Fast

Nananan NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

(KARACHI CAMPUS)

Department of Computer Science

**Spring 2022**

## Project Proposal: Checkers

**Objective:**

Parallel programming on an SMP and a multicomputer, using OpenMP for Parallel and Distributed Computing course. For this purpose, we are going to develop a serial and parallel implementation of a Checkers Game.

**Introduction:**

Checkers is a game with an 8x8, with place and white squares places alternatively on the grid or board grid played by 2 players on opposite sides with black or white pieces. The goal of each player is to eliminate the other players pieces by moving their pieces over the spaces with the other players’.

**Technologies/tools:**

- C programming language

- Ubuntu (Linux based OS)

- OpenMP for parallel programming

- Different C libraries for making

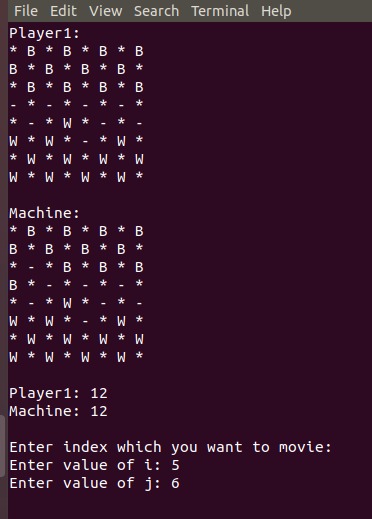
**Methodology:**

There are some set rules and constraints of the game “Checkers” which were needed to be taken into consideration before starting the coding process.

* Pieces can only move diagonally.
* Pieces can only move on the black squares.
* Each piece can only move one space at a time other than when they are capturing.
* To capture a piece the player has to move their piece over the opponent’s piece with no piece blocking it.

After we set the rules that the pieces and players will follow throughout the game, we started with a grid and exception handling that no one can make a move out of the board. Then we started making the prototype of pieces and how they are going to move. The player can move by entering the values of the x-axis and y-axis of the piece they want to move. Then we made the program to respond to the users moves. In the end we made the board more appealing to the player so they are able to understand the game better with “B” for the black pieces of the game, “W” for the white pieces, “-” for the empty spaces where the pieces can be moved on, “\*” for the blocked places on the board that none of the players can move to. We made the capture mechanism is a way that if a piece can jump over the opponent’s piece then it be would capture, otherwise it will not move in that direction. We exception handled it so there is no unwanted moves happening like capturing more than 1 piece in a move by skipping over 2 pieces at once or jumping outside the board. The AI is programmed in a way that it will check for all the pieces both its own and the opponents to make the best move possible. It does prioritizes on saving a piece over capturing opponents pieces if given the option.

**Graph:**



**Application:**

The main application for this project and its AI is for the player to learn the basics of the game “Checkers” before actually playing against a human opponent. This application can be also used to practice before playing with other human players. The AI that is used in this application is difficult to beat even for a more experienced player. It can further be modified to be played in a 2-player mode where both black and white sides can move operated by the users taken turns in entering inputs for their respective pieces.

**Conclusion:**

While creating this project we were learning about parallel distributed computing which we applied in this project to reduce process time by applying elements of PDC in our project. This did not make it easier in finishing the project faster or reduce the lines of code but significantly reduce the time it took for the AI to decide what move to make next. The AI also checks the positions of all the pieces on the board every time it needs to make a move. So applying elements of PDC we reduced that time and almost made it instantaneous to process is decision making time.