

# ASSIGNMENT

RLMCA369 - PYTHON PROGRAMMING

SUBMITTED TO:  
MS. ANIT JAMES

SUBMITTED BY:  
ELIZABATH THOMAS  
INTMCA-S9  
RNO. 23

## Create an application using Tkinter

### **Application:**

Snake Game App

### **Aim:**

Tkinter-based Snake Game provide users with an engaging and enjoyable gaming experience, challenging them to control the snake, collect food, avoid collisions, and achieve the highest possible score while offering a user-friendly interface and smooth gameplay.

### **Program:**

```
import tkinter as tk
import random
import time

# Constants
WIDTH = 300
HEIGHT = 300
DELAY = 100 # Milliseconds

class SnakeGame:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Snake Game")
        self.canvas = tk.Canvas(self.root, width=WIDTH, height=HEIGHT)
        self.canvas.pack()
        self.root.bind("<Key>", self.on_key_press)
        self.reset()

    def reset(self):
        self.snake_x = [100, 90, 80]
        self.snake_y = [100, 100, 100]
        self.food_x = random.randint(0, (WIDTH-10)//10) * 10
        self.food_y = random.randint(0, (HEIGHT-10)//10) * 10
        self.direction = "Right"
        self.score = 0
        self.game_over = False
        self.draw()

    def draw(self):
        self.canvas.delete(tk.ALL)

        # Draw food
        self.canvas.create_rectangle(self.food_x, self.food_y, self.food_x +
10, self.food_y + 10, fill="red")
```

```

        # Draw snake
        for i in range(len(self.snake_x)):
            self.canvas.create_rectangle(self.snake_x[i], self.snake_y[i],
self.snake_x[i] + 10, self.snake_y[i] + 10, fill="green")

        # Draw score
        self.canvas.create_text(10, 10, text=f"Score: {self.score}",
anchor=tk.NW)

        if self.game_over:
            self.canvas.create_text(WIDTH // 2, HEIGHT // 2, text="Game
Over!", font=("Helvetica", 20))

    def move(self):
        if not self.game_over:
            head_x, head_y = self.snake_x[0], self.snake_y[0]

            if self.direction == "Right":
                head_x += 10
            elif self.direction == "Left":
                head_x -= 10
            elif self.direction == "Up":
                head_y -= 10
            elif self.direction == "Down":
                head_y += 10

            self.snake_x.insert(0, head_x)
            self.snake_y.insert(0, head_y)

            if head_x == self.food_x and head_y == self.food_y:
                self.score += 1
                self.food_x = random.randint(0, (WIDTH-10)//10) * 10
                self.food_y = random.randint(0, (HEIGHT-10)//10) * 10
            else:
                self.snake_x.pop()
                self.snake_y.pop()

            # Check collision with wall or self
            if head_x < 0 or head_x >= WIDTH or head_y < 0 or head_y >= HEIGHT
or (head_x, head_y) in zip(self.snake_x[1:], self.snake_y[1:]):
                self.game_over = True

            self.draw()

        if not self.game_over:
            self.root.after(DELAY, self.move)

```


```
def on_key_press(self, event):
    new_direction = event.keysym
    if (new_direction == "Right" and self.direction != "Left" or
        new_direction == "Left" and self.direction != "Right" or
        new_direction == "Up" and self.direction != "Down" or
        new_direction == "Down" and self.direction != "Up"):
        self.direction = new_direction

def start(self):
    self.move()
    self.root.mainloop()

if __name__ == "__main__":
    game = SnakeGame()
    game.start()

game.start()
```


**Output:**

 Snake Game



Score: 0



 Snake Game



Score: 3

