Report

By

Het Abhijeet Patel

(Registration No.: 23FE10CSE00398)

Under the Supervision of

Jay S. Sharma

MANIPAL
UNIVERSITY JAIPUR
University u der Sectio 2(f) of the UGC Act

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

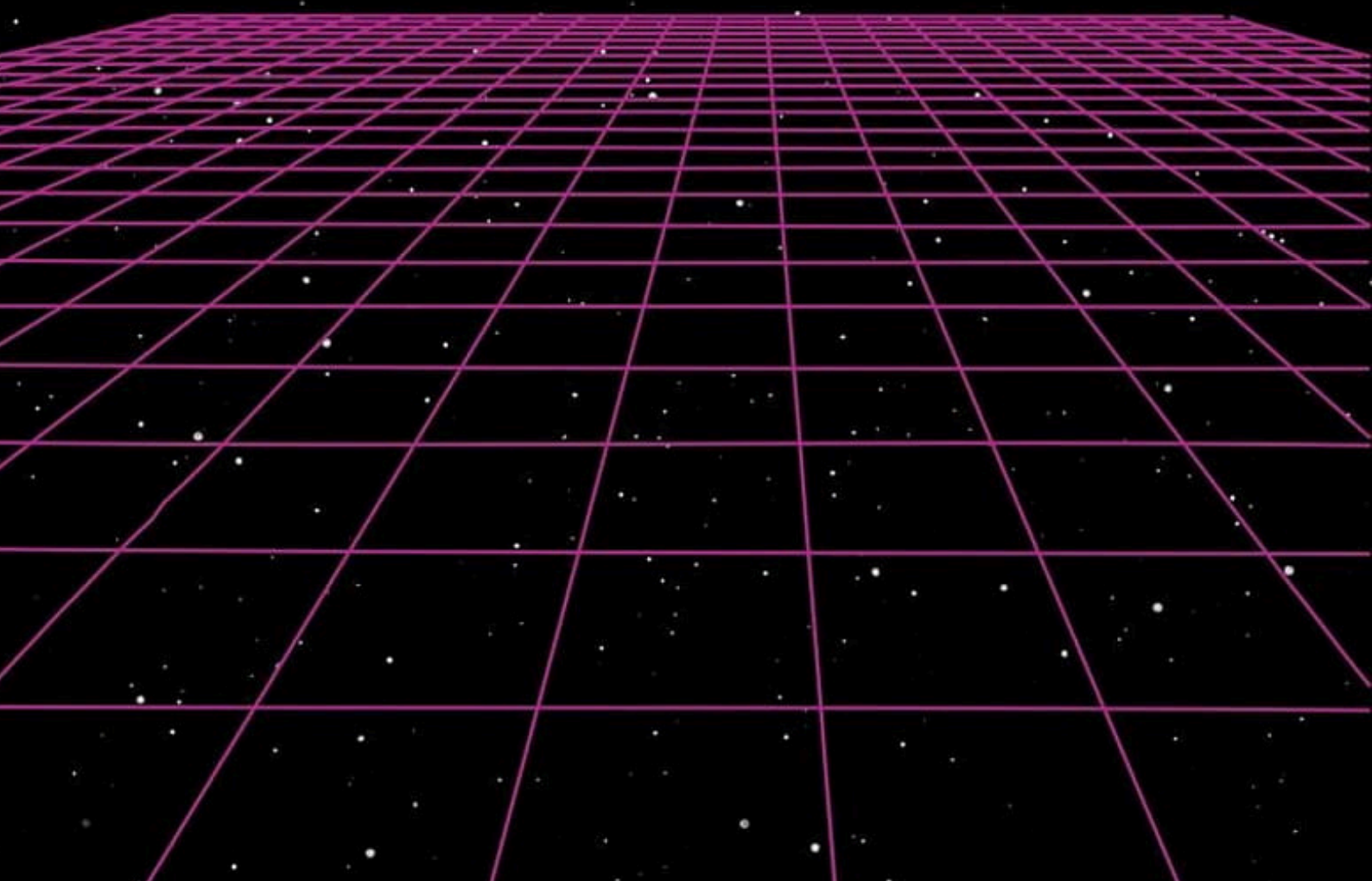
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MANIPAL UNIVERSITY JAIPUR
JAIPUR-303007
RAJASTHAN, INDIA

JULY 2024

COMPUTER PROJECT

RETRO ARCADE



Introduction

We all have played arcade games like space invaders, snake and the food, rock paper scissors. If you loved playing those games, then I guarantee you that you will love our project.

In this Softwere you'll be able to:

- Play four different types of game
    - Snake Game
    - Tic Tac toe
    - Rock Paper Scissors
    - Space invaders
- Add your Name to the database
- To store your game score individually
- Edit your player details in the database

Project Overview

Our project is an all-in-one arcade gaming platform that revives the excitement of classic games while integrating modern features for player management and score tracking. The software includes four popular games—Snake Game, Tic Tac Toe, Rock Paper Scissors, and Space Invaders—creating a versatile and entertaining experience for users. Players can register their names, play games, store their scores, and edit their profiles within the software, enhancing personalization and fostering friendly competition. This project bridges nostalgia and technology, offering a fun and engaging environment for users of all ages.

Abstract

This project is on recreating old arcade games. You can store your name and score in the game database.

There are four different games,

1. Snake Game
2. Tic Tac Toe
3. Rock Paper Scissors
4. Space Invaders

We are using Python with SQL connectivity to store all the cores and player names so it can be viewed by the players whenever they play.

Potential Use Cases

- Entertainment and Stress Relief: This platform provides a way for users to relax and have fun during breaks or leisure time.
- Skill Building and Cognitive Training:
- Tic Tac Toe encourages strategic thinking and planning.
- Snake Game and Space Invaders promote hand-eye coordination and quick reflexes.
- Score Competition: Players can challenge friends, compete for high scores, and strive to outperform their personal bests, adding a competitive element to the gameplay.
- Learning Database Management: For developers and learners, this project offers an opportunity to understand database interaction through player registration, score storage, and data retrieval.
- Game Programming Practice: Coders can explore game development techniques and the integration of gaming libraries like pygame in Python.

Features

Four Classic Games:

- Snake Game: Guide a snake to eat food and grow longer, avoiding self-collision and the game boundary.
- Tic Tac Toe: A simple yet challenging two-player game where the goal is to align three Xs or Os in a row.
- Rock Paper Scissors: A game of chance where players choose between rock, paper, or scissors to compete against the computer.
- Space Invaders: An arcade shooter game where players must protect their base by shooting down advancing alien enemies.

Player Registration and Profile Creation:

- Players can add their names to the database, creating personalized profiles for better engagement and score tracking.
- Score Tracking and Database Integration:
- Each player’s scores are individually stored in an SQLite database, allowing them to track performance over time.

Profile Editing and Management:

- Users can edit player details, ensuring the database reflects current and accurate player information.
- User-Friendly Interface:
- The software is designed with an easy-to-navigate interface, enabling users to switch between games, view scores, and manage profiles effortlessly.

Technologies Used

- **Programming Language:** Python for overall code structure and game logic.
- **Game Development Library:** pygame for creating and handling game mechanics, animations, and user interactions.
- **Graphical User Interface (GUI):** tkinter for designing a visually appealing and user-friendly interface.
- **Database Management:** SQLite for handling player data and score records in a lightweight, accessible format.
- **Data Handling Tools:** Python’s built-in sqlite3 module for database operations such as adding, updating, and retrieving data.
- **Version Control:** Git for code versioning and collaboration.
- **Additional Libraries:** Various Python libraries for handling input/output, timers, and data manipulation.

How It Works

User Registration:

- Users register their names via the application's interface, creating a personalized profile that is saved in the SQLite database.

Game Selection Menu:

- A main menu allows users to choose from the available games: Snake Game, Tic Tac Toe, Rock Paper Scissors, or Space Invaders.

Gameplay:

- Each game launches in its dedicated window or section, using pygame for smooth and interactive gameplay. Players can interact using the keyboard or mouse, depending on the game.

Score Recording:

- After completing a game session, scores are automatically recorded in the SQLite database. Each entry is linked to the player's name, providing a comprehensive record of gameplay history.

Profile Management:

- Players can update their profile information directly from the interface, allowing for quick edits to names or game data as needed.

Data Retrieval:

- The software retrieves and displays player scores upon request, allowing users to view their achievements and progress over time.

Future Enhancements

- **Online Multiplayer Capabilities:** Expand the platform to support online play, allowing users to compete against friends or other players remotely.
- **Leaderboards:** Introduce a global or local leaderboard feature that displays the highest scores, motivating players to improve and compete.
- **Additional Game Titles:** Add more games, including both classic and new original games, to increase the software's variety and appeal.
- **Improved Graphics and Sound Effects:** Upgrade game visuals and sound effects for a more immersive experience using advanced libraries.
- **User Authentication System:** Implement a secure login and user authentication system to protect data and ensure privacy.
- **Cross-Platform Support:** Develop a mobile-friendly version of the software to expand usability and access.
- **Achievements and Rewards:** Include an achievement system that unlocks rewards or badges for players when they reach certain milestones.

Main Project

- SOURCE CODE

```python
import mysql.connector as sqltor
mycon=sqltor.connect(host='localhost',user='root',password='Pass@25het',database='project_game')
cursor=mycon.cursor()



#-----------------------------------------------------------------------------------------------------------
#functions

def create():
    #To create databases
    cursor.execute('create table Game_score(Sno int primary key,Name char(40),Game1 int,Game2 int,Game3 int,Game4 int)')
    #mycon.close()
#create()

def show():
    #To show the database
    cursor.execute('select * from Game_score')
    data=cursor.fetchall()
    for row in data:
        for col in row:
            print(col,end='|')
        print()

def delete():
    # to delete a data
    show()
    pd=int(input('Enter id to be deleted:'))
    inp=(pd,)
    cursor.execute("delete from Game_score where Sno='%s'",inp)
    mycon.commit()

def insert():
    #To check if new player
    global p
    print()
    name=input('Enter name:')
    i=input('Are you new player(y/n):')
    if i=='y':
        p=int(input('Enter your id:'))
        cursor.execute('insert into Game_score(Sno,Name,Game1,Game2,Game3,Game4) values({},"{}",{},{},{},{})'.format(p,name,0,0,0,0))
        mycon.commit()
    else:
        p=int(input('Enter your id:'))

#-----------------------------------------------------------------------------------------------------------

```

```
49
50  # 1 in 5 game
51  print("_____Retro Arcade_____")
52  show()
53  insert()
54  #-------------------------------------------------------------------------------------------------------
55
56  # main database menu
57
58  def main_database():
59      while True:
60          print('-------------------------------------------------------------------------------------------------------------------------------')
61          print('1.To show the database')
62          print('2.To insert new data')
63          print('3.To delete a data')
64          print('0.To go back to main menu')
65          mr=int(input('Enter your choice: '))
66          if mr==1:
67              print()
68              show()
69          elif mr==2:
70              print()
71              insert()
72          elif mr==3:
73              print()
74              delete()
75          elif mr==0:
76              print()
77              break
78
79
80
81  #-------------------------------------------------------------------------------------------------------
82
83  #main game menu
84  def main_game():
85      maingameloop=True
86      while maingameloop:
87          #menu
88          print('-------------------------------------------------------------------------------------------------------------------------------')
89          print("_____Retro Arcade_____")
90          print("1. Snake Game")
91          print("2. Rock Paper Scissor")
92          print("3. Tik tac toe")
93          print("4. Space Invaders")
94          print('0.To go to main menu')
```

```
96          uc = int(input("Insert the game choice you want to play: "))
97
98          #to end program
99          if uc ==0:
100             maingameloop=False
101
102
103         #Game 1 code
104         if uc == 1:
105             snake()
106         elif uc == 2:
107             ropasc()
108         elif uc ==3:
109             tictactoe()
110         elif uc==4:
111             spacein()
112
113 def snake():
114     print()
115     # importing libraries
116     import pygame
117     import time
118     import random
119
120     snake_speed = 15
121
122      # Window size
123     window_x = 720
124     window_y = 480
125
126     # defining colors
127     black = pygame.Color(0, 0, 0)
128     white = pygame.Color(255, 255, 255)
129     red = pygame.Color(255, 0, 0)
130     green = pygame.Color(0, 255, 0)
131     blue = pygame.Color(0, 0, 255)
132
133     # Initialising pygame
134     pygame.init()
```

```
    # Initialise game window
    pygame.display.set_caption('Snake game')
    game_window = pygame.display.set_mode((window_x, window_y))

    # FPS (frames per second) controller
    fps = pygame.time.Clock()

    # defining snake default position
    snake_position = [100, 50]

    # defining first 4 blocks of snake body
    snake_body = [[100, 50],
                  [90, 50],
                  [80, 50],
                  [70, 50]
                  ]
    # fruit position
    fruit_position = [random.randrange(1, (window_x//10)) * 10,
                      random.randrange(1, (window_y//10)) * 10]

    fruit_spawn = True

    # setting default snake direction towards
    # right
    direction = 'RIGHT'
    change_to = direction

    # initial score
    score_1 = 0

    # displaying Score function
    def show_score(choice, color, font, size):

        # creating font object score_font
        score_font = pygame.font.SysFont(font, size)

        # create the display surface object
        # score_surface
        score_surface = score_font.render('Score : ' + str(score_1), True, color)

        # create a rectangular object for the text
        # surface object
        score_rect = score_surface.get_rect()

        # displaying text
        game_window.blit(score_surface, score_rect)

    # game over function
    def game_over():

        # creating font object my_font
        my_font = pygame.font.SysFont('times new roman', 50)
        my_font_2=pygame.font.SysFont('times new roman', 20)
        my_font_3=pygame.font.SysFont('times new roman', 20)

        # creating a text surface on which text
        # will be drawn
        game_over_surface = my_font.render('Your Score is : '+  str(score_1), True, red )
        game_over_surface_2 = my_font_2.render('Created by Het patel ', True, red )
        game_over_surface_3 = my_font_3.render('Retro games created by  Utkarshsinh Jadeja, Meet Patel, Het Patel', True, red )
        print(score_1)
        inp=(score_1,p)
        cursor.execute('update Game_score set Game1=Game1+%s where Sno=%s',inp)
        mycon.commit()

        # create a rectangular object for the text

        # surface object
        game_over_rect = game_over_surface.get_rect()
        game_over_rect_2 = game_over_surface_2.get_rect()
        game_over_rect_3 = game_over_surface_3.get_rect()

        # setting position of the text
        game_over_rect.midtop = (window_x/2, window_y/4)
        game_over_rect_2.midtop = (window_x/2, window_y/2.5)
        game_over_rect_3.midtop = (window_x/2, window_y/2)

        # blit will draw the text on screen
        game_window.blit(game_over_surface, game_over_rect)
        game_window.blit(game_over_surface_2, game_over_rect_2)
        game_window.blit(game_over_surface_3, game_over_rect_3)

        pygame.display.flip()

        # after 2 seconds we will quit the program
        time.sleep(2)

        # deactivating pygame library
        pygame.quit()

        # quit the program
```

```
    while True:

        # handling key events
        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_UP:
                    change_to = 'UP'
                if event.key == pygame.K_DOWN:
                    change_to = 'DOWN'
                if event.key == pygame.K_LEFT:
                    change_to = 'LEFT'
                if event.key == pygame.K_RIGHT:
                    change_to = 'RIGHT'

        # If two keys pressed simultaneously
        # we don't want snake to move into two
        # directions simultaneously
        if change_to == 'UP' and direction != 'DOWN':
            direction = 'UP'
        if change_to == 'DOWN' and direction != 'UP':
            direction = 'DOWN'
        if change_to == 'LEFT' and direction != 'RIGHT':
            direction = 'LEFT'
        if change_to == 'RIGHT' and direction != 'LEFT':
            direction = 'RIGHT'

        # Moving the snake
        if direction == 'UP':
            snake_position[1] -= 10
        if direction == 'DOWN':
            snake_position[1] += 10
        if direction == 'LEFT':
            snake_position[0] -= 10
        if direction == 'RIGHT':
            snake_position[0] += 10

        # Snake body growing mechanism
        # if fruits and snakes collide then scores
        # will be incremented by 10
        snake_body.insert(0, list(snake_position))
        if snake_position[0] == fruit_position[0] and snake_position[1] == fruit_position[1]:
            score_1 += 10
            fruit_spawn = False
        else:
            snake_body.pop()
```

```
        if not fruit_spawn:
            fruit_position = [random.randrange(1, (window_x//10)) * 10,
                              random.randrange(1, (window_y//10)) * 10]

        fruit_spawn = True
        game_window.fill(black)

        for pos in snake_body:
            pygame.draw.rect(game_window, green,
                             pygame.Rect(pos[0], pos[1], 10, 10))
        pygame.draw.rect(game_window, white, pygame.Rect(
            fruit_position[0], fruit_position[1], 10, 10))

        # Game Over conditions
        if snake_position[0] < 0 or snake_position[0] > window_x-10:
            game_over()
            break
        if snake_position[1] < 0 or snake_position[1] > window_y-10:
            game_over()
            break

        # Touching the snake body
        for block in snake_body[1:]:
            if snake_position[0] == block[0] and snake_position[1] == block[1]:

                game_over()
                break

        # displaying score countinuously
        show_score(1, white, 'times new roman', 20)

        # Refresh game screen
        pygame.display.update()

        # Frame Per Second /Refresh Rate
        fps.tick(snake_speed)
```

```
def ropasc():
    print()
    #This is a rock paper scissors game
    import random
    cm=["Rock","Paper","Scissors"]
    os=0
    score_2=0
    n=int(input("How many rounds do you want to play ?"))
    for x in range(n):
        print("Press 'r' for choosing Rock !")
        print("Press 'p' for choosing Paper !")
        print("Press 's' for choosing Scissors !")
        m=input("What is your move ?")
        c=random.choice(cm)
        print("Opponenet's move:",c)
        if m=="r" and c=="Paper" :
            print("Opponent wins !")
            os=os+1
        elif m=="r" and c=="Rock":
                print("It's a Draw !")
        elif m=="r" and c=="Scissors":
                print("You win !")
                score_2=score_2+1

        elif m=="p" and c=="Scissors" :
            print("Opponent wins !")
            os=os+1
        elif m=="p" and c=="Paper":
                print("It's a Draw !")
        elif m=="p" and c=="Rock":
                print("You win !")
                score_2=score_2+1

        elif m=="s" and c=="Rock" :
            print("Opponent wins !")
            os=os+1
        elif m=="s" and c=="Scissors":
            print("It's a Draw !")
        elif m=="s" and c=="Paper":
            print("You win !")
            score_2=score_2+1
```

```

        elif m!="s" and m!="p" and m!="r" :
            print("Press 'y' for Yes")
            print("Press 'n' for No")
            o=input("Invalid move ! Are you SURE you want to play a game ?")

            if o=='y':
                print("Then play properly!")
                n=n+1
            else :
                print("OK ! Goodbye ! ")
                break
    print("Opponenet's Score : ",os)
    print("Your Score : ",score_2)
    if os>score_2:
        print("The Opponent has won!")
    elif os==score_2:
        print("The game was a Draw!")
    else:
        print("You won the game!")
    inp=(score_2,p)
    cursor.execute('update Game_score set Game2=Game2+%s where Sno=%s',inp)
    mycon.commit()
    print('Created by Utkarshsinh Jadeja')

        #-----------------------------------------------------------------------------------------------------------------------
```

```
def tictactoe():
    print()
    board=['-','-','-','-','-','-','-','-','-']
    n1 = input("Insert name for player 1: ")
    n2 = 'ai'
    score_3=0
    import random
    cm=[0,1,2,3,4,5,6,7,8]


    def dis():
        print('|'+board[0]+'|'+board[1]+'|'+board[2]+'|')
        print('|'+board[3]+'|'+board[4]+'|'+board[5]+'|')
        print('|'+board[6]+'|'+board[7]+'|'+board[8]+'|')

    def check(board):
        p1='x'
        p2='o'
        if board[0] == board[1] == board[2]==p1 or board[0] == board[1] == board[2] == p2:
            return True
        elif board[3] == board[4] == board[5]==p1 or board[3] == board[4] == board[5]==p2:
            return True
        elif board[6] == board[7] == board[8]==p1 or board[6] == board[7] == board[8]==p2:
            return True
        elif board[0] == board[3] == board[6]==p1 or board[0] == board[3] == board[6]==p2:
            return True
        elif board[1] == board[4] == board[7]==p1 or board[1] == board[4] == board[7]==p2:
            return True
        elif board[2] == board[5] == board[8]==p1 or board[2] == board[5] == board[8]==p2:
            return True
        elif board[0] == board[4] == board[8]==p1 or board[0] == board[4] == board[8]==p2:
            return True
        elif board[2] == board[4] == board[6]==p1 or board[2] == board[4] == board[6]==p2:
            return True
        else:
            return False

    def inp1(board):
        print('now its your turn',n1)
        x=int(input("Enter the position:"))
        if board[x-1]!='-':
            print('value already exist')
            return inp1(board)
        else:
```

```
            return x

    def inp2(board):
        print('now its your turn',n2)
        c=random.choice(cm)
        if board[c-1]!='-':
            print('value already exist')
            return inp2(board)
        else:
            return c


    for m in range(9):
        if m%2==0:
            x=inp2(board)
            board[x-1]='o'
            dis()
            cm.remove(x)
            if check(board):
                print(n2,'win')
                break

        else:

            x=inp1(board)
            board[x-1]='x'
            dis()
            cm.remove(x)
            if check(board):
                print(n1,'win')
                score_3= score_3 + 1
                break

    inp=(score_3,p)
    cursor.execute('update Game_score set Game4=Game4+%s where Sno=%s',inp)
    mycon.commit()
    print('Created by Utkarshsinh Jadeja')
    print('game over')

#----------------------------------------------------------------------------------------------------------------------
```

```
    #Game 4
    #Space shooter
    elif uc==4:

        import os
        import turtle
        import math
        import random
        import platform
        #if on windows you import win sound or yet,just use linux
        if platform.system()=='Windows':
            try:
                import winsound
            except:
                print('winsound module not available')


        wn= turtle.Screen()
        wn.title("shooter space")
        wn.bgcolor('black')
        wn.setup(width=800, height=600)
        try:
            wn.bgpic('K:\CLASS 12A\SPACE SHOOTER\space_background.gif')
        except:
            print()
        wn.tracer(0)

        #register the shape
        try:
            turtle.register_shape('K:\CLASS 12A\SPACE SHOOTER\spaceinvader3.gif')
        except:
            print()
        try:
            turtle.register_shape('K:\CLASS 12A\SPACE SHOOTER\spaceship2.gif')
        except:
            print()

        #Draw border
        border_pen=turtle.Turtle()
        border_pen.speed(0)
        border_pen.color('white')
        border_pen.penup()
        border_pen.setposition(-300,-300)
        border_pen.pendown()
        border_pen.pensize(3)
```

```
        for side in range(4):
            border_pen.fd(600)
            border_pen.lt(90)
            border_pen.hideturtle()

        # Score
        score_4 = 0

        # Paddle A
        player = turtle.Turtle()
        player.speed(0)
        try:
            player.shape('K:\CLASS 12A\SPACE SHOOTER\spaceship2.gif')
        except:
            player.shape('triangle')
        player.color('white')
        player.setheading(90)
        player.penup()
        player.goto(0,-250)




        #no of enemies
        number_of_enemies=20
        enemies=[]

        enemy_start_x=-200
        enemy_start_y=250
        enemy_number=0
```

```
        #add enemies in list

        for i in range(number_of_enemies):
            enemies.append(turtle.Turtle())


        for enemy in enemies:
            enemy.color('blue')
            try:
                enemy.shape('K:\CLASS 12A\SPACE SHOOTER\spaceinvader3.gif')
            except:
                enemy.shape('square')
            enemy.penup()
            enemy.speed(0)
            x= enemy_start_x + (50*enemy_number)
            y= enemy_start_y
            enemy.setposition(x,y)
            #update enemy number
            enemy_number+=1
            if enemy_number==10:
                enemy_start_y-=50
                enemy_number=0

        enemyspeed=0.1

        # Pen
        pen = turtle.Turtle()
        pen.speed(0)
        pen.color('white')
        pen.penup()
        pen.hideturtle()
        pen.goto(0,260)
        pen.write('Score: 0', align ='center' , font=('Courier', 20 , 'normal'))

        # Pen2
        pen2=turtle.Turtle()
        pen2.speed(0)
        pen2.penup()
        pen2.color('white')
        pen2.hideturtle()
        pen2.goto(0,240)
        pen2.write('Created by Meet patel', align ='center' , font=('Courier', 10 , 'normal'))
```

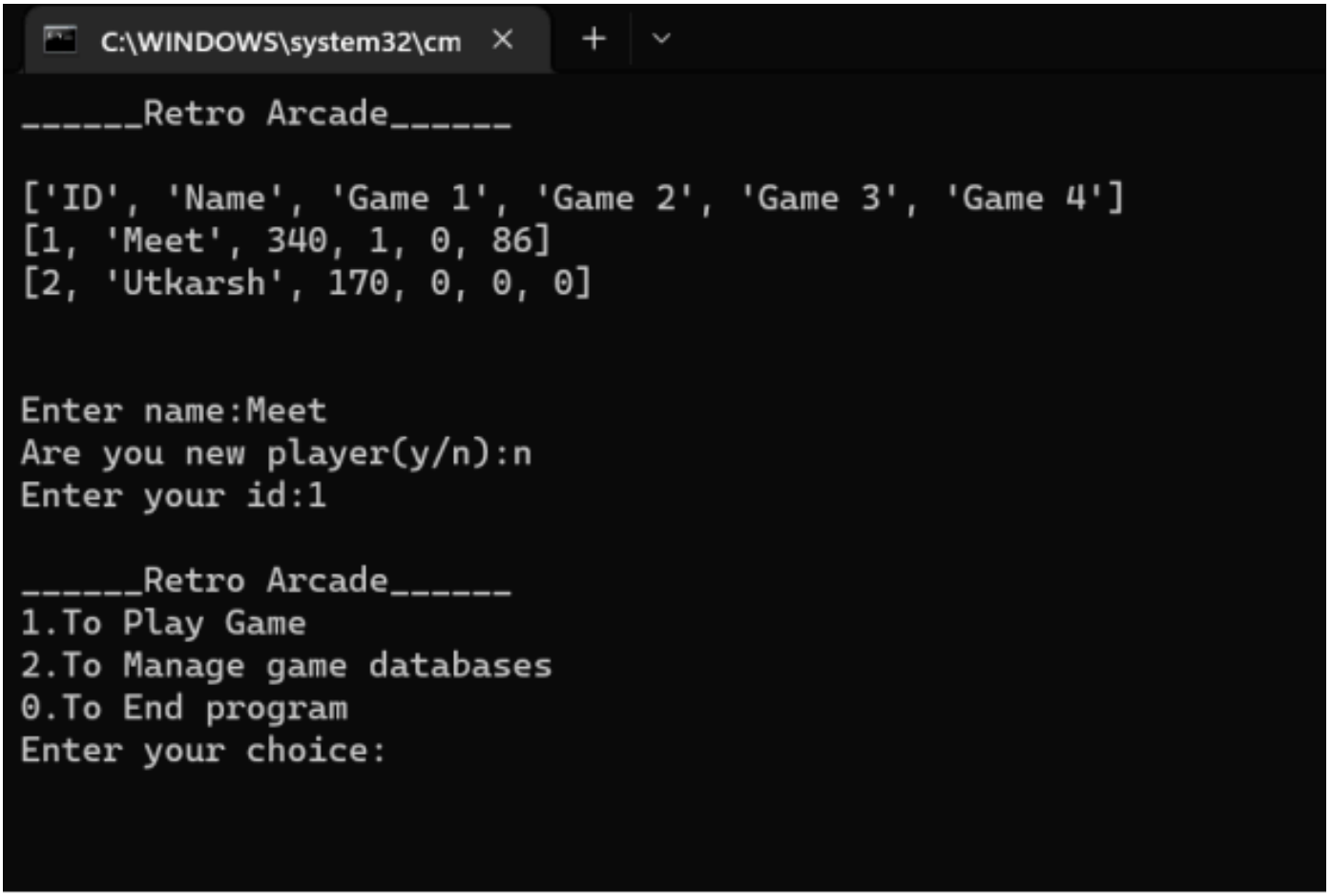
```
        # Pen3
        pen3=turtle.Turtle()
        pen3.speed(0)
        pen3.penup()
        pen3.color('white')
        pen3.hideturtle()
        pen3.goto(10,-290)
        pen3.write('Retro games created by  Utkarshsinh Jadeja, Meet Patel, Het Patel', align ='center' , font=('Courier', 10 , 'norma

        #create the player's bullet
        bullet= turtle.Turtle()
        bullet.color('yellow')
        bullet.shape('triangle')
        bullet.penup()
        bullet.speed(0)
        bullet.setheading(90)
        bullet.shapesize(0.5,0.5)
        bullet.hideturtle()

        bulletspeed=1

        #define bullet state
        #ready -ready to fire
        #fire - bullet is firing
        bulletstate = 'ready'


        #Function
        def play_sound(sound_file,time=0):
            #windows
            if platform.system()=='Windows':
                winsound.PlaySound(sound_file,winsound.SND_ASYNC)
            #linux
            elif platform.system()=='Linux':
                os.system('aplay -q{}&'.format(sound_file))
```

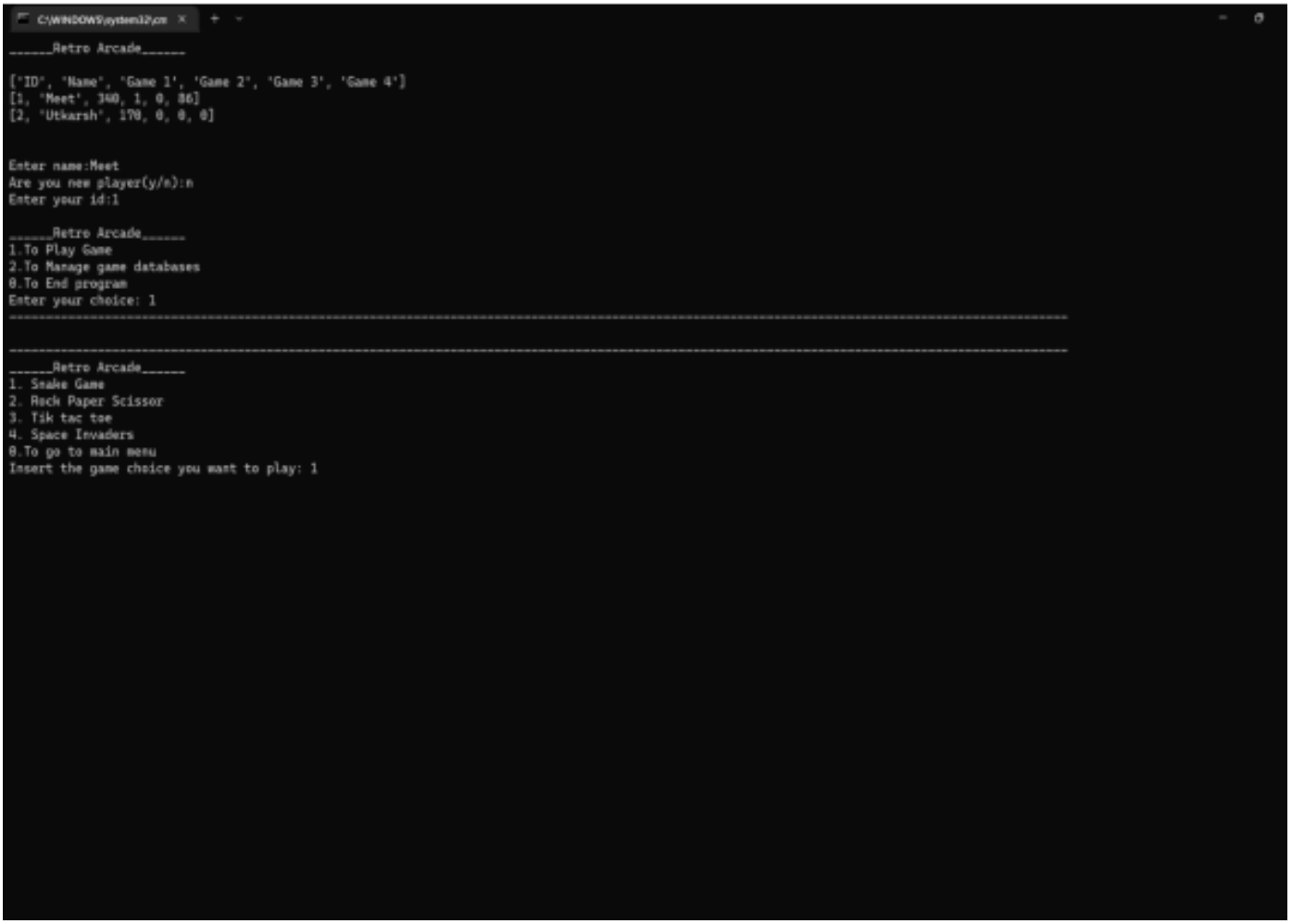
```
        def player_left():
            x=player.xcor()
            x -=25
            if x<-280:
                x=-280
            player.setx(x)

        def player_right():
            x=player.xcor()
            x +=25
            if x>280:
                x=280
            player.setx(x)

        def fire_bullet():
            #declare bullet state as a global if it needs changed
            global bulletstate
            if bulletstate =='ready':
                bulletstate= 'fire'
                #move the bullet to above
                play_sound('K:\CLASS 12A\SPACE SHOOTER\shoot.wav')
                x= player.xcor()
                y= player.ycor() +10
                bullet.setposition(x,y)
                bullet.showturtle()

        def isCollision(t1,t2):
            distance=math.sqrt(math.pow(t1.xcor()-t2.xcor(),2)+math.pow(t1.ycor()-t2.ycor(),2))
            if distance >15:
                return False
            else:
                return True


        def quit():
            global running
            running=False

        #Keyboard binding
        wn.listen()
        wn.onkeypress( player_left, 'Left')
        wn.onkeypress( player_right, 'Right')
        wn.onkeypress(fire_bullet, 'space')
        wn.onkeypress(quit,'q')
```

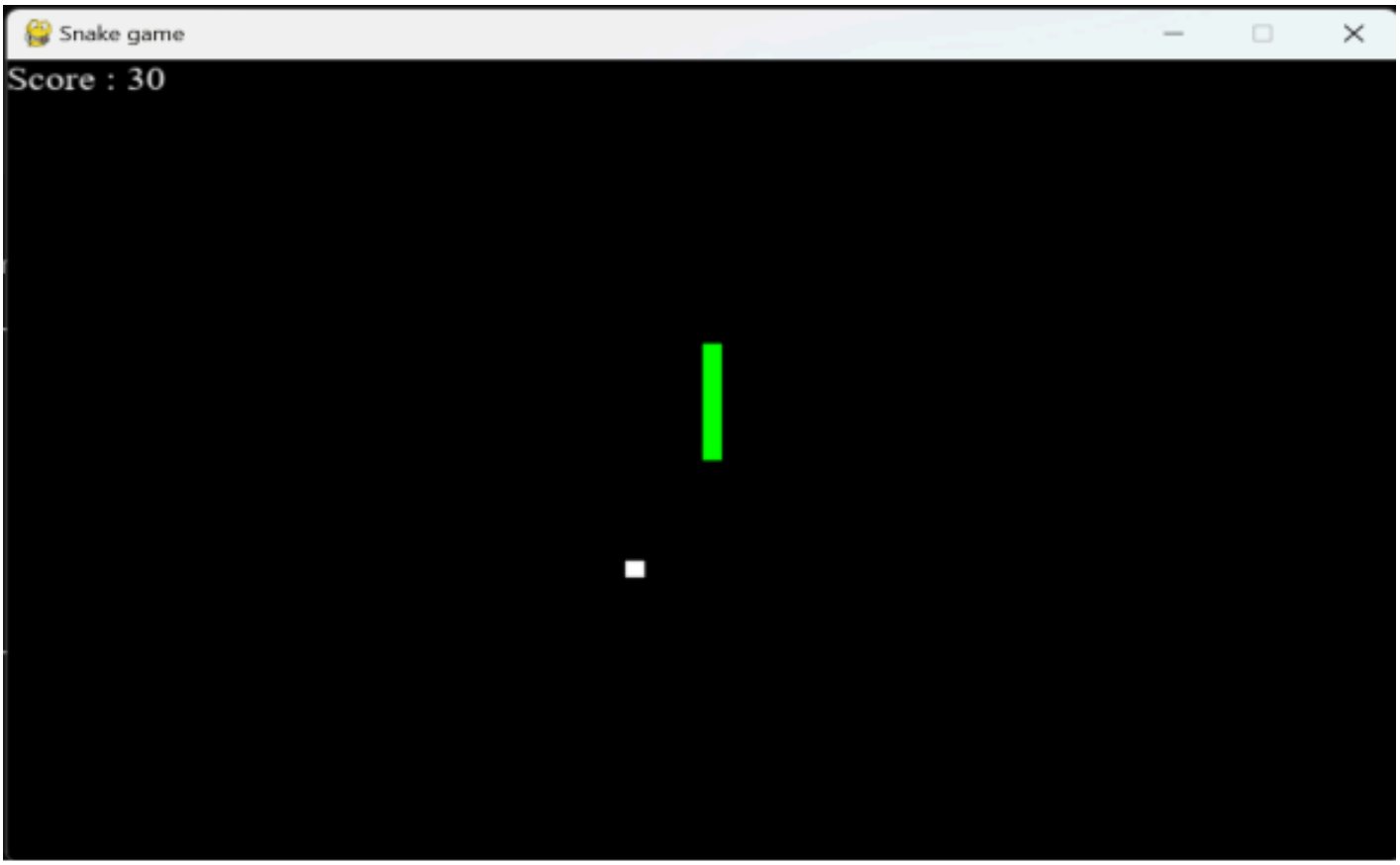
```
        running=True
        #Main game loop
        while running:
            wn.update()

            for enemy in enemies:
                #move the enemy
                x=enemy.xcor()
                x+=enemyspeed
                enemy.setx(x)


                #move enemy back and forth
                if enemy.xcor()>280:
                    for e in enemies:

                        y=e.ycor()
                        y-=40
                        e.sety(y)
                    #change enemy direction
                    enemyspeed*=-1

                if enemy.xcor()<-280:
                    for e in enemies:
                        y=e.ycor()
                        y-=40
                        e.sety(y)
                    #change enemy direction
                    enemyspeed*=-1


                #check for a collision between the bullet and the eemy
                if isCollision(bullet,enemy):
                    play_sound('K:\CLASS 12A\SPACE SHOOTER\invaderkilled.wav')
                    #reset the bullet
                    bullet.hideturtle()
                    bulletstate ='ready'
                    bullet.setposition(0,-400)
                    score_4 += +5
                    pen.clear()
                    pen.write('Score: {}'.format(score_4), align ='center' , font=('Courier', 20 , 'normal'))

                    #reset the enemy

                    enemy.setposition(0,50000)
```

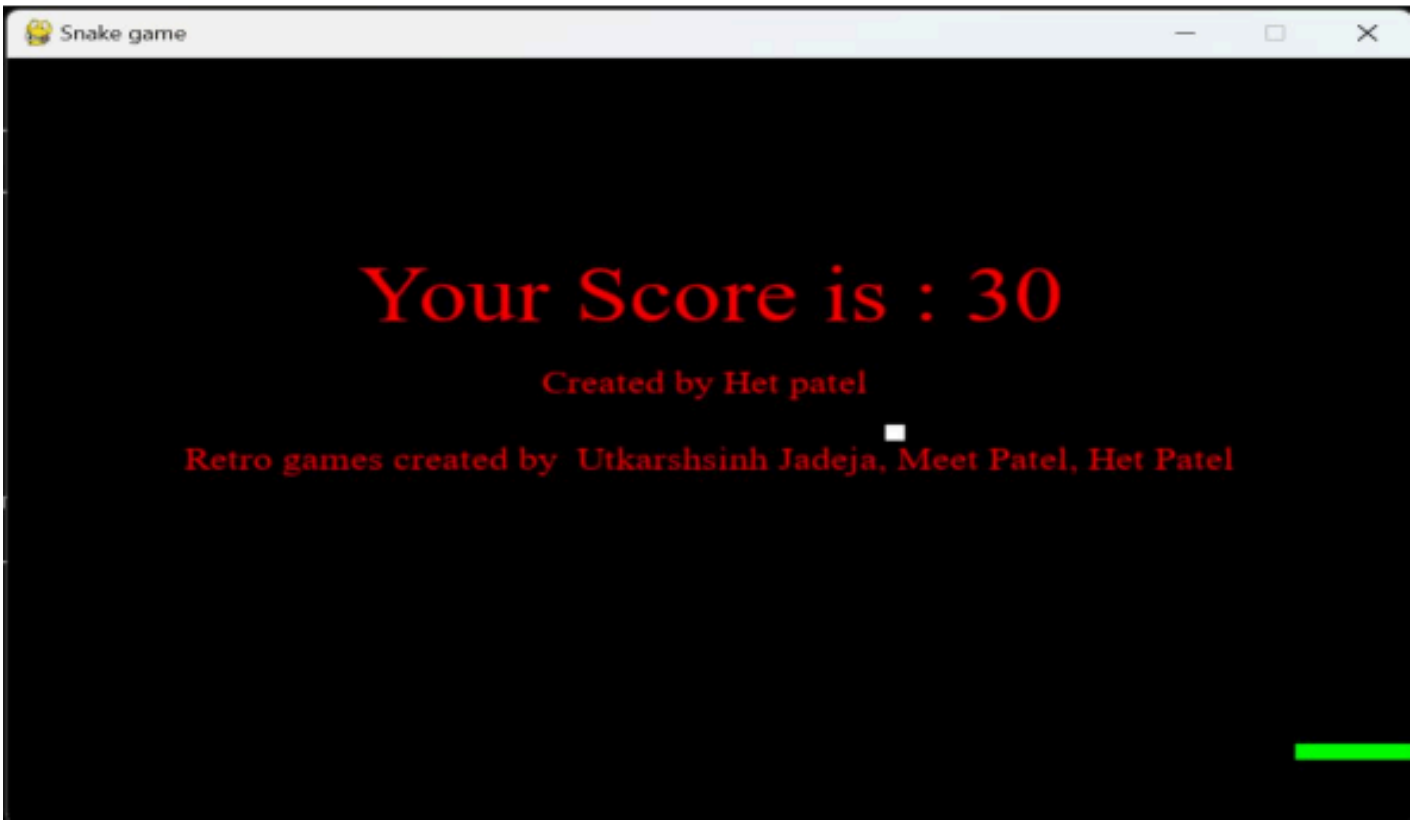
```
            if isCollision(player,enemy):
                play_sound('k:\CLASS 12A\SPACE SHOOTER\explosion.wav')
                player.hideturtle()
                enemy.hideturtle()
                print('Game over, Thank you for playing')
                break


            #move the bullet
            if bulletstate=='fire':
                y= bullet.ycor()
                y +=bulletspeed
                bullet.sety(y)

            #check to see if the bullet has gone to the top
            if bullet.ycor()> 275:
                bullet.hideturtle()
                bulletstate= 'ready'

            #Border checking

            if bullet.ycor() > 275:
                bullet.hideturtle()
                bulletstate='ready'
                score_4 += -1
                pen.clear()
                pen.write('Score: {}'.format(score_4), align ='center' , font=('Courier', 20 , 'normal'))
                x= player.xcor()
                y= player.ycor() +10
                bullet.setposition(x,y)
        wn.bye()
        inp=(score_4,p)
        cursor.execute('update Game_score set Game4=Game4+%s where Sno=%s',inp)
        mycon.commit()

    cursor.execute('select *  from Game_score')
    data=cursor.fetchall()
    for row in data:
        for col in row:
            print(col,end='|')
        print()

print('-------------------------------------------------------------------------------------------------------------------------------------------')
print('Game over!!,Thanks for playing')
print('Created by Utkarshsinh Jadeja, Meet Patel, Het Patel, Student of Class 12A(Batch 2022-2023)')
```

- OUTPUT

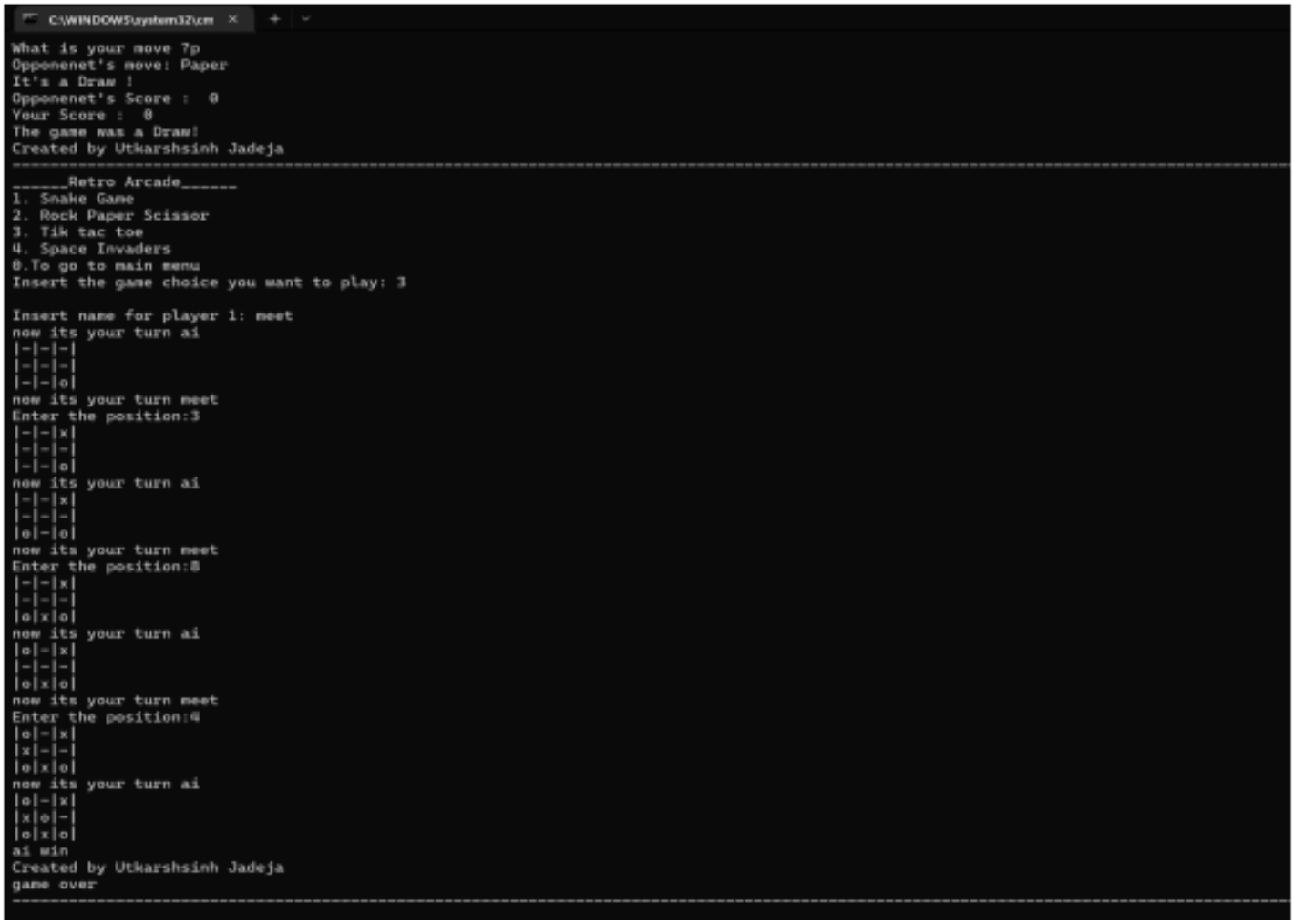

C:\WINDOWS\system32\cm
_____Retro Arcade_____
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 340, 1, 0, 86]
[2, 'Utkarsh', 170, 0, 0, 0]
Enter name:Meet
Are you new player(y/n):n
Enter your id:1
_____Retro Arcade_____
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice:


_____Retro Arcade_____
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 340, 1, 0, 86]
[2, 'Utkarsh', 170, 0, 0, 0]
Enter name:Meet
Are you new player(y/n):n
Enter your id:1
_____Retro Arcade_____
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 1
_____Retro Arcade_____
1. Snake Game
2. Rock Paper Scissor
3. Tik tac toe
4. Space Invaders
0.To go to main menu
Insert the game choice you want to play: 1


Snake game
Score : 30

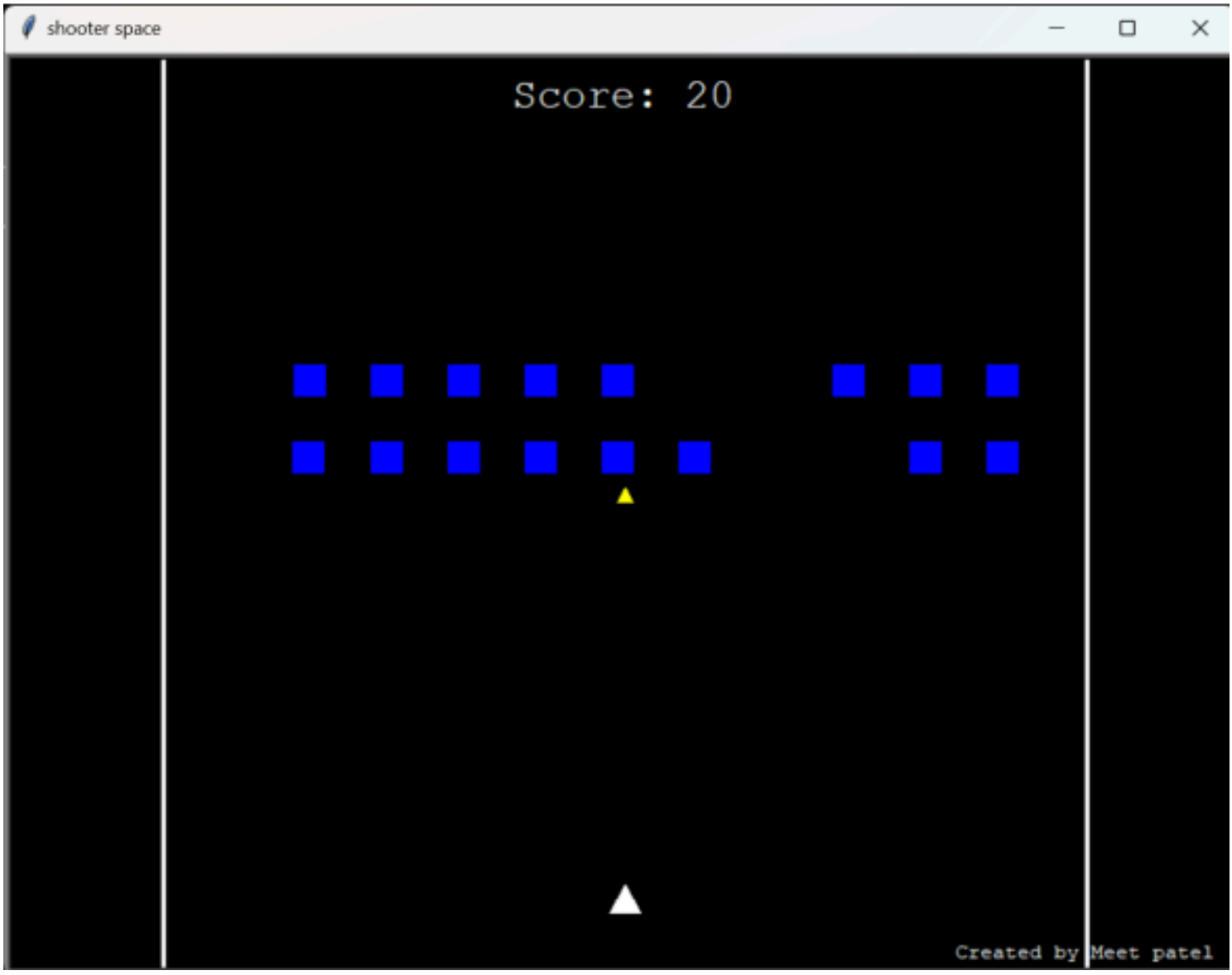

Snake game
Your Score is : 30
Created by Het patel
Retro games created by Utkarshsinh Jadeja, Meet Patel, Het Patel

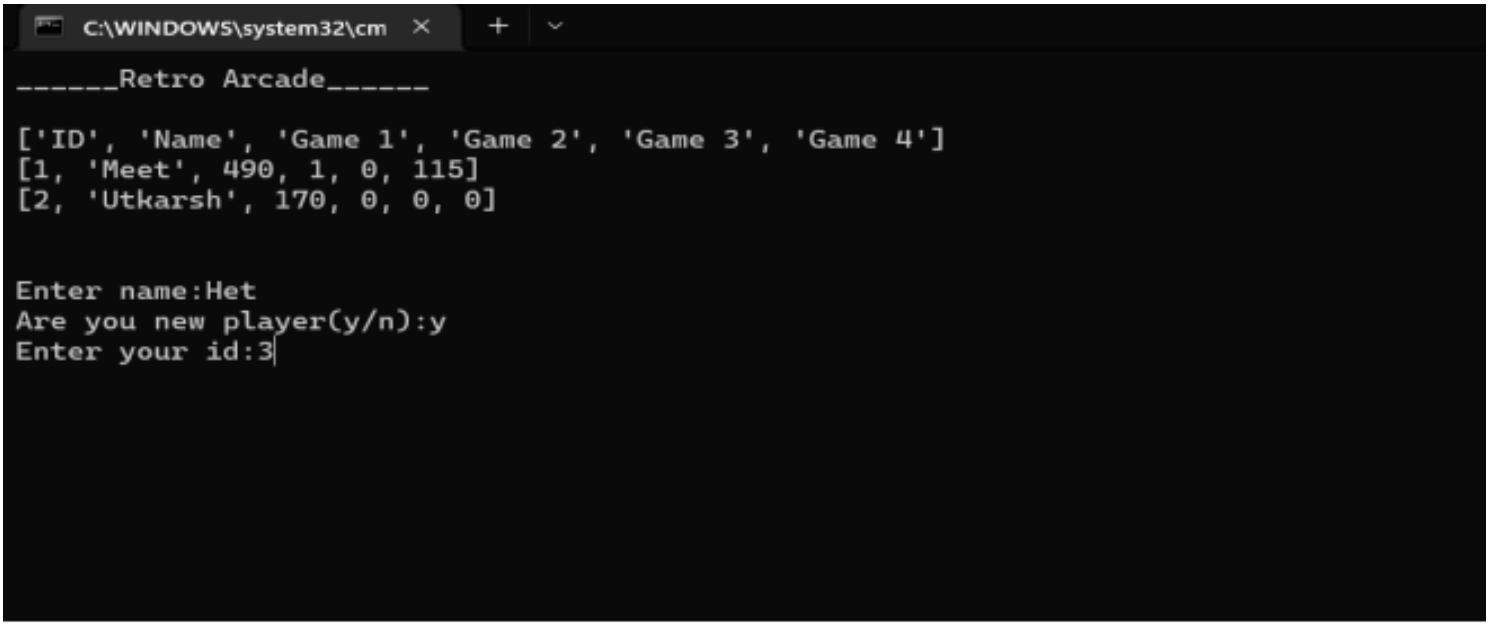
```
C:\WINDOWS\system32\cm
______Retro Arcade______

['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 86]
[2, 'Utkarsh', 170, 0, 0, 0]


Enter name:meet
Are you new player(y/n):n
Enter your id:1

______Retro Arcade______
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 1
------------------------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------
______Retro Arcade______
1. Snake Game
2. Rock Paper Scissor
3. Tik tac toe
4. Space Invaders
0.To go to main menu
Insert the game choice you want to play: 2

How many rounds do you want to play ?2
Press 'r' for choosing Rock !
Press 'p' for choosing Paper !
Press 's' for choosing Scissors !
What is your move ?r
Opponenet's move: Rock
It's a Draw !
Press 'r' for choosing Rock !
Press 'p' for choosing Paper !
Press 's' for choosing Scissors !
What is your move ?p
Opponenet's move: Paper
It's a Draw !
Opponenet's Score :  0
Your Score :  0
The game was a Draw!
Created by Utkarshsinh Jadeja
------------------------------------------------------------------------------------------------------------------------
______Retro Arcade______
1. Snake Game
2. Rock Paper Scissor
3. Tik tac toe
4. Space Invaders
0.To go to main menu
Insert the game choice you want to play:
```



```
C:\WINDOWS\system32\cm
What is your move ?p
Opponenet's move: Paper
It's a Draw !
Opponenet's Score :  0
Your Score :  0
The game was a Draw!
Created by Utkarshsinh Jadeja
------------------------------------------------------------------------------------------------------------------------
______Retro Arcade______
1. Snake Game
2. Rock Paper Scissor
3. Tik tac toe
4. Space Invaders
0.To go to main menu
Insert the game choice you want to play: 3

Insert name for player 1: meet
now its your turn ai
|-|-|-|
|-|-|-|
|-|-|o|
now its your turn meet
Enter the position:3
|-|-|x|
|-|-|-|
|-|-|o|
now its your turn ai
|-|-|x|
|-|-|-|
|o|-|o|
now its your turn meet
Enter the position:8
|-|-|x|
|-|-|-|
|o|x|o|
now its your turn ai
|o|-|x|
|-|-|-|
|o|x|o|
now its your turn meet
Enter the position:4
|o|-|x|
|x|-|-|
|o|x|o|
now its your turn ai
|o|-|x|
|x|o|-|
|o|x|o|
ai win
Created by Utkarshsinh Jadeja
game over
------------------------------------------------------------------------------------------------------------------------
```

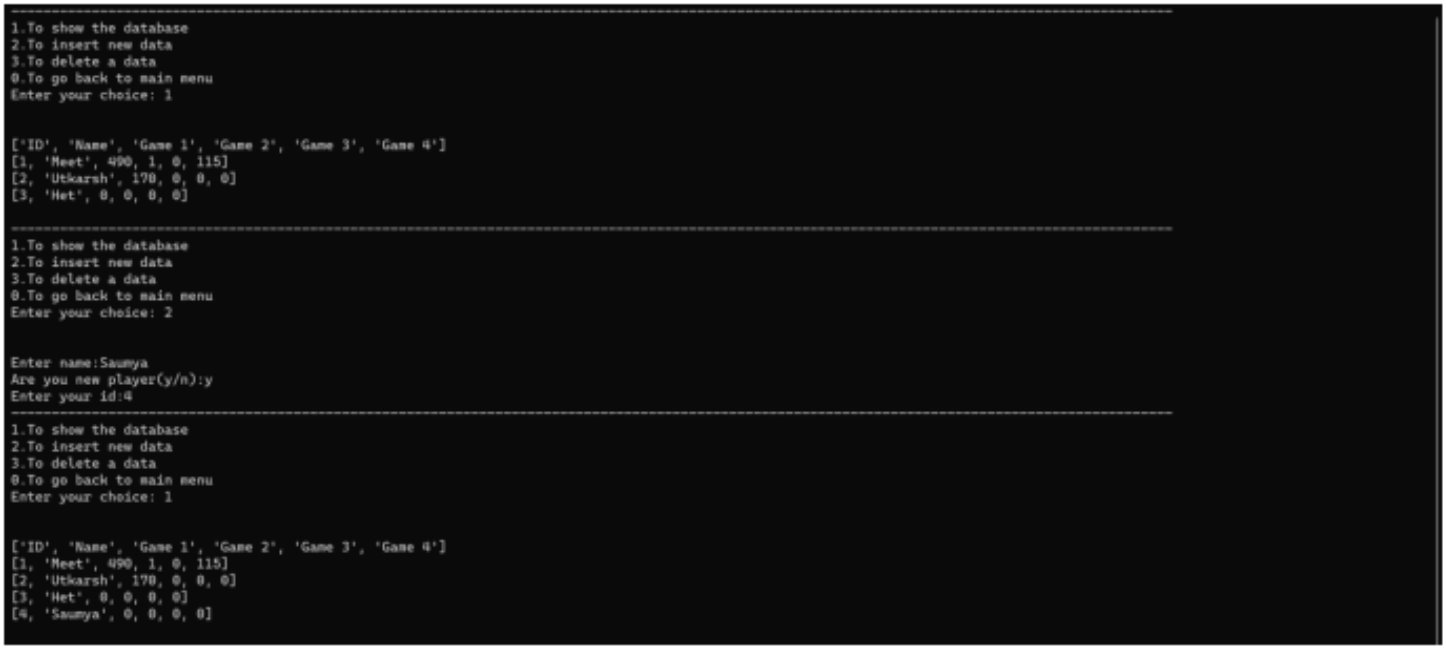
```
______Retro Arcade______

['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 86]
[2, 'Utkarsh', 170, 0, 0, 0]


Enter name:meet
Are you new player(y/n):n
Enter your id:1

______Retro Arcade______
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 1
--------------------------------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------------------------------------------
______Retro Arcade______
1. Snake Game
2. Rock Paper Scisser
3. Tik tac toe
4. Space Invaders
0.To go to main menu
Insert the game choice you want to play: 4
```




C:\WINDOWS\system32\cm
_____Retro Arcade_____
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
Enter name:Het
Are you new player(y/n):y
Enter your id:3


_____Retro Arcade_____
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
Enter name:Het
Are you new player(y/n):y
Enter your id:3
_____Retro Arcade_____
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 2
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice:


C:\WINDOWS\system32\cm
_____Retro Arcade_____
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
Enter name:Het
Are you new player(y/n):y
Enter your id:3
_____Retro Arcade_____
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 2
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice:


1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 2
Enter name:Saumya
Are you new player(y/n):y
Enter your id:4
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1
['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

```
--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 2


Enter name:Saumya
Are you new player(y/n):y
Enter your id:4
--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 3


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

Enter id to be deleted:3
--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice:
```

```
--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 3


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[3, 'Het', 0, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

Enter id to be deleted:3
--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

--------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice:
```

```
--------------------------------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 1


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]

--------------------------------------------------------------------------------------------------------------------------------------------------------
1.To show the database
2.To insert new data
3.To delete a data
0.To go back to main menu
Enter your choice: 0


['ID', 'Name', 'Game 1', 'Game 2', 'Game 3', 'Game 4']
[1, 'Meet', 490, 1, 0, 115]
[2, 'Utkarsh', 170, 0, 0, 0]
[4, 'Saumya', 0, 0, 0, 0]


______Retro Arcade______
1.To Play Game
2.To Manage game databases
0.To End program
Enter your choice: 0
--------------------------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------------------------------------
Game over!!,Thanks for playing
Created by Utkarshsinh Jadeja, Meet Patel, Het Patel, Student of Class 12A(Batch 2022-2023)

D:\CLASS 12A\retro game>
```

THANK YOU