```
import tkinter as tk

from random import randrange

from tkinter import font


from settings import *


class LudoBoard:


    def __init__(self, master):

        self.canvas = tk.Canvas(master, width=Board.BOARD_WIDTH, height=Board.BOARD_HEIGHT)

        self.frame = tk.Frame(master, width=Board.PANEL_WIDTH, height=Board.PANEL_HEIGHT,
bg=Color.CYAN)

        self.Quit = tk.Button(master, text='QUIT', command=master.quit, relief=tk.RAISED, width=20,
height=2)

        self.title_bar = tk.Label(master, text=Text.HEADER, fg=Color.DEFAULT, bg=Color.CYAN, font=(None,
40), relief=tk.RAISED)

        self.status_bar = tk.Label(master, text=Text.MADE_BY, bd=1, relief=tk.SUNKEN)

    #whenevr the ludo class will be called, all the functions in it will be called

    #he below is the one to draw the rectangles on board

    def draw_rectangle(self, lx, ly, bx, by, color, width):

        self.canvas.create_rectangle(

            lx * Board.SQUARE_SIZE,

            ly * Board.SQUARE_SIZE,

            bx * Board.SQUARE_SIZE,

            by * Board.SQUARE_SIZE,

            fill=color,

            width = width
```

```
    )


def draw_polygon(self, x1, y1, x2, y2, color, width):
    self.canvas.create_polygon(
        x1 * Board.SQUARE_SIZE,
        y1 * Board.SQUARE_SIZE,
        Board.BOARD_WIDTH // 2,
        Board.BOARD_HEIGHT // 2,
        x2 * Board.SQUARE_SIZE,
        y2 * Board.SQUARE_SIZE,
        fill=color,
        width=width
    )
#the four circle shaped tokens
def draw_circle(self, x1, y1, x2, y2, color):
    self.canvas.create_oval(
        x1 * Board.SQUARE_SIZE,
        y1 * Board.SQUARE_SIZE,
        x2 * Board.SQUARE_SIZE,
        y2 * Board.SQUARE_SIZE,
        fill=color
    )


def path(self):
```

```
    self.canvas.place(x=20, y=80)
#this is to set the position of paths for 4 tokens namely red, yellow, green and blue
    for i in range(6, 9):
        for j in range(15):
            if (j not in range(6, 9) and
                i != 7 or j == 0 or j == 14
                ):
                self.draw_rectangle(i + 0.5, j + 0.5, i + 1.5, j + 1.5, '', 1)
                self.draw_rectangle(j + 0.5, i + 0.5, j + 1.5, i + 1.5, '', 1)
            else:
                if j < 6:
                    self.draw_rectangle(i + 0.5, j + 0.5, i + 1.5, j + 1.5, Color.YELLOW, 1)
                    self.draw_rectangle(j + 0.5, i + 0.5, j + 1.5, i + 1.5, Color.GREEN, 1)
                elif j > 8:
                    self.draw_rectangle(i + 0.5, j + 0.5, i + 1.5, j + 1.5, Color.RED, 1)
                    self.draw_rectangle(j + 0.5, i + 0.5, j + 1.5, i + 1.5, Color.BLUE, 1)

    for i, j in Board.POSITIVE_V:
        if i > j:
            self.draw_rectangle(i + 0.5, j + 0.5, i + 1.5, j + 1.5, Color.YELLOW, 1)
        else:
            self.draw_rectangle(i + 0.5, j + 0.5, i + 1.5, j + 1.5, Color.RED, 1)

        self.draw_circle(i + 0.7, j + 0.7, i + 1.3, j + 1.3, Color.GRAY)
    for j, i in Board.POSITIVE_H:
```

```
        if i > j:
            self.draw_rectangle(j + 0.5, i + 0.5, j + 1.5, i + 1.5, Color.GREEN, 1)
        else:
            self.draw_rectangle(j + 0.5, i + 0.5, j + 1.5, i + 1.5, Color.BLUE, 1)
        self.draw_circle(j + 0.7, i + 0.7, j + 1.3, i + 1.3, Color.GRAY)

#this function will handle the desgin of home
def home(self):

    for i, j in Board.POINTS:
        #this is the rectangle that we drew using the function draw_rectangle()
        #we made four different rectangles for 4 tokens
        if i == 0 and j == 0:
            self.draw_rectangle(i*9 + 0.5, j*9 + 0.5, i*9 + 6.5, j*9 + 6.5, Color.GREEN, 3)
        elif i == 0 and j == 1:
            self.draw_rectangle(i*9 + 0.5, j*9 + 0.5, i*9 + 6.5, j*9 + 6.5, Color.RED, 3)
        elif i == 1 and j == 0:
            self.draw_rectangle(i*9 + 0.5, j*9 + 0.5, i*9 + 6.5, j*9 + 6.5, Color.YELLOW, 3)
        else:
            self.draw_rectangle(i*9 + 0.5, j*9 + 0.5, i*9 + 6.5, j*9 + 6.5, Color.BLUE, 3)

        self.draw_rectangle(i*9 + 1.25, j*9 + 1.25, i*9 + 5.75, j*9 + 5.75, Color.DEFAULT, 0)

    #this is to set the postion of each rectangle that we made for home
    for i, j in Board.POINTS:
```

```
#for placement of green rectangle
if i == 0 and j == 0:
    self.draw_rectangle(i*9 + 1.65, j*9 + 1.65, i*9 + 3.3, j*9 + 3.3, Color.GREEN, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 3.65, i*9 + 5.3, j*9 + 5.3, Color.GREEN, 0)
    self.draw_rectangle(i*9 + 1.65, j*9 + 3.65, i*9 + 3.3, j*9 + 5.3, Color.GREEN, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 1.65, i*9 + 5.3, j*9 + 3.3, Color.GREEN, 0)
#for placement of green rectangle
elif i == 0 and j == 1:
    self.draw_rectangle(i*9 + 1.65, j*9 + 1.65, i*9 + 3.3, j*9 + 3.3, Color.RED, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 3.65, i*9 + 5.3, j*9 + 5.3, Color.RED, 0)
    self.draw_rectangle(i*9 + 1.65, j*9 + 3.65, i*9 + 3.3, j*9 + 5.3, Color.RED, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 1.65, i*9 + 5.3, j*9 + 3.3, Color.RED, 0)
#for placement of green rectangle
elif i == 1 and j == 0:
    self.draw_rectangle(i*9 + 1.65, j*9 + 1.65, i*9 + 3.3, j*9 + 3.3, Color.YELLOW, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 3.65, i*9 + 5.3, j*9 + 5.3, Color.YELLOW, 0)
    self.draw_rectangle(i*9 + 1.65, j*9 + 3.65, i*9 + 3.3, j*9 + 5.3, Color.YELLOW, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 1.65, i*9 + 5.3, j*9 + 3.3, Color.YELLOW, 0)
#for placement of green rectangle
else:
    self.draw_rectangle(i*9 + 1.65, j*9 + 1.65, i*9 + 3.3, j*9 + 3.3, Color.BLUE, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 3.65, i*9 + 5.3, j*9 + 5.3, Color.BLUE, 0)
    self.draw_rectangle(i*9 + 1.65, j*9 + 3.65, i*9 + 3.3, j*9 + 5.3, Color.BLUE, 0)
    self.draw_rectangle(i*9 + 3.65, j*9 + 1.65, i*9 + 5.3, j*9 + 3.3, Color.BLUE, 0)
```

```
    self.draw_polygon(6.5, 6.5, 6.5, 9.5, Color.GREEN, 1)
    self.draw_polygon(6.5, 6.5, 9.5, 6.5, Color.YELLOW, 1)
    self.draw_polygon(9.5, 9.5, 6.5, 9.5, Color.RED, 1)
    self.draw_polygon(9.5, 9.5, 9.5, 6.5, Color.BLUE, 1)



 def create_panel(self):
    self.frame.place(x=700, y=80)
    self.Quit.place(x=910, y=620)
    self.title_bar.pack(side=tk.TOP, fill=tk.X)
    self.status_bar.pack(side=tk.BOTTOM, fill=tk.X)

 #this will call the above defined functions
 def create(self):
    self.path()
    self.home()
    self.create_panel()

 def get_canvas(self):
    return self.canvas

def get_frame(self):
  return self.frame
```