

2023-2 강원중 상상 플레이어

강원중학교

# Python 게임 프로그래밍

강원대학교사범대학부설고등학교  
장원재

# 학습 순서

01 게임 개발을 위한 기본 설정  
Python Idle과 Pygame

02 스네이크 게임 만들기  
게임판 생성 및 규칙 생성

03 횡스크롤 게임 만들기  
게임판, 플레이어, 규칙 생성



# 01

Python 기본

## 게임 개발을 위한 기본 설정

# Python Idle 설치

프로그래밍 환경 만들기



❖ Python 프로그래밍 환경

1. Google Colab과 같은 클라우드 환경

2. Python 기본 Idle

3. Anaconda와 Jupyter 노트북

4. Python + Vscode

-> 여기서는 2번의 기본 Idle을 설치하고 설정해볼 것!

# Python Idle 설치

<https://www.python.org/> 접속하여 파이썬 다운로드 후 설치하기

The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links: Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a dark blue header with the Python logo, a 'Donate' button, a search bar, and a 'Socialize' button. A secondary navigation bar contains links: About, Downloads, Documentation, Community, Success Stories, News, and Events. The 'Downloads' link is active, and a dropdown menu is open, listing options: All releases, Source code, Windows, macOS, Other Platforms, License, and Alternative Implementations. The 'Windows' option is highlighted with a red rectangle. To the right of this menu, a 'Download for Windows' section is visible, featuring a button for 'Python 3.11.5' (also highlighted with a red rectangle), a note about Windows 7 compatibility, and a link to view the full list of downloads. At the bottom of the page, a footer states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

# For loop on a

>>> numbers = [

>>> product = 1

>>> for number

... product

...

>>> print('The

The product is:

All releases

Source code

Windows

macOS

Other Platforms

License

Alternative Implementations

Download for Windows

Python 3.11.5

Note that Python 3.9+ cannot be used on Windows 7 or earlier.

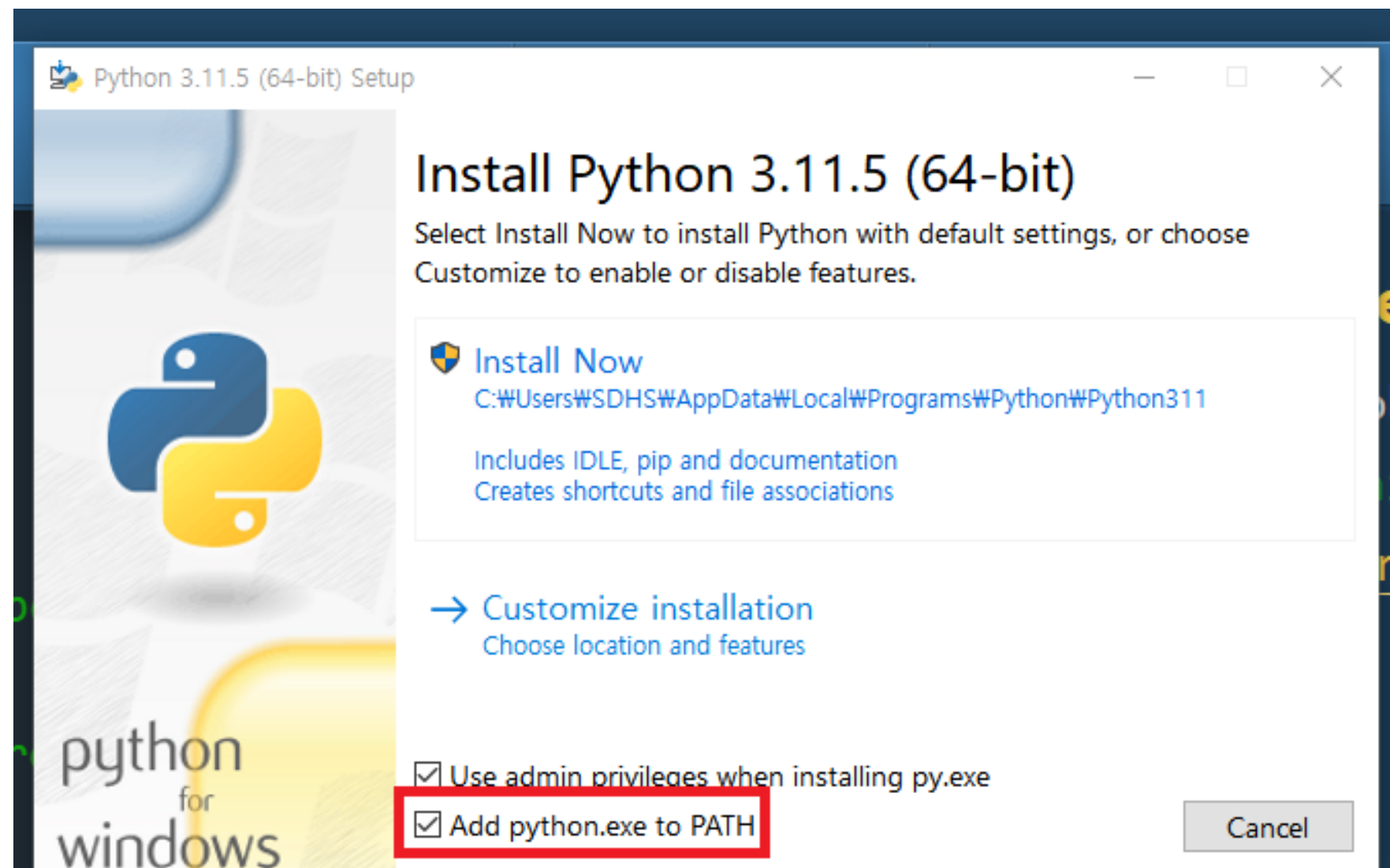
Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

# Python Idle 설치

설치파일 실행 후 반드시 환경변수 추가 체크하기!!!!!!



# Python 환경 체크하기

키보드에서 윈도우 키 + R 클릭 후 cmd누른 후 명령 프롬프트 열기

실행



프로그램, 폴더, 문서, 또는 인터넷 주소를 입력하여 해당 항목을 열 수 있습니다.

열기(O):

cmd

확인

취소

찾아보기(B)...

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19044.2965]  
(c) Microsoft Corporation. All rights reserved.

C:\Users\SDHS>python --version  
Python 3.9.5

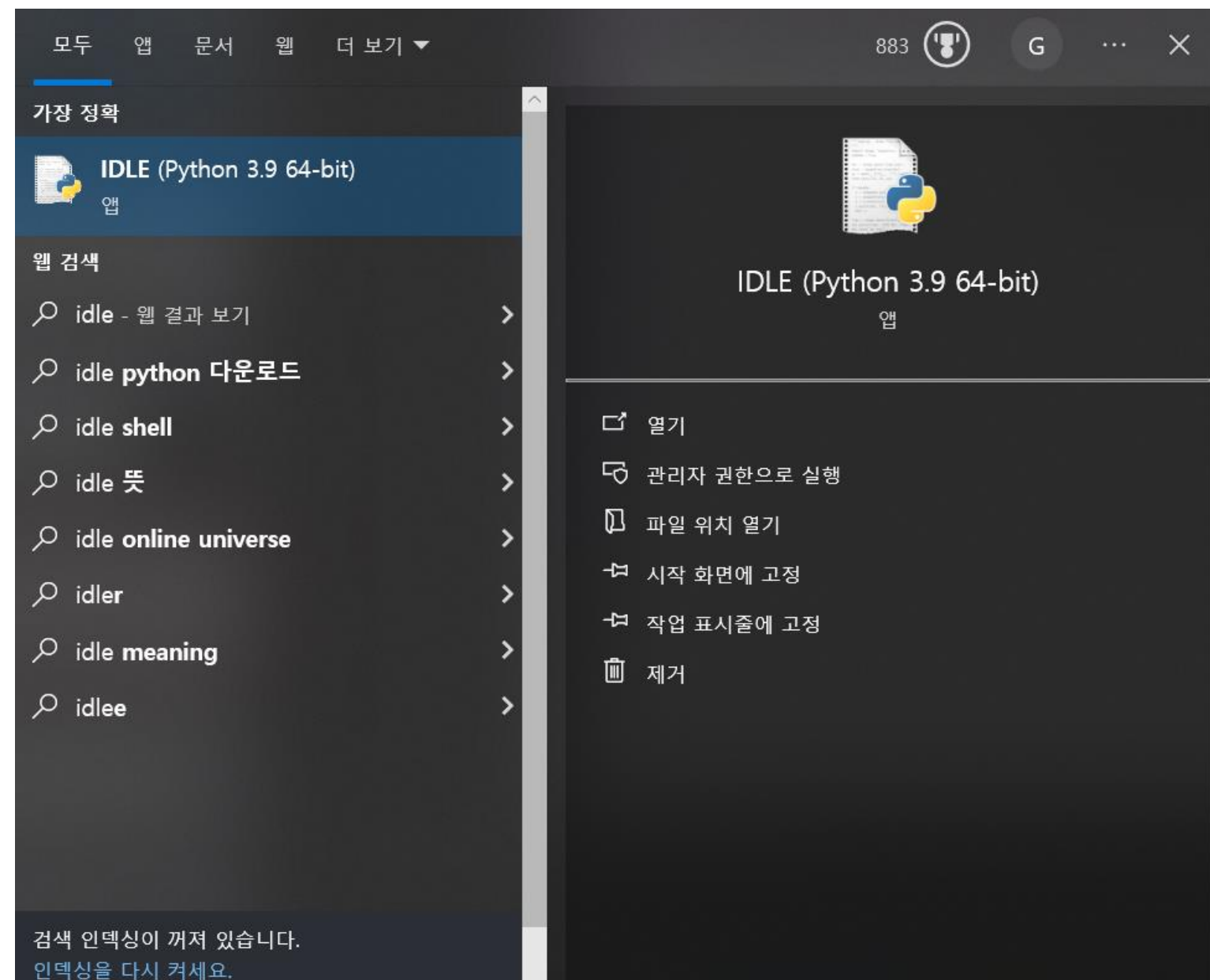
C:\Users\SDHS>pip list

Package	Version
altgraph	0.17
asttokens	2.2.1
backcall	0.2.0
certifi	2022.5.18.1
cffi	1.15.1
charset-normalizer	2.0.12
click	8.1.3
colorama	0.4.4
comm	0.1.3
cryptography	38.0.1
debugpy	1.6.7
decorator	5.1.1
executing	1.2.0
Flask	2.1.2
future	0.18.2
github3.py	3.2.0
idna	3.3
importlib-metadata	4.11.4
ipykernel	6.23.2
ipython	8.14.0
itsdangerous	2.1.2

1. Python -version (설치된 파이썬 버전 확인)
  2. Pip list (설치된 라이브러리 확인)
- > 결과가 출력이 안되면? \* 환경변수 오류 \* 설치오류

# Python Idle 실행해보기

윈도우 하단 돋보기 창에서 Idle 입력 후 Idle 실행



A screenshot of the IDLE Shell 3.9.5 window. The window title is 'IDLE Shell 3.9.5'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following information: 'Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32'. Below this, it says 'Type "help", "copyright", "credits" or "license()" for more information.' and the prompt '&gt;&gt;&gt;' is visible.



# Python 게임 프로그래밍을 위한 라이브러리 설치

Pip install Pygame

## ❖ Python 라이브러리 설치

1. Python은 프로그래밍을 위한 도구의 모음인 다양한 라이브러리를 제공한다.
2. Pip install (라이브러리명)을 사용하면 손쉽게 라이브러리를 설치할 수 있다.
3. Pip list를 입력하면 설치된 라이브러리 목록 및 버전을 확인할 수 있다.
4. 라이브러리(파일의 집합) > 모듈(파일) > 클래스, 메서드(성분)으로 이해하면 된다.
5. 명령 프롬프트(창) 실행 후 pip install pygame을 실행 해 pygame 라이브러리를 설치하자

# Python 게임 프로그래밍을 위한 라이브러리 설치

Pip install Pygame

```
C:\#Users\#SDHS>pip install pygame
Collecting pygame
  Obtaining dependency information for pygame from https://files.pythonhosted.org/packages/a6/4f/1cf0af568631286ab35cc87dfb3ab0383061c1591ef83ad2bfd9ed3bf82/pygame-2.5.1-cp39-cp39-win_amd64.whl.metadata
  Downloading pygame-2.5.1-cp39-cp39-win_amd64.whl.metadata (13 kB)
  Downloading pygame-2.5.1-cp39-cp39-win_amd64.whl (10.6 MB)
    ----- 10.6/10.6 MB 43.7 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.5.1

C:\#Users\#SDHS>_
```

# Python 게임 프로그래밍을 위한 라이브러리 설치 확인

```
numpy 1.20.3
packaging 23.1
parso 0.8.3
pefile 2021.5.24
pickleshare 0.7.5
pip 23.2.1
platformdirs 3.5.3
plotly 5.10.0
prompt-toolkit 3.0.38
psutil 5.9.5
pure-eval 0.2.2
pycparser 2.21
pygame 2.5.1
Pygments 2.15.1
pyinstaller 4.5.1
pyinstaller-hooks-contrib 2021.3
PyJWT 2.5.0
pynput 1.7.6
python-dateutil 2.8.2
pytube 12.1.3
pywin32 306
pywin32-ctypes 0.2.0
pymq 25.1.0
requests 2.27.1
setuptools 56.0.0
six 1.16.0
stack-data 0.6.2
style 1.1.0
tenacity 8.0.1
tornado 6.3.2
```

Pip list

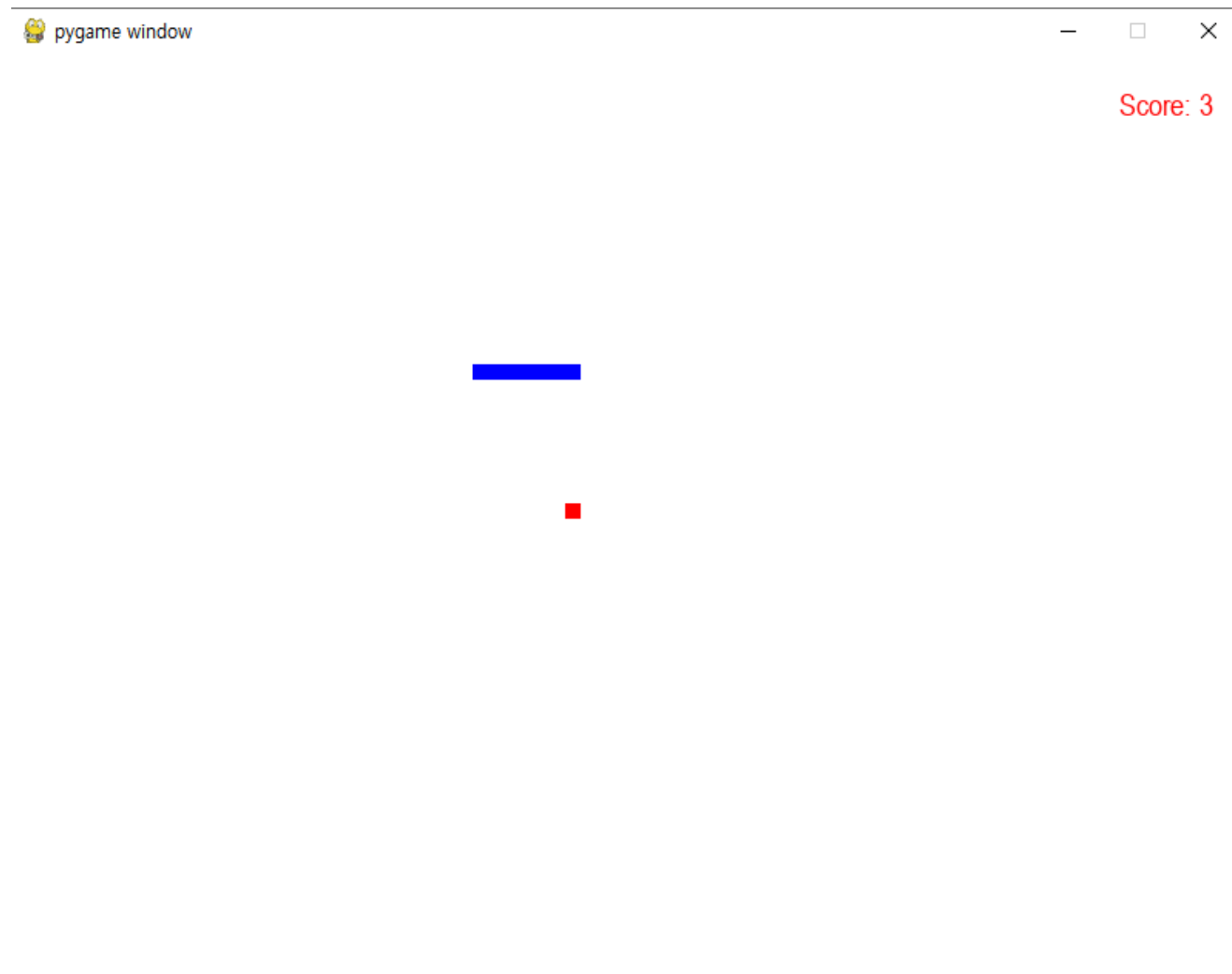
# 02

Python 게임 만들기

스네이크 게임 만들기

# Snake 게임이란? [Github.com/carlosquerman](https://github.com/carlosquerman)

뱀 오브젝트, 과일 오브젝트로 이루어진 상호작용 게임



## ❖ Snake 게임

1. 화면에 뱀을 의미하는 오브젝트와
2. 과일을 의미하는 오브젝트가 생성되고
3. 사용자가 뱀을 조종하여
4. 과일을 먹고
5. 과일을 먹으면 뱀의 몸체가 늘어나는 구조
6. 이 과정에서 뱀의 머리와 뱀의 몸통이 충돌하거나
7. 뱀이 벽과 충돌하면 게임 오버를 출력함.

# Snake 게임 만들기

게임을 만들기 위한 구성요소를 순서대로 구현하는 과정으로

## ❖ Snake 게임

1. 게임의 배경이 되는 (        )을 생성
2. 게임의 주 대상(오브젝트인) ()과 ()을 생성
3. 뱀을 ()하게 하는 코드 작성
4. 과일을 ()으면 과일을 다시 ()하는 코드 작성
5. 게임오버 조건 1 : ()와의 충돌
6. 게임오버 조건 2 : ()와의 충돌
7. 점수 표시 기능 추가

# Snake 게임 만들기

## 1. 게임의 배경이 되는 게임 판 생성

### 1 게임 기본 설정 (게임판 생성)

```
l: import pygame
import time
import random

# 색상 설정
WHITE = (255, 255, 255)
RED = (255, 0, 0)
BLUE = (0, 0, 255)
BLACK = (0,0,0)

# 화면 설정 (가로 800, 세로 600)
WIDTH, HEIGHT = ,
win = pygame.display.set_mode((WIDTH, HEIGHT))

# 게임 설정
running = True
score = 0
```

### ❖ 게임 기본 게임판 생성

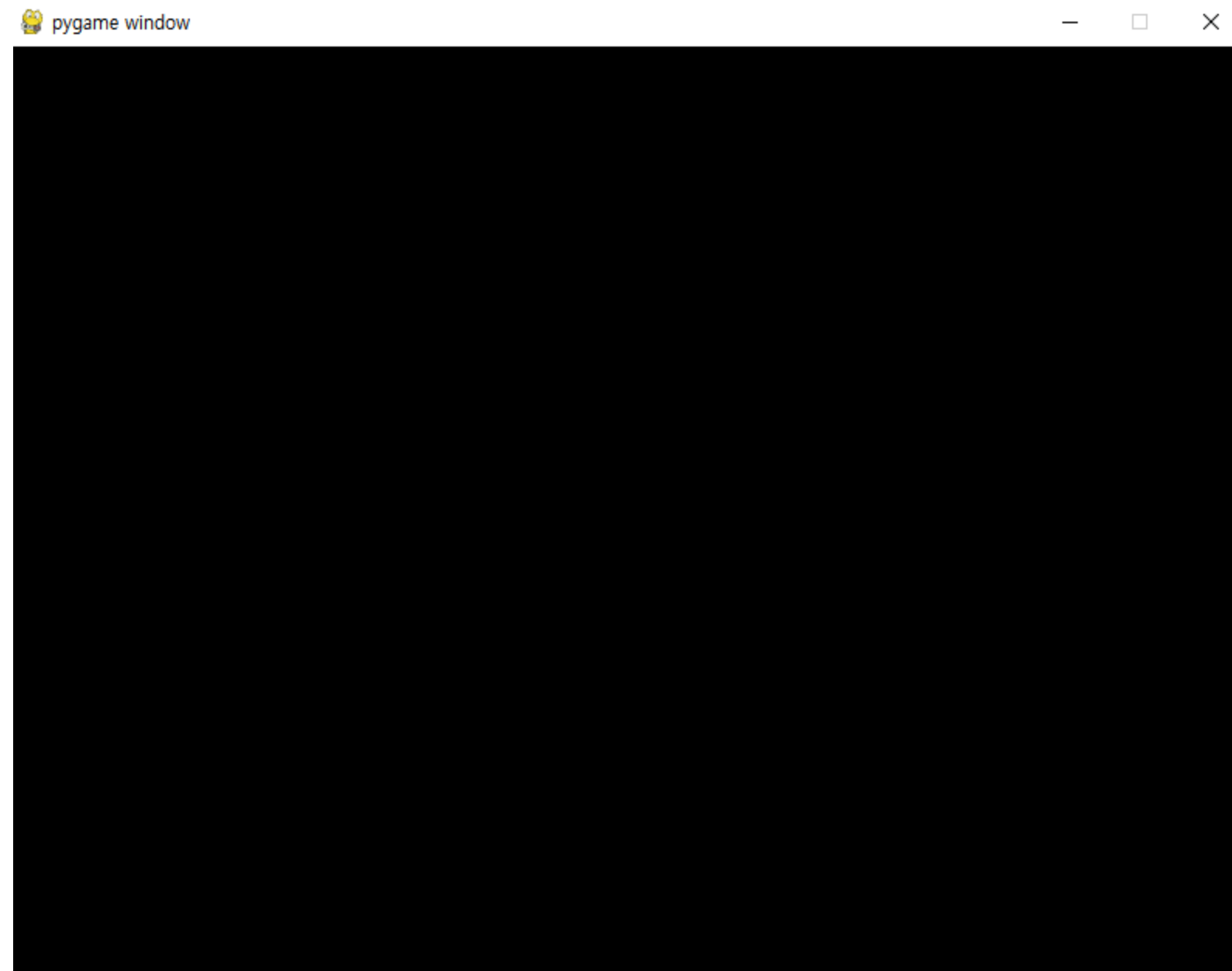
- 게임 프로그래밍을 위한 모듈 불러오기
- 색상을 변수에 저장
- 화면의 가로, 세로 길이 설정을 통해 게임

### 화면 창 만들기

- 게임 설정 및 기본 점수 설정

# Snake 게임 만들기

실행 결과





# Snake 게임 만들기

## 2. 뱀과 과일 생성

```
# 뱀 설정
```

```
snake_pos = [[100, 50], [, 50], [, 50]]
```

```
base_speed = 10
```

```
snake_speed = [, 0]
```

```
# 과일 설정
```

```
fruit_pos = [random.randrange(1, WIDTH//)*10, random.randrange(1, HEIGHT//)*10]
```

```
fruit_spawned = True
```

```
while running:
```

```
    # 화면 그리기
```

```
    win.fill(WHITE) #화면 흰색으로 채우기
```

```
    for pos in snake_pos:
```

```
        pygame.draw.rect(win, BLUE, pygame.Rect(pos[0], pos[1], , )) #win 창에 파란색 직사각형으로 x축 10, y축
```

```
        pygame.draw.rect(win, RED, pygame.Rect(fruit_pos[0], fruit_pos[1], , )) #win창에 빨간색 직사각형으로 x축 10, y축 10
```

```
    pygame.display.update()
```

```
    pygame.time.Clock().tick(30)
```

# Snake 게임 만들기

## 2. 뱀과 과일 생성

### ❖ 뱀과 과일 생성

- 사이즈 3을 가진 기본 뱀 생성 (뱀의 각 부분의 크기는 10)
- 기본 속도는 10이며, 뱀의 속도는  $\times$ 축으로 기본 속도로 움직임
- 과일은 10의 크기를 가지며 랜덤으로 생성되도록 함
- While 루프 안에서 뱀과 과일을 그려줌

# Snake 게임 만들기

## 2. 뱀과 과일 생성 실행결과



# Snake 게임 만들기

## 3. 뱀 이동 기능 추가

```
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                snake_speed = [0, -1]
            elif event.key == pygame.K_DOWN:
                snake_speed = [0, 1]
            elif event.key == pygame.K_LEFT:
                snake_speed = [-1, 0]
            elif event.key == pygame.K_RIGHT:
                snake_speed = [1, 0]

# 뱀의 움직임
snake_pos.insert(0, list(snake_pos[0])) # 먼저 뱀의 머리 부분을 복사

# 뱀이 움직이는 방향으로 머리 부분 이동
snake_pos[0][0] += snake_speed[0]
snake_pos[0][1] += snake_speed[1]
```

### ❖ 키보드 조작 통한 뱀 이동 기능 추가

- 사용자 조작 통해 뱀의 이동 속도

조절

- 뱀이 움직일 때 기본적으로 뱀의 머리를 복사하도록 함(왜? : 과일 먹기에 따른 변화 반영 위해)

# Snake 게임 만들기

## 4. 과일 먹기 기능 추가

```
# 뱀 성장 및 과일 생성
if snake_pos[] == fruit_pos:
    score += 1
    fruit_spawned =
else:
    snake_pos.pop() # 뱀이 과일을 먹지 않았을 때만 꼬리 부분 제거 (머리가 복사되어 움직이기 때문에 꼬리 누

if not fruit_spawned:
    fruit_pos = [random.randrange(1, WIDTH//10)*10, random.randrange(1, HEIGHT//10)*10]
    fruit_spawned = True
```

# Snake 게임 만들기

## 4. 과일 먹기 기능 추가

### ❖ 과일 먹기 기능 추가

- 만약 뱀의 머리와 과일의 위치가 같아진다면 (뱀이 과일을 먹었다면)
- 점수를 1 증가시키고
- 과일의 생성 여부를 False로 수정 -> 과일이 재생성 될 수 있도록
- 과일을 먹지 않았다면 생성된 뱀의 복사 부분을 제거 (pop함)

# Snake 게임 만들기

## 5. 게임오버 조건 1 (벽충돌)

```
# 벽 충돌 확인  
if snake_pos[0][0] >= 100 or snake_pos[0][0] < 0 or snake_pos[0][1] >= 100 or snake_pos[0][1] < 0:  
    running = False
```

### ❖ 게임오버 조건 1 (벽충돌)

- 뱀의 위치가 벽의 위치를 넘어간다면 (충돌한다면)
- 실행 조건을 False로 바꿈

# Snake 게임 만들기

## 6. 게임오버 조건 2 (몸과 머리 충돌)

```
# 뱀 몸통 충돌 확인  
for pos in snake_pos[:]:  
    if snake_pos[0] == :  
        running = False
```

### ❖ 게임오버 조건 2 (머리와 몸통 충돌)

- 뱀의 전체 부분에서 만약 머리가 다른 부분과 같아진다면 (충돌한다면)
- 게임 실행 조건을 False로 수정하기



# Snake 게임 만들기

## 7. 점수 표현 기능 추가

*#점수표현*

```
score_text = font.render(f'Score: {score}', True, (, 0, 0))  
score_rect = score_text.get_rect(topright=(WIDTH - 20, 20))  
win.blit(, )  #(텍스트, 위치)
```

### ❖ 점수 표현 기능 추가

- Score 점수에 저장된 게임 점수를 화면에 출력하기
- 출력의 위치는 전체 게임 화면의 크기를 고려하여 설정

# Snake 게임 만들기

## 8. 게임 오버 화면 출력

*#글씨체 설정*

```
#font = pygame.font.SysFont('arial', 20)
font = pygame.font.SysFont('kang.ttf', 20)
font_gameover = pygame.font.SysFont('kang.ttf', 72)
```

*#게임 오버 폰트 및 텍스트 설정*

```
game_over_text = font_gameover.render('Game Over', True, (0, 0, 255)) # 빨간색으로 설정
game_over_rect = game_over_text.get_rect(center=(WIDTH // 2, HEIGHT // 2)) # 화면 중앙에 위치
```

```
if not running:
```

```
    win.blit(, ) #텍스트, 위치
```

```
    pygame.display.update() # 게임 오버 텍스트를 화면에 바로 업데이트
```

```
    time.sleep(3) # 3초 동안 게임 오버 텍스트를 화면에 유지
```

## 이외에도...

게임의 기능을 많이 추가할 수 있다.

### ❖ Snake 게임의 기능 추가

1. 특정 점수에 도달하면 Stage2를 출력하도록 설정하기
2. 사용자가 난이도를 고르면 그에 따라 게임을 시작하기
3. 과일 이외에 먹으면 게임을 더 쉽게 해주거나, 게임을 더 어렵게 해주는 기능 추가하기
4. 이외에도 무한!!!!
5. 이런 방식으로 다양한 게임을 만들 수 있다!

# 03

Python 게임 만들기

윙스크롤 게임 만들기

# 횡스크롤 게임이란? Github.com/carlosquperman

배경이 횡으로 흘러가고 그런 과정에서 플레이어, 적, 보스 캐릭터를 만들고 적을 무찌르는 게임



## ❖ 횡스크롤 게임

1. 배경은 왼쪽 방향으로 흘러가는 구조
2. 게임을 실행하면 보스 캐릭터가 생성됨
3. 게임을 실행하면 쥘병 캐릭터가 생성됨
4. 키보드로 조작할 수 있는 플레이어를 배경에 만들고
5. 키보드로 미사일이나 빛의 구를 쏘아 적을 무찌르는 게임!

# 횡스크롤 게임 만들기

게임을 만들기 위한 구성요소를 순서대로 구현하는 과정으로

## ❖ 횡스크롤 게임

1. 게임의 배경이 되는 ( )을 생성하고 흘러가게 하기
2. 게임의 조작 대상인 ()를 생성하게 하기
3. 게임의 플레이어 캐릭터를 ()로 움직이게 하기
4. 게임을 방해하는 () 캐릭터를 생성해 움직이게 하기
5. 게임을 방해하는 () 캐릭터를 생성해 움직이게 하기
6. 플레이어가 키보드를 누르면 적을 무찌르는 빛의 구 쏘게 하기
7. 빛의 구를 맞으면 적을 없애고 일정 시간 이후에 재생성하게 하기
8. 기타 기능 추가!!!!

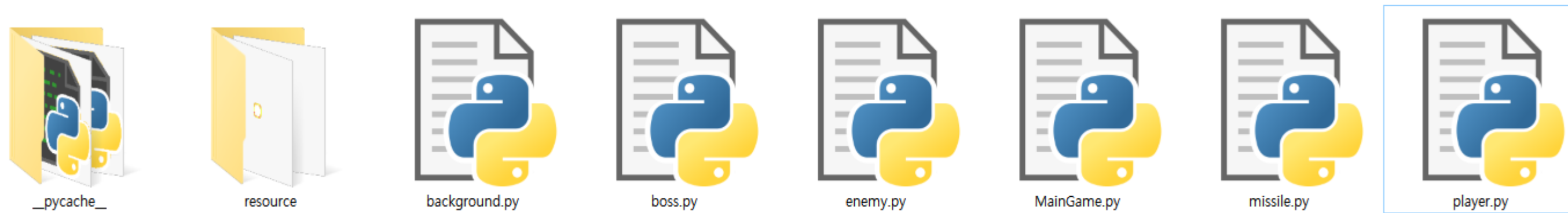
# 힉스크롤 게임 만들기

게임의 구성 요소를 별도 파일로 저장하기

- ❖ 게임 구성 요소를 클래스로 생성한 후, 생성한 클래스를 별도의 모듈로 저장하기
- 1. 객체지향 구조 프로그래밍(OOP)
- 2. 객체를 중심으로 프로그래밍을 수행하게 하는 것
- 3. 추상화, 상속, 캡슐화, 다형성
- 4. 재사용 및 유지보수가 용이함
- 5. 객체를 중심으로 프로그래밍을 수행하고, 객체를 재정의함으로 다양한 프로그램을 수행함.

# 윙스크롤 게임 만들기

게임의 구성 요소를 별도 파일로 저장하기



## ❖ 각각의 모듈 및 리소스

- MainGame.py : 게임의 가장 핵심 파일
- Background, boss, enemy, missile, player.py 프로그래밍에 사용되는 각각의 객체에 대한 정보를 파일로 저장한 것.
- Resource : 게임에 활용되는 리소스 (이미지, 사운드 등을 저장한 폴더)



# 횱스크롤 게임 만들기

## 1. Background.py

```
import pygame

class Background:
    def __init__(self, screen):
        self.screen = screen
        self.bg_img = pygame.image.load("resource/game_background.png")
        self.bg_img2 = pygame.image.load("resource/game_background.png")
        self.x1=0
        self.y1=0
        self.x2=1920
        self.y2=0

    def draw(self):
        self.screen.blit(self.bg_img, (self.x1,self.y1))
        self.screen.blit(self.bg_img2, (self.x2,self.y2))

    def update(self):
        self.x1 -= 2
        self.x2 -= 2

        if self.x1 <= -1920:
            self.x1=1920
        if self.x2 <= -1920:
            self.x2 = 1920
```

# 횱스크롤 게임 만들기

## 2. Boss.py

```
import pygame
import random

class Boss :

    def __init__(self, screen):
        self.screen= screen
        self.boss_img = [ pygame.image.load("resource/boss_run/play_mombear_run_0.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_1.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_2.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_3.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_4.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_5.png"),
                           pygame.image.load("resource/boss_run/play_mombear_run_6.png") ]

        self.bsx = 800
        self.bsy = 400
        self.std_count = 0

        self.hp_img = pygame.image.load("resource/hp.png") #초록색 = 풀피
        self.bar_img = pygame.image.load("resource/hp_bar.png") #빨간색 뺀피

        self.moveD = True
        self.moveU = False

        self.fhp = 80
        self.hp = 80
        self.isDead = False

    def draw(self):
        if not self.isDead:
            self.screen.blit(self.boss_img[self.std_count//60], (self.bsx, self.bsy))
            self.screen.blit(self.bar_img , (self.bsx + 30 , self.bsy), (0,0 ,self.fhp, 20) ) #hp는 보스의 머리위에 그린다.
            self.screen.blit(self.hp_img , (self.bsx + 30 , self.bsy), (0,0 ,self.hp, 20) ) #hp는 보스의 머리위에 그린다.

    def update(self):

        if self.moveD and self.bsy <= 1000:
            self.bsy += 1

        if self.bsy > 1000:
            self.bsx = random.randrange(0, 1800)
            self.moveD = False
            self.moveU = True

        if self.moveU and self.bsy >= 0:
            self.bsy -= 1

        if self.bsy < 0:
            self.bsx = random.randrange(0, 1800)
            self.moveU = False
            self.moveD = True

    def animation(self) :
        self.std_count += 1

        if self.std_count >= 419:
            self.std_count =0
```

# 횱스크롤 게임 만들기

## 3. Enemy.py

```
import pygame
import random

class Enemy :

    def __init__(self, screen):
        self.screen = screen
        self.enemy_img = pygame.image.load("resource/enemy/play_babybear_run_0.png")
        self.ex = 1000
        self.ey = random.randrange(0, 900)

        self.isDead = False

    def draw(self):
        if not self.isDead:
            self.screen.blit(self.enemy_img, (self.ex, self.ey))

    def update(self):

        self.ex -= 3

        if self.ex < -100:
            self.ex = 1900
            self.ey = random.randrange(0, 900)

        if self.isDead:
            self.ex = 1920
            self.ey = random.randrange(0, 900)
            self.isDead = False
```

# 횡스크롤 게임 만들기

## 4. Missile.py

```
import pygame

class Missile:
    def __init__(self, screen, boss, enemy):
        self.screen = screen

        self.boss = boss
        self.enemy = enemy

        self.missile_img = pygame.image.load('resource/bullet.png')
        self.missile_img2 = pygame.image.load('resource/bullet2.png')

        self.missile_pos = []
        self.missile_posU = []

        self.isShot = False

    def draw(self):
        if len(self.missile_pos) != 0:
            for self.x, self.y in self.missile_pos:
                self.screen.blit(self.missile_img, (self.x, self.y))

        if len(self.missile_posU) != 0:
            for self.x, self.y in self.missile_posU:
                self.screen.blit(self.missile_img2, (self.x, self.y))
```

```
    def update(self):

        if len(self.missile_pos) != 0:
            for self.i, self.mxy in enumerate(self.missile_pos):
                self.mxy[0] += 10
                self.missile_pos[self.i][0] = self.mxy[0]

                if self.missile_pos[self.i][0] > 2000:
                    del self.missile_pos[self.i]

                if self.boss.bsx <= self.mxy[0] < self.boss.bsx+100:
                    if self.boss.bsy <= self.mxy[1] <= self.boss.bsy+100:
                        self.isShot = True
                        del self.missile_pos[self.i]

                if self.enemy.ex <= self.mxy[0] < self.enemy.ex + 100:
                    if self.enemy.ey <= self.mxy[1] <= self.enemy.ey+100:
                        self.isShot = True
                        del self.missile_pos[self.i]

        if self.isShot:
            self.boss.hp -= 10
            self.isShot = False

            self.enemy.isDead = True

            if self.boss.hp <= 0:
                self.boss.isDead = True
```



# 횡스크롤 게임 만들기

## 5. Player.py

```
import pygame
```

```
class Player:
```

```
    def __init__(self, screen):  
        self.screen = screen
```

```
    self.player_img = [ pygame.image.load("resource/standing/play_atti_standing_0.png"),  
                        pygame.image.load("resource/standing/play_atti_standing_1.png"),  
                        pygame.image.load("resource/standing/play_atti_standing_2.png"),  
                        pygame.image.load("resource/standing/play_atti_standing_3.png") ]
```

```
    self.playerR_img = [ pygame.image.load("resource/run_right/play_atti_run_0.png"),  
                        pygame.image.load("resource/run_right/play_atti_run_1.png"),  
                        pygame.image.load("resource/run_right/play_atti_run_2.png"),  
                        pygame.image.load("resource/run_right/play_atti_run_3.png") ]
```

```
    self.attack_img = [pygame.image.load("resource/attack/play_atti_attack0_0.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_1.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_2.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_3.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_4.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_5.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_6.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_7.png"),  
                      pygame.image.load("resource/attack/play_atti_attack0_8.png")
```

```
    ]
```

```
    self.px = 250  
    self.py = 300  
    self.moveR= False  
    self.moveL= False  
    self.moveU=False  
    self.moveD=False
```

```
    self.std_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수  
    self.run_count = 0 #리스트 인덱스 따라가며 애니메이션 보여주기 위한 변수2
```

```
    self.attack = False  
    self.att_count = 0  
    self.speed= 2
```

```
    def draw(self):  
        if self.moveR :  
            self.screen.blit(self.playerR_img[self.run_count // 60] , (self.px, self.py))
```

```
        elif self.attack:  
            self.screen.blit(self.attack_img[self.att_count // 60] , (self.px, self.py))
```

```
        else:  
            self.screen.blit(self.player_img[self.std_count // 60] , (self.px, self.py))
```

```
    def update(self):  
        if self.moveR:  
            self.px += self.speed  
        if self.moveL:  
            self.px -= self.speed  
        if self.moveU:  
            self.py -= self.speed  
        if self.moveD:  
            self.py += self.speed
```

```
    def animation(self): ##추가한것
```

```
        self.std_count += 1 #리스트 인덱스를 증가시킨다.  
        self.run_count += 1  
        self.att_count += 1
```

```
        if self.std_count >=239 :  
            self.std_count = 0
```

```
        if self.run_count >=239 :  
            self.run_count = 0
```

```
        if self.att_count >=539 :  
            self.att_count = 0
```

# 횡스크롤 게임 만들기

## 6. MainGame.py

```
import missile

pygame.init()

screen = pygame.display.set_mode((1920, 1080))
pygame.display.set_caption("myFirst Game")

bg = background.Background(screen)
pl = player.Player(screen)
en = enemy.Enemy(screen) #이것도 추가
bo = boss.Boss(screen)
mi = missile.Missile(screen, bo, en)

while True:
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_q or event.key == pygame.K_ESCAPE:
                pygame.quit()
            if event.key == pygame.K_RIGHT:
                pl.moveR = True
            if event.key == pygame.K_LEFT:
                pl.moveL = True
            if event.key == pygame.K_UP:
                pl.moveU = True
            if event.key == pygame.K_DOWN:
                pl.moveD = True
            if event.key == pygame.K_a:
                pl.attack = True
                mi.missile_posU.append([pl.px, pl.py])
            if event.key == pygame.K_s:
                pl.attack = True
                mi.missile_pos.append([pl.px, pl.py])

        if event.type == pygame.KEYUP:
            if event.key == pygame.K_RIGHT:
                pl.moveR = False
            if event.key == pygame.K_LEFT:
                pl.moveL = False
            if event.key == pygame.K_UP:
                pl.moveU = False
            if event.key == pygame.K_DOWN:
                pl.moveD = False
            if event.key == pygame.K_a:
                pl.attack = False
            if event.key == pygame.K_s:
```

```
        bg.draw()
        bg.update()
        pl.draw()
        pl.update()
        pl.animation() #붙여주기
        en.draw()
        en.update()
        bo.draw()
        bo.update()
        bo.animation()
        mi.draw()
        mi.update()
        pygame.display.update()
```

2023 -2 강원중학교

Python 게임 만들기

감사합니다