Class Assignment (CA) 1 Report

on

SNAKE GAME

Submitted in partial fulfillment of the requirements for the award of degree

in the department of

Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

Submitted by:

Dev Munjal (Registration no: 12007537)

Ashish Sarswat (Registration no:12010488)

Section: K20TB

Date of submission: 17, November, 2021

Contents

S. No.	Topic name	Page No.
1)	Course overview (intro, abstract, requisites etc.)	3
2)	Syntax breakdown (importing module, creating game screen, food and snake etc.)	4-9
3)	Coding and game screenshots	10-16
4)	conclusion	17

Abstract

The snake game is a very popular and fun game. Every time the snake eats the fruit, its length grows longer that makes the game more difficult

INTRODUCTION

The objective of this python project is to build a snake game project. In this python project, the player has to move a snake so it touches the fruit. If the snake touches itself or the border of the game then the game will over.

Project Prerequisites

To build the snake game project we used the turtle module, random module, time module, and concept of python.

Turtle module gives us a feature to draw on a drawing board
Random module will be used to generate random numbers
Time module is an inbuilt module in python. It provides the functionality of time.
To install python modules, we use the pip install command in the command line:

```
1)pip install turtles
2)pip install random
```

Project File Structure

Steps to build a snake game project in python:

- Importing libraries
- Creating a game screen
- Creating snake and food
- Keyboard binding
- Game mainloop

COMPLETE SYNTAX:

1. Importing required module

```
import turtle

import random

import time
```

We require turtle, random, and time module to import

2. Creating game screen

```
screen = turtle.Screen()

screen.title('SNAKE GAME')

screen.setup(width = 700, height = 700)

screen.tracer(0)

turtle.bgcolor('Grey')

turtle.speed(5)

turtle.pensize(4)

turtle.penup()

turtle.goto(-310,250)

turtle.pendown()

turtle.color('Blue')

turtle.forward(600)

turtle.right(90)

turtle.forward(500)

turtle.right(90)

turtle.forward(600)

turtle.right(90)

turtle.forward(500)

turtle.penup()
```

Explanation

- **title()** will set the desired title of the screen
- **setup()** used to set the height and width of the screen
- **tracer(0)** will turn off the screen update
- **bgcolor()** will set the background color
- **forward()** will use to move the turtle in a forwarding direction for a specified amount
- **right()** used to turn the turtle clockwise and **left()** used to turn the turtle anticlockwise
- **penup()** will not draw while its move

3. Creating snake and food

```
snake = turtle.Turtle()

snake.speed(0)

snake.shape('square')

snake.color("black")

snake.penup()

snake.goto(0,0)

snake.direction = 'stop'

fruit = turtle.Turtle()

fruit.speed(0)

fruit.shape('circle')

fruit.color('red')

fruit.penup()

fruit.goto(30,30)

old_fruit=[]

scoring = turtle.Turtle()

scoring.speed(0)

scoring.color("black")

scoring.penup()

scoring.hideturtle()
scoring.goto(0,300)

scoring.write("Score :",align="center",font=("Courier",24,"bold"))
```

Explanation

- **Turtle()** will be used to create a new turtle object
- **hideturtle()** will use to hide the turtle
- **goto()** used to move the turtle at x and y coordinates

4. Keyboard binding

```
def snake_go_up():

if snake.direction != "down":

snake.direction = "up"

def snake_go_down():

if snake.direction != "up":

snake.direction = "down"

def snake_go_left():

if snake.direction != "right":

snake.direction = "left"

def snake_go_right():

if snake.direction != "left":

snake.direction = "right"

def snake_move():

if snake.direction == "up":

y = snake.ycor()

snake.sety(y + 20)

if snake.direction == "down":

y = snake.ycor()

snake.sety(y - 20)
```

```
if snake.direction == "left":
x = snake.xcor()
snake.setx(x - 20)
if snake.direction == "right":
x = snake.xcor()
snake.setx(x + 20)
screen.listen()
screen.onkeypress(snake_go_up, "Up")
screen.onkeypress(snake_go_down, "Down")
screen.onkeypress(snake_go_left, "Left")
screen.onkeypress(snake_go_right, "Right")
```

Explanation

screen.listen() function listen when key will press.

If the Up key will press then the snake will move in up direction.

If the Down key is pressed then the snake will move in the down direction.

If Left key will press then the snake will move in left direction.

If the Right key will press then the snake will move in the right direction

5. Snake and fruit collision

```
if snake.distance(fruit)< 20:
x = random.randint(-290,270)
y = random.randint(-240,240)
fruit.goto(x,y)
scoring.clear()
score+=1
scoring.write("Score:{}".format(score),align="center",font=("Courier",24,"bold"))
delay-=0.001
```

```
new_fruit = turtle.Turtle()
new_fruit .speed(0)
new_fruit .shape('square')
new_fruit .color('red')
new_fruit .penup()
old_fruit.append(new_fruit )
for index in range(len(old_fruit)-1,0,-1):
a = old_fruit[index-1].xcor()
b = old_fruit[index-1].ycor()
old_fruit[index].goto(a,b)
if len(old_fruit)>0:
a= snake.xcor()
b = snake.ycor()
old_fruit[0].goto(a,b)
snake_move()
```

Explanation

If the snake touches the fruit then the fruit will go at any random position and score will increase and the size of the snake will also increase

6. Snake and border collision

```
if snake.xcor()>280 or snake.xcor()< -300 or snake.ycor()>240 or snake.ycor()<-240:
time.sleep(1)
screen.clear()
screen.bgcolor('Red')
scoring.goto(0,0)
scoring.write(" GAME OVER \n Your Score is {}".format(score),align="center",font=("Courier",30,"bold"))
```

Explanation

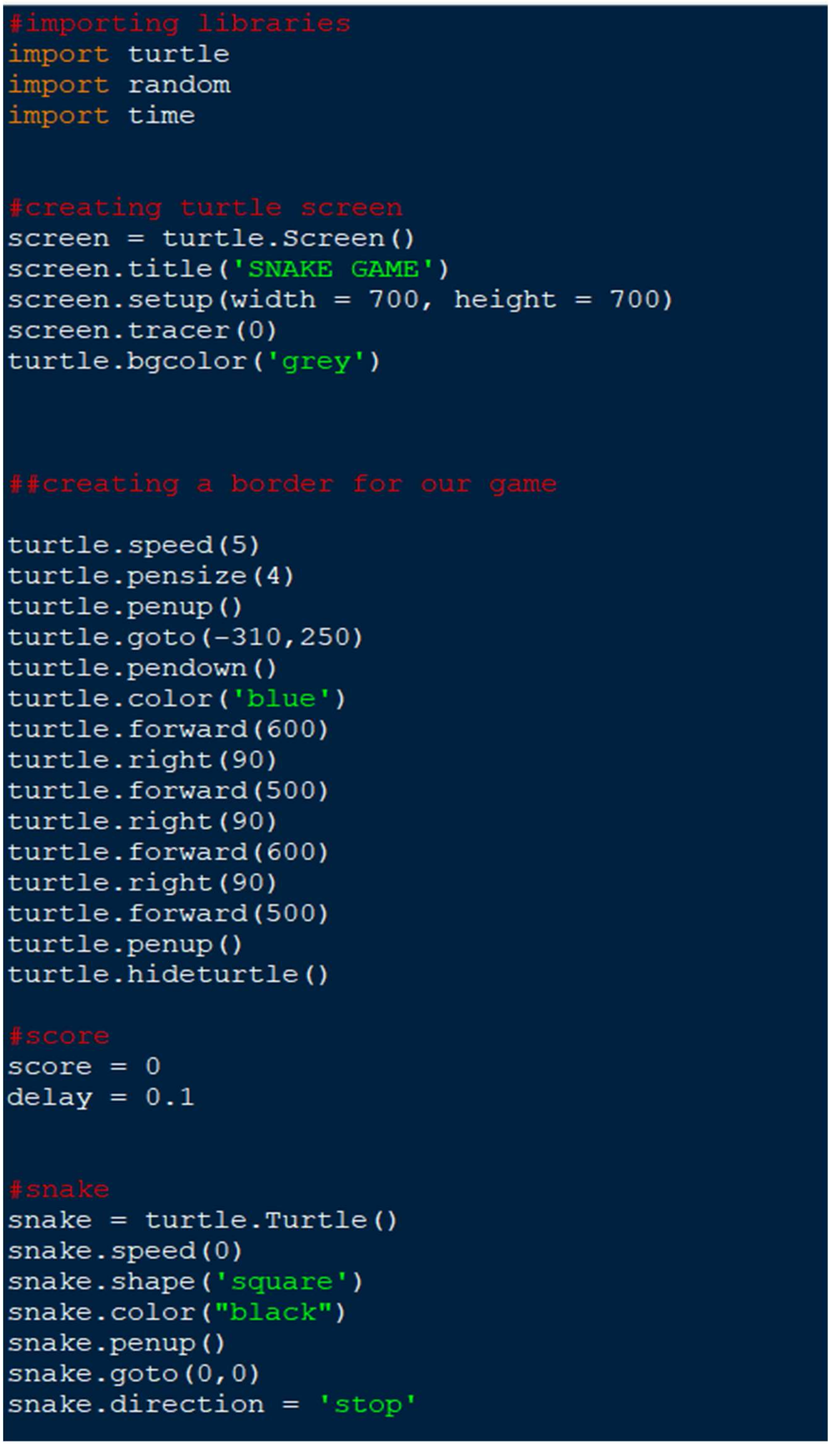
If the snake touches the border of the game then the game will over.

screen.clear() will delete all the drawing of the turtle on the screen

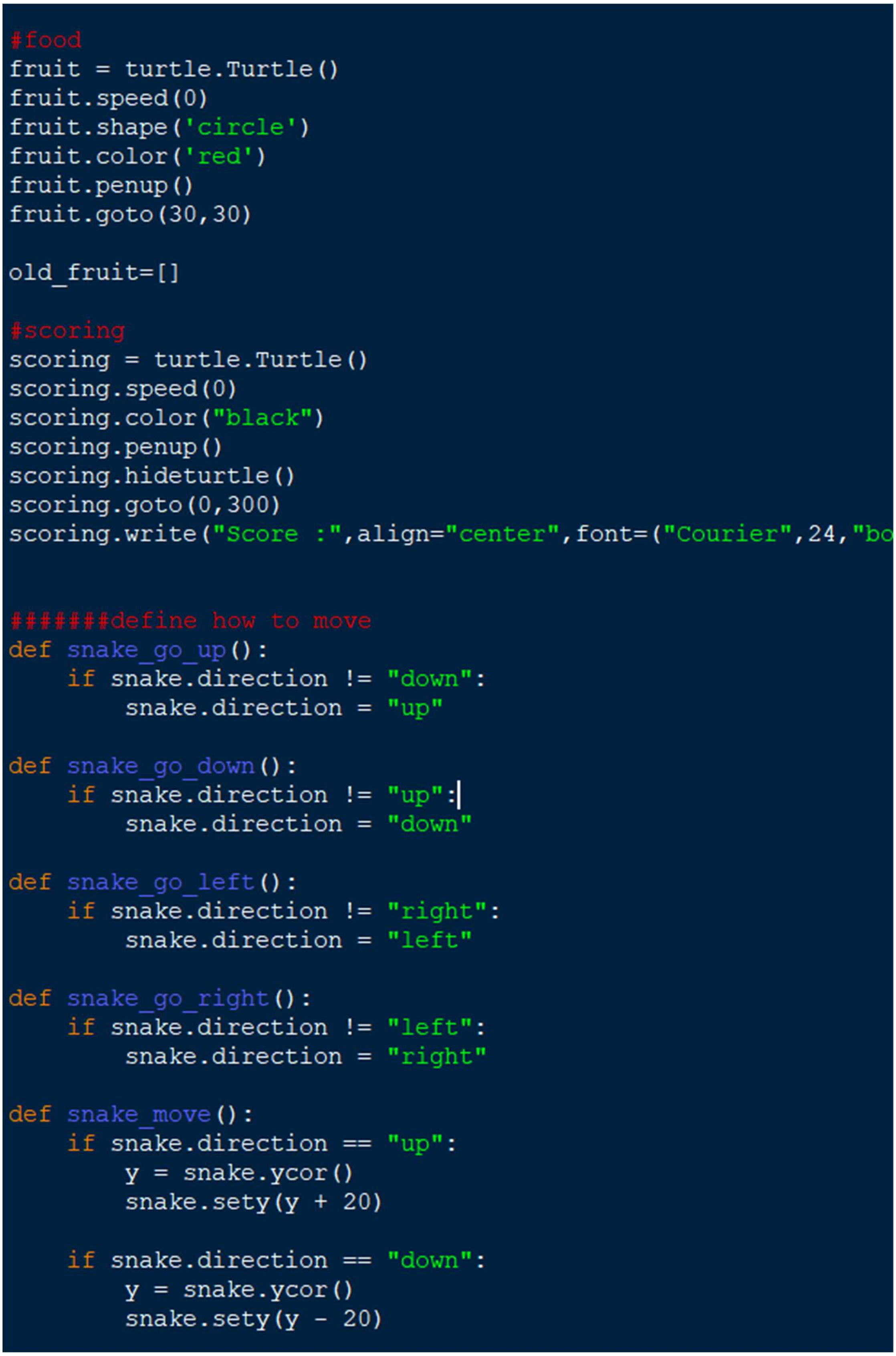
7. When snake touch itself

```
for food in old_fruit:

if food.distance(snake) < 20:

time.sleep(1)

screen.clear()

screen.bgcolor('Red')

scoring.goto(0,0)

scoring.write(" GAME OVER \n Your Score is {}".format(score),align="center",font=("Courier",30,"bold"))
```

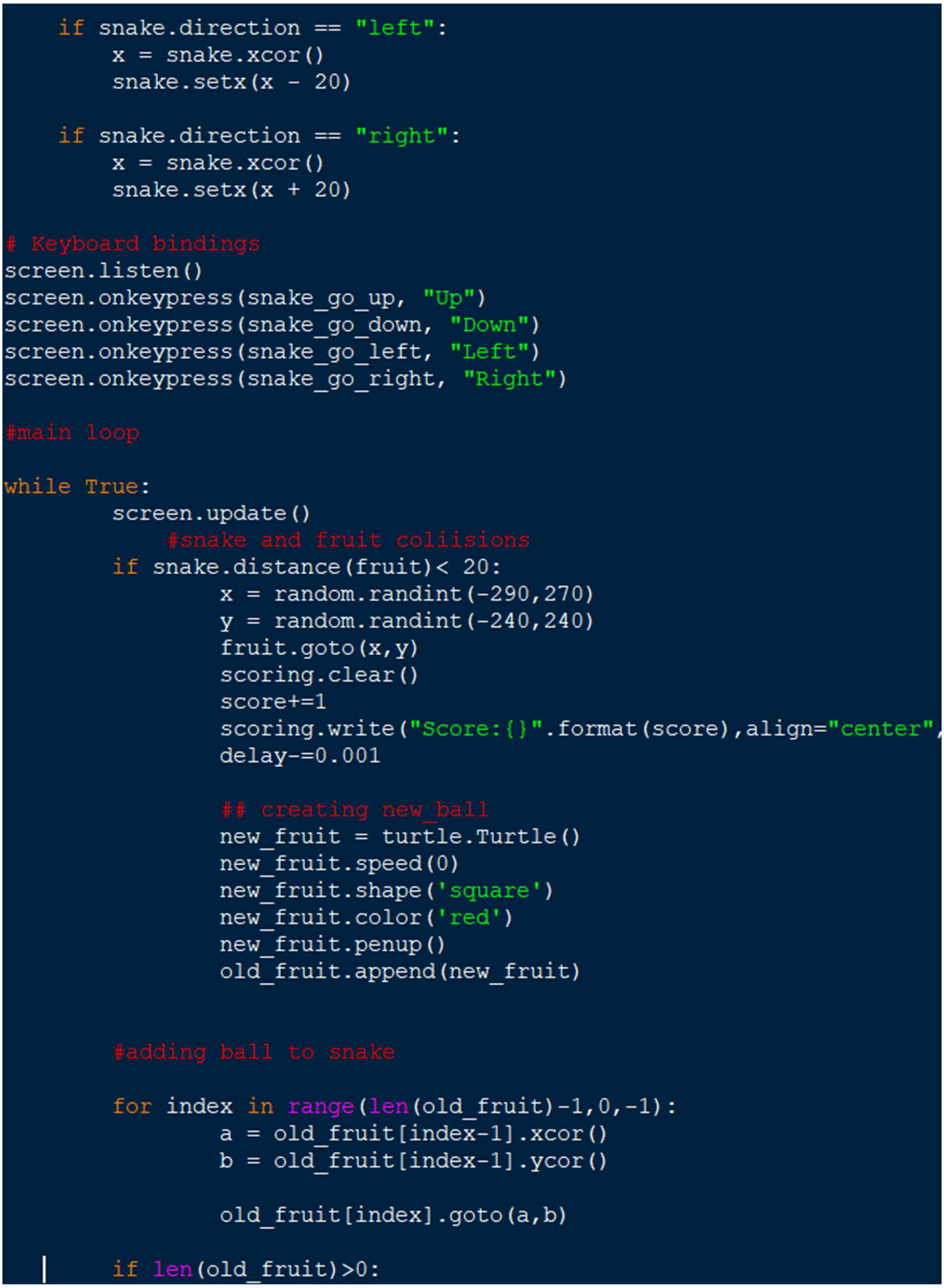
CODING AND SCREENSHOTS OF PROJECT



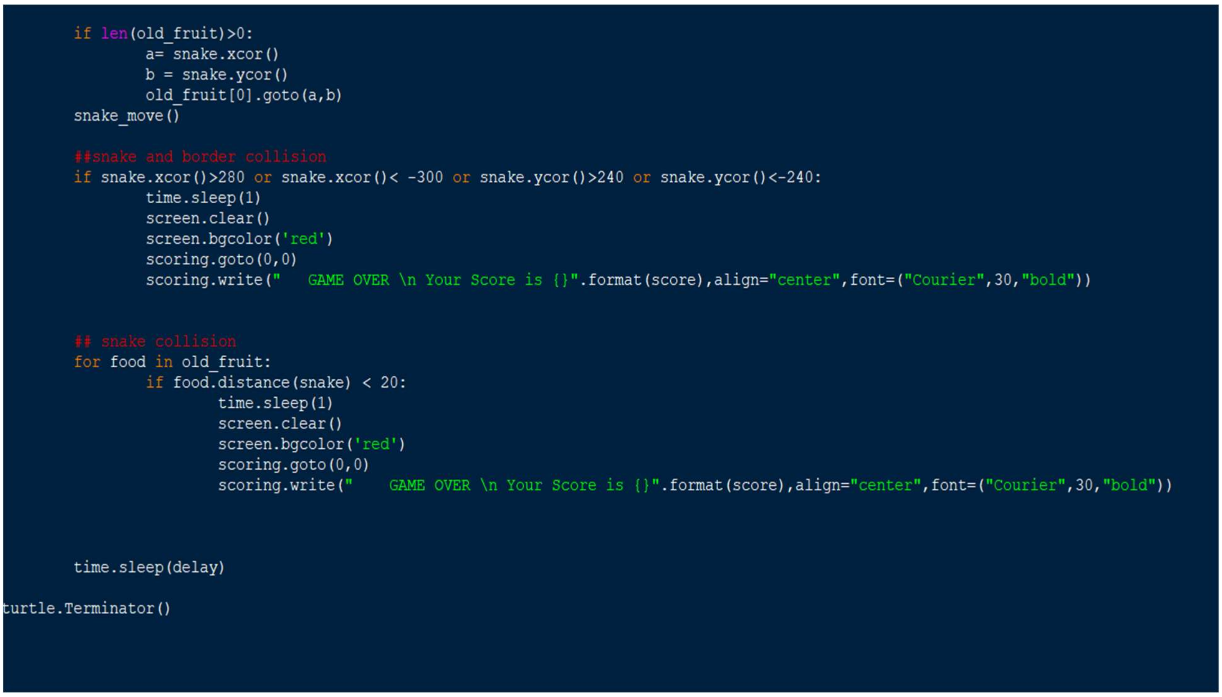
```
#importing libraries
import turtle
import random
import time


#creating turtle screen
screen = turtle.Screen()
screen.title('SNAKE GAME')
screen.setup(width = 700, height = 700)
screen.tracer(0)
turtle.bgcolor('grey')



##creating a border for our game

turtle.speed(5)
turtle.pensize(4)
turtle.penup()
turtle.goto(-310,250)
turtle.pendown()
turtle.color('blue')
turtle.forward(600)
turtle.right(90)
turtle.forward(500)
turtle.right(90)
turtle.forward(600)
turtle.right(90)
turtle.forward(500)
turtle.penup()
turtle.hideturtle()

#score
score = 0
delay = 0.1


#snake
snake = turtle.Turtle()
snake.speed(0)
snake.shape('square')
snake.color("black")
snake.penup()
snake.goto(0,0)
snake.direction = 'stop'
```



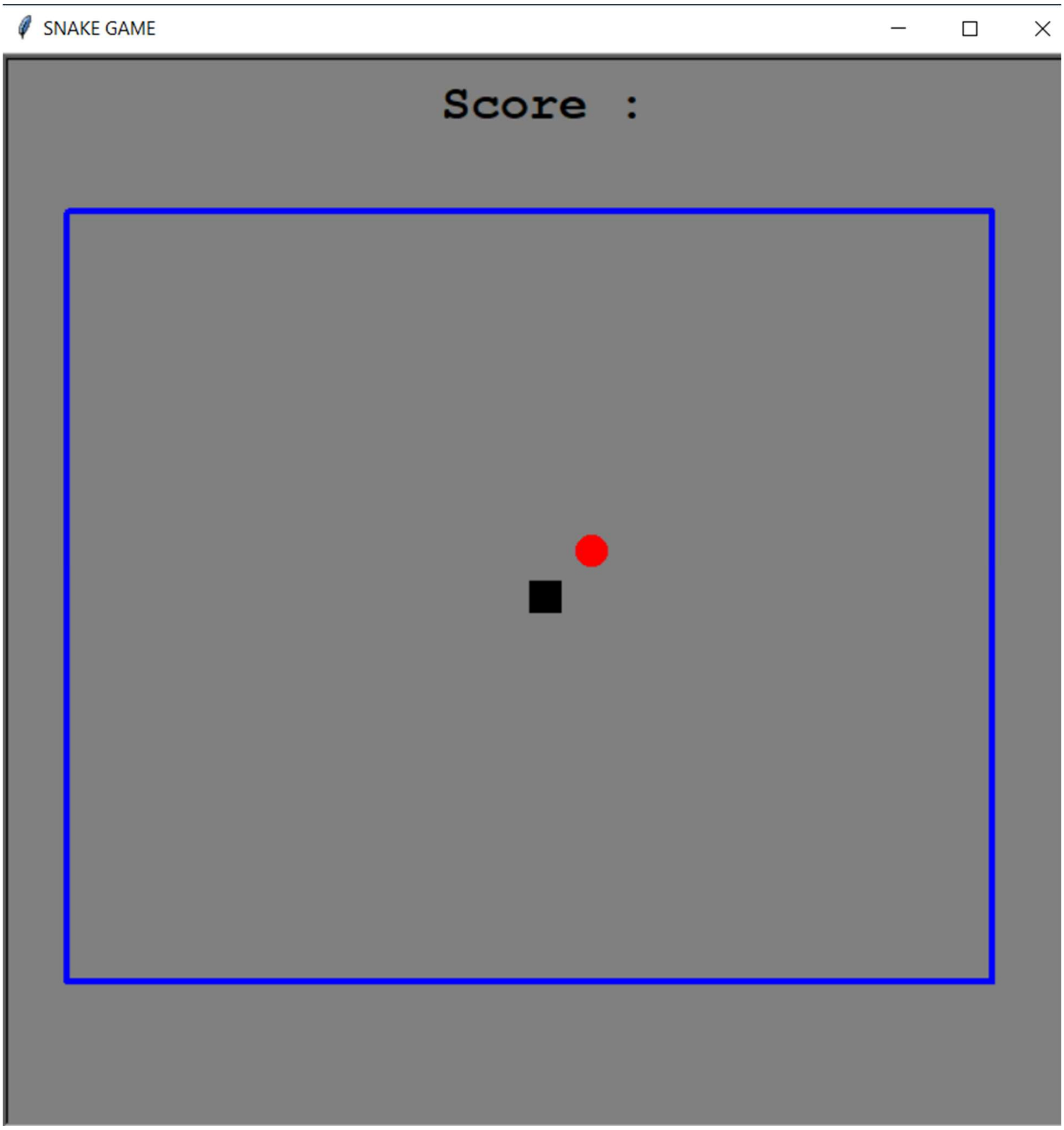
```
#food
fruit = turtle.Turtle()
fruit.speed(0)
fruit.shape('circle')
fruit.color('red')
fruit.penup()
fruit.goto(30,30)

old_fruit=[]

#scoring
scoring = turtle.Turtle()
scoring.speed(0)
scoring.color("black")
scoring.penup()
scoring.hideturtle()
scoring.goto(0,300)
scoring.write("Score :",align="center",font=("Courier",24,"bo


#######define how to move
def snake_go_up():
    if snake.direction != "down":
        snake.direction = "up"

def snake_go_down():
    if snake.direction != "up":
        snake.direction = "down"

def snake_go_left():
    if snake.direction != "right":
        snake.direction = "left"

def snake_go_right():
    if snake.direction != "left":
        snake.direction = "right"

def snake_move():
    if snake.direction == "up":
        y = snake.ycor()
        snake.sety(y + 20)

    if snake.direction == "down":
        y = snake.ycor()
        snake.sety(y - 20)
```

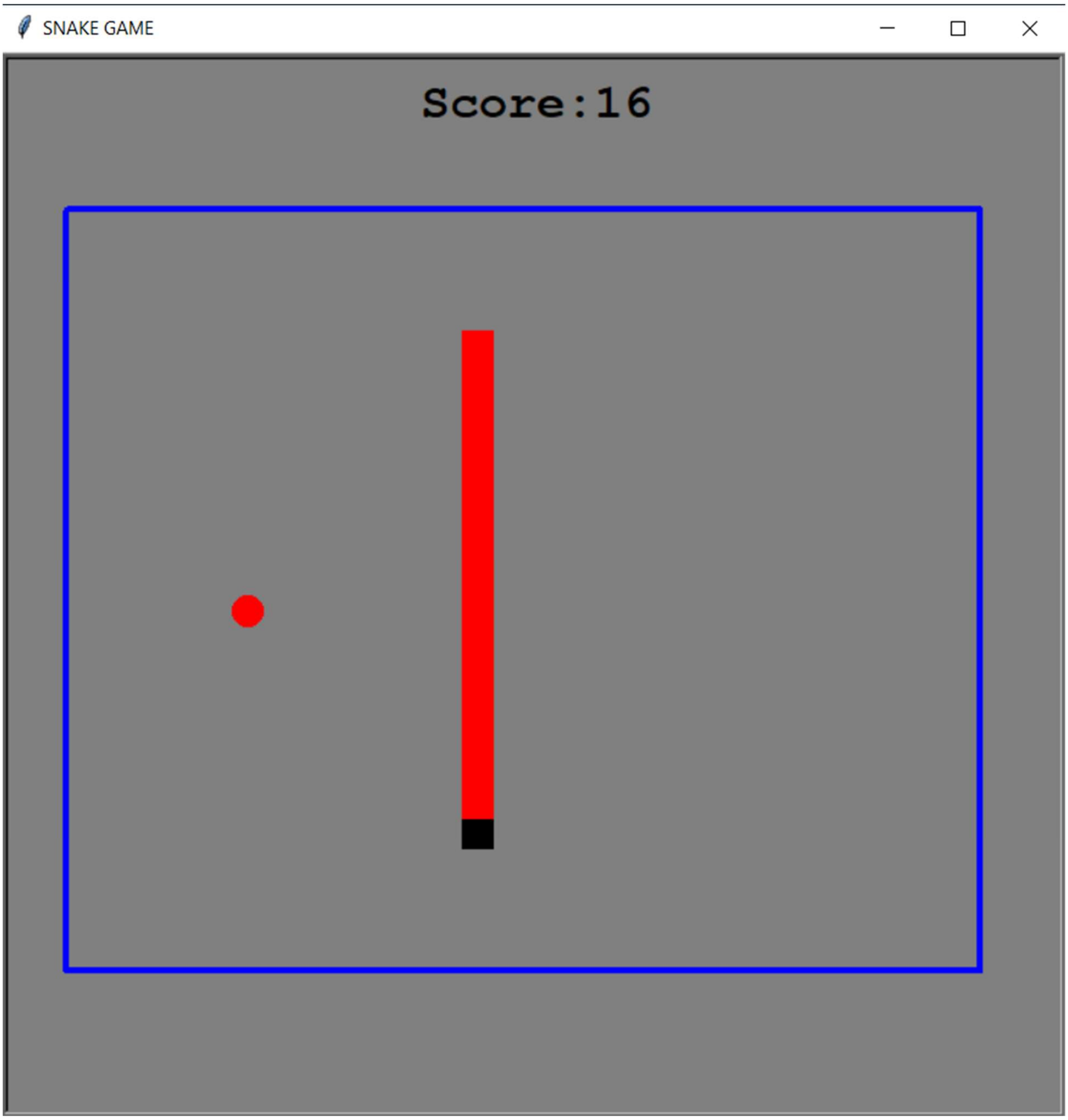


```
    if snake.direction == "left":
        x = snake.xcor()
        snake.setx(x - 20)

    if snake.direction == "right":
        x = snake.xcor()
        snake.setx(x + 20)

# Keyboard bindings
screen.listen()
screen.onkeypress(snake_go_up, "Up")
screen.onkeypress(snake_go_down, "Down")
screen.onkeypress(snake_go_left, "Left")
screen.onkeypress(snake_go_right, "Right")

#main loop

while True:
        screen.update()
            #snake and fruit coliisions
        if snake.distance(fruit)< 20:
                x = random.randint(-290,270)
                y = random.randint(-240,240)
                fruit.goto(x,y)
                scoring.clear()
                score+=1
                scoring.write("Score:{}".format(score),align="center",
                delay-=0.001

                ## creating new_ball
                new_fruit = turtle.Turtle()
                new_fruit.speed(0)
                new_fruit.shape('square')
                new_fruit.color('red')
                new_fruit.penup()
                old_fruit.append(new_fruit)


        #adding ball to snake

        for index in range(len(old_fruit)-1,0,-1):
                a = old_fruit[index-1].xcor()
                b = old_fruit[index-1].ycor()

                old_fruit[index].goto(a,b)

        if len(old_fruit)>0:
```

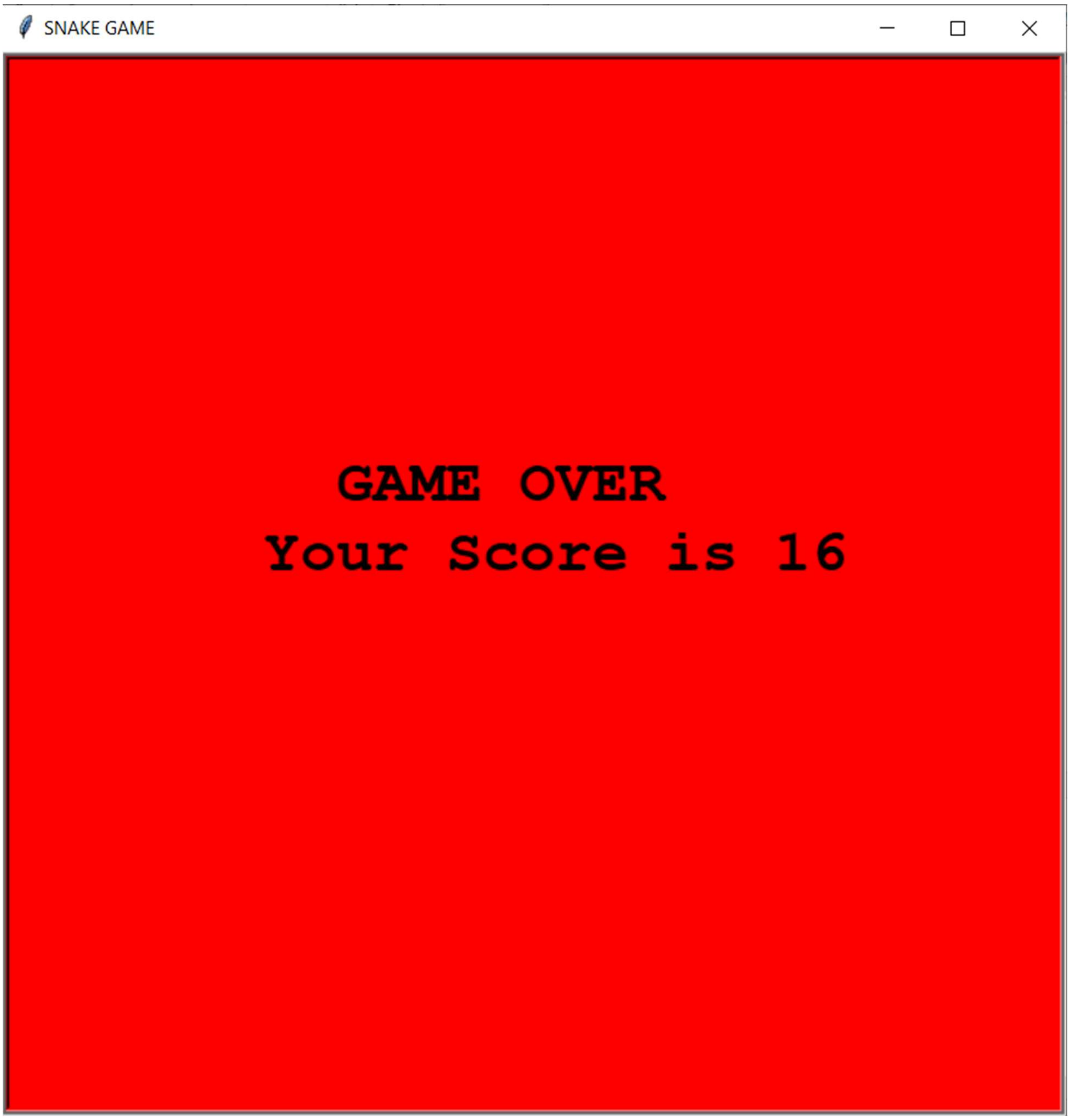


```
        if len(old_fruit)>0:
                a= snake.xcor()
                b = snake.ycor()
                old_fruit[0].goto(a,b)
        snake_move()

        ##snake and border collision
        if snake.xcor()>280 or snake.xcor()< -300 or snake.ycor()>240 or snake.ycor()<-240:
                time.sleep(1)
                screen.clear()
                screen.bgcolor('red')
                scoring.goto(0,0)
                scoring.write("   GAME OVER \n Your Score is {}".format(score),align="center",font=("Courier",30,"bold"))


        ## snake collision
        for food in old_fruit:
                if food.distance(snake) < 20:
                        time.sleep(1)
                        screen.clear()
                        screen.bgcolor('red')
                        scoring.goto(0,0)
                        scoring.write("    GAME OVER \n Your Score is {}".format(score),align="center",font=("Courier",30,"bold"))



        time.sleep(delay)

turtle.Terminator()
```


SNAKE GAME
Score :



SNAKE GAME
Score:16



SNAKE GAME

GAME OVER

Your Score is 16

……………………………………………………………………………………………

CONCLUSION AND FUTURE WORK

We successfully developed Snake game project in python. We learned how to use turtle modules and draw on the screen using a turtle. We also learn about random and time modules.

SUBMITTED TO: Dr. Siddharth Singh Chouhan

BY: Ashish Sarswat (12010488)

& Dev Munjal (12007537)