

Game development using Python

**Project report in partial fulfillment of the requirement for the award of the degree of
Bachelor of Technology**

**In
Computer Science And Engineering**

Submitted By

Arya Nath	University Roll No. 12020009029009
Abhijit Maity	University Roll No. 12020009023012
Mohan Kumar Jana	University Roll No. 12020009023009
Arigna Adhikari	University Roll No. 12020009022033
Sayak Sarkar	University Roll No.12020009022148
Suprotim Karmakar	University Roll No.12020009022174
Bratin Das	University Roll No. 12020009022127
Sagor Mondal	University Roll No. 12020009001241

Under the guidance of

Prof. Dr. Maumita Chakraborty

&

Prof. Prasenjit Kumar Das

Department of **Computer Science and Engineering**



UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled Game development using Python submitted by Arya Nath (University Roll No. 12020009029009), Abhijit Maity (University Roll No. 12020009023012), Mohan Kumar Jana (University Roll No. 12020009023009), Arigna Adhikari (University Roll No. 12020009022033), Sayak Sarkar (University Roll No. 12020009022148), Suprotim Karmakar (University Roll No. 12020009022174), Bratin Das (University Roll No. 12020009022127) and Sagor Mondal (University Roll No. 12020009001241) students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science Engineering, is a bonafide work carried out by them under the supervision and guidance of Prof. Dr. Maumita Chakraborty & Prof. Prasenjit Kumar Das during 3rd Semester of academic session of 2020-2024. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

Prof. Dr. Maumita Chakraborty

Project Guide

HOD, Department of Computer Science and Technology

UEM, Kolkata

Prof. Prasenjit Kumar Das

Project Guide and Assistant Professor

Department of Computer Science and Technology

UEM, Kolkata

Prof. Dr. Maumita Chakraborty

HOD, Department of Computer Science and Technology

UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. Dr. Maumita Chakraborty and Prof. Prasenjit Kumar Das of the Department of Computer Science Engineering, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Dr. Maumita Chakraborty HOD, Computer Science and Technology, UEM, Kolkata and all other departmental faculties for their ever-present assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Arya Nath

Abhijit Maity

Mohan Kumar Jana

Arigna Adhikari

Sayak Sarkar

Suprotim Karmakar

Bratin Das

Sagor Mondal

TABLE OF CONTENTS

ABSTRACT.....	5
CHAPTER – 1: INTRODUCTION.....	6
CHAPTER – 2: Project Details.....	7
CHAPTER – 3: Main Code.....	8
CHAPTER – 4: OUTPUT.....	13
CHAPTER – 6 : CONCLUSION & FUTURE SCOPE.....	14
BIBLIOGRAPHY	15

ABSTRACT

This Pong Game has been designed using Python Pygame. It has single player as well as double player option. It provides users a new experience that is different from the usual pong games available. It is a freeware game, so users need not pay any amount to play it. A pong game consists of two pads on both sides and a ball, and the player has to make sure that he hits the ball with the pad without missing it to win the game and score more.

INTRODUCTION

Pong, groundbreaking electronic game released in 1972 by the American game manufacturer Atari, Inc. One of the earliest video games, Pong became wildly popular and helped launch the video game industry. The original Pong consisted of two paddles that players used to volley a small ball back and forth across a screen.

Project Details

PING PONG

This game project will be an homage to the classic Pong. In the original version, a small pip (the ball) is hit left and right by "paddles" vertically moving paddles controlled by two players. The ball would bounce off the top and bottom of the display region. If the ball passed a player's paddle, a point would be added to the opponent's score. First to 11 won the game. The player winning the point would also win the serve which meant that the ball would appear in front of that player's paddle and move toward the opponents side of the display. The angle of deflection of the ball from the paddle could be controlled by the speed at which the paddle was moving when it hit the ball. The speed of the ball slowly increased during a volley, but would return to the initial slower speed at the beginning of the next server. Sounds included a blip for the paddle striking the ball, a sound for a missed return and an end of game sound

Big Ideas

- Use steps for movement
- No drawing in update part
- Use key down and up for input
- Paddle inherits BlockSprite
- Program end \neq game end
- two booleans: running, gameOver
- 'fancy' collision detection:
- sprite position + direction of movement

Main Code

1) Background:

```
wn = turtle.Screen()

wn.title("Pong")

wn.bgcolor("black")

wn.setup(width=800, height=600)

wn.tracer(0)
```

2) Score initialisation:

```
score_a = 0

score_b = 0
```

3) To create Paddle:

```
# Paddle A

paddle_a = turtle.Turtle()

paddle_a.speed(0)

paddle_a.shape("square")

paddle_a.color("white")

paddle_a.shapesize(stretch_wid=5,stretch_len=1)

paddle_a.penup()

paddle_a.goto(-350, 0)


# Paddle B

paddle_b = turtle.Turtle()
```



```
paddle_b.speed(0)

paddle_b.shape("square")

paddle_b.color("white")

paddle_b.shapesize(stretch_wid=5,stretch_len=1)

paddle_b.penup()

paddle_b.goto(350, 0)
```

4) To create Ball:

```
# Ball

ball = turtle.Turtle()

ball.speed(0)

ball.shape("square")

ball.color("white")

ball.penup()

ball.goto(0, 0)

ball.dx = 2

ball.dy = 2
```

5) To move Paddle without drawing line with function (Pen) :

```
# Pen

pen = turtle.Turtle()

pen.speed(0)

pen.shape("square")

pen.color("white")

pen.penup()

pen.hideturtle()

pen.goto(0, 260)

pen.write("Player A: 0 Player B: 0", align="center", font=("Courier", 24, "normal"))

def paddle_a_up():
```

```
# Functions

def paddle_a_up():

    y = paddle_a.ycor()

    y += 20

    paddle_a.sety(y)


def paddle_a_down():

    y = paddle_a.ycor()

    y -= 20

    paddle_a.sety(y)


def paddle_b_up():

    y = paddle_b.ycor()

    y += 20

    paddle_b.sety(y)


def paddle_b_down():

    y = paddle_b.ycor()

    y -= 20

    paddle_b.sety(y)
```

6) Keyboard Binding (input through Keyboard):

```
# Keyboard bindings

wn.listen()

wn.onkeypress(paddle_a_up, "w")

wn.onkeypress(paddle_a_down, "s")

wn.onkeypress(paddle_b_up, "Up")

wn.onkeypress(paddle_b_down, "Down")
```

7) Move the Ball:

```
ball.setx(ball.xcor() + ball.dx)
```

```
ball.sety(ball.ycor() + ball.dy)
```

8) Scoring:

```
# Top and bottom
```

```
if ball.ycor() > 290:
```

```
    ball.sety(290)
```

```
    ball.dy *= -1
```

```
    os.system("afplay bounce.wav&")
```

```
elif ball.ycor() < -290:
```

```
    ball.sety(-290)
```

```
    ball.dy *= -1
```

```
    os.system("afplay bounce.wav&")
```

```
# Left and right
```

```
if ball.xcor() > 350:
```

```
    score_a += 1
```

```
    pen.clear()
```

```
    pen.write("Player A: {} Player B: {}".format(score_a, score_b), align="center", font=("Courier", 24, "normal"))
```

```
    ball.goto(0, 0)
```

```
    ball.dx *= -1
```

```
elif ball.xcor() < -350:
```

```
    score_b += 1
```

```
    pen.clear()
```

```
pen.write("Player A: {} Player B: {}".format(score_a, score_b), align="center", font=("Courier", 24, "normal"))
```

```
ball.goto(0, 0)
```

```
ball.dx *= -1
```

9) Paddle and Ball Collision:

```
if ball.xcor() < -340 and ball.ycor() < paddle_a.ycor() + 50 and ball.ycor() > paddle_a.ycor() - 50:
```

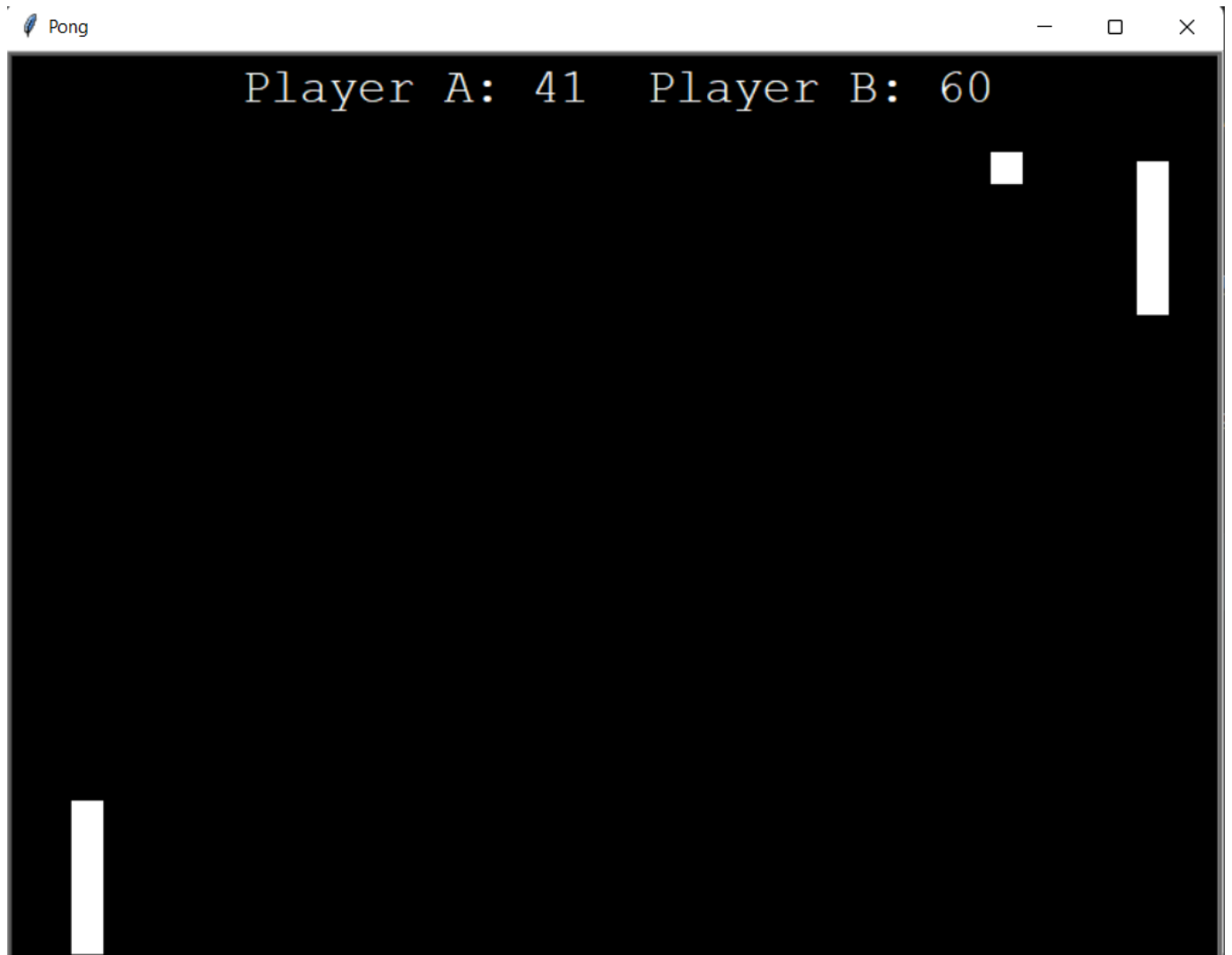
```
    ball.dx *= -1
```

```
    os.system("afplay bounce.wav&")
```

```
elif ball.xcor() > 340 and ball.ycor() < paddle_b.ycor() + 50 and ball.ycor() > paddle_b.ycor() - 50:
```

```
    ball.dx *= -1 os.system("afplay bounce.wav&")
```

OUTPUT



CONCLUSSION AND FUTURE SCOPE

The game software is good structured and easy to understand, but there could probably be improvements that are not done. Some improvements that would need small adjustment to the hardware is adding support for one additional ball and adding difficulty level. This would not increase the complexity of neither software nor hardware by much, but changes would have to be made.

BIBLIOGRAPHY

Websites:

- 1) Wikipedia: <https://www.wikipedia.org/>
- 2) YouTube: <https://www.youtube.com/>

Books:

- 1) Core Python Programming by Dr. R. Nageswara Rao
- 2) Python, PyGame, and Raspberry Pi Game Development by Sloan Kelly