Chess Manual

(Developer Edition)

v1.3

By: ZotMeUp© (Team 7)

Development Team:
Anson Do
Arian Reyes
Xianzhang Li
Adrian Gomez
Kevin Huang

Affiliation:
University of California Irvine
Department of EECS

Table of Contents

Glossary

A

AI: AI stands for Artificial Intelligence. This essentially means that a computer will be trained to perform the best moves for any given turn.

B

Bishop: A chess piece that can move diagonally across the board.

C

Castling: A special move in the game of chess involving the king and either of the original rooks of the same color. It is the only move in chess (except promotion) in which a player moves two pieces at the same time. Castling consists of moving the king two squares towards a rook on the player's first rank (row), then moving the rook onto the square over which the king crossed. Castling can only be done if the king has never moved, the rook involved has never moved, the squares between the king and the rook involved are not.

Check: If a King is threatened with capture, but has a means to escape, then it is said to be in check.

Checkmate: When a King cannot avoid capture then it is checkmated and the game is immediately over.

E

En passant: A special pawn capture which can occur immediately after a player moves a pawn two squares forward from its starting position, and an enemy pawn could have captured it had the same pawn moved only one square forward. The opponent captures the just-moved pawn as if taking it "as it passes" through the first square. The resulting position is the same as if the pawn had moved only one square forward and the enemy pawn had captured normally. The en passant capture must be done on the very next turn, or the right to do so is lost. Such a move is the only occasion in chess in which a piece captures but does not move to the square of the captured piece. If an en passant capture is the only legal move available, it must be made.

G

GUI: GUI stands for Graphical User Interface. This is the interface the user will see to navigate through the different menus the game offers.

K

King: A chess piece that can move in any direction, but only one step at a time. Also, the king must never move into check. There is also a special "castling" move for the king.

Knight: A chess piece that can jump to eight different squares which are two steps forward plus one step sideways from its current position.

M

Main: This essentially is a function that will hold all the function calls that will run the entire program.

P

Pawn: A chess piece that can move only forward towards the end of the board, but captures sideways. From its initial position, a pawn may make two steps, otherwise only a single step at a time. If the pawn reaches the end of the board, it is automatically promoted to another piece (usually a queen). There is also a special "en passant" move for the pawn.

Q

Queen: A chess piece that can move horizontally, vertically, and diagonally across the board.

R

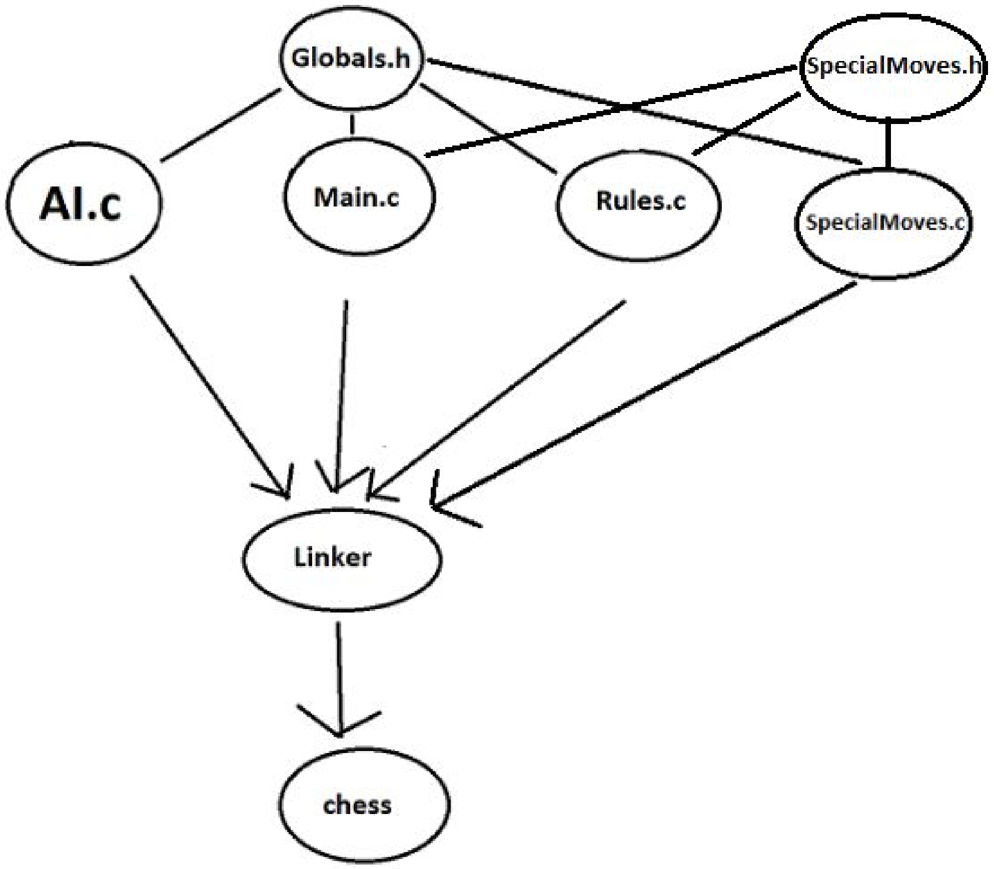
Rook: A chess piece that can move horizontally and vertically across the board.

1. Software Architecture Overview

1.1 Main data types and structures

The chessboard is made out of a 2D array.
The position and color of the pieces are stored in structs.

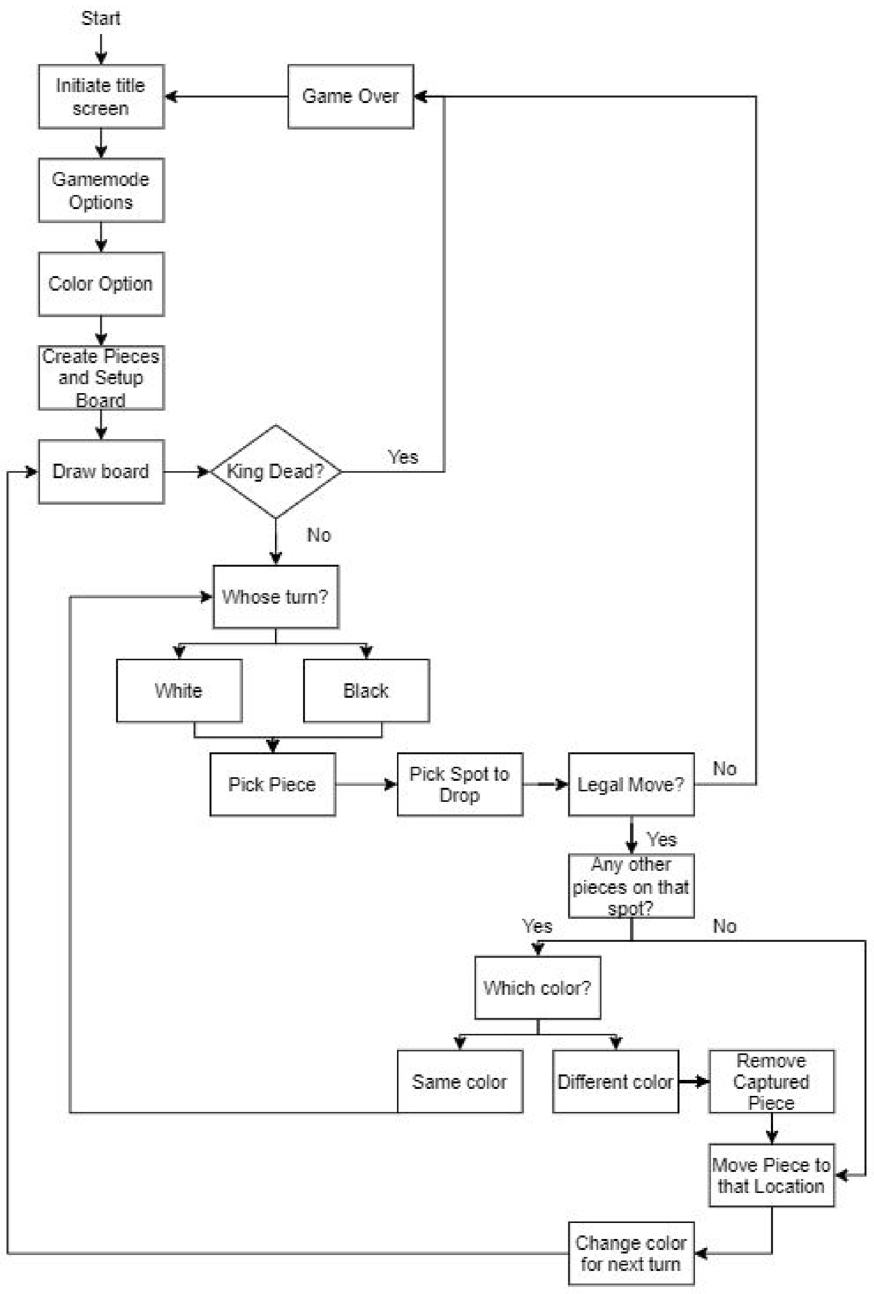
1.2 Major software components



1.3 Module interfaces

- AI: Contains the AI. This will allow the user to play against an AI or have an AI play against the same AI.
- Rules: Contains the rules for each piece. This allows the program to check if a move is legal, check for checks and checkmates, movement, capturing, etc.
- SpecialMoves: Contains functions for king checking and checkmate. These functions can be used for castling, ending the game, etc.
- Main: Contains the GUI, 2D array of the chessboard and pieces, the movement of the pieces, and calls upon other files for functions.

1.4 Overall program control flow



2. Installation

2.1 System Requirements

Linux OS

A keyboard and monitor

Intel CPU(2GHz or above)

512MB of Ram(Minimum 256MB)

350MB of Disk Space

2.2 Setup and configuration

1. Connecting to the UCI Linux server with your account, you may need the UCI VPN if you are out of campus.

2. Using the “cd” command to change the directory to find the file.

3. Extract the file with the command "tar -xvzf Chess_V1.0_src.tar.gz".

4. To compile the executable do "make all"

5. To compile and run the executable do "make test"

2.3 Uninstalling

1. Using “cd” to find the fold where the file is.

2. Using “rm -r Chess” to delete the file.

OR make clean

3. Documentation of Packages, Modules, and Interfaces

3.1 Detailed description of Data Structures

Note: When a user relocates a piece, the new location will be updated by a global array located in Global.h. From here all the C Files access this array to make some manipulation to it.

1. Initializing the board boardstate[8][8] with the side the player choses.
2. Create a new Window in GTK with boardstate to simulate the chessboard.
3. Every Click will now update the boardstate array/ perform a valid move check.

Data Structures for BoardState:

```
        int bboardstate[8][8] =
                        {{-4,-2,-3,-6,-5,-3,-2,-4}, // Initial Board State
                        {-1,-1,-1,-1,-1,-1,-1,-1},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {1, 1, 1, 1, 1, 1, 1, 1},
                        { 4, 2, 3, 6, 5, 3, 2, 4}};
int wboardstate[8][8] =
                         {{4,2,3,5,6,3,2,4}, // Initial Board State
                        {1, 1, 1, 1, 1, 1, 1, 1},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {-1, -1, -1, -1, -1, -1, -1, -1},
                        { -4, -2, -3, -5, -6, -3, -2, -4}};
```

With the array above, we can see that the board is initialized with values ranging from -6 to 6. These values represent pieces, ie: 1 acting as pawn, 2 acting as Knight, 3 acting as Bishop, 4 acting as Rook, 5 acting as Queen, and 6 acting as King. The positive or negative values correlate to what color they are. We can see that the bottom is positive, which represents black, negative will in turn be white.

These are the structures for the each Black piece:
Structure for the Chess Board:

```
struct _Cell {
	int x; //board coordinate
	int y; //board coordinate
	GtkImage* image; // containing the image for a given image of a piece
};
```

3.2 Detailed description of functions and parameters

1. Each piece will have their own function for determining if their second click is a valid move
    a. Functions in Rules.c
        i. void CheckPossibleMove(int x, int y, int piece)
        ii. int CheckMoveValid(int x, int y, int piece)
    b. Functions in Main.c
        i. void parraytransfer(int x)
        ii. void logprint(int datastate, int x, int y)
        iii. int *handle_button_click(Cell* cell, GtkButton* button)
        iv. void createChessboard (int boardstate[][8])
        v. void gamemode_select(GtkWidget *item, gpointer data)
        vi. static GtkWidget *make_menu_item (gchar *name, GCallback callback, gpointer data)
        vii. void player1_side(GtkWidget *widget, gpointer data)
        viii. static void createSetup(void)
        ix. int main(int   argc, char *argv[])
    c. Functions in SpecialMoves.c
        i. int kingCheck(int x, int y, int piece)
        ii. int checkMate(int piece)
    d. Explanation for Rules.c
        i. Here we send an x, y value from the first click to fill an array with valid movesets with a given piece value depending on that piece's location on the chessboard.
        ii. Here we send the second click with the same piece. This compares if the x,y coordinate is true within the array used in CheckPossibleMove. This will also check if there are any pieces blocking the moving piece, if it goes over the blocking piece, this will make it an unvalid move. This function returns 0 for invalid move, and 1 for a valid move.
    e. Explanation for Main.c

i. Transfers the piece into a string so it can be printed into the Log file for the user.
ii. This function essentially takes in 2 array coordinates and switches the x coordinate into a character and
iii. This is the bulk of where the program will run. There are multiple versions in order to handle which feature the user selects. For instance, there are two for AI and two for player versus player. Each has their own purpose. For AI, rather than looping for player clicks, it calls the AI function to update the array in which the pieces are located. For player versus player, it handles the click the user inputs and calls the CheckPossibleMove to create an array of possible moves for the piece that was selected. After, it will compare the second click with the array with other rules and decide if it's a legal move or not. After, it will update the image in the button, write the movement into the log file and switch the playerstate flag.
iv. This function essentially makes a chessboard window from a two dimensional array that was set up from player1_side.
v. This function takes the input from the user to determine what gamemode was chosen. From here it sets the flag to determine which handle_button_click to use.
vi. This function makes the drop down menu.
vii. This function takes the user input from black/white and updates a third two dimensional array to set up the creation of the chessboard window. This function also determines whether the white AI makes a first move before the chessboard is created.
viii. This makes the GUI menus when called from main.
ix. This handles the initiation of GTK and creates the GUI.

f. Explanation for SpecialMoves.c
    i. This function takes in an x,y coordinate and piece type to decide if the king is in check or not.
    ii. This function takes in the King piece value and decides whether the king is in checkmate or not.

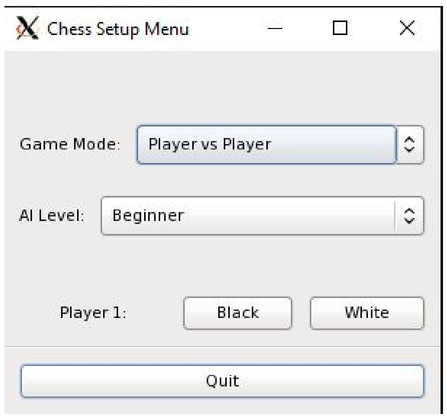
3.3 Detailed description of input and output format

User Input:

- The user will select the color of the board through the GUI.
- Their input will be converted into an integer. Depending on the selection, an initialized board will be sent to boardstate for the entirety of the game.
- Users are also able to select the quit option.

User Output:

- Will display an updated board each time user makes a legal play
- If there was a game winning play, or an illegal move, the windows will terminate but the log file will be updated with all the plays made.


Chess Setup Menu

Game Mode: Player vs Player

AI Level: Beginner

Figure a.

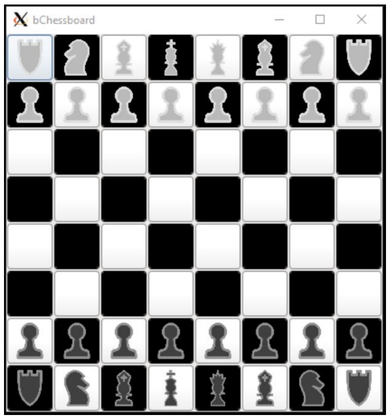


Figure b.

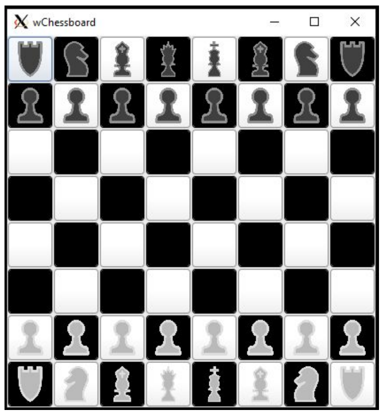


Figure c.

Figure a, initially shows the menu for the user to select which side of the board they choose to play. The user has the option to select the white side or black side. The menu also shows the game mode they wish to play either player vs player or AI vs . player. Additionally, they have the option to select a possible new feature within the game. Figure b and c, show the two different board colors.

MAKE SURE NOT TO MOVE CHESSBOARD WINDOW. IN CERTAIN SCENARIOS A NEW WINDOW WILL BE MADE IN PLACE OF THE OLD CHESSBOARD.

A NEW BOARD WILL BE PRINTED IN PvP IF THERE A PLAYER MADE A SPECIAL MOVE.

A NEW BOARD WILL BE PRINTED IN PvAI WHENEVER AI MAKES A MOVE.

****When a pawn is promoted, it automatically selects the Queen piece

4. Development Plan and Timeline

Figure 4.1 : Projected Timeline

Features	**Week 1** (3/30/2020)	**Week 2** (4/6/2020)	**Week 3** (4/13/2020)	**Week 4** (4/20/2020)	**Week 5** (4/27/2020)	**Week 6** (5/4/2020)
Functions	X	X	X	START	X	X
Modules	X	X	X	START	X	X
Timer(?)	X	X	START	X	X	N/A
GUI	X	X	WIP	FINISH	X	X
AI	X	X	X	START	X	X
Withdraw Moves(?)	X	X	X	START	N/A	N/A
Readable Log	X	X	X	START	FINISH	X
Choose Board Side	X	X	FINISH	X	X	X

Team Member Responsibilities:

Anson Do: Chessboard GUI, Setup Menu, AI
Adrian Gomez: Updating Manuals, AI, Log Function
Kevin Huang: Creating and deleting pieces, Coding rules for pieces, Rules
Xianzhang Li: Main GUI(option1), add the Moving in the GUI,Rules
Arian Reyes: Creating Header Files, Functions for Events, Rules

Copyright

References

- https://freesvg.org/
    - *For all Images used for Pieces and Board on page 9.*

Index