

Sakshi Patel

CPS 3320

Professor Robert Domanski

Project #2

March 25, 2021

## **Write-up**

### **Snake Game**

So I decided to write a code for a snake game. The snake game is a very popular and fun game. Every time the snake eats the fruit, its length grows longer, which makes the game more difficult. The objective of this python project was 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 be over. To build the snake game project I 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.

The first step is to import the libraries so I imported Turtle, Random and Time libraries.

The second step is to create the game screen, so for that I used:

- 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

```
#creating turtle screen
screen = turtle.Screen()
screen.title('SNAKE GAME BY SAKSHI')
screen.setup(width = 800, height = 800)
screen.tracer(0)
turtle.bgcolor('Lavender')

#creating a border for our game
turtle.speed(5)
turtle.pensize(4)
turtle.penup()
turtle.goto(-310,250)
turtle.pendown()
turtle.color('black')
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()
```

Third step was to create a snake and its food and for that I used:

- 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

```
#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,"bold"))
```

Fourth step was to Keyboard binding for that I used:

- 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 is pressed then the snake will move in the left direction.
- If the Right key will be press then the snake will move in the right direction

Fifth step was to write a function for Snake and fruit collision. So, 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.

Moreover, I also added a function for Snake and border collision.

```
##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('Lavender')
    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('Lavender')
        scoring.goto(0,0)
        scoring.write("  GAME OVER \n Your Score is {}".format(score),align="center",font=("Courier",30,"bold"))
```

So basically if the snake touches the border of the game or if the snake touches itself then the game will be over. screen.clear() will delete all the drawing of the turtle on the screen

So here is the final Output:

