

Portfolio St.

Python을 이용한 슈팅게임 제작

[조장 : 이재욱 조원 : 김호준, 박기범, 박대희, 정운학]

목차

a table of contents

- 1 초기코드작성 및 오류검출
- 2 Game 기획
- 3 Game Code 추가
- 4 Image 다운 및 적용
- 5 Q&A

Part 1

초기코드작성 및 오류검출

초기코드작성

과제를 시작하며 강사님에게 받은 문서로 만들어질 게임의 기본 틀이 될 코드를 작성합니다.

코드를 작성하며 해석이 되지 않는 코드를 Chat GPT를 이용하여 해석합니다

```
def writeMessage(text):  
    global gamePad, gameOverSound  
    textfont = pygame.font.Font('NanumGothic.ttf', 40)  
    text = textfont.render(text, True, (255, 0, 0))  
    textpos = text.get_rect()  
    textpos.center = (padWidth / 2, padHeight / 2)  
    gamePad.blit(text, textpos)  
    pygame.display.update()  
    sleep(2) 한국어로설명해줘
```

5. 텍스트를 화면에 그리기:

python

코드 복사

```
gamePad.blit(text, textpos)  
pygame.display.update()
```

- ``gamePad.blit(text, textpos)``를 통해 텍스트를 ``gamePad``라는 화면 객체에 그립니다.
``pygame.display.update()``는 화면에 변경 사항을 즉시 반영하도록 업데이트합니다.

6. 일시 정지:

python

코드 복사

```
sleep(2)
```

- ``sleep(2)``는 파이썬의 ``time`` 모듈에서 제공하는 함수로, 2초 동안 실행을 멈추게 합니다.
이렇게 하면 텍스트가 화면에 2초 동안 표시됩니다.

ALTools

초기코드작성

```

1 import pygame
2 import sys
3 import random
4 from time import sleep
5
6 pygame.init() # pygame 초기화
7
8 # 색상 정의
9 BLACK = (0, 0, 0)
10 # 게임 화면 크기
11 paddlth = 425
12 padheight = 640
13 # 이미지 및 사운드 파일 목록
14 rockImage = [
15     'rock01.png', 'rock02.png', 'rock03.png', 'rock04.png', 'rock05.png',
16     'rock06.png', 'rock07.png', 'rock08.png', 'rock09.png', 'rock10.png',
17     'rock11.png', 'rock12.png', 'rock13.png', 'rock14.png', 'rock15.png',
18     'rock16.png', 'rock17.png', 'rock18.png', 'rock19.png', 'rock20.png',
19     'rock21.png', 'rock22.png', 'rock23.png', 'rock24.png', 'rock25.png',
20     'rock26.png', 'rock27.png', 'rock28.png', 'rock29.png', 'rock30.png'
21 ]
22 explosionSound = [
23     'explosion01.wav', 'explosion02.wav', 'explosion03.wav', 'explosion04.wav'
24 ]
25
26 # 게임 객체를 그리는 함수
27 def drawObject(obj, x, y):
28     global gamePad
29     gamePad.blit(obj, (x, y))
30
31 # 게임 초기화
32 def initGame():
33     global gamePad, clock, background, fighter, missile, explosion, missileSound, gameOverSound
34     pygame.init()
35     gamePad = pygame.display.set_mode((paddlth, padheight))
36     pygame.display.set_caption('PyShooting')
37     background = pygame.image.load('D:\python-workspace/background.png') # 파일 확장자 포함
38     fighter = pygame.image.load('D:\python-workspace/fighter.png')
39     missile = pygame.image.load('D:\python-workspace/missile.png')
40     explosion = pygame.image.load('D:\python-workspace/explosion.png')
41     #pygame.mixer.music.load('music.wav')
42     #3pygame.mixer.music.play(-1)
43     #4missileSound = pygame.mixer.Sound('missile.wav')
44     #5gameOverSound = pygame.mixer.Sound('gameover.wav')
45     clock = pygame.time.Clock()
46
47 # 게임 실행 함수
48 def runGame():
49     global gamePad, clock, background, fighter, missile, explosion, missileSound
50     fighterSize = fighter.get_rect().size
51     fighterWidth = fighterSize[0]
52     fighterHeight = fighterSize[1]
53     x = paddlth * 0.45
54     y = padheight * 0.9
55     fighterX = 0
56     missileXY = []
57     rock = pygame.image.load(random.choice(rockImage))
58     rockSize = rock.get_rect().size
59     rockWidth = rockSize[0]
60     rockHeight = rockSize[1]
61     #destroySound = pygame.mixer.Sound(random.choice(explosionSound))
62     rockX = random.randrange(0, paddlth - rockWidth)
63     rockY = 0
64     rockSpeed = 2
65     isShot = False
66     shotCount = 0
67     rockPassed = 0
68     onGame = True
69
70 while onGame:
71     for event in pygame.event.get():
72         if event.type == pygame.QUIT:
73             pygame.quit()
74             sys.exit()
75         elif event.type == pygame.KEYDOWN:
76             if event.key == pygame.K_LEFT:
77                 fighterX -= 5
78             elif event.key == pygame.K_RIGHT:
79                 fighterX += 5
80             elif event.key == pygame.K_SPACE:
81                 #destroySound.play()
82                 missileX = x + fighterWidth / 2
83                 missileY = y - fighterHeight
84                 missileXY.append([missileX, missileY])
85             elif event.type == pygame.KEYUP:
86                 if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
87                     fighterX = 0
88
89 drawObject(background, 0, 0)
90 x += fighterX
91 if x < 0:

```

```

92 if x < 0:
93     x = 0
94 elif x > paddlth - fighterWidth:
95     x = paddlth - fighterWidth
96 if y < rockY + rockHeight:
97     if (rockX > x and rockX < x + fighterWidth) or \
98         (rockX + rockWidth > x and rockX + rockWidth < x + fighterWidth):
99         crash()
100 drawObject(fighter, x, y)
101 if len(missileXY) != 0:
102     for i, by in enumerate(missileXY):
103         by[1] -= 10
104         missileXY[i][1] = by[1]
105         if by[1] <= 0:
106             try:
107                 missileXY.remove(by)
108             except:
109                 pass
110         elif by[1] < rockY:
111             if by[0] > rockX and by[0] < rockX + rockWidth:
112                 missileXY.remove(by)
113                 isShot = True
114                 shotCount += 1
115 if len(missileXY) != 0:
116     for bx, by in missileXY:
117         drawObject(missile, bx, by)
118 if isShot:
119     drawObject(explosion, rockX, rockY)
120     #destroySound.play()
121     rock = pygame.image.load(random.choice(rockImage))
122     rockSize = rock.get_rect().size
123     rockWidth = rockSize[0]
124     rockHeight = rockSize[1]
125     rockX = random.randrange(0, paddlth - rockWidth)
126     rockY = 0
127     #destroySound = pygame.mixer.Sound(random.choice(explosionSound))
128     isShot = False
129     rockSpeed += 0.02
130     if rockSpeed >= 10:
131         rockSpeed = 10
132     rockY += rockSpeed
133 if rockY > padheight:
134     rock = pygame.image.load(random.choice(rockImage))
135     rockSize = rock.get_rect().size
136     rockWidth = rockSize[0]
137     rockHeight = rockSize[1]
138     rockX = random.randrange(0, paddlth - rockWidth)
139     rockY = 0
140     rockPassed += 1
141 if rockPassed == 3:
142     gameOver()
143 drawObject(rock, rockX, rockY)
144 writeScore(shotCount)
145 writePassed(rockPassed)
146 pygame.display.update()
147 clock.tick(60)
148
149 # 폰트 및 글자 크기
150 def writeScore(count):
151     global gamePad
152     font = pygame.font.Font('NanumGothic.ttf', 20)
153     text = font.render('파괴한 몬스터 수: ' + str(count), True, (255, 255, 255))
154     gamePad.blit(text, (10, 0))
155
156 # 폰트를 통한 개수 표시
157 def writePassed(count):
158     global gamePad
159     font = pygame.font.Font('NanumGothic.ttf', 20)
160     text = font.render('통과한 몬스터: ' + str(count), True, (255, 0, 0))
161     gamePad.blit(text, (310, 0))
162
163 # 게임 메시지 출력
164 def writeMessage(text):
165     global gamePad, gameOverSound
166     textfont = pygame.font.Font('NanumGothic.ttf', 40)
167     text = textfont.render(text, True, (255, 0, 0))
168     textpos = text.get_rect()
169     textpos.center = (paddlth / 2, padheight / 2)
170     gamePad.blit(text, textpos)
171     pygame.display.update()
172     #pygame.mixer.music.stop()
173     #destroySound.play()
174     sleep(2)
175     #pygame.mixer.music.play(-1)
176
177 # 전투기가 파괴와 충돌했을 때
178 def crash():
179     writeMessage('전투기 파괴!!')
180     runGame()
181
182 # 게임 오버 메시지 및 종료
183 ShotCount = 0 # 전역 변수로 선언
184
185 def gameOver():
186     writeMessage('게임 오버!!')
187     writeScore(ShotCount) # 전역 변수 사용
188     pygame.quit()
189     sys.exit()
190
191 # 게임 종료 ShotCount 값을 수정하는 코드
192 ShotCount += 1 # 메시지: 값이 발생할 때 ShotCount를 증가시킴
193
194 # 게임 초기화 및 실행
195 initGame()
196 runGame()

```

오류검출

```
rockImage = [
    'rock01.png', 'rock02.png', 'rock03.png', 'rock04.png', 'rock05.png',
    'rock06.png', 'rock07.png', 'rock08.png', 'rock09.png', 'rock10.png',
    'rock11.png', 'rock12.png', 'rock13.png', 'rock14.png', 'rock15.png',
    'rock16.png', 'rock17.png', 'rock18.png', 'rock19.png', 'rock20.png',
    'rock21.png', 'rock22.png', 'rock23.png', 'rock24.png', 'rock25.png',
    'rock26.png', 'rock27.png', 'rock28.png', 'rock29.png', 'rock30.png'
]

background = pygame.image.load('D:\python-workspace/background.png') # 파일 확장자 포함
fighter = pygame.image.load('D:\python-workspace/fighter.png')
missile = pygame.image.load('D:\python-workspace/missile.png')
explosion = pygame.image.load('D:\python-workspace/explosion.png')
```

이미지오류: 코드에 설정된 이미지 파일이 지정된 위치에 없어 발생한다. 이미지파일을 확장자에 맞춰 경로를 설정하여 위치에 두거나 경로를 지정하지 않고 코드 파일 위치에 두면 해결된다.

```
def writeScore(count):
    global gamePad
    font = pygame.font.Font('None', 20)
    text = font.render('파괴한 운석 수: ' + str(count), True, (255, 255, 255))
    gamePad.blit(text, (10, 0))
# 운석을 놓친 개수 표시
def writePassed(count):
    global gamePad
    font = pygame.font.Font('NanumGothic.ttf', 20)
    text = font.render('놓친 운석 : ' + str(count), True, (255, 0, 0))
    gamePad.blit(text, (310, 0))
# 게임 메시지 출력
def writeMessage(text):
    global gamePad, gameOverSound
    textfont = pygame.font.Font('NanumGothic.ttf', 40)
    text = textfont.render(text, True, (255, 0, 0))
    textpos = text.get_rect()
    textpos.center = (padWidth / 2, padHeight / 2)
    gamePad.blit(text, textpos)
    pygame.display.update()
```

폰트오류: 폰트도 마찬가지로 폰트를 다운받아 확장자명을 일치시켜 경로를 지정하거나 코드파일 위치에 저장하면 해결된다.

```
#2pygame.mixer.music.load('music.wav')
#3pygame.mixer.music.play(-1)
#4missileSound = pygame.mixer.Sound('missile.wav')
#5gameOverSound = pygame.mixer.Sound('gameover.wav')

#11destroySound.play()
rock = pygame.image.load(random.choice(rockImage))
rockSize = rock.get_rect().size
rockWidth = rockSize[0]
rockHeight = rockSize[1]
rockX = random.randrange(0, padWidth - rockWidth)
rockY = 0
#8destroySound = pygame.mixer.Sound(random.choice(explosionSound))

#9pygame.mixer.music.stop()
#10gameOverSound.play()
sleep(2)
#11pygame.mixer.music.play(-1)
```

오디오오류: 학원 PC에 오디오 출력 장치가 없어 오디오 출력 코드를 주석처리 하였다.

Part 2

Game 기획

필살기

무기성장

보스

인간형 적군

아이템

운석체력

목숨

기획

어떤 기능을 넣어서 만들지 어떤 배경으로 할지 등 간단한 토의를 진행하였다

Part 3

Game코드추가

운석의 체력생성 및 동작

운석의 체력을 설정하고
생성된 운석의 체력에 맞게 동작하도록 만든다.

```
class Rock:
    def __init__(self, image, x, y, speed):
        self.image = pygame.image.load(image)
        self.rect = self.image.get_rect(topleft=(x, y))
        self.speed = speed
        self.health = random.randint(1, 3) # 체력을 랜덤으로 설정 (1~3)

    def draw(self):
        gamePad.blit(self.image, self.rect.topleft)

    def move(self):
        self.rect.y += self.speed

    def is_off_screen(self):
        return self.rect.y > padHeight

    def hit(self):
        self.health -= 1
        if self.health <= 0:
            return True # 체력이 0 이하로 떨어지면 True 반환
        return False # 그렇지 않으면 False 반환
```

Game코드추가

운석파괴시 무기강화

운석을 30개 처리할 때마다
발사되는 미사일 수가 한발씩 늘어나며
총 3개까지 늘어나도록 설정하였다.

```
if len(missileXY) != 0:
    for i, bxy in enumerate(missileXY):
        bxy[1] -= 10
        missileXY[i][1] = bxy[1]
        if bxy[1] <= 0:
            try:
                missileXY.remove(bxy)
            except:
                pass
        elif bxy[1] < rock.rect.y:
            if bxy[0] > rock.rect.x and bxy[0] < rock.rect.x + rock.image.get_width():
                if rock.hit(): # 운석이 파괴되었는지 확인
                    isShot = True
                    shotCount += 1
                    # 30개를 파괴한 경우 미사일 두 배로 발사
                    if shotCount % 30 == 0:
                        missile_upgrade += 1
                        if missile_upgrade > 3: # 최대 3단계로 제한
                            missile_upgrade = 3
                    # 새로운 운석 생성
                    rock = Rock(random.choice(rockImage), random.randrange(0, padWidth - 64), 0, 2)
            missileXY.remove(bxy)
```

Game코드추가

```
# 충돌 감지 및 목숨 감소
if y < rock.rect.y + rock.image.get_height():
    if (rock.rect.x > x and rock.rect.x < x + fighterWidth) or \
        (rock.rect.x + rock.image.get_width() > x and rock.rect.x + rock.image.get_width() < x + fighterWidth):
        lives -= 1 # 목숨 감소
        if lives <= 0:
            gameOver() # 목숨이 0이 되면 게임 오버
        else:
            # 충돌 후 새로운 운석 생성
            rock = Rock(random.choice(rockImage), random.randrange(0, padWidth - 64), 0, 2)
```

```
def initgame():
    global gamePad, clock, background, fighter, missile, explosion, lives
    pygame.init()
    gamePad = pygame.display.set_mode((padWidth, padHeight))
    pygame.display.set_caption('PyShooting')
    background = pygame.image.load('background.png')
    fighter = pygame.image.load('fighter.png')
    missile = pygame.image.load('missile.png')
    explosion = pygame.image.load('explosion.png')
    clock = pygame.time.Clock()
    lives = 3 # 초기 목숨 설정
```

초기목숨 설정 및 목숨감소

운석의 좌표값이 플레이어의 좌표값과 겹쳐질 때 목숨이 줄어들며 운석이 재생성되고 총 3번 목숨이 줄어들 때 게임이 끝나도록 만들었다

플레이어 움직임 및 기능

이 코드는 플레이어의 전반적인 동작코드다.
플레이어가 좌 우로만 움직이던 코드를 위아래로도 움직이며 키를 땔 때 움직임이 멈추도록 만들어졌으며 s 키를 클릭하여 필살기를 사용하면 화면에 있는 운석이 모두 제거된다.

```
while onGame:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                fighterX = -5
            elif event.key == pygame.K_RIGHT:
                fighterX = 5
            elif event.key == pygame.K_UP: # 위쪽 이동
                fighterY = -5
            elif event.key == pygame.K_DOWN: # 아래쪽 이동
                fighterY = 5
            elif event.key == pygame.K_SPACE:
                missileX = x + fighterWidth / 2
                missileY = y - fighterHeight
                # 미사일 업그레이드 단계에 따라 미사일 발사 개수 조정
                missile_spacing = 15 # 미사일 간격 설정
                for i in range(missile_upgrade):
                    # 발사 위치를 나란히 배열
                    missileXY.append([missileX + (i - (missile_upgrade - 1) / 2) * missile_spacing, missileY])
            elif event.key == pygame.K_s: # 필살기 발사 키 (s키)
                if special_missile_count > 0:
                    special_missile_count -= 1
                    isShot = True # 모든 운석이 제거되도록 설정
            elif event.key == pygame.K_u: # 무기 업그레이드 키 (u키)
                missile_upgrade += 1 # 업그레이드 단계 증가
                if missile_upgrade > 3: # 최대 3단계로 제한
                    missile_upgrade = 3
        elif event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                fighterX = 0
            if event.key == pygame.K_UP or event.key == pygame.K_DOWN: # 키를 떼면 움직임을 멈춤
                fighterY = 0
```

화면의 기능표시

```
def writeScore(count):  
    global gamePad  
    font = pygame.font.Font('NanumGothic.ttf', 20)  
    text = font.render('파괴한 운석 수: ' + str(count), True, (255, 255, 255))  
    gamePad.blit(text, (10, 0))  
  
def writePassed(count):  
    global gamePad  
    font = pygame.font.Font('NanumGothic.ttf', 20)  
    text = font.render('놓친 운석 : ' + str(count), True, (255, 0, 0))  
    gamePad.blit(text, (310, 0))  
  
def writeSpecialMissileCount(count):  
    global gamePad  
    font = pygame.font.Font('NanumGothic.ttf', 20)  
    text = font.render('필살기: ' + str(count), True, (0, 255, 0))  
    gamePad.blit(text, (10, 30))  
  
def writeMissileUpgrade(level):  
    global gamePad  
    font = pygame.font.Font('NanumGothic.ttf', 20)  
    text = font.render('무기 레벨: ' + str(level), True, (255, 255, 0))  
    gamePad.blit(text, (310, 30))  
  
def writeLives(count):  
    global gamePad  
    font = pygame.font.Font('NanumGothic.ttf', 20)  
    text = font.render('목숨: ' + str(count), True, (255, 255, 255))  
    gamePad.blit(text, (10, 60))
```

파괴한 운석 수
놓친 운석
필살기 수
무기레벨
목숨

등을 화면에 위치와 폰트 크기를 지정하여 표시한다.

Part 4

Image다운 및 적용

Image다운 및 적용

STEP 1

게임의 쓸 이미지를
다운받는다

>>

STEP 2

다운받은 이미지를
그림판을 이용하여
픽셀크기를 지정하고
png 파일로 저장한다

>>

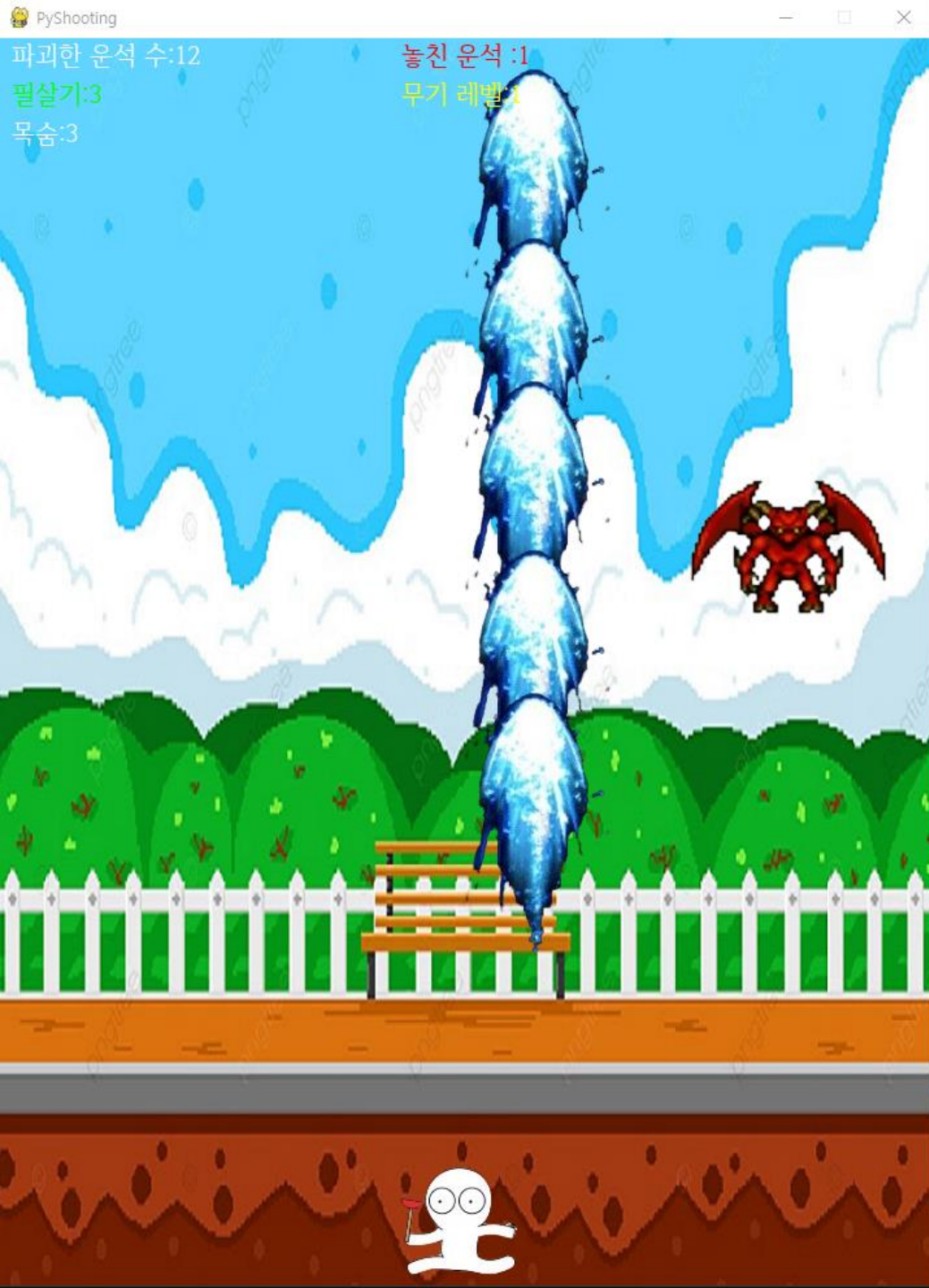
STEP 3

그림판을 사용하여
뒤에 배경이 생겼기
때문에 이미지 배경
삭제 사이트를 이용
하여 배경을 지운다

>>

STEP 4

게임에 적용한다!



Design.



Q & A

감사합니다.