파이썬 제안서

홍주안

# 1. 내가 사용할 모듈, 파이게임이란?

파이썬에는 다양한 라이브러리가 있어 파이썬이란 언어에 대한 응용도는 굉장히 우수한 편이고 방대한 양의 라이브러리 덕분에 파이썬이 지금의 자리에 있다고 해도 과언이 아니라고 한다. 내가 프로젝트를 만들면서 이번에 사용할 라이브러리는 바로 파이게임이라는 라이브러리, 모듈이다 파이게임이란 무엇일까?

SDL(Simple DirectMedia Layer) 기반 라이브러리라고 한다. 오픈 소스이자 무료이다! 파이썬을 돌릴 수 있는 플랫폼이라면 어디든지 실행할 수 있다고 한다.게임 개발도구이지만 이미지 프로세스, 입력기능등이 편해서 원하는 기능만 때다가 사 할 수 있다고 한다.

# 2. 나는 왜 파이게임을 골랐나?

내가 프로젝트를 만들기 위해서 파이썬 모듈들을 검색해보고 조사를 해보면서 기본적인 게임을 만들고 싶다는 생각이 들었다. 게임을 만드는 것은 알 사람은 알테지만 매우 어 렵다. 하지만 파이썬에 게임 만들기를 편하게 해준다는 라이브러리가 있다는 것을 알자 마자 꼭 파이게임을 나의 프로젝트에 사용하고 싶다는 생각이 들었고 그로 인해서 파이게임을 선택을 했고 게임을 만들게 되었다

# 3. 파이게임으로 어떠한 것을 할 수 있을까?

파이게임은 위에도 이야기 했듯이 다양한 도구들을 쉽게 사용할 수 있어서 굳이 게임을 만드는 것이 아니여도 이미지를 블러오는 프로세스를 만들거나, 조이스틱, 마우스, 키보드로부터 입력을 받아야 하는 경우에도 사용이 된다.

파이게임을 공부하고 또 사용해본 결과 RPG등 엄청 복잡한 게임을 만들기는 힘든 것 같다. 파이게임의 기능이 부족하기 보단 파이게임은 속도가 느리다. 밑에서 이야기를 할 것이지만 내가 만든 유성 피하기 게임은 주석처리된 줄을 합쳐서 150줄 정도 된다 하지만 시작하는데 처음 시작할 때 5초 정도 걸리는 것을 감안하여 보면 조금 복잡한 게임도 가능하지만, RPG등(게임 캐릭터를 키우는 게임)등이나 복잡한 상호작용이 있는 게임은 힘들어보인다.

# 4. 나는 파이게임으로 어떤 게임을 구현할 것인가?

내가 구현할 게임은 유성 피하기 게임이다. 유성 피하기 게임이라고 하면 감을 잘 못 잡으실 수도 있지만 비행기처럼 내가 조종하는 물체가 하늘을 날라다니며 위에서부터 떨어지는 유성을 피하는 그런 단순한 게임이다.

내 게임에서 나는 더 다양한 기능들을 넣고 싶었지만 파이게임이라는 라이브러리를 내가 원하는 그런 깊은 정도의 지식을 알려주는 곳을 찾기가 쉽지않아 그냥 파이게임 예제를 올려주는 곳에서 설명해주는 영상, 글을 보며 파이게임의 함수에 대해 공부 할 수 밖에 없었다.

# 5. 내가 배운 것들

먼저 내가 쓴 코드를 보자

import pygame

import random

###################################################################

###################################################################

def restart(): #게임이 끝나지 않는이상 게임을 계속 실행시킨다

    global isGameOver, score

    isGameOver = False

    score = 0

    for i in range(len(star)):

        recStar[i].y = -1

###################################################################

###################################################################

def eventProcess(): # 게임에 움직임을 담당한다. 키보드로 부터 정보를 받아드림

    for event in pygame.event.get():

        if event.type == pygame.KEYDOWN:

            if event.key == pygame.K\_ESCAPE:

                pygame.quit() # 부딪히면 끝

            if event.key == pygame.K\_SPACE:

                move.x = 0

                move.y = 0

            if event.key == pygame.K\_LEFT: # 키보드 좌클릭

                move.x = -1

                move.y = 0

            if event.key == pygame.K\_RIGHT:# 키보드 우클릭

                move.x = 1

                move.y = 0

            if event.key == pygame.K\_UP:   # 키보드 위클릭

                move.y = -1

                move.x = 0

            if event.key == pygame.K\_DOWN: # 키보드 밑클릭

                move.y = 1

                move.x = 0

            if event.key == pygame.K\_r:

                restart()

###################################################################

###################################################################

def movePlayer(): # 키보드로부터 받은 정보를 토대로 게임 캐릭터를 움직인다

    if not isGameOver:

        recPlayer.x += move.x # 입력받은 x좌표대로 계속해서 캐릭터를 움직임

        recPlayer.y += move.y # 입력받은 y좌표대로 계속해서 캐릭터를 움직임

    if recPlayer.x < 0: # 만약 왼쪽 벽과 만난다면 x좌표의 움직임을 멈춤

        recPlayer.x = 0

    if recPlayer.x > SCREEN\_WIDTH-recPlayer.width: # 만약 오른쪽 벽과 만난다면 x좌표의 움직임을 멈춤

        recPlayer.x = SCREEN\_WIDTH-recPlayer.width

    if recPlayer.y < 0: # 만약 아래쪽 벽과 만난다면 y 좌표를 멈춤

        recPlayer.y = 0

    if recPlayer.y > SCREEN\_HEIGHT-recPlayer.height: # 만약 위쪽 벽과 만난다면 y좌표를 멈춤 하지만 가능성 매우 희박함

        recPlayer.y = SCREEN\_HEIGHT-recPlayer.height

    SCREEN.blit(player, recPlayer)

###################################################################

###################################################################

def timeDelay500ms():

    global time\_delay\_500ms

    if time\_delay\_500ms > 5:

        time\_delay\_500ms = 0

        return True

    time\_delay\_500ms += 1

    return False

def makeStar(): ## 별을 만드는 함수

    if isGameOver: # 게임이 끝났다면 더이상 별을 만들지 않는다

        return

    if timeDelay500ms():

        idex = random.randint(0, len(star)-1)

        if recStar[idex].y == -1:

            recStar[idex].x = random.randint(0, SCREEN\_WIDTH)

            recStar[idex].y = 0

def moveStar(): # 별들을 움직여주는 함수

    makeStar() # 가장 먼저 별들을 생성한다

    for i in range(len(star)):

        if recStar[i].y == -1:  # 별들을 아래로 떨어트린다

            continue

        if not isGameOver:

            recStar[i].y += 1

        if recStar[i].y > SCREEN\_HEIGHT:

            recStar[i].y = 0

        SCREEN.blit(star[i], recStar[i])

###################################################################

###################################################################

def CheckCollision():   #캐릭터가 별에 충돌하는지 계속 확인해주는 함수

    global score, isGameOver

    if isGameOver: #충돌했다면 끝낸다

        return

    for rec in recStar:

        if rec.y == -1:

            continue

        if rec.top < recPlayer.bottom \

            and recPlayer.top < rec.bottom \

            and rec.left < recPlayer.right \

            and recPlayer.left < rec.right:

            print('충돌')

            isGameOver = True #게임을 끝낸다

            break

    score += 1

###################################################################

###################################################################

def blinking():

    global time\_dealy\_4sec, toggle

    time\_dealy\_4sec += 1

    if time\_dealy\_4sec > 40:

        time\_dealy\_4sec = 0

        toggle = ~toggle

    return toggle

def setText(): # 게임의 시작과 끝에 메세지를 출력해준다

    mFont = pygame.font.SysFont("arial",20, True, False)

    SCREEN.blit(mFont.render(

        f'score : {score}', True, 'green'), (10, 10, 0, 0))

    if isGameOver and blinking():

        SCREEN.blit(mFont.render(

            'Game Over!!', True, 'red'), (150, 300, 0, 0))

        SCREEN.blit(mFont.render(

            'press R - Restart', True, 'red'), (140, 320, 0, 0))

###################################################################

###################################################################

#1.변수초기화

isActive = True

SCREEN\_WIDTH = 600 # 게임판의 가로크기

SCREEN\_HEIGHT = 900 # 게임판의 세로크기

move = pygame.rect.Rect(0,0,0,0) # x,y와 음,양으로 움직임을 받아드린다

time\_delay\_500ms = 0

time\_dealy\_4sec = 0

toggle = False

score = 0 # 점수

isGameOver = False # True가 되면 게임을 끝낸다

#2.스크린생성

pygame.init() ###라이브러리를 사용하기위한 첫 동작### 중요하다고 한다

SCREEN = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT)) #스크린의 크기를 정하는 줄

pygame.display.set\_caption('홍주안의 달 피하기 게임') #제목 정하는 줄

#3. player 생성

player = pygame.image.load('player.png') #플레이어의 이미지를 로드한다.

player = pygame.transform.scale(player,(20,30)) # 플레이어의 크기

recPlayer = player.get\_rect() # 플레이어의 행동

recPlayer.centerx = (SCREEN\_WIDTH/2) #플레이어의 시작위치

recPlayer.centery = (SCREEN\_HEIGHT/2) # 플레이어의 시작위치

#4. 유성 생성

star = [pygame.image.load('star.png') for i in range(40)] # 유성의 개수를 정함

recStar = [None for i in range(len(star))]

for i in range(len(star)):

    star[i] = pygame.transform.scale(star[i], (20, 20))

    recStar[i] = star[i].get\_rect()

    recStar[i].y = -1 # 별들을 아래로 내린다

#5. 기타

clock = pygame.time.Clock()

#####반복####

while isActive:

    #1.흰 화면으로 채움

    SCREEN.fill((255,255,255)) # 255,255,255는 흰 색상의 숫자번호이다

    #2.이벤트처리

    eventProcess()

    #3.플레이어 이동

    movePlayer()

    #4.유성 생성 및 이동

    moveStar()

    #5.충돌 확인

    CheckCollision()

    #6.text업데이트

    setText()

    #7.화면 갱신

    pygame.display.flip() #스크린 여는것을 유지시켜줌

    clock.tick(100)

#####반복#####

##################

원래는 함수별로 정리하려고 했지만 생각보다 함수가 너무나도 많기에 그냥 코드에 주석 처리를 해서 함수에 대한 설명, 변수에 대한 설명을 적었다.

# 6. 이 게임의 한계

내가 만든 프로젝트의 한계를 따로 내가 적는 이유는 이 한계를 해결하지 못한게 아쉽기 도 하고 또 언젠간 이 한게를 극복할 수 있지 않을까 싶어 적는다

내가 이루지 못한 것은 다음과 같다.

.1 달의 내려오는 모양이 고정되어 있음. >> 함수상으로 위의 게임에 달이 떨어지는 방 설정된 화면 크기에 내가 넣은 별의 개수를 무작위로 흩뿌린다. 그 다음 그 사진을 -1씩 내리는 방법을 사용하는데 그 덕분에 한 별이 화면을 다 내려가면 그 별은 위에 똑같이 나오게 된다. 해결하고 싶지만 무슨 방법을 모르겠다ㅠㅠ

2. 로켓을 멈출 수가 없다. >> 내가 키보드를 한번 한 방향으로 누른다면 로켓은 그 방 향으로 벽을 만날 때 까지 계속 간다. 나도 키보드를 누르는동안만 가게 구현하고 싶다

제안서를 적다가 2번을 어떻게 해결할지 생각이나서 2번을 해결했다.

해결과정은 새로운 키 다운을 만든다 바로 SPACE바이다.

기존에 나의 로켓은 한번 누르면 그 방향으로 계속 가는 단점이 있었는데 SPACE바 다운 변수를 추가하면서 로켓이 움직이다가도 SPACE를 누르면 멈추게 만들었다

# 7. 후기

솔직히 프로젝트를 만들라는 이야기를 들었을 때 혼자서 프로젝트를 만들 수 있을까? 라는 생각이 가장 먼저 들었고 내가 파이썬에 이제 기초를 땠는데 내가 네이버에 나오는 걸 모방해서 프로젝트를 만드는 것이 의미가 있을까? 싶었지만 프로젝트를 복사해와서 분석하고 공부하면서 파이게임이라는 라이브러리 뿐만 아니라 파이썬 그 자체의 실력도 올랐다는 느낌을 받아서 굉장히 좋은 느낌으로 프로젝트를 마무리 했다.

감사합니다