



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

First Semester Python Project – Ball Paddle game

Soumyadipto De – PES1202100880

Pranav Sridhar – PES1202102194

Table Of Contents:

[Abstract:](#) Page 2

[Introduction:](#) Page 2

[Design & Implementation:](#) [Main Menu Code:](#) Page 3

[Singleplayer Code:](#) Page 4 - 7

[Multiplayer Code:](#) Page 8 – 12

[Result:](#) Page 12 – 15

[Testing:](#) Page 15

[Conclusion:](#) Page 16

[References:](#) Page 16

Abstract:

The goal of this project is to create a simple and enjoyable game using our knowledge of programming. Python is a powerful programming language, used for various purposes.

In our project we have used the module – ‘tkinter’ to create the interface and the time module to configure the movement of objects in the window.

As more variety increases the quality of any game, we have made two modes to play the game- single player and multiplayer, where two players can play simultaneously.

This project increased our knowledge of Python programming knowledge and helped us discover more ways to program, and broadened our view on the scope of computer science and programming.

Introduction:

Our project – ‘Paddle Ball Game’ – is a game that is created to be as simple game, yet as enjoyable as possible. The objective of the game is to keep the ‘ball’ on the field using the ‘paddle’.

The ‘ball’ is a class created in tkinter that changes speed in the window using the time module. The ‘paddle’ is a class that is controlled by the user, using fairly easy controls.

In the single player version of the game, there is a single paddle to be controlled and the game ends when the ball reaches the bottom of the window. In the multiplayer version, there are two paddles on either side of the window, and the ball starts of in the center of the screen and is to be blocked off the sides of the window. If the ball touches either wall, the opponent receives a point and the game continues

The modules used for this project are:

tkinter: Used to create the interface, the objects, classes, message boxes, etc.

time: Used to change the frames in the window to create a smooth animation of the ball and paddle.

random: Used to randomly vary the velocity and colour of the ball to add an element of surprise.

subprocess: Used to call different programs.

Design & Implementation:

The code for the menu:

```
import os
import subprocess
from tkinter import *
from tkinter import messagebox
import time

w=Tk()
canvas=Canvas(w,width=500,height=300)
w.title("Welcome message")
w.resizable(0,0)
canvas.create_text(250,100,fill='orange',text="Welcome to Ball-Paddle game",
                  font=('Bernard MT',20))
canvas.pack()

ob=Tk()
ob.title("Team-Project")
lbl=Label(ob,text="Choose mode of play:",font=('Arial',20)).grid()

def play1():
    messagebox.showinfo(title="Rules",message=''Move the paddle bar using left
and right arrow keys.
Prevent the ball from hitting bottom of the screen.
Click on the gamescreen to get started'')

    subprocess.call('paddle1.py',shell=True)

def play2():
    messagebox.showinfo(title="Rules",message=''Keys to press
Player1:E and X
Player2: Up and Down arrow keys
Prevent the ball from getting past your paddle.Click on gamescreen to get
started'')
    subprocess.call('ballgame_final.py',shell=True)

bt1=Button(ob,text="Player 1",command=play1,font=('Arial',16)).grid()

bt2=Button(ob,text="Player 2",command=play2,font=('Arial',16)).grid()

ob.mainloop()
```

The code for single player game:

```
from tkinter import *
from tkinter import messagebox
import random
import time
import os
import sys
import subprocess

a=Tk()
canvas=Canvas(a,width=500,height=500)
a.title("Ball-Paddle")
a.resizable(0,0)
canvas.pack()
a.update()

count=0

def score():
    canvas.itemconfig(cscore,text='Score:'+str(count))

class Ball:
    def __init__(self,canvas,paddle,color):

        self.canvas=canvas
        self.paddle=paddle
        self.circ=canvas.create_oval(10,10,27,27,fill=color)
        self.canvas.move(self.circ,220,220)

        startx=[-5,-4,-3,-2,-1,1,2,3,4,5]
        self.x=random.choice(startx)
        self.y=-3
```

```
self.cht=self.canvas.winfo_height()
self.cwd=self.canvas.winfo_width()
self.hitbottom=False

def hit_paddle(self,pos):
    paddle_pos=self.canvas.coords(self.paddle.pad)

    if pos[2]>=paddle_pos[0] and pos[0]<=paddle_pos[2]:
        if pos[3]>=paddle_pos[1] and pos[3]<=paddle_pos[3]:

            self.x+=self.paddle.x

            return True
    return False

def draw(self,v):
    self.canvas.move(self.circ,self.x,self.y)
    pos=self.canvas.coords(self.circ)

    if pos[0]<=0:
        canvas.itemconfig(self.circ,fill='blue')
        self.x=3
    if pos[2]>=self.cwd:
        self.x=-3
        canvas.itemconfig(self.circ,fill='yellow')
    if pos[1]<=0:
        self.y=v
        canvas.itemconfig(self.circ,fill='orange')
    if pos[3]>=self.cht:
        self.hitbottom=True
    if self.hit_paddle(pos)==True:
        self.y=-v
        canvas.itemconfig(self.circ,fill='red')
        global count
        count+=1
        score()

class Paddle:
```

```
def __init__(self, canvas, color):
    self.canvas = canvas
    self.pad = canvas.create_rectangle(100, 100, 180, 110, fill=color)
    self.canvas.move(self.pad, 200, 300)
    self.x = 0
    self.cwd = self.canvas.winfo_width()
    self.start = False
    self.canvas.bind_all('<KeyPress-Left>', self.turn_left)
    self.canvas.bind_all('<KeyPress-Right>', self.turn_right)
    self.canvas.bind_all('<KeyRelease>', self.stop)
    self.canvas.bind_all('<Button-1>', self.st)

def draw(self):
    self.canvas.move(self.pad, self.x, 0)
    self.pos = self.canvas.coords(self.pad)
    if self.pos[0] <= 0:
        self.x = 0
    if self.pos[2] >= self.cwd:
        self.x = 0

def turn_left(self, event):
    if self.pos[0] >= 0:
        self.x = -5

def turn_right(self, event):
    if self.pos[2] <= self.cwd:
        self.x = 5

def st(self, event):
    self.start = True

def stop(self, event):
    self.x = 0

def msgprompt():
    r = messagebox.askyesno('Replay', 'Do you want a replay?')
    if r:
        a.destroy()
        subprocess.call(sys.executable + ' "' +
```

```
os.path.realpath(__file__) + '')

else:
    a.destroy()

paddle=Paddle(canvas,'blue')
ball=Ball(canvas,paddle,'green')
gmover=canvas.create_text(250,270,state='hidden',
                           fill='purple',font=('Algerian',20))
cscore=canvas.create_text(250,20,font=('Footlight MT',16))

score()
i=3

while 1:
    if ball.hitbottom==False and paddle.start==True:
        ball.draw(i)
        paddle.draw()
    if ball.hitbottom==True:
        time.sleep(1)
        myimage=PhotoImage(file='D:\\download.gif')
        canvas.create_image(120,0,anchor=NW,image=myimage)

        canvas.itemconfig(gmover,state='normal',text='Game over\\n'
                        +'Score: '+str(count))

        break

    i+=0.008
    a.update_idletasks()
    a.update()
    time.sleep(0.01)

msgprompt()
a.mainloop()
```


The code for multiplayer:

```
from tkinter import *
import time as t
import random

count1=0;count2=0

def score1():
    canvas.itemconfig(cscore1,text='Player1 Score: '+str(count1))

def score2():
    canvas.itemconfig(cscore2,text='Player2 Score: '+str(count2))

class Ball:
    def __init__(self,canvas,paddle1,paddle2,color):
        self.canvas=canvas
        self.paddle1=paddle1
        self.paddle2=paddle2
        self.circle=canvas.create_oval(10,10,30,30,fill=color)

        self.canvas.move(self.circle,220,210)
        self.x=3
        self.y=3
        self.height=self.canvas.winfo_height()
        self.width=self.canvas.winfo_width()
        self.hitside=False

    def reset(self):
        self.canvas.delete(self.circle)
        self.__init__(canvas,paddle1,paddle2,'green')

    def hitpaddle1(self,pos):
        paddle_pos=self.canvas.coords(self.paddle1.p)
```

```
if pos[3]>=paddle_pos[1] and pos[1]<=paddle_pos[3]:
    if pos[0]>=paddle_pos[0] and pos[0]<=paddle_pos[2]:
        self.y+=self.paddle1.y
        return True
    return False

def hitpaddle2(self,pos):
    paddle_pos=self.canvas.coords(self.paddle2.p)

    if pos[3]>=paddle_pos[1] and pos[1]<=paddle_pos[3]:
        if pos[2]>=paddle_pos[0] and pos[2]<=paddle_pos[2]:
            self.y+=self.paddle2.y
            return True
        return False

def draw(self,v):
    self.canvas.move(self.circle,self.x,self.y)
    pos=self.canvas.coords(self.circle)

    if pos[0]<=0:
        self.x=v
        self.hitside=True
        global count2
        count2+=1
        score2()

    if pos[1]<=0:
        self.y=v
        canvas.itemconfig(self.circle,fill="green")

    if pos[2]>=self.width:
        self.x=-v
        self.hitside=True
        global count1
        count1+=1
        score1()

    if pos[3]>=self.height:
        self.y=-v
        canvas.itemconfig(self.circle,fill="yellow")
```

```
if self.hitpaddle1(pos)==True:
    self.x=3
    canvas.itemconfig(self.circle,fill="brown")

if self.hitpaddle2(pos)==True:
    self.x=-3
    canvas.itemconfig(self.circle,fill="pink")

class Paddle:
    def __init__(self,canvas,paddle,color):
        self.canvas=canvas
        self.paddle=paddle
        self.p=canvas.create_rectangle(10,10,20,110,fill=color)
        self.y=0
        self.cht=self.canvas.winfo_height()

        if self.paddle=='paddle1':
            self.canvas.bind_all('<KeyPress-e>',self.up)
            self.canvas.bind_all('<KeyPress-x>',self.down)

        if self.paddle=='paddle2':
            self.canvas.bind_all('<KeyPress-Up>',self.up)
            self.canvas.bind_all('<KeyPress-Down>',self.down)

    def move(self,x,y):
        self.canvas.move(self.p,x,y)

    def draw(self):
        self.canvas.move(self.p,0,self.y)
        self.pospad=self.canvas.coords(self.p)

        if self.pospad[1]<=0:
```

```
        self.y=0
    if self.pospad[3]>=self.cht:
        self.y=0

    def up(self,event):
        if self.pospad[1]>=0:
            self.y=-6

    def down(self,event):
        if self.pospad[3]<=self.cht:
            self.y=6

tk=Tk()
canvas=Canvas(tk,width=600,height=600)
tk.title("air hockey")
tk.resizable(0,0)
canvas.pack()
tk.update()

paddle1=Paddle(canvas,'paddle1',"red")
paddle1.move(60,220)

paddle2=Paddle(canvas,'paddle2',"blue")
paddle2.move(520,220)

ball=Ball(canvas,paddle1,paddle2,"green")

cscore1=canvas.create_text(100,20,font=('Footlight MT',10))
cscore2=canvas.create_text(400,20,font=('Footlight MT',10))

score1()
score2()

def startgame(event):
    i=3
    while 1:
        i+=0.005
        if ball.hitside==True:
            t.sleep(1)
```

```
        ball.reset()
        pass

    if ball.hitside==False:
        ball.draw(i)
        paddle1.draw()
        paddle2.draw()

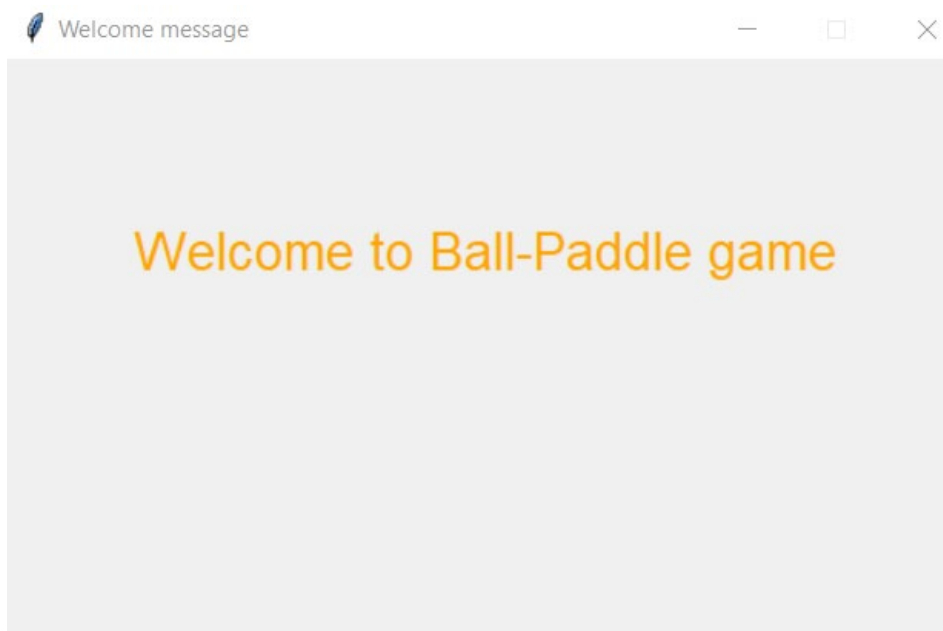
    tk.update_idletasks()
    tk.update()
    t.sleep(0.01)

canvas.bind_all('<Button-1>',startgame)

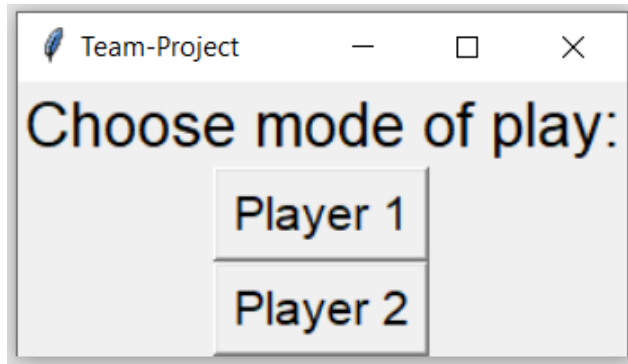
tk.mainloop()
```

Results:

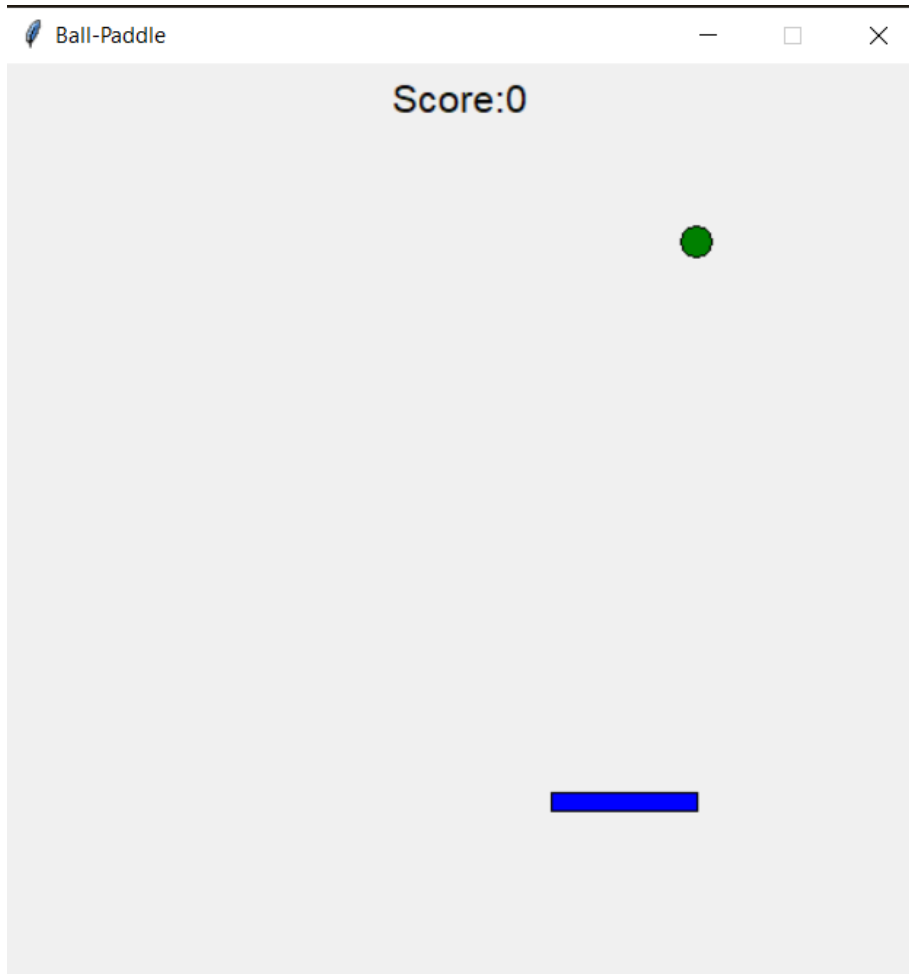
Screenshots



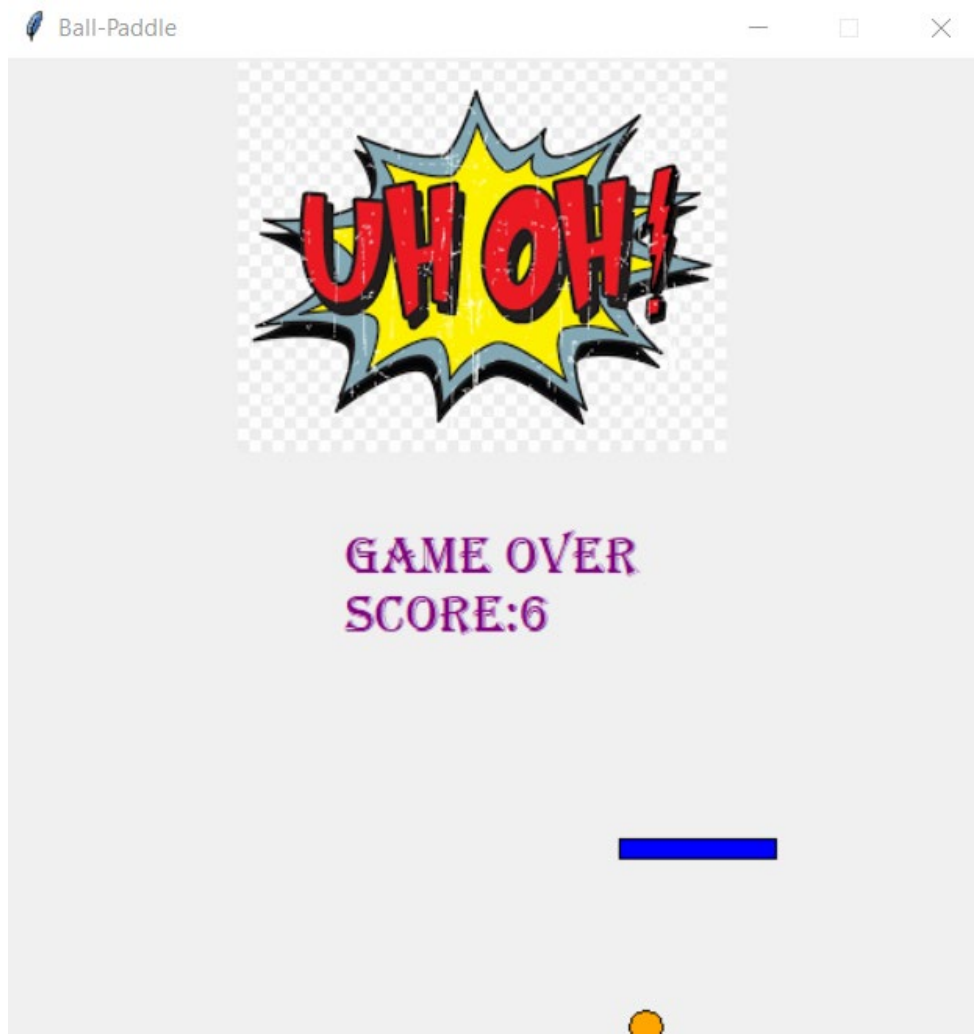
Welcome Screen



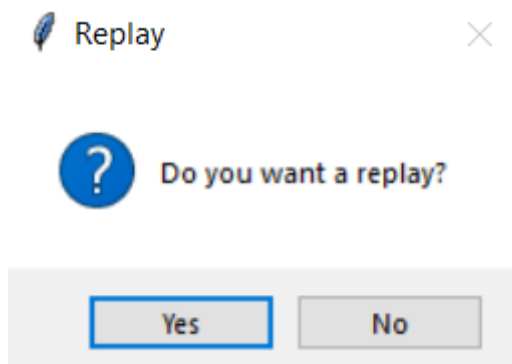
Selection of game mode

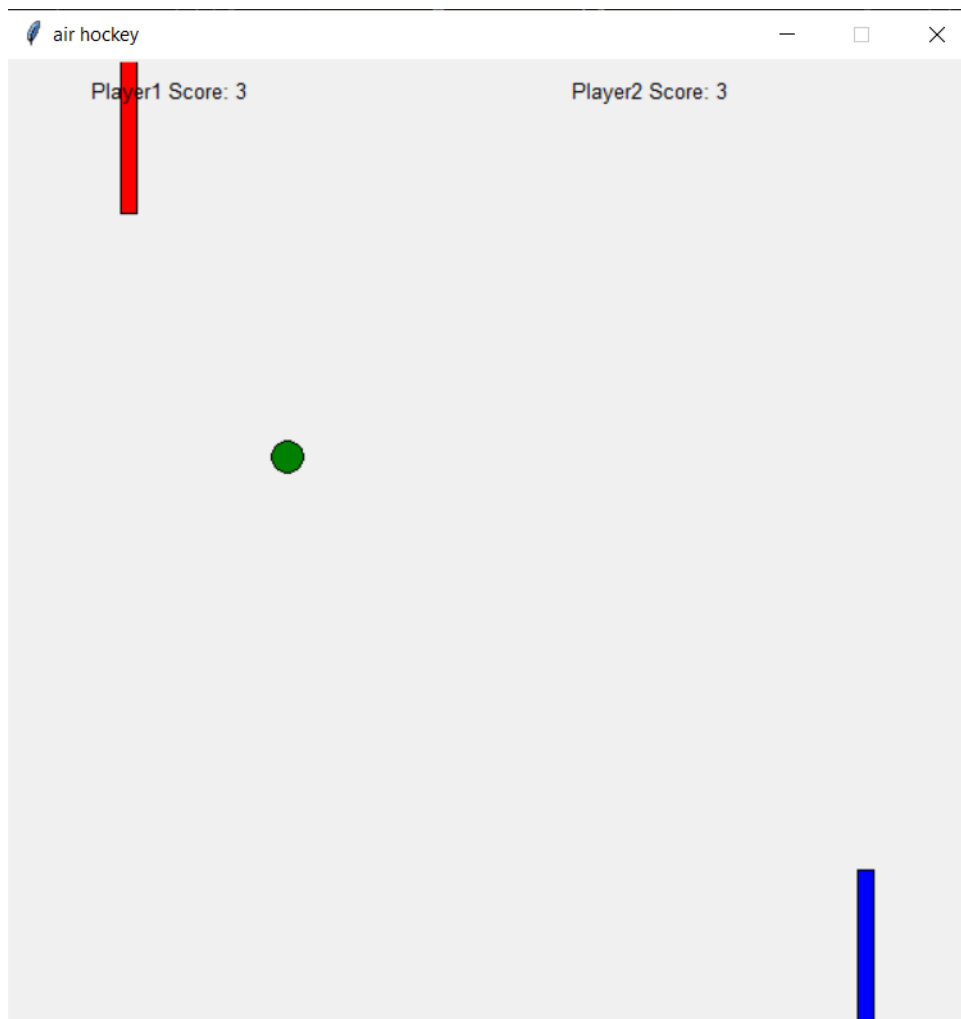


Single Player Game



Game Over Screen





Multiplayer Game

Testing:

The Main Menu opens with a selection and welcome screen.

The two game modes can be selected here.

In the single player game, when the ball touches the bottom of the screen, a game over screen appears accompanied by a message box, which has options to restart or quit the game.

In the multiplayer game, when ball touches the sides of the screen, the score updates for the opponent and the ball recentres.

Conclusion:

This project has enhanced our knowledge on Python programming language by providing an insight into several new functions and modules. Further, we could try to work on the same program using different modules such as pygame. We wish to make the game more interactive and developed by increasing visual and audio appeal and introducing more obstacles that will keep the player engaged

References:

Python for Kids – by *Jason. R. Briggs*

[GeeksforGeeks](#)

[TutorialsPoint](#)

[StackOverflow](#)