

# 코드 설명

```
import pygame
import math #math 함수를 이용해 마우스에 따라 움직이는 우주선이나 얼굴 미사일 등을 표현하였다.
import random #말 그대로 랜덤한 수를 만들기 위해 사용
from tkinter import * # Tkinter 사용

pygame.init()

#행성 범위 설정 (131,88),(410,138) <-- 클릭 시 행성 변경.

#play 범위 (131, 426), (353, 483)

#quit 범위 (457, 432), (631, 477)

size = [800, 600] #pygame의 창 크기 표현
screen = pygame.display.set_mode(size) #창 크기를 정의해 주었다.
angle=None #각도를 저장할 함수 정의

#이미지 로드 시작
logo=pygame.image.load('image/logo.png') #이미지로드를 통해 이미지 추가
play=[pygame.image.load('image/play.png'),pygame.image.load('image/play2.png')]
quitt=[pygame.image.load('image/quit.png'),pygame.image.load('image/quit2.png')]

cursor=[pygame.image.load('image/cursor/cursor1.png'),pygame.image.load('image/cursor/cursor2.png'),pygame
image.load('image/cursor/cursor3.png')]
pygame.display.set_caption("Planet Save")

ship=[pygame.image.load('image/ship/ship'+str(random.randrange(0,3)+1)+'_1.png'),pygame.image.load('image/s
hip/ship'+str(random.randrange(0,3)+1)+'_1.png')]
shot=pygame.image.load('image/shot.png')
planet=[]
for i in range(4) :
    planet.append(pygame.image.load('image/planet/planet'+str(i+1)+'_1.png'))
eyes=[]
for i in range(4) :
    eyes.append(pygame.image.load('image/eyes/eyes'+str(i+1)+'_1.png'))

bg=pygame.image.load('image/background.png')
shadow=pygame.image.load('image/planet/shadow.png')
```

```
healthbar=pygame.image.load('image/health/healthbar.png')
healthbarin=[]
for i in range(5) :
    healthbarin.append(pygame.image.load('image/health/health_bar'+str(i)+'.png'))
```

#이미지 로드 끝

# 사운드

```
crush_sound=pygame.mixer.Sound('sound/crush.wav') #pygame안에 있는 Sound를 사용
hits=pygame.mixer.Sound('sound/hits.wav')
pygame.mixer.music.load('sound/background.mp3')
gameover=pygame.mixer.Sound('sound/gameover.wav')
#pygame.mixer.music.load('sound/background2.wav')
pygame.mixer.music.play(-1) #-1무한 재생
```

#사운드 끝

```
screen.blit(bg, (0,0)) #백그라운드 이미지를 화면에 표시
centerX=size[0]/2 #가운데 X 좌표를 정의
centerY=size[1]/2 #가운데 Y 좌표를 정의
```

```
pygame.mouse.set_visible(0) #기본 마우스 숨김 이유 : 다른 이미지를 쓰고 싶어서
```

#필요한 변수 정의 시작

```
done = False
clock = pygame.time.Clock()
FPS = 60
click = 0
planet_speed = 0.0
shot_speed = 85
ship_state=0
position=[0,0]
```

```
play_state=0
quit_state=0
eyes_state=0
start_state=0
check=0
```

```
moving=25
```

```
planet_char=random.randrange(0,4)
shot_state=[]
```

```
star_angle=[0.0,0.0,0.0,0.0]
t=0
q=0
area=5
st=0
```

```
ti=0
time=0
checktime=0
shot_sw=0
level=0
```

```
HP=4
```

```
score=[]
score_name=[]
```

#필요한 변수 정의 끝

```
myfont=pygame.font.Font('avant_pixel.ttf',25) #글씨 폰트 정의
```

#운석 클래스 운석은 여러개가 필요하기 때문에 클래스로 지정하였다.

```
class starClass:
```

```
    hap=0 #운석 거리에 대한 합
```

```
    rotate=None #운석 회전에 대한 이미지 변수
```

```
    def __init__(self,num): #운석이 어떤 이미지가 될지 클래스 정의할 때 수를 랜덤하게 받아 결정
```

```
        self.image =
```

```
[pygame.image.load('image/star/star'+str(num+1)+'_1.png'),pygame.image.load('image/star/star'+str(num+1)+'_
2.png'),pygame.image.load('image/star/star'+str(num+1)+'_3.png')]
```

```
    self.angle=random.randrange(1,36)+random.random() #운석의 각도
```

```
    self.crush=random.randrange(0,3) #운석 충돌 여부
```

```
    self.speed=random.random()+0.1 #운석 스피드
```

```
    self.distance=random.randrange(450,5000) #운석 위치
```

```
    self.t=random.randrange(2,6) #운석 회전 속도
```

```
cometru=[] #운석 클래스가 들어갈 변수
```

#스크린 위치 함수 설명 : 이미지가 화면 중앙에 출력되도록 하는 함수

```
def pos(img_position):
```

```
    position=img_position.get_size()
```

```
    return centerX-position[0]/2,centerY-position[1]/2
```

#앵글 방향으로 회전 함수 설명 : math함수를 이용해서 이미지 회전할 때 변하는 이미지 크기대로 좌표가 변경하게 하는 함수

```
def vision(shape,angle,radius):
```

```

    return
    centerX-(shape.get_size()[0]/2)+(math.cos(angle)*radius),centerY-(shape.get_size()[1]/2)+(math.sin(angle)*radius)
)

```

#### #운석 회전 함수

```

def star_rotate(shape):
    global t
    return pygame.transform.rotate(shape,t)

```

#### #충돌 판정 함수

```

def crush(shape1,shape2,star):
    global area
    if (shape1[0]-area<shape2[0] and shape2[0]<shape1[0]+(star[0]-area)) and
(shape1[1]-area<shape2[1] and shape2[1]<shape1[1]+(star[1]-area)) :
        return 1
    return 0

```

#### #시간 함수 time을 입력받아 HH:MM:SS 포맷으로 반환해주는 함수

```

def timegender(time):
    HH,MM,SS=0,0,0
    HH=time//3600
    if HH!=0: time=time%3600
    MM=time//60
    if MM!=0: time=time%60
    SS=time
    if HH<10:HH='0'+str(HH)
    else: HH=str(HH)
    if MM<10:MM='0'+str(MM)
    else: MM=str(MM)
    if SS<10:SS='0'+str(SS)
    else: SS=str(SS)
    return HH+":"+MM+":"+SS

```

#### #랭크 화면

```

def rank(score,score_name):
    global cursor,click,clock,FPS,position,myfont
    bg=pygame.image.load('image/rankbackground.png')
    size = [400, 600]
    done=0
    screen2 = pygame.display.set_mode(size)
    ranktxt=myfont.render(u'RANK',True,(255,255,255))
    ranking=[ ]
    tmp=0;
    text=None

```

```

print(len(score))
if(len(score)>1):
    lent=len(score)
    for i in (0,lent-2):
        for j in (0,(lent-i)-2):
            if score[j]<score[j+1] :
                tmp=score[j]
                score[j]=score[j+1]
                score[j+1]=tmp
                text=score_name[j]
                score_name[j]=score_name[j+1]
                score_name[j+1]=text
j=len(score)
if len(score)>5:
    j=5
#print(score)
while not done:
    clock.tick(FPS)
    screen2.blit(bg, (0,0))
    screen2.blit(ranktxt, (200-(ranktxt.get_size()[0]/2),75))
    #score.sort(reverse=True)
    for i in range(0,j):
        ranking.append(myfont.render(u''+str(i+1)+' Rank : '+timegender(score[i])+
"+score_name[i],True,(255,255,255)))
        screen2.blit(ranking[i], (200-(ranking[i].get_size()[0]/2),200+(i*50)))
#마우스 모양
if click == 0 :
    screen2.blit(cursor[0],(position[0],position[1]))
elif click == 1:
    screen2.blit(cursor[1],(position[0],position[1]))

#마우스 모션
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        done = True
    if event.type == pygame.MOUSEMOTION:
        position=pygame.mouse.get_pos()
    if event.type == pygame.MOUSEBUTTONDOWN:
        if event.button==1:
            click=1
        elif event.type == pygame.MOUSEBUTTONUP:
            click=0
pygame.display.flip()

```

#랭크 이름 입력 Tkinter

```

def rankname():
    def close():
        global score_name
        score_name.append(txt.get())
        root.quit()
        root.destroy()
    root = Tk()

    root.geometry('250x100')
    root.title("랭크 이름을 입력하세요.")
    root.resizable(True, True)

    label01 = Label(root, text="랭크 이름 : ")
    label01.pack()
    txt=Entry(root)
    txt.pack()

    quit01 = Button(root, text="확인", command=close, bg='white', fg='blue')
    quit01.pack()
    root.mainloop()

```

#글씨 출력 시작

```

starttxt=myfont.render(u'Game Start! 5',True,(255,255,255))
txt=myfont.render(u'time : '+timegenger(time),True,(255,255,255))
cometruTxt=myfont.render(u'Remaining meteorite : ',True,(255,255,255))
#stark1=pygame.transform.rotate(star[0][0],t)

```

#글씨 출력 끝

#게임 시작

```

while not done:
    clock.tick(FPS)
    #배경, 행성, 게임시작,나가기 초기화
    screen.blit(bg, (0,0))
    if(start_state==0):
        if HP==0:
            if moving>=0:
                screen.blit(logo,(pos(logo)[0]+moving,pos(logo)[1]-185))

    screen.blit(play[play_state],(pos(play[play_state])[0]-150-moving,pos(play[play_state])[1]+160))

    screen.blit(quitt[quit_state],(pos(quitt[quit_state])[0]+150+moving,pos(quitt[quit_state])[1]+160))
        moving=moving-25
        if moving== -25:
            HP=4
            pygame.time.wait(100)
            rankname() #랭크 이름 입력 함수 호출

```

```

rank(score,score_name) #랭크 출력 함수
screen = pygame.display.set_mode(size)
else:
    screen.blit(logo,(pos(logo)[0],pos(logo)[1]-185))
    screen.blit(play[play_state],(pos(play[play_state])[0]-150,pos(play[play_state])[1]+160))
    screen.blit(quit[quit_state],(pos(quit[quit_state])[0]+150,pos(quit[quit_state])[1]+160))
elif(moving<600):
    screen.blit(logo,(pos(logo)[0]+moving,pos(logo)[1]-185))
    screen.blit(play[play_state],(pos(play[play_state])[0]-150-moving,pos(play[play_state])[1]+160))
    screen.blit(quit[quit_state],(pos(quit[quit_state])[0]+150+moving,pos(quit[quit_state])[1]+160))
    moving=moving+25

```

```

planet_speed=(planet_speed+0.3)%360
planet2=pygame.transform.rotozoom(planet[planet_char],planet_speed,1)
screen.blit(planet2,pos(planet2))

```

```

screen.blit(shadow,(pos(shadow)[0],pos(shadow)[1]+30))
# 우주선 앵글 변환
angle=math.atan2(position[1]-centerY,position[0]-centerX)
angle2 = angle * (180/math.pi) #호도법을 육십분법으로 바꿔줌
angle2 = (angle2 + 90) % 360

```

```

ship2=pygame.transform.rotate(ship[ship_state],-angle2)
ship_state=0

```

#-(ship2.get\_size()[0]/2) 하는 이유는 rotate를 할때 이미지 좌표가 달라진다 그러므로 바뀔때마다 이미지 크기를 불러와서 다시 빼줘야한다

```

x2,y2=vision(ship2,angle,85) #85는 반지름 길이이다.
screen.blit(ship2,(x2,y2))
#우주선 끝

```

```

#눈 표시
screen.blit(eyes[eyes_state],vision(eyes[eyes_state],angle,20))

```

```

#star_angle=math.atan2(,)

```

#시간초 표시 게임시작

```

if moving>=600:
    #HP가 0이 되었다.
    if HP!=0:
        screen.blit(healthbar,(10,10))
        screen.blit(healthbarin[HP],(10,10))
        ti=ti+1
        if ti==FPS :

```

```

ti=ti%FPS
time=time+1
if time<=5 and checktime==0:
    starttxt=myfont.render(u'Game Start! '+str(5-time),True,(255,255,255))
elif checktime==0 :
    time=0
    checktime=1
if checktime==1:
    txt=myfont.render(u'time : '+timegender(time),True,(255,255,255))
#시간에 따라 운석 속도를 더 빨리 바꾼다
if time==30 :
    level=0.5
if time==60:
    level=1
if time==90:
    level=1.5
if time==120:
    level=2
#시간에 따라 운석 속도 끝
#운석 화면 출력
if checktime==1 :
    if len(cometru)==0 :
        HP=0
        score.append(time)
        checktime=0
        ti=0
        time=0
        start_state=0
    for i in range(0,len(cometru)):
        cometru[i].hap=(cometru[i].hap+cometru[i].t)%360
        cometru[i].distance=cometru[i].distance-(cometru[i].speed+level)

cometru[i].ratate=pygame.transform.rotate(cometru[i].image[cometru[i].crush],cometru[i].hap)
#행성에 운석이 박히면 운석을 지운다.
if
crush((centerX-(planet2.get_size()[0]/2),centerY-(planet2.get_size()[1]/2)),vision(cometru[i].ratate,cometru[i].angle,cometru[i].distance),planet[planet_char].get_size()):
    pygame.mixer.Sound.play(crush_sound)
    cometru.remove(cometru[i])
    eyes_state=3
    HP=HP-1
    screen.blit(healthbar,(10,10))
    pygame.mixer.Sound.play(gameover)
    if HP==0:
        del cometru[:]

```



```

        score.append(time)
        checktime=0
        ti=0
        time=0
        start_state=0
        break
    screen.blit(cometru[i].ratate,vision(cometru[i].ratate,cometru[i].angle,cometru[i].distance))
    #텍스트 출력 0일때는 54321 출력하는거고 1일때는 시간초 표시
    if checktime==0: screen.blit(starttxt,(296,95))
    if checktime==1:
        screen.blit(txt,(296,95))
        cometruTxt=myfont.render(u'Remaining meteorite : '+str(len(cometru)),True,(255,255,255))
        screen.blit(cometruTxt,(30,550))

```

#마우스 클릭 할 때와 시작할 때 모양 변경해서 출력

```

if click == 0 and start_state==0:
    screen.blit(cursor[0],(position[0],position[1]))
elif click == 1 and start_state==0:
    screen.blit(cursor[1],(position[0],position[1]))

if start_state==1:
    screen.blit(cursor[2],(position[0]-cursor[2].get_size()[0]/2,position[1]-cursor[2].get_size()[1]/2))

```

#마우스 클릭 끝

#슈팅 게임때 쏘꺼. 만약 총알이 범위 밖으로 나가면 총알을 지운다.

```

for i in range(0,len(shot_state)) :
    x,y=vision(shot_state[i][0],shot_state[i][1],shot_state[i][2])
    shot_state[i][2]=shot_state[i][2]+10
    screen.blit(shot_state[i][0],(x,y))
    if checktime==1 :
        #미사일과 운석이 충돌할 때 운석을 지워주는 부분
        for j in range(0,len(cometru)):
            if moving>=600 and
crush(vision(cometru[j].ratate,cometru[j].angle,cometru[j].distance),(x,y),cometru[j].ratate.get_rect().size):
            cometru[j].crush=cometru[j].crush+1
            pygame.mixer.Sound.play(crush_sound)
            if cometru[j].crush==3:
                cometru.remove(cometru[j])
            #print("충돌!" +str(st))
            shot_sw=1
            break

```

```

if shot_sw==1:
    shot_state.remove(shot_state[i])
    shot_sw=0
    break
if (x<0 or x>820) or (y<0 or y>620) :
    shot_state.remove(shot_state[i])
    break

```

### #마우스 이벤트 발생

```

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        done = True
    if event.type == pygame.MOUSEMOTION:
        position=pygame.mouse.get_pos()
        #eyes_state=0
        #게임시작 나가기 얼굴 표정 변환
        if(start_state==0):
            if (131<position[0] and position[0]<353) and (426<position[1] and position[1]<483):
                play_state=1
                eyes_state=1
            else:
                play_state=0
            if (457<position[0] and position[0]<631) and (432<position[1] and position[1]<477):
                quit_state=1
                eyes_state=2
            else:
                quit_state=0
        #print(position[0],position[1],angle)
    if event.type == pygame.MOUSEBUTTONDOWN:
        if event.button==1:
            eyes_state=0
            pygame.mixer.Sound.play(hits)
            if(start_state==0):
                #행성 변경 영역
                if (132<position[0] and position[0]<412) and (83<position[1] and position[1]<135):
                    planet_char=(planet_char+1)%4
                #게임시작 영역
                if (131<position[0] and position[0]<353) and (426<position[1] and position[1]<483):
                    start_state=1
                    moving=25
                    for i in range(0,100):
                        cometru.append(starClass(random.randrange(0,4)))
                    starttxt=myfont.render(u'Game Start! 5',True,(255,255,255))
                    level=0

```

**#나가기 클릭 영역**

if (457<position[0] and position[0]<631) and (432<position[1] and position[1]<477):

done = True

#print(pygame.mouse.get\_pos(),eyes\_state)

click=1

ship\_state=1

shot\_state.append([pygame.transform.rotate(shot,-angle2),angle,shot\_speed])

elif event.type == pygame.MOUSEBUTTONUP:

click=0

pygame.display.flip()

pygame.quit()

# 후기

오픈소스 하나 없이 게임을 만들었기 때문에 힘든 부분이 좀 많았습니다.

그렇지만 게임을 만들면서 코사인과 사인 아크 탄젠트 등 수학적인 부분을 좀 더 많이 알 수 있는 계기가 되었습니다.

이미지 하나하나와 음향, 효과음 등 많은 부분이 얼마나 필연적인 존재인지 다시 한번 알 수 있게 되었고 혼자 게임을 만들게 되니까 이 많은 작업을 왜 팀별로 나눠서 하는지 알 수 있는 시간이었습니다.

수학적인 부분과 함수 활용 등 부족한 부분이 너무 많았는데 게임을 만들면서 조금 더 다듬는 법을 배웠습니다.

객체 지향적인 사고를 키울 수 있었던 작업이었고 클래스가 얼마나 편한지 같지만 다른 객체들을 이렇게 편리하게 만들 수 있는지 알게 되었습니다.

생각했던 것보다 좋은 퀄리티의 결과물이 나와 만족스럽고 아쉬운 부분이 참 많지만, 시간상 부족한 부분이 많고 코드상 불필요한 부분이 있지만, 시간이 된다면 다른 기능과 새로운 동작들을 추가해보고 싶습니다.