Checkers But Chess

By Angelica Miranda Luna, Jeffrey Yu, Zachary Qin, Kunal Paode, Tommy Kung

Table of Contents

Glossary

1. AI (Artificial intelligence)

   -the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, decision-making, etc., …

2. GUI (Graphical User Interface)

   -an interface that uses icons/menus, and a mouse to manage interaction with the system

3. Doubly linked list

   -a linked data structure that consists of nodes that have pointers to the previous and next node. In which navigation is possible in both directions.

4. Enum

   -a data type that allows assignment of naming conventions to constants

5. Struct

   -structure is another user defined data type available in C that allows to combine data items of different kinds.

6. King ♔
   Slow piece that can move only one step in every direction – forward, backward, to the sides or diagonally. The King can capture any of the opponent's pieces that are standing in any square surrounding the King.

7. Queen ♕
   Can move in every direction – horizontally, vertically and diagonally. Unlike the King, however, the Queen can move in a straight line all the way to the other side of the board, stepping on every square that isn't taken up by another piece – making her the most powerful piece in the game of Chess. The Queen can capture a piece by landing directly on the other player's square.

8. Bishop ♗
   Can only move diagonally and step on any square that isn't taken up by another piece. If a Bishop starts the game on a Black square, he will only be able to step on Black squares for the rest of the game, and the same goes for a Bishop that starts the game on a White square. Similarly to the other pieces, the Bishop can only capture an opponent's piece by landing on the square that the piece is standing on.

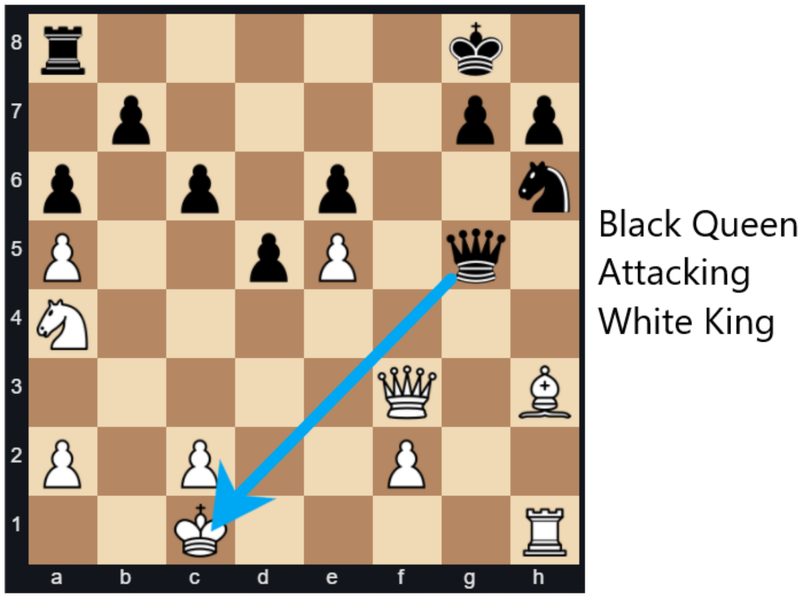
9. Knight ♘
   Can move two squares forward or backward and one square to the side, or two squares to the side and one square forward or backward, so that his movements resemble the shape

of an L. The Knight is the only piece in the game of Chess that can skip over the other pieces when it moves. Even though he can skip over squares while they are occupied by other pieces, the Knight can only capture a piece that is standing on the square he lands on.

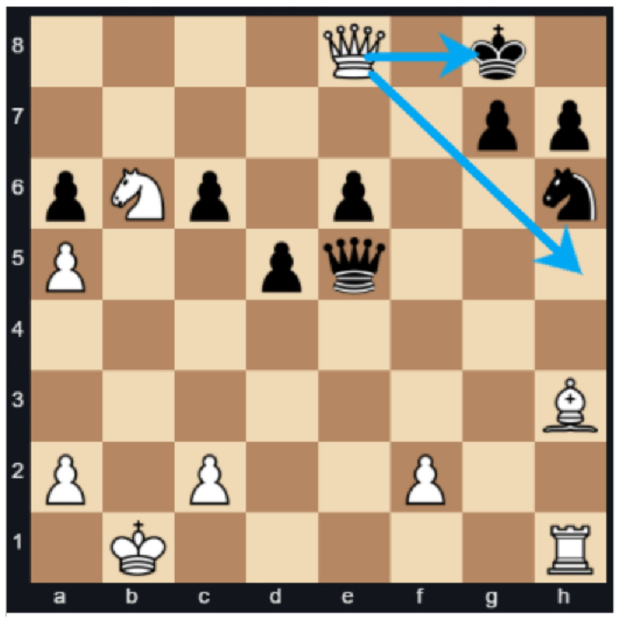
10. Rook ♖
    Can move in a straight line forwards and backwards through any square on the board that isn't occupied by another piece. To capture the other player's pieces, the Rook needs to land directly on the piece's square.

11. Pawn ♙
    Can only move forwards one step at a time. When they capture the other pieces they can only do so when the opponent's piece is on a square diagonally in front of them. If another piece is standing in front of the Pawn, he will not be able to keep moving. Only when they first move from the starting position, they can choose to jump over the square immediately in front of them, therefore moving forward by two squares. If a Pawn manages to reach the other side of the board he can be promoted to any piece of his liking except the King. Once the Pawn is promoted, he can move in the same way as the piece he was promoted to moves.

12. Check
    When the player's king is under the attack of a piece. The player with the king that is in check must get out of check before being able to move otherwise.

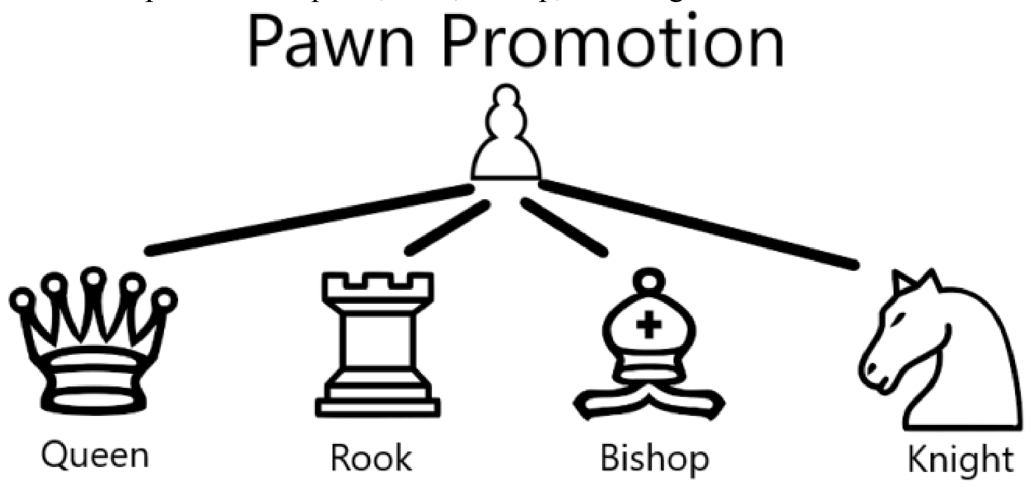


13. Checkmate
    When a player's king has no possible escape while they are in check.

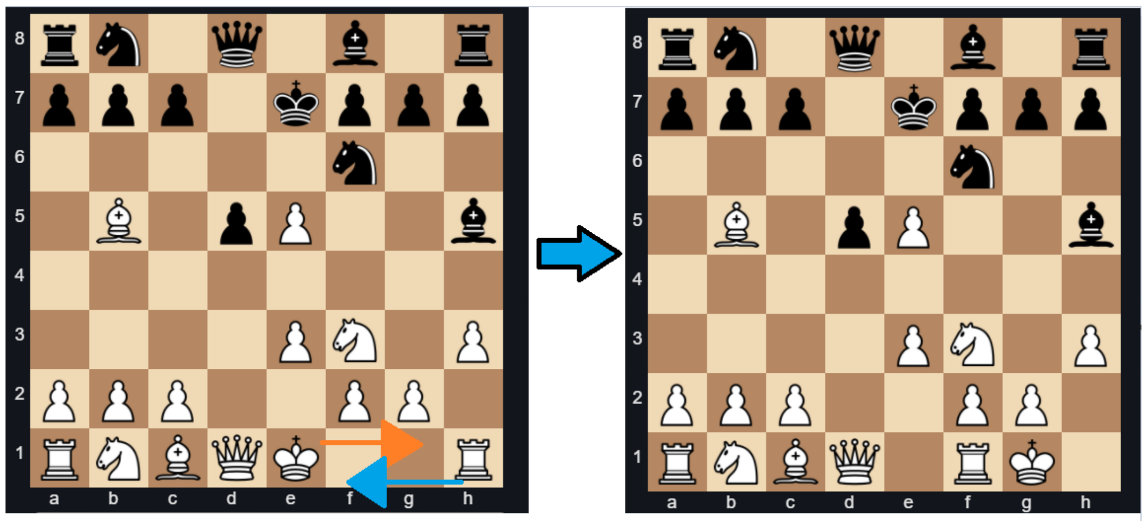

White Queen attacks Black King.
White Queen also blocks escape route.
8
7
6
5
4
3
2
1
a
b
c
d
e
f
g
h

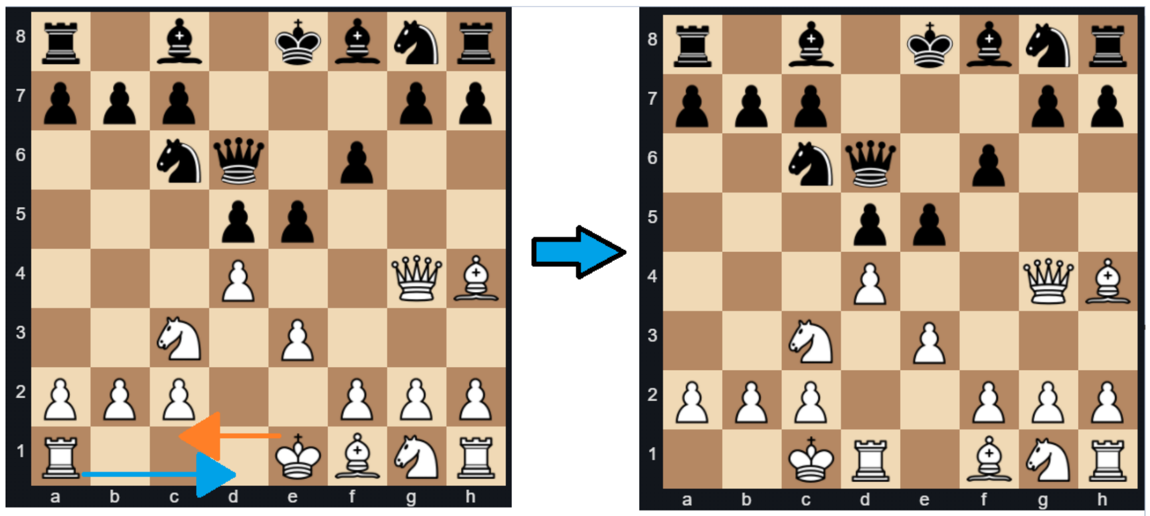
14. Resign
    When a player concedes the game.

15. Pawn Promotion
    When a pawn advances to the 8th rank, the end of the board, the player can choose to promote the pawn into a queen, rook, bishop, or a knight.

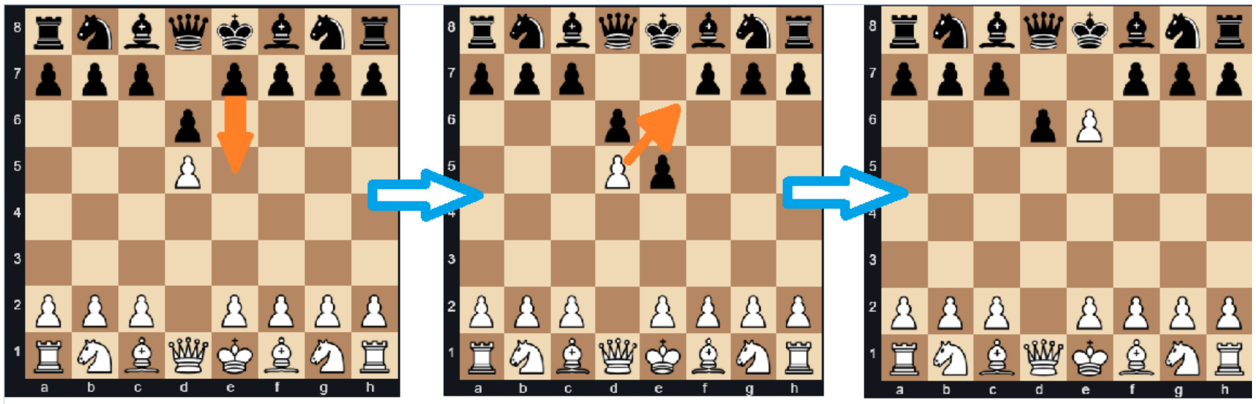


16. Castling
    Moving the king two squares toward a rook, then placing the rook on the other side , adjacent to it. The conditions being the king and rook involved must not have moved previously to castling. There are no pieces in between the rook and the king. The king cannot be currently in check or will end up in a check position.


Ex. Castling to the right


Ex. Castling to the left

17. <u>En Passant</u>
    When a pawn advances two squares from its original square and ends the turn adjacent to a pawn of the opponent's on the same rank. The pawn can capture the adjacent pawn.



Ex. En Passant

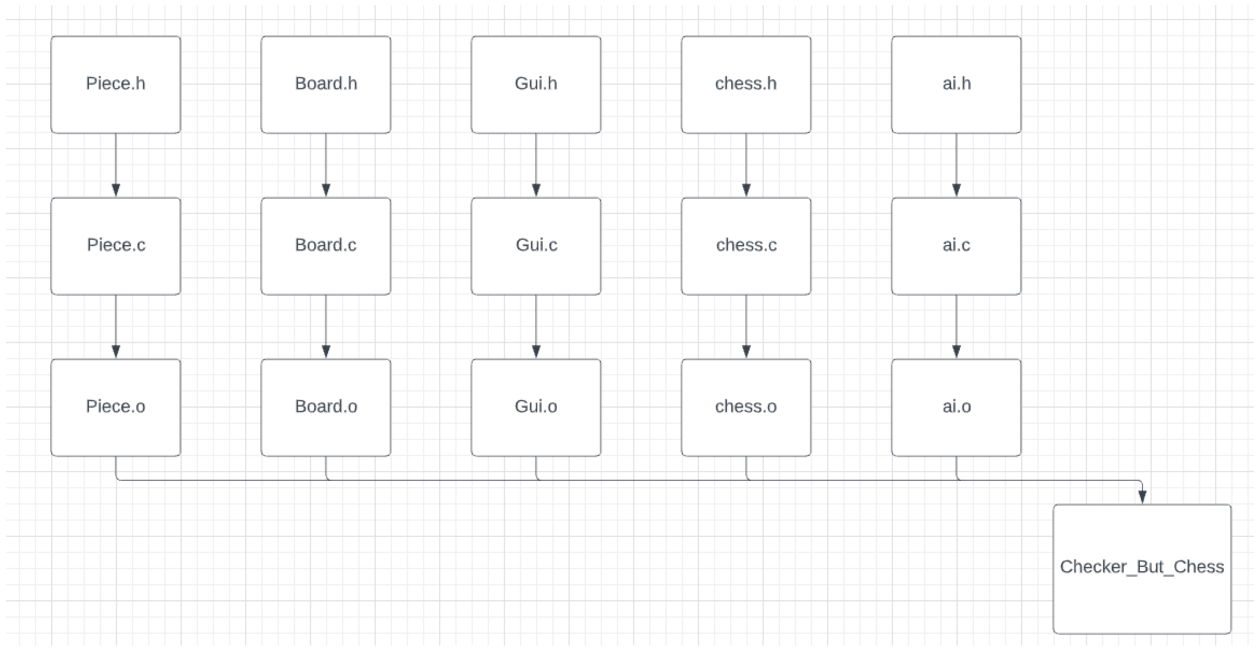
Software Architecture Overview

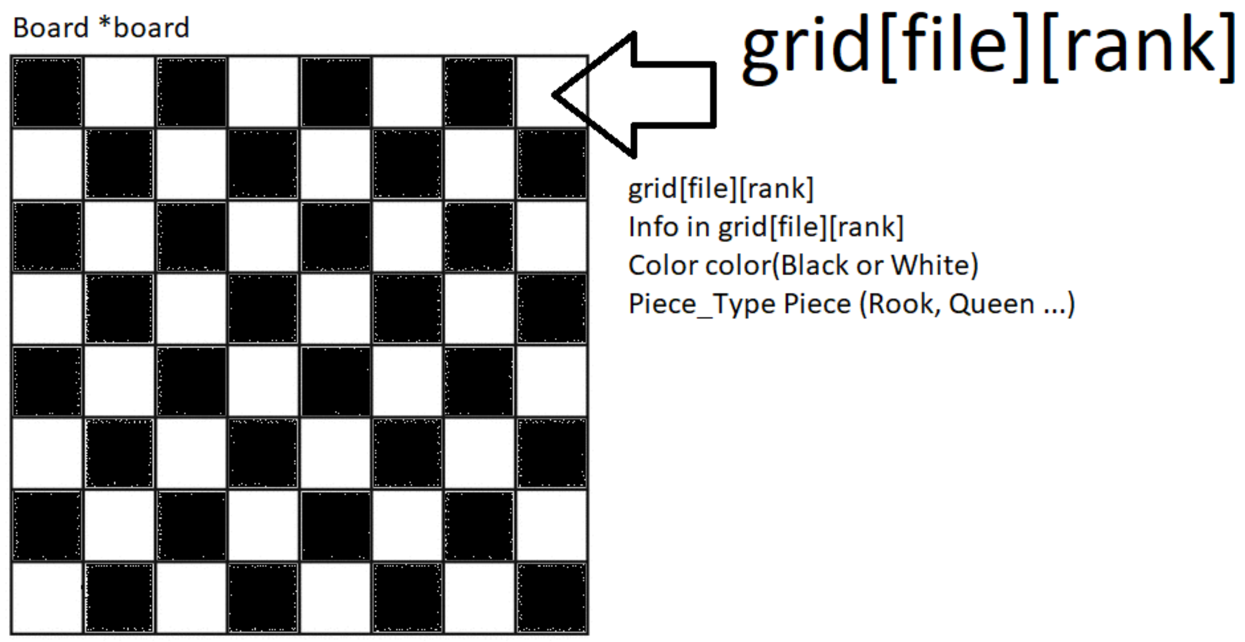
1.1 Main data types and structures

Main enums:
enum Piece_type
enum Color
enum Rank
enum File

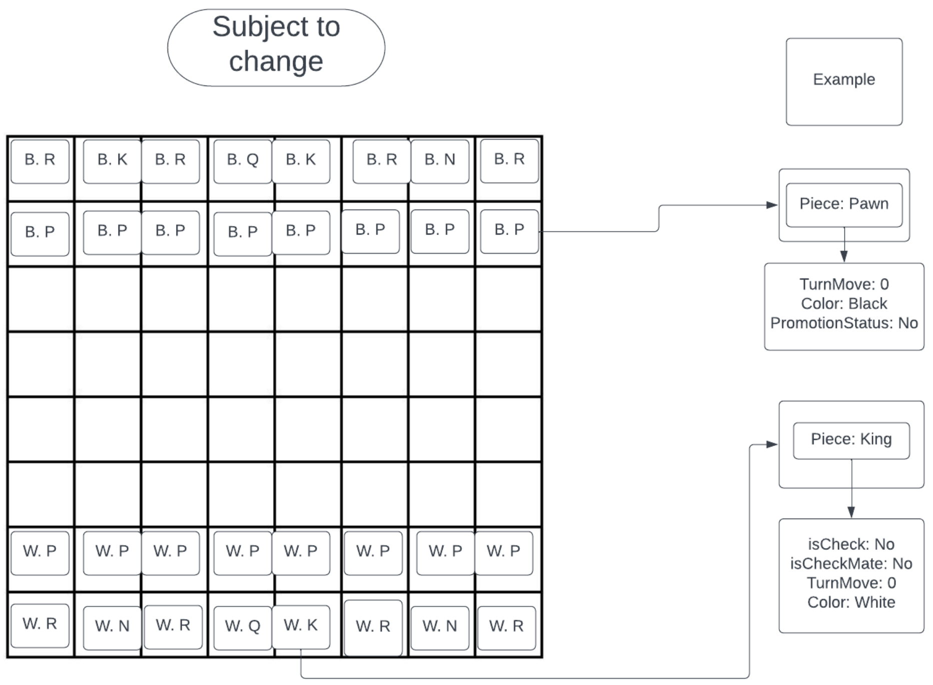
Main structs:
struct Piece
struct Position
struct Board
struct ChessGame
struct AI
struct Move
Struct PositionList
Struct MoveList

1.2 Major software components

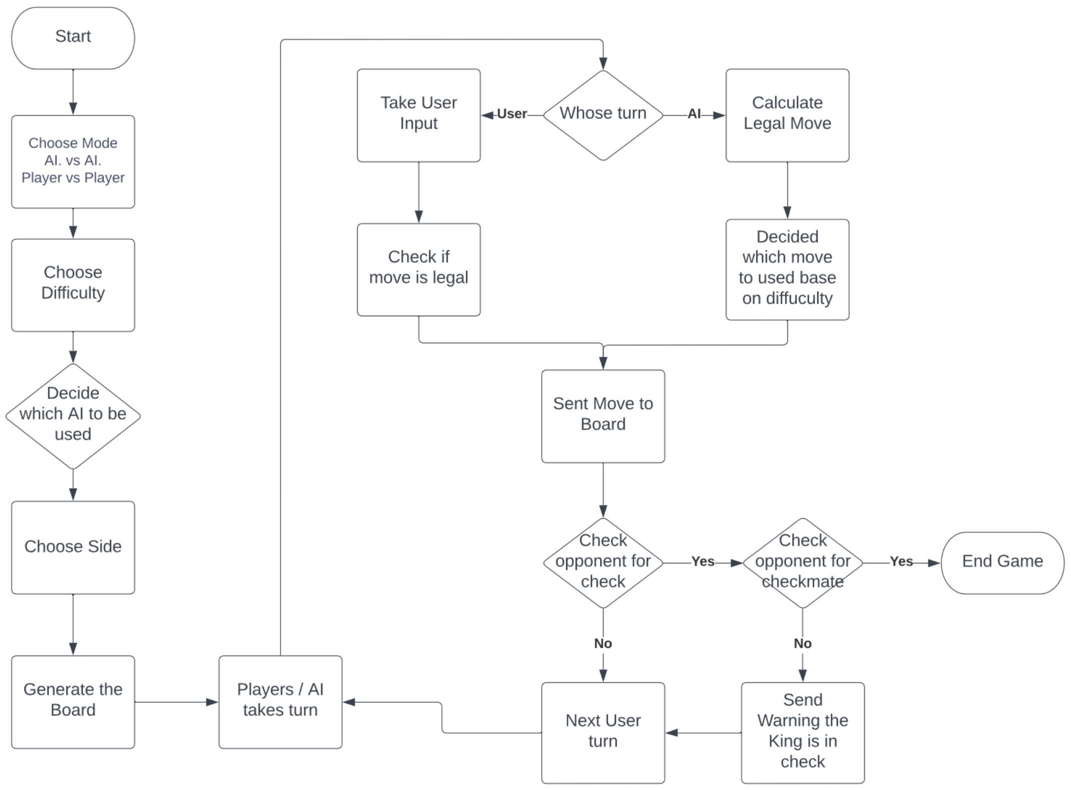




1.3 Module interfaces



1.4 Overall program control flow



Installation

2.1 System requirements

Linux, std=c11, gtk library

2.2 Setup and configuration

Obtain Linux environment with C11
Download tar.gz file and Makefile
Extract tar.gz file with gtar xvzf tar.gz file

2.3 Building, compiling and installation

[Linux] Type “make all” to build program
[Linux] Type “./Checkers_But_Chess”

2.4 Uninstalling

[Linux] Type “make clean”
[Linux] Type “rm Makefile”

Documentation of Packages

3.1 Data structures

We use an 8 by 8 array to store the struct of Pieces inside. Inside each element tell us the position, color, and piece type such as a White Rook on A8. We can then send the board to another function to operate on the pieces in the array.

3.2 Functions and parameters

Board Module Enums

```
typedef struct Board {
    //[file][rank]
    Piece grid[8][8];

    //[file]
    Bool blackUpForEnPassant[8];
    Bool whiteUpForEnPassant[8];

    Bool whiteKingMoved;
    Bool blackKingMoved;
    Bool whiteRookMoved[2]; // = {left, right}
    Bool blackRookMoved[2]; // = {left, right}
} Board;
```

Notable Board Module Functions:

```
int SmartMovePiece(Position initialPosition, Position finalPosition, Board*
board, Piece_type promotedType);

int Rules(Position initialPosition, Position finalPosition, Board* board);
```

```
Bool CheckForCheck(Position kingPosition, Board *board);

Bool CheckForCheckMate(Position kingPosition, Board *board);

MoveList GetAllMoves(Color teamColor, Board *board);
```

Piece Module Enums

```
typedef enum Piece_type {
    king,
    queen,
    bishop,
    knight,
    rook,
    pawn,
    empty
} Piece_type;

typedef enum Color {
    black,
    white,
    no_color
} Color;

typedef struct Piece {
    Piece_type type;
    Color color;
} Piece;
```

```
typedef enum Rank {
    one = 0,
    two = 1,
    three = 2,
    four = 3,
    five = 4,
    six = 5,
    seven = 6,
    eight = 7,
    unknownRank = 8
} Rank;

typedef enum File {
    a, b, c, d, e, f, g, h, unknownFile
} File;

typedef struct Position {
    Rank rank;
    File file;
} Position;

typedef struct Move {
    Position iPos; //initial position
    Position fPos; //final position
} Move;

typedef struct PositionList {
    Position* list;
    int size;
} PositionList;

typedef struct MoveList {
    Move* list;
    int size;
} MoveList;
```

Notable Ai Functions:

```
int minimax(Board* board, int depth, Color teamColor)

Move createBestMove(Board *board, int depth, Color teamColor);
```

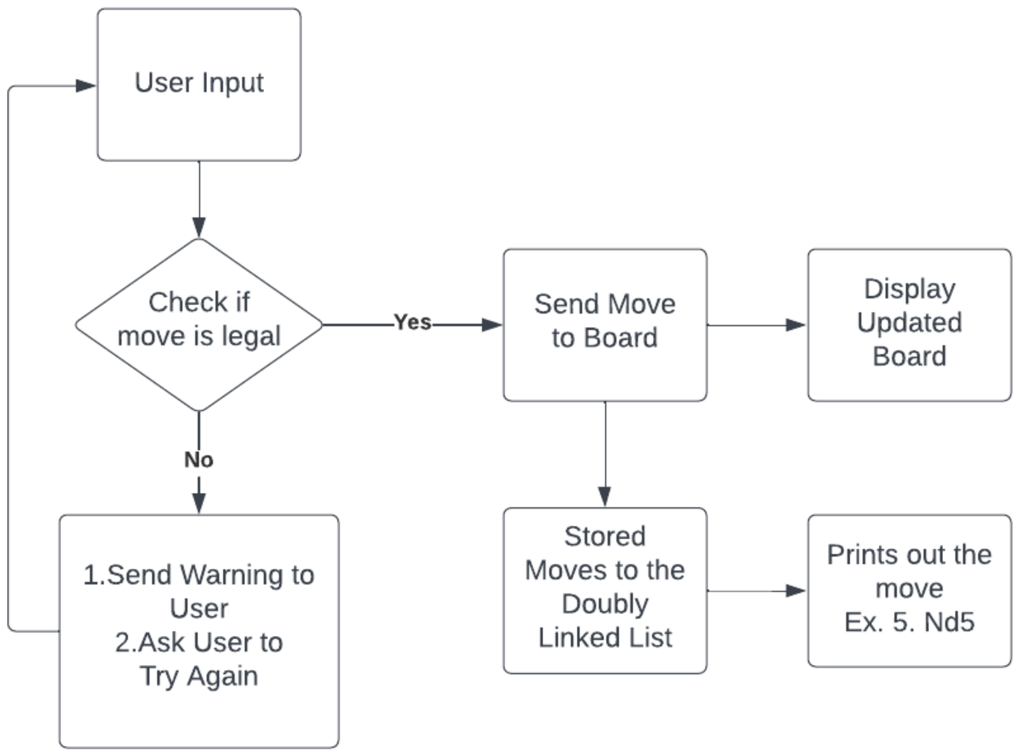
Notable GUI Functions: Created the color board and assign the beginning position of the chess

```
int counterpiecearray = 0;

  int i, j;
    for(j = 0; j< 8; j ++)
  {
    for(i = 0; i < 8; i ++)
    {
      piecelabel = (GtkLabel *) gtk_label_new(asciipieces[counterpiecearray]);
      gtk_widget_set_size_request((GtkWidget *) piecelabel, 50, 50); //size of
ascii to be 35,35 to fit into 50,50 square
      originboard[i][j] = piecelabel;
      chessbox = gtk_event_box_new();

      switch(BoardWhiteBlack[i][j])
      {
        case BLACK:
          gdk_color_parse("#37b04d", &bgcolor);
          gtk_widget_modify_bg((GtkWidget *)chessbox, GTK_STATE_NORMAL,
&bgcolor);


          break;
        case WHITE:
          gdk_color_parse("#94e1a2", &bgcolor);
          gtk_widget_modify_bg((GtkWidget *)chessbox, GTK_STATE_NORMAL,
&bgcolor);
          break;
        default:
          break;

      }

      gtk_event_box_set_above_child(GTK_EVENT_BOX(chessbox), FALSE);
      //Put the chess unicode into the chessbox
      gtk_container_add(GTK_CONTAINER(chessbox), (GtkWidget *) piecelabel);
      //Add the chessbox inot the table
      gtk_table_attach(GTK_TABLE(table), (GtkWidget *) chessbox, i, i + 1, j, j +
1, GTK_FILL, GTK_FILL, 0, 0) ;


      //Connects signal when the individual squares are pressed
      g_signal_connect(G_OBJECT(chessbox), "button_press_event", G_CALLBACK
(chess_board_pressed), (gpointer) originboard);

      gtk_widget_set_events(chessbox, GDK_BUTTON_PRESS_MASK);



      counterpiecearray++;//Increment to access the next char
```

3.3 Input and output formats

For inputs by the user, the user would click on the original position and a click on the final position with mouse inputs.

The log will save all the moves that both players have moved in the game. When the player clicks on the positions, it will display the moves into the log. The format of the moves would be starting position, ending position b2 to b4. It would tell who is in check in the log.


User Input
Check if move is legal
Yes
No
Send Move to Board
Display Updated Board
1.Send Warning to User
2.Ask User to Try Again
Stored Moves to the Doubly Linked List
Prints out the move
Ex. 5. Nd5

Development Plan

4.1 Partitioning of tasks

Week	Goals
1	User Manual
2	Software Specifications Begin initial development of data structures - Pieces - Board - Moving pieces, etc Draw board Rough Draft of user v. user game
3	Have board completed Start with the implementation of the AI
4	Start the testing of the game Modify any last-minute changes

4.2 Team member responsibilities

Team member	Responsibility
Jeffrey	GUI
Zachary	Chess Piece Rules, AI, I/O
Kunal	Board module, AI, Chess Piece Rules
Tommy	GUI, Connecting Game to GUI
Angelica	AI, Chess Piece Rules

Back Matter

Copyright

References

1. Hyatt, Robert. "Chess program board representations." February 28, 2022.
2. https://developer-old.gnome.org/gtk2/stable/index.html (For GTK library)
3. https://www.gtk.org/ (For GTK library)
4. https://www.youtube.com/watch?v=l-hh51ncgDI&t=333s (For minimax chess ai)

Index

1. GUI: Graphical User Interface
   Def. An interface that uses icons/menus and a mouse to manage interaction with the system
2. AI: Artificial intelligence
   Def. The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, decision-making, etc., …