

```
1 import sys
2 from os import path
3 import tkinter as tk
4 from PIL import ImageTk, Image
5 from tkinter import ttk
6 import operator
7 # 1907, Bishop - 1640, Bishop and Queen - 1344, Rook - 1088, Rook and Queen - 802, 721, 711
8 # version1 completed 5/10/20
9
10 class Board(ttk.Frame):
11     def __init__(self, parent, whitePawn, whiteRook, whiteKnight, whiteBishop, whiteKing, whiteQueen,
12                 blackKing, blackQueen, blackBishop, blackKnight, blackRook, blackPawn):
13         super().__init__(parent)
14         self.buttonSize = 70
15         self.borderColor = tk.StringVar()
16         self.firstClick = False
17         self.secondClick = True
18         self.firstButton = None
19         self.secondButton = None
20         self.turn = "white"
21         self.row1 = None
22         self.row2 = None
23         self.column1 = None
24         self.column2 = None
25         self.flip = False
26         self.isClear = True
27         self.doneLoop = False
28         self.buttonRook = ""
29         self.buttonRookTo = ""
30         self.whitePawn = whitePawn
31         self.whiteRook = whiteRook
32         self.whiteKnight = whiteKnight
33         self.whiteBishop = whiteBishop
34         self.whiteKing = whiteKing
35         self.whiteQueen = whiteQueen
36         self.blackKing = blackKing
37
38         self.blackQueen = blackQueen
39         self.blackBishop = blackBishop
```

```
39 self.blackKnight = blackKnight
40 self.blackRook = blackRook
41 self.blackPawn = blackPawn
42
43 style = ttk.Style(self)
44 style.theme_use("clam")
45
46 style.configure("black.TButton", borderwidth=0, padding=0, background="#783c00")
47 style.configure("tan.TButton", borderwidth=0, padding=0, background="#DEB887")
48
49 print(style.element_options("Frame.border"))
50
51 self.tagToIm = {"whitePawn": self.whitePawn, "whiteKnight": self.whiteKnight, "whiteRook": self.whiteRook,
52               "whiteBishop": self.whiteBishop, "whiteQueen": self.whiteQueen, "whiteKing": self.whiteKing,
53               "blackPawn": self.blackPawn, "blackRook": self.blackRook, "blackKnight": self.blackKnight,
54               "blackBishop": self.blackBishop, "blackQueen": self.blackQueen, "blackKing": self.blackKing
55               }
56 self.board = [
57     [],
58     [],
59     [],
60     [],
61     [],
62     [],
63     [],
64     []
65 ]
66 self.buttonNumber = 0
67 for row in self.board:
68     for x in range(0, 8):
69         if self.buttonNumber % 2 == 1:
70             color = "black.TButton"
71         else:
72             color = "tan.TButton"
73         if self.buttonNumber in range(8, 17):
74             im = self.blackPawn
75             id = "blackPawn"
76         elif self.buttonNumber == 0 or self.buttonNumber == 7:
77             im = self.blackRook
78             id = "blackRook"
79         elif self.buttonNumber == 1 or self.buttonNumber == 6:
80             im = self.blackKnight
```

```
81         id = "blackKnight"
82     elif self.buttonNumber == 2 or self.buttonNumber == 5:
83         im = self.blackBishop
84         id = "blackBishop"
85     elif self.buttonNumber == 3:
86         im = self.blackQueen
87         id = "blackQueen"
88     elif self.buttonNumber == 4:
89         im = self.blackKing
90         id = "blackKing"
91     elif self.buttonNumber in range(53, 62):
92         im = self.whitePawn
93         id = "whitePawn"
94     elif self.buttonNumber == 63 or self.buttonNumber == 70:
95         im = self.whiteRook
96         id = "whiteRook"
97     elif self.buttonNumber == 64 or self.buttonNumber == 69:
98         im = self.whiteKnight
99         id = "whiteKnight"
100    elif self.buttonNumber == 65 or self.buttonNumber == 68:
101        im = self.whiteBishop
102        id = "whiteBishop"
103    elif self.buttonNumber == 66:
104        im = self.whiteQueen
105        id = "whiteQueen"
106    elif self.buttonNumber == 67:
107        im = self.whiteKing
108        id = "whiteKing"
109    else:
110        im = ""
111        id = ""
112    row.append({ttk.Button(self, image=im, style=color): [id, []]})
113    self.buttonNumber += 1
114    self.buttonNumber += 1
115
116    self.rowconfigure((0, 1, 2, 3, 4, 5, 6, 7), minsize=self.buttonSize)
117    self.columnconfigure((0, 1, 2, 3, 4, 5, 6, 7), minsize=self.buttonSize)
118
119    self.row = 0
120    for row in self.board:
121        self.column = 0
122        for buttons in row:
```

```
123         for button, ids in buttons.items():
124             id, place = ids
125             place.append(self.row)
126             place.append(self.column)
127             button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
128             button.grid(row=self.row, column=self.column, sticky="NSEW", padx=0, pady=0)
129         self.column += 1
130     self.row += 1
131
132     settingsFrame = ttk.Frame(self)
133     settingsFrame.grid(row=8, column=0, columnspan=7)
134     self.isFlip = tk.StringVar(value=True)
135     self.isTurn = tk.StringVar(value="White's Move")
136     FlipButton = ttk.Checkbutton(settingsFrame, text="Flip Board", variable=self.isFlip, onvalue=True, offvalue=False)
137     FlipButton.grid(row=0, column=0, sticky="E")
138     turnLabel = ttk.Label(settingsFrame, textvariable=self.isTurn)
139     turnLabel.grid(row=0, column=1, padx=30)
140
141     def clicked(self, event, row, column):
142         button = event.widget
143         for item in self.board:
144             for items in item:
145                 for dict in items:
146                     if button == dict:
147                         self.tag1 = items[dict].copy()
148             if button["image"] != "" and self.tag1[0][:5] == self.turn:
149                 self.firstButton = button
150                 self.row1 = row
151                 self.column1 = column
152                 self.firstClick = True
153             elif self.firstClick == True:
154                 self.row2 = row
155                 self.column2 = column
156                 self.secondButton = button
157                 self.move()
158
159     def move(self):
160         global buttonRook, buttonRook
161         for item in self.board:
162             for items in item:
163                 for dict in items:
164                     if self.firstButton == dict:
```

```
165         self.tag1 = items[dict].copy()
166     elif self.secondButton == dict:
167         self.tag2 = items[dict].copy()
168
169     if self.tag1[0][5:] == "Pawn":
170         if self.tag1[0][:5] == "white":
171             if self.secondButton["image"] == "":
172                 if self.row1 - self.row2 == 1 and self.column1 - self.column2 == 0:
173                     self.finishTurn()
174                 elif self.tag1[1][0] == 6 and self.row1 - self.row2 == 2 and self.column1 - self.column2 == 0:
175                     self.finishTurn()
176             else:
177                 if self.row1 - self.row2 == 1 and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "black":
178                     self.finishTurn()
179         elif self.tag1[0][:5] == "black":
180             if self.secondButton["image"] == "":
181                 if self.row2 - self.row1 == 1 and self.column2 - self.column1 == 0:
182                     self.finishTurn()
183                 elif self.tag1[1][0] == 1 and self.row2 - self.row1 == 2 and self.column1 - self.column2 == 0:
184                     self.finishTurn()
185             else:
186                 if self.row2 - self.row1 == 1 and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "white":
187                     self.finishTurn()
188
189     elif self.tag1[0][5:] == "Knight":
190         if self.tag1[0][:5] == "white":
191             if self.secondButton["image"] == "":
192                 if (self.row1 - self.row2 in (-2, 2) and self.column2 - self.column1 in (-1, 1)) or \
193                     (self.row1 - self.row2 in (-1, 1) and self.column2 - self.column1 in (-2, 2)):
194                     self.finishTurn()
195             else:
196                 if (self.row1 - self.row2 in (-1, 2) and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "black") or \
197                     (self.row1 - self.row2 in (-1, 1) and self.column2 - self.column1 in (-2, 2) and self.tag2[0][:5] == "black"):
198                     self.finishTurn()
199         elif self.tag1[0][:5] == "black":
200             if self.secondButton["image"] == "":
201                 if (self.row2 - self.row1 in (-2, 2) and self.column1 - self.column2 in (-1, 1)) or \
202                     (self.row2 - self.row1 in (-1, 1) and self.column1 - self.column2 in (-2, 2)):
203                     self.finishTurn()
204             else:
205                 if (self.row2 - self.row1 in (-2, 2) and self.column1 - self.column2 in (-1, 1) and self.tag2[0][:5] == "white") or \
206                     (self.row2 - self.row1 in (-1, 1) and self.column1 - self.column2 in (-2, 2) and self.tag2[0][:5] == "white"):
```

```
207         self.finishTurn()
208
209     elif self.tag1[0][5:] == "King":
210         if self.tag1[0][:5] == "white":
211             if self.secondButton["image"] == "":
212                 if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1):
213                     self.finishTurn()
214             elif self.row1 - self.row2 == 0 and self.row1 == 7:
215                 if self.column2 - self.column1 == 2:
216                     for item in self.board:
217                         for items in item:
218                             for dict, key in items.items():
219                                 if key[1][0] == self.row1 and key[1][1] == 5:
220                                     if key[0] != "":
221                                         self.isClear = False
222                                         self.doneLoop = True
223                                     break
224                                 elif key[1][0] == self.row1 and key[1][1] == 6:
225                                     if key[0] != "":
226                                         self.isClear = False
227                                         self.doneLoop = True
228                                     break
229                                 elif key[1][0] == self.row1 and key[1][1] == 7:
230                                     if key[0] != "whiteRook":
231                                         self.isClear = False
232                                         self.doneLoop = True
233                                     break
234         if self.isClear:
235             for item in self.board:
236                 for items in item:
237                     for button, key in items.items():
238                         if key[1][0] == self.row1 and key[1][1] == 7:
239                             self.buttonRook = button
240                         elif key[1][0] == self.row1 and key[1][1] == 5:
241                             self.buttonRookTo = button
242             self.buttonRook["image"] = ""
243             self.buttonRookTo["image"] = self.whiteRook
244             for item in self.board:
245                 for items in item:
246                     for dict in items:
247                         if self.buttonRook == dict:
248                             items[self.buttonRook][0] = ""
```

```
249         elif self.buttonRookTo == dict:
250             items[self.buttonRookTo][0] = "whiteRook"
251         self.finishTurn()
252     elif self.column2 - self.column1 == -2:
253         for item in self.board:
254             for items in item:
255                 for dict, key in items.items():
256                     if key[1][0] == self.row1 and key[1][1] == 1:
257                         if key[0] != "":
258                             self.isClear = False
259                             self.doneLoop = True
260                         break
261                     elif key[1][0] == self.row1 and key[1][1] == 2:
262                         if key[0] != "":
263                             self.isClear = False
264                             self.doneLoop = True
265                         break
266                     elif key[1][0] == self.row1 and key[1][1] == 3:
267                         if key[0] != "":
268                             self.isClear = False
269                             self.doneLoop = True
270                         break
271                     elif key[1][0] == self.row1 and key[1][1] == 0:
272                         if key[0] != "whiteRook":
273                             self.isClear = False
274                             self.doneLoop = True
275                         break
276     if self.isClear:
277         for item in self.board:
278             for items in item:
279                 for button, key in items.items():
280                     if key[1][0] == self.row1 and key[1][1] == 0:
281                         self.buttonRook = button
282                     elif key[1][0] == self.row1 and key[1][1] == 3:
283                         self.buttonRookTo = button
284     self.buttonRook["image"] = ""
285     self.buttonRookTo["image"] = self.whiteRook
286     for item in self.board:
287         for items in item:
288             for dict in items:
289                 if self.buttonRook == dict:
290                     items[self.buttonRook][0] = ""
```

```
291         elif self.buttonRookTo == dict:
292             items[self.buttonRookTo][0] = "whiteRook"
293         self.finishTurn()
294     else:
295         if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1) and self.tag2[0][:5] == "black":
296             self.finishTurn()
297     elif self.tag1[0][:5] == "black":
298         if self.secondButton["image"] == "":
299             if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1):
300                 self.finishTurn()
301             elif self.row1 - self.row2 == 0 and self.row1 == 0:
302                 if self.column2 - self.column1 == 2:
303                     for item in self.board:
304                         for items in item:
305                             for dict, key in items.items():
306                                 if key[1][0] == self.row1 and key[1][1] == 5:
307                                     if key[0] != "":
308                                         self.isClear = False
309                                         self.doneLoop = True
310                                     break
311                                 elif key[1][0] == self.row1 and key[1][1] == 6:
312                                     if key[0] != "":
313                                         self.isClear = False
314                                         self.doneLoop = True
315                                     break
316                                 elif key[1][0] == self.row1 and key[1][1] == 7:
317                                     if key[0] != "blackRook":
318                                         self.isClear = False
319                                         self.doneLoop = True
320                                     break
321             if self.isClear:
322                 for item in self.board:
323                     for items in item:
324                         for button, key in items.items():
325                             if key[1][0] == self.row1 and key[1][1] == 7:
326                                 self.buttonRook = button
327                             elif key[1][0] == self.row1 and key[1][1] == 5:
328                                 self.buttonRookTo = button
329             self.buttonRook["image"] = ""
330             self.buttonRookTo["image"] = self.blackRook
331         for item in self.board:
332             for items in item:
```



```
333         for dict in items:
334             if self.buttonRook == dict:
335                 items[self.buttonRook][0] = ""
336             elif self.buttonRookTo == dict:
337                 items[self.buttonRookTo][0] = "blackRook"
338         self.finishTurn()
339     elif self.column2 - self.column1 == -2:
340         for item in self.board:
341             for items in item:
342                 for dict, key in items.items():
343                     if key[1][0] == self.row1 and key[1][1] == 1:
344                         if key[0] != "":
345                             self.isClear = False
346                             self.doneLoop = True
347                         break
348                     elif key[1][0] == self.row1 and key[1][1] == 2:
349                         if key[0] != "":
350                             self.isClear = False
351                             self.doneLoop = True
352                         break
353                     elif key[1][0] == self.row1 and key[1][1] == 3:
354                         if key[0] != "":
355                             self.isClear = False
356                             self.doneLoop = True
357                         break
358                     elif key[1][0] == self.row1 and key[1][1] == 0:
359                         if key[0] != "blackRook":
360                             self.isClear = False
361                             self.doneLoop = True
362                         break
363     if self.isClear:
364         for item in self.board:
365             for items in item:
366                 for button, key in items.items():
367                     if key[1][0] == self.row1 and key[1][1] == 0:
368                         self.buttonRook = button
369                     elif key[1][0] == self.row1 and key[1][1] == 3:
370                         self.buttonRookTo = button
371     self.buttonRook["image"] = ""
372     self.buttonRookTo["image"] = self.blackRook
373
374     for item in self.board:
375         for items in item:
```

```
375         for dict in items:
376             if self.buttonRook == dict:
377                 items[self.buttonRook][0] = ""
378             elif self.buttonRookTo == dict:
379                 items[self.buttonRookTo][0] = "blackRook"
380             self.finishTurn()
381     else:
382         if self.row1 - self.row2 in (-1, 0, 1) and self.column2 - self.column1 in (-1, 0, 1) and self.tag2[0][:5] == "white":
383             self.finishTurn()
384
385 elif self.tag1[0][:5] == "Bishop":
386     if self.tag1[0][:5] == "white":
387         if self.secondButton["image"] == "":
388             if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
389                 if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
390                     self.diagMove("<", "<")
391                 elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
392                     self.diagMove("<", ">")
393                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
394                     self.diagMove(">", ">")
395                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
396                     self.diagMove(">", "<")
397         else:
398             if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "black":
399                 if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
400                     self.diagMove("<", "<", True)
401                 elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
402                     self.diagMove("<", ">", True)
403                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
404                     self.diagMove(">", ">", True)
405                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
406                     self.diagMove(">", "<", True)
407     if self.tag1[0][:5] == "black":
408         if self.secondButton["image"] == "":
409             if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
410                 if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
411                     self.diagMove("<", "<")
412                 elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
413                     self.diagMove("<", ">")
414                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
415                     self.diagMove(">", ">")
416                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
```

```
417         self.diagMove(">", "<")
418     else:
419         if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "white":
420             if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
421                 self.diagMove("<", "<", True)
422             elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
423                 self.diagMove("<", ">", True)
424             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
425                 self.diagMove(">", ">", True)
426             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
427                 self.diagMove(">", "<", True)
428
429     elif self.tag1[0][5:] == "Rook":
430         if self.tag1[0][:5] == "white":
431             if self.secondButton["image"] == "":
432                 if self.row2 - self.row1 == 0 and self.column2 < self.column1:
433                     self.diagMove("==", "<", False, 1)
434                 elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
435                     self.diagMove("==", ">", False, 1)
436                 elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
437                     self.diagMove(">", "==", False, 0)
438                 elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
439                     self.diagMove("<", "==", False, 0)
440             else:
441                 if self.tag2[0][:5] == "black":
442                     if self.row2 - self.row1 == 0 and self.column2 < self.column1:
443                         self.diagMove("==", "<", True, 1)
444                     elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
445                         self.diagMove("==", ">", True, 1)
446                     elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
447                         self.diagMove(">", "==", True, 0)
448                     elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
449                         self.diagMove("<", "==", True, 0)
450         if self.tag1[0][:5] == "black":
451             if self.secondButton["image"] == "":
452                 if self.row2 - self.row1 == 0 and self.column2 > self.column1:
453                     self.diagMove("==", ">", False, 1)
454                 elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
455                     self.diagMove("==", "<", False, 1)
456                 elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
457                     self.diagMove(">", "==", False, 0)
458                 elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
```

```
459         self.diagMove("<", "==", False, 0)
460     else:
461         if self.tag2[0][:5] == "white":
462             if self.row2 - self.row1 == 0 and self.column2 > self.column1:
463                 self.diagMove("==", ">", True, 1)
464             elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
465                 self.diagMove("==", "<", True, 1)
466             elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
467                 self.diagMove(">", "==", True, 0)
468             elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
469                 self.diagMove("<", "==", True, 0)
470
471         elif self.tag1[0][5:] == "Queen":
472             if self.tag1[0][:5] == "white":
473                 if self.secondButton["image"] == "":
474                     if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
475                         if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
476                             if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
477                                 self.diagMove("<", "<")
478                             elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
479                                 self.diagMove("<", ">")
480                             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
481                                 self.diagMove(">", ">")
482                             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
483                                 self.diagMove(">", "<")
484                         elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
485                             self.diagMove("==", "<", False, 1)
486                         elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
487                             self.diagMove("==", ">", False, 1)
488                         elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
489                             self.diagMove(">", "==", False, 0)
490                         elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
491                             self.diagMove("<", "==", False, 0)
492                     else:
493                         if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "black":
494                             if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
495                                 self.diagMove("<", "<", True)
496                             elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
497                                 self.diagMove("<", ">", True)
498                             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
499                                 self.diagMove(">", ">", True)
500                             elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
```

```

501         self.diagMove(">", "<", True)
502     elif self.tag2[0][:5] == "black":
503         if self.row2 - self.row1 == 0 and self.column2 < self.column1:
504             self.diagMove("==", "<", True, 1)
505         elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
506             self.diagMove("==", ">", True, 1)
507         elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
508             self.diagMove(">", "==", True, 0)
509         elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
510             self.diagMove("<", "==", True, 0)
511     if self.tag1[0][:5] == "black":
512         if self.secondButton["image"] == "":
513             if abs(self.row2 - self.row1) == abs(self.column2 - self.column1):
514                 if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
515                     self.diagMove("<", "<")
516                 elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
517                     self.diagMove("<", ">")
518                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
519                     self.diagMove(">", ">")
520                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
521                     self.diagMove(">", "<")
522             elif self.row2 - self.row1 == 0 and self.column2 < self.column1:
523                 self.diagMove("==", "<", False, 1)
524             elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
525                 self.diagMove("==", ">", False, 1)
526             elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
527                 self.diagMove(">", "==", False, 0)
528             elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
529                 self.diagMove("<", "==", False, 0)
530         else:
531             if abs(self.row2 - self.row1) == abs(self.column2 - self.column1) and self.tag2[0][:5] == "white":
532                 if self.row2 - self.row1 < 0 and self.column2 - self.column1 < 0:
533                     self.diagMove("<", "<", True)
534                 elif self.row2 - self.row1 < 0 and self.column2 - self.column1 > 0:
535                     self.diagMove("<", ">", True)
536                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 > 0:
537                     self.diagMove(">", ">", True)
538                 elif self.row2 - self.row1 > 0 and self.column2 - self.column1 < 0:
539                     self.diagMove(">", "<", True)
540             elif self.tag2[0][:5] == "white":
541                 if self.row2 - self.row1 == 0 and self.column2 < self.column1:
542                     self.diagMove("==", "<", True, 1)

```

```
543         elif self.row2 - self.row1 == 0 and self.column2 > self.column1:
544             self.diagMove("==", ">", True, 1)
545         elif self.row2 > self.row1 and self.column2 - self.column1 == 0:
546             self.diagMove(">", "==", True, 0)
547         elif self.row2 < self.row1 and self.column2 - self.column1 == 0:
548             self.diagMove("<", "==", True, 0)
549
550     self.firstClick = False
551     self.isClear = True
552     self.doneLoop = False
553
554     def diagMove(self, operator1, operator2, touch=False, isRook=-1):
555         operators = {"<": operator.lt, ">": operator.gt, "==": operator.eq}
556         if isRook == 1:
557             self.increment1 = 0
558             if operator2 == "<":
559                 self.increment2 = 1
560             elif operator2 == ">":
561                 self.increment2 = -1
562             secondOperator = operators[operator2](self.column2, self.column1)
563         else:
564             secondOperator = operators[operator2](self.column2 - self.column1, 0)
565         if isRook == 0:
566             self.increment2 = 0
567             if operator1 == "<":
568                 self.increment1 = 1
569             elif operator1 == ">":
570                 self.increment1 = -1
571             firstOperator = operators[operator1](self.row2, self.row1)
572         else:
573             firstOperator = operators[operator1](self.row2 - self.row1, 0)
574         if touch:
575             if operator1 == "<" and isRook == -1:
576                 self.increment1 = 1
577             elif operator1 == ">" and isRook == -1:
578                 self.increment1 = -1
579             if operator2 == "<" and isRook == -1:
580                 self.increment2 = 1
581             elif operator2 == ">" and isRook == -1:
582                 self.increment2 = -1
583         else:
584             self.increment1 = 0
```

```
585         self.increment2 = 0
586     if firstOperator and secondOperator:
587         while not self.doneLoop:
588             if self.tag2[1][0] + self.increment1 == self.row1 and self.tag2[1][1] + self.increment2 == self.column1:
589                 break
590             if isRook == -1:
591                 if operator1 == "<":
592                     self.row1 -= 1
593                 elif operator1 == ">":
594                     self.row1 += 1
595                 if operator2 == "<":
596                     self.column1 -= 1
597                 elif operator2 == ">":
598                     self.column1 += 1
599             elif isRook == 0:
600                 if operator1 == "<":
601                     self.row1 -= 1
602                 elif operator1 == ">":
603                     self.row1 += 1
604             elif isRook == 1:
605                 if operator2 == "<":
606                     self.column1 -= 1
607                 elif operator2 == ">":
608                     self.column1 += 1
609             for item in self.board:
610                 for items in item:
611                     for dict, key in items.items():
612                         if key[1][0] == self.row1 and key[1][1] == self.column1:
613                             if key[0] != "":
614                                 self.isClear = False
615                                 self.doneLoop = True
616                                 break
617             if self.isClear:
618                 self.finishTurn()
619
620     def finishTurn(self):
621         if self.tag2[0] == "blackKing":
622             gameOver("White")
623             self.switchTag()
624         elif self.tag2[0] == "whiteKing":
625             gameOver("Black")
626             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
668                     place.append(self.row)
669                     place.append(self.column)
```



```
669         button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
670         button.grid(row=7 - self.row, column=7 - self.column, sticky="NSEW", padx=0, pady=0)
671         self.column += 1
672         self.row += 1
673         self.turn = "black"
674         self.isTurn.set("Black's Move")
675         self.flip = False
676     else:
677         self.row = 0
678         for row in self.board:
679             self.column = 0
680             for buttons in row:
681                 for button, ids in buttons.items():
682                     id, place = ids
683                     place = [self.row, self.column]
684                     button.bind("<Button-1>", lambda event, x=self.row, y=self.column: self.clicked(event, x, y))
685                     button.grid(row=self.row, column=self.column, sticky="NSEW", padx=0, pady=0)
686                     self.column += 1
687                     self.row += 1
688                     self.turn = "white"
689                     self.isTurn.set("White's Move")
690             self.flip = False
691
692     def changePiece(self, piece):
693         board.tkraise()
694         self.tag1[0] = piece
695         self.finishTurn()
696
697 root = tk.Tk()
698
699 # All of the following images were cropped from: https://pixabay.com/illustrations/chess-black-and-white-pieces-3413429/
700 bundle_dir = getattr(sys, "_MEIPASS", path.abspath(path.dirname(__file__)))
701 pathToWhitePawn = path.join(bundle_dir, "assets", "whitePawn.png")
702 whitePawnImage = Image.open(pathToWhitePawn)
703 whitePawnImage = whitePawnImage.resize((40, 60))
704 whitePawn = ImageTk.PhotoImage(whitePawnImage)
705
706 pathToBlackPawn = path.join(bundle_dir, "assets", "blackPawn.png")
707 blackPawnImage = Image.open(pathToBlackPawn)
708 blackPawnImage = blackPawnImage.resize((40, 60))
709 blackPawn = ImageTk.PhotoImage(blackPawnImage)
710
```

```
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)
```

```
753 whiteBishopImage = whiteBishopImage.resize((40, 60))
754 whiteBishop = ImageTk.PhotoImage(whiteBishopImage)
755
756 pathToWhiteKing = path.join(bundle_dir, "assets", "whiteKing.png")
757 whiteKingImage = Image.open(pathToWhiteKing)
758 whiteKingImage = whiteKingImage.resize((40, 60))
759 whiteKing = ImageTk.PhotoImage(whiteKingImage)
760
761 pathToWhiteQueen = path.join(bundle_dir, "assets", "whiteQueen.png")
762 whiteQueenImage = Image.open(pathToWhiteQueen)
763 whiteQueenImage = whiteQueenImage.resize((40, 60))
764 whiteQueen = ImageTk.PhotoImage(whiteQueenImage)
765
766 def endRow(turn):
767     if turn == "white":
768         frameWhite.tkraise()
769     else:
770         frameBlack.tkraise()
771
772 def gameOver(winner):
773     gameOverFrame = ttk.Frame(root, width=560, height=560, padding=5)
774     gameOverFrame.grid(row=0, column=0)
775     style = ttk.Style(frameWhite)
776     style.theme_use("clam")
777     gameOverLabel = ttk.Label(gameOverFrame, text="Game Over", font=("Courier", 30))
778     winnerLabel = ttk.Label(gameOverFrame, text=f"{winner} Wins!", font=("Courier", 30))
779     gameOverLabel.grid()
780     winnerLabel.grid()
781
782 board = Board(root, whitePawn, whiteRook, whiteKnight, whiteBishop, whiteKing, whiteQueen,
783               blackKing, blackQueen, blackBishop, blackKnight, blackRook, blackPawn)
784 board.grid(row=0, column=0)
785
786 frameWhite = ttk.Frame(root, width=560, height=560, padding=5)
787 frameWhite.grid(row=0, column=0)
788 style = ttk.Style(frameWhite)
789
790 whiteLabel = ttk.Label(frameWhite, text="Choose a piece", font=("Courier", 25), padding=10)
791 whiteRookFrame = ttk.Button(frameWhite, im=whiteRook, padding=30, command=lambda: board.changePiece("whiteRook"))
792 whiteKnightFrame = ttk.Button(frameWhite, im=whiteKnight, padding=30, command=lambda: board.changePiece("whiteKnight"))
793 whiteBishopFrame = ttk.Button(frameWhite, im=whiteBishop, padding=30, command=lambda: board.changePiece("whiteBishop"))
794 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](https://bakerfranke.github.io/codePrint/)