```python
# Python code for catching the ball game

# importing suitable packages
from tkinter import Tk,Button,Label
from tkinter import Canvas
from random import randint

# defining Tk from Tkinter
root = Tk()
root.title("Catch the ball Game")
root.resizable(False,False)

# for defining the canvas
canvas = Canvas(root, width=600, height=60
0)
canvas.pack()

# variable for the vertical distance
# travelled by ball
limit = 0

# variable for horizontal distance
# of bar from x-axis
dist = 5

# variable for score
score = 0
```

```
# Class for the Creating and moving ball
class Ball:

    # for creation of ball on the canvas
    def __init__(self, canvas, x1, y1, x2,
 y2):
        self.x1 = x1
        self.y1 = y1
        self.x2 = x2
        self.y2 = y2
        self.canvas = canvas

        # for creation of ball object
        self.ball = canvas.create_oval(sel
f.x1, self.y1, self.x2, self.y2,

      fill = "red",tags = 'dot1')

    # for moving the ball
    def move_ball(self):

        # defining offset
        offset = 10
        global limit
        # checking if ball lands ground or
 bar
        if limit >= 510:
            global dist,score,next

            # checking that ball falls on
the bar
            if(dist -
 offset <= self.x1 and
            dist + 40 + offset >= self.x2)
:

                # incrementing the score
                score += 10

                # dissappear the ball
                canvas.delete('dot1')

                # calling the function for
 again
                # creation of ball object
                ball_set()

            else:
                # dissappear the ball
                canvas.delete('dot1')
                bar.delete_bar(self)
                # display the score
                score_board()
            return

        # incrementing the vertical distan
ce
        # travelled by ball by deltay
        limit += 1

        # moving the ball in vertical dire
ction
        # by taking x=0 and y=deltay
        self.canvas.move(self.ball,0,1)

        # for continuous moving of ball ag
ain call move_ball
        self.canvas.after(10,self.move_bal
l)

# class for creating and moving bar
class bar:

    # method for creating bar
    def __init__(self,canvas,x1,y1,x2,y2):

        self.x1 = x1
        self.y1 = y1
        self.x2 = x2
        self.y2 = y2
        self.canvas = canvas

        # for creating bar using create_re
ctangle
        self.rod=canvas.create_rectangle(s
elf.x1, self.y1, self.x2, self.y2,

      fill="yellow",tags='dot2')

    # method for moving the bar
    def move_bar(self,num):
        global dist

        # checking the forward or backward
 button
        if(num == 1):

            # moving the bar in forward di
rection by
            # taking x-
axis positive distance and
            # taking vertical distance y=0
            self.canvas.move(self.rod,20,0
)

            # incrementing the distance of
 bar from x-axis
            dist += 20
        else:

            # moving the bar in backward d
irection by taking x-axis
            # negative distance and taking
 vertical distance y=0
            self.canvas.move(self.rod,-
20,0)

            # decrementing the distance of
 bar from x-axis
            dist-=20

    def delete_bar(self):
        canvas.delete('dot2')


# Function to define the dimensions of the
 ball
def ball_set():
    global limit
    limit=0

    # for random x-axis distance from
    # where the ball starts to fall
    value = randint(0,570)

    # define the dimensions of the ball
    ball1 = Ball(canvas,value,20,value+30,
50)

    # call function for moving of the ball

    ball1.move_ball()

# Function for displaying the score
# after getting over of the game
def score_board():
    root2 = Tk()
    root2.title("Catch the ball Game")
    root2.resizable(False,False)
    canvas2 = Canvas(root2,width=300,heigh
t=300)
    canvas2.pack()

    w = Label(canvas2,text="\nOOPS...GAME
IS OVER\n\nYOUR SCORE = "
```

```
  + str(score) + "\n\n")
    w.pack()

    button3 = Button(canvas2, text="PLAY A
GAIN", bg="green",
                        command=lambda:pla
y_again(root2))
    button3.pack()

    button4 = Button(canvas2,text="EXIT",b
g="green",
                    command=lambda:exit_ha
ndler(root2))
    button4.pack()

# Function for handling the play again req
uest
def play_again(root2):
    root2.destroy()
    main()

# Function for handling exit request
def exit_handler(root2):
    root2.destroy()
    root.destroy()
```

```
# Main function
def main():
    global score,dist
    score = 0
    dist = 0

    # defining the dimensions of bar
    bar1=bar(canvas,5,560,45,575)

    # defining the text,colour of buttons 
and
    # also define the action after click o
n
    # the button by calling suitable metho
ds
    button = Button(canvas,text="==>", bg=
"green",
                    command=lambda:bar1.mo
ve_bar(1))

    # placing the buttons at suitable loca
tion on the canvas
    button.place(x=300,y=580)

    button2 = Button(canvas,text="<==",bg=
"green",
                    command=lambda:bar1.mo
ve_bar(0))
    button2.place(x=260,y=580)

    # calling the function for defining
    # the dimensions of ball
    ball_set()
    root.mainloop()

# Driver code
if(__name__=="__main__"):
    main()
```