

# CS PROJECT

Nama N V S S Hari Krishna  
Chagam Dileep Kumar Reddy  
Ashrut Vaddela

170050077  
170050080  
170050079

## DESCRIPTION OF THE PROBLEM :

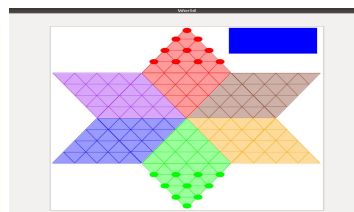
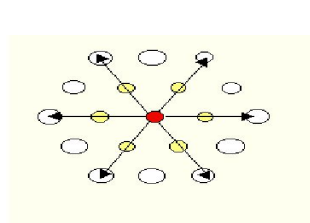
The main motto of the project is to simulate a basic model of **chinese checkers game**. Here we can choose to play either two player or single player game.

### INTRODUCTION TO GAME :

- 1 or 2 players can play the game ,having 10 same-colored marbles
- At the start - player's marbles are in the ten holes of the star point that has the same color as his marbles
- Goal - move all marbles of one color from starting point to the star point on the opposite side of the board
- No game pieces are removed from the board.

### CONSTRAINTS :

- ☐ marble can move by rolling to a hole next to it.
- ☐ by jumping over one marble, of any color, to a free hole, along the lines connecting the holes in a hexagonal pattern.
- ☐ several jumps in a row, but only one roll.
- ☐ cannot both roll the marble and jump with it at the same turn.



## IMPLEMENTATION IDEA :

### A. Image Part :

We used **2htdp/image** for image part.

**Board** is drawn with predefined functions of teachpack, on which pawns of the players are placed.

We used a struct called **PLAYER** which stores the colour of the pawn and list of positions of the pawns. Image part finally gives frame which contains board with all pawns placed on it and block of skip-move is placed at top-right corner.

### B. Graphics Part :

We used **2htdp/universe** for graphics part.

For the complete game, we only used **mouse events** (for moving pawns), which takes list of PLAYERS as input and also gave list of PLAYERS as output. A predefined function called **big-bang function** is used to change from one frame to other frame on changing with mouse events.

### C. AI FOR SINGLE PLAYER :

#### STATIC SCORE :

One of the most important part of AI, which allows computer to decide.

The method I used to estimate the score is based on considering Vertical, Horizontal displacements, Back-piece moving strategy, sometimes random moves based on opponent positions.

#### MINIMAX FUNCTION:

It takes current depth, alpha, beta, max depth as arguments.

It calls **maximiser** function when it is the turn of player1.

It calls **minimiser** function when it is the turn of player2.

The function recurses till depth reaches max depth where it gives score.

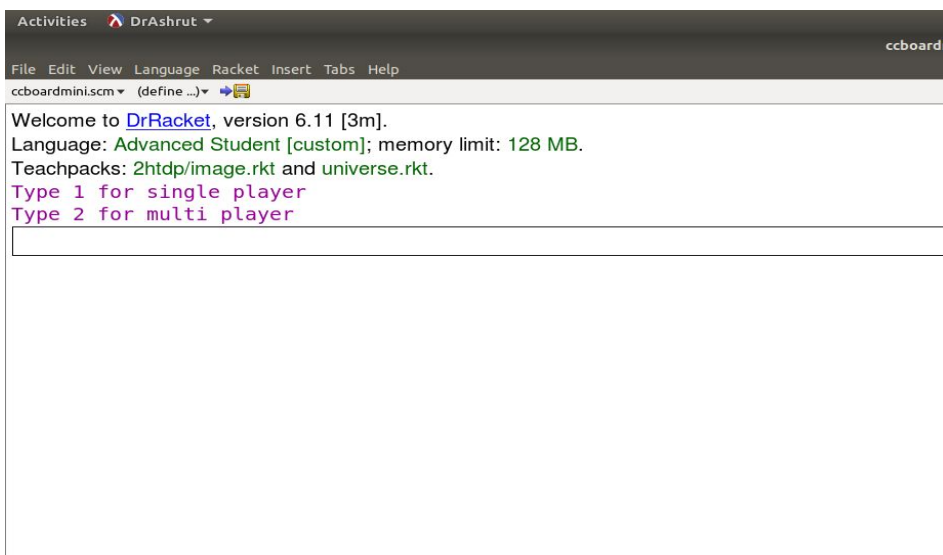
**Alpha beta pruning algorithm** is also used along with minimax.

### BESTMOVE FUNCTION:

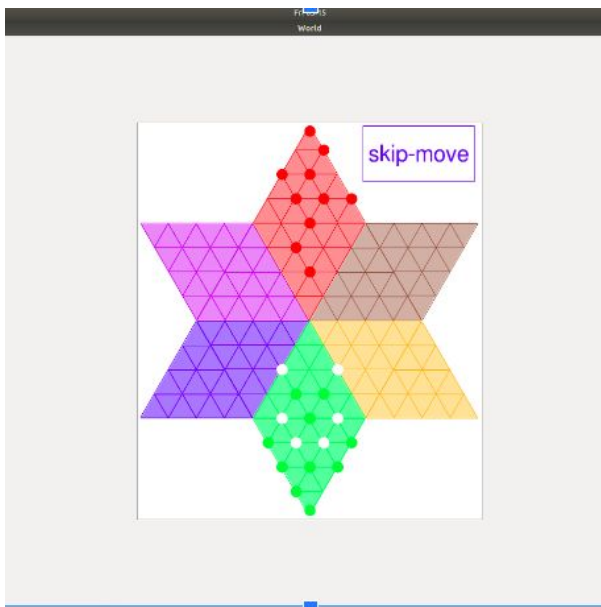
Returns the best move for the player at the current situation.

It uses minimax function to find the best move.

## INPUT AND OUTPUT INSTANCES :



On running the code we are asked to choose for 1 player or 2 player. The game window looks like the one below.



## POINTS OF INTEREST :

### CLEVER CODING:

- **ALPHA-BETA PRUNING:** We implemented this standard method to cut down on the number of leaves in the minimax tree for which the score must be evaluated. It removes the parts of the tree for which we won't get a better value than the current value. This increases the efficiency of minimax.
- **HEURISTICS:** This is an important part of the AI. Back-piece moving strategy ensures that the pieces move in clusters. This is included in the scoring.

## LIMITATIONS AND BUGS :

1. We can run minimax up to a max depth of 3 in a considerable time (within 3 s). It takes around 10 seconds if we check up to a depth of 4.
2. Heuristics or incorporating tactics to the game, is not an easy thing and vary from mind to mind.
3. After completion of each move player has to press on skip-move block placed on top-right corner for the computer to play the move in single player case.

### PACKAGES:

We used image and universe packages.

## REFERENCES :

- Racket documentation.

- [https://en.wikipedia.org/wiki/Chinese\\_checkers](https://en.wikipedia.org/wiki/Chinese_checkers)
- <https://hem.passagen.se/baolan/release/china.pdf>