**NAGARJUNA VIDYANIKETAN**



**ACADEMIC YEAR : 2020-21**

**PROJECT REPORT ON**

**‘PONG’**

**ROLL NO : 29**

**NAME : AYUSH SURESH BHARSHETTY**

**CLASS : XII A**

**SUBJECT : COMPUTER SCIENCE**

**SUB CODE : 083**

**PROJECT GUIDE: Mr NARASA REDDY**

**PGT (CS)**

**NAGARJUNA VIDYANIKETAN**

**#104, IVRI ROAD**

**BANGALORE**

## **NAGARJUNA VIDYANIKETAN**



# **CERTIFICATE**

This is to certify that Cadet **Ayush Suresh Bharshetty** of CBSE,

Roll No: 29 has successfully completed the project Work entitled **"Pong"**in the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in **NAGARJUNA VIDYANIKETAN** on\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**(Narasa Reddy)**

PGT Comp Sci

***Master IC***

**Examiner:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature:

Date:

| **TABLE OF CONTENTS [ T O C ]** | | | |
| --- | --- | --- | --- |
| **SER** | **DESCRIPTION** | **PAGE NO** | |
| 01 | ACKNOWLEDGEMENT | 4 | |
| 02 | INTRODUCTION TO PYTHON | 5 | |
| 03 | HISTORY OF PYTHON | 6 | |
| 04 | INTRODUCTION TO THE PROJECT | 7 | |
| 05 | OBJECTIVES OF THE PROJECT | 7 | |
| 06 | ALGORITHM OF THE PROJECT | 8 | |
| 07 | FLOW CHART | 9 | |
| 08 | SOURCE CODE | 10 - 32 | |
| 09 | SCOPE OF THE PROJECT | 33 | |
| 10 | THE EXISTING SYSTEM | 34 | |
| 11 | SCREENSHOTS OF PROGRAM EXECUTION | 35 - 36 | |
| 12 | MODULES USED | 37 | |
| 13 | BIBLIOGRAPHY | 38 | |

**ACKNOWLEDGEMENT**

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to the luminary **The Principal, Leena BH** who has been continuously motivating and extending their helping hand to us.

My sincere thanks to **NARASA REDDY**, Master In-charge, A guide, Mentor all the above a friend, who critically reviewed my project and helped in solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

**INTRODUCTION TO PYTHON**

Python is a high-level general-purpose programming language. Its design philosophy emphasises code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

Python consistently ranks as one of the most popular programming languages.

Guido Van Rossum began working on Python in the late 1980s, as a successor to the ABC Programming Language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

**HISTORY OF PYTHON**

Python consistently ranks as one of the most popular programming languages.

Python was conceived in the late 1980s by Guido Van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC Programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end of life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No further security patches or other improvements will be released for it. With Python 2's end of life, only Python 3.6.x and later are supported.

Python 3.9.2 and 3.8.8 were expedited as all versions of Python had security issues leading to possible remote code execution and web cache poisoning.

**INTRODUCTION TO THE PROJECT**

"**PONG**" is a remake of the first video game in the history of mankind. This old school classic arcade game comes with different balanced difficulty modes for players of every level of experience and skill. A light hearted video game, which takes just a minute to learn but a lifetime to master.

**OBJECTIVES OF THE PROJECT**

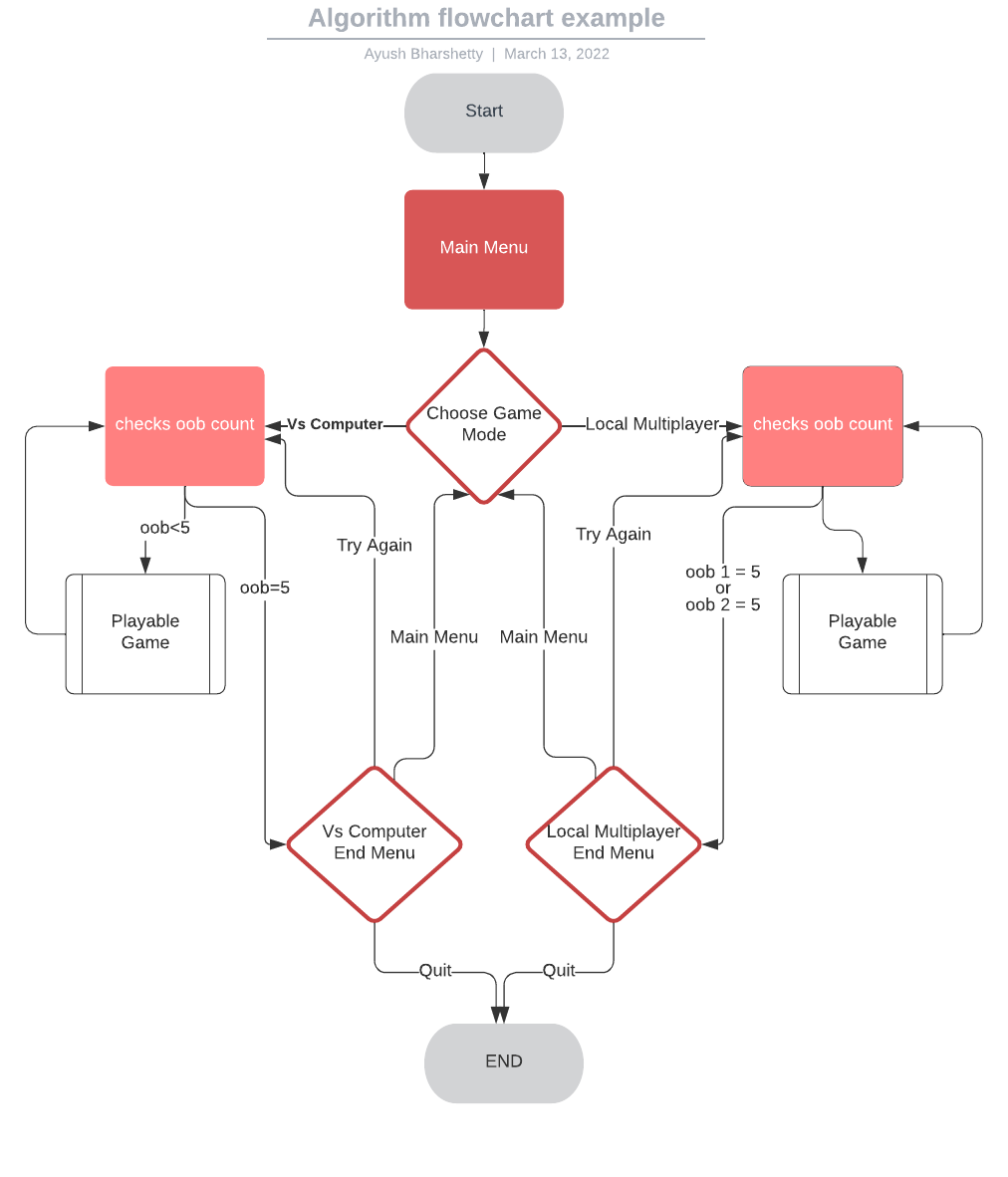
The objective of this project is to be an easy to learn but fun and engaging game. It aims to :

* Be a simple easy to learn game
* Fun for all age groups
* Take players down a trip of nostalgia with its arcadey retro vibe
* Introducing the kids of this generation to a legendary game of its time.

**ALGORITHM**

1. Start
2. Choose game mode
3. Vs Computer
   1. Oob < 5
      1. Playable game
   2. Oob = 5
      1. End Menu
4. Local Multiplayer
   * 1. Oob1 < 5 and oob2 < 5
        1. Playable game
     2. Oob1 = 5 or oob2 = 5
        1. End Menu
5. End Menu
6. End

**FLOW CHART**

****

**SOURCE CODE**

**import pygame, sys**

**from pygame.locals import \***

**import os**

**import pygame.freetype**

**#main display**

**screen\_width = 1200**

**screen\_height = 750**

**screen = pygame.display.set\_mode((screen\_width,screen\_height))**

**pygame.display.set\_caption('Pong')**

**#initializing required values**

**score = 0**

**score1, score2 = 0,0**

**oob, oob1, oob2 = 5, 5, 5**

**a,j,i = 0,0,0**

**k = screen\_width - 370**

**inc = 30**

**global rectx, thic**

**rectx = screen\_width - 30**

**thic = 50**

**ball1x,ball1y = 400,400**

**select,select1 = 0,0**

**tryagain = 0**

**hitx, hity = 0, 0**

**hitxs, hitys = 0, 0**

**#initializing modules**

**pygame.init()**

**clock=pygame.time.Clock()**

**pygame.freetype.init()**

**#Initializing Fonts**

**font1 = pygame.freetype.Font('Pokemon\_Gb.ttf', 80)**

**letter = pygame.freetype.Font('Pokemon\_Gb.ttf', 40)**

**heading = pygame.freetype.Font('Heading.ttf', 40)**

**heading1 = pygame.freetype.Font('FlynnHollow.ttf', 140)**

**heading2 = pygame.freetype.Font('FlynnHollow.ttf', 100)**

**heading3 = pygame.freetype.Font('SelawikSemilight.ttf', 20)**

**options = pygame.freetype.Font('Heading.ttf', 30)**

**options1 = pygame.freetype.Font('Heading.ttf', 25)**

**mscore = pygame.freetype.Font('Heading.ttf', 40)**

**lives = pygame.freetype.Font('MfLoveDings.ttf',40)**

**def lives1():**

**if oob1 == 5:**

**lives.render\_to(screen ,(190, screen\_height/2 - 110), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 - 65), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 - 20), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 25), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 65), 'g', grey)**

**elif oob1 ==4:**

**lives.render\_to(screen ,(190, screen\_height/2 - 87.5), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 - 42.5), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 2.5), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 47.5 ), 'g', grey)**

**elif oob1 == 3:**

**lives.render\_to(screen ,(190, screen\_height/2 - 65), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 - 20), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 25), 'g', grey)**

**elif oob1 == 2:**

**lives.render\_to(screen ,(190, screen\_height/2 - 42.5), 'g', grey)**

**lives.render\_to(screen ,(190, screen\_height/2 + 2.5), 'g', grey)**

**elif oob1 == 1:**

**lives.render\_to(screen ,(190, screen\_height/2 - 20), 'g', grey)**

**if oob2 == 5:**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 110), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 65), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 20), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 25), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 65), 'g', grey)**

**elif oob2 ==4:**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 87.5), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 42.5), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 2.5), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 47.5 ), 'g', grey)**

**elif oob2 == 3:**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 65), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 20), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 25), 'g', grey)**

**elif oob2 == 2:**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 42.5), 'g', grey)**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 + 2.5), 'g', grey)**

**elif oob2 == 1:**

**lives.render\_to(screen ,(screen\_width - 190, screen\_height/2 - 20), 'g', grey)**

**def lives2():**

**global x**

**if oob == 5:**

**x = screen\_width/2-100**

**elif oob == 4:**

**x = screen\_width/2-77.5**

**elif oob == 3:**

**x = screen\_width/2-55**

**elif oob == 2:**

**x = screen\_width/2-35**

**elif oob == 1:**

**x = screen\_width/2-14**

**def computer\_boundary():**

**pygame.draw.line(screen,(255,255,255),(20, 20),(screen\_width/2 - 115, 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 + 125 , 20),(screen\_width - 20, 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, 20),(screen\_width - 20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(20, 20),(20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 - 115, screen\_height - 20),(20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 + 125, screen\_height - 20),(screen\_width - 20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 - 115, 45 ),(screen\_width/2 + 125, 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 - 115, 20),(screen\_width/2 - 115, 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 + 125, 20),(screen\_width/2 + 125, 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 - 115, screen\_height - 20),(screen\_width/2 - 115, screen\_height - 55), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 + 125, screen\_height - 20),(screen\_width/2 + 125, screen\_height - 55), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2 - 115, screen\_height - 55),(screen\_width/2 + 125, screen\_height - 55), 2)**

**def multiplayer\_boundary():**

**pygame.draw.line(screen, white,(20, 20),(20, screen\_height - 20), 2)**

**pygame.draw.line(screen, white,(20, screen\_height - 20),(screen\_width - 20, screen\_height - 20), 2)**

**pygame.draw.line(screen, white,(screen\_width - 20, screen\_height - 20),(screen\_width- 20, 20), 2)**

**pygame.draw.line(screen, white,(20, 20),(screen\_width - 20, 20), 2)**

**pygame.draw.line(screen, white, (screen\_width/2, 20),(screen\_width/2, 60), 2)**

**pygame.draw.line(screen, white, (screen\_width/2, screen\_height - 20),(screen\_width/2, screen\_height - 60), 2)**

**def intro\_words():**

**options.render\_to(screen ,(screen\_width/2 - 75, screen\_height/2 + 15), 'LOCAL', grey)**

**options.render\_to(screen ,(screen\_width/2 - 160, screen\_height/2 + 50), 'MULTIPLAYER', grey)**

**options.render\_to(screen ,(screen\_width/2 - 150, screen\_height/2+150), 'VS COMPUTER', grey)**

**def intro\_boundary():**

**pygame.draw.line(screen,(255,255,255),(20, 20),(screen\_width - 20, 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, 20),(screen\_width - 20, 190), 2)**

**pygame.draw.line(screen,(255,255,255),(20, 20),(20, 190), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, 280),(screen\_width - 20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(20, 280),(20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, screen\_height - 20),(20, screen\_height - 20), 2)**

**def welcome\_words():**

**heading.render\_to(screen ,(screen\_width/2 - 190, 125), 'WELCOME TO', lightgrey)**

**heading1.render\_to(screen ,(screen\_width/2 - 227 , 180), 'PONG', white)**

**pygame.draw.line(screen, blue, (screen\_width/2 + 235, 230), (screen\_width,230), 50)**

**pygame.draw.line(screen, red, (screen\_width/2 - 235, 230), (0, 230), 50)**

**#VsComputerWords**

**def VsComputer\_words():**

**if choice == 2:**

**letter.render\_to(screen ,(150, 200), 'C', grey)**

**letter.render\_to(screen ,(150, 250), 'O', grey)**

**letter.render\_to(screen ,(150, 300), 'M', grey)**

**letter.render\_to(screen ,(150, 350), 'P', grey)**

**letter.render\_to(screen ,(150, 400), 'U', grey)**

**letter.render\_to(screen ,(150, 450), 'T', grey)**

**letter.render\_to(screen ,(150, 500), 'E', grey)**

**letter.render\_to(screen ,(150, 550), 'R', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 250), 'P', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 300), 'L', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 350), 'A', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 400), 'Y', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 450), 'E', grey)**

**letter.render\_to(screen ,(screen\_width - 150, 500), 'R', grey)**

**#MultiplayerWords**

**def Multiplayer\_words():**

**if choice == 1:**

**letter.render\_to(screen ,(screen\_width - 125, 225), 'P', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 275), 'L', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 325), 'A', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 375), 'Y', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 425), 'E', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 475), 'R', grey)**

**letter.render\_to(screen ,(screen\_width - 125, 525), '1', grey)**

**letter.render\_to(screen ,(125, 225), 'P', grey)**

**letter.render\_to(screen ,(125, 275), 'L', grey)**

**letter.render\_to(screen ,(125, 325), 'A', grey)**

**letter.render\_to(screen ,(125, 375), 'Y', grey)**

**letter.render\_to(screen ,(125, 425), 'E', grey)**

**letter.render\_to(screen ,(125, 475), 'R', grey)**

**letter.render\_to(screen ,(125, 525), '2', grey)**

**#making player, opponent and ball**

**global ball, player, opponent**

**ball = pygame.Rect(screen\_width/2-15, screen\_height/2-15,30,30)**

**player = pygame.Rect(screen\_width - 45 ,screen\_height/2 - 70,15,150)**

**opponent = pygame.Rect(30 ,screen\_height/2 - 70, 15, 150)**

**ball1 = pygame.Rect(400, 400,inc,inc)**

**#player, opponent and ball speed**

**ball\_speed\_x = -8**

**ball\_speed\_y = -8**

**player\_speed = 6**

**opponent\_speed =6**

**#colors**

**bg\_color = (0, 0, 0)**

**white = (255, 255, 255)**

**blue = (255,94,94)**

**red = (95,163,254)**

**grey = (124, 124, 124)**

**lightgrey = (204, 204, 204)**

**darkgrey = (10,10,10)**

**def countdown3():**

**while True:**

**screen.fill(bg\_color)**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height/2 - 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2,screen\_height/2 + 37),(screen\_width/2, screen\_height - 55), 2)**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**font1.render\_to(screen ,(screen\_width/2-30, screen\_height/2-40),'3' , grey)**

**pygame.display.update()**

**clock.tick(30)**

**pygame.time.delay(800)**

**break**

**def countdown2():**

**while True:**

**screen.fill(bg\_color)**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height/2 - 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2,screen\_height/2 + 37),(screen\_width/2, screen\_height - 55), 2)**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**font1.render\_to(screen ,(screen\_width/2-30, screen\_height/2-40),'2' , grey)**

**pygame.display.update()**

**clock.tick(30)**

**pygame.time.delay(800)**

**break**

**def countdown1():**

**while True:**

**screen.fill(bg\_color)**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height/2 - 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2,screen\_height/2 + 37),(screen\_width/2, screen\_height - 55), 2)**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**font1.render\_to(screen ,(screen\_width/2-30, screen\_height/2-40),'1' , grey)**

**pygame.display.update()**

**clock.tick(30)**

**pygame.time.delay(800)**

**break**

**def countdown0():**

**while True:**

**screen.fill(bg\_color)**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height/2 - 45), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2,screen\_height/2 + 37),(screen\_width/2, screen\_height - 55), 2)**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**font1.render\_to(screen ,(screen\_width/2-72, screen\_height/2-40),'GO' , grey)**

**pygame.display.update()**

**clock.tick(30)**

**pygame.time.delay(800)**

**a = 2**

**break**

**def end\_boundary():**

**pygame.draw.line(screen, white, (screen\_width - 20, 20), (20, 20), 3)**

**pygame.draw.line(screen, white, (screen\_width - 20, 95), (screen\_width - 20, 20), 3)**

**pygame.draw.line(screen, white, (20, 95), (20, 20), 3)**

**pygame.draw.line(screen, white, (screen\_width - 20, 290), (screen\_width - 20, screen\_height - 20), 3)**

**pygame.draw.line(screen, white, (20, 290), (20, screen\_height - 20), 3)**

**pygame.draw.line(screen, white, (20, screen\_height - 20), (screen\_width - 340, screen\_height - 20), 3)**

**pygame.draw.line(screen, white, (screen\_width - 20, screen\_height - 20), (screen\_width - 170, screen\_height - 20), 3)**

**heading3.render\_to(screen ,(screen\_width - 330, screen\_height - 28.5), 'Ayush Bharshetty', white)**

**def outerblack():**

**#outerbalck boundary**

**pygame.draw.line(screen,bg\_color,(0,0),(screen\_width,0), 38) #top**

**pygame.draw.line(screen,bg\_color,(screen\_width,screen\_height),(screen\_width,0),42) #right**

**pygame.draw.line(screen,bg\_color,(screen\_width,screen\_height),(0,screen\_height),48)#bottom**

**pygame.draw.line(screen,bg\_color,(0,screen\_height),(0,0),32)#left**

**def gameover\_text():**

**heading2.render\_to(screen ,(screen\_width/2 - 230, 110), 'GAME', lightgrey)**

**heading2.render\_to(screen ,(screen\_width/2 - 80, 205), 'OVER', lightgrey)**

**pygame.draw.line(screen, blue, (screen\_width/2 - 245, 145), (0 ,145), 60)**

**pygame.draw.line(screen, blue, (screen\_width/2 + 115, 145), (screen\_width,145), 60)**

**pygame.draw.line(screen, red, (screen\_width/2 - 95, 235), (0, 235), 60)**

**pygame.draw.line(screen, red, (screen\_width, 235), (screen\_width - 340, 235), 60)**

**def reset():**

**#resetting all actor locations**

**player.y = screen\_height/2 - 70**

**opponent.y = screen\_height/2 - 70**

**ball.x = screen\_width/2 - 15**

**ball.y = screen\_height/2-15**

**def end\_menu():**

**options1.render\_to(screen ,(screen\_width/2 - 106, screen\_height/2), 'RETURN TO', grey)**

**options1.render\_to(screen ,(screen\_width/2 - 110, screen\_height/2 + 30), 'MAIN MENU', grey)**

**options1.render\_to(screen ,(screen\_width/2 - 106, screen\_height/2 + 105), 'TRY AGAIN', grey)**

**options1.render\_to(screen ,(screen\_width/2 - 40, screen\_height/2 + 180), 'QUIT', grey)**

**#MainGame**

**while True:**

**#START MENU**

**while a == 0:**

**while True:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**pygame.quit()**

**sys.exit()**

**select = 0**

**a = 0**

**choice = 0**

**screen.fill(bg\_color)**

**mouse = pygame.mouse.get\_pos()**

**if tryagain == 2:**

**gameover\_text()**

**end\_menu()**

**end\_boundary()**

**pygame.draw.rect(screen, white, closing1)**

**rectx -= 25**

**thic += 26**

**if rectx <= 1:**

**rectx = 0**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**#pygame.draw.rect(screen, red, [screen\_width/2 - 200,screen\_height/2,400,100])**

**#pygame.draw.rect(screen, blue, [screen\_width/2 - 200,screen\_height/2+120,400,100])**

**intro\_words()**

**#outer boundary**

**intro\_boundary()**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**if screen\_width/2-150 <= mouse[0] <= screen\_width/2+150 and screen\_height/2 <= mouse[1] <= screen\_height/2+80:**

**choice = 1**

**select = 1**

**a = 1**

**elif screen\_width/2-150 <= mouse[0] <= screen\_width/2+150 and screen\_height/2+100 <= mouse[1] <= screen\_height/2+180:**

**choice = 2**

**select = 1**

**a = 1**

**if screen\_width/2-150 <= mouse[0] <= screen\_width/2+150 and screen\_height/2 <= mouse[1] <= screen\_height/2+80:**

**#pygame.draw.rect(screen, white, [screen\_width/2 - 2000,screen\_height/2,400,100])**

**options.render\_to(screen ,(screen\_width/2 - 75, screen\_height/2 + 15), 'LOCAL', lightgrey)**

**options.render\_to(screen ,(screen\_width/2 - 160, screen\_height/2 + 50), 'MULTIPLAYER', lightgrey)**

**elif screen\_width/2-150 <= mouse[0] <= screen\_width/2+150 and screen\_height/2+100 <= mouse[1] <= screen\_height/2+180:**

**#pygame.draw.rect(screen, white, [screen\_width/2 - 2000,screen\_height/2+120,400,100])**

**options.render\_to(screen ,(screen\_width/2 - 150, screen\_height/2+150), 'VS COMPUTER', lightgrey)**

**welcome\_words()**

**if select == 1:**

**a = 1**

**break**

**pygame.display.update()**

**clock.tick(30)**

**#START MENU TO GAME**

**while a == 1:**

**screen.fill(bg\_color)**

**closing1 = pygame.Rect (rectx, 0, thic, screen\_height)**

**cover = pygame.Rect (0, 0, 30, screen\_height)**

**outerblack()**

**if rectx > 0:**

**if tryagain == 0:**

**options.render\_to(screen ,(screen\_width/2 - 75, screen\_height/2 + 15), 'LOCAL', grey)**

**options.render\_to(screen ,(screen\_width/2 - 160, screen\_height/2 + 50), 'MULTIPLAYER', grey)**

**options.render\_to(screen ,(screen\_width/2 - 150, screen\_height/2+150), 'VS COMPUTER', grey)**

**#outer boundary**

**pygame.draw.line(screen,(255,255,255),(20, 20),(screen\_width - 20, 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, 20),(screen\_width - 20, 190), 2)**

**pygame.draw.line(screen,(255,255,255),(20, 20),(20, 190), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, 280),(screen\_width - 20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(20, 280),(20, screen\_height - 20), 2)**

**pygame.draw.line(screen,(255,255,255),(screen\_width - 20, screen\_height - 20),(20, screen\_height - 20), 2)**

**#Main Heading**

**heading.render\_to(screen ,(screen\_width/2 - 190, 125), 'WELCOME TO', lightgrey)**

**heading1.render\_to(screen ,(screen\_width/2 - 227 , 180), 'PONG', white)**

**#SideLines**

**pygame.draw.line(screen, blue, (screen\_width/2 + 235, 230), (screen\_width,230), 50)**

**pygame.draw.line(screen, red, (screen\_width/2 - 235, 230), (0, 230), 50)**

**pygame.draw.rect(screen, white, closing1)**

**rectx -= 25**

**thic += 26**

**elif tryagain == 1:**

**gameover\_text()**

**end\_menu()**

**end\_boundary()**

**pygame.draw.rect(screen, white, closing1)**

**rectx -= 25**

**thic += 26**

**if rectx <= 1:**

**rectx = 0**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height - 55), 2)**

**font1.render\_to(screen ,(screen\_width/2-30, screen\_height/2-40), '3', grey)**

**pygame.draw.rect(screen, white, closing1)**

**thic -= 25**

**if thic <= -2:**

**a = 2**

**rectx = screen\_width - 30**

**thic = 1**

**pygame.display.update()**

**clock.tick(60)**

**#COUNTDOWN**

**while a == 2:**

**outerblack()**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**pygame.quit()**

**sys.exit()**

**countdown3()**

**countdown2()**

**countdown1()**

**countdown0()**

**tryagain = 0**

**a = 3**

**#MAIN GAME MECHANISM**

**while a == 3:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**pygame.quit()**

**sys.exit()**

**if event.type == pygame.KEYDOWN:**

**if event.key == pygame.K\_DOWN:**

**player\_speed = 0**

**player\_speed += 8**

**if event.key == pygame.K\_UP:**

**player\_speed = 0**

**player\_speed -= 8**

**if choice == 1:**

**if event.key == pygame.K\_s:**

**opponent\_speed = 0**

**opponent\_speed += 8**

**if event.key == pygame.K\_w:**

**opponent\_speed = 0**

**opponent\_speed -= 8**

**#so that ball bounces of rectangles in local multiplayer**

**if choice == 2:**

**if ball.bottom >= screen\_height - 55 and ball.right <= screen\_width/2 + 120 and ball.left >= screen\_width/2 - 110:**

**ball\_speed\_y \*= -1**

**ball.bottom = screen\_height - 55**

**hitx = ball.x**

**hity = ball.bottom**

**if ball.top <= 45 and ball.right <= screen\_width/2 + 120 and ball.left >= screen\_width/2 - 110:**

**ball\_speed\_y \*= -1**

**ball.top = 45**

**hitx = ball.x**

**hity = ball.top**

**#ball boundary limits**

**if ball.top <= 21:**

**ball\_speed\_y \*= -1**

**ball.top = 21**

**hitx = ball.x**

**hity = ball.top**

**if ball.bottom >= screen\_height - 21:**

**ball\_speed\_y \*= -1**

**ball.bottom = screen\_height - 21**

**hitx = ball.x**

**hity = ball.bottom**

**if ball.left <= 21:**

**ball\_speed\_x \*= -1**

**ball.left = 21**

**hitx = 21**

**hity = ball.y**

**if choice == 1:**

**oob1 -= 1**

**if ball.right >= screen\_width - 21:**

**ball\_speed\_x \*= -1**

**ball.right = screen\_width - 21**

**hitx = screen\_width-21**

**hity = ball.y**

**if choice == 1:**

**oob2 -= 1**

**if choice == 2:**

**oob -= 1**

**#ball player hit interaction**

**if ball.colliderect(player):**

**ball\_speed\_x \*= -1**

**if choice == 1:**

**score2 += 1**

**if choice == 2:**

**score += 1**

**if ball.colliderect(opponent):**

**ball\_speed\_x \*= -1**

**if choice == 1:**

**score1 += 1**

**#player boundaries**

**if player.top <= 33:**

**player.top = 33**

**if player.bottom >= screen\_height - 31:**

**player.bottom = screen\_height - 31**

**#opponent boundaries**

**if choice == 2:**

**if opponent.bottom >= ball.y + 100:**

**opponent\_speed = 0**

**opponent\_speed -= 8**

**if opponent.top <= ball.y - 70:**

**opponent\_speed = 0**

**opponent\_speed += 8**

**if opponent.top <= 33:**

**opponent.top = 33**

**if opponent.bottom >= screen\_height - 31:**

**opponent.bottom = screen\_height - 31**

**#initialising ball and player speed**

**ball.x += ball\_speed\_x**

**ball.y += ball\_speed\_y**

**player.y += player\_speed**

**opponent.y += opponent\_speed**

**#boundary hit animation**

**if hitxs < 50:**

**hitxs += 0.5**

**elif hitxs >= 50:**

**hitxs = 0**

**if hitys < 50:**

**hitys += 0.5**

**elif hitys >= 50:**

**hitys = 0**

**#rendering all actors**

**screen.fill(bg\_color)**

**hit = pygame.Rect(hitx, hity,hitxs,hitys)**

**pygame.draw.circle(screen, lightgrey, (hitx, hity), hitxs, width=0)**

**pygame.draw.circle(screen, bg\_color, (hitx, hity), hitxs-1, width=0)**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.ellipse(screen,white , ball)**

**#pygame.draw.ellipse(screen,grey , hit)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height - 55), 2)**

**#rendering fonts**

**font1.render\_to(screen ,(screen\_width/2-74, screen\_height/2-30), 'VS', grey)**

**mscore.render\_to(screen ,(250, screen\_height/2 - 65), 'g', grey)**

**#game bg and outline structure**

**outerblack()**

**if choice == 1:**

**lives1()**

**mscore.render\_to(screen, (screen\_width - 260, screen\_height/2 - 20), str(score2), grey)**

**mscore.render\_to(screen, (260, screen\_height/2 - 20), str(score1), grey)**

**if choice == 2:**

**lives2()**

**options.render\_to(screen ,(screen\_width/2-106, 10), 'SCORE:'+str(score), grey)**

**lives.render\_to(screen ,(x, screen\_height - 50), 'g '\*oob, grey)**

**Multiplayer\_words()**

**VsComputer\_words()**

**if choice == 1:**

**multiplayer\_boundary()**

**elif choice == 2:**

**computer\_boundary()**

**pygame.display.update()**

**clock.tick(120)**

**ball1x = ball.x**

**ball1y = ball.y**

**if oob <= 0:**

**a = 4**

**score = 0**

**oob = 5**

**pygame.time.delay(1200)**

**if oob1 <= 0 or oob2 <= 0:**

**a =4**

**score1 = 0**

**score2 = 0**

**oob1 = 5**

**oob2 = 5**

**pygame.time.delay(1200)**

**#GAME TO END MENU**

**while a == 4:**

**screen.fill(bg\_color)**

**outerblack()**

**if rectx >= 1:**

**pygame.draw.rect(screen, red, player)**

**pygame.draw.rect(screen, blue, opponent)**

**pygame.draw.ellipse(screen,white , ball)**

**pygame.draw.line(screen,(255,255,255),(screen\_width/2, 45),(screen\_width/2,screen\_height - 55), 2)**

**if choice == 1 and rectx >= 1:**

**multiplayer\_boundary()**

**elif choice == 2 and rectx >= 1:**

**computer\_boundary()**

**closing1 = pygame.Rect (rectx, 0, thic, screen\_height)**

**cover = pygame.Rect (0, 0, 30, screen\_height)**

**if rectx > 0:**

**pygame.draw.rect(screen, white, closing1)**

**rectx -= 18**

**thic += 19**

**if rectx <= 1:**

**rectx = 0**

**gameover\_text()**

**end\_menu()**

**end\_boundary()**

**pygame.draw.rect(screen, white, closing1)**

**thic -= 18**

**if thic <= -2:**

**a = 5**

**rectx = screen\_width - 30**

**thic = 1**

**pygame.display.update()**

**clock.tick(60)**

**#END MENU**

**while a == 5:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**pygame.quit()**

**sys.exit()**

**select = 0**

**outerblack()**

**reset()**

**screen.fill(bg\_color)**

**gameover\_text()**

**end\_boundary()**

**mouse = pygame.mouse.get\_pos()**

**end\_menu()**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**if screen\_width/2-150 <= mouse[0] <= screen\_width/2+150 and screen\_height/2 <= mouse[1] <= screen\_height/2+80:**

**a = 6**

**ball.x = screen\_width/2 - 15**

**ball.y = screen\_height/2 - 15**

**tryagain = 1**

**elif screen\_width/2-115 <= mouse[0] <= screen\_width/2+135 and screen\_height/2+100 <= mouse[1] <= screen\_height/2+160:**

**a = 1**

**ball.x = screen\_width/2 - 15**

**ball.y = screen\_height/2 - 15**

**tryagain = 1**

**elif screen\_width/2-40 <= mouse[0] <= screen\_width/2+40 and screen\_height/2+175 <= mouse[1] <= screen\_height/2+210:**

**pygame.quit()**

**sys.exit()**

**if screen\_width/2-115 <= mouse[0] <= screen\_width/2+135 and screen\_height/2 <= mouse[1] <= screen\_height/2+70:**

**#pygame.draw.rect(screen, white, [screen\_width/2 - 2000,screen\_height/2,400,100])**

**options1.render\_to(screen ,(screen\_width/2 - 106, screen\_height/2), 'RETURN TO', white)**

**options1.render\_to(screen ,(screen\_width/2 - 110, screen\_height/2 + 30), 'MAIN MENU', white)**

**elif screen\_width/2-115 <= mouse[0] <= screen\_width/2+135 and screen\_height/2+100 <= mouse[1] <= screen\_height/2+160:**

**#pygame.draw.rect(screen, white, [screen\_width/2 - 2000,screen\_height/2+120,400,100])**

**options1.render\_to(screen ,(screen\_width/2 - 106, screen\_height/2 + 105), 'TRY AGAIN', white)**

**elif screen\_width/2-40 <= mouse[0] <= screen\_width/2+40 and screen\_height/2+175 <= mouse[1] <= screen\_height/2+1210:**

**options1.render\_to(screen ,(screen\_width/2 - 40, screen\_height/2 + 180), 'QUIT', white)**

**pygame.display.update()**

**clock.tick(120)**

**while a == 6:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**pygame.quit()**

**sys.exit()**

**screen.fill(bg\_color)**

**outerblack()**

**if rectx >= 1:**

**gameover\_text()**

**end\_boundary()**

**end\_menu()**

**closing1 = pygame.Rect (rectx, 0, thic, screen\_height)**

**cover = pygame.Rect (0, 0, 30, screen\_height)**

**if rectx > 0:**

**pygame.draw.rect(screen, white, closing1)**

**rectx -= 18**

**thic += 19**

**if rectx <= 1:**

**rectx = 0**

**welcome\_words()**

**intro\_boundary()**

**intro\_words()**

**pygame.draw.rect(screen, white, closing1)**

**thic -= 18**

**if thic <= 0:**

**a = 0**

**#options1.render\_to(screen ,(screen\_width/2 - 110, screen\_height/2 + 30), 'WORKS LOL', white)**

**pygame.display.update()**

**clock.tick(60)**

**SCOPE OF THE PROJECT**

Just like any game, the player can get bored of the same repetitive movements, playstyle and methods so it is essential for any game developer to keep adding new modes, features and play styles to the game to keep the players involved and invested, and pong isnt any different.

Even though pong has the nostalgia triggering advantage compared to other newer games of comparable scale, this in itself might be beneficial in attracting a large player base but won't be enough to retain the same for an extended period of time.

Hence the development of new game featured/modes such as:

* Adding levels to the ‘vs computer’ game mode
* An entirely new game mode - ‘online multiplayer’
* Leaderboards with player name and score in for every game mode

And many more quality of life improvements such as:

* a controls menu to the welcome screen
* A feature to pause the game at any time, with a pause menu
* Customization of the player and computer, or player 1 and player 2 control bars
* Ball color customization
* Custom key binds

Are in the works which will not only bring new fresh and engaging modes to the game, but will also let the players customize their game colors and controls as per their choices, which will by a huge margin eliminate the problem of getting used to a new game and its controls for the player, which is a major interest breaker for most of the players trying out a new game for the first time.

**THE EXISTING SYSTEM**

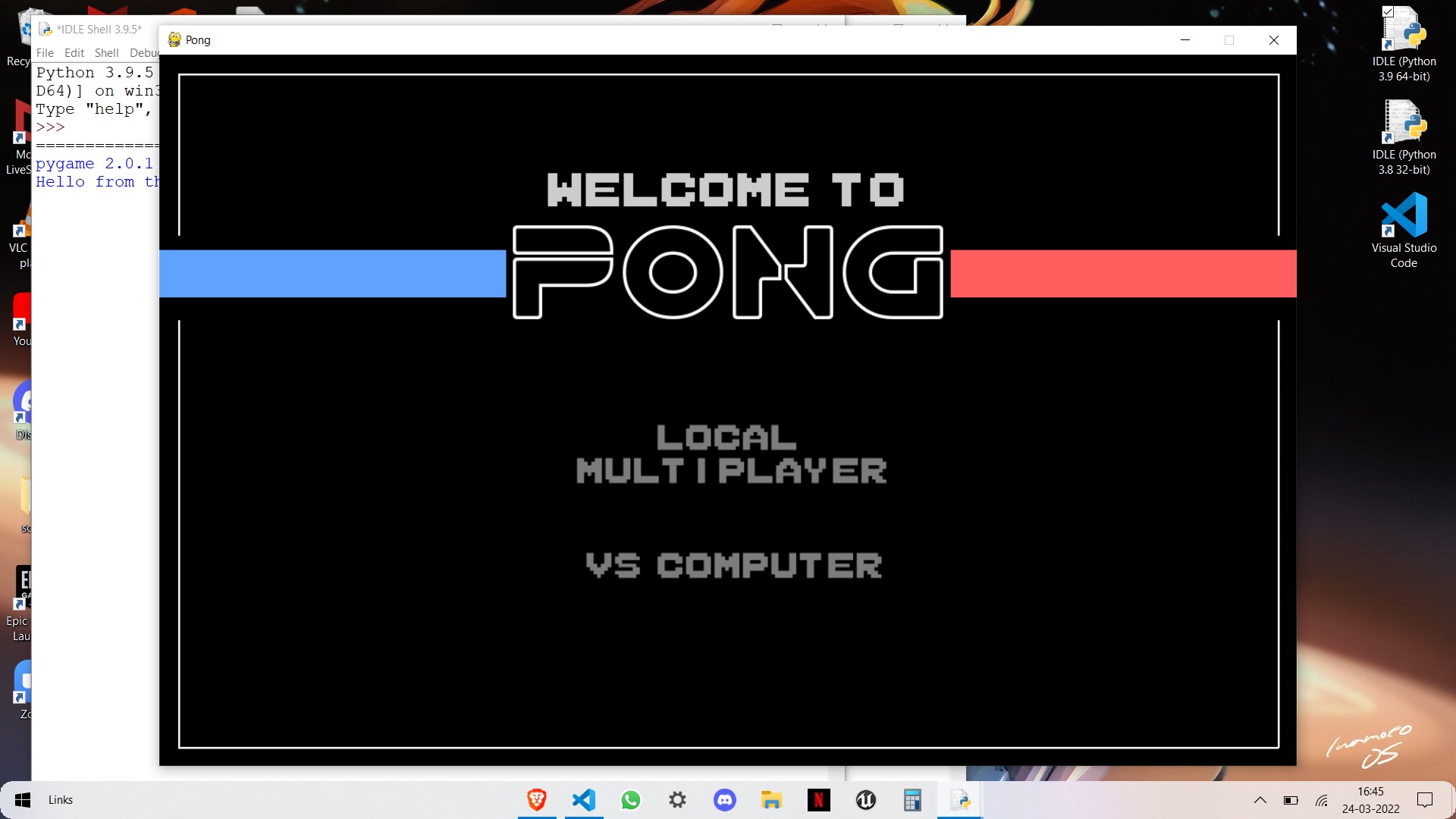
The project uses the PyGame module in python for the interactable objects, to menu display and selection, to the clock system determining the pace of the game. The game works with simple mechanics and controls.

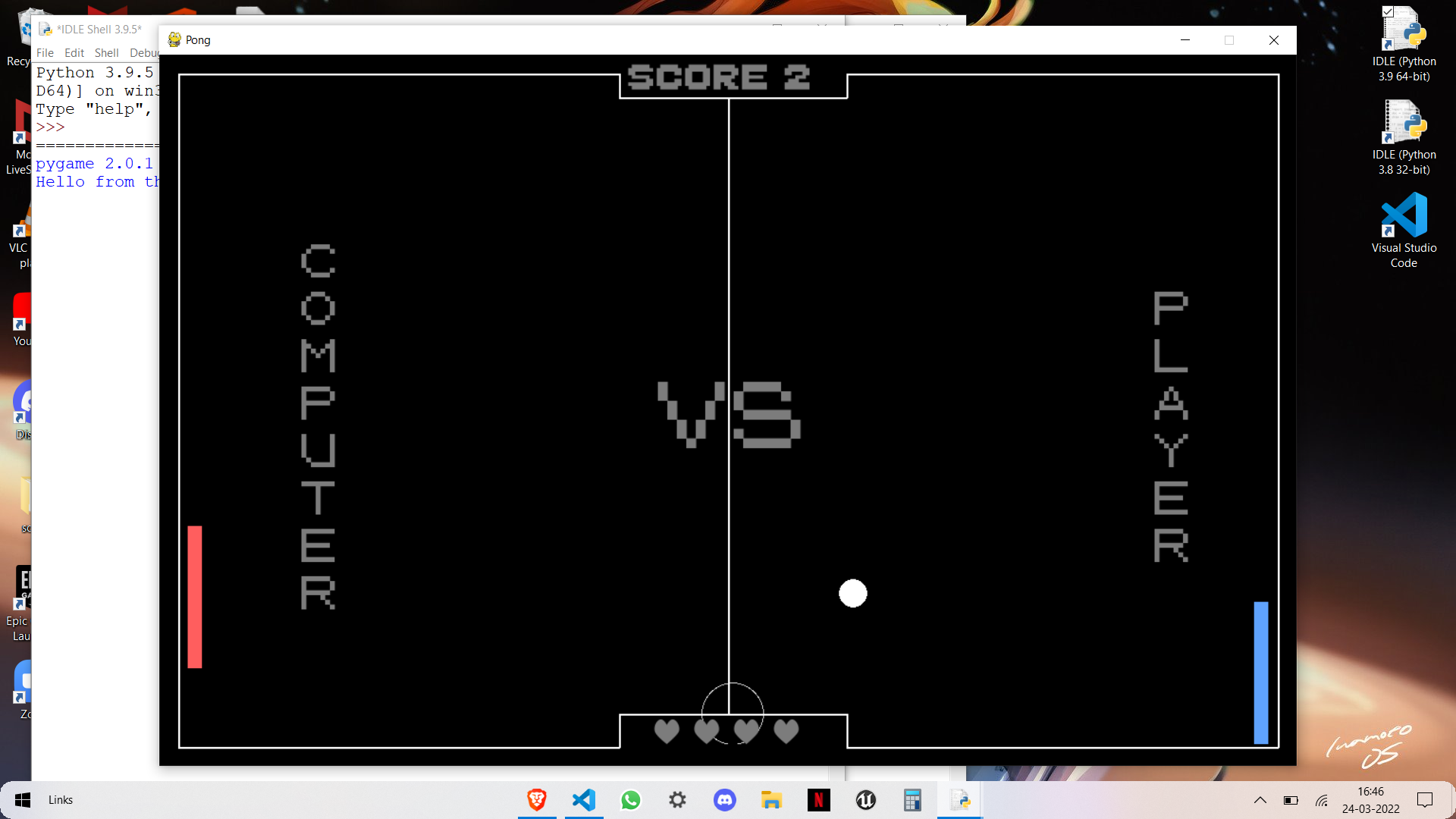
It has a ball bouncing off between the sides of the screen and players use a controllable bar to deflect the ball away from their side to the opponent's side while making sure it doesn't hit the wall on their side.

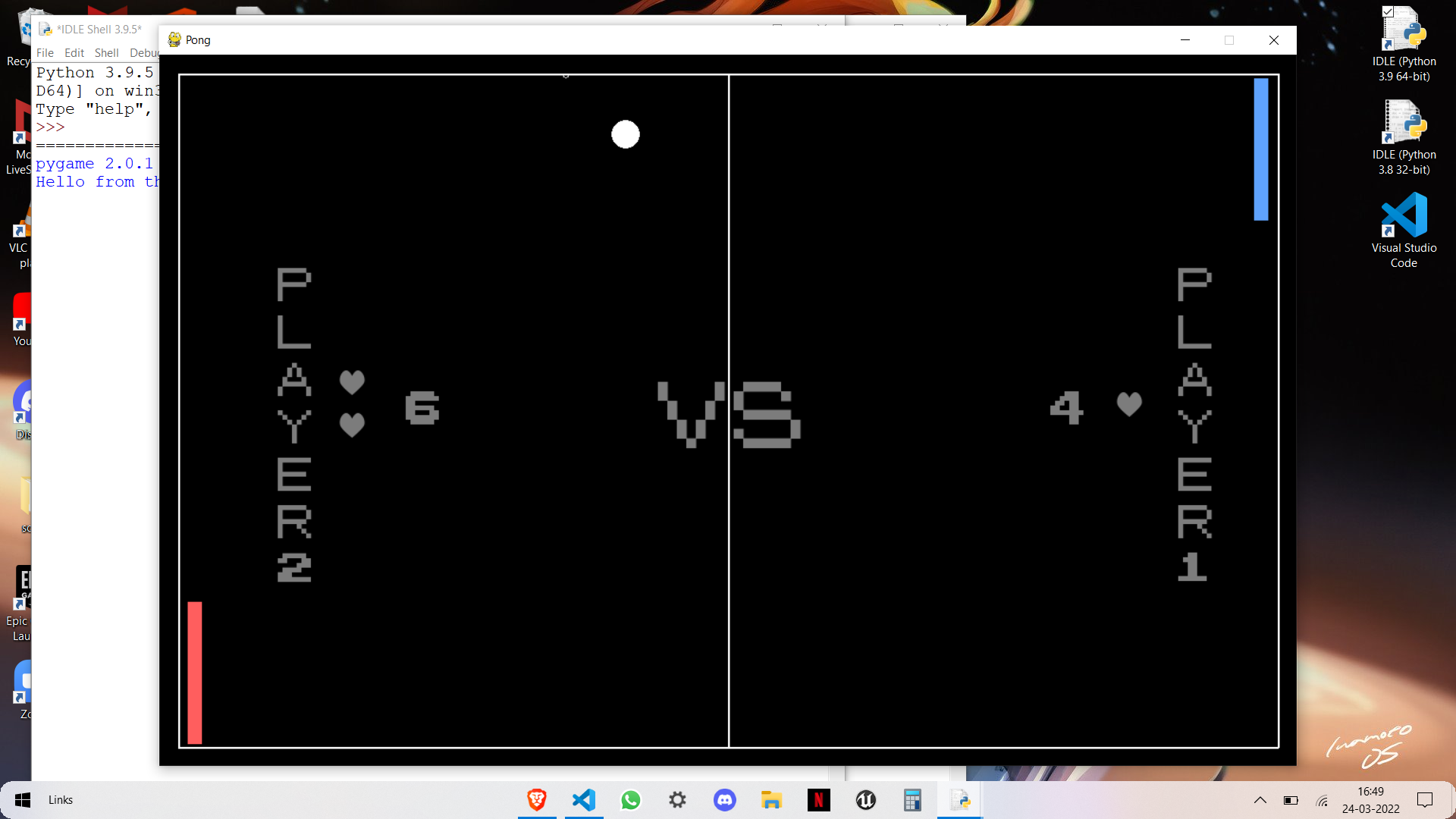
Each player spawns with 5 lives and loses one life every time the ball hits the wall on their side, once either player loses 5 lives the game comes to an end and the player standing with remaining lives and a higher score is the winner.

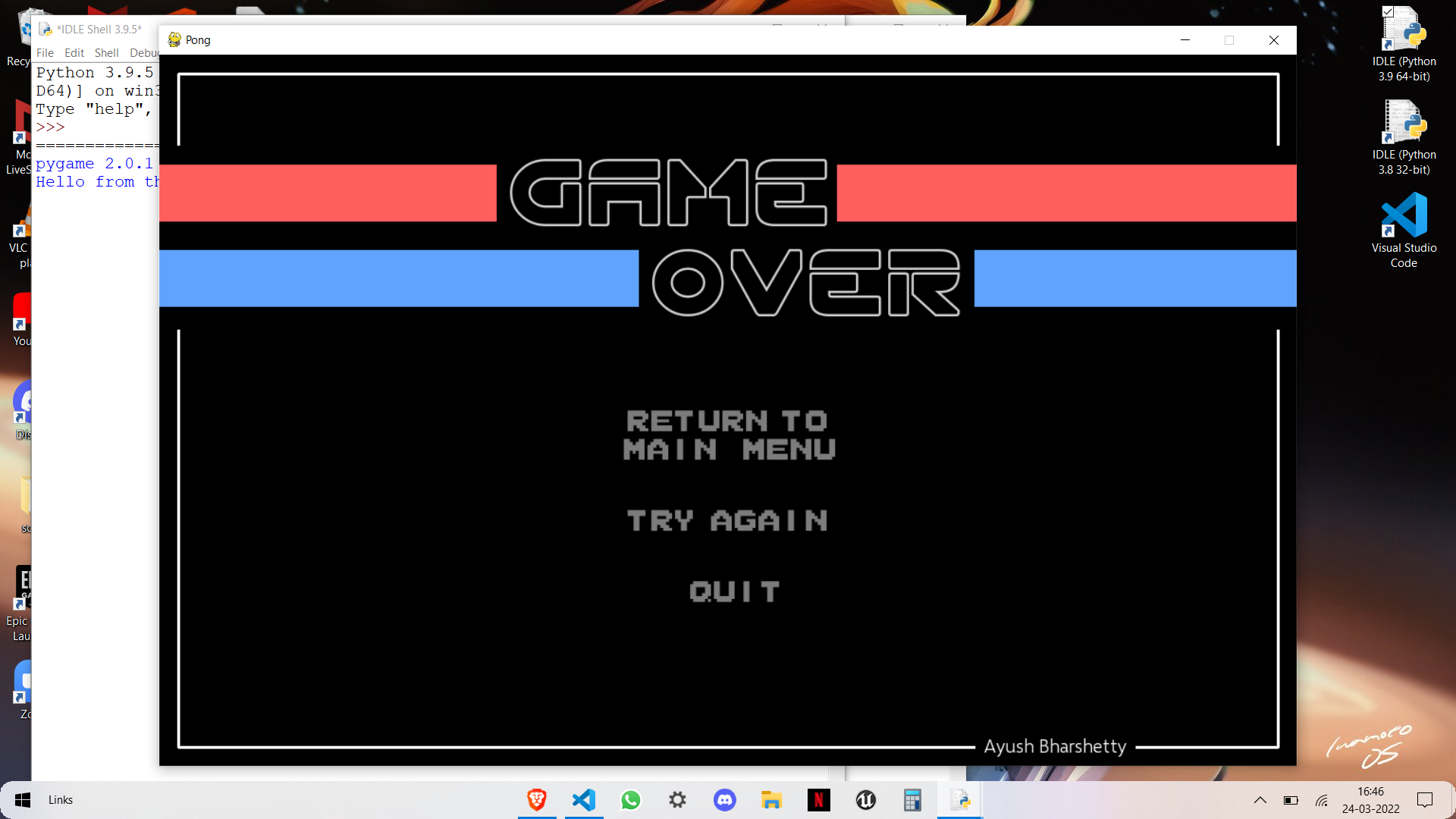
As for the multiplayer mode, it game bar is controlled by the player while the other game bar is controlled by the unbeatable computer AI, the game gets tougher as you progress into a higher score, players spawn with 5 lives and ones they run out of lives the game ends with their existing score being their final score which will determine their place in the score leaderboard. A player gets an unlimited number of tries, giving them the best chance to practice as much as they want and climb the ranks of the leaderboard and become the ultimate PONG champion.

**SCREENSHOTS OF PROGRAM EXECUTION**

****

****

****

****

**MODULES AND TOOLS USED**

**Modules:**

* Pygame
* Sys
* OS

**Editors:**

* Visual Studio Code

**BIBLIOGRAPHY**

* <https://www.geeksforgeeks.org/>
* <https://www.youtube.com/>
* <https://realpython.com/>