talalz94 / Ludo-Python-game- Public

Code Issues Pull requests Actions Projects Wiki Security Insights

master

Ludo-Python-game- / Ludo-game.py / Jump to

talalz94 Add files via upload

1 contributor

847 lines (682 sloc) | 32.2 KB

```
1  from tkinter import *  # Tkinter is used as the GUI.
2  from tkinter import messagebox
3  import sys
4  import os
5  import random
6  import tkinter.messagebox
7
8  root = Tk()
9
10 root.resizable(width=False, height=False)  # The window size of the game.
11 root.geometry('1000x750')
12 root.configure(background='green')
13 root.title("Checkers")
14
15 logo = PhotoImage(file="whitebox.gif")      # Loading all the image files that are required in
16 logo2 = PhotoImage(file="red side.gif")     # Loading all the image files that are required in
17 logo3 = PhotoImage(file="red.gif")          # Loading all the image files that are required in
18 logo4 = PhotoImage(file="blue side.gif")
19 logo5 = PhotoImage(file="green side.gif")
20 logo6 = PhotoImage(file="yellow side.gif")
21 logo7 = PhotoImage(file="center.gif")
22 logoxx = PhotoImage(file="test.gif")
23 logog = PhotoImage(file="greenbox.gif")
24 logogs = PhotoImage(file="greenstop.gif")
25 logoy = PhotoImage(file="yellowbox.gif")
26 logoys = PhotoImage(file="yellowstop.gif")
27 logob = PhotoImage(file="bluebox.gif")
28 logobs = PhotoImage(file="bluestop.gif")
29 logor = PhotoImage(file="redbox.gif")
30 logors = PhotoImage(file="redstop.gif")
31 logoh = PhotoImage(file="head.gif")
32 logot = PhotoImage(file="tail.gif")
```

```
33  logoh1 = PhotoImage(file="head1.gif")
34  logot1 = PhotoImage(file="tail1.gif")
35  logoh2 = PhotoImage(file="head2.gif")
36  logot2 = PhotoImage(file="tail2.gif")
37  logoh3 = PhotoImage(file="head3.gif")
38  logot3 = PhotoImage(file="tail3.gif")
39  logoab= PhotoImage(file="blue.gif")
40  logoay= PhotoImage(file="yellow.gif")
41  logoag= PhotoImage(file="green.gif")
42
43  Label(image=logo2, width=298, height=298).place(x=-1, y=-1)               #setting up board ima
44  Label(image=logo4, width=300, height=300).place(x=(-2), y=(448))
45  Label(image=logo5, width=296, height=296).place(x=(450), y=(0))
46  Label(image=logo6, width=294, height=294).place(x=(450), y=(450))
47  Label(image=logo7, width=150, height=150).place(x=(298), y=(298))
48
49  c = 0                                #initializing variable and flags that are to be used in th
50  lx = 0
51  bb =0
52  nc = 0
53  rollc = 0
54  rolls = []
55  RED = True
56  BLUE = False
57  GREEN = False
58  YELLOW = False
59  TURN = True
60  REDKILL = False
61  BLUEKILL = False
62  GREENKILL = False
63  YELLOWKILL = False
64
65
66  def board():                             #Drawing the board, piece by piece.
67
68                                           #Splash Screen.
69      tkinter.messagebox.showinfo(title=None, message="TO START GAME PRESS OKAY & TO EXIT PRESS C
70      v = 0
71      z = 0
72
73      while (v != 300):           #Drawing White boxes
74          z = 0
75          while (z != 150):
76              Label(image=logo, width=46, height=46).place(x=(300 + z), y=(0 + v))
77              z = z + 50
78          v = v + 50
79
80      z = 0
81      v = 0
82      while (v != 300):          #Drawing White boxes
83          z = 0
84          while (z != 150):
```

```
85              Label(image=logo, width=46, height=46).place(x=(0 + v), y=(300 + z))
86              z = z + 50
87          v = v + 50
88
89      ######################
90
91      v = 0
92      z = 0
93
94      while (v != 300):            #Drawing White boxes
95          z = 0
96          while (z != 150):
97              Label(image=logo, width=46, height=46).place(x=(300 + z), y=(450 + v))
98              z = z + 50
99          v = v + 50
100
101     z = 0
102     v = 0
103     while (v != 300):         #Drawing White boxes
104         z = 0
105         while (z != 150):
106             Label(image=logo, width=46, height=46).place(x=(450 + v), y=(300 + z))
107             z = z + 50
108         v = v + 50
109
110     v = 0
111     while (v != 250):     #Drawing Green boxes
112         Label(image=logog, width=46, height=46).place(x=(350), y=(50 + v))
113         v = v + 50
114
115     Label(image=logog, width=46, height=46).place(x=(300), y=(100))
116     Label(image=logogs, width=46, height=46).place(x=(400), y=(50))
117
118     v = 0
119     while (v != 250):     #Drawing Yellow boxes
120         Label(image=logoy, width=46, height=46).place(x=(450 + v), y=(350))
121         v = v + 50
122
123     Label(image=logoy, width=46, height=46).place(x=(600), y=(300))
124     Label(image=logoys, width=46, height=46).place(x=(650), y=(400))
125
126     v = 0
127     while (v != 250):    #Drawing Red Boxes
128         Label(image=logor, width=46, height=46).place(x=(50 + v), y=(350))
129         v = v + 50
130
131     Label(image=logor, width=46, height=46).place(x=(100), y=(400))
132     Label(image=logors, width=46, height=46).place(x=(50), y=(300))
133
134     v = 0
135     while (v != 250):    #Drawing Blue Boxes
136         Label(image=logob, width=46, height=46).place(x=(350), y=(450 + v))
```

```
137         v = v + 50
138 
139     Label(image=logobs, width=46, height=46).place(x=(300), y=(650))
140     Label(image=logob, width=46, height=46).place(x=(400), y=(600))
141 
142     Label(image=logoh, width=46, height=46).place(x=250, y=400)        #Drawing arrows
143     Label(image=logot, width=46, height=46).place(x=300, y=450)
144     Label(image=logoh1, width=46, height=46).place(x=400, y=450)
145     Label(image=logot1, width=46, height=46).place(x=450, y=400)
146     Label(image=logoh2, width=46, height=46).place(x=450, y=300)
147     Label(image=logot2, width=46, height=46).place(x=400, y=250)
148     Label(image=logoh3, width=46, height=46).place(x=300, y=250)
149     Label(image=logot3, width=46, height=46).place(x=250, y=300)
150 
151 class YBox:                              #Class of yellow box
152     rap = None
153 
154     def __init__(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):
155         self.num = num                #no of gamepiece acc to box
156         self.x = x                    #initial and final co-ordinates of the boxes
157         self.y = y
158         self.x0 = x0
159         self.y0 = y0
160         self.rap = Label(image=logoay, width=20, height=20)        #image of game piece.
161         self.double = double                                       #if one game piece on top o
162 
163     def swap(self):                     #Swaps the position of gamepiece according to the numbe
164         self.rap.place(x=self.x0 + 13, y=self.y0 + 14)
165 
166 class GBox:                              #Class of green box
167     rap = None
168 
169     def __init__(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):
170         self.num = num
171         self.x = x
172         self.y = y
173         self.x0 = x0
174         self.y0 = y0
175         self.rap = Label(image=logoag, width=20, height=20)
176         self.double = double
177 
178     def swap(self):
179         self.rap.place(x=self.x0 + 13, y=self.y0 + 14)
180 
181 class BBox:                             #Class of Blue box
182     rap = None
183 
184     def __init__(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):
185         self.num = num
186         self.x = x
187         self.y = y
188         self.x0 = x0
```

```
        self.y0 = y0
        self.rap = Label(image=logoab, width=20, height=20)
        self.double = double

    def swap(self):
        self.rap.place(x=self.x0 + 13, y=self.y0 + 14)

class Box:                          #class of red box
    rap = None

    def __init__(self, num=-1, x=0, y=0, x0=0, y0=0, double=False, ):
        self.num = num
        self.x = x
        self.y = y
        self.x0 = x0
        self.y0 = y0
        self.rap = Label(image=logo3, width=20, height=20)
        self.double = double

    def swap(self):
        self.rap.place(x=self.x0 + 13, y=self.y0 + 14)


def main():                             # Main game function.

    global box, redbox, bluebox, greenbox, yellowbox, redhome, bluehome, yellowhome, greenhome
    global red, blue, yellow, green, rap, RED, BLUE, GREEN, YELLOW, dice, nc, TURN, bb

    if c == 0:                          #constructs the game pieces first time the code is

        board()

        box = [Box() for i in range(52)]  # list of co-ordinates of all the outer boxes

        redbox = [Box() for i in range(57)]  # list of co-ordinates of all the colored boxes, e
        bluebox = [Box() for i in range(57)]
        greenbox = [Box() for i in range(57)]
        yellowbox = [Box() for i in range(57)]

        redhome = [Box() for i in range(4)]  # list co-ordinates of all the home positions
        bluehome = [Box() for i in range(4)]
        greenhome = [Box() for i in range(4)]
        yellowhome = [Box() for i in range(4)]

        red = [Box() for i in range(4)]  # list of co-ordinates of all the game pieces in their
        blue = [BBox() for i in range(4)]  # that is equal to their respective home co-ordinate
        green = [GBox() for i in range(4)]
        yellow = [YBox() for i in range(4)]

        for i in range(2):                      #Populates list of homeboxes, colored boxes,
            redhome[i].x = (100 + (100 * i))
            redhome[i].y = 100
```

```
241             red[i].x0 = redhome[i].x
242             red[i].y0 = redhome[i].y
243             red[i].x = (red[i].x0) + 25
244             red[i].y = (red[i].y0) + 25
245 
246             bluehome[i].x = (100 + (100 * i))
247             bluehome[i].y = (550)
248             blue[i].x0 = bluehome[i].x
249             blue[i].y0 = bluehome[i].y
250             blue[i].x = (blue[i].x0) + 25
251             blue[i].y = (blue[i].y0) + 25
252 
253             yellowhome[i].x = (550 + (100 * i))
254             yellowhome[i].y = (550)
255             yellow[i].x0 = yellowhome[i].x
256             yellow[i].y0 = yellowhome[i].y
257             yellow[i].x = (yellow[i].x0) + 25
258             yellow[i].y = (yellow[i].y0) + 25
259 
260             greenhome[i].x = (550 + (100 * i))
261             greenhome[i].y = (100)
262             green[i].x0 = greenhome[i].x
263             green[i].y0 = greenhome[i].y
264             green[i].x = (green[i].x0) + 25
265             green[i].y = (green[i].y0) + 25
266 
267         for i in range(2, 4):
268             redhome[i].x = (100 + (100 * (i - 2)))
269             redhome[i].y = 200
270             red[i].x0 = redhome[i].x
271             red[i].y0 = redhome[i].y
272             red[i].x = (red[i].x0) + 25
273             red[i].y = (red[i].y0) + 25
274 
275             bluehome[i].x = (100 + (100 * (i - 2)))
276             bluehome[i].y = (650)
277             blue[i].x0 = bluehome[i].x
278             blue[i].y0 = bluehome[i].y
279             blue[i].x = (blue[i].x0) + 25
280             blue[i].y = (blue[i].y0) + 25
281 
282             yellowhome[i].x = (550 + (100 * (i - 2)))
283             yellowhome[i].y = (650)
284             yellow[i].x0 = yellowhome[i].x
285             yellow[i].y0 = yellowhome[i].y
286             yellow[i].x = (yellow[i].x0) + 25
287             yellow[i].y = (yellow[i].y0) + 25
288 
289             greenhome[i].x = (550 + (100 * (i - 2)))
290             greenhome[i].y = 200
291             green[i].x0 = greenhome[i].x
292             green[i].y0 = greenhome[i].y
```

```
            green[i].x = (green[i].x0) + 25
            green[i].y = (green[i].y0) + 25

        for i in range(6):
            box[i].x = 300
            box[i].y = (700 - (50 * i))

        for i in range(6, 12):
            box[i].x = (250 - (50 * (i - 6)))
            box[i].y = (400)

        box[12].x = 0
        box[12].y = 350

        for i in range(13, 19):
            box[i].x = (0 + (50 * (i - 13)))
            box[i].y = (300)

        for i in range(19, 25):
            box[i].x = (300)
            box[i].y = (250 - (50 * (i - 19)))

        box[25].x = 350
        box[25].y = 0

        for i in range(26, 32):
            box[i].x = (400)
            box[i].y = (0 + (50 * (i - 26)))

        for i in range(32, 38):
            box[i].x = (450 + (50 * (i - 32)))
            box[i].y = (300)

        box[38].x = 700
        box[38].y = 350

        for i in range(39, 45):
            box[i].x = (700 - (50 * (i - 39)))
            box[i].y = (400)

        for i in range(45, 51):
            box[i].x = (400)
            box[i].y = (450 + (50 * (i - 45)))

        box[51].x = 350
        box[51].y = 700

        # teshh
        lx = 14
        for i in range(52):
            redbox[i].x = box[lx].x
            redbox[i].y = box[lx].y
```

```
            lx = lx + 1
            if lx > 51:
                lx = 0

        lx = 50
        for i in range(7):
            redbox[lx].x = (0 + (50 * i))
            redbox[lx].y = 350
            lx = lx + 1
        # blue
        lx = 1
        for i in range(52):

            bluebox[i].x = box[lx].x
            bluebox[i].y = box[lx].y
            lx = lx + 1
            if lx > 51:
                lx = 0

        lx = 50
        for i in range(7):
            bluebox[lx].x = 350
            bluebox[lx].y = (700 - (50 * i))
            lx = lx + 1
        # yellow
        lx = 40
        for i in range(52):
            yellowbox[i].x = box[lx].x
            yellowbox[i].y = box[lx].y
            lx = lx + 1
            if lx > 51:
                lx = 0

        lx = 50
        for i in range(7):
            yellowbox[lx].x = (700 - (50 * i))
            yellowbox[lx].y = (350)
            lx = lx + 1

        # green
        lx = 27
        for i in range(52):

            greenbox[i].x = box[lx].x
            greenbox[i].y = box[lx].y

            lx = lx + 1
            if lx > 51:
                lx = 0

        lx = 50
        for i in range(7):
```

```
            greenbox[lx].x = 350
            greenbox[lx].y = (0 + (50 * i))
            lx = lx + 1

        for i in range(4):
            red[i].swap()
            blue[i].swap()
            green[i].swap()
            yellow[i].swap()                        #Population of all list is completed. Now ga


    else:  # HERE ALL THE GAME OCCURS ... IF WAGHAIRA, MOVEMENT IDHAR HOGI !!!

        if c >= 1:                                  #This condition is true when a click is made

            if RED == True and TURN == False:       #Red players turn
                print("Red's Turn")
                print("moves available: ", rolls)
                la = "RED"
                if (movecheck(red, redhome, redbox, la)) == False:  #Checks if player can take
                    BLUE = True
                    RED = False
                    clear()                                          #clears variable, next pla

                if RED == True:                                    # searches if click is made
                    for i in range(len(red)):
                        if ((((cx > red[i].x0 + 13) and (cx < red[i].x + 13)) and (
                            (cy > red[i].y0 + 14) and (cy < red[i].y + 14)))
                            and (red[i].x0 == redhome[i].x) and (red[i].y0 == redhome[i].y)):
                            print("woila ")

                            if rolls[0 + nc] == 6:                 #If a six occurs and gamepie
                                                                    #Game piece is moved onto
                                red[i].x0 = redbox[0].x
                                red[i].y0 = redbox[0].y
                                red[i].x = redbox[0].x + 25
                                red[i].y = redbox[0].y + 25
                                red[i].num = 0
                                red[i].swap()
                                nc = nc + 1

                                if nc > len(rolls) - 1:           # check if all moves are mad
                                    BLUE = True
                                    RED = False
                                    clear()
                                break

                            if ((((cx > red[i].x0 + 13) and (cx < red[i].x + 13)) and (       #if
                                (cy > red[i].y0 + 14) and (cy < red[i].y + 14)))
                                and ((red[i].x0 > 270) or (red[i].y0 > 270))):
                                print("woila ")
                                bb = ((red[i].num) + rolls[0 + nc])
```

```
449                             # Winning condition
450
451                             if bb > 57:                         #prevents moves greater than al
452                                 break
453                                 #bb = ((red[i].num) + rolls[0 + nc]) - 57
454
455                             kill(redbox,blue,yellow,green,bluehome,yellowhome,greenhome)      #
456
457                             red[i].x0 = redbox[bb].x
458                             red[i].y0 = redbox[bb].y
459                             red[i].x = redbox[bb].x + 25
460                             red[i].y = redbox[bb].y + 25
461                             red[i].swap()
462                             red[i].num = bb
463                             doublecheck(red)                            #checks if the gamepiec
464
465                             nc = nc + 1
466                             if bb == 57:                               #checks if game has trav
467                                 # del red[i]
468                                 red.remove(red[i]);
469
470                             if nc > len(rolls) - 1:
471                                 BLUE = True                         #next players turn.
472                                 RED = False
473                                 clear()
474                             break
475
476
477                         # BLUES TURN!!!!!!!!!!!!!!!!!!!
478
479             if BLUE == True and TURN == False:                          #same as REDS CODE
480                 print("Blue's Turn")
481                 print("moves available: ", rolls)
482                 la="BLUE"
483                 if (movecheck(blue, bluehome, bluebox, la)) == False:
484                     print("NO MOVES SIR JEE")
485                     BLUE = False
486                     YELLOW = True
487                     clear()
488
489                 if BLUE == True:
490
491                     for i in range(len(blue)):
492                         if ((((cx > blue[i].x0 + 13) and (cx < blue[i].x + 13)) and (
493                             (cy > blue[i].y0 + 14) and (cy < blue[i].y + 14)))
494                             and (blue[i].x0 == bluehome[i].x) and (blue[i].y0 == bluehome[i].y
495                             print("woila ")
496
497                             if rolls[0 + nc] == 6:
498
499                                 blue[i].x0 = bluebox[0].x
500                                 blue[i].y0 = bluebox[0].y
```

```
501                             blue[i].x = bluebox[0].x + 25
502                             blue[i].y = bluebox[0].y + 25
503                             blue[i].num = 0
504                             blue[i].swap()
505                             nc = nc + 1
506 
507                             if nc > len(rolls) - 1:
508                                 YELLOW = True
509                                 BLUE = False
510                                 clear()
511                             break
512 
513                         if ((((cx > blue[i].x0 + 13) and (cx < blue[i].x + 13)) and (
514                             (cy > blue[i].y0 + 14) and (cy < blue[i].y + 14)))
515                             and ((blue[i].x0 > 270) or (blue[i].y0 < 470))):
516                             print("woila ")
517                             bb = ((blue[i].num) + rolls[0 + nc])
518                             if bb > 57:
519                                 break
520                                 # bb= ((blue[i].num) + rolls[0 + nc]) - 52
521 
522                             kill(bluebox,red,yellow,green,redhome,yellowhome,greenhome)
523 
524                             blue[i].x0 = bluebox[bb].x
525                             blue[i].y0 = bluebox[bb].y
526                             blue[i].x = bluebox[bb].x + 25
527                             blue[i].y = bluebox[bb].y + 25
528                             blue[i].swap()
529                             blue[i].num = bb
530                             doublecheck(blue)
531                             nc = nc + 1
532                             if bb == 57:
533                                 # del red[i]
534                                 blue.remove(blue[i]);
535 
536                             if nc > len(rolls) - 1:
537                                 YELLOW = True
538                                 BLUE = False
539                                 clear()
540                             break
541 
542                         # YELLOWS TURN!!!!!!!!!!!!!!!!!!!
543 
544             if YELLOW == True and TURN == False:                    #Same as RED's code
545                 print("Yellows's Turn")
546                 print("moves available: ", rolls)
547                 la="YELLOW"
548                 if (movecheck(yellow, yellowhome, yellowbox,la)) == False:
549                     print("NO MOVES SIR JEE")
550                     YELLOW = False
551                     GREEN = True
552                     clear()
```

```
553
554                 if YELLOW == True:
555
556                     for i in range(len(yellow)):
557                         if ((((cx > yellow[i].x0 + 13) and (cx < yellow[i].x + 13)) and (
558                                     (cy > yellow[i].y0 + 14) and (cy < yellow[i].y + 14)))
559                             and (yellow[i].x0 == yellowhome[i].x) and (yellow[i].y0 == yellowho
560                             print("woila ")
561
562                             if rolls[0 + nc] == 6:
563
564                                 yellow[i].x0 = yellowbox[0].x
565                                 yellow[i].y0 = yellowbox[0].y
566                                 yellow[i].x = yellowbox[0].x + 25
567                                 yellow[i].y = yellowbox[0].y + 25
568                                 yellow[i].num = 0
569                                 yellow[i].swap()
570                                 nc = nc + 1
571
572                                 if nc > len(rolls) - 1:
573                                     YELLOW = False
574                                     GREEN = True
575                                     clear()
576                                 break
577
578                             if ((((cx > yellow[i].x0 + 13) and (cx < yellow[i].x + 13)) and (
579                                     (cy > yellow[i].y0 + 14) and (cy < yellow[i].y + 14)))
580                             and ((yellow[i].x0 < 470) or (yellow[i].y0 < 470))):
581                             print("woila ")
582                             bb = ((yellow[i].num) + rolls[0 + nc])
583                             if bb > 57:
584                                 break
585                                 #bb = ((yellow[i].num) + rolls[0 + nc]) - 52
586
587                             kill(yellowbox,blue,red,green,bluehome,redhome,greenhome)
588
589                             yellow[i].x0 = yellowbox[bb].x
590                             yellow[i].y0 = yellowbox[bb].y
591                             yellow[i].x = yellowbox[bb].x + 25
592                             yellow[i].y = yellowbox[bb].y + 25
593                             yellow[i].swap()
594                             yellow[i].num = bb
595                             doublecheck(yellow)
596                             nc = nc + 1
597                             if bb == 57:
598                                 # del red[i]
599                                 yellow.remove(yellow[i]);
600
601                             if nc > len(rolls) - 1:
602                                 YELLOW = False
603                                 GREEN = True
604                                 clear()
```

```
605                             break
606
607
608                         # GREENS TURN!!!!!!!!!!!!!!!!!!!
609
610             if GREEN == True and TURN == False:                     #Same as RED's code
611                 print("Green's Turn")
612                 print("moves available: ", rolls)
613                 la="GREEN"
614                 if (movecheck(green, greenhome, greenbox,la)) == False:
615                     print("NO MOVES SIR JEE")
616                     GREEN = False
617                     RED = True
618                     clear()
619
620                 if GREEN == True:
621
622                     for i in range(len(green)):
623                         if ((((cx > green[i].x0 + 13) and (cx < green[i].x + 13)) and (
624                                     (cy > green[i].y0 + 14) and (cy < green[i].y + 14)))
625                             and (green[i].x0 == greenhome[i].x) and (green[i].y0 == greenhome[
626                             print("woila ")
627
628                             if rolls[0 + nc] == 6:
629
630                                 green[i].x0 = greenbox[0].x
631                                 green[i].y0 = greenbox[0].y
632                                 green[i].x = greenbox[0].x + 25
633                                 green[i].y = greenbox[0].y + 25
634                                 green[i].num = 0
635                                 green[i].swap()
636                                 nc = nc + 1
637                                 print("green x.y: ", green[i].x0, green[i].y0)
638
639                                 if nc > len(rolls) - 1:
640                                     GREEN = False
641                                     RED = True
642                                     clear()
643                                 break
644
645                         if ((((cx > green[i].x0 + 13) and (cx < green[i].x + 13)) and (
646                                     (cy > green[i].y0 + 14) and (cy < green[i].y + 14)))
647                             and ((green[i].x0 < 470) or (green[i].y0 < 470))):
648                             print("woila ")
649                             bb = ((green[i].num) + rolls[0 + nc])
650                             if bb > 57:
651                                 break
652                                 # bb = ((green[i].num) + rolls[0 + nc]) - 52
653
654                             kill(greenbox,blue,yellow,red,bluehome,yellowhome,redhome)
655
656                             green[i].x0 = greenbox[bb].x
```

```
                            green[i].y0 = greenbox[bb].y
                            green[i].x = greenbox[bb].x + 25
                            green[i].y = greenbox[bb].y + 25
                            green[i].swap()
                            green[i].num = bb
                            nc = nc + 1
                            doublecheck(green)
                            if bb == 57:
                                # del red[i]
                                green.remove(green[i]);

                            if nc > len(rolls) - 1:
                                GREEN = False
                                RED = True
                                clear()
                            break


main()    #Main functin is called once when c==0 to intialize all the gamepieces.


def leftClick(event):  # Main play function is called on every left click.

    global c, cx, cy, RED, YELLOW
    c = c + 1
    cx = root.winfo_pointerx() - root.winfo_rootx()  # This formula returns the x,y co-ordinate
    cy = root.winfo_pointery() - root.winfo_rooty()

    print("Click at: ", cx, cy)

    main()           #Main function called on every click to progress the game


root.bind("<Button-1>", leftClick)


def turn():   #Prints whoose turn is it

    if RED == True:
        L2 = Label(root, text="   Red's Turn    ", fg='Black', background='green', font=("Arial
        L2.place(x=770, y=50)

    if BLUE == True:
        L2 = Label(root, text="   Blue's Turn   ", fg='Black', background='green', font=("Arial
        L2.place(x=770, y=50)

    if GREEN == True:
        L2 = Label(root, text="Green's Turn  ", fg='Black', background='green', font=("Arial",
        L2.place(x=770, y=50)

    if YELLOW == True:
        L2 = Label(root, text="Yellow's Turn", fg='Black', background='green', font=("Arial", 1
```

```
709         L2.place(x=770, y=50)
710 
711 
712 def roll():   #Rolling function that rolls a dice, goes again if its a six
713     global rollc, dice, dice1, dice2, TURN, rolls
714 
715     if TURN == True:
716 
717         rollc = rollc + 1
718         print("roll: ", rollc)
719 
720         if rollc == 1:
721             dice = random.randint(1, 6)
722             L1 = Label(root, text=dice, fg='Black', background='green', font=("Arial", 24, "bo
723             L1.place(x=800, y=200)
724             print("dice: ", dice)
725             rolls.append(dice)
726             if dice != 6:
727                 rollc = 0
728                 TURN = False
729 
730         if rollc == 2:
731             if dice == 6:
732                 dice1 = random.randint(1, 6)
733                 L3 = Label(root, text=dice1, fg='Black', background='green', font=("Arial", 24
734                 L3.place(x=800, y=250)
735                 rolls.append(dice1)
736                 if dice1 != 6:
737                     rollc = 0
738                     TURN = False
739 
740         if rollc == 3:
741             if dice1 == 6:
742                 dice2 = random.randint(1, 6)
743                 L4 = Label(root, text=dice2, fg='Black', background='green', font=("Arial", 24
744                 L4.place(x=800, y=300)
745                 rolls.append(dice2)
746                 rollc = 0
747                 TURN = False
748 
749 
750 def clear():        #clears all the variable prior to next player's turn
751     global nc, rolls, TURN, L1, L3, L4
752     nc = 0
753     del rolls[:]
754     TURN = True
755     L1 = Label(root, text="        ", fg='Black', background='green', font=("Arial", 24, "bold'
756     L1.place(x=800, y=200)
757     L3 = Label(root, text="        ", fg='Black', background='green', font=("Arial", 24, "bold'
758     L3.place(x=800, y=250)
759     L4 = Label(root, text="        ", fg='Black', background='green', font=("Arial", 24, "bold'
760     L4.place(x=800, y=300)
```

```
761         print("cleared")
762         turn()
763 
764 
765     def movecheck(r, rh, rb, la):       #Check if the player can make a move
766 
767         if (dice == 6 and dice1 == 6 and dice2 == 6):
768             return False
769 
770         win=True                                                #Checking if the game is won or t
771         for j in range(4):
772             if (r[j].x0 != rb[56].x) and (r[j].y0 != rb[56].y):
773                  win=False
774 
775         if win == True:                                     #If all gamepieces home, prints tha
776             print("YOU HAVE WON")
777             L2 = Label(root, text=(la + "Wins"), fg='Black', background='green', font=("Arial", 24
778             L2.place(x=770, y=500)
779             return False
780 
781         if win == False and dice != 6:                  #if its not a 6 and all game pieces insi
782             for i in range(len(r)):
783                 if(r[i].num != -1):
784                     (print("good hai"))
785                     return True
786             print("jani all in")
787             return False
788 
789     def kill(a,b,c,d,bh,ch,dh):   #function that determines if a gamepiece can be killed
790 
791         #if the game piece is not on a stop
792         if ((a[bb].x0 != box[1].x and a[bb].y0 != box[1].y) and (a[bb].x0 != box[14].x and a[bb].y0
793             (a[bb].x0 != box[9].x and a[bb].y0 != box[9].y) and (a[bb].x0 != box[22].x and a[bb].y0
794             (a[bb].x0 != box[27].x and a[bb].y0 != box[27].y) and (a[bb].x0 != box[35].x and a[bb]
795             (a[bb].x0 != box[40].x and a[bb].y0 != box[40].y) and (a[bb].x0 != box[48].x and a[bb]
796 
797 
798             #if the game piece of another color and its on the same block and it is not a double, a
799             for i in range (len(b)):
800                 if (b[i].x0 == a[bb].x and b[i].y0 == a[bb].y and (b[i].double == False)):
801                     b[i].x0 = bh[i].x
802                     b[i].y0 = bh[i].y
803                     b[i].x = bh[i].x + 25
804                     b[i].y = bh[i].y + 25
805                     b[i].num=-1
806                     b[i].swap()
807                     break
808 
809             for i in range (len(c)):
810                 if (c[i].x0 == a[bb].x and c[i].y0 == a[bb].y and (c[i].double == False)):
811                     c[i].x0 = ch[i].x
812                     c[i].y0 = ch[i].y
```

```
                c[i].x = ch[i].x + 25
                c[i].y = ch[i].y + 25
                c[i].num=-1
                c[i].swap()
                break

        for i in range (len(d)):
            if (d[i].x0 == a[bb].x and d[i].y0 == a[bb].y and (d[i].double == False)):
                d[i].x0 = dh[i].x
                d[i].y0 = dh[i].y
                d[i].x = dh[i].x + 25
                d[i].y = dh[i].y + 25
                d[i].num=-1
                d[i].swap()
                break

def doublecheck(a):        #makes a double is two or more gamepieces on top of another.

    for k in range (len(a)):
        a[k].double = False

    for i in range (len(a)):
        for j in range (len(a)):
            if (a[i].num == a[j].num) and (i != j):
                a[j].double = True
                a[i].double = True


turn()            #prints the "red player's turn" initially

button = Button(root, text="   ROLL   ", relief="raised", font=("Arial", 20),
                command=roll)  # call roll function evertime this button is clicked
button.place(x=805, y=120)

root.mainloop()
```