Algorithm and Programming Final Project

By: Hansel Faren (2501990350)

L1AC / Computer Science

Snake

I. Game Overview

1. Genre:

   Snake game is a puzzle video game.

2. Visual:

   The art that is used is 8-bit.

3. Game Flow:

   It starts with a 1 block of snake and eat as much food to grow as long as it can.

II. Gameplay

1. Objectives:

   To grow as long as it can until it may fill in the whole screen.

2. Controls:

   Players can move up, down, left, and right using the arrow key.

```
    elif event.type == pygame.KEYDOWN: # Create movements for the snake
        if event.key == pygame.K_LEFT:
            x1Change = -snake_block
            y1Change = 0
        elif event.key == pygame.K_RIGHT:
            x1Change = snake_block
            y1Change = 0
        elif event.key == pygame.K_UP:
            x1Change = 0
            y1Change = -snake_block
        elif event.key == pygame.K_DOWN:
            x1Change = 0
            y1Change = snake_block
```

III. Interface

1. Display Screen:

The place where the game will take place

```
# to adjust the width and height for display screen
display_width = 600
display_height = 400

# Create the screen board
display = pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption("Snake Game")
```

Declaring the color for the text and object

```
# Coloring
yellow = (255, 255, 102)
red = (213, 50, 80)
green = (0, 255, 0)
blue = (50,153, 213)
black = (0, 0, 0)
white = (255, 255, 255)
```

The following code will be randomizing the food for the snake

```
# Create food
foodx = round(random.randrange(0, display_width - snake_block) / 10.0) * 10.0
foody = round(random.randrange(0, display_height - snake_block) / 10.0) * 10.0
```

Below is for drawing the food in 1 block size (the pygame.draw.rect)

```
display.fill(blue)
pygame.draw.rect(display, green, [foodx, foody, snake_block, snake_block]
SnakeHead = []
```

2. Snake:
    1. The code where to adjust the size and the speed for snake.

```
# To adjust the size of the game
snake_block = 10

# To adjust the speed for the snake
snake_speed = 15
```

    2. Increasing the length of the snake
        - Drawing the snake.

```
# Define snake and for increasing the length
def Snake(snake_block, SnakeList):
    for x in SnakeList:
        pygame.draw.rect(display, black, [x[0], x[1], snake_block, snake_block])
```

        - When the snake eats the food and add another food when it is eaten

```
# Increase the length for snake
if x1 == foodx and y1 == foody:
    foodx = round(random.randrange(0, display_width - snake_block) / 10.0) * 10.0
    foody = round(random.randrange(0, display_height - snake_block) / 10.0) * 10.0
    SnakeLength += 1
```

        - When the snake hits it owns body.

```
if len(SnakeList) > SnakeLength:
    del SnakeList[0]
# if it hits the body of the snake
for x in SnakeList[:-1]:
    if x == SnakeHead:
        GameClose = True
```

3. Texts
    a. Declaring the style and size font for the score and message.

```
# To declare the font for score and message
FontStyle = pygame.font.SysFont("cambria", 25)
ScoreFont = pygame.font.SysFont("georgia", 35)
```

b. Defining and display the Score Board

```
# Create score board
def YourScore(score):
    number = ScoreFont.render("Your Score " + str(score), True, red)
    display.blit(number, [0, 0])
```

c. Defining the text message when you hit the edges or snake's body

```
# Define message for later
def Message(msg, color):
    message = FontStyle.render(msg, True, color)
    display.blit(message, [display_width / 6, display_height / 3])
```

d. Displaying the text if you lose the game

```
display.fill(blue)
Message("Too Bad! Press P to play again or Q to quit", yellow)
YourScore(SnakeLength - 1)
pygame.display.update()
```

4. Retry and Quit game command

1. Declaring False

```
# Create the loop
def GameLoop():
    GameOver = False # For closing the game
    GameClose = False # To show the quit or retry text if the snake hit edges
```

2. Code for exit or play again the game

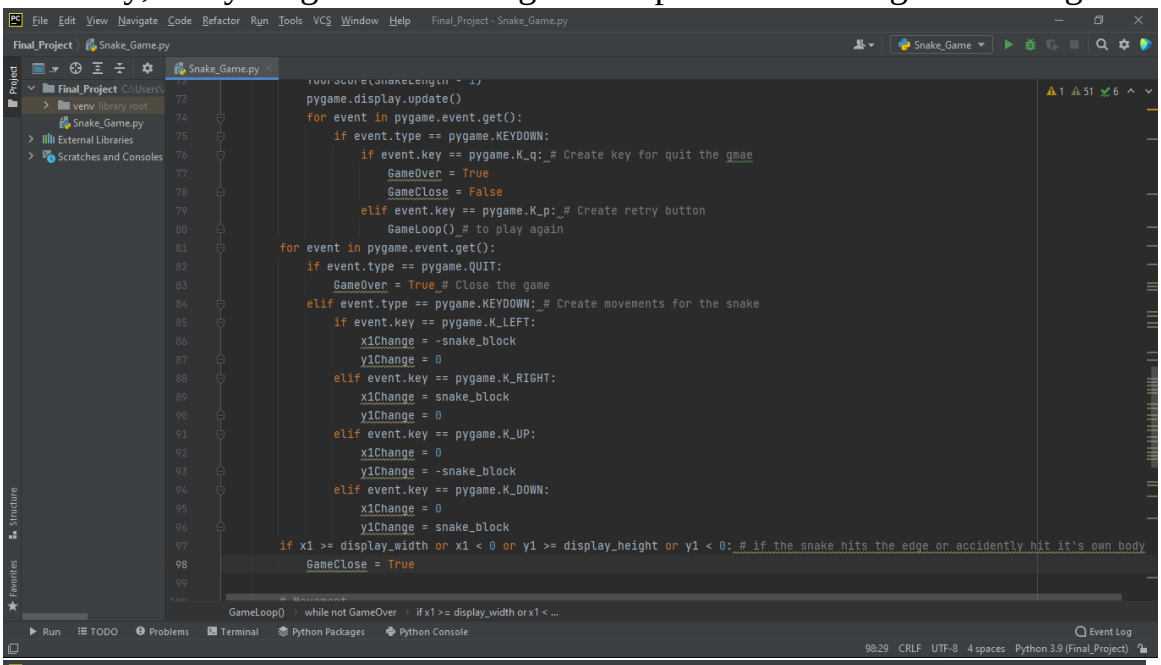
```
while not GameOver:
    while GameClose == True:
        display.fill(blue)
        Message("Too Bad! Press P to play again or Q to quit", yellow) #Text display if you lose
        YourScore(SnakeLength - 1)
        pygame.display.update()
        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_q: # Create key for quit the gmae
                    GameOver = True
                    GameClose = False
                elif event.key == pygame.K_p: # Create retry button
                    GameLoop() # to play again
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            GameOver = True # Close the game
```

- If the snake hits the edges of the screen

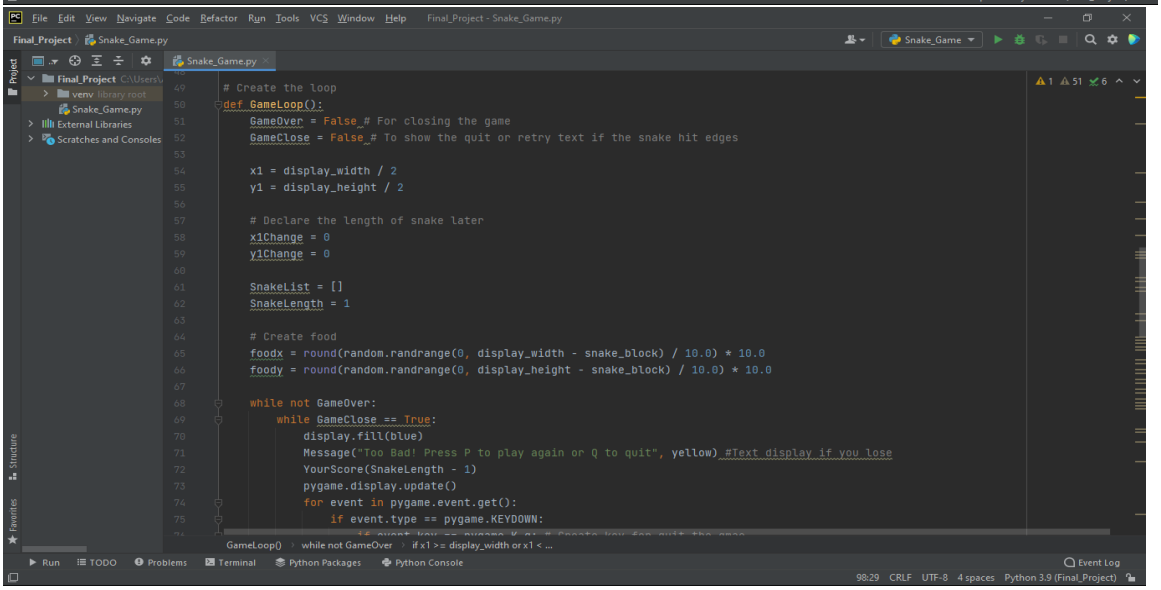
```
if x1 >= display_width or x1 < 0 or y1 >= display_height or y1 < 0:
    GameClose = True
```

5. Game Starts

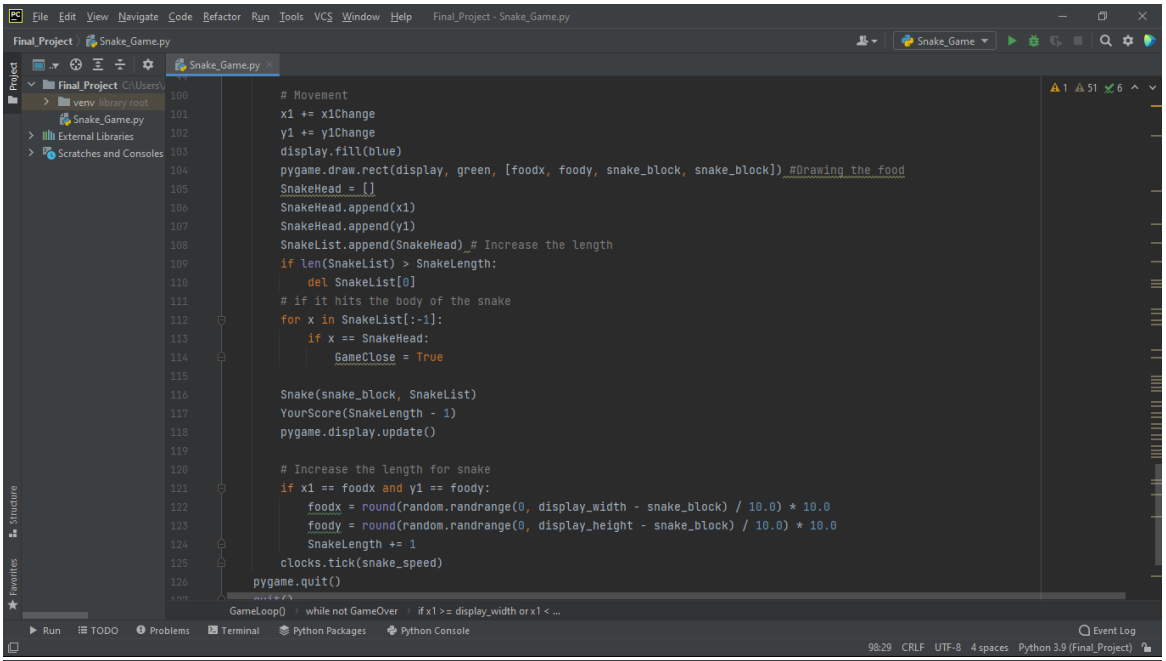
Basically, everything inside of the game loop will be running the entire game

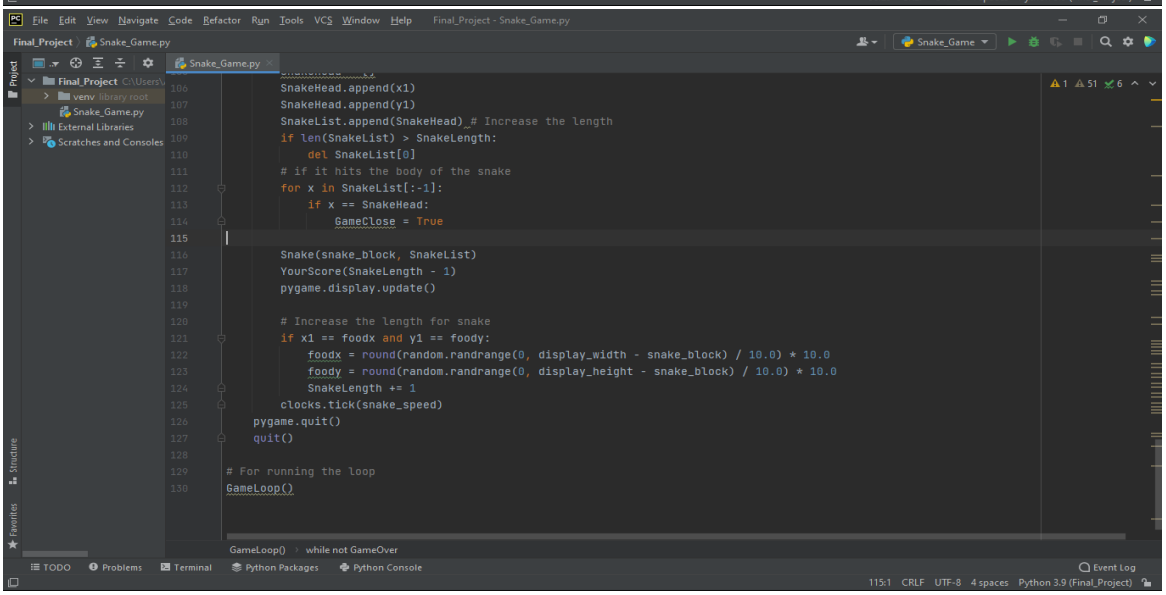


```
            pygame.display.update()
            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_q:  # Create key for quit the gmae
                        GameOver = True
                        GameClose = False
                    elif event.key == pygame.K_p:  # Create retry button
                        GameLoop()  # to play again
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                GameOver = True  # Close the game
            elif event.type == pygame.KEYDOWN:  # Create movements for the snake
                if event.key == pygame.K_LEFT:
                    x1Change = -snake_block
                    y1Change = 0
                elif event.key == pygame.K_RIGHT:
                    x1Change = snake_block
                    y1Change = 0
                elif event.key == pygame.K_UP:
                    x1Change = 0
                    y1Change = -snake_block
                elif event.key == pygame.K_DOWN:
                    x1Change = 0
                    y1Change = snake_block
        if x1 >= display_width or x1 < 0 or y1 >= display_height or y1 < 0:  # if the snake hits the edge or accidently hit it's own body
            GameClose = True
```



```
# Create the loop
def GameLoop():
    GameOver = False  # For closing the game
    GameClose = False  # To show the quit or retry text if the snake hit edges

    x1 = display_width / 2
    y1 = display_height / 2

    # Declare the length of snake later
    x1Change = 0
    y1Change = 0

    SnakeList = []
    SnakeLength = 1

    # Create food
    foodx = round(random.randrange(0, display_width - snake_block) / 10.0) * 10.0
    foody = round(random.randrange(0, display_height - snake_block) / 10.0) * 10.0

    while not GameOver:
        while GameClose == True:
            display.fill(blue)
            Message("Too Bad! Press P to play again or Q to quit", yellow) #Text display if you lose
            YourScore(SnakeLength - 1)
            pygame.display.update()
            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN:
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help Final_Project - Snake_Game.py
Final_Project Snake_Game.py
Snake_Game
Snake_Game.py
Final_Project C:\Users\
venv library root
Snake_Game.py
External Libraries
Scratches and Consoles
Movement
x1 += x1Change
y1 += y1Change
display.fill(blue)
pygame.draw.rect(display, green, [foodx, foody, snake_block, snake_block]) #Drawing the food
SnakeHead = []
SnakeHead.append(x1)
SnakeHead.append(y1)
SnakeList.append(SnakeHead) # Increase the length
if len(SnakeList) > SnakeLength:
del SnakeList[0]
# if it hits the body of the snake
for x in SnakeList[:-1]:
if x == SnakeHead:
GameClose = True
Snake(snake_block, SnakeList)
YourScore(SnakeLength - 1)
pygame.display.update()
# Increase the length for snake
if x1 == foodx and y1 == foody:
foodx = round(random.randrange(0, display_width - snake_block) / 10.0) * 10.0
foody = round(random.randrange(0, display_height - snake_block) / 10.0) * 10.0
SnakeLength += 1
clocks.tick(snake_speed)
pygame.quit()
GameLoop() while not GameOver if x1 >= display_width or x1 < ...
Run TODO Problems Terminal Python Packages Python Console Event Log
98:29 CRLF UTF-8 4 spaces Python 3.9 (Final_Project)


File Edit View Navigate Code Refactor Run Tools VCS Window Help Final_Project - Snake_Game.py
Final_Project Snake_Game.py
Snake_Game
Snake_Game.py
Final_Project C:\Users\
venv library root
Snake_Game.py
External Libraries
Scratches and Consoles
SnakeHead.append(x1)
SnakeHead.append(y1)
SnakeList.append(SnakeHead) # Increase the length
if len(SnakeList) > SnakeLength:
del SnakeList[0]
# if it hits the body of the snake
for x in SnakeList[:-1]:
if x == SnakeHead:
GameClose = True
Snake(snake_block, SnakeList)
YourScore(SnakeLength - 1)
pygame.display.update()
# Increase the length for snake
if x1 == foodx and y1 == foody:
foodx = round(random.randrange(0, display_width - snake_block) / 10.0) * 10.0
foody = round(random.randrange(0, display_height - snake_block) / 10.0) * 10.0
SnakeLength += 1
clocks.tick(snake_speed)
pygame.quit()
quit()
# For running the loop
GameLoop()
GameLoop() while not GameOver
TODO Problems Terminal Python Packages Python Console Event Log
115:1 CRLF UTF-8 4 spaces Python 3.9 (Final_Project)

6. Games Working