

**M U N I**  
**F S S**

## **CHECKS POST FACTO**

Shri Vishnu Engineering College For Women

**22B05A1211:K. Varshini: IT.**  
**22B05A0509:M.Lohitha:CSE.**  
**22B05A1207:G.Sangeetha:IT.**  
**21B01A05J9:P.Rashmitha:CSE.**  
**22B05A0516:V.Yamini:CSE.**  
**21B01A05D2:P.Tejaswini:CSE.**

February 10, 2023

# INTRODUCTION

- It played on the dark squares of an 8\*8 checkboard.
- There are two players,Black and white,who alternate turns moving their pieces. **Each piece occupies a single dark square.**
- A normal man can be either a man or promoted king.
- A turn consists of choosing one piece and moving it in one of two ways.

# GAME RULES:

## RULE:1<sup>1</sup>

The pieces always moves diagonally and single pieces are always limited to forward moves.

## RULE:2

If a player is able to make a capture, then the jump must be made.

## RULE:3

**Kings are limited to moving diagonally but can move both forward and backward.**

---

1

# APPROACH

WE USED RATIONAL APPROACH WHICH IS

1. Define the problem.
2. Identify possible causes.
3. Brainstorm options to solve the problem .
4. Select an option.
5. Create an implementation plan.
6. Execute the plan and monitor the results.
7. Evaluate the solution.

## Learnings

**Mini-max algorithm** is a recursive or backtracking algorithm which is used in decision-making and game theory. It provides an optimal move for the player assuming the opponent is also playing optimally. In this algorithm two players play the game, one is called MAX and other is called MIN.

A deep copy is a process where create a new object and add a copy elements recursively. We will use the **deepcopy()** method which present on copy module. The independent copy is created of original object and its entire object.

## Challenges

Over the last five days we have spent many hours programming a complex checkers onto python. Overall, the project has definitely taught us many lessons about organising the variables and use of functions. We have faced many problems in the initial stage to understand the game.

So, we searched and took the reference from the internet and we played the game then we have understood the game and its rules clearly. Then we started writing the program. When it comes to programming the logic of the game, somewhat complicated but it has been fun making the project successfully.

# STATISTICS:

2

Number of lines of code:470

Number of functions: 10

Node()

Checkers()

Getplayerinput()

Findavailablemoves()

Checkjumps()

Checkmoves()

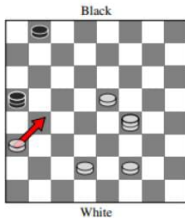
Evaluatestates()

Minmax()

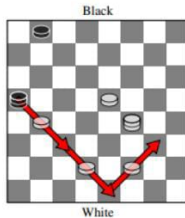
2an

22B05A1211:K. Varshini: IT.22B05A0509:M.Lohitha:CSE. 22B05A1207:G.Sangeetha:IT. 21B01A05J9:P.Rashmitha:CSE. 22B

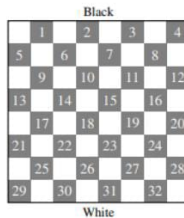
## Demo/Screenshots



(a) Simple move



(b) Jump move



(c) Board numbering



## Demo/Screenshots

**Sample Input 1**

```
W 3
21-17
13x22x31x24
19x28
```

**Sample Output 1**

```
-b-.-.-. -b-.-.-.
.-.---.-. -.---.-.
-.---.-. -.---.-.
B-.w-.-. -.w-.-.
-.--W-. -.---.-.
W-.---.-. -.---.-.
-.w-w-. -.---.-W
.-.---.-. -.---.-.
```

**Sample Input 2**

```
B 5
2-7
9x2
32-27
2x11x18
5-9
```

**Sample Output 2**

```
-.b-.w -.---.-W
b-b-.--- -.---.-
-w-.---. -b-.---.
B-w-b-. B-w-.---
-.---.-. -.W-.-.
.-.---.-. -.---.-
-.---.-. -.---.-B-
.-.---B- -.---.-
```

# Demo/Screenshots

```
#### WELCOME TO CHECKERS ####
```

Some basic rules:

- 1.You enter the coordinates in the form i,j.
  - 2.You can quit the game at any time by pressing enter.
  - 3.You can surrender at any time by pressing 's'.
- Now that you've familiarized yourself with the rules, enjoy!

First, we need to know, is jumping mandatory?[Y/n]: **y**

```
0 | - w01 - w03 - w05 - w07
1 |w10 - w12 - w14 - w16 -
2 | - w21 - w23 - w25 - w27
3 | - - - - - - -
4 | - - - - - - -
5 |b50 - b52 - b54 - b56 -
6 | - b61 - b63 - b65 - b67
7 |b70 - b72 - b74 - b76 -

    0  1  2  3  4  5  6  7
```

Player's turn.

Which piece[i,j]: **5,2**

Where to[i,j]: **4,0**

22B05A1211-K. Varshini:IT,22B05A0509-M.Lohitha:CSE, 22B05A1207-G.Sangeetha:IT, 21B01A05J9:P.Rashmitha:CSE, 22B

■ *THANK YOU*