CS101 Project 2015

BRAND MONOPOLY BOARD GAME PROJECT REPORT

Team ID 419

Avineil Jain - 14D170002 (Group Leader)

Sharnam Shah - 14D110001

Rajas Pawar - 140070012

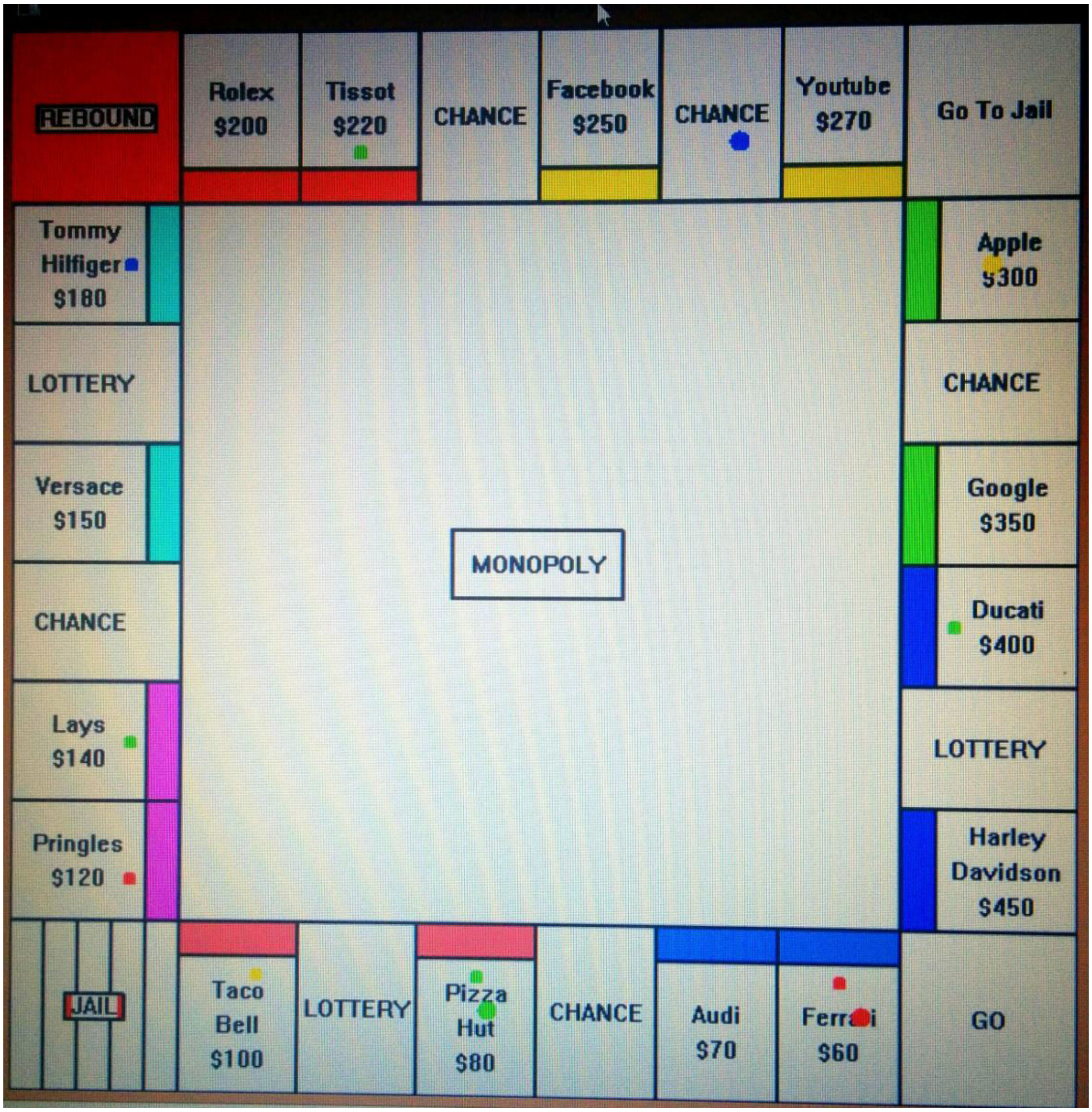
Kamini SuryaMohan Reddy - 140010049

TABLE OF CONTENTS

1. INTRODUCTION

Brand Monopoly is an innovative and fun version of the original monopoly board game which involves shares of famous brands and companies instead of places and properties.

Monopoly is a board game which is wound around the life of the player. It involves rolling of a die which decides the advancements the player makes on the board. The game can run for few hours or even days. All transactions are virtual money based. Basic rules are kept unchanged.



2.PROBLEM STATEMENT

To make a monopoly board game for 2 or 4 players using C++ and simplecpp graphics library.

TASKS AIMED FOR :

i. To make a working game for 2 or 4 players

ii. The game should end when one player loses his money in 2 player, and the game should continue without that player in 4 player game, ultimately ending with a winner

iii. To introduce new elements of gameplay to make the game better

iv. To use simplecpp library to its maximum to try and make a very good game

v. To make all the transactions run very smoothly, without any errors or bugs

TASKS COMPLETED

- Referring to the tasks that we set out on, we can happily say that we did complete all the tasks set out by us
- The game works for 2 or 4 players
- The game ends without error in 2 or 4 players
- The gameplay works smoothly with all transactions happening without error

3.REQUIREMENTS

Codeblocks IDE with simplecpp library installed.

4.CHALLENGES-

There were numerous challenges that we faced during the execution of our project.

In the code-

- To continue the 4 player game, when a player was out
- To add the rent, sell and auction function
- To add the rectangles after a property was bought
- To move the token without any error
- To make the board itself
- To make the gameplay better by adding new elements

In real life -

- To coordinate between all the members of our group and work distribution
- To ensure that each member contributes in one way or another for the project

5.HOW WE TACKLED THE CHALLENGES –

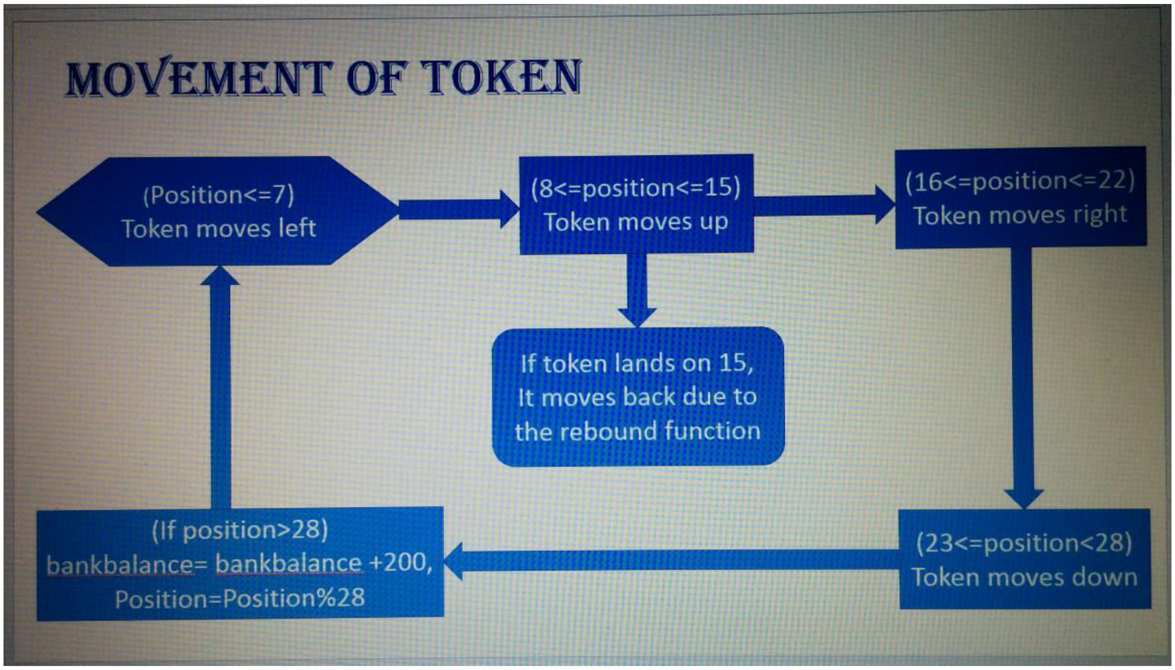
- Work was distributed so that each member could contribute in a significant manner. Weekly checks by the group leader for progress was made.

- The movement of token was perfected by repeated testing-

  1) The token had the tendency of not staying at the corners, that was corrected

  2) Rebound function had to be written separately, if the token landed on Rebound

  3) Cyclic motion of token has been explained in the following diagram -



MOVEMENT OF TOKEN

(Position<=7)
Token moves left

(8<=position<=15)
Token moves up

(16<=position<=22)
Token moves right

If token lands on 15,
It moves back due to
the rebound function

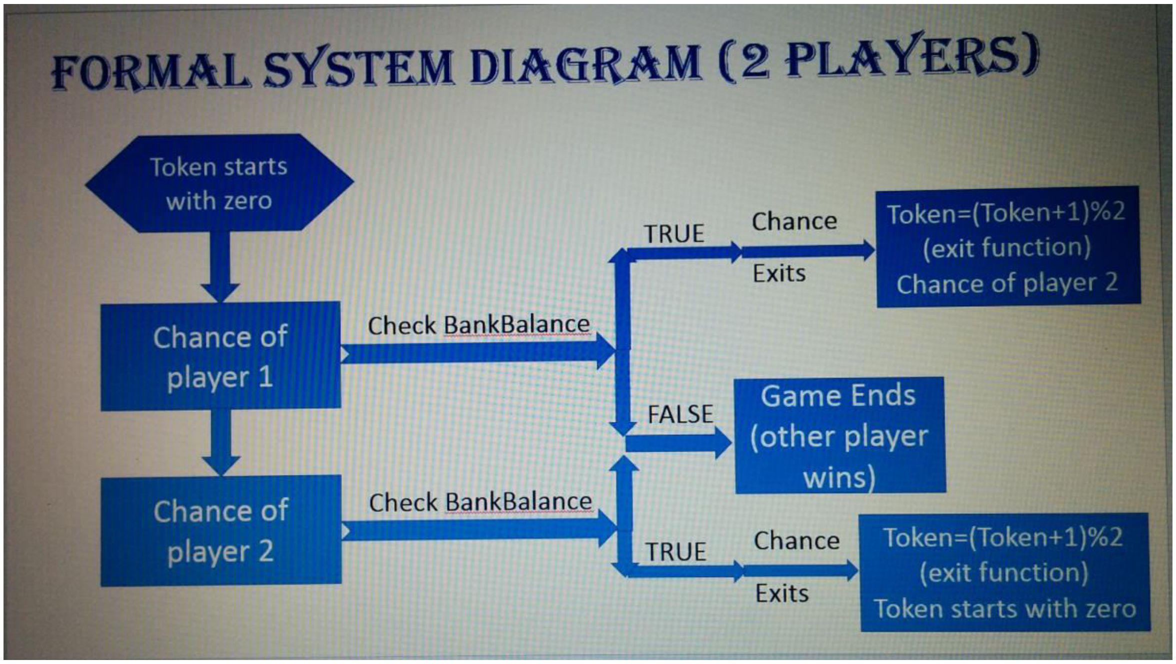
(23<=position<28)
Token moves down

(If position>28)
bankbalance= bankbalance +200,
Position=Position%28

❖ The 4 player game was designed in a new way-

1) The game initially had a tendency of not ending properly

2) First 2 player game was designed-



FORMAL SYSTEM DIAGRAM (2 PLAYERS)

Token starts with zero

Chance of player 1

Check BankBalance

TRUE

Chance Exits

Token=(Token+1)%2 (exit function) Chance of player 2

FALSE

Game Ends (other player wins)

Chance of player 2

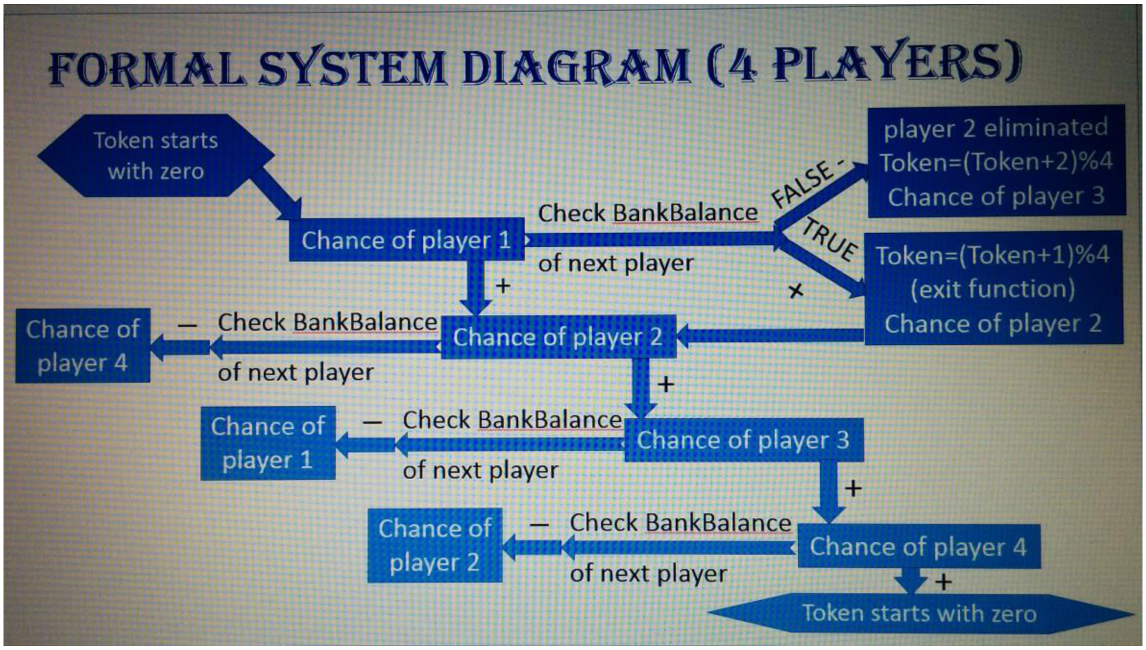
Check BankBalance

TRUE

Chance Exits

Token=(Token+1)%2 (exit function) Token starts with zero

That was modified to give the following 4 player mode –



FORMAL SYSTEM DIAGRAM (4 PLAYERS)

Token starts with zero

Chance of player 1

Check BankBalance of next player

FALSE -

player 2 eliminated Token=(Token+2)%4 Chance of player 3

TRUE

Token=(Token+1)%4 (exit function) Chance of player 2

+

×

Chance of player 4

– Check BankBalance of next player

Chance of player 2

+

Chance of player 1

– Check BankBalance of next player

Chance of player 3

+

Chance of player 2

– Check BankBalance of next player

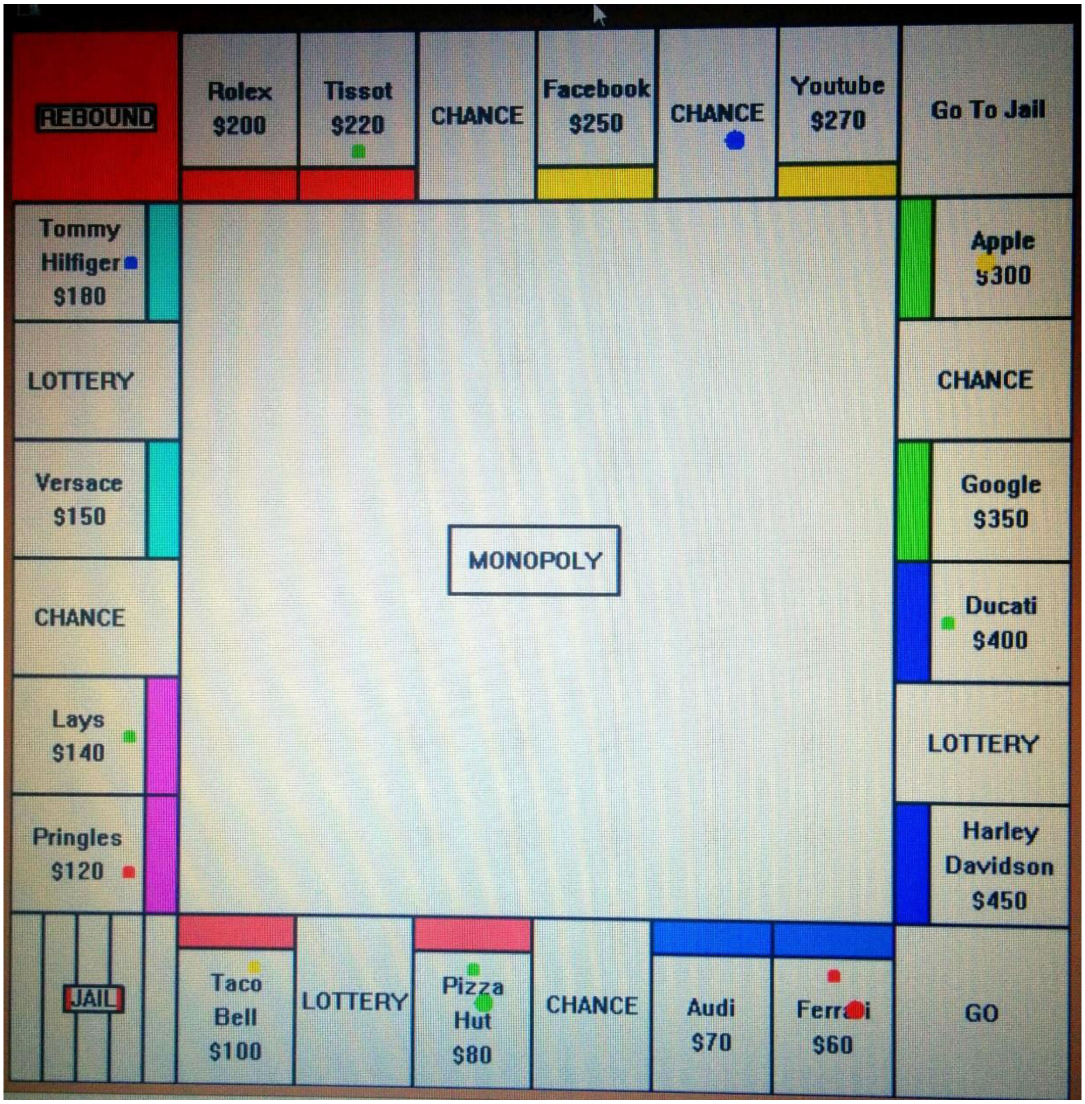
Chance of player 4

+

Token starts with zero

3) After each turn, balance of next player is checked and progress is made in that way. A sort of recursion takes place.

- The board was made by careful planning of the coordinates, and noting that each rectangle is uniform.



- Rent, sell and auction functions were made using certain parameters defined in our code, namely balance and property-index.

6.INNOVATION

- The whole code is written all by ourselves, we have not taken a single code snippet from the net.
- The working of the 4 player game, required lots of thought and planning (as shown in the flow diagram).
- The movement of token had to designed based on a single parameter - The current position.
- The board was made completely by us, and all the new gameplay elements- rebound, lottery were thought by us and implemented.
- The new gameplay elements make the game a lot of fun to play.

7.TESTING

TEST CRITERIA : To test all the elements in the game and see whether they work as predicted without error.

TEST DESCRIPTION :

- The movement of token had to be improved a lot, initially running into a lot of errors(such as not turning upon reaching the edge of the board, the rebound function when the token moves backwards and not forwards.
- The game ending(especially for 4 players) had to be improved again and again. The basic logic is that the chance of eliminated player gets skipped.
- Buying and rent function had to made separately for the 4 player game , as concluded from the tests done on them. They required more parameters than the 2 player functions, and certainly a lot more coding.

8.DISCUSSION OF SYSTEM

A) WHAT WORKED AS PER OUR PLAN?

Most of the things we aimed for worked as per our plan.

1) The board was completed and developed as per planned.
2) All the functions were successfully created that are required for the gameplay of monopoly.
3) A rule book was created as per plan.
4) The elements we mentioned in the SRS were successfully created.

B) WHAT WE ADDED MORE THAN WE DISCUSSED IN THE SRS?

We have added a sell function and an auction function that we originally didn’t plan to add. The sell function allows a player to sell any of his/her properties when his/her bank balance is lower than $500. The auction function allows players to buy a property through auction when another player lands on a property and doesn’t buy it.

C) What changes we made in our plan?

We started making the two player and four player game together, but then we only made the two player game for our Prototype-I and then using that, we developed the code for the four player game.

9.Future Directions

I. This game represents what this team could do best, in the finite resources and knowledge that the members have acquired over the whole CS 101 course.

II. However, the game could be improved by using a better graphics library thus improving the visual appeal of the game.

III. More of mouse and less of keyboard user interface could be made, using that better graphics library.

IV. The working may be subjected to stringent rules in the game.

10.Conclusions

We have successfully created a brand based monopoly game based on simplecpp graphics. This can contains the factor of risk which brings the fun in the game, this game can be played for fun and time pass. It shows that students can successfully create a working game using just the knowledge that we have gained through the whole CS 101 course.

REFERENCES

For installation of simplecpp and code blocks.

1) http://www.cse.iitb.ac.in/~ranade/simplecpp/

For the rules of the game.

2) http://en.wikipedia.org/wiki/Monopoly
3) http://en.wikipedia.org/wiki/Monopoly_(game)
4) http://monopoly-game.net/Classic_Monopoly_Rules.html
5) http://en.wikibooks.org/wiki/Monopoly/Official_Rules
6) http://www.wikihow.com/Play-Monopoly