



SUMMER VACATION

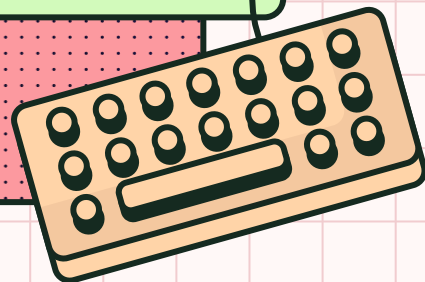
2022 SMARCLE PYTHON WEEK



22 양지윤, 22 윤수경, 22 천승원

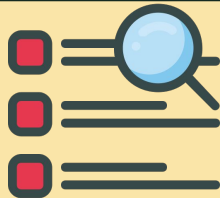


파이리





CONTENTS



01

WHAT
게임 소개

02

HOW
파이썬 코드 설명

03

PRODUCT
결과물

04

THOUGHTS
소감

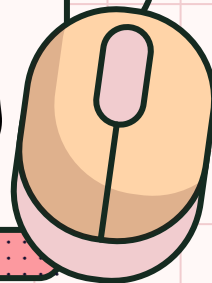




EARTHWORM GAME



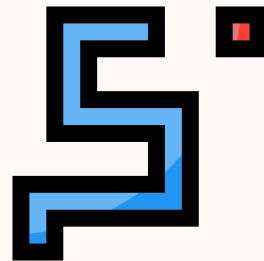
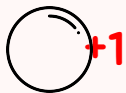
파이썬을 이용한 지렁이 게임



EARTHWORM GAME

방향키를 눌러 지렁이가 갈 위치를 설정하여 위치가 지정된 먹이를 먹으면
점수를 획득!

하얀색 공을 먹으면 1점이 추가되고,
파란색 삼각형 먹이를 먹으면 2점이 추가된다.



GREAT!



참고 YouTube
'코드덤code your dreams'

파이리팀만의 특별한 지렁이 게임

추가 기능

SCORE!

획득 가능 점수

파란색 삼각형 먹이를 추가하여
2점을 획득할 수 있다



플레이 시간

사용자가 게임을 진행한 시간을
확인할 수 있다



코드 기본 세팅

```
from turtle import Turtle, Screen // Turtle 그래픽 모듈 실행
                                   // Turtle()로 개체 생성 가능
import time                       // time 모듈 실행
import random                     // random 모듈 실행
```

GAME

방향 조정 함수

```
def up(): // up 함수
    if worm[0].heading() != 270: // 지렁이 머리가 아래쪽(270도)을 향하고 있지 않은 경우
        snakes[0].setheading(90)
def down(): // down 함수
    if snakes[0].heading() != 90: // 지렁이 머리가 위쪽(90도)을 향하고 있지 않은 경우
        snakes[0].setheading(270)
def right(): // right 함수
    if snakes[0].heading() != 180: // 지렁이 머리가 왼쪽(180도)을 향하고 있지 않은 경우
        snakes[0].setheading(0)
def left(): // left 함수
    if snakes[0].heading() != 0:
        snakes[0].setheading(180)
```

DIRECTION

지렁이 몸체 만들기 함수

```
def create_snake(pos):  
    snake_body = Turtle()           // 지렁이를 하나의 객체로 지정  
    snake_body.shape("square")       // 지렁이 몸체 모양 설정  
    snake_body.color("orangered")    // 지렁이 색깔 설정  
    snake_body.up()  
    snake_body.goto(pos)  
    snakes.append(snake_body)        // 지렁이 몸체 추가
```

SNAKE

랜덤위치 지정 함수(먹이 위치)

```
def rand_pos():  
    rand_x = random.randint(-250,250) // 함수를 통해 X,Y값 랜덤 지정  
    rand_y = random.randint(-250,250)  
    return rand_x, rand_y // 랜덤으로 지정된 X,Y값 반환
```




RANDOM

점수 갱신 함수

```
def score_update():  
    global score  
    score += 1  
    score_pen.clear()  
    score_pen.write(f"점수 : {score}", font = ("", 15, "bold"))
```

// 하얀색 공(먹이 1)을 먹었을때
// score을 전역변수로 선언
// 하얀색 공을 먹었을때 1점 추가
// 전에 표시했던 내용을 지우고 새롭게 갱신한 내용 표시



```
def score_update2():  
    global score  
    score += 2  
    score_pen.clear()  
    score_pen.write(f"점수 : {score}", font = ("", 15, "bold"))
```

// 파란색 삼각형(먹이2)을 먹었을때
// 파란색 공을 먹었을때 2점

SCORE

버틴 시간 갱신 함수

```
def resist_time_update():  
    global resist_time          // resist_time을 전역변수로 선언  
    resist_time += 0.1          // 시간이 지날때마다 0.1초씩 추가(스크린 갱신 주기)  
    resist_time_pen.clear()  
    resist_time_pen.write(f"시간: {int(resist_time)} 초", font = ("", 15, "bold"))
```



TIME

Game Over 표시 함수

```
def game_over():  
    score_pen.goto(0,0)           // 중앙에 글자를 표시하도록 위치 조정  
    score_pen.write("Game Over", False, "center", (10, 30, "bold"))  
    // game over 시스템 메시지 표시
```



THE END

Screen 기본 세팅

```
screen = Screen()  
screen.setup(600,600)  
screen.bgcolor("khaki")  
screen.title("Snake Game")  
screen.tracer(0)
```

// 스크린 세팅
// 스크린 크기 조정
// 스크린 색깔 조정
// 스크린 이름 조정
// 스크린 트레이서 0(버퍼링 최소화)



SCREEN

변수들 초기화/ 기본 몸체 세팅

```
start_pos = [(0,0), (-20,0), (-40,0)] // 지렁이 초기위치 설정
snakes = []
score = 0 // 점수 초기화
resist_time = 0 // 시간 초기화

for pos in start_pos: // 처음 몸체 세팅
    create_snake(pos)
```



RESET

먹이 설정

```
food = Turtle()           // 하얀색 공(먹이 1)을 객체로 지정
food.shape("circle")       // 하얀색 공(먹이 1) 모양 지정
food.color("snow")         // 하얀색 공(먹이 1) 색깔 지정
food.up()
food.speed(0)
food.goto(rand_pos())      // 하얀색 공(먹이 1)을 랜덤 위치에 생성

food2 = Turtle()           // 파란색 삼각형(먹이 2)을 객체로 지정
food2.shape("triangle")    // 파란색 삼각형(먹이 2) 모양 지정
food2.color("blue")        // 파란색 삼각형(먹이 2) 색깔 지정
food2.up()
food2.speed(0)
food2.goto(rand_pos())     // 파란색 삼각형(먹이 2) 랜덤 위치
```

FOOD



점수와 시간 표시 초기화/ 표시

```
score_pen = Turtle()           // 점수 표시를 객체로 지정
score_pen.ht()
score_pen.up()
score_pen.goto(-270,250)       // 점수 표시 위치 지정
score_pen.write(f"점수 : {score}", font = ("", 15, "bold")) // 점수 표시

resist_time_pen = Turtle()     // 시간 표시를 객체로 지정
resist_time_pen.ht()
resist_time_pen.up()
resist_time_pen.goto(-270, 230) // 시간 표시 위치
resist_time_pen.write(f"시간 : {resist_time}", font = ("", 15, "bold"))
```

사용자의 입력 받기

```
screen.listen()  
screen.onkeypress(up,"Up")           // 키보드 방향키 입력 따라 방향 조정  
screen.onkeypress(down,"Down")       // 각 방향 조정 함수 호출  
screen.onkeypress(left,"Left")  
screen.onkeypress(right,"Right")
```

INPUT



게임 실행코드

```
game_on = True
while game_on:
    screen.update()
    time.sleep(0.1)
    resist_time_update()
```

```
// 임의의 game_on변수를 True로 설정
// game_on 변수가 False가 되기 전까지
// 스크린 update를 진행하며 계속 실행
// 버퍼링 방지 + 시간 측정
// 측정한 시간에 따라 시간 갱신
```

RUN

지렁이가 먹이를 먹을때

```
if snakes[0].distance(food) < 15:    // 하얀색 공(먹이1)을 먹을때
    score_update()                  // 점수 갱신 함수 호출
    food.goto(rand_pos())           // 랜덤 위치 함수 호출 -> 먹이 옮김
    create_snake(snakes[-1].pos())  // 몸체 길이 추가 x1

if snakes[0].distance(food2) < 15:   // 파란색 삼각형(먹이2)을 먹을때
    score_update2()                 // 점수 갱신 함수 호출
    food2.goto(rand_pos())          // 몸체 길이 추가 x2
    create_snake(snakes[-1].pos())
    create_snake(snakes[-1].pos())
```

EATING

Game Over(종료) 조건 설정

```
if snakes[0].xcor() > 280 or snakes[0].xcor() < -280 \
    or snakes[0].ycor() > 280 or snakes[0].ycor() < -280:
    game_on = False // 벽의 X위치, Y위치에 몸체가 닿았을때
    game_over()      // game_on 변수를 False로 바꿔 반복문 종료
                    // game_over표시 함수 호출

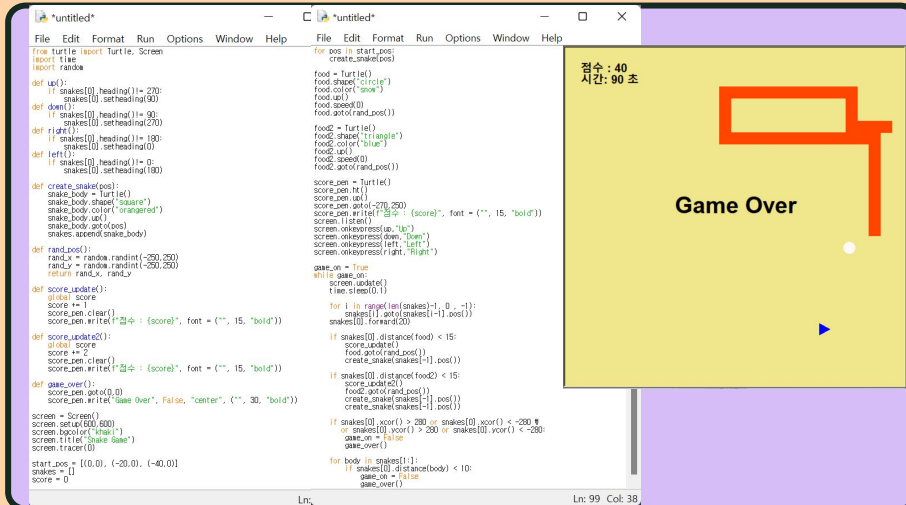
for body in snakes[1:]: // 지렁이가 자기 몸체에 닿았을때
    if snakes[0].distance(body) < 10:
        game_on = False // game_on 변수를 False로 바꿔 반복문 종료
        game_over()      // game_over표시 함수 호출
```

OVER





실행 화면



```
File Edit Format Run Options Window Help File Edit Format Run Options Window Help
from turtle import Turtle, Screen
import time
import random

def up():
    if snakes[0].heading() != 270:
        snakes[0].setheading(90)

def down():
    if snakes[0].heading() != 90:
        snakes[0].setheading(270)

def right():
    if snakes[0].heading() != 180:
        snakes[0].setheading(0)

def left():
    if snakes[0].heading() != 0:
        snakes[0].setheading(180)

def create_snake(pos):
    snake_body = Turtle()
    snake_body.shape("square")
    snake_body.color("orange")
    snake_body.penup()
    snake_body.goto(pos)
    snakes.append(snake_body)

def rand_pos():
    rand_x = random.randint(-250, 250)
    rand_y = random.randint(-250, 250)
    return rand_x, rand_y

def score_update():
    score += 1
    score_pen.clear()
    score_pen.write(f"점수 : {score}", font=("", 15, "bold"))

def score_update2():
    if snake_score:
        score += 2
        score_pen.clear()
        score_pen.write(f"점수 : {score}", font=("", 15, "bold"))

def game_over():
    score_pen.goto(0, 0)
    score_pen.write("Game Over", False, "center", ("", 30, "bold"))

screen = Screen()
screen.setup(500, 500)
screen.bgcolor("black")
screen.title("Snake Game")
screen.tracer(0)

start_pos = [(0, 0), (-20, 0), (-40, 0)]
snakes = []
score = 0

for pos in start_pos:
    create_snake(pos)

food = Turtle()
food.shape("circle")
food.color("red")
food.penup()
food.goto(rand_pos())

food2 = Turtle()
food2.shape("triangle")
food2.color("blue")
food2.penup()
food2.goto(rand_pos())

score_pen = Turtle()
score_pen.penup()
score_pen.goto(-250, 250)
score_pen.write(f"점수 : {score}", font=("", 15, "bold"))
screen.listen()
screen.onkeypress(up, "Up")
screen.onkeypress(down, "Down")
screen.onkeypress(left, "Left")
screen.onkeypress(right, "Right")

game_on = True
while game_on:
    screen.update()
    time.sleep(0.1)

    for i in range(len(snakes)-1, 0, -1):
        snakes[i].goto(snakes[i-1].pos())
        snakes[i].forward(20)

    if snakes[0].distance(food) < 15:
        score_update()
        food.goto(rand_pos())
        create_snake(snakes[-1].pos())

    if snakes[0].distance(food2) < 15:
        score_update2()
        food2.goto(rand_pos())
        create_snake(snakes[-1].pos())

    if snakes[0].xcor() > 250 or snakes[0].xcor() < -250 or snakes[0].ycor() > 250 or snakes[0].ycor() < -250:
        game_on = False
        game_over()

    for body in snakes[1:]:
        if snakes[0].distance(body) < 10:
            game_on = False
            game_over()
```



04

IMPRESSION

소감

우리의 소감이 궁금해여?



JIYUN

파이썬을 이용하여 게임을
실행할 수 있다는 것을 알 수
있었고 게임을 만들며 여러
코드를 접해볼 수 있어 좋았다.



SOOKYOUNG

파이썬으로 게임을
만들어보며 다양한 코드를
배워보는 시간이 된 것 같아
정말 유익하고 재미있었다.



SEUNGWON

파이썬으로 새롭게 turtle
모듈을 공부하고 사용하여
게임 프로젝트를 완성할 수
있어 좋았다.

