```
import sys
from os import path
import tkinter as tk
from PIL import ImageTk, Image
from tkinter import ttk
import operator
# 1907, Bishop - 1640, Bishop and Queen - 1344, Rook - 1088, Rook and Queen - 802, 721, 711
# version1 completed 5/10/20

class Board(ttk.Frame):
    def __init__(self, parent, whitePawn, whiteRook, whiteKnight, whiteBishop, whiteKing, whiteQueen,
                 blackKing, blackQueen, blackBishop, blackKnight, blackRook, blackPawn):
        super().__init__(parent)
        self.buttonSize = 70
        self.borderColor = tk.StringVar()
        self.firstClick = False
        self.secondClick = True
        self.firstButton = None
        self.secondButton = None
        self.turn = "white"
        self.row1 = None
        self.row2 = None
        self.column1 = None
        self.column2 = None
        self.flip = False
        self.isClear = True
        self.doneLoop = False
        self.buttonRook = ""
        self.buttonRookTo = ""
        self.whitePawn = whitePawn
        self.whiteRook = whiteRook
        self.whiteKnight = whiteKnight
        self.whiteBishop = whiteBishop
        self.whiteKing = whiteKing
        self.whiteQueen = whiteQueen
        self.blackKing = blackKing

        self.blackQueen = blackQueen
        self.blackBishop = blackBishop
```

```
        self.blackKnight = blackKnight
        self.blackRook = blackRook
        self.blackPawn = blackPawn

        style = ttk.Style(self)
        style.theme_use("clam")

        style.configure("black.TButton", borderwidth=0, padding=0, background="#783c00")
        style.configure("tan.TButton", borderwidth=0, padding=0, background="#DEB887")

        print(style.element_options("Frame.border"))

        self.tagToIm = {"whitePawn": self.whitePawn, "whiteKnight": self.whiteKnight, "whiteRook": self.whiteRook,
                        "whiteBishop": self.whiteBishop, "whiteQueen": self.whiteQueen, "whiteKing": self.whiteKing,
                        "blackPawn": self.blackPawn, "blackRook": self.blackRook, "blackKnight": self.blackKnight,
                        "blackBishop": self.blackBishop, "blackQueen": self.blackQueen, "blackKing": self.blackKing
                        }
        self.board = [
            [],
            [],
            [],
            [],
            [],
            [],
            [],
            []
        ]
        self.buttonNumber = 0
        for row in self.board:
            for x in range(0, 8):
                if self.buttonNumber % 2 == 1:
                    color = "black.TButton"
                else:
                    color = "tan.TButton"
                if self.buttonNumber in range(8, 17):
                    im = self.blackPawn
                    id = "blackPawn"
                elif self.buttonNumber == 0 or self.buttonNumber == 7:
                    im = self.blackRook
                    id = "blackRook"
                elif self.buttonNumber == 1 or self.buttonNumber == 6:
                    im = self.blackKnight
```

```
                    id = "blackKnight"
                elif self.buttonNumber == 2 or self.buttonNumber == 5:
                    im = self.blackBishop
                    id = "blackBishop"
                elif self.buttonNumber == 3:
                    im = self.blackQueen
                    id = "blackQueen"
                elif self.buttonNumber == 4:
                    im = self.blackKing
                    id = "blackKing"
                elif self.buttonNumber in range(53, 62):
                    im = self.whitePawn
                    id = "whitePawn"
                elif self.buttonNumber == 63 or self.buttonNumber == 70:
                    im = self.whiteRook
                    id = "whiteRook"
                elif self.buttonNumber == 64 or self.buttonNumber == 69:
                    im = self.whiteKnight
                    id = "whiteKnight"
                elif self.buttonNumber == 65 or self.buttonNumber == 68:
                    im = self.whiteBishop
                    id = "whiteBishop"
                elif self.buttonNumber == 66:
                    im = self.whiteQueen
                    id = "whiteQueen"
                elif self.buttonNumber == 67:
                    im = self.whiteKing
                    id = "whiteKing"
                else:
                    im = ""
                    id = ""
                row.append({ttk.Button(self, image=im, style=color): [id, []]})
                self.buttonNumber += 1
            self.buttonNumber += 1

        self.rowconfigure((0, 1, 2, 3, 4, 5, 6, 7), minsize=self.buttonSize)
        self.columnconfigure((0, 1, 2, 3, 4, 5, 6, 7), minsize=self.buttonSize)

        self.row = 0
        for row in self.board:
            self.column = 0
            for buttons in row:
```

```
                for button, ids in buttons.items():
                    id, place = ids
                    place.append(self.row)
                    place.append(self.column)
                    button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
                    button.grid(row=self.row, column=self.column, sticky="NSEW", padx=0, pady=0)
                self.column += 1
            self.row += 1

        settingsFrame = ttk.Frame(self)
        settingsFrame.grid(row=8, column=0, columnspan=7)
        self.isFlip = tk.StringVar(value=True)
        self.isTurn = tk.StringVar(value="White's Move")
        FlipButton = ttk.Checkbutton(settingsFrame, text="Flip Board", variable=self.isFlip, onvalue=True, offvalue=False)
        FlipButton.grid(row=0, column=0, sticky="E")
        turnLabel = ttk.Label(settingsFrame, textvariable=self.isTurn)
        turnLabel.grid(row=0, column=1, padx=30)

    def clicked(self, event, row, column):
        button = event.widget
        for item in self.board:
            for items in item:
                for dict in items:
                    if button == dict:
                        self.tag1 = items[dict].copy()
        if button["image"] != "" and self.tag1[0][:5] == self.turn:
            self.firstButton = button
            self.row1 = row
            self.column1 = column
            self.firstClick = True
        elif self.firstClick == True:
            self.row2 = row
            self.column2 = column
            self.secondButton = button
            self.move()

    def move(self):
        global buttonRook, buttonRook
        for item in self.board:
            for items in item:
                for dict in items:
                    if self.firstButton == dict:
```

```
                    self.tag1 = items[dict].copy()
                elif self.secondButton == dict:
                    self.tag2 = items[dict].copy()

        if self.tag1[0][5:] == "Pawn":
            if self.tag1[0][:5] == "white":
                if self.secondButton["image"] == "":
                    if self.row1 - self.row2 == 1 and self.column1 - self.column2 == 0:
                        self.finishTurn()
                    elif self.tag1[1][0] == 6 and self.row1 - self.row2 == 2 and self.column1 - self.column2 == 0:
                        self.finishTurn()
                else:
                    if self.row1 - self.row2 == 1 and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "black":
                        self.finishTurn()
            elif self.tag1[0][:5] == "black":
                if self.secondButton["image"] == "":
                    if self.row2 - self.row1 == 1 and self.column2 - self.column1 == 0:
                        self.finishTurn()
                    elif self.tag1[1][0] == 1 and self.row2 - self.row1 == 2 and self.column1 - self.column2 == 0:
                        self.finishTurn()
                else:
                    if self.row2 - self.row1 == 1 and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "white":
                        self.finishTurn()

        elif self.tag1[0][5:] == "Knight":
            if self.tag1[0][:5] == "white":
                if self.secondButton["image"] == "":
                    if (self.row1 - self.row2 in (-2, 2) and self.column2 - self.column1 in (-1, 1)) or \
                            (self.row1 - self.row2 in (-1, 1) and self.column2 - self.column1 in (-2, 2)):
                        self.finishTurn()
                else:
                    if (self.row1 - self.row2 in (-1, 2) and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "black") or \
                            (self.row1 - self.row2 in (-1, 1) and self.column2 - self.column1 in (-2, 2) and self.tag2[0][:5] == "black"):
                        self.finishTurn()
            elif self.tag1[0][:5] == "black":
                if self.secondButton["image"] == "":
                    if (self.row2 - self.row1 in (-2, 2) and self.column1 - self.column2 in (-1, 1)) or \
                            (self.row2 - self.row1 in (-1, 1) and self.column1 - self.column2 in (-2, 2)):
                        self.finishTurn()
                else:
                    if (self.row2 - self.row1 in (-2, 2) and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "white") or \
                            (self.row2 - self.row1 in (-1, 1) and self.column1 - self.column2 in (-2, 2) and self.tag2[0][:5] == "white"):
```

```
                        self.finishTurn()

        elif self.tag1[0][5:] == "King":
            if self.tag1[0][:5] == "white":
                if self.secondButton["image"] == "":
                    if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1):
                        self.finishTurn()
                    elif self.row1 - self.row2 == 0 and self.row1 == 7:
                        if self.column2 - self.column1 == 2:
                            for item in self.board:
                                for items in item:
                                    for dict, key in items.items():
                                        if key[1][0] == self.row1 and key[1][1] == 5:
                                            if key[0] != "":
                                                self.isClear = False
                                                self.doneLoop = True
                                            break
                                        elif key[1][0] == self.row1 and key[1][1] == 6:
                                            if key[0] != "":
                                                self.isClear = False
                                                self.doneLoop = True
                                            break
                                        elif key[1][0] == self.row1 and key[1][1] == 7:
                                            if key[0] != "whiteRook":
                                                self.isClear = False
                                                self.doneLoop = True
                                            break
                            if self.isClear:
                                for item in self.board:
                                    for items in item:
                                        for button, key in items.items():
                                            if key[1][0] == self.row1 and key[1][1] == 7:
                                                self.buttonRook = button
                                            elif key[1][0] == self.row1 and key[1][1] == 5:
                                                self.buttonRookTo = button
                                self.buttonRook["image"] = ""
                                self.buttonRookTo["image"] = self.whiteRook
                                for item in self.board:
                                    for items in item:
                                        for dict in items:
                                            if self.buttonRook == dict:
                                                items[self.buttonRook][0] = ""
```

```
                                    elif self.buttonRookTo == dict:
                                        items[self.buttonRookTo][0] = "whiteRook"
                            self.finishTurn()
                    elif self.column2 - self.column1 == -2:
                        for item in self.board:
                            for items in item:
                                for dict, key in items.items():
                                    if key[1][0] == self.row1 and key[1][1] == 1:
                                        if key[0] != "":
                                            self.isClear = False
                                            self.doneLoop = True
                                        break
                                    elif key[1][0] == self.row1 and key[1][1] == 2:
                                        if key[0] != "":
                                            self.isClear = False
                                            self.doneLoop = True
                                        break
                                    elif key[1][0] == self.row1 and key[1][1] == 3:
                                        if key[0] != "":
                                            self.isClear = False
                                            self.doneLoop = True
                                        break
                                    elif key[1][0] == self.row1 and key[1][1] == 0:
                                        if key[0] != "whiteRook":
                                            self.isClear = False
                                            self.doneLoop = True
                                        break
                        if self.isClear:
                            for item in self.board:
                                for items in item:
                                    for button, key in items.items():
                                        if key[1][0] == self.row1 and key[1][1] == 0:
                                            self.buttonRook = button
                                        elif key[1][0] == self.row1 and key[1][1] == 3:
                                            self.buttonRookTo = button
                            self.buttonRook["image"] = ""
                            self.buttonRookTo["image"] = self.whiteRook
                            for item in self.board:
                                for items in item:
                                    for dict in items:
                                        if self.buttonRook == dict:
                                            items[self.buttonRook][0] = ""
```

```
                                        elif self.buttonRookTo == dict:
                                            items[self.buttonRookTo][0] = "whiteRook"
                                self.finishTurn()
                    else:
                        if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1) and self.tag2[0][:5] == "black":
                            self.finishTurn()
                elif self.tag1[0][:5] == "black":
                    if self.secondButton["image"] == "":
                        if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1):
                            self.finishTurn()
                        elif self.row1 - self.row2 == 0 and self.row1 == 0:
                            if self.column2 - self.column1 == 2:
                                for item in self.board:
                                    for items in item:
                                        for dict, key in items.items():
                                            if key[1][0] == self.row1 and key[1][1] == 5:
                                                if key[0] != "":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                            elif key[1][0] == self.row1 and key[1][1] == 6:
                                                if key[0] != "":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                            elif key[1][0] == self.row1 and key[1][1] == 7:
                                                if key[0] != "blackRook":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                if self.isClear:
                                    for item in self.board:
                                        for items in item:
                                            for button, key in items.items():
                                                if key[1][0] == self.row1 and key[1][1] == 7:
                                                    self.buttonRook = button
                                                elif key[1][0] == self.row1 and key[1][1] == 5:
                                                    self.buttonRookTo = button
                                    self.buttonRook["image"] = ""
                                    self.buttonRookTo["image"] = self.blackRook

                                    for item in self.board:
                                        for items in item:
```

```
                                    for dict in items:
                                        if self.buttonRook == dict:
                                            items[self.buttonRook][0] = ""
                                        elif self.buttonRookTo == dict:
                                            items[self.buttonRookTo][0] = "blackRook"
                                self.finishTurn()
                            elif self.column2 - self.column1 == -2:
                                for item in self.board:
                                    for items in item:
                                        for dict, key in items.items():
                                            if key[1][0] == self.row1 and key[1][1] == 1:
                                                if key[0] != "":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                            elif key[1][0] == self.row1 and key[1][1] == 2:
                                                if key[0] != "":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                            elif key[1][0] == self.row1 and key[1][1] == 3:
                                                if key[0] != "":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                            elif key[1][0] == self.row1 and key[1][1] == 0:
                                                if key[0] != "blackRook":
                                                    self.isClear = False
                                                    self.doneLoop = True
                                                break
                                if self.isClear:
                                    for item in self.board:
                                        for items in item:
                                            for button, key in items.items():
                                                if key[1][0] == self.row1 and key[1][1] == 0:
                                                    self.buttonRook = button
                                                elif key[1][0] == self.row1 and key[1][1] == 3:
                                                    self.buttonRookTo = button
                                    self.buttonRook["image"] = ""
                                    self.buttonRookTo["image"] = self.blackRook

                                    for item in self.board:
                                        for items in item:
```

```
                                    for dict in items:
                                        if self.buttonRook == dict:
                                            items[self.buttonRook][0] = ""
                                        elif self.buttonRookTo == dict:
                                            items[self.buttonRookTo][0] = "blackRook"
                                self.finishTurn()
                    else:
                        if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1) and self.tag2[0][:5] == "white":
                            self.finishTurn()

            elif self.tag1[0][5:] == "Bishop":
                if self.tag1[0][:5] == "white":
                    if self.secondButton["image"] == "":
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<")
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<")
                    else:
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "black":
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<", True)
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<", True)
                if self.tag1[0][:5] == "black":
                    if self.secondButton["image"] == "":
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<")
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
```

```
                            self.diagMove(">", "<")
                    else:
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "white":
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<", True)
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<", True)

            elif self.tag1[0][5:] == "Rook":
                if self.tag1[0][:5] == "white":
                    if self.secondButton["image"] == "":
                        if self.row2 - self.row1 == 0 and self.column2 < self.column1:
                            self.diagMove("==", "<", False, 1)
                        elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", False, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", False, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove("<", "==", False, 0)
                    else:
                        if self.tag2[0][:5] == "black":
                            if self.row2 - self.row1 == 0 and self.column2 < self.column1:
                                self.diagMove("==", "<", True, 1)
                            elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                                self.diagMove("==", ">", True, 1)
                            elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                                self.diagMove(">", "==", True, 0)
                            elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                                self.diagMove("<", "==", True, 0)
                if self.tag1[0][:5] == "black":
                    if self.secondButton["image"] == "":
                        if self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", False, 1)
                        elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
                            self.diagMove("==", "<", False, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", False, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
```

```
                        self.diagMove("<", "==", False, 0)
                else:
                    if self.tag2[0][:5] == "white":
                        if self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", True, 1)
                        elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
                            self.diagMove("==", "<", True, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", True, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove("<", "==", True, 0)

        elif self.tag1[0][5:] == "Queen":
            if self.tag1[0][:5] == "white":
                if self.secondButton["image"] == "":
                    if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<")
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<")
                    elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
                        self.diagMove("==", "<", False, 1)
                    elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                        self.diagMove("==", ">", False, 1)
                    elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                        self.diagMove(">", "==", False, 0)
                    elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                        self.diagMove("<", "==", False, 0)
                else:
                    if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "black":
                        if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                            self.diagMove("<", "<", True)
                        elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                            self.diagMove("<", ">", True)
                        elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                            self.diagMove(">", ">", True)
                        elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
```

```
                            self.diagMove(">", "<", True)
                    elif self.tag2[0][:5] == "black":
                        if self.row2 - self.row1 == 0 and self.column2 < self.column1:
                            self.diagMove("==", "<", True, 1)
                        elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", True, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", True, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove("<", "==", True, 0)
                if self.tag1[0][:5] == "black":
                    if self.secondButton["image"] == "":
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<")
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">")
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<")
                        elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
                            self.diagMove("==", "<", False, 1)
                        elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", False, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", False, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove("<", "==", False, 0)
                    else:
                        if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "white":
                            if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
                                self.diagMove("<", "<", True)
                            elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
                                self.diagMove("<", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
                                self.diagMove(">", ">", True)
                            elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
                                self.diagMove(">", "<", True)
                        elif self.tag2[0][:5] == "white":
                            if self.row2 - self.row1 == 0 and self.column2 < self.column1:
                                self.diagMove("==", "<", True, 1)
```

```
                        elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
                            self.diagMove("==", ">", True, 1)
                        elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove(">", "==", True, 0)
                        elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
                            self.diagMove("<", "==", True, 0)

        self.firstClick = False
        self.isClear = True
        self.doneLoop = False

    def diagMove(self, operator1, operator2, touch=False, isRook=-1):
        operators = {"<": operator.lt, ">": operator.gt, "==": operator.eq}
        if isRook == 1:
            self.increment1 = 0
            if operator2 == "<":
                self.increment2 = 1
            elif operator2 == ">":
                self.increment2 = -1
            secondOperator = operators[operator2](self.column2, self.column1)
        else:
            secondOperator = operators[operator2](self.column2 - self.column1, 0)
        if isRook == 0:
            self.increment2 = 0
            if operator1 == "<":
                self.increment1 = 1
            elif operator1 == ">":
                self.increment1 = -1
            firstOperator = operators[operator1](self.row2, self.row1)
        else:
            firstOperator = operators[operator1](self.row2 - self.row1, 0)
        if touch:
            if operator1 == "<" and isRook == -1:
                self.increment1 = 1
            elif operator1 == ">" and isRook == -1:
                self.increment1 = -1
            if operator2 == "<" and isRook == -1:
                self.increment2 = 1
            elif operator2 == ">" and isRook == -1:
                self.increment2 = -1

        else:
            self.increment1 = 0
```

```
            self.increment2 = 0
        if firstOperator and secondOperator:
            while not self.doneLoop:
                if self.tag2[1][0] + self.increment1 == self.row1 and self.tag2[1][1] + self.increment2 == self.column1:
                    break
                if isRook == -1:
                    if operator1 == "<":
                        self.row1 -= 1
                    elif operator1 == ">":
                        self.row1 += 1
                    if operator2 == "<":
                        self.column1 -= 1
                    elif operator2 == ">":
                        self.column1 += 1
                elif isRook == 0:
                    if operator1 == "<":
                        self.row1 -= 1
                    elif operator1 == ">":
                        self.row1 += 1
                elif isRook == 1:
                    if operator2 == "<":
                        self.column1 -= 1
                    elif operator2 == ">":
                        self.column1 += 1
                for item in self.board:
                    for items in item:
                        for dict, key in items.items():
                            if key[1][0] == self.row1 and key[1][1] == self.column1:
                                if key[0] != "":
                                    self.isClear = False
                                    self.doneLoop = True
                                break
            if self.isClear:
                self.finishTurn()

    def finishTurn(self):
        if self.tag2[0] == "blackKing":
            gameOver("White")
            self.switchTag()
        elif self.tag2[0] == "whiteKing":
            gameOver("Black")
            self.switchTag()
```

```
627             elif self.tag1[0] == "whitePawn" and self.tag2[1][0] == 0:
628                 endRow(self.turn)
629             elif self.tag1[0] == "blackPawn" and self.tag2[1][0] == 7:
630                 endRow(self.turn)
631             else:
632                 if self.isFlip.get() == "1":
633                     self.flipBoard()
634                 else:
635                     if self.turn == "white":
636                         self.turn = "black"
637                         self.isTurn.set("Black's Move")
638                     elif self.turn == "black":
639                         self.turn = "white"
640                         self.isTurn.set("White's Move")
641                 self.switchTag()
642                 self.firstClick = False
643 
644     def switchTag(self):
645         self.firstButton["image"] = ""
646         self.secondButton["image"] = self.tagToIm[self.tag1[0]]
647         for item in self.board:
648             for items in item:
649                 for dict in items:
650                     if self.secondButton == dict:
651                         items[self.secondButton][0] = self.tag1[0]
652                     elif self.firstButton == dict:
653                         items[self.firstButton][0] = ""
654 
655     def flipBoard(self):
656         if not self.flip:
657             self.flip = True
658             self.after(500, self.flipBoard)
659         elif self.turn == "white":
660             self.row = 0
661             for row in self.board:
662                 self.column = 0
663                 for buttons in row:
664                     for button, ids in buttons.items():
665                         id, place = ids
666                         place = []
667                         place.append(self.row)
668                         place.append(self.column)
```

```
                        button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
                        button.grid(row=7 - self.row, column=7 - self.column, sticky="NSEW", padx=0, pady=0)
                    self.column += 1
                self.row += 1
                self.turn = "black"
                self.isTurn.set("Black's Move")
            self.flip = False
        else:
            self.row = 0
            for row in self.board:
                self.column = 0
                for buttons in row:
                    for button, ids in buttons.items():
                        id, place = ids
                        place = [self.row, self.column]
                        button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
                        button.grid(row=self.row, column=self.column, sticky="NSEW", padx=0, pady=0)
                    self.column += 1
                self.row += 1
                self.turn = "white"
                self.isTurn.set("White's Move")
            self.flip = False

    def changePiece(self, piece):
        board.tkraise()
        self.tag1[0] = piece
        self.finishTurn()

root = tk.Tk()

# All of the following images were croped from: https://pixabay.com/illustrations/chess-black-and-white-pieces-3413429/
bundle_dir = getattr(sys, "_MEIPASS", path.abspath(path.dirname(__file__)))
pathToWhitePawn = path.join(bundle_dir, "assets", "whitePawn.png")
whitePawnImage = Image.open(pathToWhitePawn)
whitePawnImage = whitePawnImage.resize((40, 60))
whitePawn = ImageTk.PhotoImage(whitePawnImage)

pathToBlackPawn = path.join(bundle_dir, "assets", "blackPawn.png")
blackPawnImage = Image.open(pathToBlackPawn)
blackPawnImage = blackPawnImage.resize((40, 60))
blackPawn = ImageTk.PhotoImage(blackPawnImage)

```

```
711 pathToBlackRook = path.join(bundle_dir, "assets", "blackRook.png")
712 blackRookImage = Image.open(pathToBlackRook)
713 blackRookImage = blackRookImage.resize((40, 60))
714 blackRook = ImageTk.PhotoImage(blackRookImage)
715
716 pathToBlackKnight = path.join(bundle_dir, "assets", "blackKnight.png")
717 blackKnightImage = Image.open(pathToBlackKnight)
718 blackKnightImage = blackKnightImage.resize((40, 60))
719 blackKnight = ImageTk.PhotoImage(blackKnightImage)
720
721 pathToBlackBishop = path.join(bundle_dir, "assets", "blackBishop.png")
722 blackBishopImage = Image.open(pathToBlackBishop)
723 blackBishopImage = blackBishopImage.resize((40, 60))
724 blackBishop = ImageTk.PhotoImage(blackBishopImage)
725
726 pathToBlackKing = path.join(bundle_dir, "assets", "blackKing.png")
727 blackKingImage = Image.open(pathToBlackKing)
728 blackKingImage = blackKingImage.resize((40, 60))
729 blackKing = ImageTk.PhotoImage(blackKingImage)
730
731 pathToBlackQueen = path.join(bundle_dir, "assets", "blackQueen.png")
732 blackQueenImage = Image.open(pathToBlackQueen)
733 blackQueenImage = blackQueenImage.resize((40, 60))
734 blackQueen = ImageTk.PhotoImage(blackQueenImage)
735
736 pathToWhitePawn = path.join(bundle_dir, "assets", "whitePawn.png")
737 whitePawnImage = Image.open(pathToWhitePawn)
738 whitePawnImage = whitePawnImage.resize((40, 60))
739 whitePawn = ImageTk.PhotoImage(whitePawnImage)
740
741 pathToWhiteRook = path.join(bundle_dir, "assets", "whiteRook.png")
742 whiteRookImage = Image.open(pathToWhiteRook)
743 whiteRookImage = whiteRookImage.resize((40, 60))
744 whiteRook = ImageTk.PhotoImage(whiteRookImage)
745
746 pathToWhiteKnight = path.join(bundle_dir, "assets", "whiteKnight.png")
747 whiteKnightImage = Image.open(pathToWhiteKnight)
748 whiteKnightImage = whiteKnightImage.resize((40, 60))
749 whiteKnight = ImageTk.PhotoImage(whiteKnightImage)
750
751 pathToWhiteBishop = path.join(bundle_dir, "assets", "whiteBishop.png")
752 whiteBishopImage = Image.open(pathToWhiteBishop)
```

```
whiteBishopImage = whiteBishopImage.resize((40, 60))
whiteBishop = ImageTk.PhotoImage(whiteBishopImage)

pathToWhiteKing = path.join(bundle_dir, "assets", "whiteKing.png")
whiteKingImage = Image.open(pathToWhiteKing)
whiteKingImage = whiteKingImage.resize((40, 60))
whiteKing = ImageTk.PhotoImage(whiteKingImage)

pathToWhiteQueen = path.join(bundle_dir, "assets", "whiteQueen.png")
whiteQueenImage = Image.open(pathToWhiteQueen)
whiteQueenImage = whiteQueenImage.resize((40, 60))
whiteQueen = ImageTk.PhotoImage(whiteQueenImage)

def endRow(turn):
    if turn == "white":
        frameWhite.tkraise()
    else:
        frameBlack.tkraise()

def gameOver(winner):
    gameOverFrame = ttk.Frame(root, width=560, height=560, padding=5)
    gameOverFrame.grid(row=0, column=0)
    style = ttk.Style(frameWhite)
    style.theme_use("clam")
    gameOverLabel = ttk.Label(gameOverFrame, text="Game Over", font=("Courier", 30))
    winnerLabel = ttk.Label(gameOverFrame, text=f"{winner} Wins!", font=("Courier", 30))
    gameOverLabel.grid()
    winnerLabel.grid()

board = Board(root, whitePawn, whiteRook, whiteKnight, whiteBishop, whiteKing, whiteQueen,
                blackKing, blackQueen, blackBishop, blackKnight, blackRook, blackPawn)
board.grid(row=0, column=0)

frameWhite = ttk.Frame(root, width=560, height=560, padding=5)
frameWhite.grid(row=0, column=0)
style = ttk.Style(frameWhite)

whiteLabel = ttk.Label(frameWhite, text="Choose a piece", font=("Courier", 25), padding=10)
whiteRookFrame = ttk.Button(frameWhite, im=whiteRook, padding=30, command=lambda: board.changePiece("whiteRook"))
whiteKnightFrame = ttk.Button(frameWhite, im=whiteKnight, padding=30, command=lambda: board.changePiece("whiteKnight"))
whiteBishopFrame = ttk.Button(frameWhite, im=whiteBishop, padding=30, command=lambda: board.changePiece("whiteBishop"))
whiteQueenFrame = ttk.Button(frameWhite, im=whiteQueen, padding=30, command=lambda: board.changePiece("whiteQueen"))
```

```
795
796 whiteLabel.grid(row=0, column=0, columnspan=2)
797 whiteKnightFrame.grid(row=1, column=0, pady=10)
798 whiteBishopFrame.grid(row=1, column=1, pady=10)
799 whiteRookFrame.grid(row=2, column=0, pady=10)
800 whiteQueenFrame.grid(row=2, column=1, pady=10)
801
802 frameBlack = ttk.Frame(root, width=560, height=560, padding=5)
803 frameBlack.grid(row=0, column=0)
804
805 blackLabel = ttk.Label(frameBlack, text="Choose a piece", font=("Courier", 25), padding=10)
806 blackRookFrame = ttk.Button(frameBlack, im=blackRook, padding=30, command=lambda: board.changePiece("blackRook"))
807 blackKnightFrame = ttk.Button(frameBlack, im=blackKnight, padding=30, command=lambda: board.changePiece("blackKnight"))
808 blackBishopFrame = ttk.Button(frameBlack, im=blackBishop, padding=30, command=lambda: board.changePiece("blackBishop"))
809 blackQueenFrame = ttk.Button(frameBlack, im=blackQueen, padding=30, command=lambda: board.changePiece("blackQueen"))
810
811 blackLabel.grid(row=0, column=0, columnspan=2)
812 blackKnightFrame.grid(row=1, column=0, pady=10)
813 blackBishopFrame.grid(row=1, column=1, pady=10)
814 blackRookFrame.grid(row=2, column=0, pady=10)
815 blackQueenFrame.grid(row=2, column=1, pady=10)
816
817 board.tkraise()
818
819 root.mainloop()
```

---

*PDF* document made with CodePrint using Prism