Chess Testing

⇒ for this phase we will go with ~~White Box~~ Black-Box Testing, and in Black-Box Testing we will check if the chess Game Performs all the Specified Requirements of the Chess Game.

I] Specifying the Requirements that will Be Tested:

- first we need to specify ~~the~~ the requirements That we are going to test, and they are as following:

{ Requirements }

1 - Identifying the User Turn
2 - Identifying all the possible moves a piece can make
3 - Making it not possible for a Chess piece to move if there is a Check and this move will not get rid of this check
4 - Identifying if a player won in case of Check mate
5 - Enabling the players to Restart the Game
6 - Identifying if its a Draw in Case of Stalemate
7 - Identifying if its a Draw in Case of Inssuficient Material
8 - Identifying if its a Draw in case of a mutual agreement between the 2-players

9 — Identifying if it is a Draw in Case 50 moves were made without the capturing of any pieces ~~Identify.~~

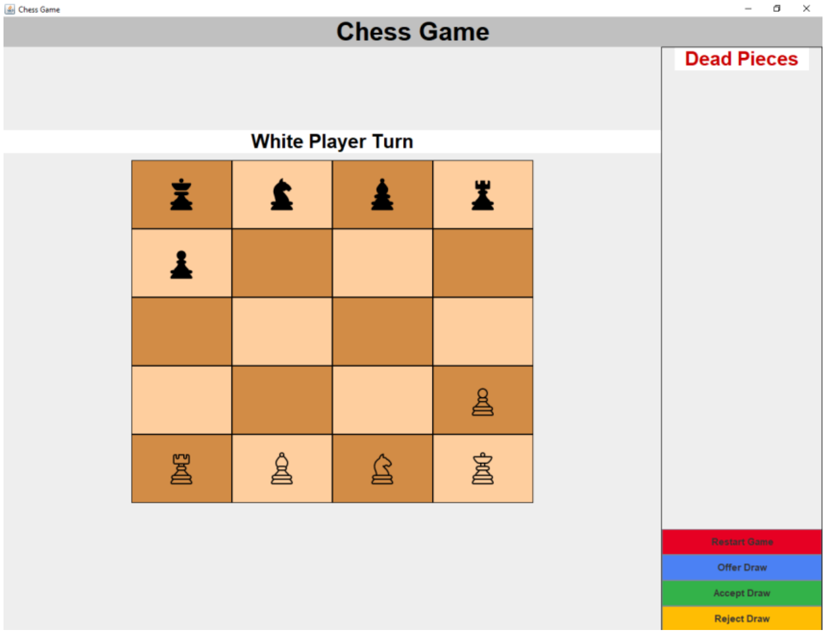
10 — Identifying the Chess Pieces ~~during the~~ ~~Game~~ that died during the Game

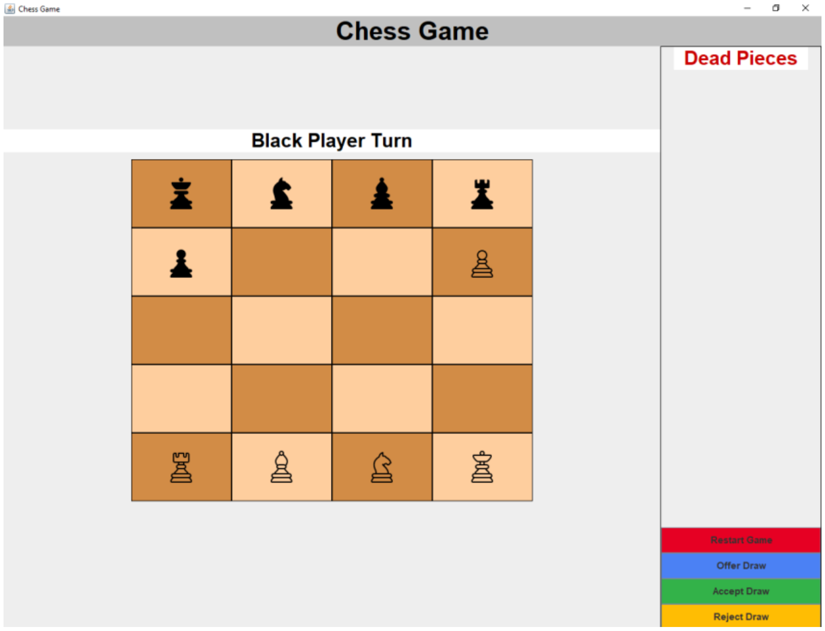
11 — Identify that the White Pawn & the Black Pawn both Can get Promoted

Test Execution

Here we will test that every Requirement that we have specified earlier is working correctly.

1) Identifying the User Turn:





2) Identifying all the possible moves a Piece can make

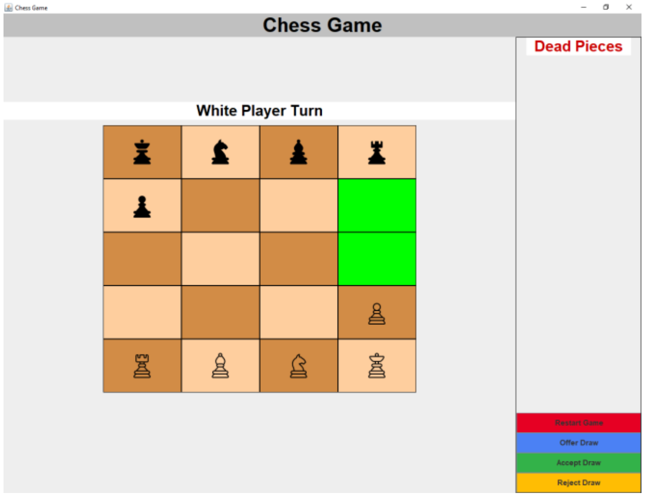


Figure 1: White Pawn Possible Moves

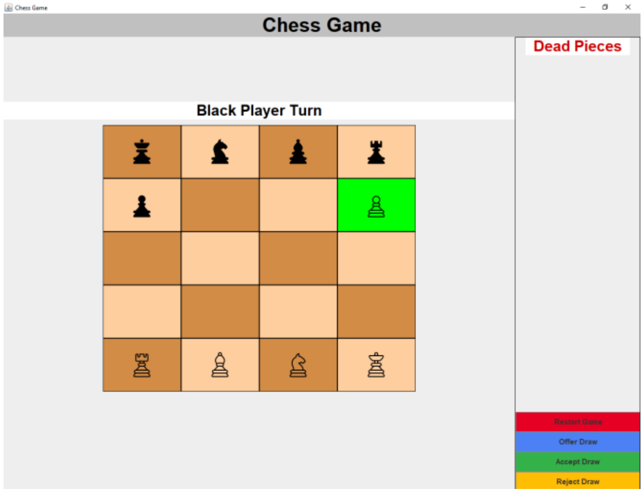


Figure 2: Black Tower Possible Moves

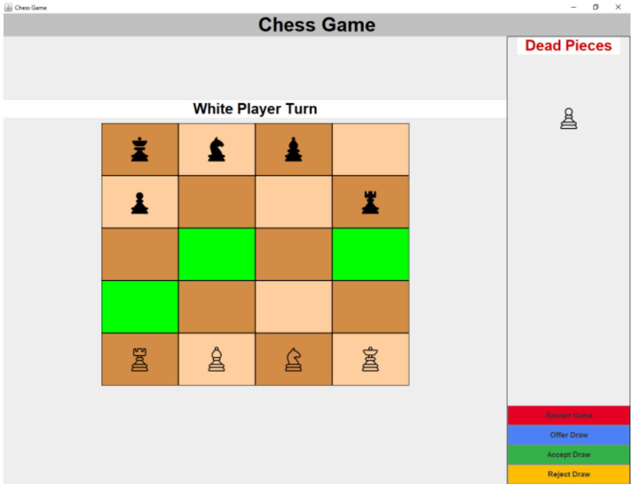


Figure 3: White Jumper Possible Moves

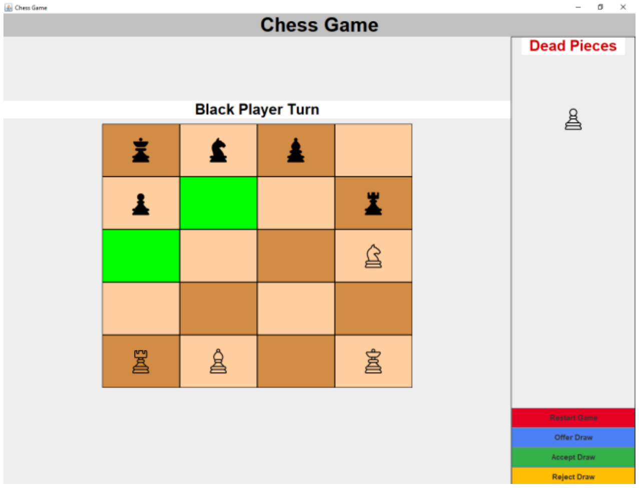


Figure 4: Black Runner Possible Moves

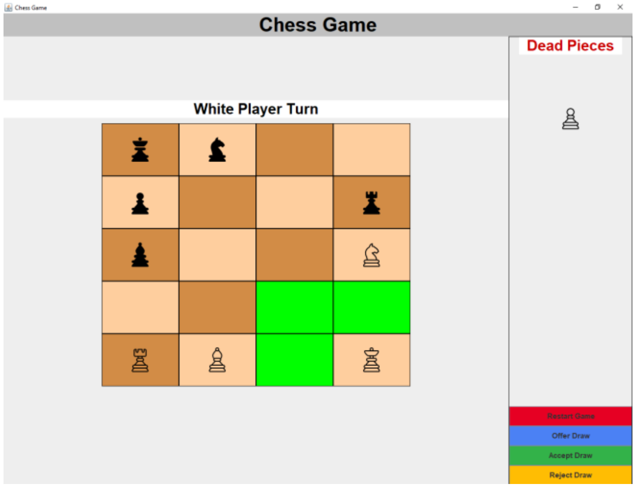
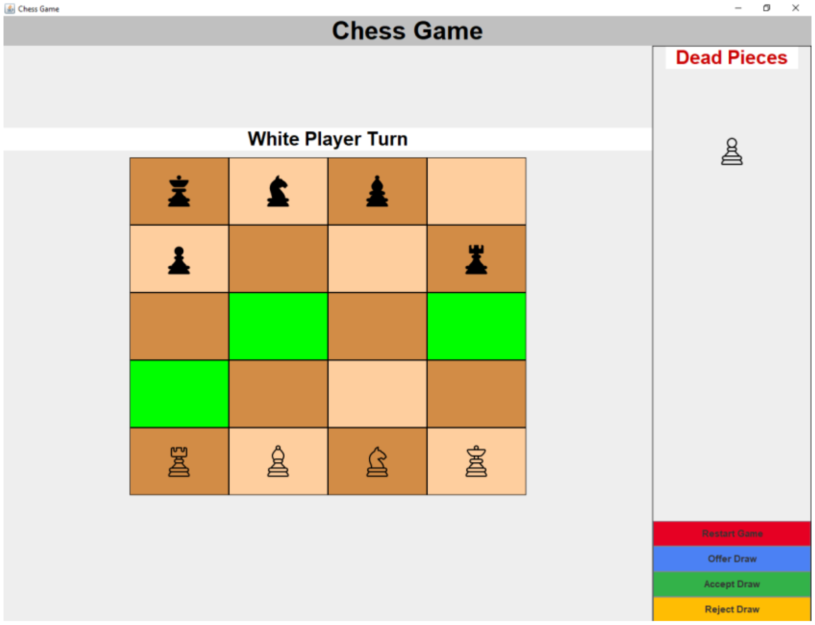


Figure 5: White King Possible Moves

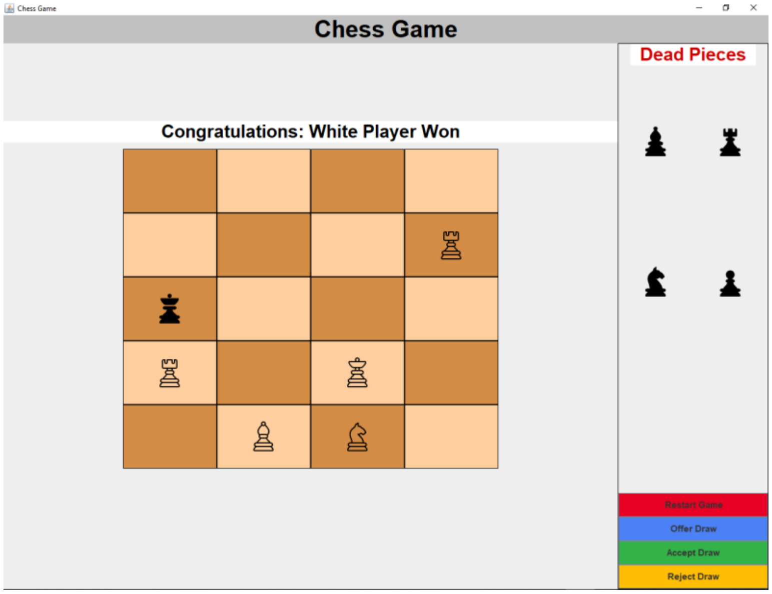
3) Making it not possible for a chess piece to move if there is a check and this move will not get rid of this check



White Player Turn

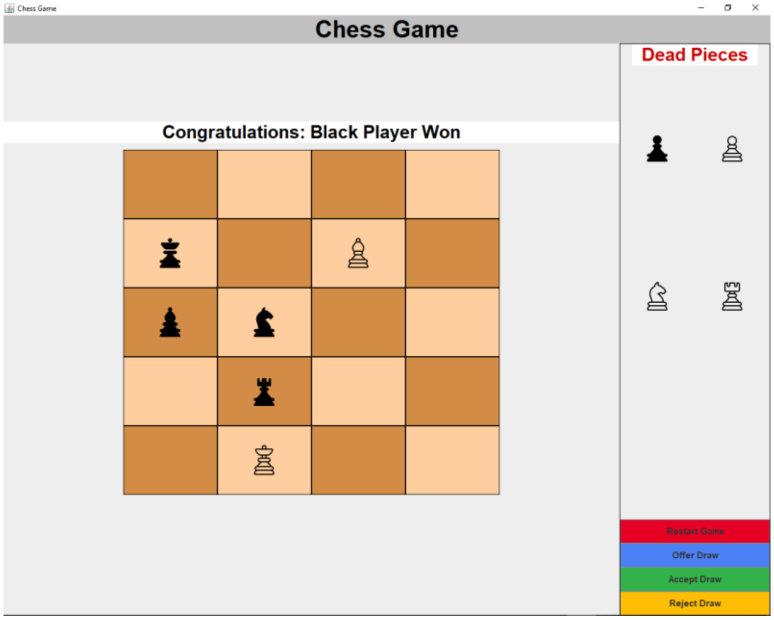
Here we are trying to move the White Jumper to the second row and first column this move cannot be done automatically by the program because the White King is in check by the black Tower.

4) Identifying if a player won in case of checkmate



Congratulations: White Player Won

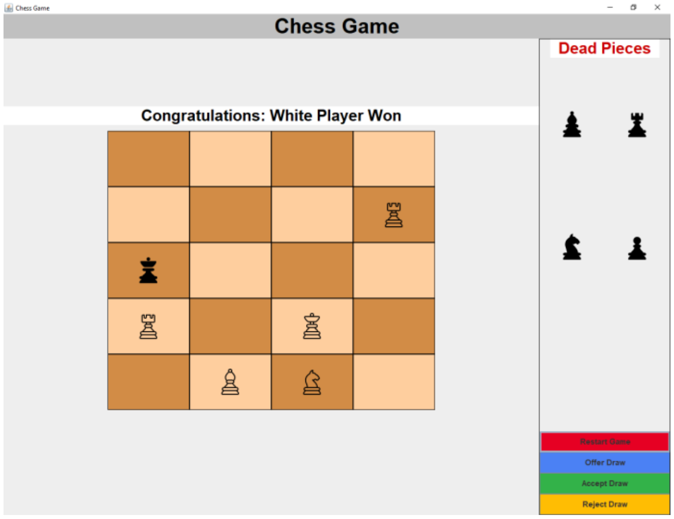
Here the Program states that the White Player has Won, because the Black King is in Checkmate because he is in a Check in his current spot and moving to any spot will also lead to a check.



Congratulations: Black Player Won

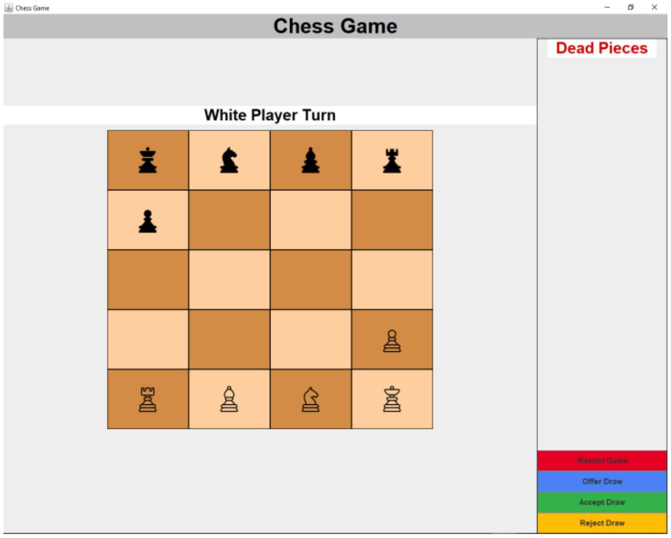
This is another Example where the Black Player has Won, because the White king is in Checkmate.

5) Enabling the players to Restart the game



Congratulations: White Player Won

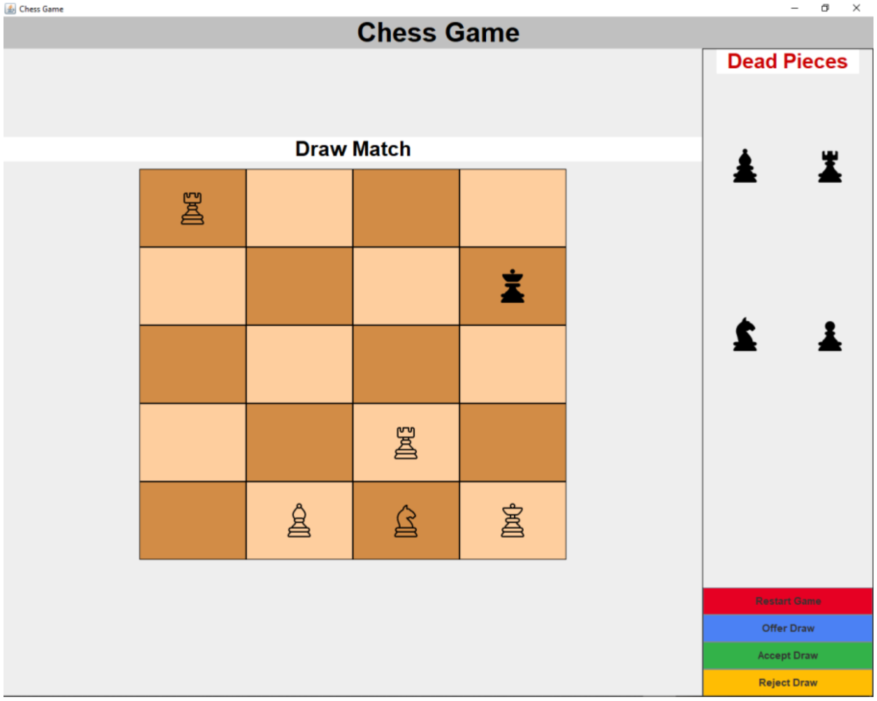
Figure 6: The Game Before Clicking the Restart Button



White Player Turn

Figure 7: The Game After Clicking the Restart Button

6) Identifying if it's a Draw in case of stalemate



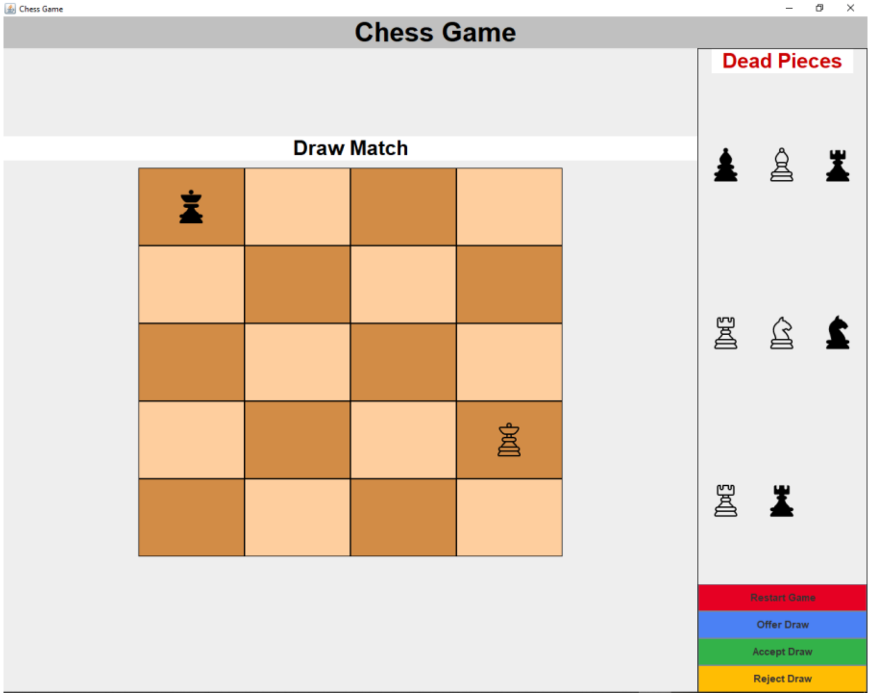
Draw Match

Here the Program states that it is a Draw Match, because the Black King is in Stalemate which means that it is the Black King turn and the current spot that he is currently on is not in check, but every other spot that he might move to will lead to a check.

7) Identifying if it's a Draw in case of Insufficient material

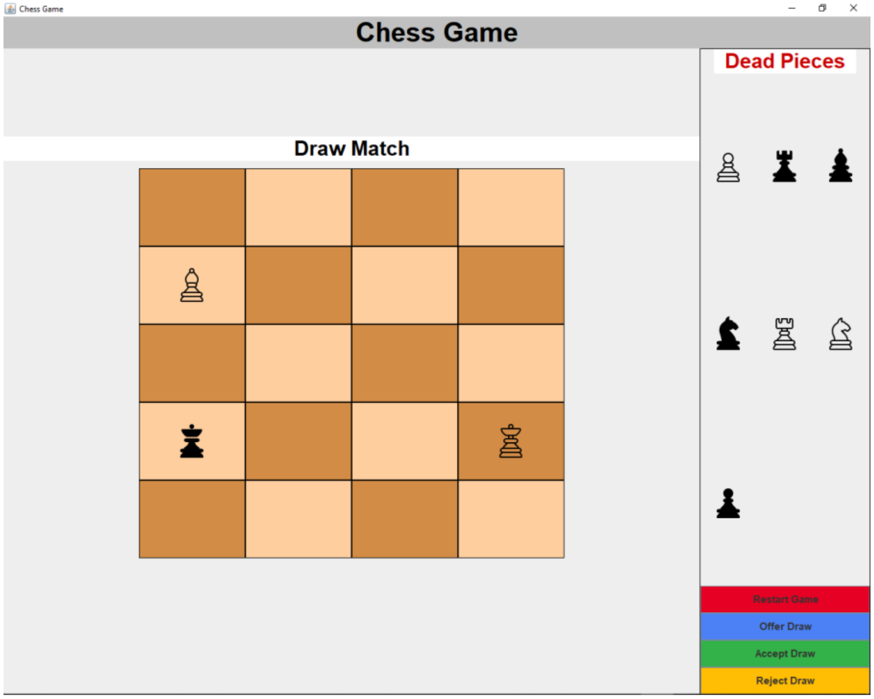
Insufficient Material in my program can happen in 2 cases:

a) If there is only 2 kings on the Board
b) If there is a King and a Runner VS. a King



Draw Match

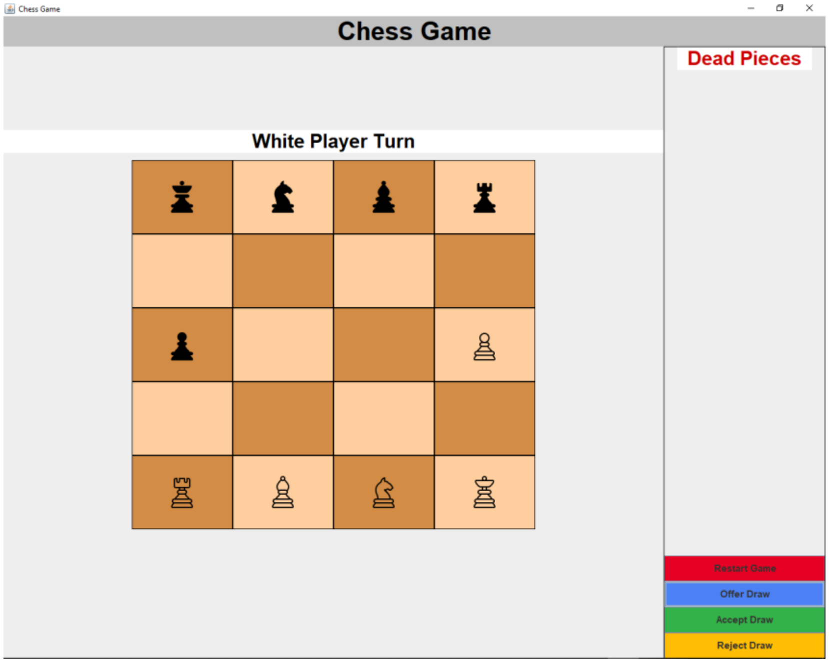
Figure 8: Only 2 Kings on the Board



Draw Match

Figure 9: a King and a Runner VS. a King

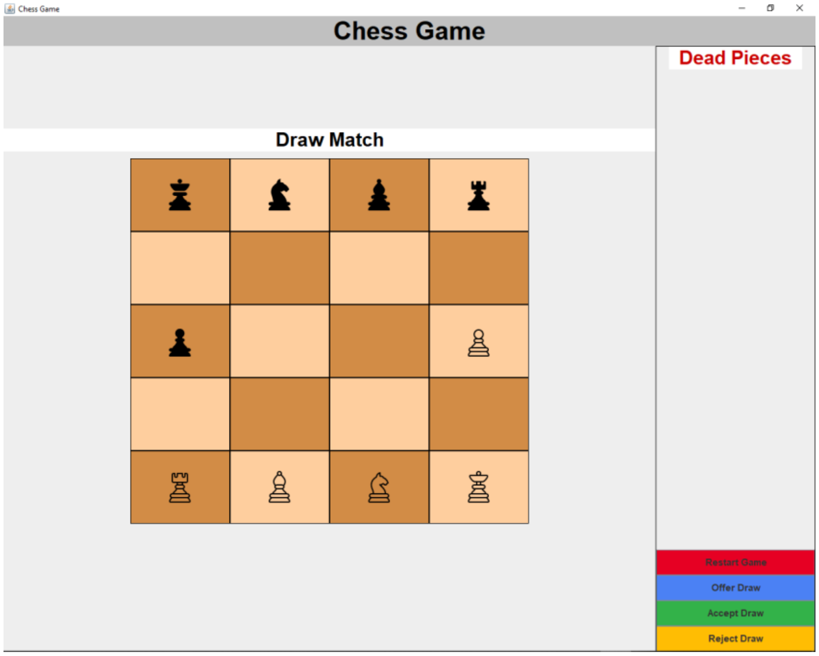
8) Identifying if it’s a Draw in case of mutual agreement between the 2-players


Chess Game

Dead Pieces

White Player Turn

Figure 10: Offer Draw Button was Clicked

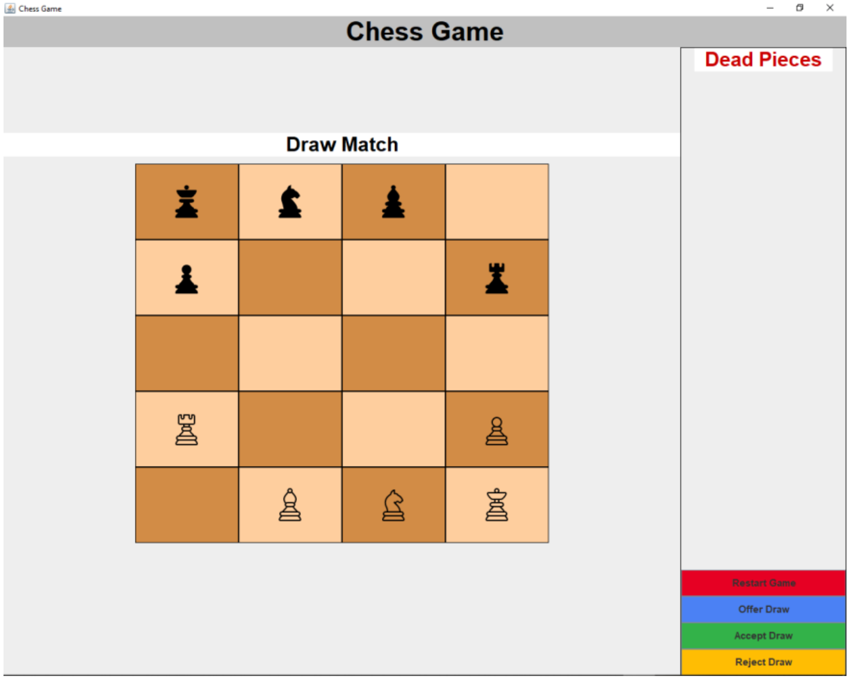

Chess Game

Dead Pieces

Draw Match

Figure 11: Accept Draw Button was Clicked

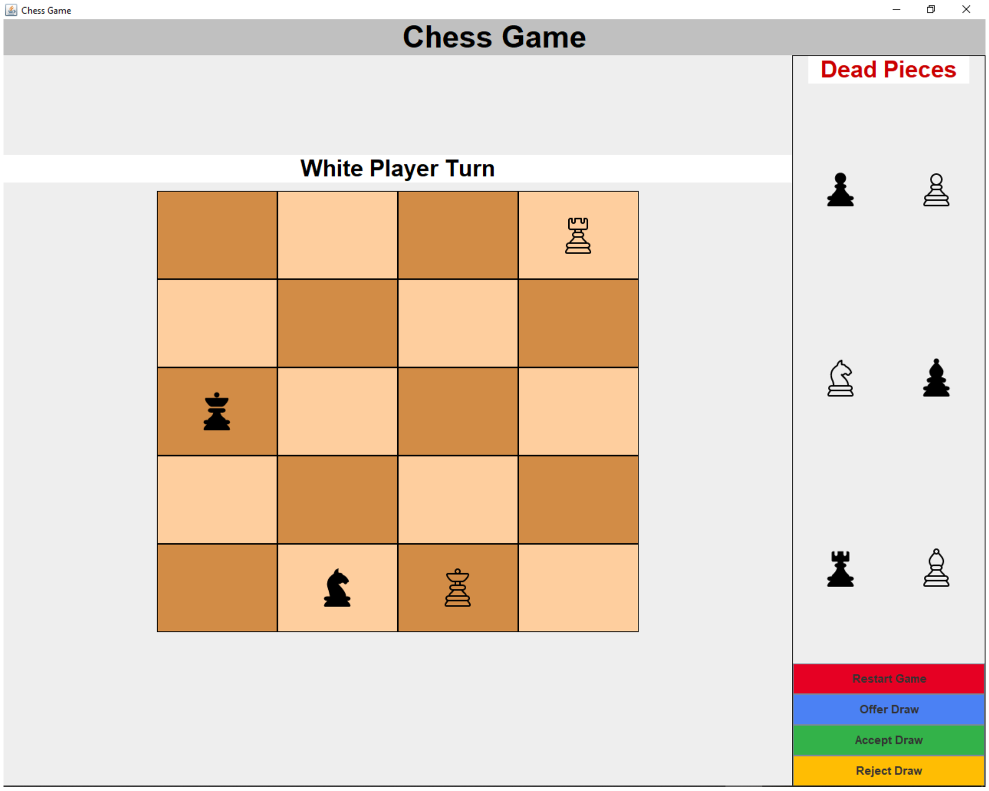
9) Identifying if it's a Draw in case 50 successive moves were made without any of the players capturing any of the chess pieces



Draw Match

The Program Stated that it is a Draw match after moving the Towers back and forth 50 times.

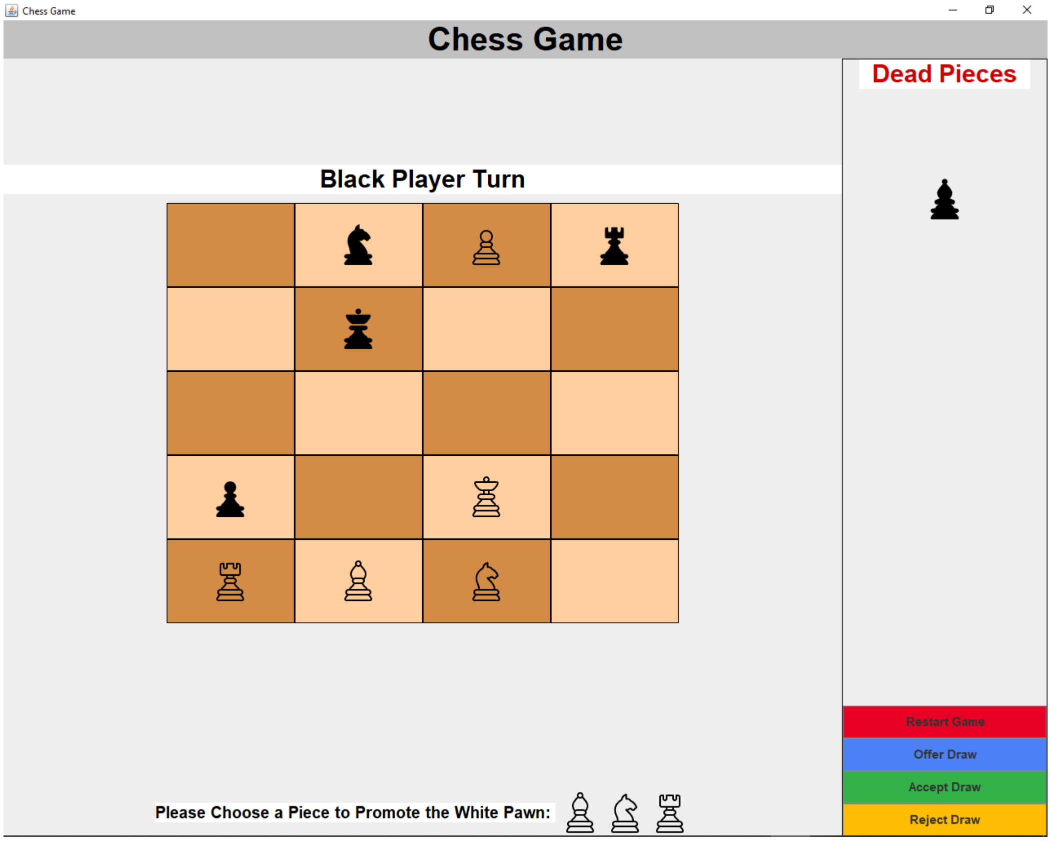
10) Identifying the Chess Pieces that died during the Game



White Player Turn

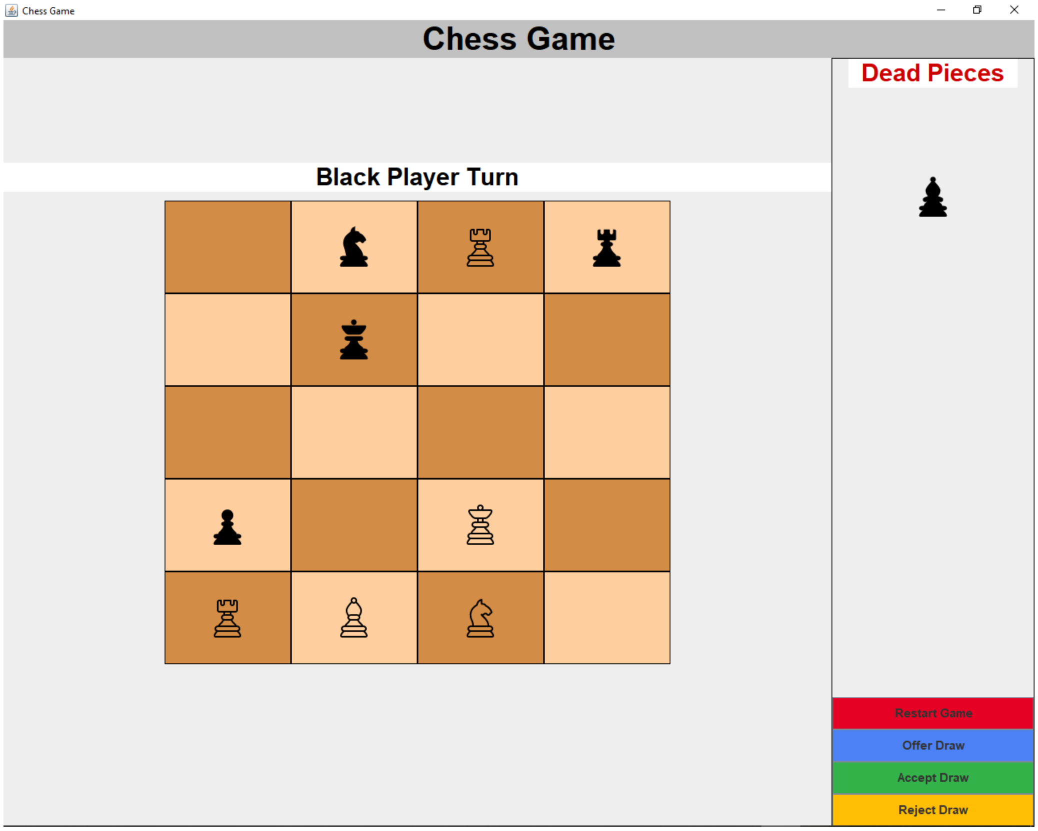
Here we are showing the Chess Pieces that have died during the Game in the Dead Pieces Panel on the Right

11) Identify that the White Pawn & the Black Pawn both can get promoted when reaching the end row from their perspective



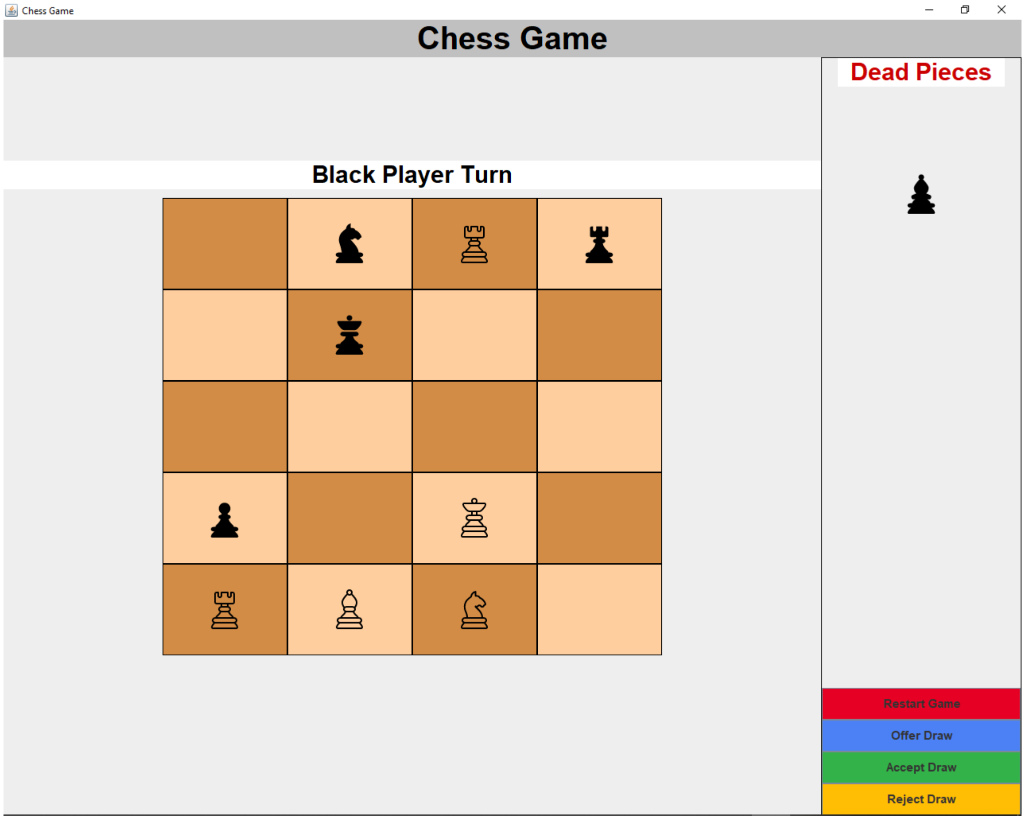
Please Choose a Piece to Promote the White Pawn:

Here we are promoting the White Pawn by giving the user the choice to Promote his Pawn to either a Tower, Runner or Jumper. Also, the User cannot click on any piece on the chess board until he promotes his Pawn to another Piece.



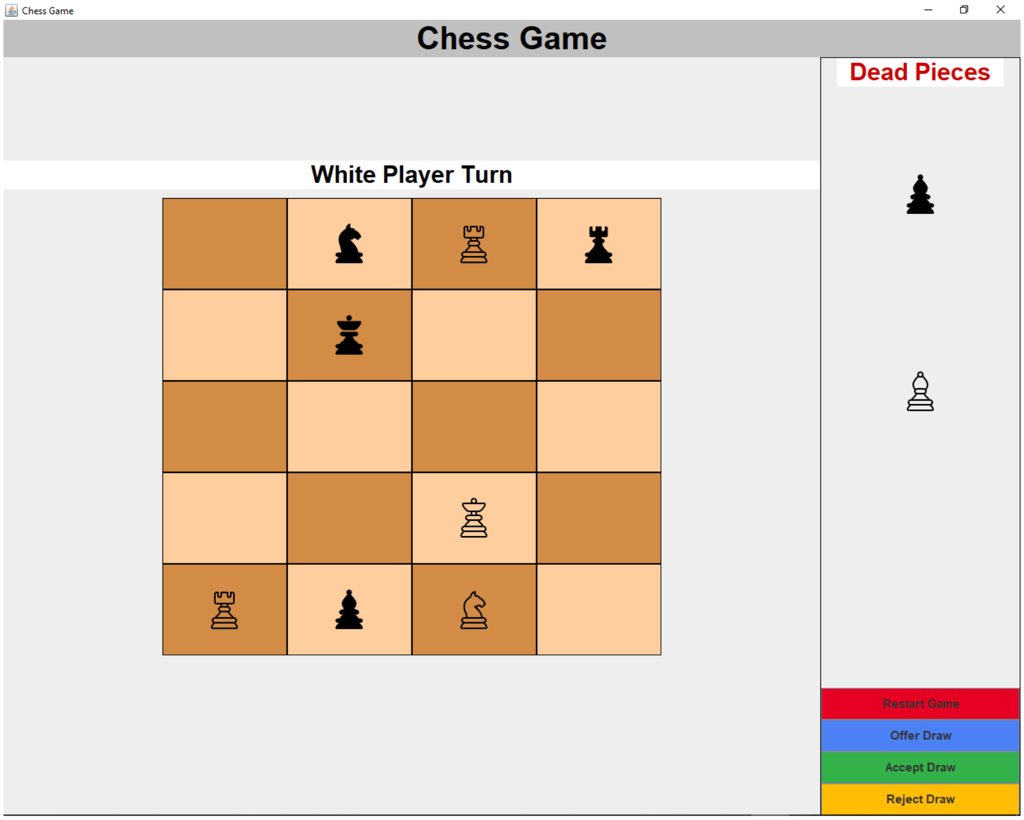
Black Player Turn

Here the White Player Promoted his Pawn to a Tower.



Black Player Turn

Here we are promoting the Black Pawn by giving the user the choice to Promote his Pawn to either a Tower, Runner or Jumper. Also, the User cannot click on any piece on the chess board until he promotes his Pawn to another Piece.



Here the White Player Promoted his Pawn to a Runner.

3]Report :

⇒ After Testing all ~~R~~ the Requirements that needs to be tested all of the test Checks have passed, but still we need to test this Program more by trying and using it as often as possible to try to detect weird behavior.

⇒ Also a behavior that ~~Shed~~ is not intendend to happen in the Game that when a pawn is getting Promoted the user is Supplied with a Panel to choose a chess piece to promote his pawn, and in that Case the user can offer a Draw and accept it while the Pawn is being Promoted, and the Solution for this is to either ~~Accept~~ State that it is a Draw match and hide the Pawn Promotion Panel, else ~~don't~~ don't allow the user to offer a draw while promoting ~~a~~ a Pawn.