Snake Game in Python (Pygame)

Purpose of the Program:

This program implements the classic Snake game using the Pygame library. The objective of the game is to control a snake, eat food, and avoid crashing into the walls or the snake's own body. Each time the snake eats food, it grows longer, and the player earns points.

Imports:

python

Copy

```
import pygame

import time

import random
```

- pygame: A library used for making games in Python. It helps to create graphics, handle events (like keyboard input), and more.
- time: Provides functions for handling time (e.g., pauses).
- random: Generates random numbers, which is used here for placing food randomly on the screen.

Initializing Pygame:

python

Copy

```
pygame.init()
```

This line initializes all the necessary modules of Pygame to start using it.

Defining Colors:

python

Copy

```
white = (255, 255, 255)

yellow = (255, 255, 102)

black = (0, 0, 0)

red = (213, 50, 80)

green = (0, 255, 0)
```

blue = (50, 153, 213)

These are RGB color values for different colors used in the game like white, yellow, black, red, green, and blue.

Setting up the Game Window:

python

Copy

```
dis_width = 600

dis_height = 400

dis = pygame.display.set_mode((dis_width, dis_height))

pygame.display.set_caption('Snake Game In Python')
```

- dis_width and dis_height: These define the width (600 pixels) and height (400 pixels) of the game window.
- pygame.display.set_mode(): Initializes the window with the specified dimensions.
- pygame.display.set_caption(): Sets the title of the window to 'Snake Game In Python'.

Other Important Variables:

python

Copy

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

snake_block = 10

snake_speed = 15

font_style = pygame.font.SysFont("bahnschrift", 25)

score_font = pygame.font.SysFont("comicsansms", 35)
```

- clock: Controls the game’s framerate.
- snake_block: The size of each block that makes up the snake and the food. It's set to 10 pixels.
- snake_speed: The speed at which the snake moves. Higher values make the snake move faster.
- font_style: Font used for displaying messages during the game.
- score_font: Font used for displaying the score.

Functions:

1. Your_score(score):

Displays the current score on the screen.

python

Copy

```
def Your_score(score):
    value = score_font.render("Your Score: " + str(score), True, yellow)
    dis.blit(value, [0, 0])
```

- score_font.render(): Creates the text for displaying the score in yellow color.
- dis.blit(): Draws the text on the screen at the top-left corner.

2. our_snake(snake_block, snake_list):

Draws the snake on the screen.

python

Copy

```
def our_snake(snake_block, snake_list):
    for x in snake_list:
        pygame.draw.rect(dis, black, [x[0], x[1], snake_block, snake_block])
```

- Iterates over each part of the snake (snake_list) and draws a black rectangle for each block of the snake.

3. message(msg, color):

Displays a message on the screen.

python

Copy

```
def message(msg, color):
    mesg = font_style.render(msg, True, color)
    dis.blit(mesg, [dis_width / 6, dis_height / 3])
```

- Renders the message in the given color and displays it in the center of the screen.

Main Game Loop:

The gameLoop() function runs the main game. Here’s an overview:

python

Copy

```
def gameLoop():
    game_over = False
    game_close = False
    # Initialize the snake's position and movement
    x1 = dis_width / 2
    y1 = dis_height / 2
    x1_change = 0
    y1_change = 0
    snake_List = []
    Length_of_snake = 1

    # Randomly place food
    foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
    foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0

    while not game_over:
        while game_close == True:
            dis.fill(blue)
            message("You Lost! Press 'C' to Play Again or 'Q' To Quit The Game", red)
            Your_score(Length_of_snake - 1)
            pygame.display.update()
            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_q:
                        game_over = True
                        game_close = False
                    if event.key == pygame.K_c:
                        gameLoop()  # Restart the game
```

```
# Event handling (keyboard inputs)
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        game_over = True
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            x1_change = -snake_block
            y1_change = 0
        elif event.key == pygame.K_RIGHT:
            x1_change = snake_block
            y1_change = 0
        elif event.key == pygame.K_UP:
            y1_change = -snake_block
            x1_change = 0
        elif event.key == pygame.K_DOWN:
            y1_change = snake_block
            x1_change = 0

# Check if snake goes out of bounds
if x1 >= dis_width or x1 < 0 or y1 >= dis_height or y1 < 0:
    game_close = True
x1 += x1_change
y1 += y1_change
dis.fill(blue)

# Draw food
pygame.draw.rect(dis, green, [foodx, foody, snake_block, snake_block])

# Update snake's position
snake_Head = []
```

```
    snake_Head.append(x1)
    snake_Head.append(y1)
    snake_List.append(snake_Head)
    if len(snake_List) > Length_of_snake:
        del snake_List[0]

    # Check for collisions with itself
    for x in snake_List[:-1]:
        if x == snake_Head:
            game_close = True

    # Draw the snake and update the score
    our_snake(snake_block, snake_List)
    Your_score(Length_of_snake - 1)

    pygame.display.update()

    # If snake eats food
    if x1 == foodx and y1 == foody:
        foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
        foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
        Length_of_snake += 1

    clock.tick(snake_speed)  # Control the snake's speed

pygame.quit()
quit()
```

- **Game Over & Restart:** The game checks if the player loses by either hitting the wall or the snake itself. It gives the option to restart (C) or quit (Q).
- **Snake Movement:** Arrow keys are used to control the snake’s direction.

- **Food Collision:** When the snake eats food (i.e., its head coordinates match the food's), the snake grows longer, and new food is placed randomly on the screen.
- **Score:** The score increases as the snake eats food.

Starting the Game:

python

Copy

gameLoop()

This line starts the game when the program is run.

Conclusion:

This code provides a fully functional Snake game in Python. You can control the snake using the arrow keys, and the game continues until the snake either runs into a wall or itself. After that, you can restart or quit the game.