherever needed to keep program output clean.
2. Use **usleep()** function as applicable along with user-friendly messages.

Submission

After completing the assignment, upload the **zip file** containing the source code file from the progress report.

You can download the solution to the assignment from the progress report once you have uploaded the solution. Compare your program with the provided solution and explore how a complex program can be broken down into smaller components by defining functions for each task. All the best.