MINI PROJECT
Automated Cricket Scoreboard

NAME	: G.D.M. PRAMODI
REGISTRATION NO	: 102829
INDEX NO	: 21/ENG/028
SUBMISSION DATE	: 25/09/2023

Design Overview

This is a brief overview of designing and implementing an automated cricket scoreboard.

In this design, it is mainly focused on the key functionality of a scoreboard which is displaying the scores and the wickets of each team. In order to calculate score, an automated cricket match is simulated. The cricket match simulation is displayed on the output console and a menu option was added to analyse the Score, Wickets, Boundaries (fours, sixes), Extras (Wide/No ball) for each team. Analysing the score and the bowling of each player was not included in this design. The run score of 5 is omitted as it is a rare occasion in a cricket match.

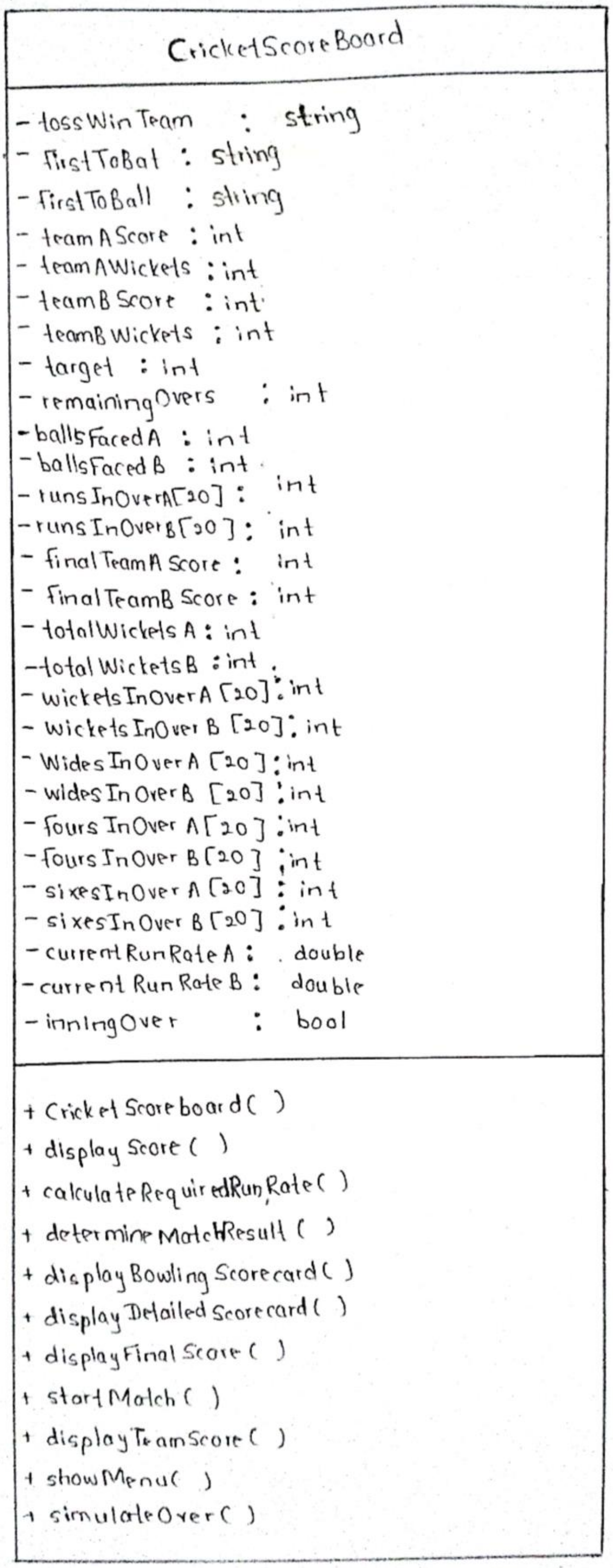
The choice of the winning team of the toss is given by a user input and the cricket match is simulated automatically accordingly. A T20 match (20 overs) was considered and the two teams were named as team A and team B in this design for easy implementation.

The entire match was automated and only the choices can be entered as a user input in this design.
The program code was divided into a header file, source file and a main file for better organization.

UML Class Diagram

The class diagram for the Cricket Scoreboard consists with a class named CricketScoreboard.

The class diagram was created by mainly focusing on the main function of a scoreboard which is displaying the score and the wickets in each team. The function parameters can be found in the code/the doxygen file provided.



Flow of the Program

1. Initialization

   The program begins by initializing various variables and arrays in the “CricketScoreboard constructor.
2. Main Function

   The main function is the entry point of the program. It initializes the random number generator, creates an instance of the “CricketScoreboard” class, and starts the match.
3. Match initialization

   The match starts with a coin toss to determine which team to bat first and ball first according to the choice of the winning team of the toss. The choice is entered as a user input and it is validated. The program will exit if the user input is invalid.
4. First Inning

   The program simulates the first inning based on the test result for 20 overs, each consisting of 6 balls. For each ball, it calculates the runs scored, handles wide balls, no balls and wickets and display the ball-by-ball commentary. The inning may end if 10 wickets fall.
5. Target Calculation
6. After the first inning, the program calculates the target score for the second inning based on the team that batted first.
7. Second Inning

   The program simulates the second inning similar to the first inning.
8. Match Result Determination

   After completing the both innings, the program determines the match result. The program displays the match result, indicating which team won and by how many runs and wickets or else whether it is a tie.
9. Menu Option

   After simulating the match, the program offers menu options to the user. The user can view the match result, detailed scorecard, bowling scorecard, or exit the program.
10. Display Functions

    The program contains functions to display the items in the menu option.
11. User Interaction

    Tried to minimize the user interaction, since this is designed to be an automated scoreboard. Therefore user interacts only by entering the choices and navigating the menu option throughout the program.
12. Exit

    The program runs after the simulation until the user selects the exit option in the menu list.

Challenges faced when implementing the system

- Understanding the cricket scoring system.

  The initial challenge is to have a comprehensive understanding of the cricket scoring system as it is vital to accurately calculate the scores and the statistics during the match.
- Get familiarize with the rules and regulations in cricket.

  Getting familiarize with the rules and regulations help to simulate a realistic cricket match accurately.
- Displaying user friendly output.

  Another key challenge was to display the output not only informative but also in a user-friendly manner which is clear and understandable for the user.
- Getting the same output when the program was run for several times successively.

  Since the system is designed to be automated, the rand() function was used for generating random numbers. In that case, when the program was executed for several times in succession, it was lead to the same output.

  Therefore to address this issue, the srand() function was used to ensure that each execution of the program produces different and dynamic results.
- Since some variables are reset to zero before starting the second inning, some data cannot be displayed accurately.

  Therefore, arrays are used to store data and provide the desired output.
- The current run rate was exceptionally higher than the range of the typical run rate of a cricket match (7 - 9). In order to make it similar to a realistic match, the probability of scoring runs was decreased.

Limitations

- This Score board is only suitable for T20 Cricket matches (only 20 overs).
- Statistics related to each player cannot be evaluated in this design.
- The number of players in each team is limited for 11 and no extra players.

Further Improvements

This design of the scoreboard can be developed further to evaluate each player separately and the details of each player (name, age, etc.) can be entered as a user input.