

# PROJECT REPORT

---

## CHINESE CHECKERS

---

120050017      VIKAS GARG

120050009      DEEPANJAN KUNDU

Problem :- To make chinese checkers game computer v/s player. The board is a rhombus of  $9 \times 9$  dimension. It is the part used by two players in professional chinese checkers. The main problem is that how computer identify the best move according to situation of board.

### Overall IDEA of the PROBLEM:-

- 1) The board used here is a state variable carrying the positions occupied as true and the rest as false. The board is a collection of points of a parallelogram coordinate-system. There is a sensible use of mutable lists because of set!-mcdr.
- 2) The players and pieces are represented as classes.
- 3) The tree is also a general tree with a position list, list of sub-trees and a value.
- 4) The aim was to make a tree representing the possible moves computer and player would make using alpha-beta pruning. The tree removes the unnecessary possibility of moving behind or entering the corners. The artificial intelligence is a function devised by us by experimenting the game play.
- 5) The Artificial Intelligence would decide the value of the moves. We have made long jumps, avoiding moving into the corners as far as possible, carrying all the pieces altogether and blocking the opposite team. The formula used here makes (its value very high) if it is too long move or if there is any winning move else it uses a formula which includes terms such as sum of coordinates of the pieces, their differences and their squares for the last four of the pieces.

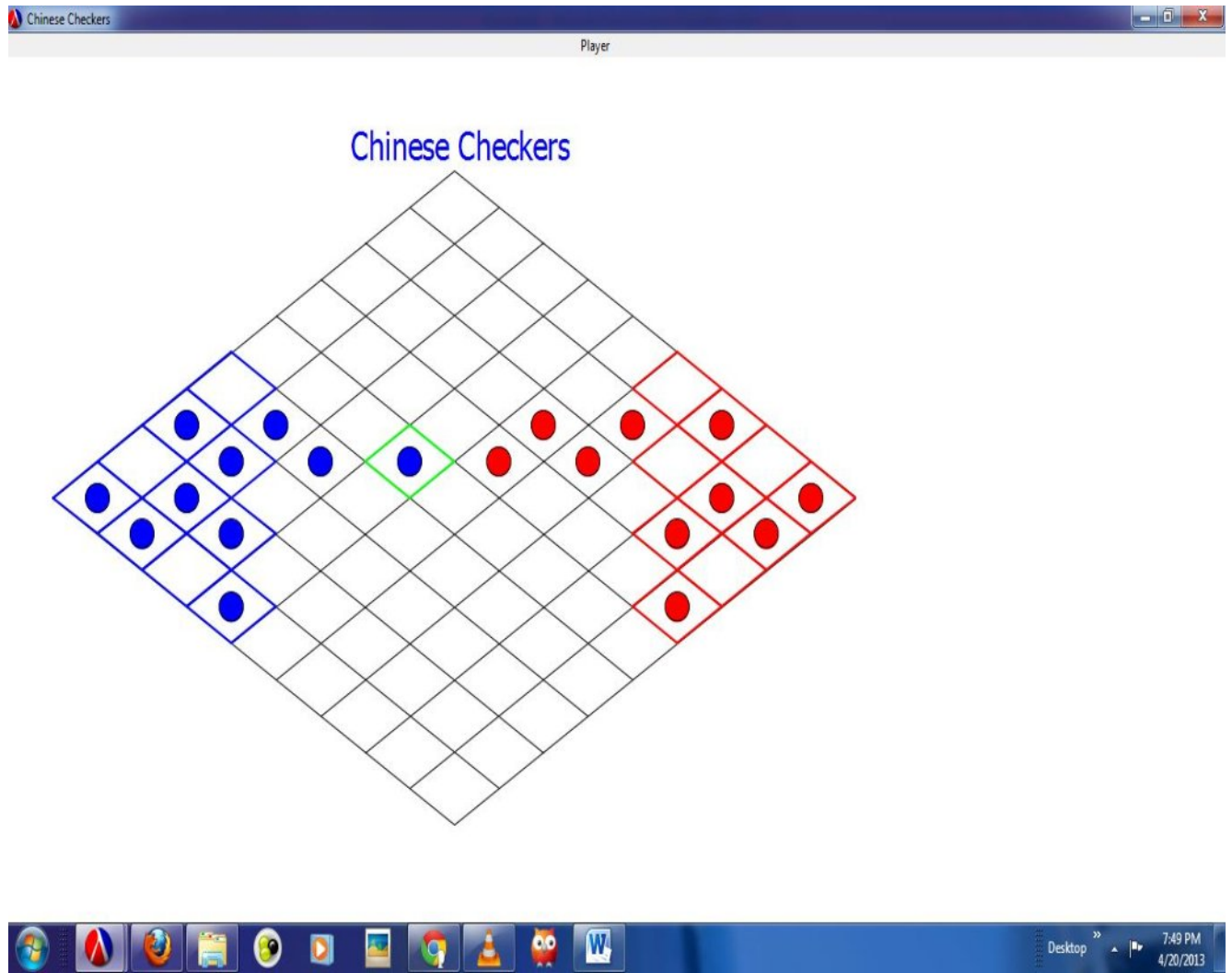
6)The tree goes to the depth of 4 in the starting and filling part,but to a depth of 3 in the core interaction part as it takes about 5-10 seconds to make a move.

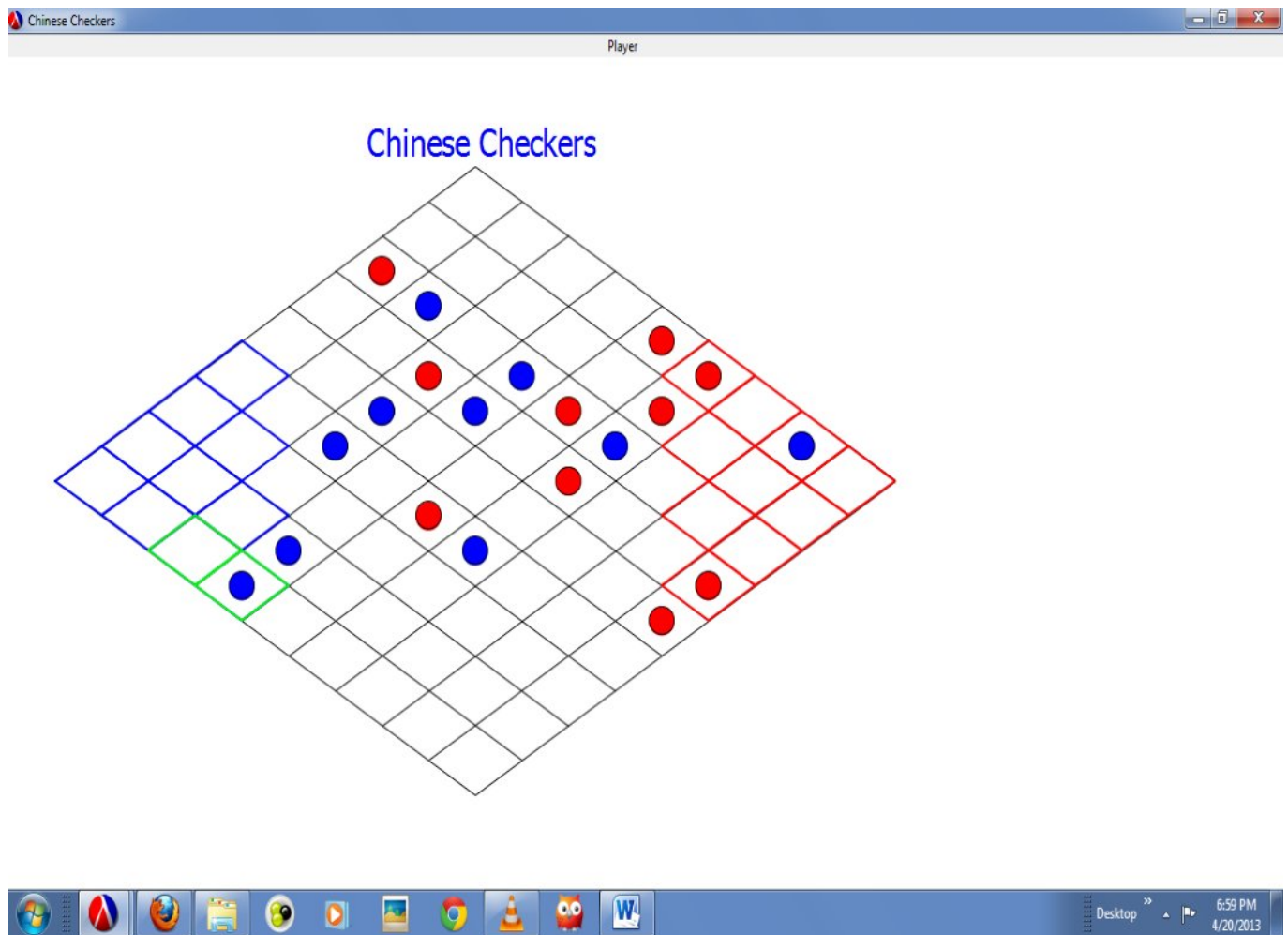
The game is of fairly moderate level.Good for amateurs to learn good moves.

There is a quite frequent use of abstractions eg: classes(based on object oriented programming),"let" has been used quite frequently ,higher order functions like foldr,map etc.

The graphics have been made using the gui and draw library in the racket version 5.3.1.

Some of the glimpse of the gameplay





### Sample Input And Output:-

Input has to be in form of mouse clicks at the initial and final position and the output will be given visually in the frame. Both user and computer moves are executed simultaneously.

Limitations or bugs:- May take a little time when there is too much interaction possibilities with the opposite team.

Both user and computer moves are executed simultaneously.

### Others:-

The rules have been mentioned in the Readme.txt.