# Project Chess Demonstration

Collaborated by Kunling Yang, Zichu Wu, and Min Suk Kim

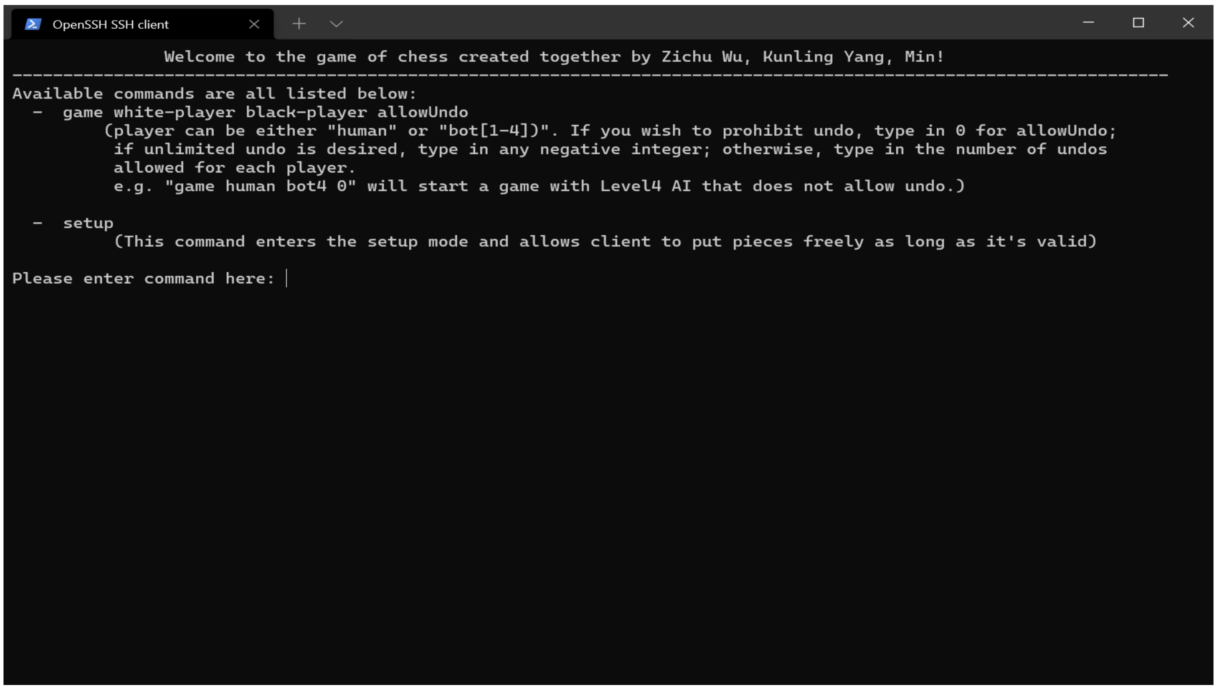
University of Waterloo, CS246, F21

Final Project

# Content

# Main Title Page

Below are the welcoming messages when the client opens the program. At the very first line is the name of all laborers. After that is the list of all acceptable commands (or modes), including the “game” command and “setup” command.



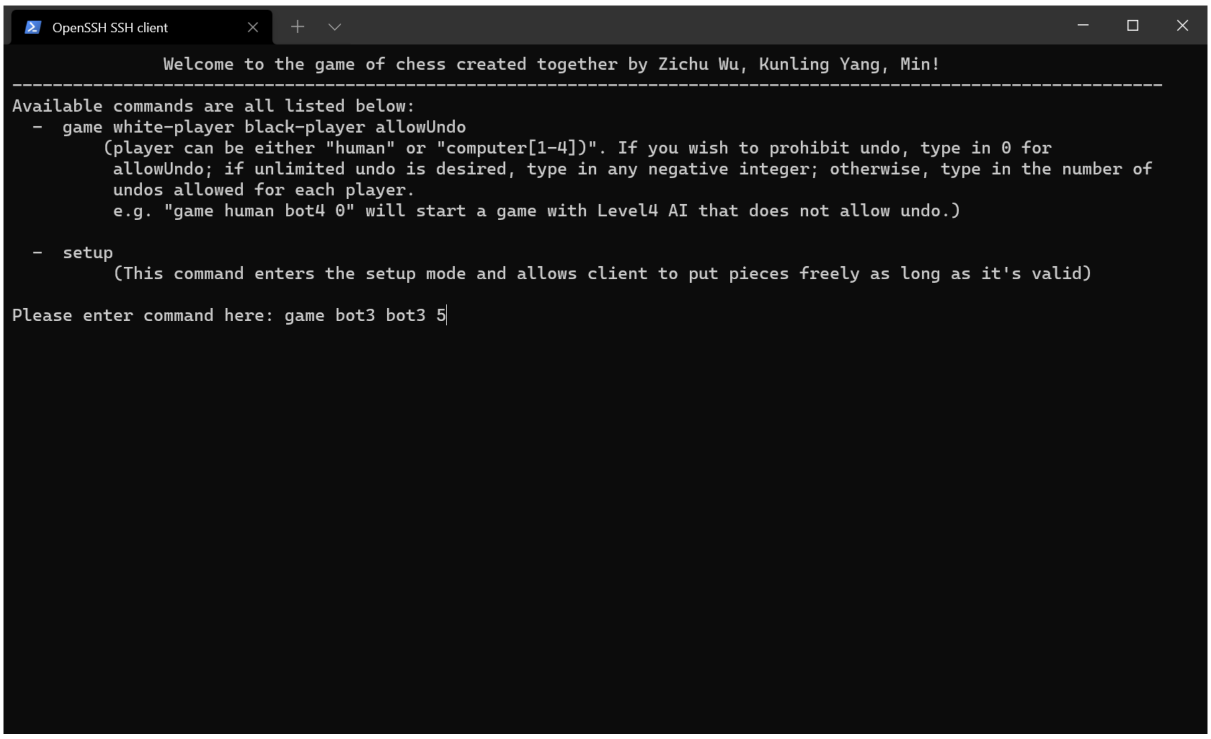
```
               Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
------------------------------------------------------------------------------------------------------------------
Available commands are all listed below:
  -  game white-player black-player allowUndo
         (player can be either "human" or "bot[1-4])". If you wish to prohibit undo, type in 0 for allowUndo;
          if unlimited undo is desired, type in any negative integer; otherwise, type in the number of undos
          allowed for each player.
          e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
          (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here: 
```

## Game Mode

Below is the walkthrough of the "*game*" command of this project. To enter the game mode (that is to play the game), the client is expected to type in "*game*" first, then two strings (one of "human", "computer1", "computer2", computer3", "computer4"), and an integer at last. Each component must be separated by at least one whitespace, and any leading whitespace and ending whitespace of this command is simply ignored.

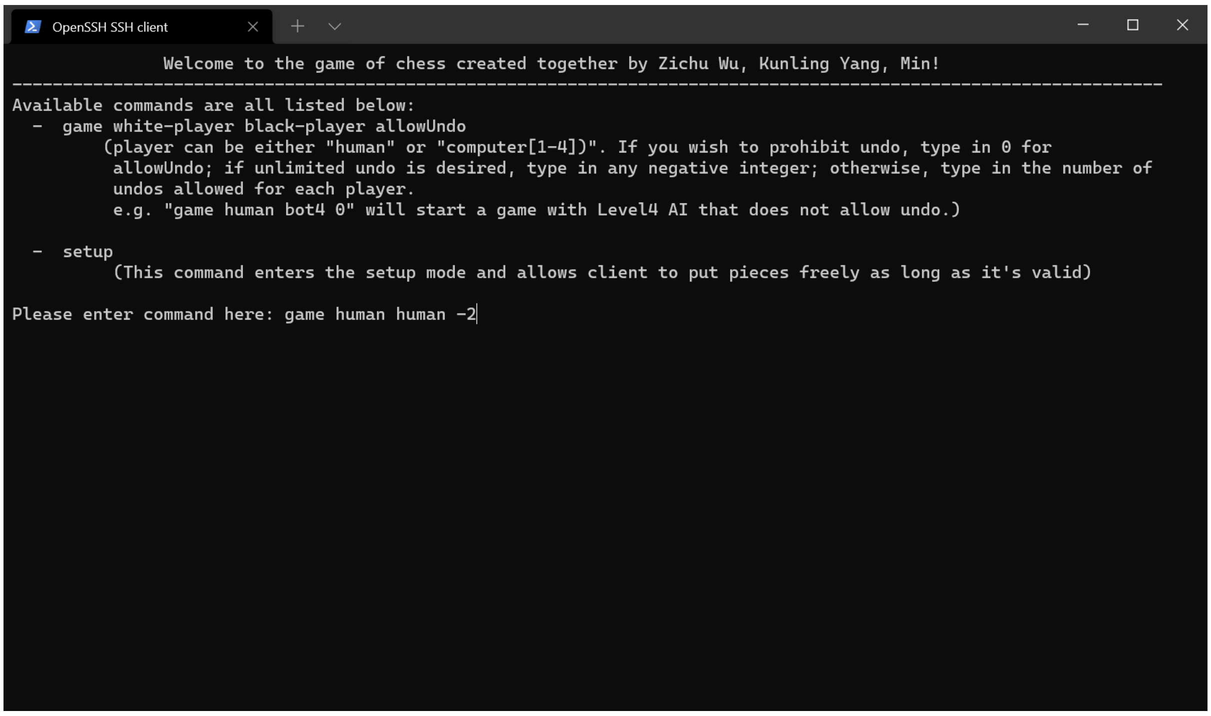
In the example below, the client typed in the command "*game bot3 bot3 5*", and this command initiated a game between Level 3 Computer player and another Level3 Computer players with at most 5 undos allowed for each player (despite the fact that AIs are unlikely to undo their moves).

Since the client never setup and directly entered the game mode, all the pieces were placed at the default positions just like a regular literally on-board chess game.



```
               Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
-------------------------------------------------------------------------------------------------------------------
Available commands are all listed below:
  -  game white-player black-player allowUndo
         (player can be either "human" or "computer[1-4]")". If you wish to prohibit undo, type in 0 for
          allowUndo; if unlimited undo is desired, type in any negative integer; otherwise, type in the number of
          undos allowed for each player.
          e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
          (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here: game bot3 bot3 5
```

By typing a negative integer followed by "game white-side player and black-side player", the program will give an unlimited number of undos for both players. This particular game will provide a human player vs human player chess game with an unlimited number of undos.



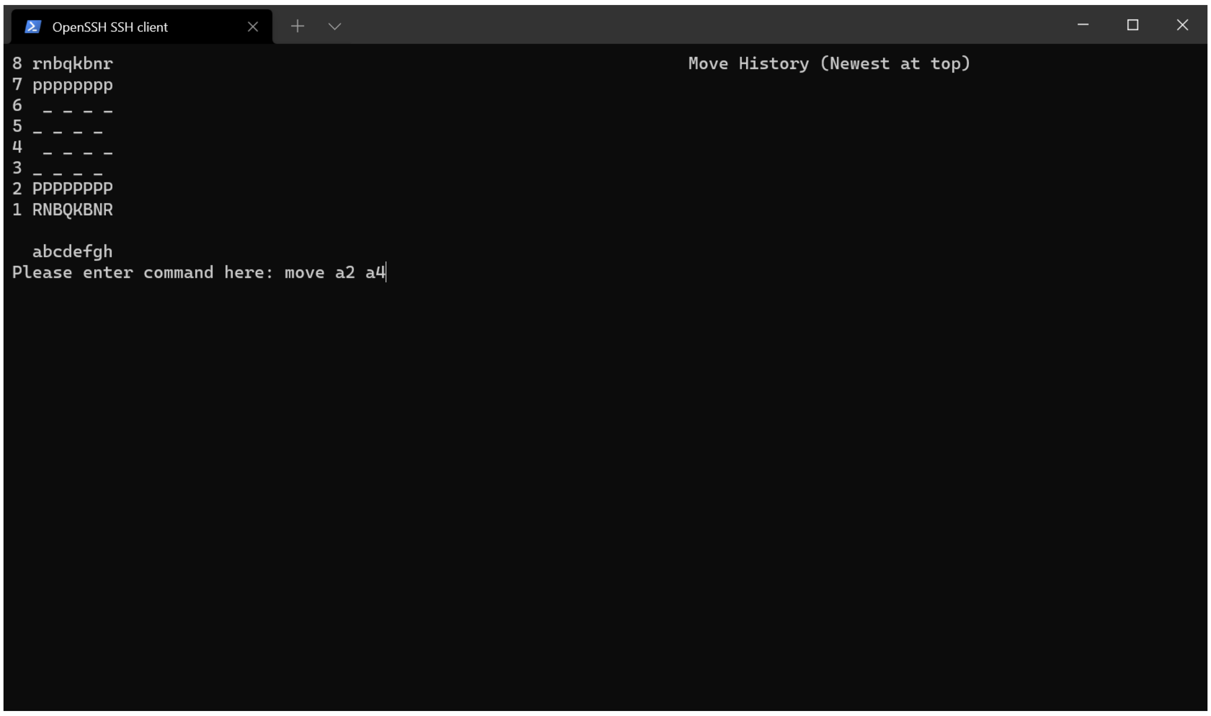
```
               Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
--------------------------------------------------------------------------------------------------------------------
Available commands are all listed below:
  -  game white-player black-player allowUndo
         (player can be either "human" or "computer[1-4])". If you wish to prohibit undo, type in 0 for
          allowUndo; if unlimited undo is desired, type in any negative integer; otherwise, type in the number of
          undos allowed for each player.
          e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
          (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here: game human human -2
```

However, if the client has ever entered the setup mode and saved his or her setup, that setup would take effect when a valid game command is provided, and the pieces would be placed at the location exactly as in the setup mode. Please refer to the Setup Mode for more related information.

## Move

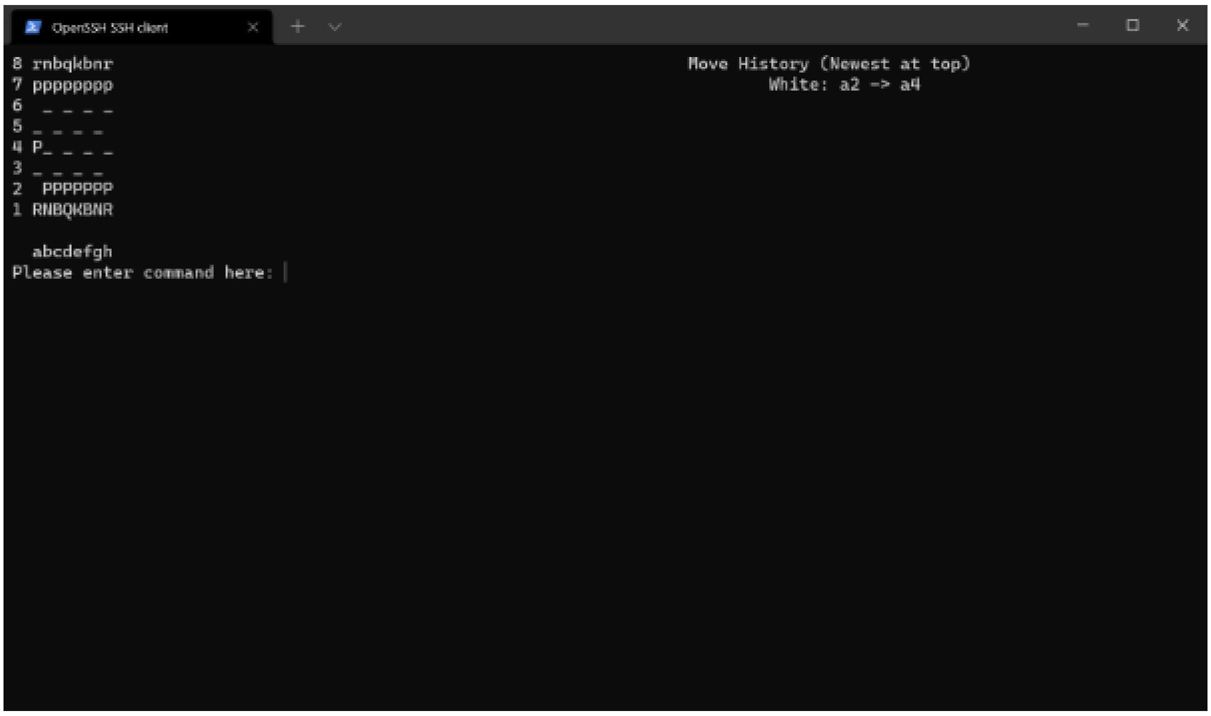
### Moving a Pawn

By typing “move a2 a4” the program will move pawn at a2 to a4. This is a legal move from white-side player. If players type in a legal move as a command, the program will automatically ask the next player to enter the command.



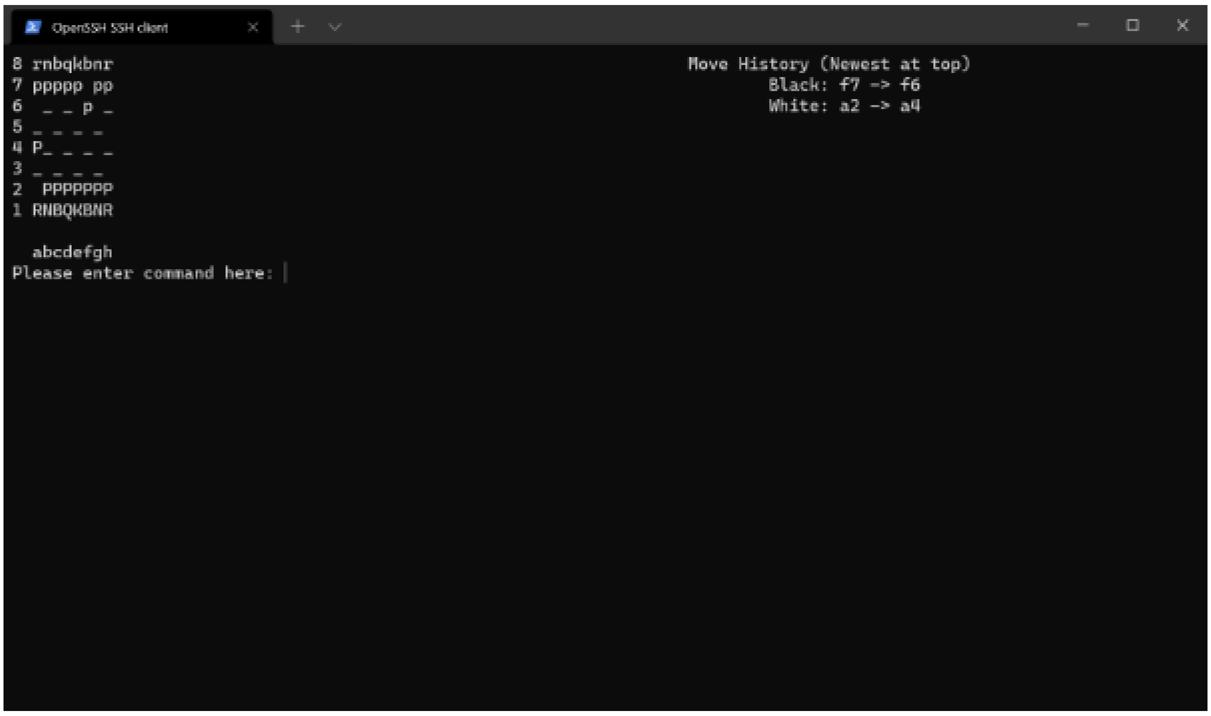
```
8 rnbqkbnr                                                          Move History (Newest at top)
7 pppppppp
6  _ _ _ _
5 _ _ _ _ 
4  _ _ _ _
3 _ _ _ _ 
2 PPPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here: move a2 a4
```

Pawn at a2 has been successfully moved to a4 and the program asks for black-side player to enter the command. Now black-side player should make any legal moves.



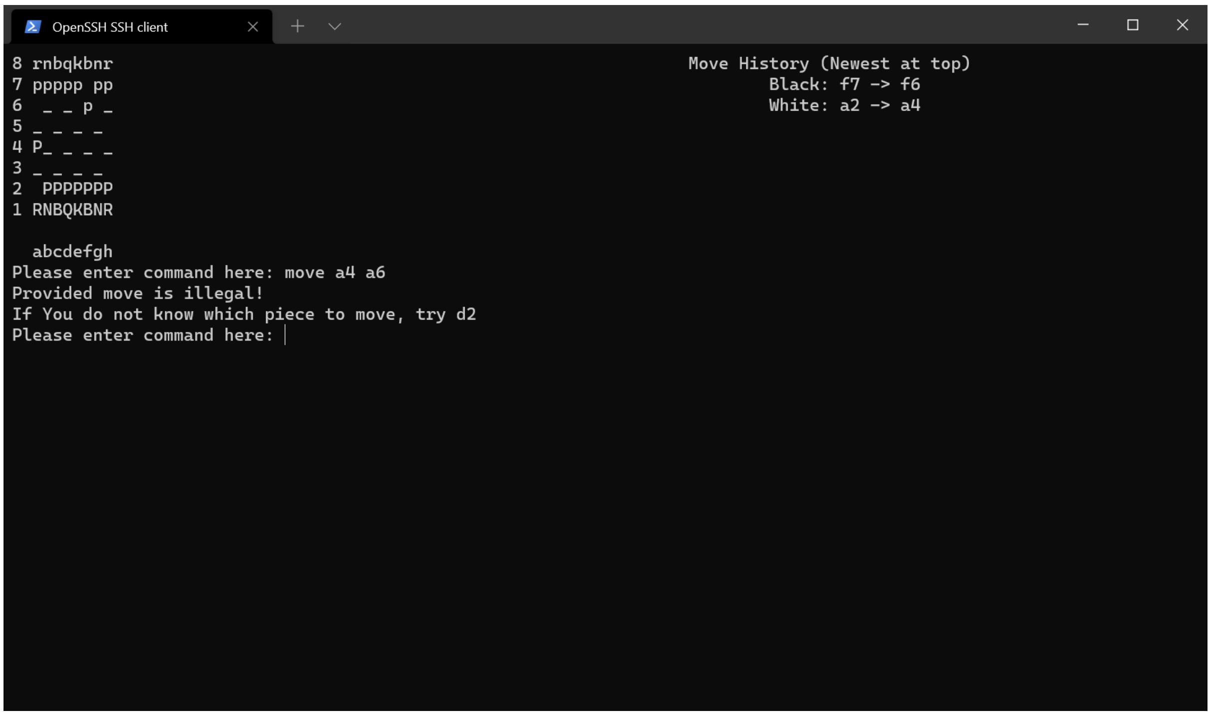
```
8 rnbqkbnr                                                Move History (Newest at top)
7 pppppppp                                                        White: a2 -> a4
6  _ _ _ _
5 _ _ _ _
4 P_ _ _ _
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here:
```

By typing “move f7 f6”, the program moves black-side player’s pawn at f7 to f6. This is also a legal move made by the player, so the program asks for white-side player to enter the next move. As it can be seen from the diagram, all the players’ move history are being shown on the right top of the diagram.



```
8 rnbqkbnr                                                Move History (Newest at top)
7 ppppp pp                                                        Black: f7 -> f6
6  _ _ p _                                                        White: a2 -> a4
5 _ _ _ _
4 P_ _ _ _
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here:
```

Here’s the case of an illegal move made by white-side player. White-side player is trying to move an already-moved pawn two positions forward and the program suggests a player which available piece to move. In the case of an illegal move by players, move history will not be updated and the program will ask the same player to enter another available command.

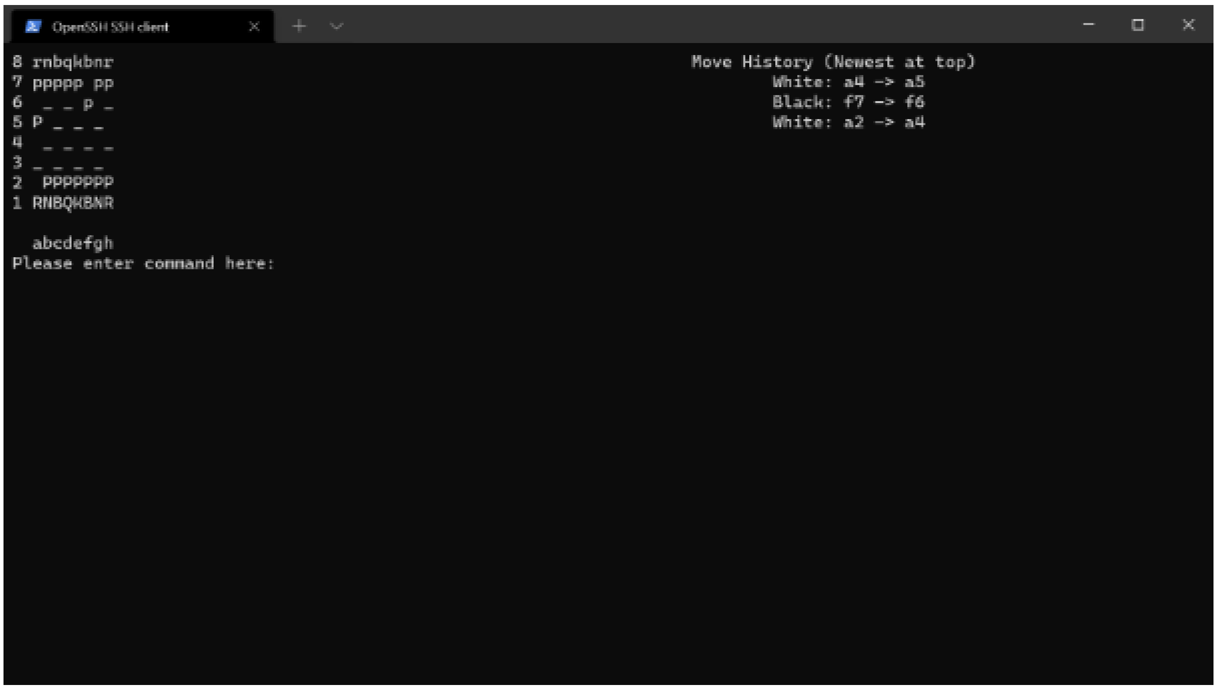


```
8 rnbqkbnr                                                        Move History (Newest at top)
7 ppppp pp                                                                Black: f7 -> f6
6  _ _ p _                                                                White: a2 -> a4
5 _ _ _ _
4 P_ _ _ _
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here: move a4 a6
Provided move is illegal!
If You do not know which piece to move, try d2
Please enter command here: 
```

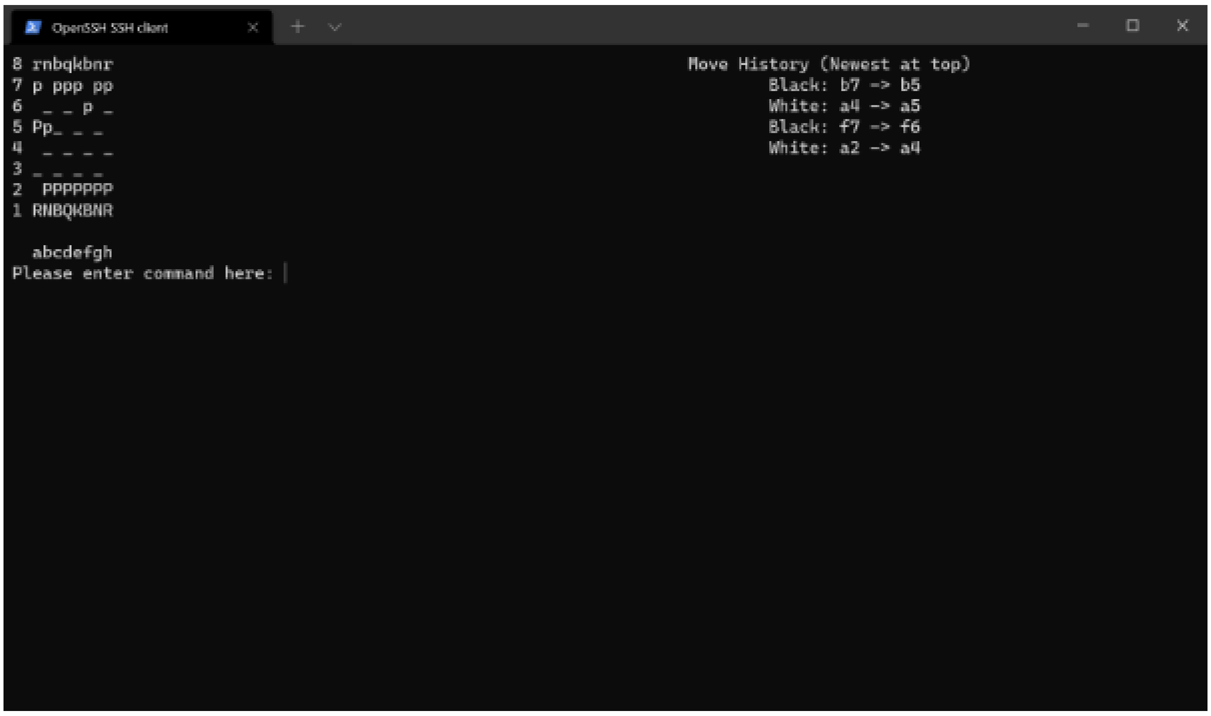
## Pawn Capturing

## En Passant

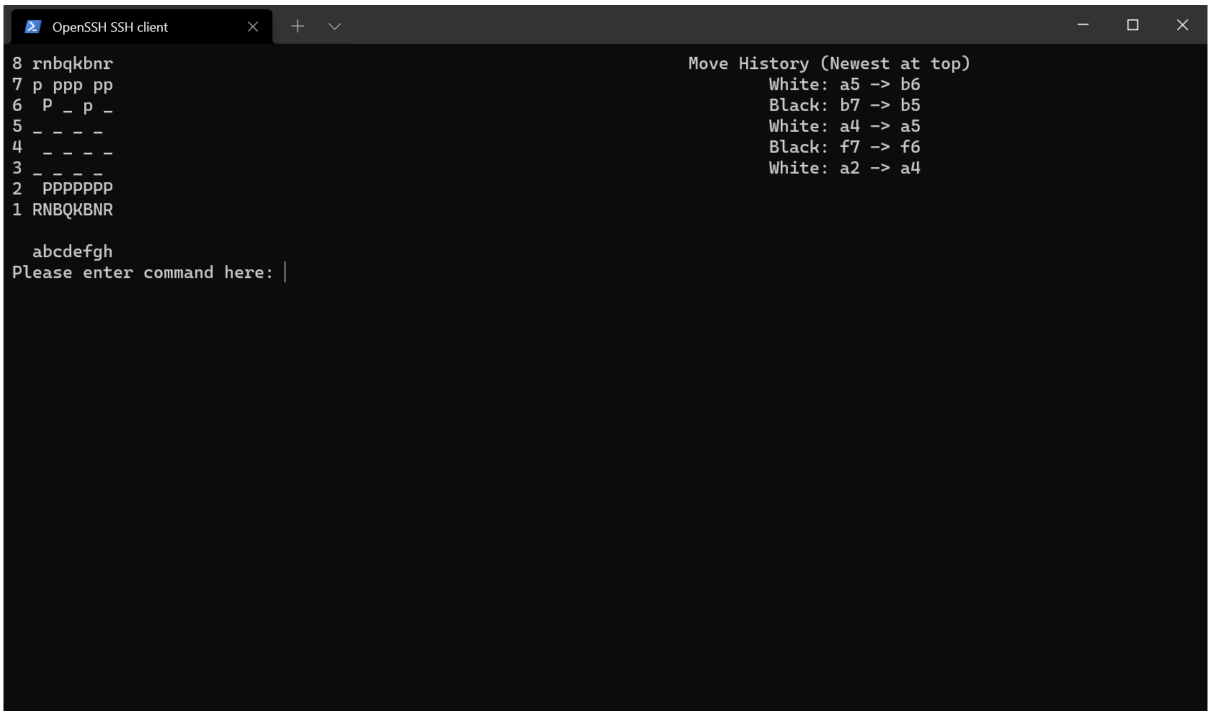
En Passant is a special pawn capture that can only occur immediately after an opponent's pawn makes a move of two squares from its starting square, and the later could have been captured by an friendly pawn had it advanced only one square. The client could capture the just-moved pawn "in passing" through the first square, as if the opponent's pawn had advanced only one square and the friendly pawn had captured it normally.



```
8 rnbqkbnr                                                  Move History (Newest at top)
7 ppppp pp                                                          White: a4 -> a5
6  _ _ p _                                                          Black: f7 -> f6
5 P _ _ _                                                           White: a2 -> a4
4  _ _ _ _
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter connand here:
```



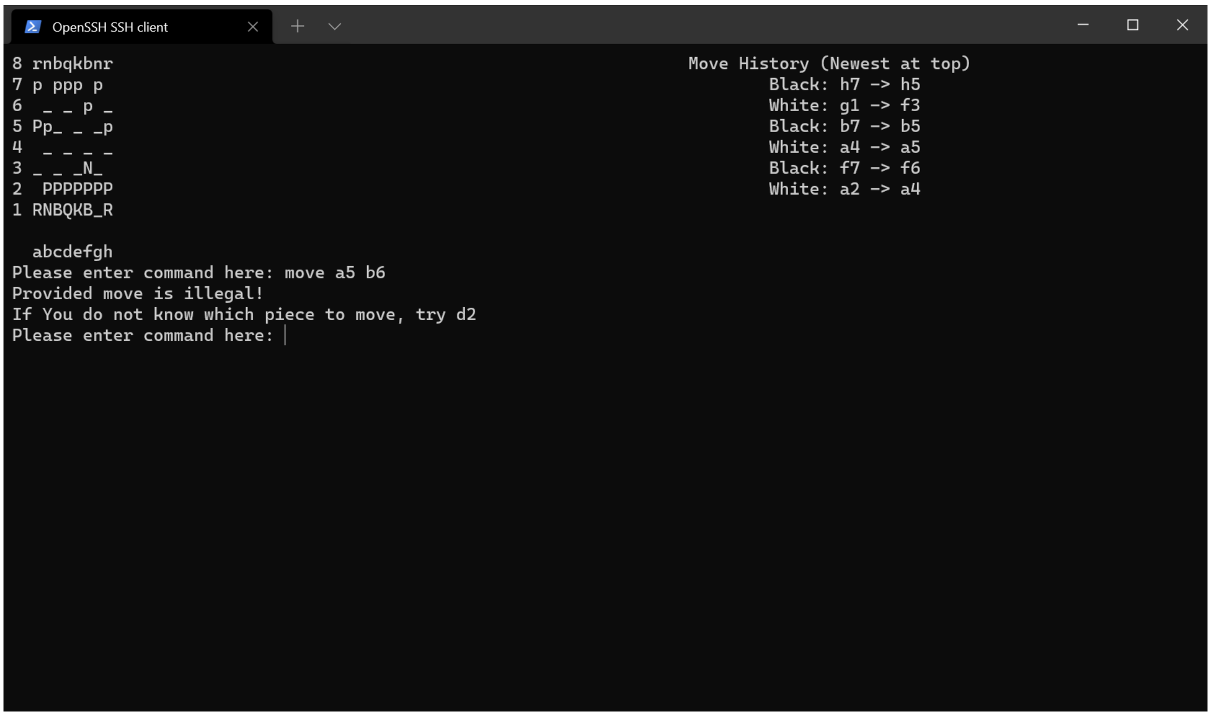
```
8 rnbqkbnr                                                  Move History (Newest at top)
7 p ppp pp                                                          Black: b7 -> b5
6  _ _ p _                                                          White: a4 -> a5
5 Pp_ _ _                                                           Black: f7 -> f6
4  _ _ _ _                                                          White: a2 -> a4
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here:
```



```
8 rnbqkbnr                                                         Move History (Newest at top)
7 p ppp pp                                                                 White: a5 -> b6
6  P _ p _                                                                 Black: b7 -> b5
5 _ _ _ _                                                                  White: a4 -> a5
4  _ _ _ _                                                                 Black: f7 -> f6
3 _ _ _ _                                                                  White: a2 -> a4
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here: 
```

For each enemy piece, there is only one chance to en passant it. In the example below, after the black side moved the pawn from b7 to b5, the white side did not choose to en passant it immediately. Instead, he or she chose to move a knight from g1 to f3.. According to the rules, this en passant chance was, therefore, wasted. Black side then chose to move a pawn from h7 to h5.

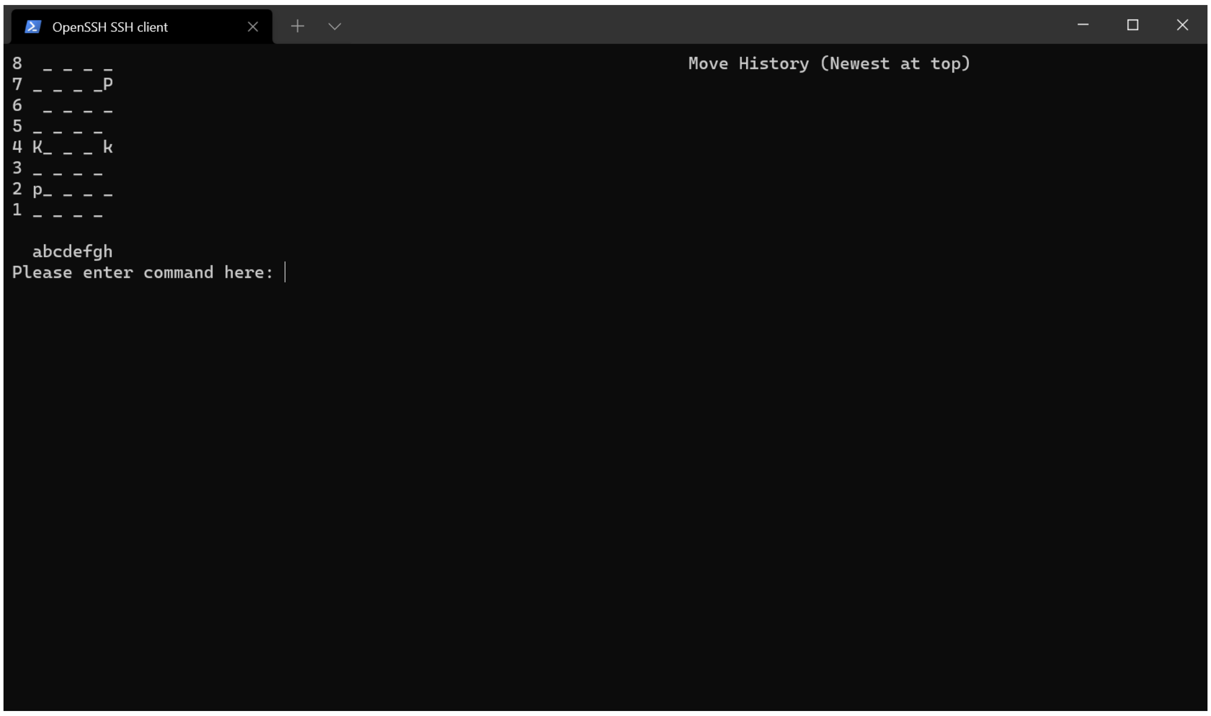
When it went back to the white side to make a move, the program simply rejected the command “*move a6 b6*” which could have captured the black pawn if that chance was not wasted.



```
8 rnbqkbnr                                                         Move History (Newest at top)
7 p ppp p                                                                  Black: h7 -> h5
6  _ _ p _                                                                 White: g1 -> f3
5 Pp_ _ _p                                                                 Black: b7 -> b5
4  _ _ _ _                                                                 White: a4 -> a5
3 _ _ _N_                                                                  Black: f7 -> f6
2  PPPPPPP                                                                 White: a2 -> a4
1 RNBQKB_R

  abcdefgh
Please enter command here: move a5 b6
Provided move is illegal!
If You do not know which piece to move, try d2
Please enter command here: 
```

## Promotion

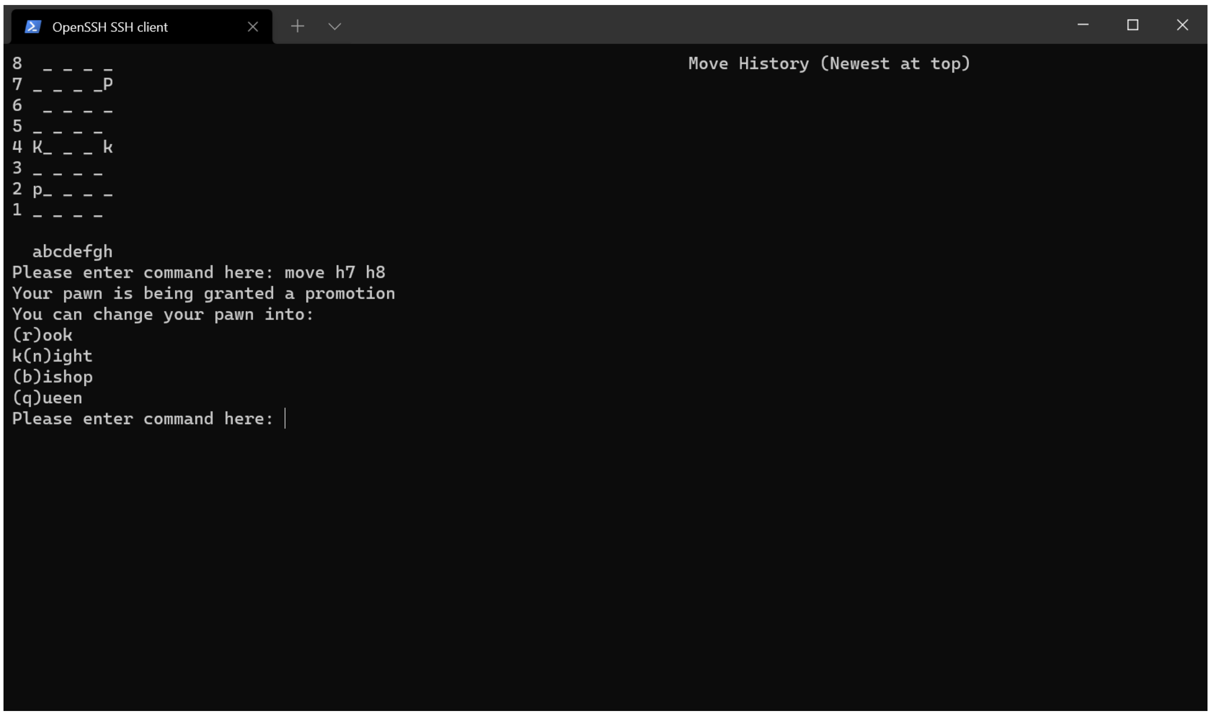
This is an example of a promotion of pawn at h7 in game mode. Pawn at h7 will be moved to h8 by typing the command “move h7 h8”. We’ve used the setup mode to show the illustration of the promotion of the pawn. That is the reason why the mover history of past moves aren’t available in the diagram below.



OpenSSH SSH client

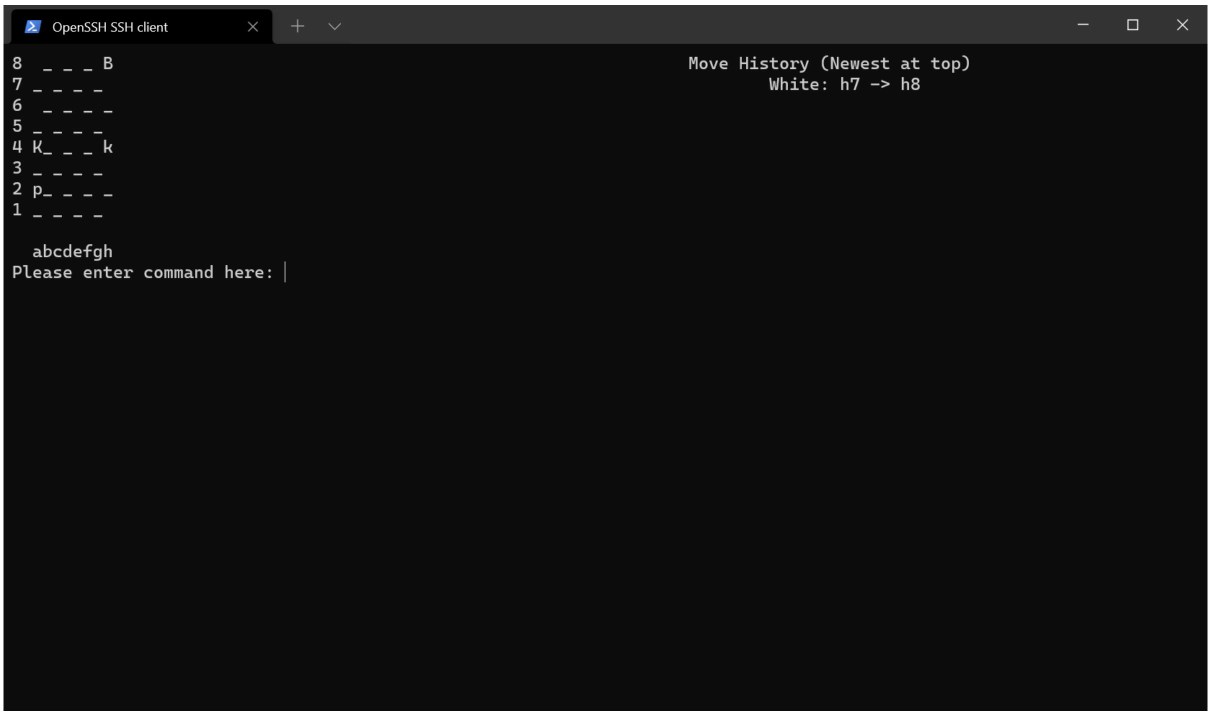
```
8  _ _ _ _                                                  Move History (Newest at top)
7 _ _ _ _P
6  _ _ _ _
5 _ _ _ _
4 K_ _ _ k
3 _ _ _ _
2 p_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here:
```

After, the program will tell the user about the information of a promotion. By typing the letter “b” (case sensitive) , the pawn at position h8 will be promoted to a bishop. Instead of showing “P” for the pawn at position h8, bishop “B” will be outputted at position h8. Then it’s black-side player’s turn to make the next move.



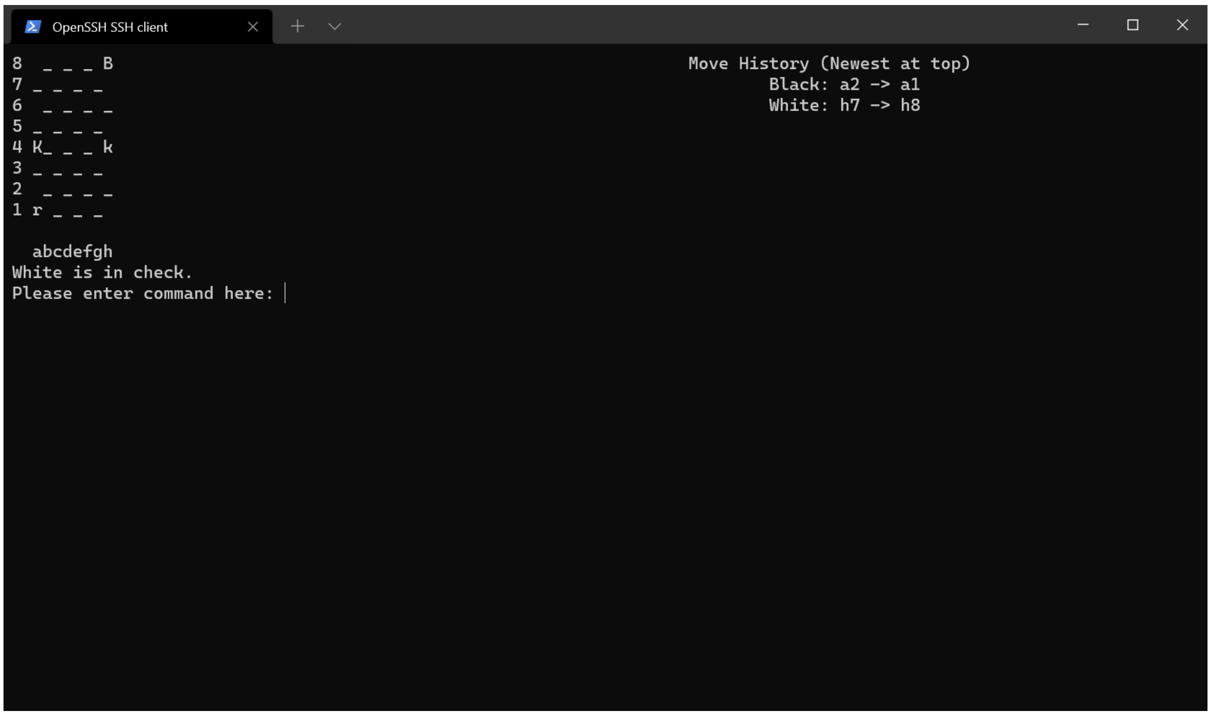
```
8  _ _ _ _                                                         Move History (Newest at top)
7 _ _ _ _P
6  _ _ _ _
5 _ _ _ _
4 K_ _ _ k
3 _ _ _ _
2 p_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: move h7 h8
Your pawn is being granted a promotion
You can change your pawn into:
(r)ook
k(n)ight
(b)ishop
(q)ueen
Please enter command here: 
```

As it was mentioned above, all players must be aware of the letter case, since the program will be case sensitive for the particular promotion commands.
Now the program is showing the move history after the pawn moved from the position h7 to h8.



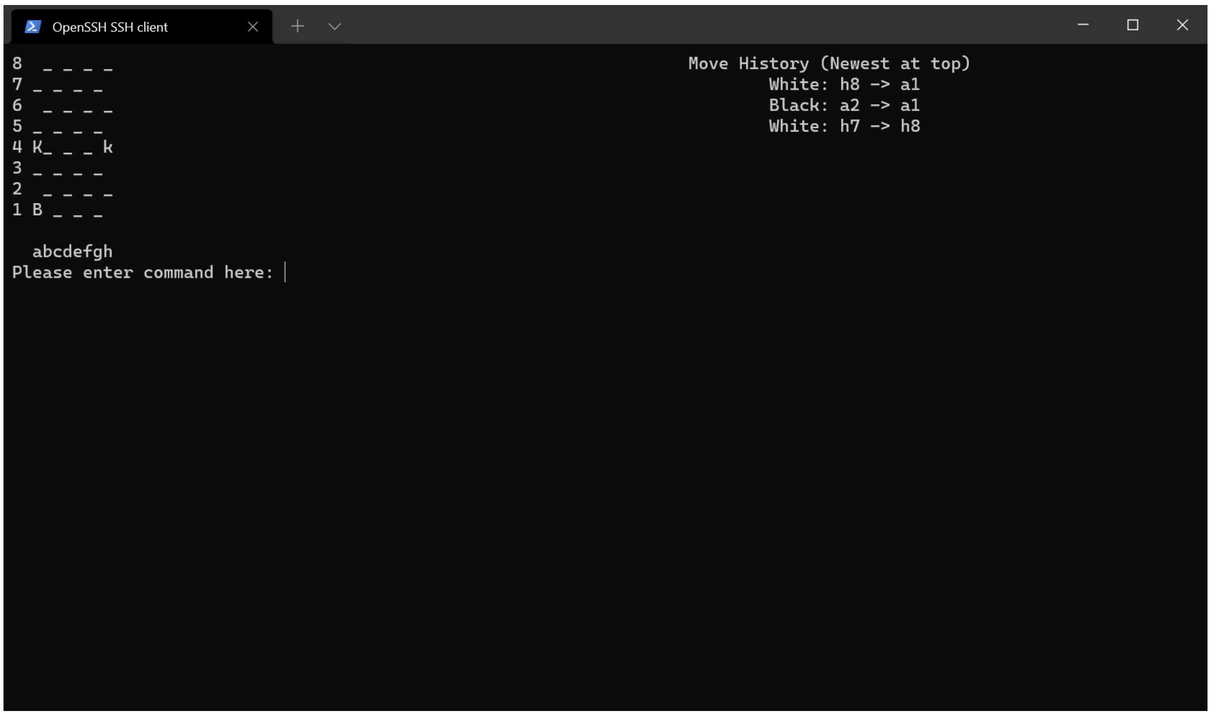
```
8  _ _ _ B                                                         Move History (Newest at top)
7 _ _ _ _                                                                  White: h7 -> h8
6  _ _ _ _
5 _ _ _ _
4 K_ _ _ k
3 _ _ _ _
2 p_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: 
```

For black-side player command, by typing “move a2 a1” and promotion of pawn to a rook will make the white-side player’s king in check. Similar to white-side player’s pawn promotion, the diagram is showing “r” instead of “p” since the promotion has already happened.



```
8  _ _ _ B                                                         Move History (Newest at top)
7 _ _ _ _                                                                  Black: a2 -> a1
6  _ _ _ _                                                                 White: h7 -> h8
5 _ _ _ _
4 K_ _ _ k
3 _ _ _ _
2  _ _ _ _
1 r _ _ _

  abcdefgh
White is in check.
Please enter command here: 
```

By typing the command “move h8 a1” to capture rook and save king from being checked.



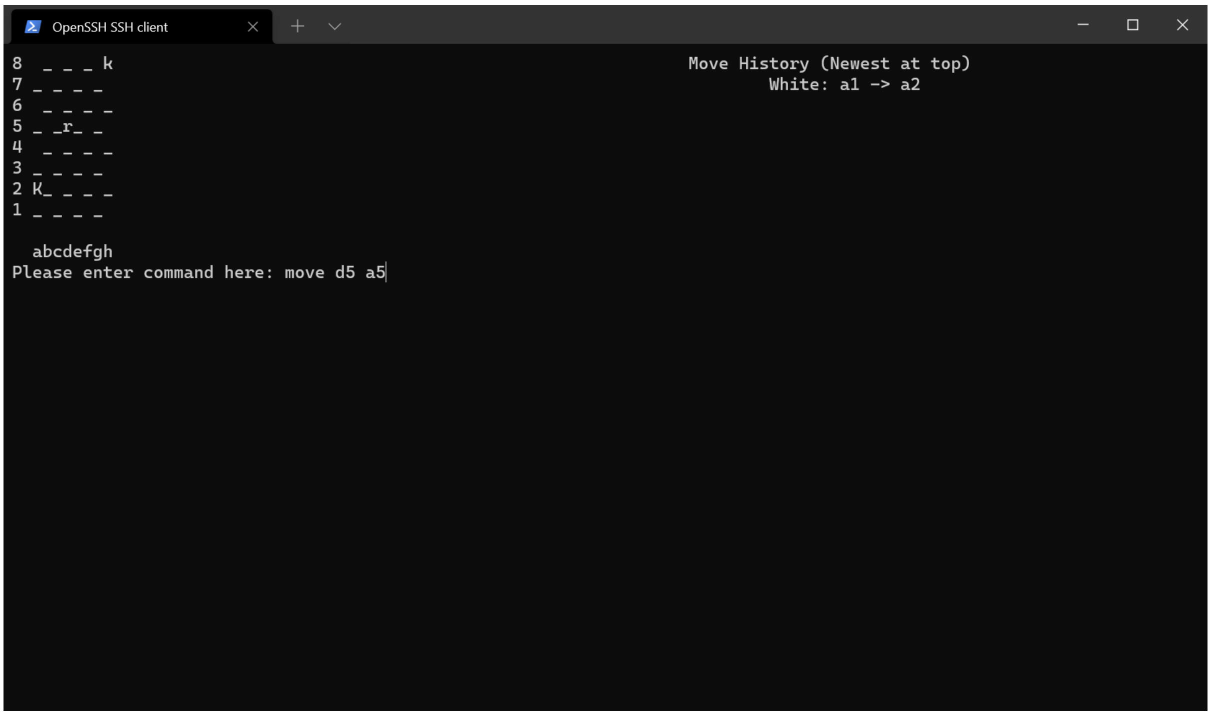
```
8  _ _ _ _                                                        Move History (Newest at top)
7 _ _ _ _                                                                 White: h8 -> a1
6  _ _ _ _                                                                Black: a2 -> a1
5 _ _ _ _                                                                 White: h7 -> h8
4 K_ _ _ k
3 _ _ _ _
2  _ _ _ _
1 B _ _ _

  abcdefgh
Please enter command here:
```

Now we have shown the complete promotion of pawns for both white and black-side players. Next we will show how different pieces move by their own characteristics.

## Moving a Rook

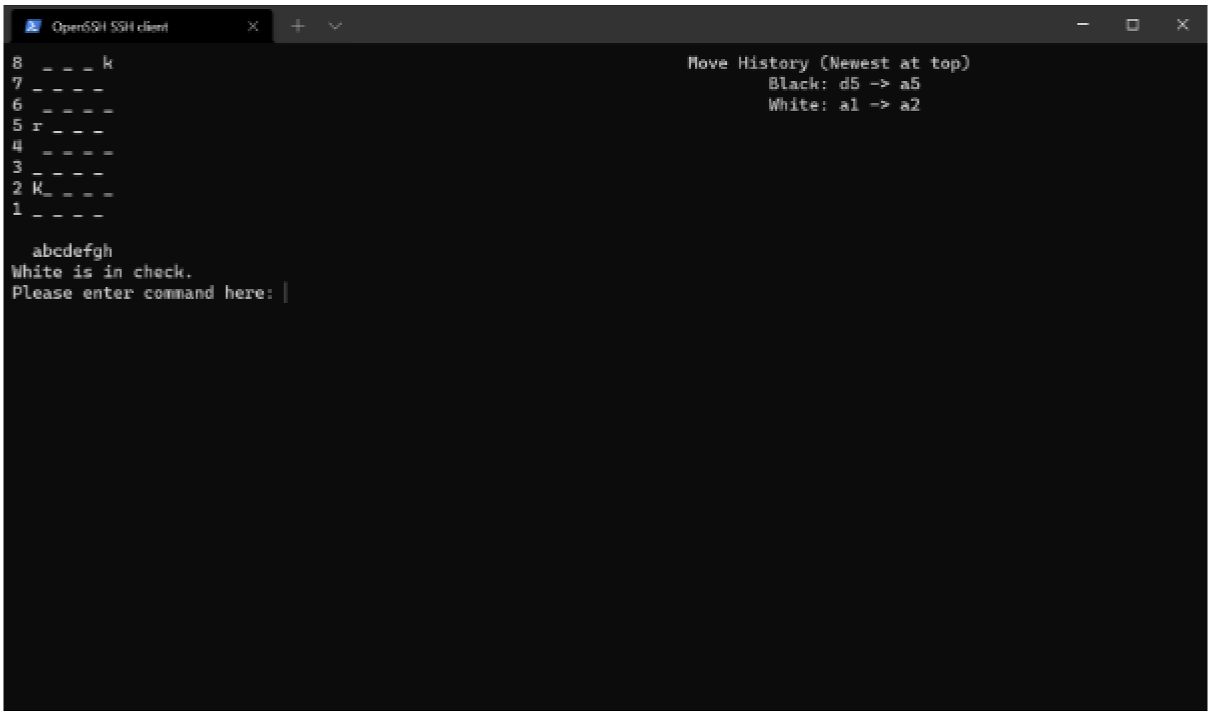
In a normal chess game, rooks can move in any number of positions that players desire, either horizontally or vertically. Unlike knights, it cannot jump over other pieces and must be stopped whenever the rooks encounter other pieces.

We have created another setup mode to illustrate the movement of a rook. In order to run the game properly, we had to place a king for each side to show the characteristics of a rook for our chess game.



```
8  _ _ _ k                                                  Move History (Newest at top)
7 _ _ _ _                                                          White: a1 -> a2
6  _ _ _ _
5 _ _r_ _
4  _ _ _ _
3 _ _ _ _
2 K_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: move d5 a5
```

By typing the command “move d5 a5” and since this is a legal move that can be made by the player, the rook at position d5 will be moved to the position a5. Of course the program updates the move history to show the movement of the rook from d5 to a5. Now the king at position a2 from white-side player is in check by the black-side player’s rook at position a5.



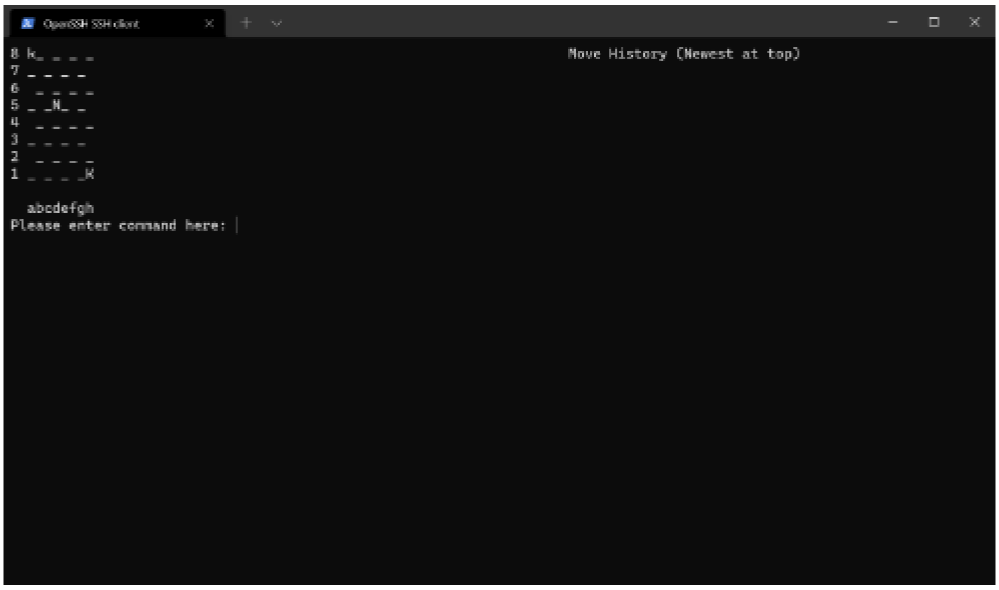
```
8  _ _ _ k                                                    Move History (Newest at top)
7 _ _ _ _                                                             Black: d5 -> a5
6  _ _ _ _                                                            White: a1 -> a2
5 r _ _ _
4  _ _ _ _
3 _ _ _ _
2 K_ _ _ _
1 _ _ _ _

  abcdefgh
White is in check.
Please enter command here:
```

Note that a rook has a special move called “castling”, it requires neither the rook and the king to be never moved, and it moves these two pieces in the same turn. To avoid ambiguity (whether the player wants to mobilize the rook and the king at the same time or just the rook itself), this special move is put in the section under Moving a King, please refer to that part.

## Moving a Knight

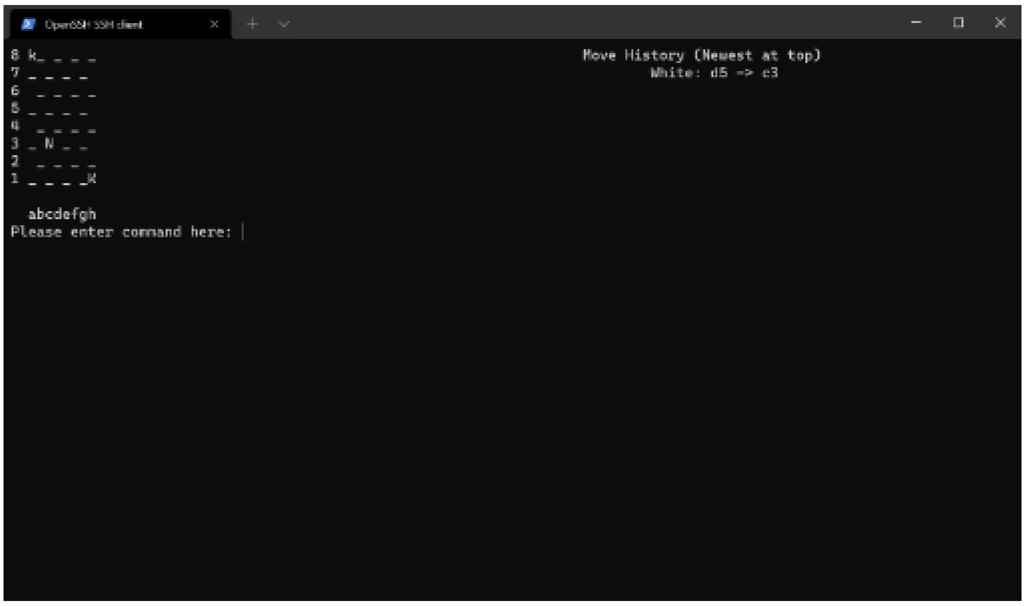
In the chess game,  the knight's movement is unique: it may move two squares vertically and one square horizontally, or two squares horizontally and one square vertically (with both forming the shape of an L). This way, a knight can have a maximum of 8 moves. While moving, the knight can jump over pieces to reach its destination. And clearly, knights can not go outside the board just like the other pieces.

In the following example, it is now the white's turn, and the white knight is at d5. Eight different positions are reachable from there, including: e7, f6, f4, e3, c3, b4, b6, c7 (in clockwise sequence).



```
8 k_ _ _ _                                                        Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _N_ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _K

  abcdefgh
Please enter command here: 
```

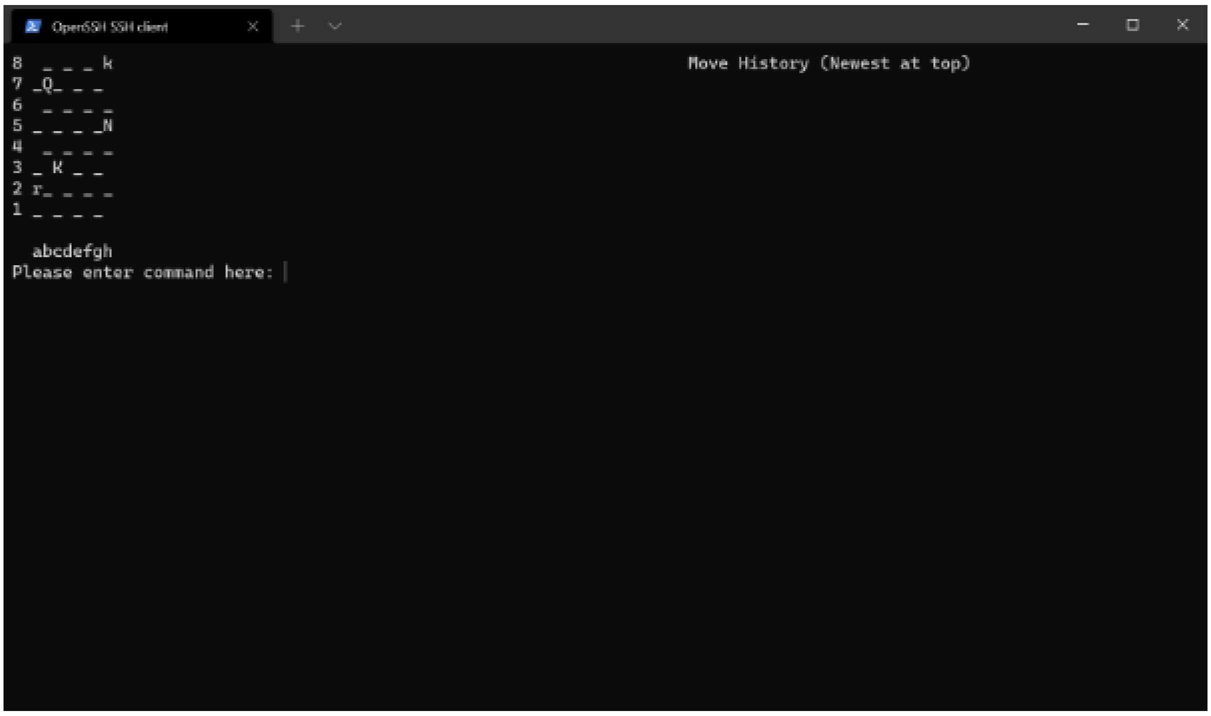
For example, the white player typed in the command "*move d5 c3*", and pressed the [enter] key. Then the real-time board will be like this:



```
8 k_ _ _ _                                                        Move History (Newest at top)
7 _ _ _ _                                                                White: d5 -> c3
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ N _ _
2  _ _ _ _
1 _ _ _ _K

  abcdefgh
Please enter command here: 
```

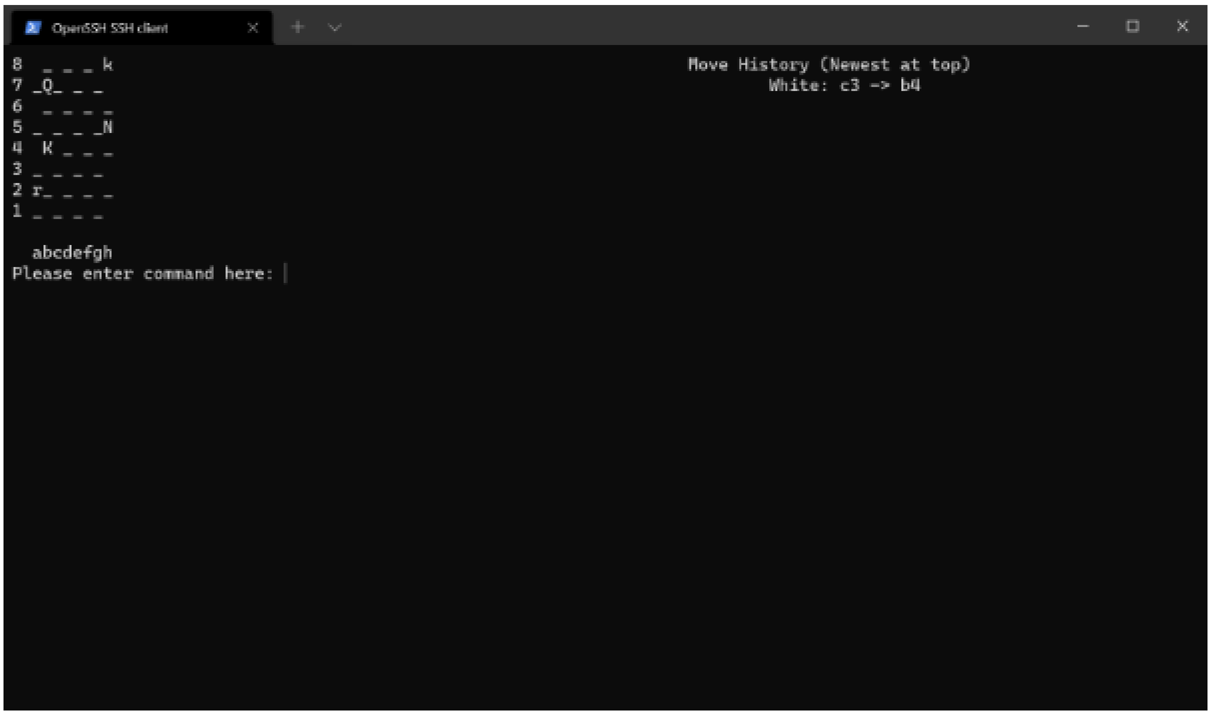
## Moving a King

In a chess game, the king can move one square in any direction (horizontally, vertically, or diagonally), unless the square is already occupied by a friendly piece, or the move would place the king in check. Besides, the king is also involved in the special move called castling, which is also implemented in this project.



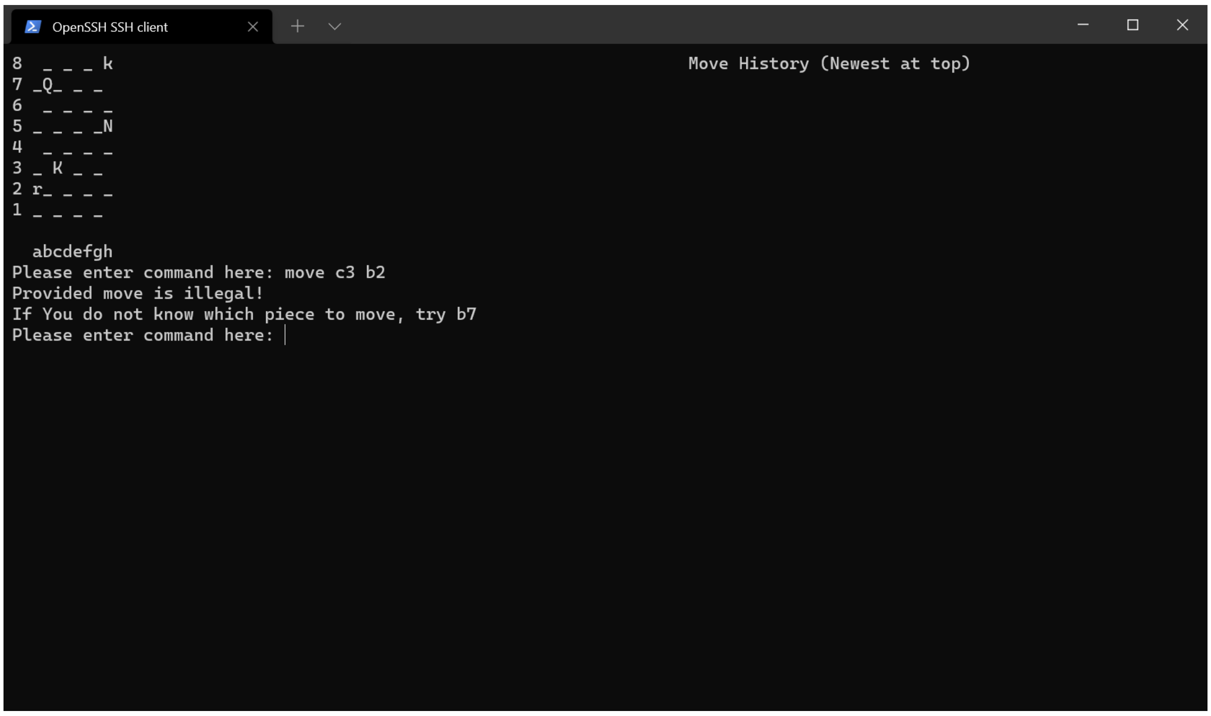
```
8  _ _ _ k                                                      Move History (Newest at top)
7 _Q_ _ _
6  _ _ _ _
5 _ _ _ _N
4  _ _ _ _
3 _ K _ _
2 r_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here:
```

The picture above demonstrated a setup and now it was the white's turn.  By typing "*move c3 b4*" and pressing [enter], the white king was moved, just as the picture below shows.



```
8  _ _ _ k                                                      Move History (Newest at top)
7 _Q_ _ _                                                              White: c3 -> b4
6  _ _ _ _
5 _ _ _ _N
4  K _ _ _
3 _ _ _ _
2 r_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here:
```

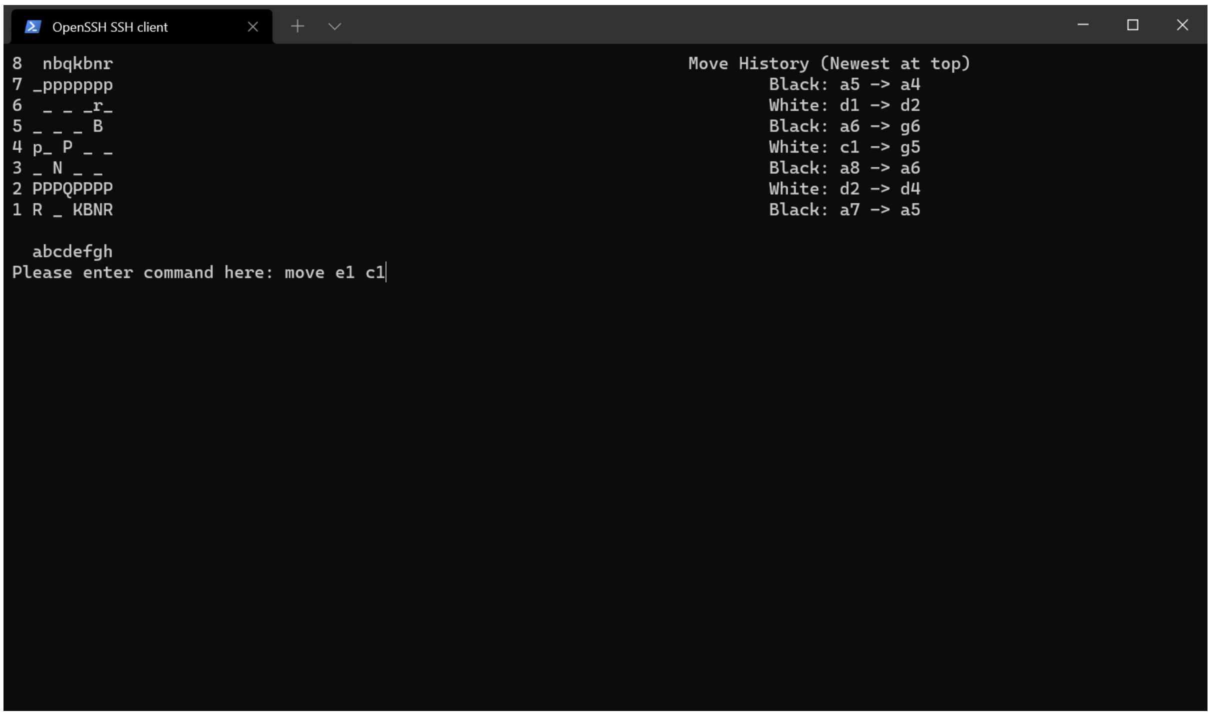
But however, the command “*move c3 b2*” was rejected by the program since this move put the white king in check, disallowed by the rules. For more detailed information about check, please refer to the check & checkmate part.



```
8  _ _ _ k                                                         Move History (Newest at top)
7 _Q_ _ _
6  _ _ _ _
5 _ _ _ _N
4  _ _ _ _
3 _ K _ _
2 r_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: move c3 b2
Provided move is illegal!
If You do not know which piece to move, try b7
Please enter command here:
```

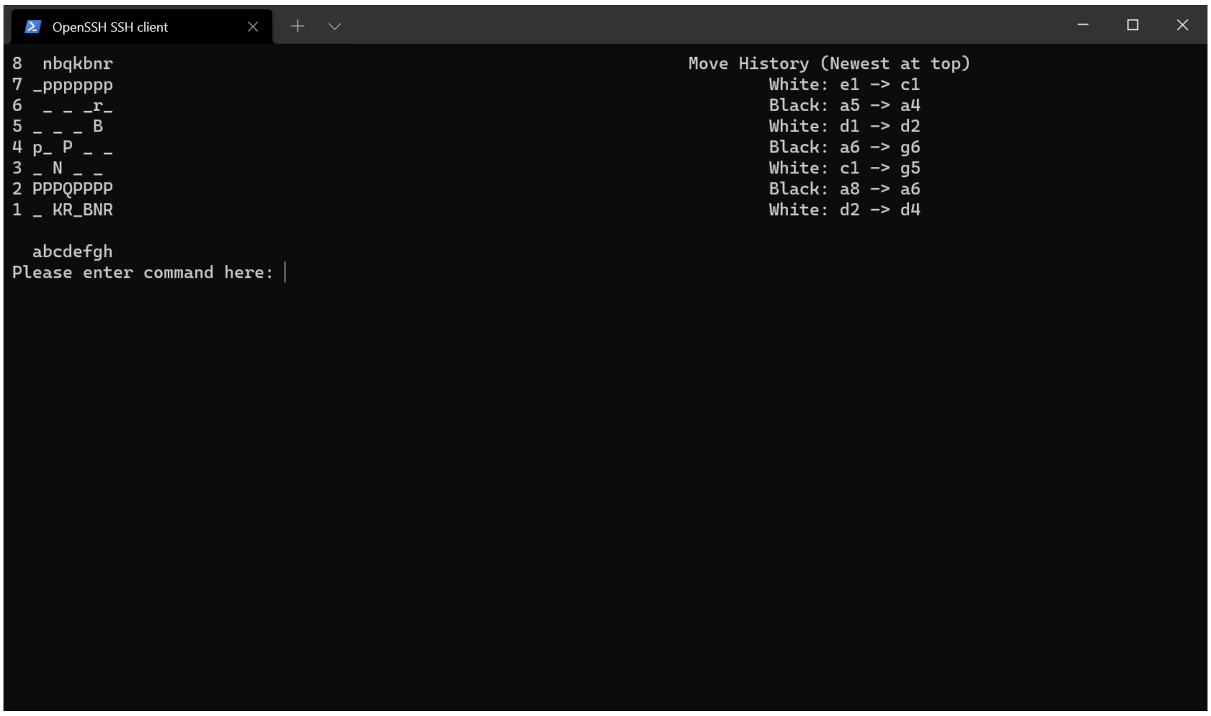
## Castling

Castling is a special move in the game of chess involving a player's king and either of the player's original rooks. It is the only move in chess in which a player moves two pieces in the same move,



```
8  nbqkbnr                                                          Move History (Newest at top)
7 _ppppppp                                                                  Black: a5 -> a4
6  _ _ _r_                                                                  White: d1 -> d2
5 _ _ _ B                                                                   Black: a6 -> g6
4 p_ P _ _                                                                  White: c1 -> g5
3 _ N _ _                                                                   Black: a8 -> a6
2 PPPQPPPP                                                                  White: d2 -> d4
1 R _ KBNR                                                                  Black: a7 -> a5

  abcdefgh
Please enter command here: move e1 c1
```

Consider the situation above, and currently it is the white player’s turn. At this time, neither the white rook at a1 nor the white king at e1 has been moved so far, thus a castling can be carried. To avoid ambiguity, a castling must be initiated by the king rather than a rook. The client can type in “*move e1 c1*” to start a castling.

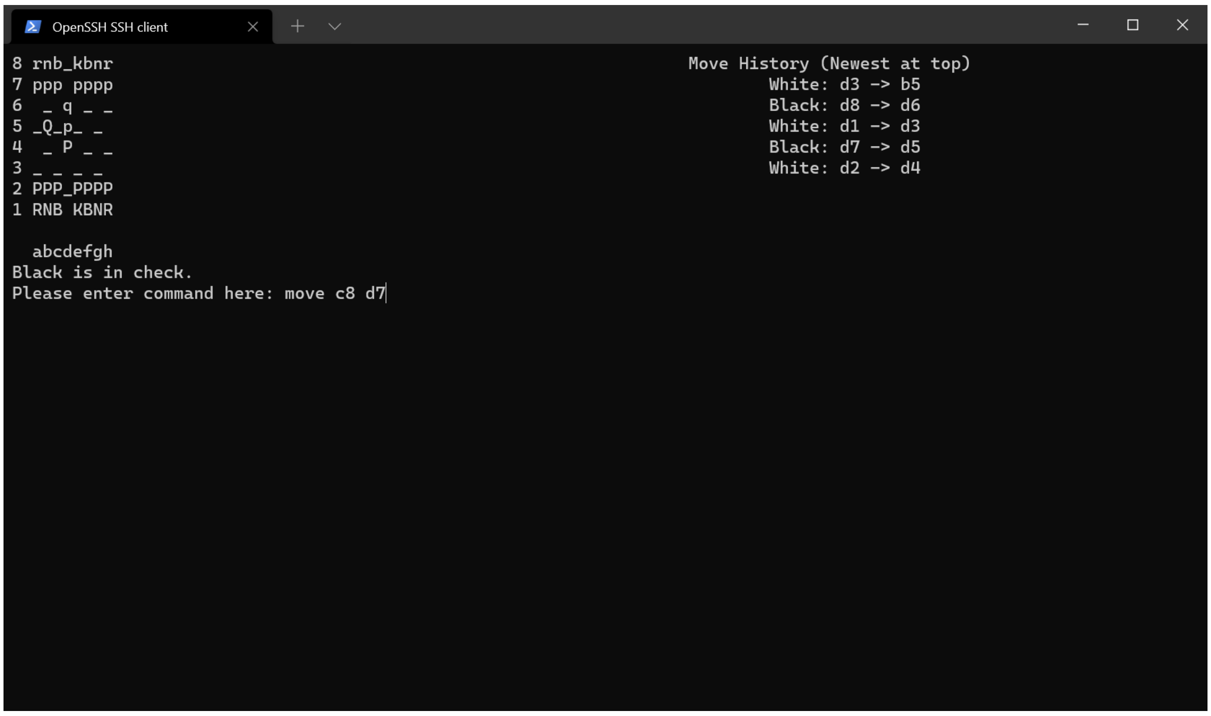


```
8  nbqkbnr                                                         Move History (Newest at top)
7 _ppppppp                                                                 White: e1 -> c1
6  _ _ _r_                                                                 Black: a5 -> a4
5 _ _ _ B                                                                  White: d1 -> d2
4 p_ P _ _                                                                 Black: a6 -> g6
3 _ N _ _                                                                  White: c1 -> g5
2 PPPQPPPP                                                                 Black: a8 -> a6
1 _ KR_BNR                                                                 White: d2 -> d4

  abcdefgh
Please enter command here: 
```

After putting this command, the white king moved from e1 to c1 and the white rook moved from a1 to d1. This whole castling was counted as a single move, thus only one piece of move information is added to the move history, which is reflected by the move history column on the right side.

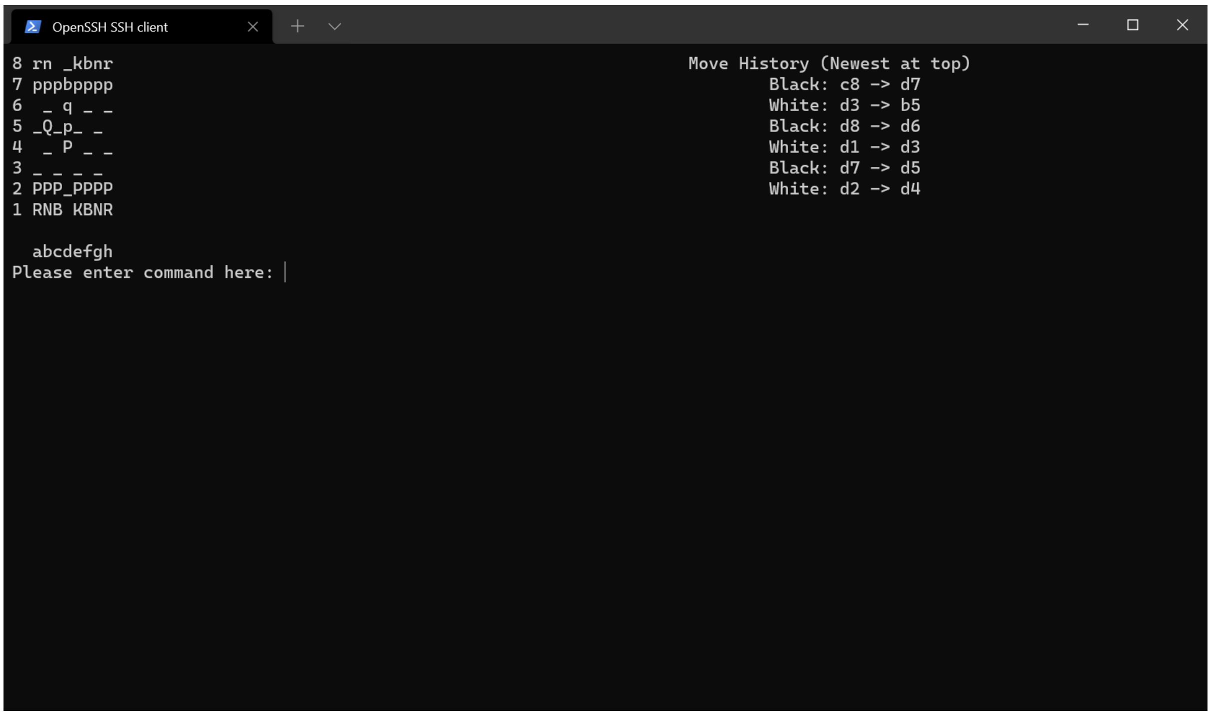
## Moving a Bishop

In a chess game, bishops have similar properties as rooks but they can only move in diagonal directions. Of course, they cannot jump over other pieces unlike knights. Move history from the diagram below We will be demonstrating the movements of bishops with the diagrams below.



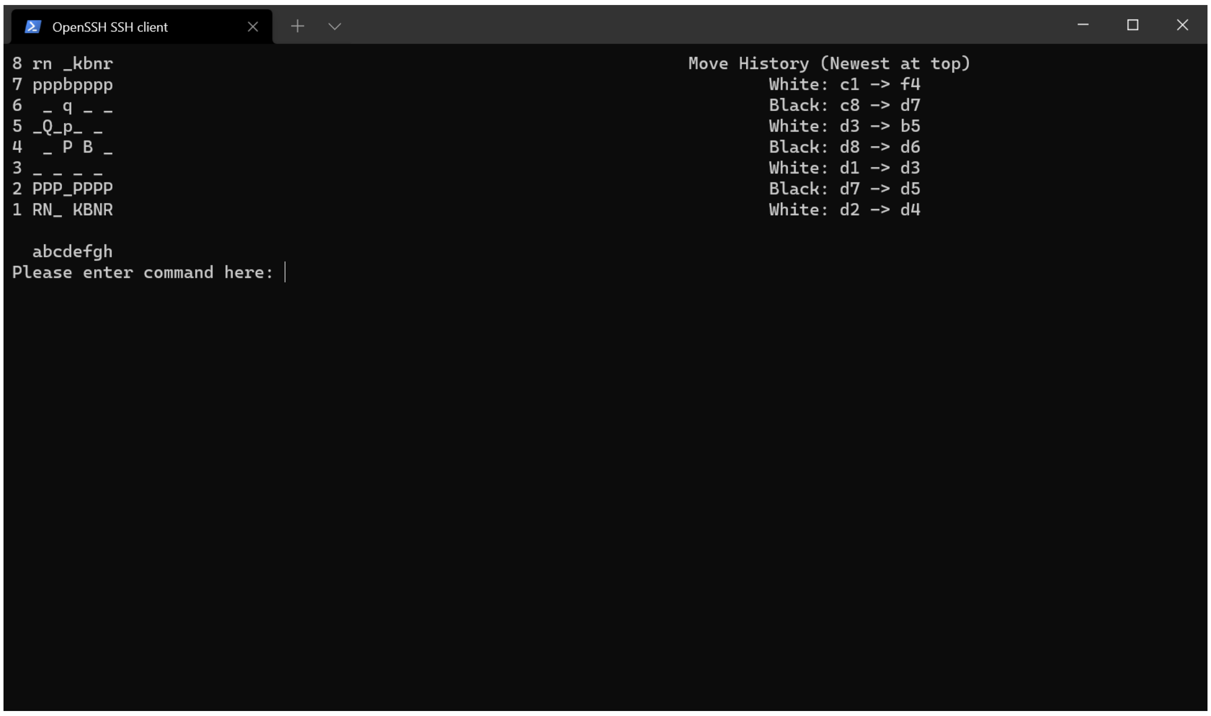
```
8 rnb_kbnr                                                         Move History (Newest at top)
7 ppp pppp                                                                 White: d3 -> b5
6  _ q _ _                                                                 Black: d8 -> d6
5 _Q_p_ _                                                                  White: d1 -> d3
4  _ P _ _                                                                 Black: d7 -> d5
3 _ _ _ _                                                                  White: d2 -> d4
2 PPP_PPPP
1 RNB KBNR

  abcdefgh
Black is in check.
Please enter command here: move c8 d7
```

By typing the command “move c8 d7”, the program will move the bishop from the position c8 to d7. As specified in the property of bishops, the program moved the bishop one position in a diagonal direction.



```
8 rn _kbnr                                                         Move History (Newest at top)
7 pppbpppp                                                                 Black: c8 -> d7
6  _ q _ _                                                                 White: d3 -> b5
5 _Q_p_ _                                                                  Black: d8 -> d6
4  _ P _ _                                                                 White: d1 -> d3
3 _ _ _ _                                                                  Black: d7 -> d5
2 PPP_PPPP                                                                 White: d2 -> d4
1 RNB KBNR

  abcdefgh
Please enter command here: 
```

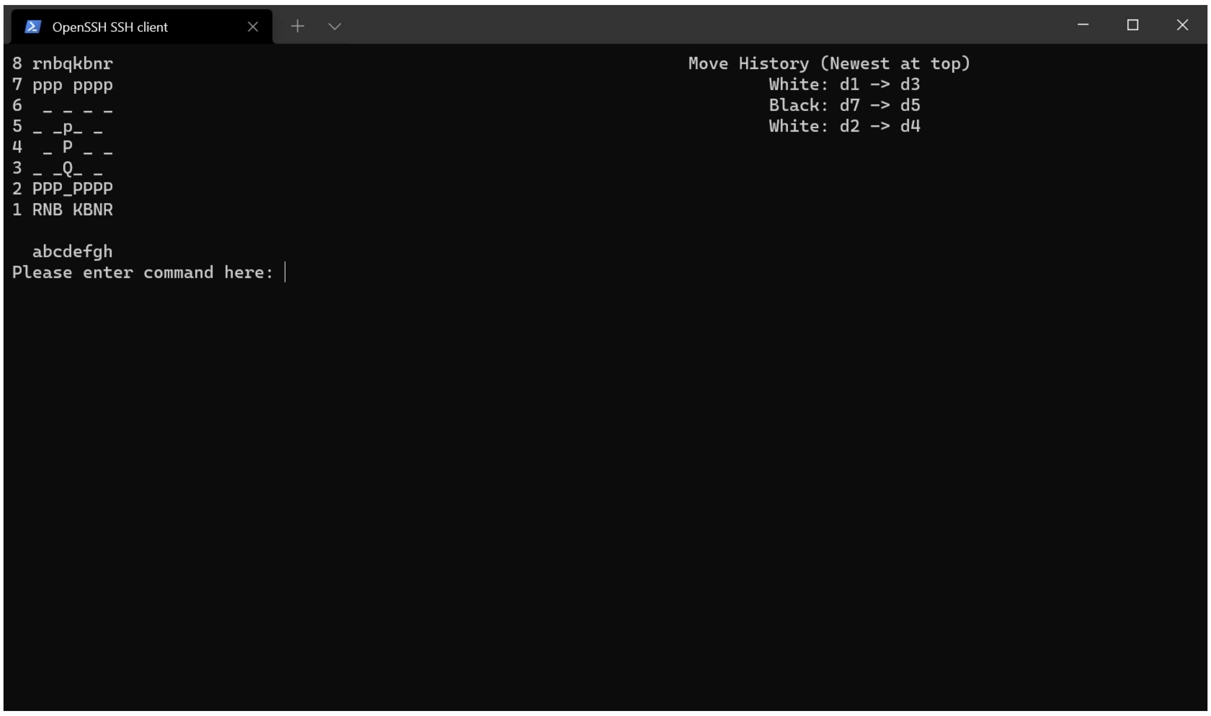
The diagram below shows the movement of a bishop of the white-side player.



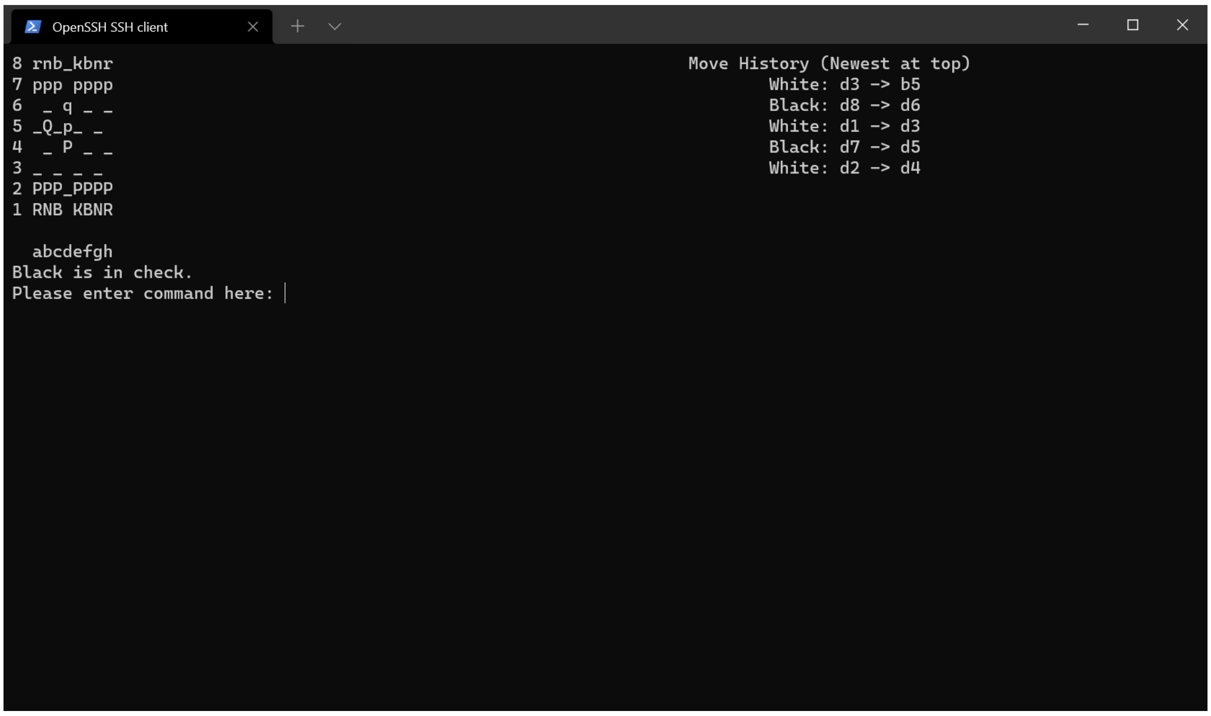
```
8 rn _kbnr                                                         Move History (Newest at top)
7 pppbpppp                                                                 White: c1 -> f4
6  _ q _ _                                                                 Black: c8 -> d7
5 _Q_p_ _                                                                  White: d3 -> b5
4  _ P B _                                                                 Black: d8 -> d6
3 _ _ _ _                                                                  White: d1 -> d3
2 PPP_PPPP                                                                 Black: d7 -> d5
1 RN_ KBNR                                                                 White: d2 -> d4

  abcdefgh
Please enter command here: 
```

## Moving a Queen

how a queen move



```
8 rnbqkbnr                                                         Move History (Newest at top)
7 ppp pppp                                                                 White: d1 -> d3
6  _ _ _ _                                                                 Black: d7 -> d5
5 _ _p_ _                                                                  White: d2 -> d4
4  _ P _ _
3 _ _Q_ _
2 PPP_PPPP
1 RNB KBNR

  abcdefgh
Please enter command here:
```



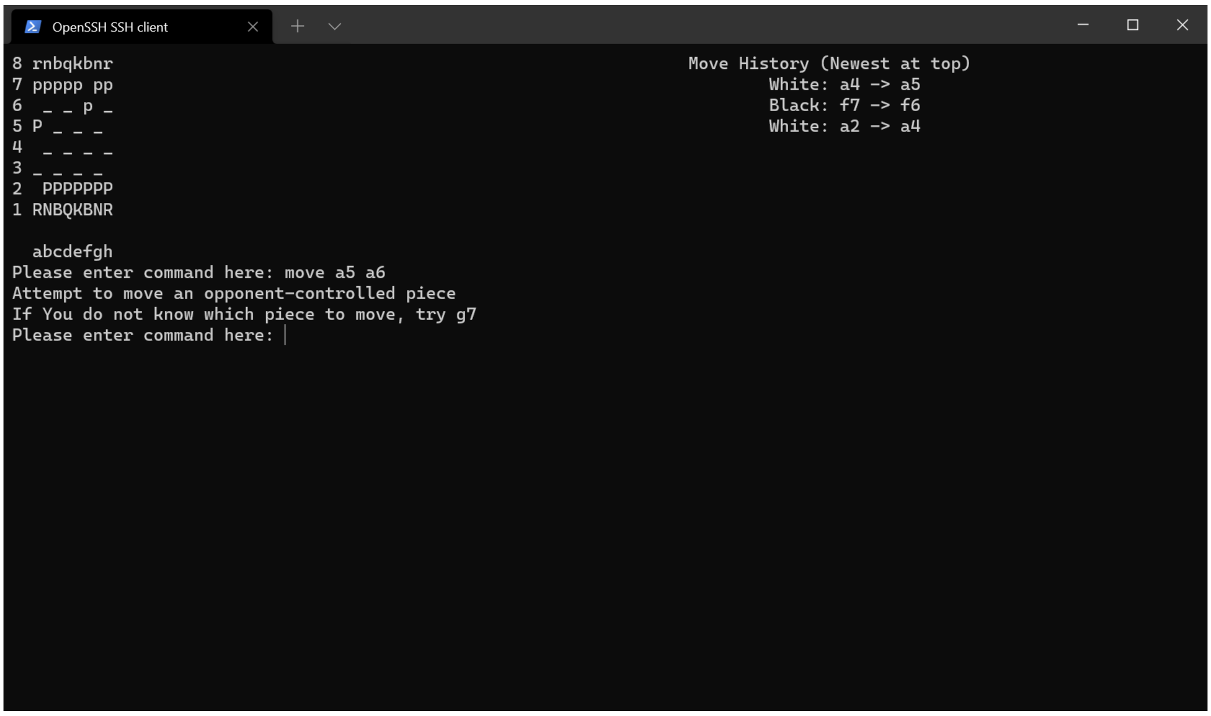
```
8 rnb_kbnr                                                         Move History (Newest at top)
7 ppp pppp                                                                 White: d3 -> b5
6  _ q _ _                                                                 Black: d8 -> d6
5 _Q_p_ _                                                                  White: d1 -> d3
4  _ P _ _                                                                 Black: d7 -> d5
3 _ _ _ _                                                                  White: d2 -> d4
2 PPP_PPPP
1 RNB KBNR

  abcdefgh
Black is in check.
Please enter command here:
```

## Illegal Move Command

### Moving an Opponent-Controlled Piece

From the Move History on the right side, we can see it is the black side’s turn now, if that player instead moves a white piece (in this example below, the white pawn at a5), the program prompts, “*Attempting to move an opponent-controlled piece*”, so that the client knows he or she has chose to move a piece of another side.

If the player does not know which piece could be moved, he or she could refer to the recommendation, such as in this example, it is suggested to move the piece at g7, which is a black pawn.

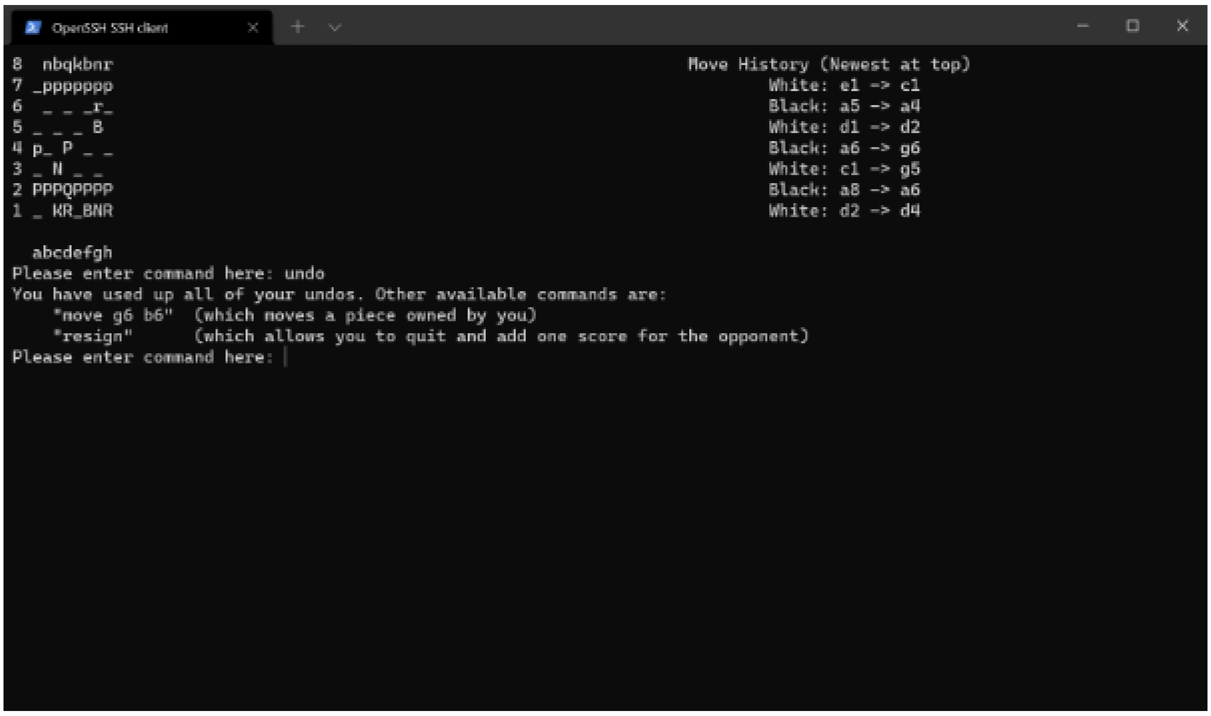


```
8 rnbqkbnr                                                          Move History (Newest at top)
7 ppppp pp                                                                  White: a4 -> a5
6  _ _ p _                                                                  Black: f7 -> f6
5 P _ _ _                                                                   White: a2 -> a4
4  _ _ _ _
3 _ _ _ _
2  PPPPPPP
1 RNBQKBNR

  abcdefgh
Please enter command here: move a5 a6
Attempt to move an opponent-controlled piece
If You do not know which piece to move, try g7
Please enter command here: 
```

It is worthy to point out that this suggested move does not involve tactical consideration at all (otherwise it is considered as a hint and a hint really impacts the chess game in a very bad way). Instead, this suggested piece is only randomly chosen from every piece on the board controlled by the current player and has available move(s).

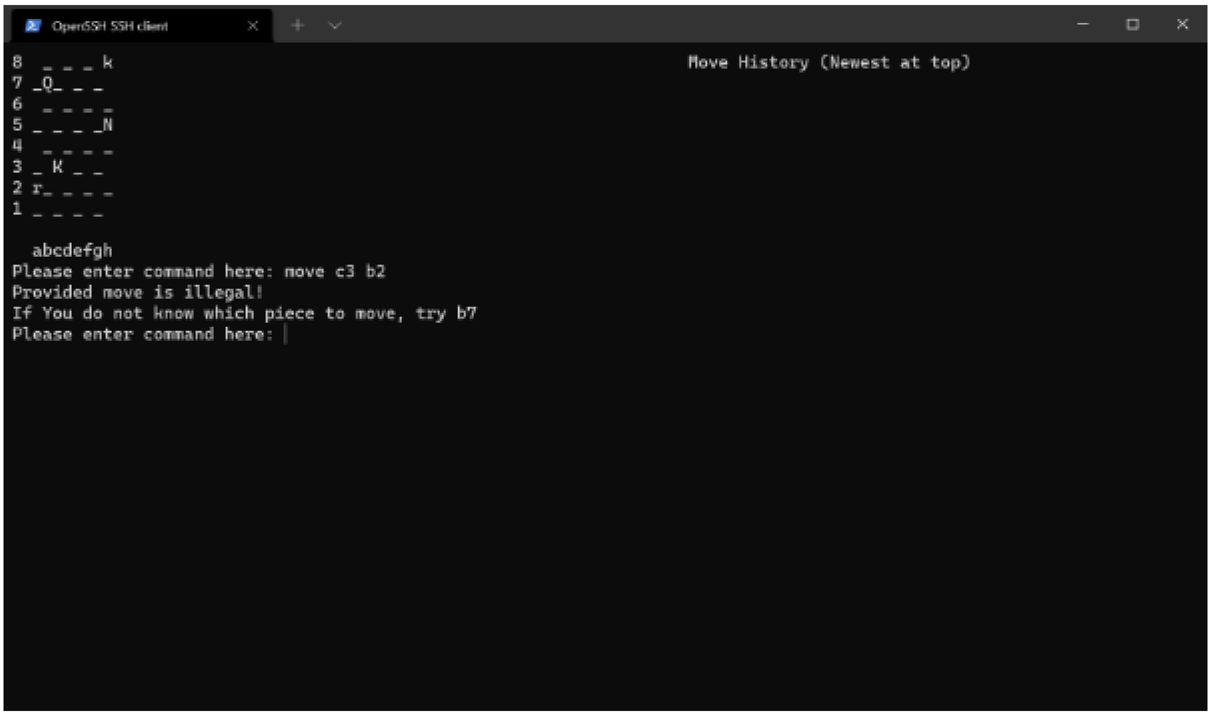
## Undo & Move History

The number of undo was specified before starting the game, once a player has used up all its undo chances or just simply, undoing is not allowed (specified as 0 in the game command), further undoing is prohibited. Just as the picture below shows, there were no more undo chances for the black player, thus the program rejected the undo command. Instead, the program prompted all other available commands, including “move” and “resign”. Again, this move command was randomly rolled from all available movements, it did not involve any tactics or anticipations.



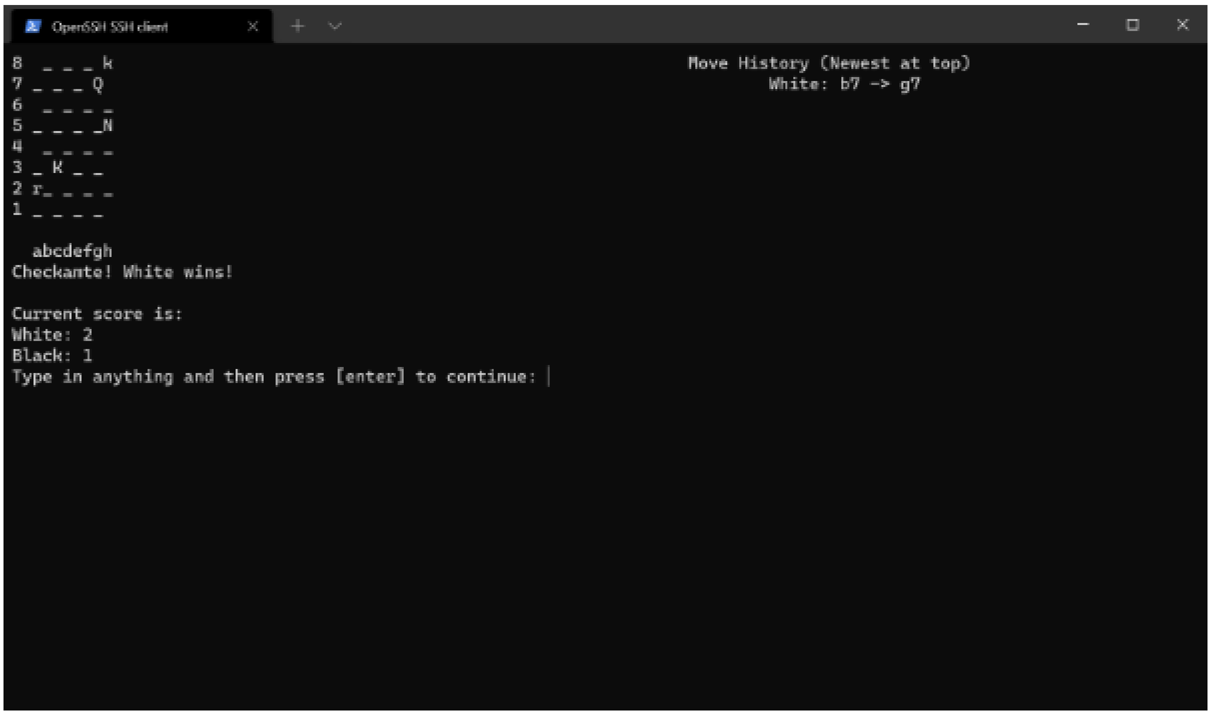
```
8  nbqkbnr                                                          Move History (Newest at top)
7 _ppppppp                                                                  White: e1 -> c1
6  _ _ _r_                                                                  Black: a5 -> a4
5 _ _ _ B                                                                   White: d1 -> d2
4 p_ P _ _                                                                  Black: a6 -> g6
3 _ N _ _                                                                   White: c1 -> g5
2 PPPQPPPP                                                                  Black: a8 -> a6
1 _ KR_BNR                                                                  White: d2 -> d4

  abcdefgh
Please enter command here: undo
You have used up all of your undos. Other available commands are:
    "move g6 b6"  (which moves a piece owned by you)
    "resign"      (which allows you to quit and add one score for the opponent)
Please enter command here: 
```

## Check & Checkmate



OpenSSH SSH client

```
8  _ _ _ k                                                   Move History (Newest at top)
7 _Q_ _ _
6  _ _ _ _
5 _ _ _ _N
4  _ _ _ _
3 _ K _ _
2 r_ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: move c3 b2
Provided move is illegal!
If You do not know which piece to move, try b7
Please enter command here: 
```



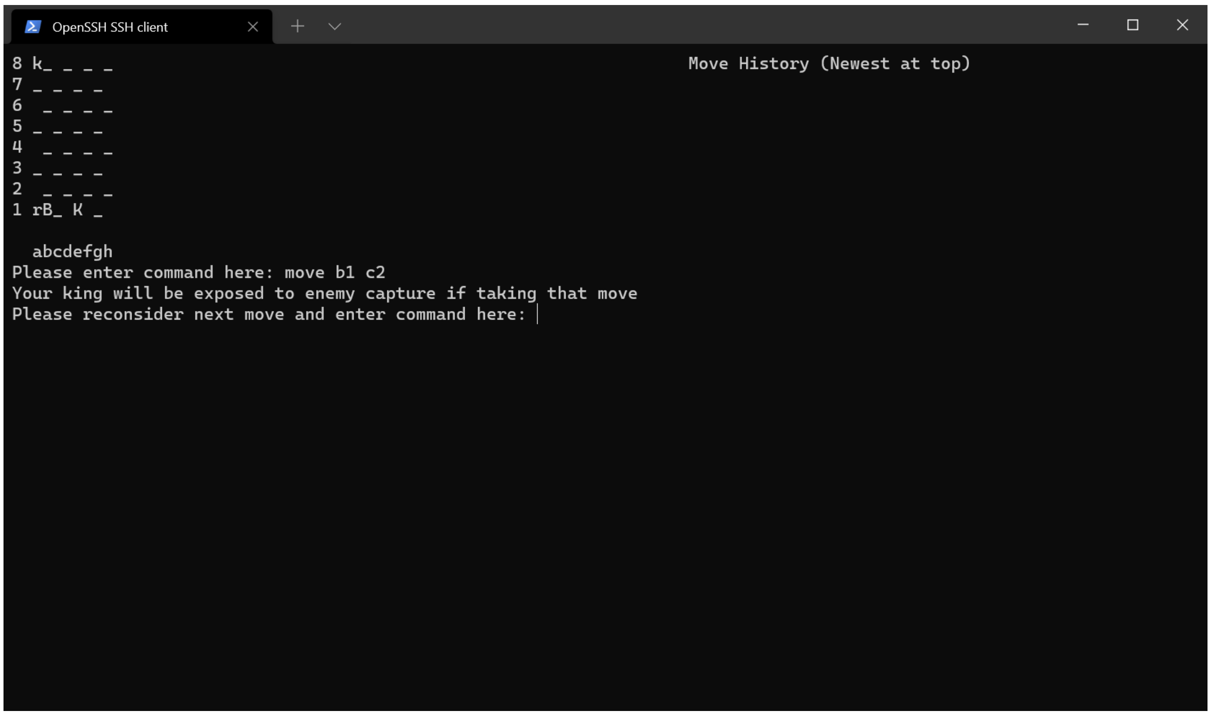
OpenSSH SSH client

```
8  _ _ _ k                                                   Move History (Newest at top)
7 _ _ _ Q                                                            White: b7 -> g7
6  _ _ _ _
5 _ _ _ _N
4  _ _ _ _
3 _ K _ _
2 r_ _ _ _
1 _ _ _ _

  abcdefgh
Checkamte! White wins!

Current score is:
White: 2
Black: 1
Type in anything and then press [enter] to continue: 
```

b7 -> g7, checkmate (describe). no undo allowed in this one thus directly win

## Exposing the King to Enemy Check

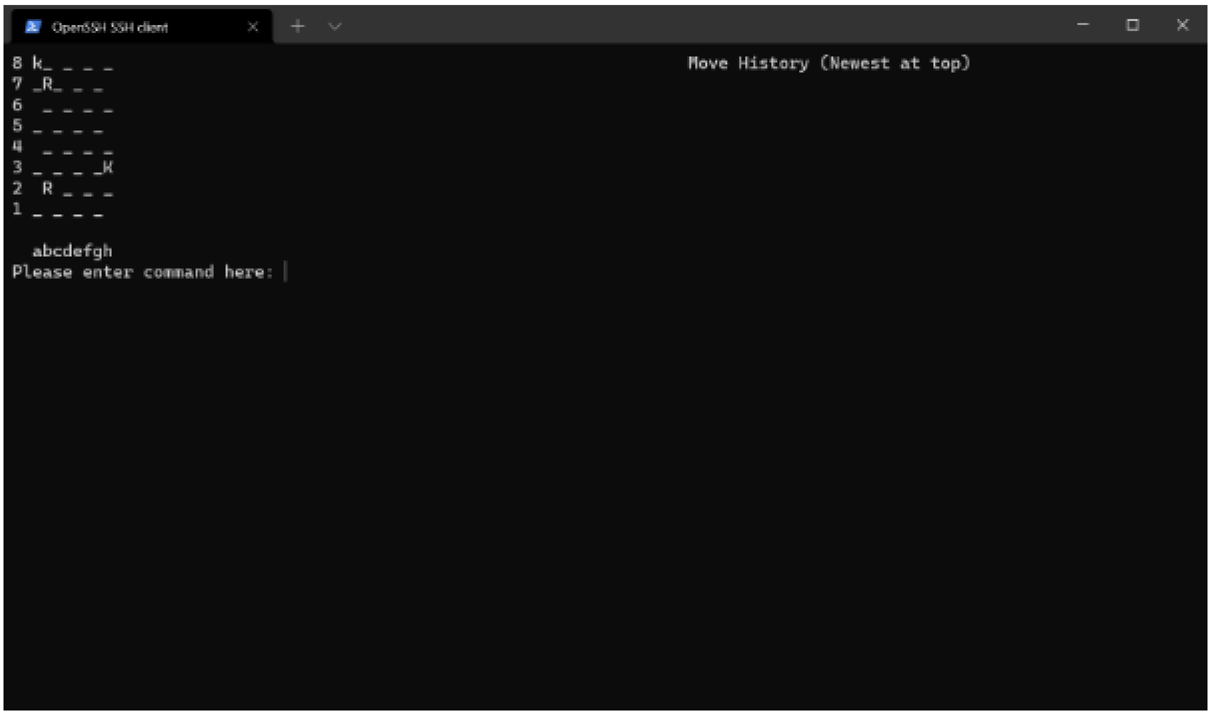
Players of both sides are disallowed to put their own kings under the opponent's check. Take the situation below as an example. A white player typed in command “*move b1 c2*” and pressed [enter], attempting to move the bishop. But this action was rejected by the program since this move put the white king under the check of a black rook (at a1) . The program considered such a move as “illegal move” and a corresponding prompt was printed.



```
8 k_ _ _ _                                                          Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 rB_ K _

  abcdefgh
Please enter command here: move b1 c2
Your king will be exposed to enemy capture if taking that move
Please reconsider next move and enter command here: 
```

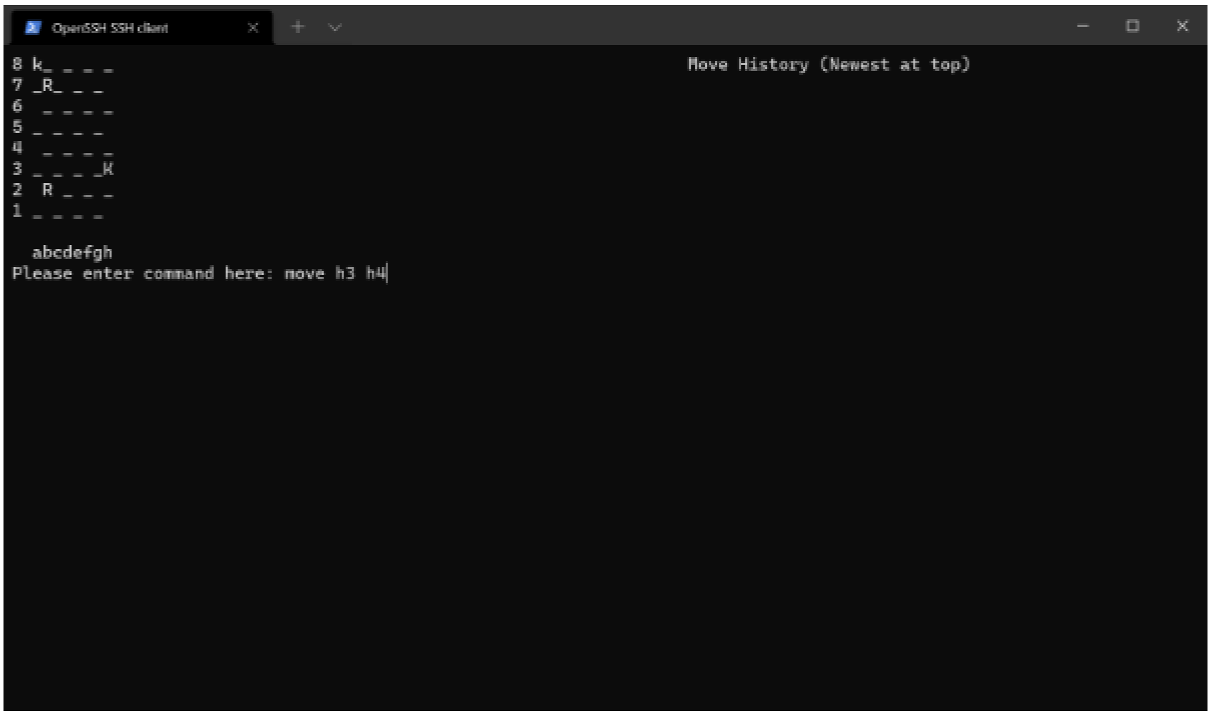
## Stalemate

what is a stalemate. consider the setup below and it is currently white's turn.



```
8 k_ _ _ _                                                    Move History (Newest at top)
7 _R_ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _K
2  R _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here:
```

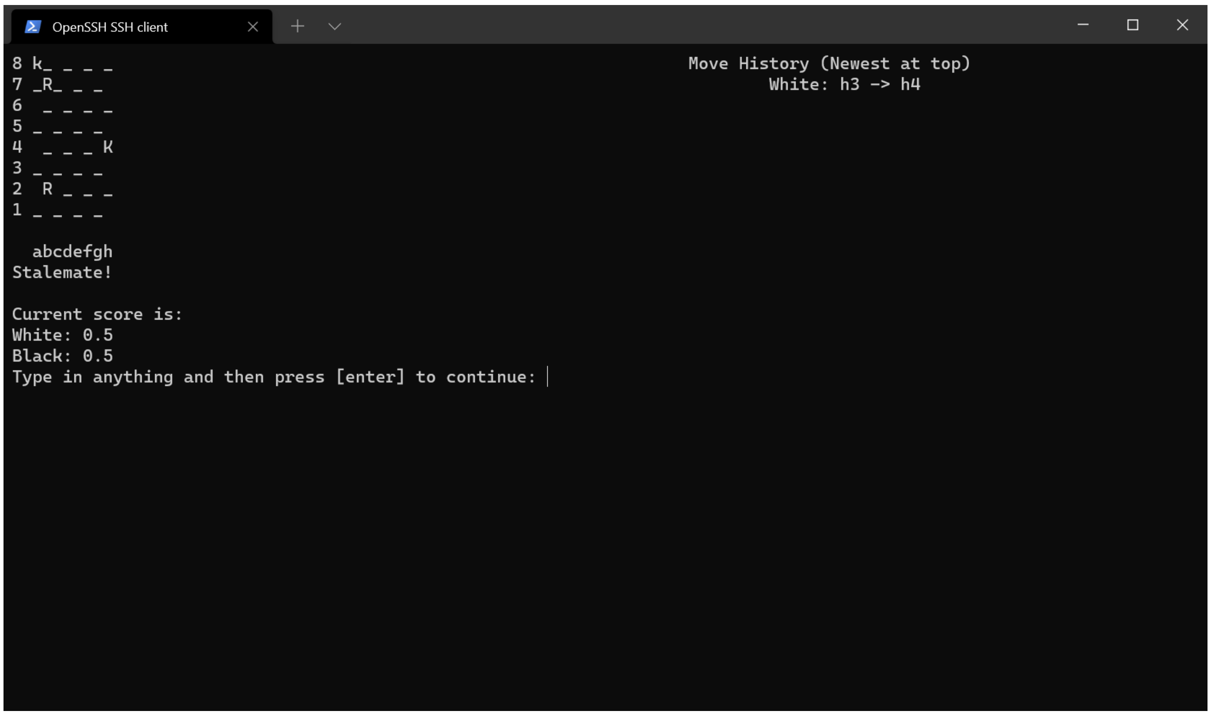
move h3 h4 and pressed [enter]



```
8 k_ _ _ _                                                    Move History (Newest at top)
7 _R_ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _K
2  R _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here: move h3 h4
```

Now it is black's turn, but black is currently not in check, however, the black player does not have any available moves. The only possible moving positions are a7, b7, b8, but a7 and b8 are controlled by the white rook at b7, and moving black king to b7 and capturing the rook at b7 exposes the king under check from white rook at b2. In other words, the black player has no available moves that does not put his or her king under check.

But be advised that the black king is not under check right now, the black king itself was just not able to move. In this situation, a stalemate has happened, just as the program prompts below. A stalemate added 0.5 to both white and black players grand score.



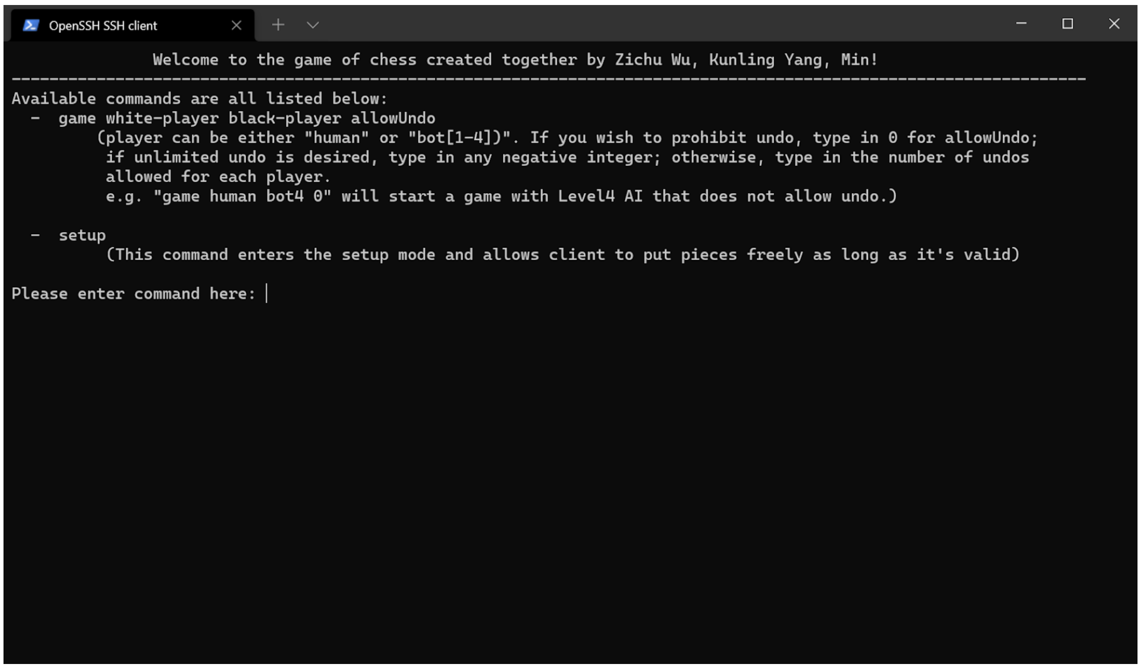
```
8 k_ _ _ _                                                        Move History (Newest at top)
7 _R_ _ _                                                                 White: h3 -> h4
6  _ _ _ _
5 _ _ _ _
4  _ _ _ K
3 _ _ _ _
2  R _ _ _
1 _ _ _ _

  abcdefgh
Stalemate!

Current score is:
White: 0.5
Black: 0.5
Type in anything and then press [enter] to continue:
```

## Setup Mode

Below is the walkthrough of the "*setup"* command of this project.

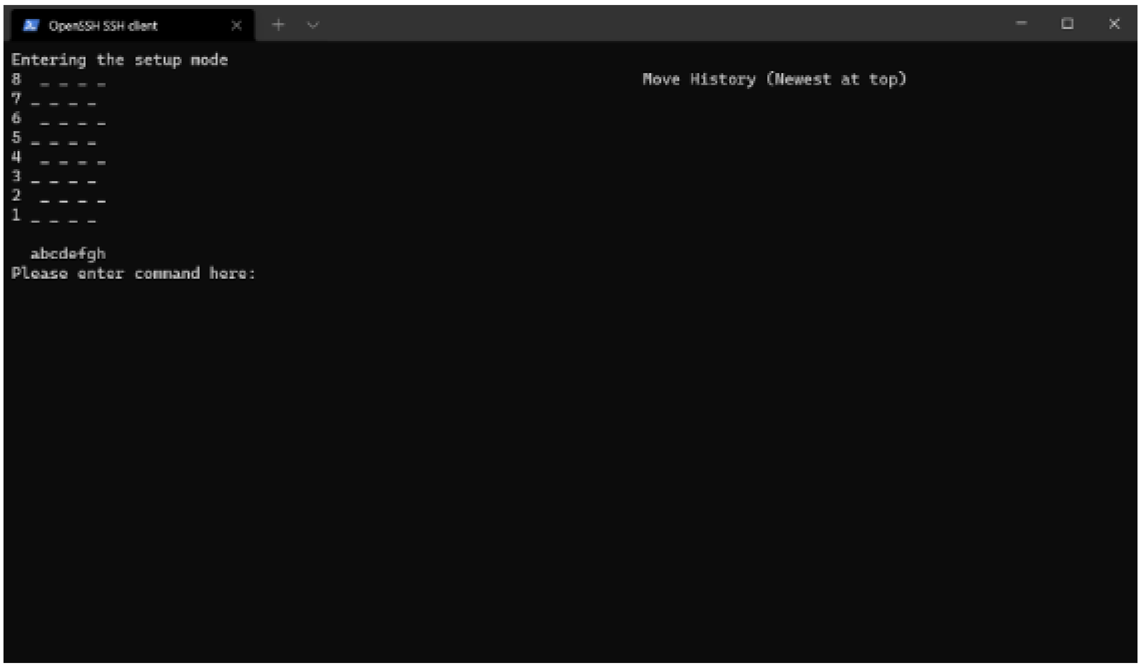
The setup command will enter the setup mode, which removes all the chess pieces from the current game and the client will be able to freely place the chess pieces on both sides. However, the client is not allowed to go back to the main title screen unless the current setup has passed the valid check.

To enter the setup mode, enter "*setup*" at the main title page.



```
               Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
-----------------------------------------------------------------------------------------------------------------
Available commands are all listed below:
  -  game white-player black-player allowUndo
         (player can be either "human" or "bot[1-4]")". If you wish to prohibit undo, type in 0 for allowUndo;
          if unlimited undo is desired, type in any negative integer; otherwise, type in the number of undos
          allowed for each player.
          e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
          (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here: |
```

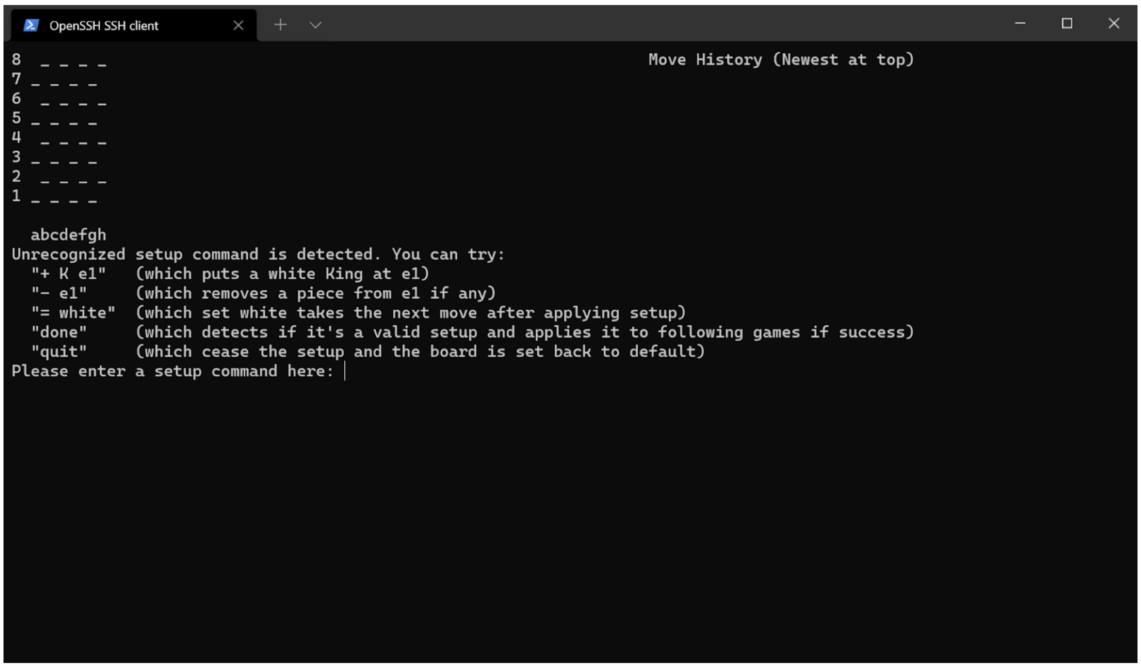
Type in this command guides the program into the setup mode so the client can place the pieces as they like. Firstly, a welcome is printed, "*Entering the setup mode*" .



```
Entering the setup mode
8  _ _ _ _                                                         Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _

  abcdefgh
Please enter command here:
```

## Unrecognized Command

If the client does not know what commands are available or just simply types in a wrong command, and then provides an unrecognized command (or simply press the [enter] button on the keyboard), the program would first clear whatever is on the screen, and then prompt the client to enter a valid setup command again, along with a list of all possible commands in the setup mode.



```
8  _ _ _ _                                                      Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _

  abcdefgh
Unrecognized setup command is detected. You can try:
  "+ K e1"   (which puts a white King at e1)
  "- e1"     (which removes a piece from e1 if any)
  "= white"  (which set white takes the next move after applying setup)
  "done"     (which detects if it's a valid setup and applies it to following games if success)
  "quit"     (which cease the setup and the board is set back to default)
Please enter a setup command here: 
```

In the picture above, all acceptable setup command types are listed inside the double quotation marks and inside the following parenthesis is a brief explanation of what this command does.

| | |
|---|---|
| *"+ K e1"* | This commands puts a white King at e1 position. "K" can be replaced by any acceptable pieces(please refer to the Acceptable Pieces Type), an uppercase letter for a white piece and a lower case letter for a black piece. "*e1*" can be replaced by any available position on the board. |
| *"- e1"* | This command removes a piece from position *"e1"* if there is any piece. "*e1*" can be replaced by any available position on the board. |
| *"= white"* | This command sets the next player to take a move. *"white*" can be replaced by *"black"* if the client desires black side to take the next turn. |
| *"done"* | This command triggers the setup inspection, and guides the client back to the main title screen once this setup has been considered as valid. |
| *"quit"* | This command ceases all changes, restores all the pieces back to the default, and directly guides the client back to the main title page. |

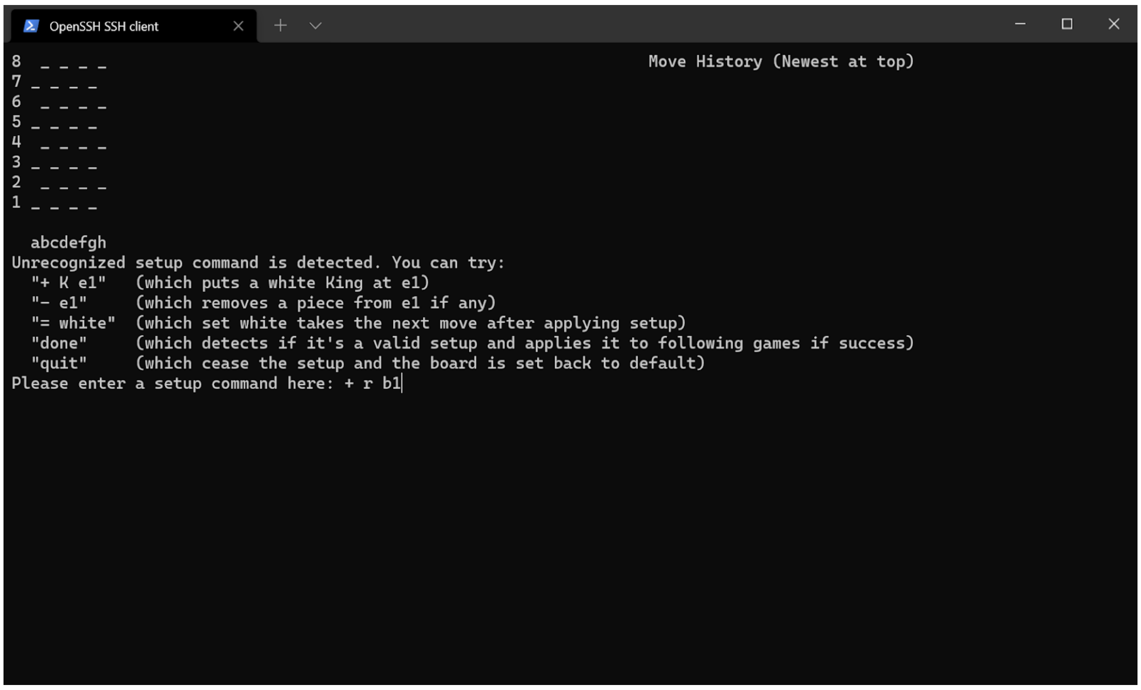
## Adding a Piece to Setup Board

By typing a command starting with "+", the client can place any acceptable pieces (please refer to the Acceptable Pieces Type) onto the chessboard, where an uppercase letter represents a white piece and a lowercase letter represents a black piece; "*e1*" can be replaced by any available position on the board.

Be advised that this command must follow the template so that "+", piece type and position are in the right order and are separated by at least one whitespace , and the client must put the whole command in a single line (i.e. type in all required parts and then press [enter]). Any whitespace at the beginning or at the end of this command is ignored.

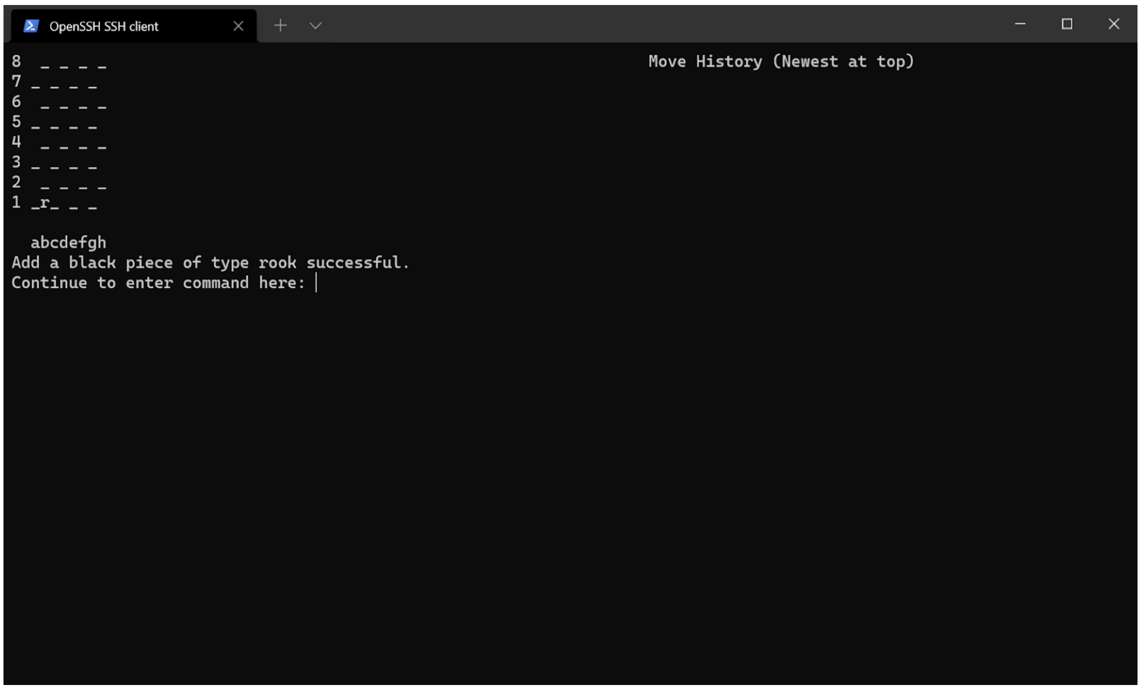
It is worthy to point out that adding a piece is not considered as a MOVE, thus this command does not affect the Move History on the right side at all. It remains empty during the whole setup mode, in fact.

For example, the client puts in "+ r b1" and then presses the [enter] key as below:



OpenSSH SSH client

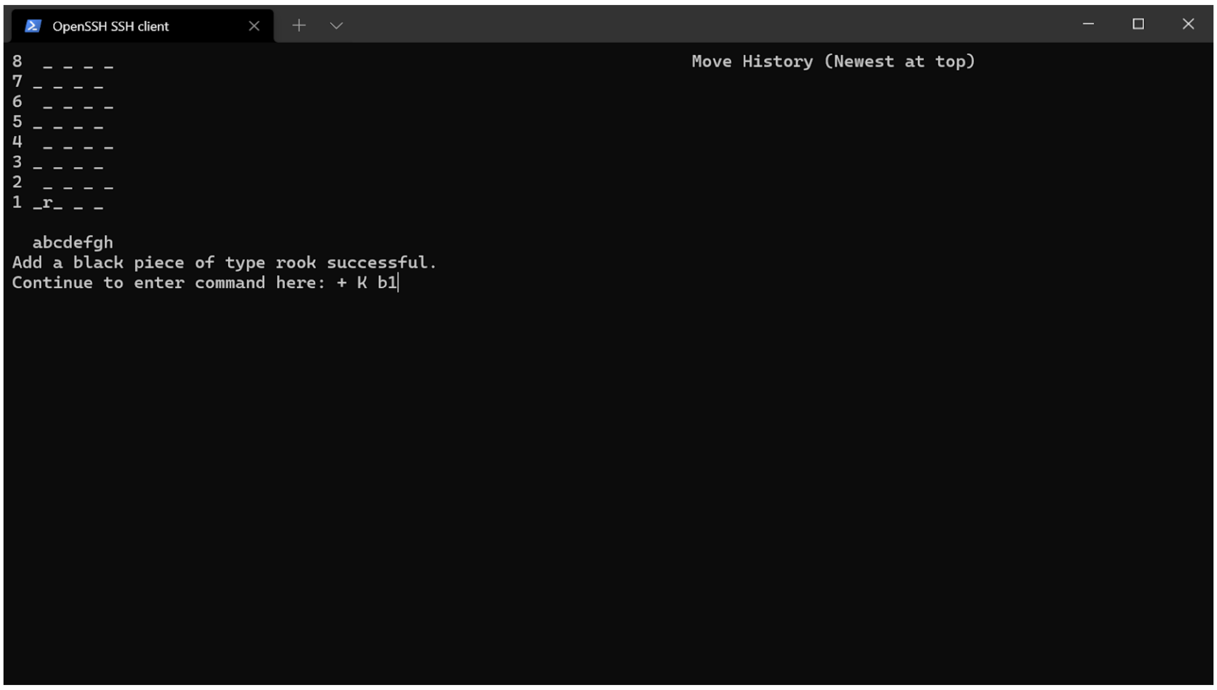
```
8  _ _ _ _                                                         Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _

  abcdefgh
Unrecognized setup command is detected. You can try:
  "+ K e1"   (which puts a white King at e1)
  "- e1"     (which removes a piece from e1 if any)
  "= white"  (which set white takes the next move after applying setup)
  "done"     (which detects if it's a valid setup and applies it to following games if success)
  "quit"     (which cease the setup and the board is set back to default)
Please enter a setup command here: + r b1
```



```
8  _ _ _ _                                                           Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _r_ _ _

  abcdefgh
Add a black piece of type rook successful.
Continue to enter command here: 
```

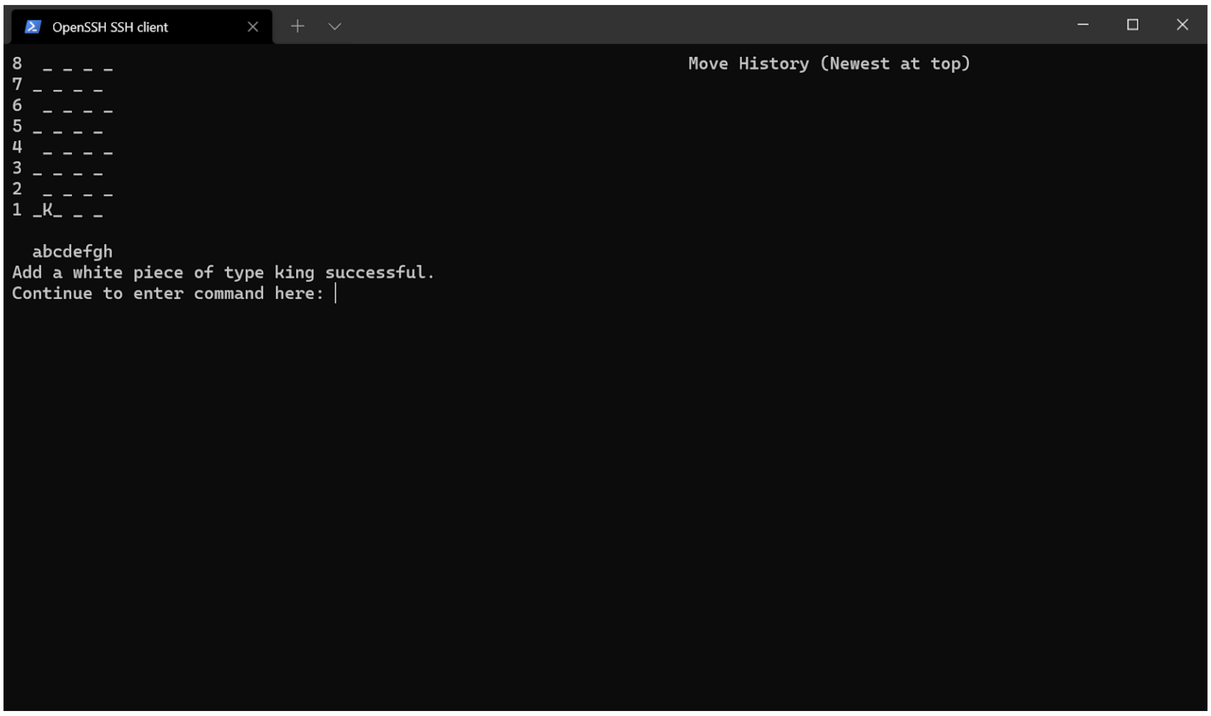
The real-time board is updated, and a lowercase r is already there, representing a black rook. If an invalid position or piece type is entered, this whole command is considered as an unrecognized command, and all effects of this command are removed immediately. The program prompts the client for all possible setup commands.

If the designated position is already occupied by another piece, the new piece placed by the add setup command will overwrite the old one (the one already on the chessboard). For example, if the client continues to type in “+ K b1” and presses then [enter] key:



```
8  _ _ _ _                                                           Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _r_ _ _

  abcdefgh
Add a black piece of type rook successful.
Continue to enter command here: + K b1
```

As the picture below shows, after the client typing in “+ K b1”, a white king is generated at b1 that was previously occupied by the rook previously. Thus, the black rook is removed and a white king is placed there.



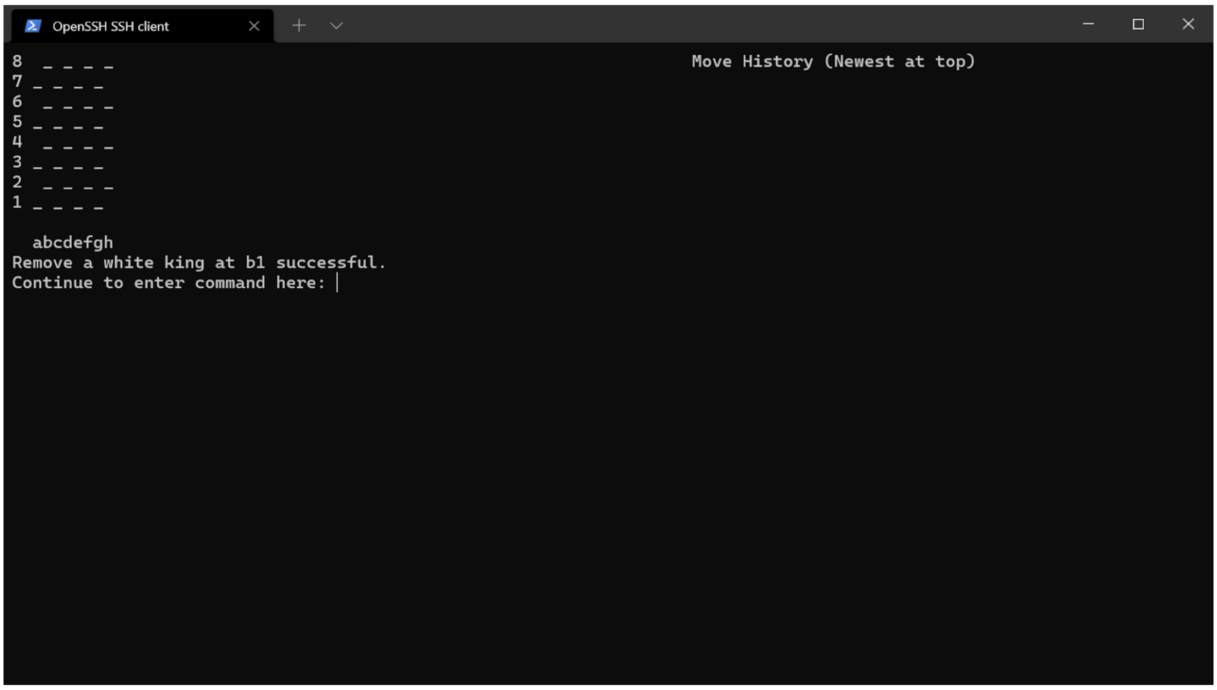
```
8  _ _ _ _                                                       Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _K_ _ _

  abcdefgh
Add a white piece of type king successful.
Continue to enter command here:
```

In the setup mode, the number of pieces are unrestricted (that means the client can put 10 white knights onto the board, or 4 queens, as long as it seems interesting, but at most one king for each side and this is only checked when “done” is entered).

## Removing a Piece

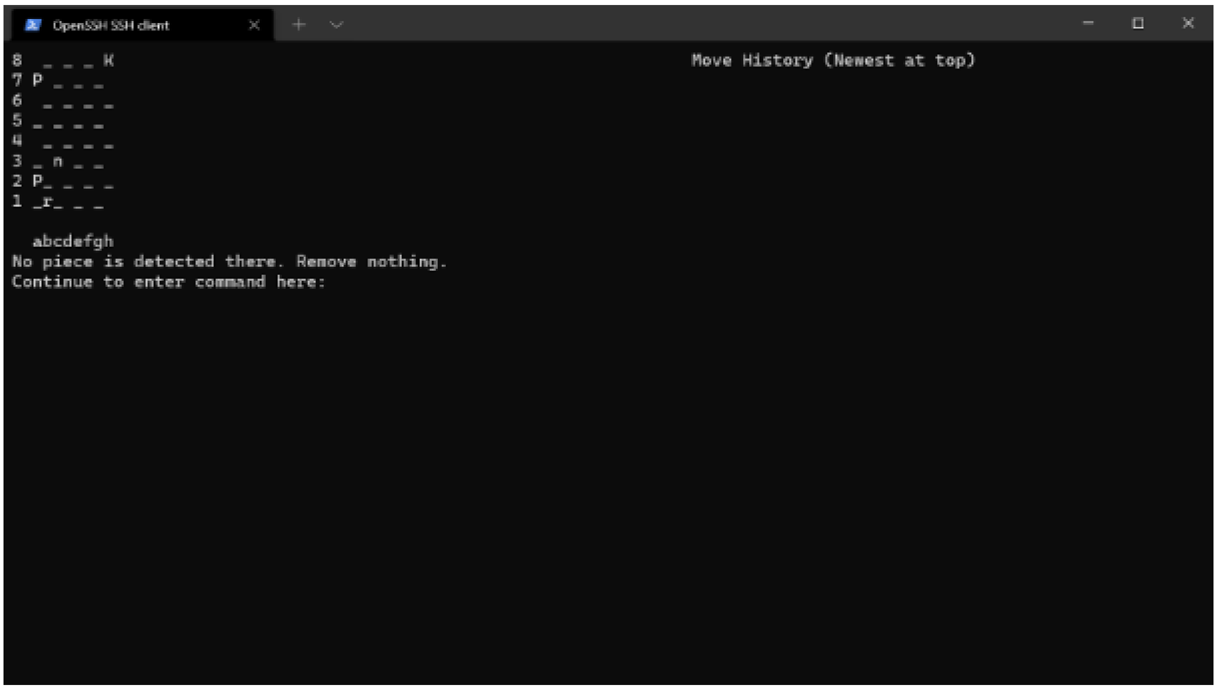
This command removes a piece from a designated position on the chessboard. If the designated location is beyond the chessboard, this whole command is considered as an unrecognized command, and all effects of this command are removed immediately. The program prompts the client for all possible setup commands.

Continuing the example from the last part, the client types in “- b1” and presses [enter], then the white king already there is immediately removed from the board, just as the picture below shows. The program prompts the user what has been removed from the board, and at which position it is, “Remove a white king at b1 successful”, so that the client can have a quick double-check if an accidental removal has happened.



```
8  _ _ _ _                                                  Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _

  abcdefgh
Remove a white king at b1 successful.
Continue to enter command here: |
```

However, if the designated position is indeed a valid position and is just simply not occupied by any piece, the program is prompting a message, “No piece is detected there. Remove nothing.”, just as the picture below:

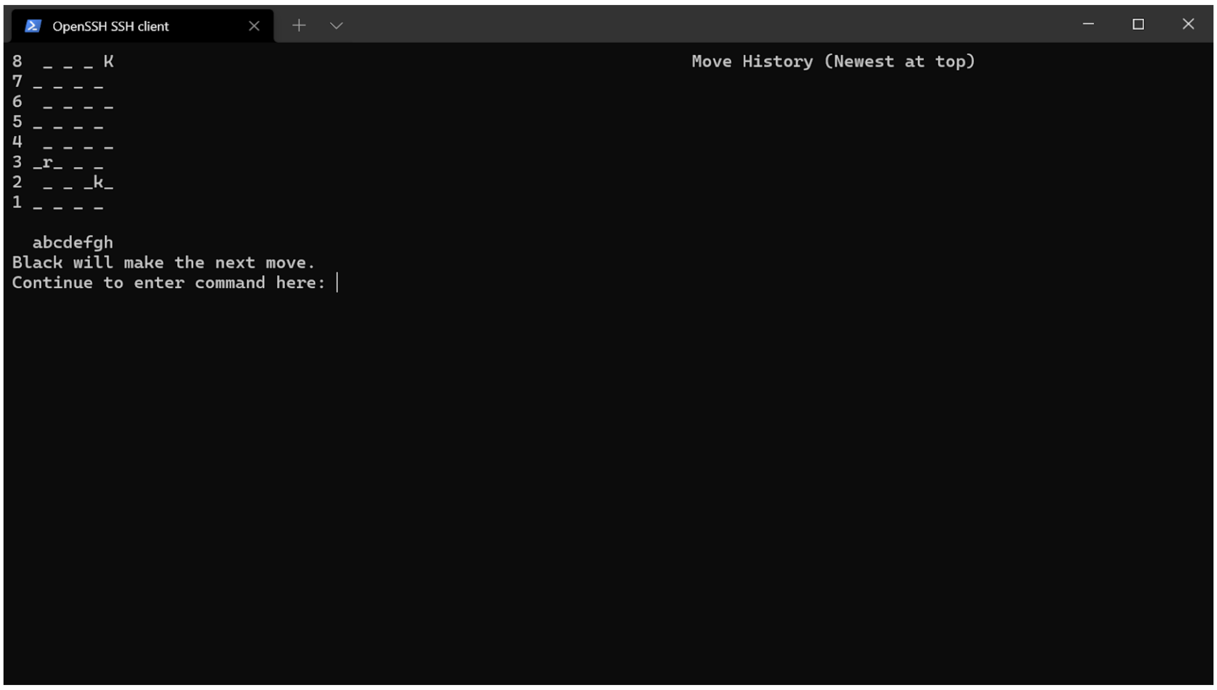


```
8  _ _ _ K                                                         Move History (Newest at top)
7 P _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _ n _ _
2 P_ _ _ _
1 _r_ _ _

  abcdefgh
No piece is detected there. Remove nothing.
Continue to enter command here:
```

## Set Next Turn Player

This command sets the next player to take a move after this setup has been applied and takes effect in all subsequential games. Possible options are either “white” or “black”, and the player of the selected side takes the next move in the subsequential games.

For example, if the client types in “= black” and presses [enter]:



```
8  _ _ _ K                                                  Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _r_ _ _
2  _ _ _k_
1 _ _ _ _

  abcdefgh
Black will make the next move.
Continue to enter command here: 
```

The program prompts “Black will make the next move” so that the client knows his or her command has already taken effect. Once this setup has been checked and applied, the black side (maybe a human or a computer player)  will take the next move.

And the situation is generally the same when “=white” is entered except for different sides inside the prompts, and thus is omitted.

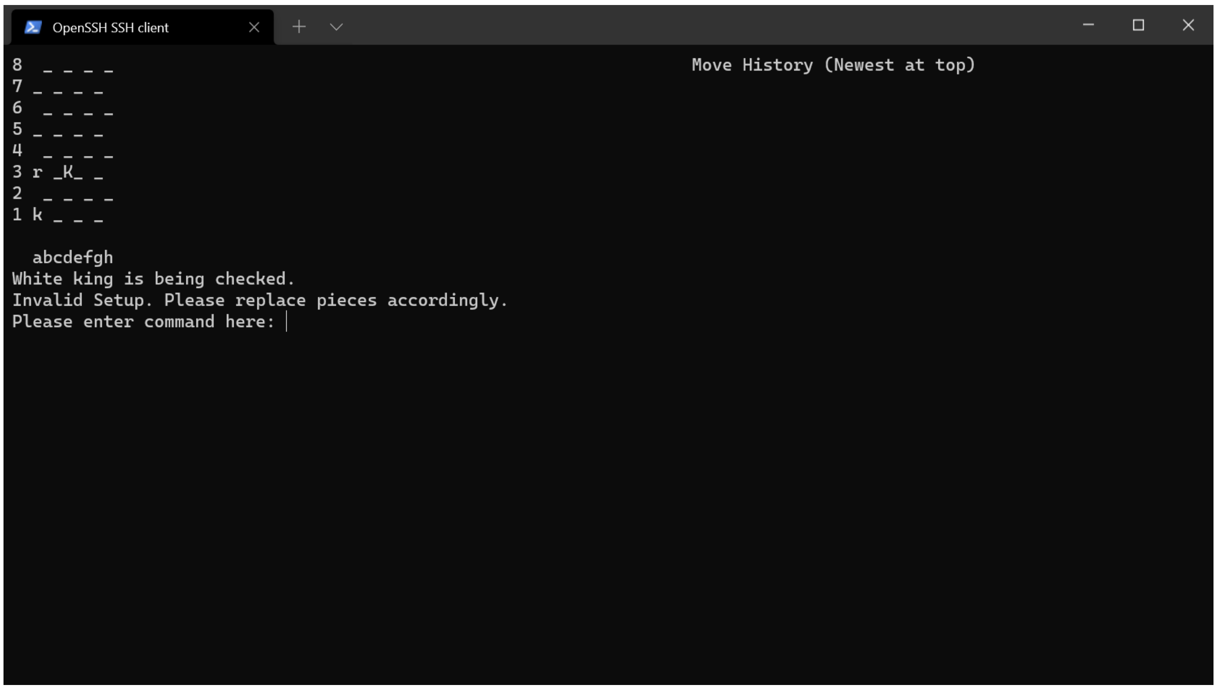
## Finishing Setup

- ### Invalid Setup

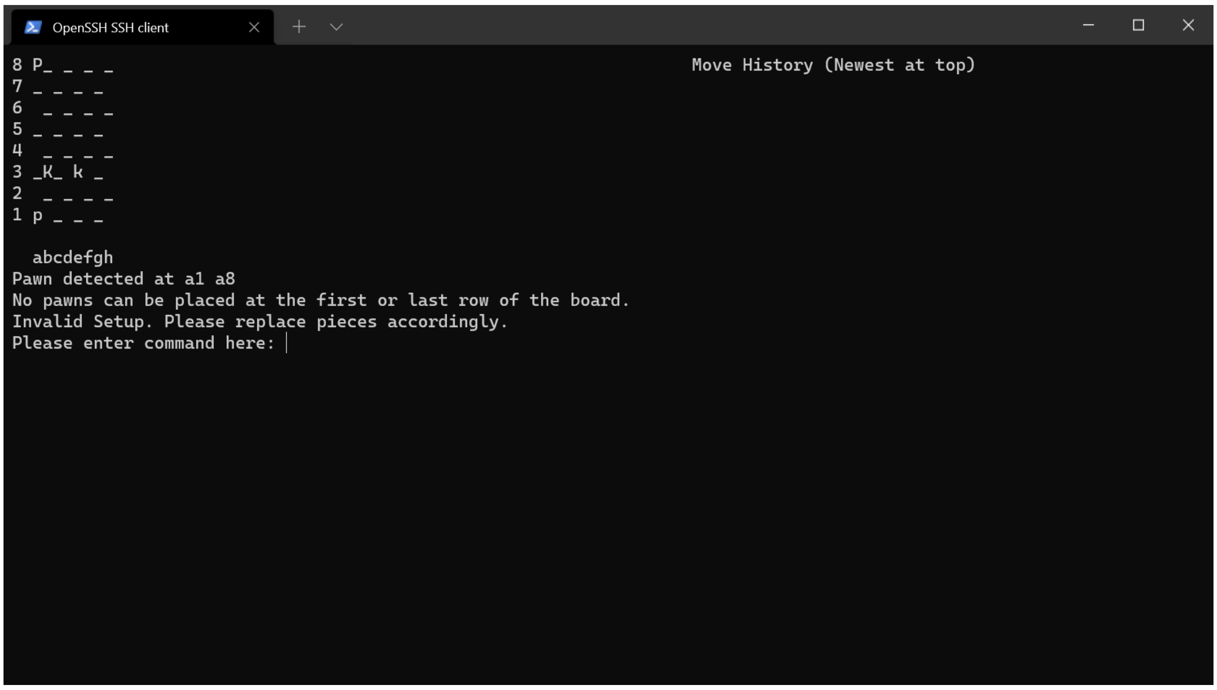
Despite the freedom to place pieces on the board in the setup mode, the command "*done*" is rather restrictive and rigorous. If at least one of the following rules is not meet after the client puts in "*done"* and presses [enter], the program forces the client to remain in the setup mode and further modifies the setup until all requirements are met:

- ☐ There must be exactly one white king (represented by "K", a uppercase letter) and exactly one black king (represented by "k", a lowercase letter) on the board.
- ☐ There is no pawn at the first row nor last row (marked by Row 1 and Row 8, respectively) since these pawns require immediate promotion.
- ☐ Neither white king nor the black king is being checked.

In the picture below is an invalid setup since the black rook at a3 puts the white king at d3 in check thus this violates the third rule. Immediately after the real-time board, goes the prompts, "*White king is being checked.*" To tell the user which king is being checked.



```
8  _ _ _ _                                                 Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 r _K_ _
2  _ _ _ _
1 k _ _ _

  abcdefgh
White king is being checked.
Invalid Setup. Please replace pieces accordingly.
Please enter command here: 
```

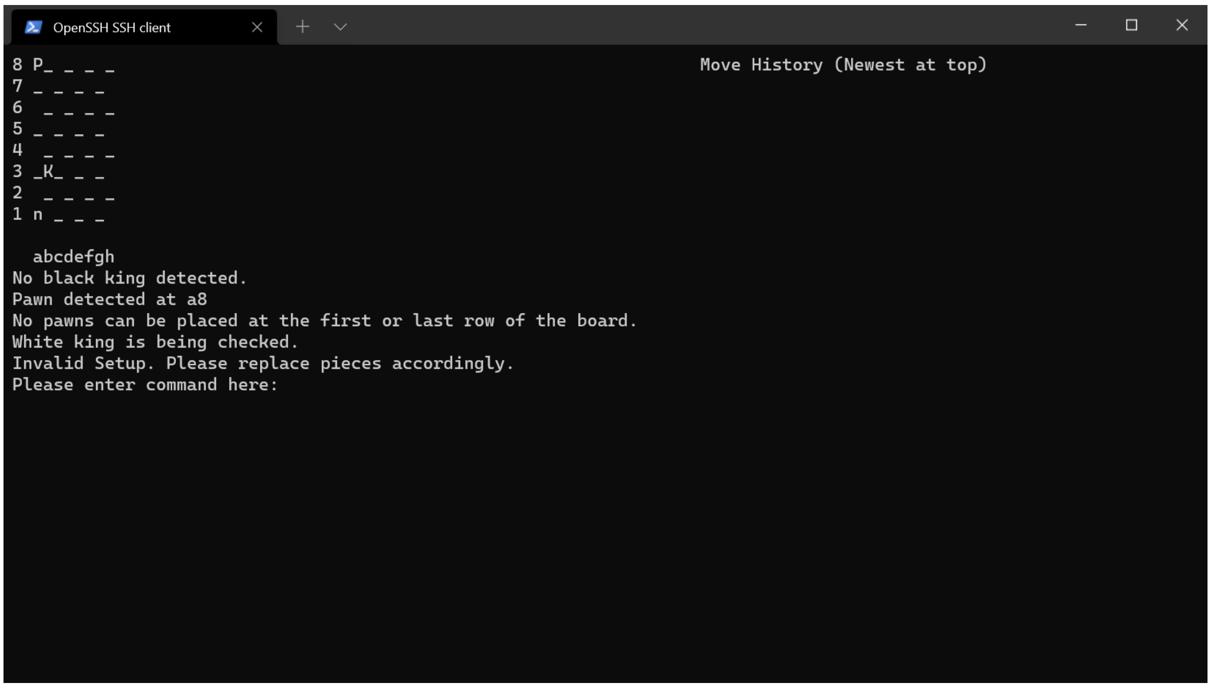


```
8 P_ _ _ _                                                           Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _K_ k _
2  _ _ _ _
1 p _ _ _

  abcdefgh
Pawn detected at a1 a8
No pawns can be placed at the first or last row of the board.
Invalid Setup. Please replace pieces accordingly.
Please enter command here: 
```

In the picture above is an invalid setup, and the reason is stated just as the program prompts, “*Pawn detected at a1 a8*”. This is a violation of rule2, “*No pawns can be placed at the first or last row of the board”*.  Be advised that the black pawn will only get promotion when reaches the first row (Row 1 on the board) but it is still impossible for it to be placed at a place that it can never reach in a normal chess game (pawn can only move forward, not backward clearly, thus it can never get to the last row, Row 8).

To fix this problem, all pawns at Row 1 and Row 8 must be removed and in this example, two pawns at a1 and a8 must be removed so that this setup can be applied.

Note that this setup checking can check all requirements at the same time so that the client can adjust the setup accordingly once and for all. In the example below, every rule is violated at the same time.

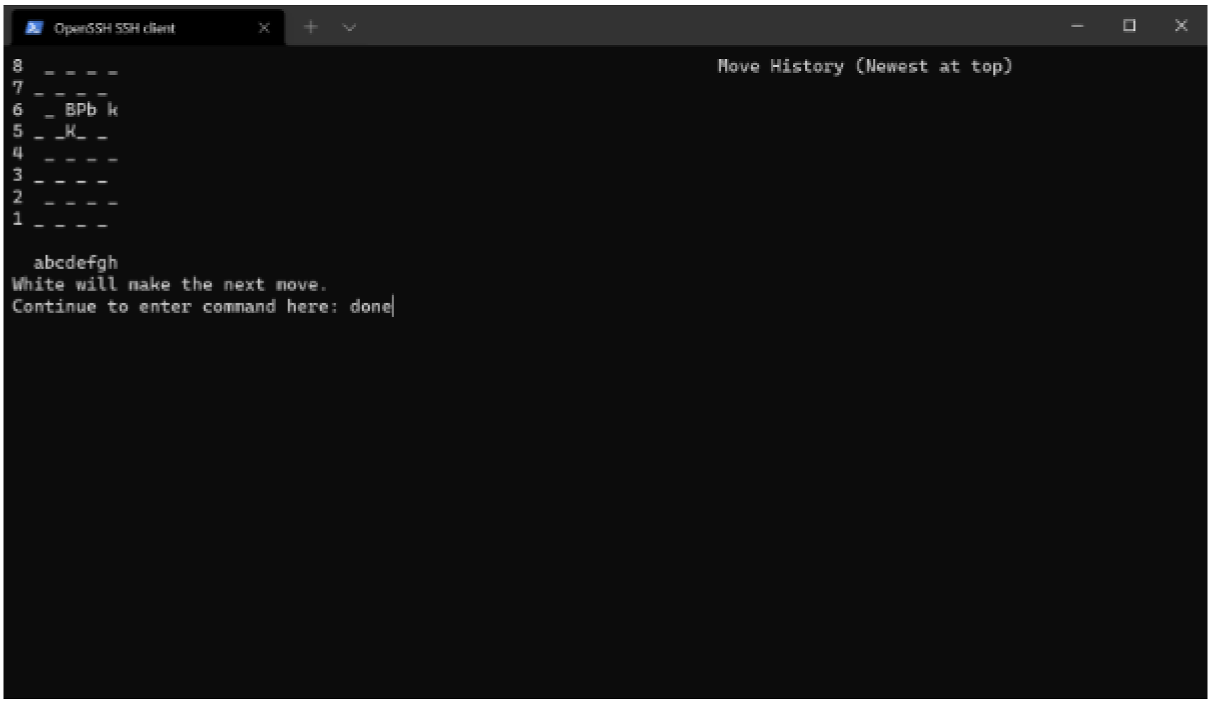


```
8 P_ _ _ _                                                        Move History (Newest at top)
7 _ _ _ _
6  _ _ _ _
5 _ _ _ _
4  _ _ _ _
3 _K_ _ _
2  _ _ _ _
1 n _ _ _

  abcdefgh
No black king detected.
Pawn detected at a8
No pawns can be placed at the first or last row of the board.
White king is being checked.
Invalid Setup. Please replace pieces accordingly.
Please enter command here:
```

There is no black king detected, which violates the first rule; there is a pawn at a8 which is the last row of the chessboard, which is violation of the second rule; there is a black knight at a1 that puts white king at b3 in check, which is violation of the third rule.

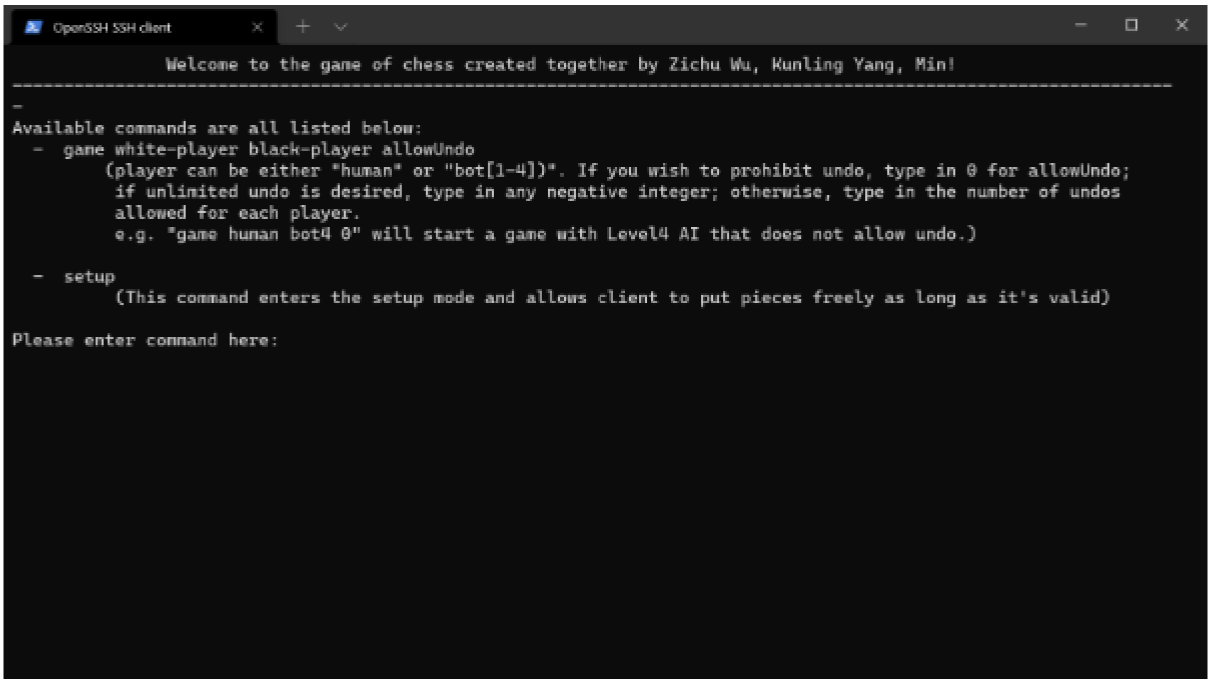
It is worthy to point out that when there are multiple kings of the same side detected, the program does not check if that side is in check, since having multiple kings already seems to be unreasonable and there is no good reason to check if any of them is being checked.

- Valid Setup



```
8  _ _ _ _                                                  Move History (Newest at top)
7 _ _ _ _
6  _ BPb k
5 _ _K_ _
4  _ _ _ _
3 _ _ _ _
2  _ _ _ _
1 _ _ _ _

  abcdefgh
White will make the next move.
Continue to enter command here: done
```

Above Is a very classical endgame in chess. There is exactly one white king, exactly one black king, and none of them is being checked, and there are no pawns at the first or last row (Row 1 and Row 8). Thus, it is considered as a valid setup and by typing in "*done*" and pressing [enter], the client is brought back to the main title page.

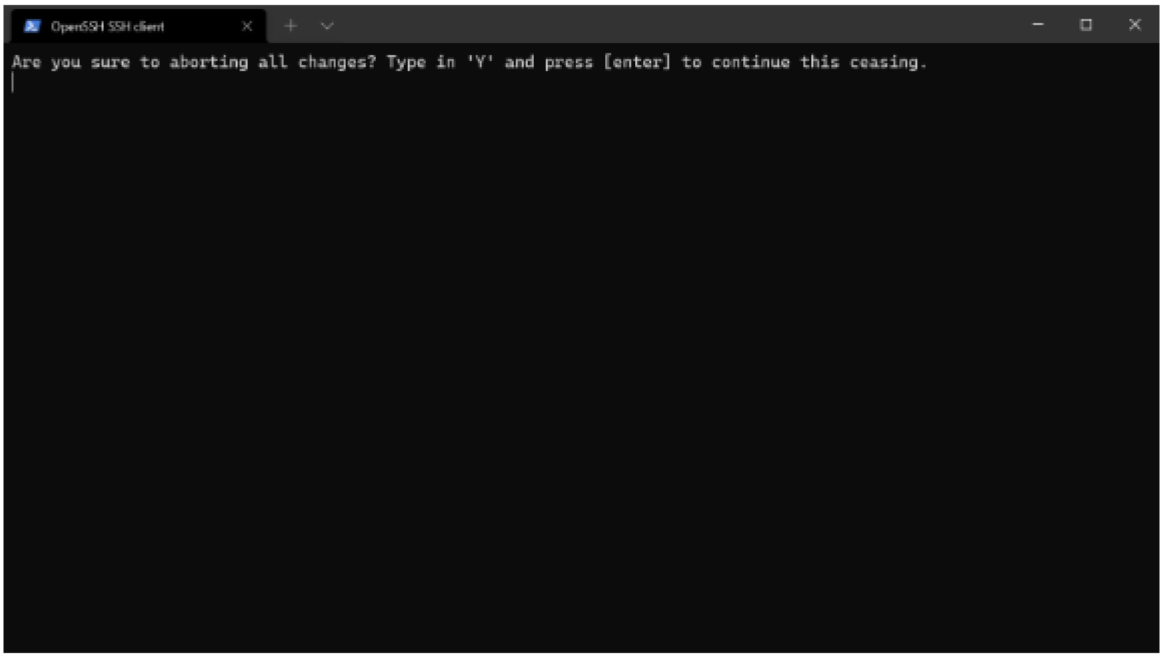


```
            Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
-----------------------------------------------------------------------------------------------------------
-
Available commands are all listed below:
  -  game white-player black-player allowUndo
        (player can be either "human" or "bot[1-4])". If you wish to prohibit undo, type in 0 for allowUndo;
         if unlimited undo is desired, type in any negative integer; otherwise, type in the number of undos
         allowed for each player.
         e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
         (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here:
```

By this time, this setup has already taken effect in all subsequent games, and the client can observe this setup once a new game is started.

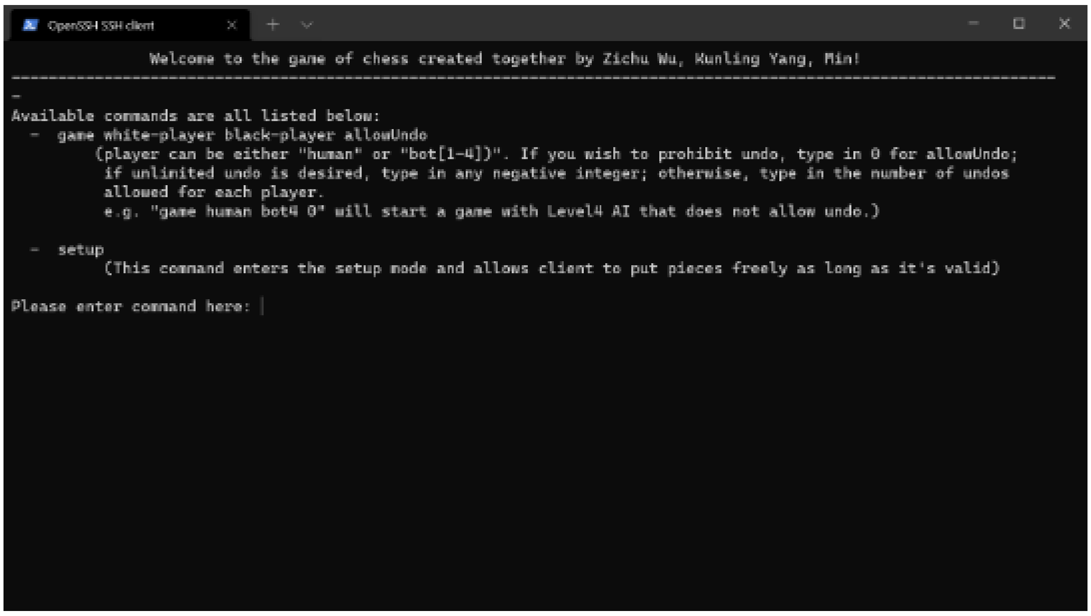
## Ceasing the Setup Mode

The command "*quit*" ceases all changes, restores all the pieces back to the default positions, and guides the client back to the main title page. But before that, a double confirmation is required. If the client types in "*quit*" and presses [enter], the previous content is refreshed with a new double confirmation printed, shown below:



```
Are you sure to aborting all changes? Type in 'Y' and press [enter] to continue this ceasing.
```

If the client has accidentally typed in "*quit*" and pressed the [enter] key, which is not very likely, the client can type in anything but a "*Y*" to get back to the setup mode and continue playing with pieces until satisfied. But once a single "*Y*" is typed and the client pressed the [enter] key, the board will be set back to the lastly saved chess board (if the client has already set up the board last time, otherwise the board will be set back to the default status).



```
              Welcome to the game of chess created together by Zichu Wu, Kunling Yang, Min!
--------------------------------------------------------------------------------------------------------------------
-
Available commands are all listed below:
  -  game white-player black-player allowUndo
         (player can be either "human" or "bot[1-4])". If you wish to prohibit undo, type in 0 for allowUndo;
          if unlimited undo is desired, type in any negative integer; otherwise, type in the number of undos
          allowed for each player.
          e.g. "game human bot4 0" will start a game with Level4 AI that does not allow undo.)

  -  setup
          (This command enters the setup mode and allows client to put pieces freely as long as it's valid)

Please enter command here:
```

Assume the client aborts the current setup, he or she is redirected to the main title page.

# Appendix

## Acceptable Piece Type

Currently this chess program supports six different types of chess pieces:

1. Pawn (represented by “p”)
2. Rook (represented by a “r”)
3. Knight (represented by a “k”)
4. Bishop (represented by a “b”)
5. Queen (represented by a “q”)
6. King (represented by a “k”)

Since all these pieces inherit from a Piece class, more pieces types can be added to the chess game very easily.