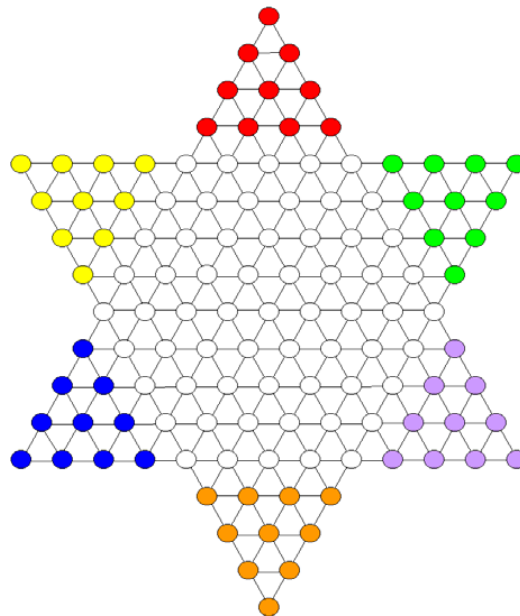


Project Report

CS152

Professor : Sanyal Amitabh

Chinese checkers



Chinese Checkers Board

Team Members

- Vipul Harsh 110050034
- Niles Kulkarni 110050007
- Ravi Kumar Roshan 110050033

Description of the game

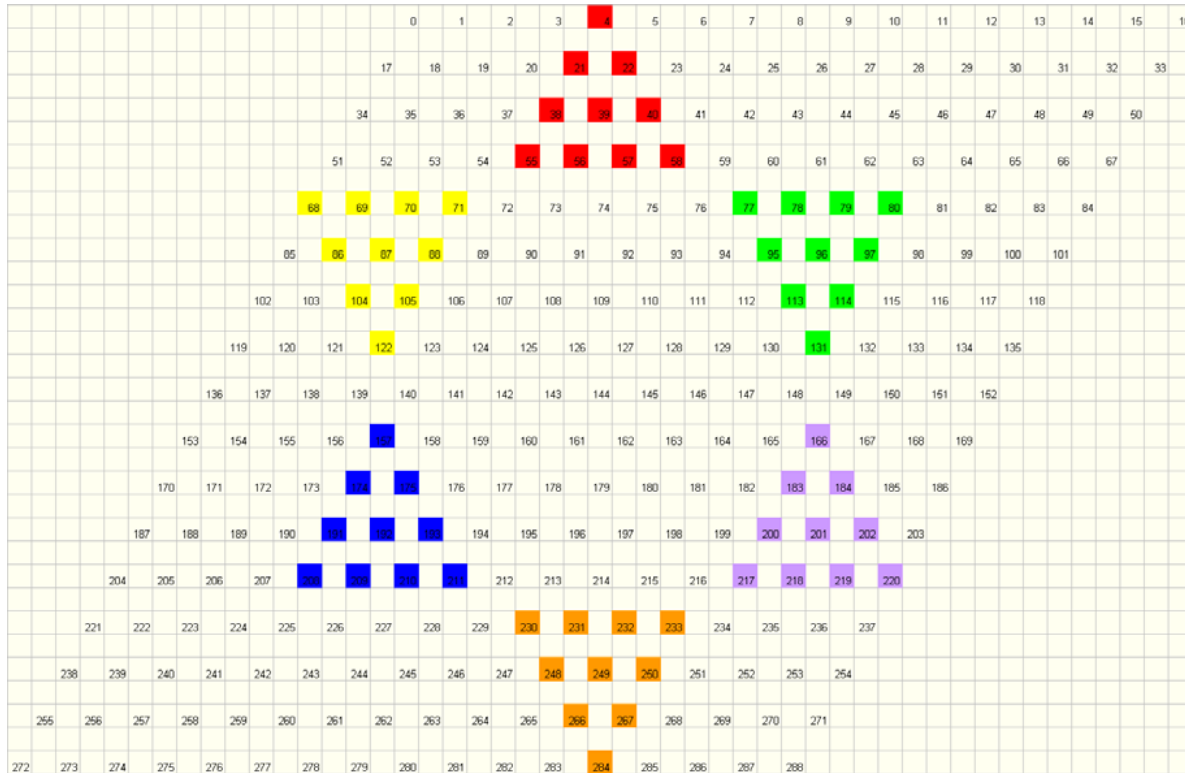
Introduction:

The project is centered on the classic board game of Chinese checkers. There is an option for number of human players, the computer controls the rest. The goal is to create a program that would play good Chinese checkers.

Project Idea Design:

- The project uses GUI(Graphics User Interface)
- The project extensively uses features of object oriented programming.

- There are 4 classes – hole, ball, board and ball-set (player that is).
- The board is represented by a 2d array of holes which are initialized as soon as an object of class board is created. The balls are also initialized at this stage. Shown below is a pictorial representation of the board
- Representation of marbles on the board would be through a 2-D array or a list.
- Representation of marbles on the board would be through a 2-D array or a list. As shown below



- Single and multiplayer options.
- There are two algorithms , one the popular minmax and the other indigenous(based on our experience of Chinese checkers)
- The minmax algorithm first creates a search tree of all sets of possible moves upto next 3 players (the number can be changed) and then evaluates current player's position according to which it returns the most profitable move.
- The other algorithm considers the next 2 moves of the same player (using the normal list functions) and returns the best-move. This is the normal way we play checkers we think two three moves ahead about our own move so as to make a long chain of jumps
- Initially the program uses the second algorithm and after a certain stage it switches to minmax.

Sample Input Output:

- The project being a GUI interface would be self explanatory and would not require any sample input output!

Bugs

- The graphics library fails to respond sometimes.

Future Improvisations

- Feasibility: Thought about coding an adoptive i.e. self-learning strategy. However , abandoned the idea as it seemed too ambitious and out of reach. The idea of alpha-beta pruning was also ditched as it proved too be too costly in terms of time!!.
- Implementing better algorithms.
- There can be more improvement in the graphics part. We can have better interface.
- We can have a computer versus human one-on-one game.

Acknowledgements

Many people have put their efforts in this project . At this stage we want to thank them for all the pain they took .

Vibhor Kanojia: for making eye-catching images in photoshop.

Sushant , Ayush ,Sanchit ,Anmol,Himanshu,Dhanesh: for various random helps

Anand: helping us with graphics library.

Ashish: The debugging show function!!