Here is our project about

THE SNAKE GAME PROGRAM USING PYTHON

CONTENTS OF THIS PROJECT

We make this program wholeheartedly using:

1. **Visual Studio Code (VS Code)**

   It is an open source development application developed by Microsoft. It is a powerful text editor powered by IntelliSense, debugging, automatic code formatting, Git management, language support and many other featuresnux.

2. **Tkinter**

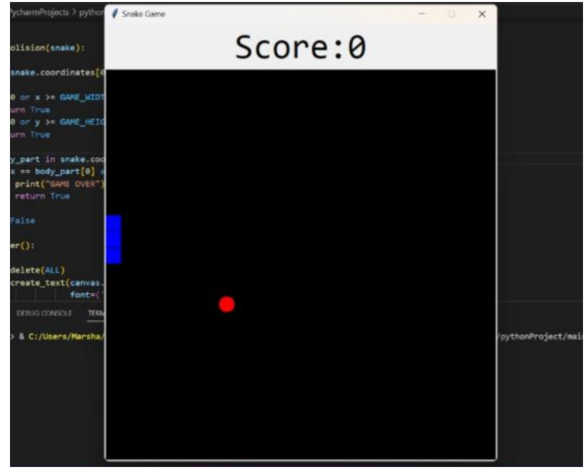
   Tkinter is a library used to create Python interface (GUI) applications. Tkinter provides a fast and easy, object-oriented, powerful way of creating GUI-based python applications. Tkinter is usually bundled with Python by default. So when you install Python, Tkinter will also be installed too. Tkinter is actually the OOP form of TCL/TK. TCL (Tool Command Language) is a programming language and TK is a library used by TCL to create GUI applications.

3. **Perulangan (Looping)**

   Perulangan or looping in Python is program code instructions that are executed repeatedly. Its function is to instruct the computer to do something repeatedly with a certain number of times as long as a predetermined condition is still fulfilled

THE GAMES

When start playing :

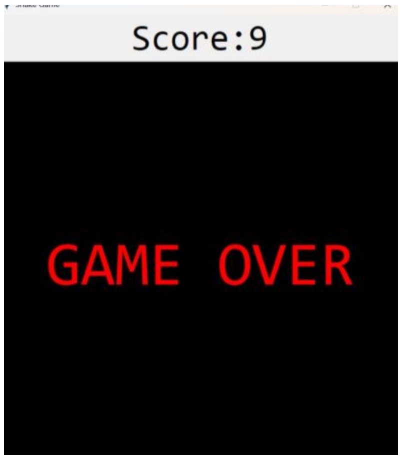


```
olision(snake):

snake.coordinates[0

0 or x >= GAME_WIDT
urn True
0 or y >= GAME_HEIG
urn True

y_part in snake.coo
x == body_part[0] a
 print("GAME OVER")
 return True

False

er():

delete(ALL)
create_text(canvas.
            font=('
```

```
> & C:/Users/Marsha/                              /pythonProject/mai
```

Score:0

When you hit the wall or the body of the snake itself



Score:9

GAME OVER

THE PLAYERS

001

Marsha Trista Aulia
DS02_02_1206220007

Muh Abid Atmira
DS02_02_1206220036

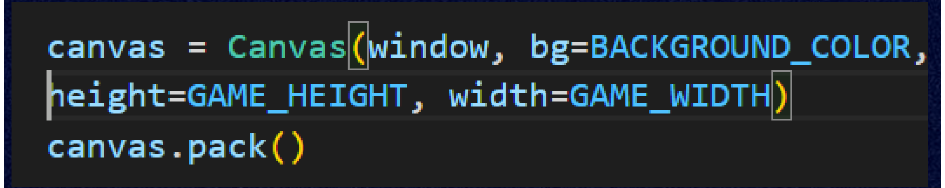
002

003

Firanitsy Shania K.
DS02_02_1206220013

The Program

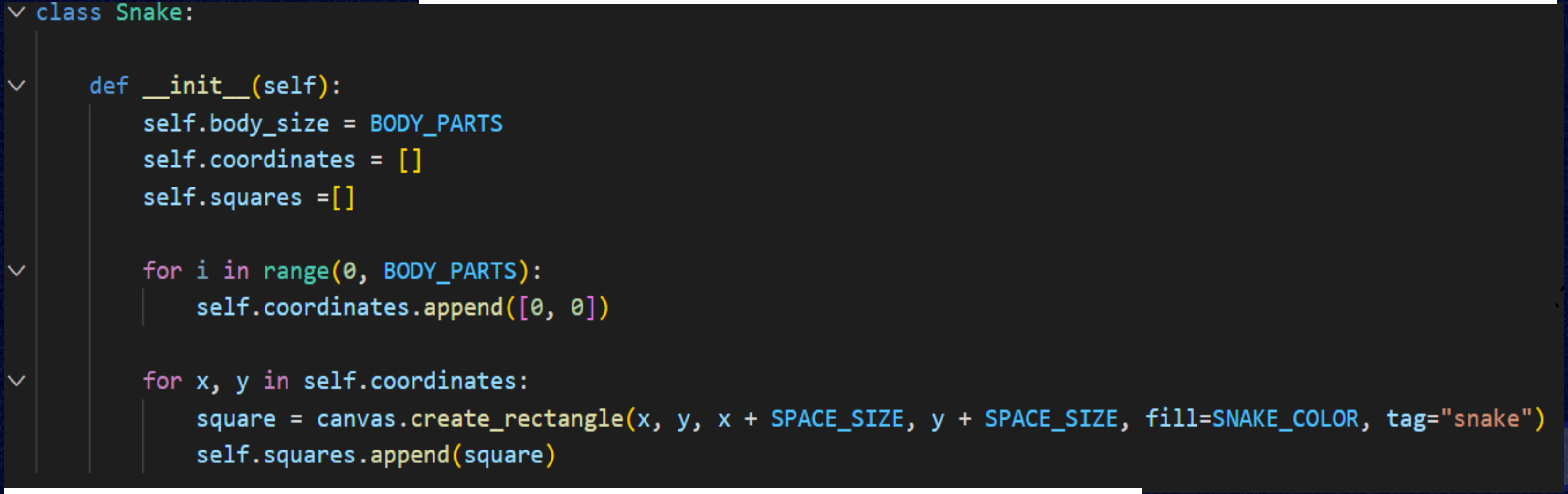
1. Membuat Background



```
canvas = Canvas(window, bg=BACKGROUND_COLOR,
height=GAME_HEIGHT, width=GAME_WIDTH)
canvas.pack()
```

Second Program

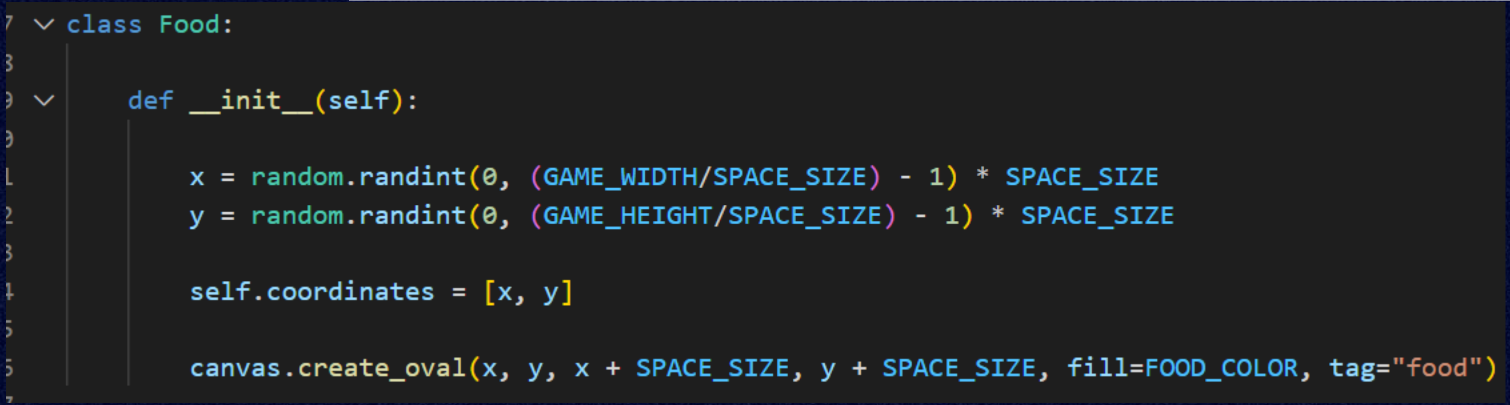
2. Membuat Bodypart (ular



```
class Snake:

    def __init__(self):
        self.body_size = BODY_PARTS
        self.coordinates = []
        self.squares =[]

        for i in range(0, BODY_PARTS):
            self.coordinates.append([0, 0])

        for x, y in self.coordinates:
            square = canvas.create_rectangle(x, y, x + SPACE_SIZE, y + SPACE_SIZE, fill=SNAKE_COLOR, tag="snake")
            self.squares.append(square)
```

Thtird Program

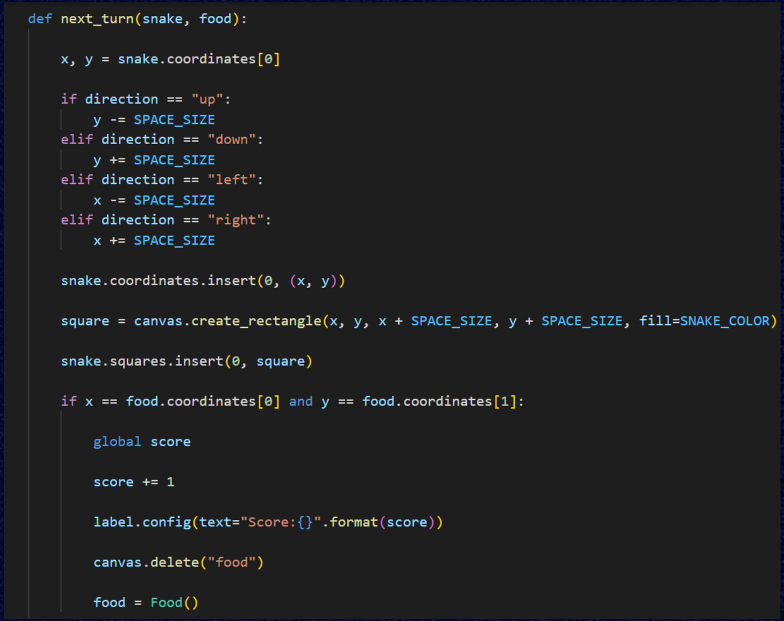
3. Membuat Makanan



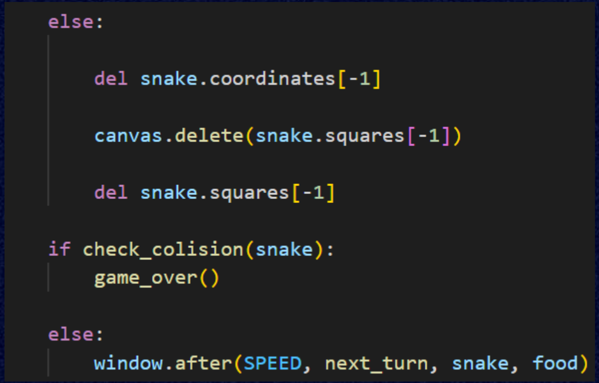
```python
class Food:

    def __init__(self):

        x = random.randint(0, (GAME_WIDTH/SPACE_SIZE) - 1) * SPACE_SIZE
        y = random.randint(0, (GAME_HEIGHT/SPACE_SIZE) - 1) * SPACE_SIZE

        self.coordinates = [x, y]

        canvas.create_oval(x, y, x + SPACE_SIZE, y + SPACE_SIZE, fill=FOOD_COLOR, tag="food")
```

Fourth Program

4. Membuat Perintah Bergerak



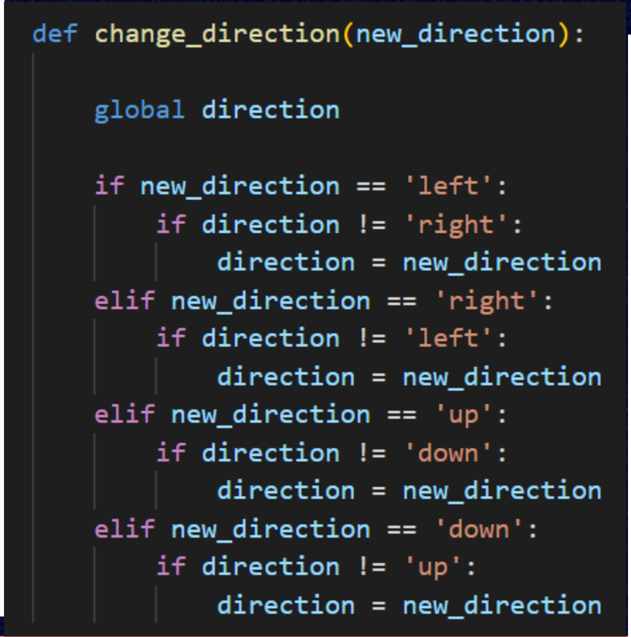
```
def next_turn(snake, food):

    x, y = snake.coordinates[0]

    if direction == "up":
        y -= SPACE_SIZE
    elif direction == "down":
        y += SPACE_SIZE
    elif direction == "left":
        x -= SPACE_SIZE
    elif direction == "right":
        x += SPACE_SIZE

    snake.coordinates.insert(0, (x, y))

    square = canvas.create_rectangle(x, y, x + SPACE_SIZE, y + SPACE_SIZE, fill=SNAKE_COLOR)

    snake.squares.insert(0, square)

    if x == food.coordinates[0] and y == food.coordinates[1]:

        global score

        score += 1

        label.config(text="Score:{}".format(score))

        canvas.delete("food")

        food = Food()
```



```
    else:

        del snake.coordinates[-1]

        canvas.delete(snake.squares[-1])

        del snake.squares[-1]

    if check_colision(snake):
        game_over()

    else:
        window.after(SPEED, next_turn, snake, food)
```

Fifth Program

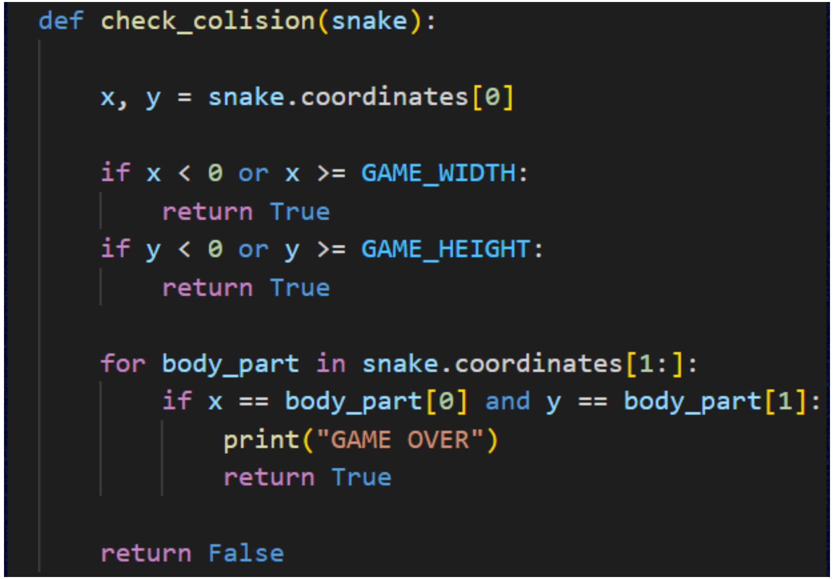
5. Membuat Arah Gerak Baru



```python
def change_direction(new_direction):

    global direction

    if new_direction == 'left':
        if direction != 'right':
            direction = new_direction
    elif new_direction == 'right':
        if direction != 'left':
            direction = new_direction
    elif new_direction == 'up':
        if direction != 'down':
            direction = new_direction
    elif new_direction == 'down':
        if direction != 'up':
            direction = new_direction
```

Sixth Program

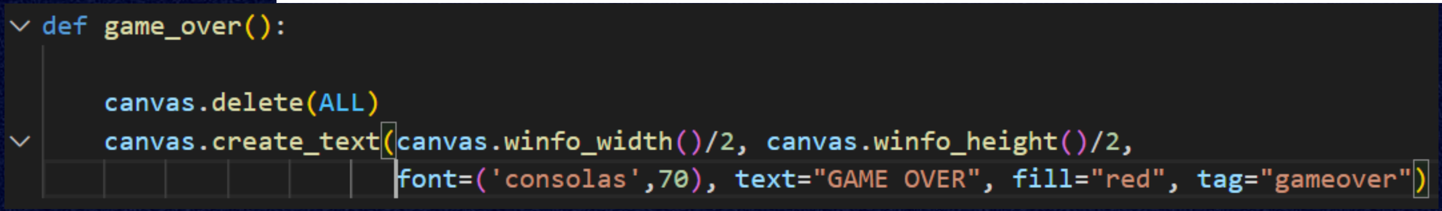
6. Membuat Prediksi Tabrakan



```
def check_colision(snake):

    x, y = snake.coordinates[0]

    if x < 0 or x >= GAME_WIDTH:
        return True
    if y < 0 or y >= GAME_HEIGHT:
        return True

    for body_part in snake.coordinates[1:]:
        if x == body_part[0] and y == body_part[1]:
            print("GAME OVER")
            return True

    return False
```

Seventh Program

7. Game Over



```python
def game_over():

    canvas.delete(ALL)
    canvas.create_text(canvas.winfo_width()/2, canvas.winfo_height()/2,
                       font=('consolas',70), text="GAME OVER", fill="red", tag="gameover")
```

How to play the GAMES

START

01

02

03

Play With arrows

The snake is moved with the arrows on the keyboard

Catch food and avoid crashes

To increase the score we have to move the snake to catch food

Game Over

You will "game over" if you hit the wall or hit your own body

Thanks!

Do you have any questions?

mahasiswa@ittelkom.ac.id
0811-3278-005
Ittelkom-sby.ac.id