*Journal of Computer Science & Computational Mathematics, Volume 6, Issue 1, March 2016*
DOI: 10.20967/jcscm.2016.01.002

# Random Variables and Probability in Cricket Game Management

V. R. Lakshmi Gorty[*], Shashank Prabhu, Shaila Parekh, Dhwani Shah, Sachin Mistry, Mayank Tolia

Mukesh Patel School of Technology Management & Engineering, SVKM's NMIMS University,
[*]Corresponding author email: vr.lakshmigorty@nmims.edu

***Abstract***: This paper focusses on the concept of random variables and probability in making a cricket management game through C++ programming. For a finite number of outcomes which occur on each ball that is bowled, a discrete random variable is generated and the outcome probabilities are estimated from the data being used in the game. The probability in the game has an effect on the batsman, number of wickets lost and number of balls remaining. The game behaves like a simulator which is used to predict the outcome in the matches being played by the user. The game is designed in such a way that the sample space changes, and the probability of the random variable generated changes as well according to the in-game options and decisions made by the user. The concept of random variables and probability is applied to make the foundation logic for this game. The skills help in understanding what strategy can be implemented to be on the winning side. The study is developed with an attempt to realistically model a T20 cricket match, and to help in reflecting major tendencies.

***Keywords***: C++, Cricket, Probability, Random Variables.

## 1. Introduction

The game "Cricket Fever", which is developed for the purpose of simulation in this study, is one game which uses the concept of probability, random variables and mean. The basic idea in developing this game is by combining the mathematical concept with programming knowledge. The author in [1] states that students are expected to program in some procedural programming language such as C/C++, and to be able to deal with discrete mathematics, so some familiarity with basic data structure and algorithm analysis technique is also assumed. In [2] and [3], it is affirmed that luck can play a big part in tournament success, and progress is not necessarily the best measure of performance. The paper [3] also explains that optimum strategies may be expected to differ fundamentally in the first and second innings, typically involving an increasing rate when setting a target but a run-rate which may decline over the course of an innings when chasing a target. For calculating some elements like run-rate, we use the concept of mean, finding an average of runs scored for every over. In this way we can calculate the expected score for the team at the end of 20 overs.

## 2. Algorithm

The study exhibits the development of a cricket game in C++ programming language. This game demonstrates a good usage of strategy planning in cricket game so as to win the tournament. This section of the paper focuses on the algorithm and details about the game development and on how the game is played. The game is designed in such a way that the users, who stick to it, will be curious for each and every ball throughout the 20 overs. Knock opponent out or knock yourself out is the motto of the game. The game starts with a title screen, as shown in Figure 1.

```
Welcome to the game of CRICKET !!
```

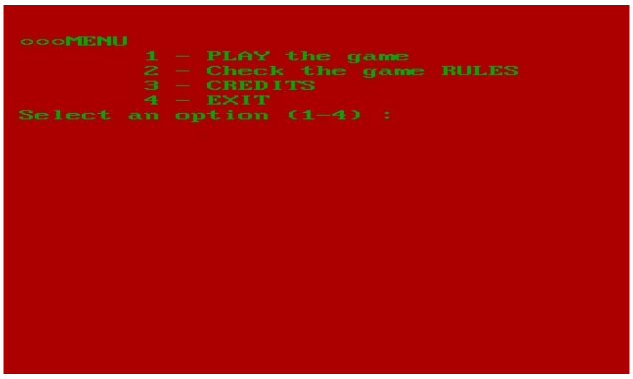
**Figure 1.** Title screen

```
The tournament begins here..._
```

**Figure 2.** Screen displaying the beginning of tournament

After a user enters any key, the game proceeds to the menu screen. The Menu screen consists of the following options:

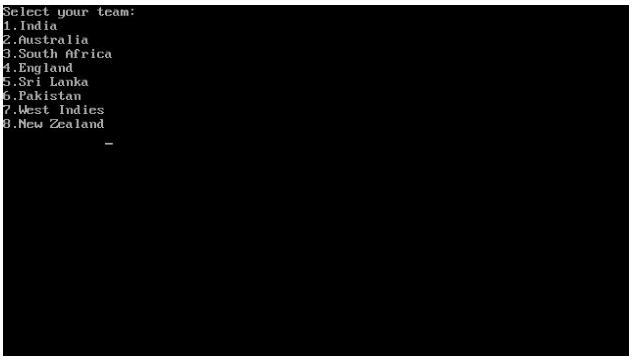
1. *Play the Game* : Select to play the tournament
2. *Check the game rules*: Select to get familiar with the game rules
3. *Credits*: Select to view details of the developers
4. *Exit:* Select to leave the game



```
°°°MENU
         1 - PLAY the game
         2 - Check the game RULES
         3 - CREDITS
         4 - EXIT
Select an option (1-4) :
```

**Figure 3.** Screen displaying the menu

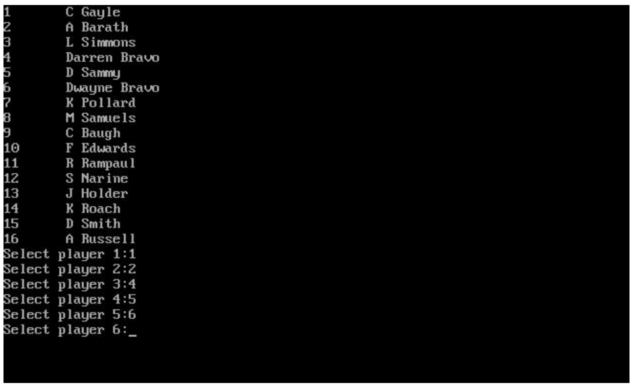
On selection of option 1, the user is navigated to the screen where he/she needs to select the team for whom he/she wants to play, as shown in Figure 4.



```
Select your team:
1.India
2.Australia
3.South Africa
4.England
5.Sri Lanka
6.Pakistan
7.West Indies
8.New Zealand
             _
```

**Figure 4.** Team selection screen

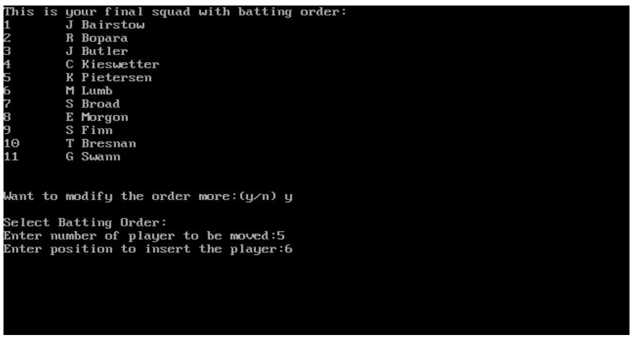
Once the user has made up his/her mind and selected his/her desired team, he/she now needs to select 11 players from a list of 16 pre-defined ones, which are available for each team including the team the user has selected. These 11 players will be the final squad for the team selected by the user and will play the tournament against the team that is randomly selected by the computer.



```
1       C Gayle
2       A Barath
3       L Simmons
4       Darren Bravo
5       D Sammy
6       Dwayne Bravo
7       K Pollard
8       M Samuels
9       C Baugh
10      F Edwards
11      R Rampaul
12      S Narine
13      J Holder
14      K Roach
15      D Smith
16      A Russell
Select player 1:1
Select player 2:2
Select player 3:4
Select player 4:5
Select player 5:6
Select player 6:_
```

**Figure 5.** Screen for selecting the 11 players from a list of 16 players

In the real cricketing world, the captain has the ability and power to change the batting order of batsman. Keeping the look & feel similar to the real world and using it as a metaphor, the program is developed in such a way that the user can also change the batting order of the 11 players he/she selected in the final squad. Once the user is ready with the batting order, he/she can press enter to start playing the match.



```
This is your final squad with batting order:
1       J Bairstow
2       R Bopara
3       J Butler
4       C Kieswetter
5       K Pietersen
6       M Lumb
7       S Broad
8       E Morgon
9       S Finn
10      T Bresnan
11      G Swann


Want to modify the order more:(y/n) y

Select Batting Order:
Enter number of player to be moved:5
Enter position to insert the player:6
```

**Figure 6**. Screen for selecting the batting order

Fundamentally, the program is developed in such a way that it uses a random function which arbitrarily selects any number from the pre-determined limit, and batting of each player depends on that random number. For instance, if the computer generates a random number ‘5’ which has a function of increasing the run by 1, thus this will be shown to the user in a way that his/her batsman has scored 1 run for that particular ball.

A feature that is striking in this program is that a user gets to select the mode in which he/she wants his/her batman to play each over to be on the winning side. Modes available to the user are: defensive, aggressive and normal. As stated in [4], either batsman can choose to bat aggressively or defensively which alter their chances of scoring runs or being dismissed.

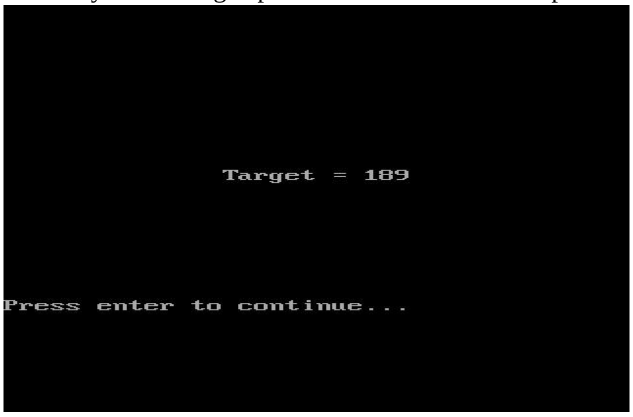
*Mode 1 – Defensive:* Here the program is designed in such a way that the team players have maximum probability of hitting 1 or 2 runs per ball. Also, the probability of a wicket falling is less.

*Mode 2 – Aggressive:* Here the program is designed in such a way that the team players have maximum probability of hitting fours and sixes. The probability of a wicket falling is high in this mode.

*Mode 3 – Normal:* Here the program is designed in such a way that the team players have properly distributed probability of hitting any shot. The probability of wickets falling is intermediate between the aggressive and defensive modes.

In [5] the author conveys that a team captain or management can consider an aggressive or defensive batting strategy for the coming session. User’s task is to press the “enter” key to control his/her every over. As the user presses the “enter” key, statistics of every ball of that over is displayed. The paper [6] presents a simulation component that generates runs ball by ball during an innings taking into account the state of the match and estimated characteristics of individual batsman. A user has an option to choose the mode in which he/she would like the batsman to play the next over and the game continues till 20 overs are completed or all 11 players are out. The author, in [7], presents for interrupted cricket matches, an adjustment rule that equalizes the probability of winning before and after the interruption, which we claim to be fair and free of incentive effects. It is the basic way on how the game works. The game currently provides the user to bat and compete with the target generated at random by the computer. Following screenshots show the way in which the game is played.

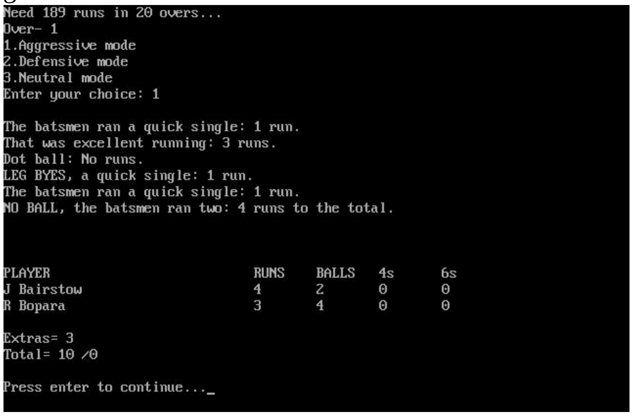
The flow of the program developed is in such a way that firstly, the opponent team is randomly selected by the computer and secondly, the target score is generated as shown in figure 7. The author emphasizes in [8] that the probability of each team winning from any game situation and proposes adjusting the target score in such a way that the probability of winning is preserved across the interruptions.



```
                    Target = 189



Press enter to continue...
```

**Figure 7.** Random target generated by computer & user is expected to chase this target

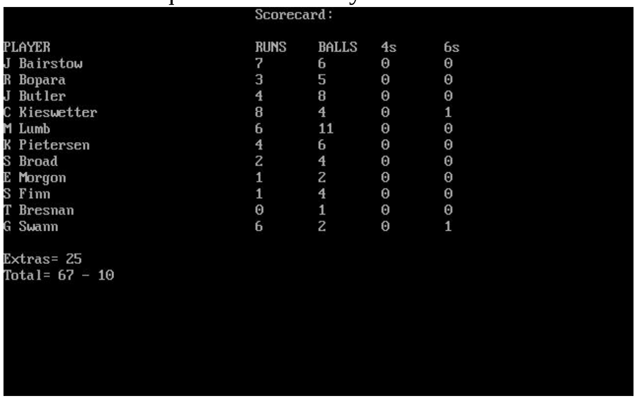
The user has to chase the computer's score to win the game. Once the user has selected his/her team, the players and the orders, the user is ready to play the tournament. According to the batting order selected by the user the batsman is selected and the user gets a choice to select one of the modes. After the user presses "enter", statistics of the 1st over is displayed to the user with details and existing commentary for each run just as in real world cricket. Run-rate and projected score of the team are updated after each over. In [9], the author implements a visual basic program, SimScore, to simulate batsman score from the probability distributions pertaining to the match stage of interest. Total runs needed, the commentary and the details of the players batting are displayed to the user after each over as shown in figure 8.



```
Need 189 runs in 20 overs...
Over- 1
1.Aggressive mode
2.Defensive mode
3.Neutral mode
Enter your choice: 1

The batsmen ran a quick single: 1 run.
That was excellent running: 3 runs.
Dot ball: No runs.
LEG BYES, a quick single: 1 run.
The batsmen ran a quick single: 1 run.
NO BALL, the batsmen ran two: 4 runs to the total.


PLAYER                          RUNS    BALLS   4s      6s
J Bairstow                      4       2       0       0
R Bopara                        3       4       0       0

Extras= 3
Total= 10 /0

Press enter to continue..._
```

**Figure 8.** Screen displaying the stats after each over and giving freedom to the user to select a mode

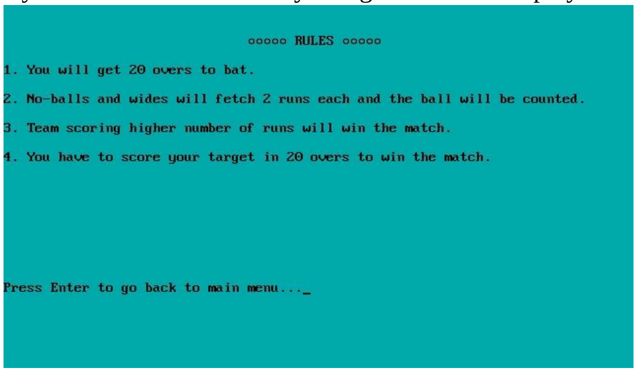
In this fashion, the statistics of each over are presented till 20 overs are completed or all the batsmen are out.
Finally, the scoreboard of batsman is displayed to the user as shown in figure 9. This will lead to the next innings of the match if the user wins the current match. This game is designed to have 3 matches i.e. the team has to play 3 matches in the game and if it wins all the matches in the tournament, then the team will be declared as the winner. The author states in [10] that using a dynamic programming formulation, an analysis is presented for both the first and second innings of one-day cricket match assuming variation in type of ball bowled and subsequent selection of a strategy by the batsman. The paper [11] estimates the systematic co-variation among various dimensions pertaining to batting and bowling capabilities of T-20 cricket using the advanced statistical technique of factor analysis.



```
                                Scorecard:

PLAYER                          RUNS    BALLS   4s      6s
J Bairstow                      7       6       0       0
R Bopara                        3       5       0       0
J Butler                        4       8       0       0
C Kieswetter                    8       4       0       1
M Lumb                          6       11      0       0
K Pietersen                     4       6       0       0
S Broad                         2       4       0       0
E Morgon                        1       2       0       0
S Finn                          1       4       0       0
T Bresnan                       0       1       0       0
G Swann                         6       2       0       1

Extras= 25
Total= 67 - 10
```

**Figure 9**. Scoreboard of batsman

The author in [12], talks about probabilities that are estimated from historical data involving one day international cricket matches. The probabilities depend on the batsman, the bowler, the number of wickets lost, the number of balls bowled and the innings. Figure 10 displays Rules page for players to understand the way this game should be played.



```
                    ooooo RULES ooooo

1. You will get 20 overs to bat.

2. No-balls and wides will fetch 2 runs each and the ball will be counted.

3. Team scoring higher number of runs will win the match.

4. You have to score your target in 20 overs to win the match.




Press Enter to go back to main menu..._
```

**Figure 10**. Game rules

This game is a metaphor of the real world cricket game viewed in stadiums or TV's. The program can be played by any age group. The format of each stat is user friendly and the parsimony is maintained in the program as well as the paper. The commentary displayed for each ball helps to gain the users interests and helps the user to be engrossed in the game. As it is a knock out tournament, the user enjoys playing the game and he/she develops curiosity to win the tournament. Playing on a regular basis will facilitate the user to formulate strategies to win all the following tournaments.

Thus, to win the tournament the user needs to plan a strategy and to select the modes accordingly after each match such that it minimizes the number of wickets lost and maximizes the number of runs gained.

## 3. Mathematical interpretation

The data collected from the game can be used to win the game by deciding which mode to play. The data collected here is:

1: Number of fours in an innings
2: Number of sixes in an innings
3: Number of wickets in an innings
4: Number of runs per innings.

Calculating the data:-

1. For number of fours there is counter placed in the program which increments by 1 every time a four is hit.
2. Similarly another counter is placed for number of sixes which increments by 1 for every six hit.
3. Wickets counter increments by 1 every time a wicket falls and this counter also helps to stop the program when wicket counter reaches 10, as the maximum wickets which can fall is 10.
4. Number of runs is the total runs of the team.

*Procedure*: Data is taken for 50 matches. Mean is taken of every member of the 50 games. After the mean is taken, the mean works like a cheat code for the game.
Supposedly, the mean has come out per innings, say:

(1) Number of fours= 12
(2) Number of six= 5
(3) Wickets= 4
(4) Runs= 166

In the game a player has to chase supposedly 195 runs, and then the player has to choose more aggressive overs than normal or defensive as the team needs to hit more number of sixes and fours to surpass 195 runs, because on an average the team can hit 166 runs, 12 fours and 5 sixes with 4 wickets down.

These statistics could be said to be achieved in normal mode, because when we take the data for 50 games the aggressive mode and defensive mode get nullified.

To get 195 runs in a game the user needs more runs than 166, therefore he/she needs to play more aggressively to win the game. To find the

$$\text{Mean} = (\sum x) \div n$$

Where x= number of fours, or sixes, or wickets, or runs per Innings, and
n= Data collected for n number of games

**EXAMPLE 1:**

The Total numbers of fours that is: $\sum x$, where
x= number of fours per innings

Considering $\sum x$= 300, and the data collected is for 50 games then,

$$Mean = \frac{300}{50}.$$

It means that on an average 6 fours are hit in a game. The following is a table consisting of readings taken for 4s, 6s, wickets and runs across 50 games played:

**Table 1.** 4s, 6s, wickets and runs across 50 games played

| Sr. No. | Fours | Sixes | Wickets | Runs | Win/Lose |
|---|---|---|---|---|---|
| 1 | 4 | 8 | 10 | 90 | L |
| 2 | 10 | 5 | 8 | 176 | W |
| 3 | 4 | 6 | 3 | 153 | W |
| 4 | 1 | 4 | 10 | 130 | L |
| 5 | 3 | 4 | 10 | 106 | L |
| 6 | 9 | 5 | 7 | 167 | L |
| 7 | 6 | 4 | 9 | 156 | W |
| 8 | 5 | 5 | 7 | 160 | W |
| 9 | 7 | 3 | 10 | 117 | L |
| 10 | 4 | 1 | 10 | 131 | L |
| 11 | 9 | 4 | 1 | 164 | W |
| 12 | 3 | 1 | 10 | 100 | L |
| 13 | 3 | 3 | 8 | 156 | W |
| 14 | 6 | 6 | 10 | 157 | L |
| 15 | 6 | 2 | 10 | 112 | L |
| 16 | 6 | 4 | 6 | 172 | W |
| 17 | 10 | 3 | 2 | 155 | W |
| 18 | 9 | 4 | 0 | 167 | W |
| 19 | 4 | 5 | 2 | 167 | W |
| 20 | 4 | 4 | 8 | 182 | W |
| 21 | 2 | 2 | 2 | 152 | W |
| 22 | 5 | 6 | 5 | 155 | W |
| 23 | 8 | 1 | 0 | 158 | W |
| 24 | 0 | 3 | 0 | 164 | W |
| 25 | 5 | 3 | 5 | 172 | W |
| 26 | 5 | 7 | 0 | 180 | W |
| 27 | 4 | 4 | 2 | 168 | W |
| 28 | 1 | 4 | 10 | 76 | L |
| 29 | 6 | 9 | 3 | 170 | W |
| 30 | 1 | 3 | 5 | 174 | W |
| 31 | 0 | 1 | 10 | 75 | L |
| 32 | 13 | 0 | 4 | 162 | W |
| 33 | 6 | 6 | 5 | 170 | W |
| 34 | 0 | 3 | 10 | 35 | L |
| 35 | 4 | 1 | 3 | 158 | W |
| 36 | 1 | 5 | 2 | 172 | W |
| 37 | 3 | 4 | 7 | 158 | W |
| 38 | 3 | 1 | 10 | 81 | L |
| 39 | 3 | 1 | 10 | 107 | L |
| 40 | 7 | 9 | 5 | 171 | W |

| Sr. No. | Fours | Sixes | Wickets | Runs | Win/Lose |
|---|---|---|---|---|---|
| 41 | 6 | 5 | 3 | 167 | W |
| 42 | 6 | 1 | 3 | 174 | W |
| 43 | 1 | 0 | 10 | 26 | L |
| 44 | 3 | 0 | 10 | 57 | L |
| 45 | 1 | 5 | 9 | 160 | W |
| 46 | 7 | 2 | 2 | 172 | W |
| 47 | 8 | 4 | 4 | 157 | W |
| 48 | 5 | 8 | 8 | 159 | W |
| 49 | 5 | 1 | 4 | 173 | W |
| 50 | 0 | 0 | 10 | 49 | L |
| **Total:-** | 232 | 180 | 302 | 7070 | W=33 L=17 |
| **MEAN :-** | 4.64 | 3.6 | 6.04 | 141.4 | W=66% L=34% |

## 4. Discussion

In this article, the cricket game based on average run-rate, and average number of 4s, 6s and wickets per match is analyzed.

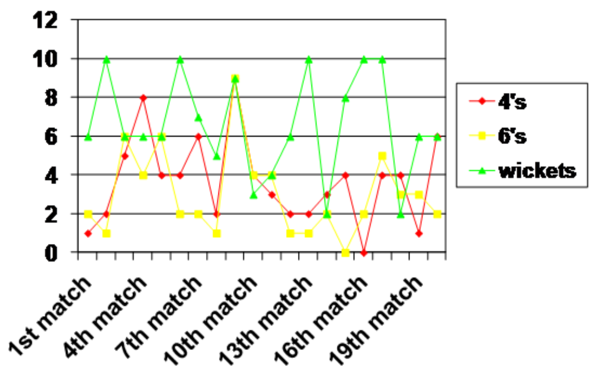


**Figure 11.** Number of matches versus wickets falling per match

In this graph, the number of matches is plotted on the x-axis. On y-axis the numbers of 4s, 6s and wickets falling per match are plotted. The red, yellow and green lines indicate the numbers of 4s, 6s and wickets per match respectively.

The number of 4s, 6s and wickets keep on changing per match. Thus, the variations are graphically represented. The graph represents the runs per match, and the match is a mixture of normal, aggressive and defensive modes. In the match some overs will be played in aggressive mode, while some are in normal and the others in defensive modes depending on the user, and runs are generated accordingly and plotted on the graph, as shown in Fig. 11. The graph helps the user to know whether his/her team is performing better or worse than the average score. The study is developed with an attempt to realistically model a T20 cricket match, and it appears to do a reasonable job of reflecting major tendencies.

## 5. Results

In this article we successfully simulated a T20 cricket match. The concept of random variables and probability is applied to make the foundation logic for this game. The concept of probability is used to analyze the game and extract information such as run-rate, average numbers of 4s, 6s and wickets and the expected score. Using the concept of probability, the average numbers of 4s, 6s and wickets per match are closely matching with the numbers of 4s, 6s and wickets in every match. This knowledge helps in understanding what strategy should be implemented to be on the winning side.

## 6. Future scope

According to the interests of the programming group the game could be designed in the way as presented in this paper. Some of the elements which can be modified later will include features of visual graphics and sound effects. For instance, while choosing the players for the team, the players can be displayed with a visual appearance of them. Video clips can be inserted at the start of the match showing the stadium and the environment. The sound effect of an audience applause after every over in which a four or six is generated will boost the mood of user. At the end of the innings, video clips of few random moments of the match can be shown as highlights of the match. Also it can be converted into other programming languages, and any other cricket format (e.g. 50 overs, test matches, etc) can be implemented using this logic. Effects of selecting a particular ground may be added in the future to enhance this game.

Currently, the selected player's name does not have any effect on the outcome of the game. This is because the players are simply names in the present game. In the future there is scope to add attributes and characteristics to the players. Many factors have been considered such as form, playing style, attitude, approach, etc. that will shape the players' persona in the game. This will allow a realistic effect to be added to each player along with his/her name. This will make the players behave, or play the game, like they do in real life. Thus, selecting an aggressive player would increase the probability of an aggressive outcome to the game, similarly with defensive players.

## References

[1] I. Parberry, "Lecture Notes on Algorithm Analysis and Computational, Complexity (Fourth Edition)," *Department of Computer Science (University of North Texas),* Dec 2001.

[2] S. R. Clarke, P. Allsopp, "Fair Measures of Performance: the World Cup of Cricket," *Journal of the Operational Research Society (2004)*, vol. 52, no. 4, pp. 471-479, 2001.

[3] I. Preston, J. Thomas, "Batting Strategy in Limited Overs Cricket," *Journal of the Royal Statistical Society*: *Series D*, vol.49, no. 1, pp. 95-106, 2000.

[4] S. R. Clarke, J.M. Norman, "To Run or Not? Some Dynamic Programming Models in Cricket," *Journal of*

*the operational Research Society*, vol. 50, no. 5, pp. 536-545, 1999.

[5] S. Akhtar, P. Scarf, "Forecasting Test Cricket Match outcomes in play," University of Salford, *Salford Business School Working Paper Series*, vol. 28, no. 3, pp. 632-643, 2012.

[6] T. B. Swatrz, P. S. Gill, D. Beaudoin, B. M. deSilva, "Optimal Batting Orders in One-Day Cricket," *Computers and Operations Research*, vol. 33, no. 7, pp. 1939-1950, 2006.

[7] P. Allsopp, "Measuring Team Performance and Modeling the Home Advantage effect in Cricket", *Thesis*,researchbank.swinburne.edu.au/vital/access/services/.../swin.../SOURCE2, 2005.

[8] M. Ovens, B. Bukiet,"A Mathematical Modeling approach to One-Day Cricket Batting orders", *Journal of Sports Science and Medicine*, vol. 5. no, 4, pp. 495–502, 2006.

[9] J. R. T. Sargent, A. Bedford, "Using Conditional Estimates to Simulate In-Play Outcomes in Limited Overs Cricket," *Journal of Quantitative Analysis in Sports*, vol. 8, no. 2, Page-ISSN (Online) 1559-0410, DOI: 1515/1559-0410.1430, June 2012, Published: 2012-06-08.

[10] P. Norton, R. Phatarfod, "Optimal Strategies in One-Day Cricket," *Asia-Pacific Journal of Operations Research*, World Scientific Publishing Co. & Operational Research Society of Singapore, vol. 25, no. 4, pp. 495-511, 2008.

[11] S. K. Sharma, "A Factor Analysis Approach in Performance Analysis of T-20 Cricket," *Journal of Reliability and Statistical Studies*, vol. 6, no. 1, pp. 69-76, 2013.

[12] T. B. Swartz, P. S. Gill, S. Muthukumarana, "Modeling and Simulation for One-Day Cricket," *The Canadian Journal of Statistics*, vol. 37, no. 2, pp. 143-160, 2009.