

명령어 : Pip install pygame

Pygame은 SDL(Simple directMedia Layer) 기반의 게임 제작을 위해 사용되는 파이썬 확장 라이브러리 C언어로 구현된 크로스플랫폼 멀티미디어 라이브러리.

비디오, 오디오, 사용자 입력 등과 관련된 처리를 편하게 수행할수 있도록 해줌.

```
I:\python_example_student>pip install pygame
Collecting pygame
  Using cached https://files.pythonhosted.org/packages/b0/30/4df62b0eb0f09a8ad1c329ff5479b3b115e8dce25475fd0c2f73cb790aa4/pygame-1.9.4-cp37-cp37m-win32.whl
Installing collected packages: pygame
Successfully installed pygame-1.9.4
You are using pip version 10.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

설치 과정.

```
C:\Users\chameleon001>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pygame
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> import pygame
>>>
```

명령창에 **python**을 들어가서 **pygame** 을 **import**를 했을때 아무런 에러가 없으면 설치가 된걸 확인할수있다.

```
import pygame
import random
from time import sleep

# 게임에 사용되는 전역 변수 정의
BLACK = (0, 0, 0) # 게임 바탕화면의 색상값
RED = (255, 0, 0)
WHITE = (255, 255, 255)
pad_width = 480 # 게임 화면의 가로값
pad_height = 640 # 게임 화면의 세로값
fight_width = 36 # User 이미지 가로 크기
fight_height = 38 # User 이미지 세로 크기
enemy_width = 26 # 적 이미지 가로 크기
enemy_height = 20 # 적 이미지 세로 크기
```

```
def gameover():  
    global gamepad  
    dispMessage('Game Over')
```

적을 맞춘 개수 계산

```
def drawScore(count):  
    global gamepad  
  
    font = pygame.font.SysFont(None, 20)  
    text = font.render('Enemy Kills: ' + str(count), True, WHITE)  
    gamepad.blit(text, (0, 0))
```

필살기 개수 확인

```
def drawskill(count):
```

```
    global gamepad
```

```
    font = pygame.font.SysFont(None, 20)
```

```
    text = font.render('Skill: ' + str(count), True, WHITE)
```

```
    gamepad.blit(text, (180, 0))
```

```
def drawPassed(count):
```

```
    global gamepad
```

```
    font = pygame.font.SysFont(None, 20)
```

```
    text = font.render('Enemy Passed: ' + str(count), True, RED)
```

```
    gamepad.blit(text, (360, 0))
```

화면에 글씨 보이게 하기

```
def dispMessage(text):  
    global gamepad  
  
    textfont = pygame.font.Font('freesansbold.ttf', 80)  
    text = textfont.render(text, True, RED)  
    textpos = text.get_rect()  
    textpos.center = (pad_width/2, pad_height/2)  
    gamepad.blit(text, textpos)  
    pygame.display.update()  
    sleep(2)  
    runGame()
```

```
def crash():  
    global gamepad  
    dispMessage('Crashed!')
```

게임에 등장하는 객체를 그려줌

```
def drawObject(obj, x, y):  
    global gamepad  
    gamepad.blit(obj, (x,y))
```

게임 실행 메인 함수

def runGame():

 global gamepad, fighter, clock

 # gamepad : 게임이 진행될 게임 화면 전역변수

 # clock 게임의 초당 프레임 변수 pygame이 제공하는 clock 객체

 global bullet, enemy

 isShot = False

 shotcount = 0

 enemypassed = 0

 x = pad_width*0.45

 y = pad_height*0.9

 x_change = 0 # 전투기 좌우로 이동시키기 위한 변수

 bullet_xy = []

 enemy_x = random.randrange(0, pad_width-enemy_width) #적 생성 위치. (게임 화면 - 적크기)

 enemy_y = 0

 enemy_speed = 3 # 적 내려오는 속도 조절 변수


```
skill = 1
```

```
c = 0
```

```
ongame = False
```

Tap : 1 `while not ongame:`

```
    #키 이벤트 관련 if문
```

```
    #키 입력이 될때마다 설정된 코드가 실행된다.
```

Tap : 2 `for event in pygame.event.get():`

```
    if event.type == pygame.QUIT:
```

```
        ongame = True
```

```
    if event.type == pygame.KEYDOWN:
```

```
        if event.key == pygame.K_LEFT:
```

```
            x_change -= 5
```

```
        elif event.key == pygame.K_RIGHT:
```

```
            x_change += 5
```

```
    #왼쪽 컨트롤 키를 누르면 무기 발사.
```

```
    elif event.key == pygame.K_LCTRL:
```

```
        if len(bullet_xy) < 5: #오른쪽 숫자만큼 한 화면에 미사일이 존재할수있음
```

```
gamepad.fill(BLACK) |
```

Tap : 2

```
#전투기의 위치 재조정
```

```
x += x_change
```

Tap : 2

```
if x < 0:
```

```
    x = 0
```

```
elif x > pad_width - fight_width:
```

```
    x = pad_width - fight_width
```

Tap : 2

```
# 게이머 전투기가 적과 충돌했는지 체크
```

```
if y < enemy_y + enemy_height:
```

```
    if (enemy_x > x and enemy_x < x + fight_width) or \
```

```
        (enemy_x + enemy_width > x and enemy_x + enemy_width < x + fight_width)
```

```
        crash()
```

Tap : 2

```
drawObject(fighter, x, y)
```

전투기 무기 발사 구현

Tap : 2 `if len(bullet_xy) != 0:`

Tap : 3 `for i, bxy in enumerate(bullet_xy):`

Tap : 4 `bxy[1] -= 10`

`bullet_xy[i][1] = bxy[1]`

Tap : 4 `if bxy[1] < enemy_y:`

Tap : 5 `if bxy[0] > enemy_x and bxy[0] < enemy_x + enemy_width:`

`bullet_xy.remove(bxy)`

`isShot = True`

`shotcount += 1`

`if bxy[1] <= 0:`

`try:`

`bullet_xy.remove(bxy)`

`except:`

`pass`

Tap : 2 `if len(bullet_xy) != 0:`

`for bx, by in bullet_xy:`

`drawObject(bullet, bx, by)`

```
Tap : 2    drawScore(shotcount)
           drawskill(skill)

           # 적 아래로 움직임 구현

Tap : 2    enemy_y += enemy_speed

Tap : 2    if enemy_y > pad_height:
           |     enemy_x = random.randrange(0, pad_width-enemy_width)
           |     enemy_y = 0
           |     enemypassed += 1

           # 정해진 숫자만큼 적을 놓치면 게임이 끝난다.

Tap : 2    if enemypassed == 3:
           |     gameover()

           drawPassed(enemypassed)

           if isShot: #isShot은 발사한 무기가 적을 맞췄을경우 True로 설정된다.
           |     enemy_speed += 1 # 점점 내려오는 속도가 빨라진다.
           |     if enemy_speed >= 10:
           |         enemy_speed = 10
```


Tap : 3

```
enemy_x = random.randrange(0, pad_width-enemy_width)
enemy_y = 0
isShot = False
```

Tap : 2

```
drawObject(enemy, enemy_x, enemy_y)

pygame.display.update()
clock.tick(60) #프레임 설정 높을 수록 더 자연스럽다.
```

 pygame.quit()

Tap : 1

```
def initGame():
```

```
    global gamepad, fighter, clock
```

Tap : 1

```
    global bullet, enemy
```

```
    pygame.init()
```

```
    gamepad = pygame.display.set_mode((pad_width, pad_height))
```

```
    pygame.display.set_caption('One shot one opportunity') # 게임 타이틀 이름.
```

```
    fighter = pygame.image.load('fighter.png') # User 이미지
```

```
    enemy = pygame.image.load('enemy.png') # 적 이미지
```

```
    bullet = pygame.image.load('bullet.png') # 총알 이미지
```

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

```
initGame()
```

```
runGame()
```