

# 소프트웨어 프로젝트 2 Adventure Design

## 최종 보고서

별똥별 피하기 게임

20191595 박근우

20191595 박승현

## 목차

### 1.기획

-프로젝트 구상

### 2.요구사항

-기능적 요구사항

-인터페이스

-비기능적 요구사항

### 3.코드 구현

-코드 리뷰

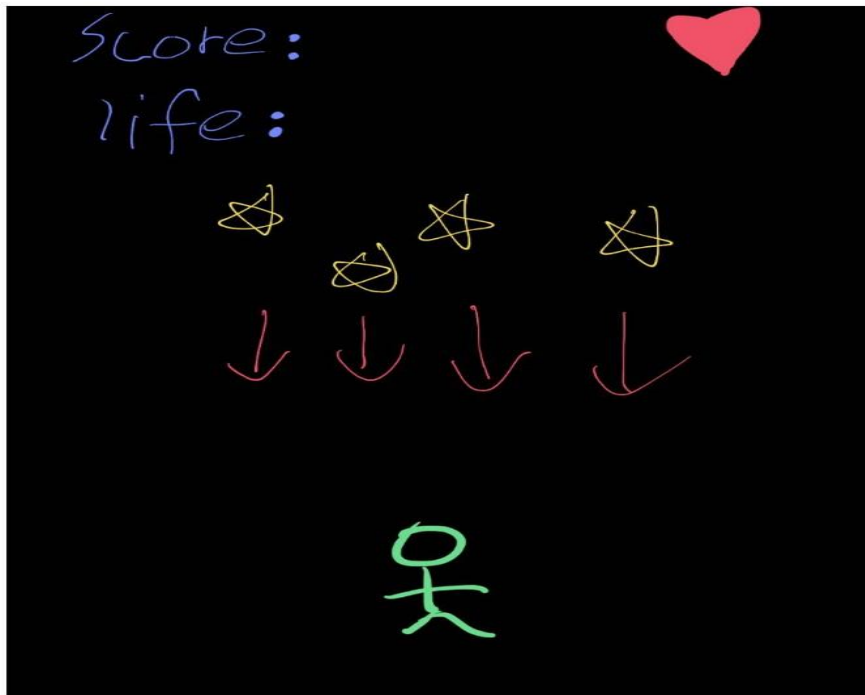
### 4.결과

### 5.느낀 점과 후기

# 계획

## 1\_1 프로젝트 구상

이번 AD 프로젝트에서는 3-rd party 라이브러리를 이용 하는게 핵심 키워드였다. 구글링을 통해 찾아보던 도중 Pygame 을 알게 되었고 우리만의 특별한 게임을 하나 만들자는 마인드로 같이 AD 프로젝트를 시작하였다. 다른 조들의 상황을 들어보니 복잡하면서 고 수준의 기능들을 기획한 조들이 꽤 많았지만 현실적으로 무리하지 말고 우리가 하고 싶은 것을 끝까지 제대로 만들어보자 라는 마음가짐으로 시작하게 되었다. 어릴 적 많이 볼 수 있었던 게임을 모티브 삼아 오브젝트를 피하거나 획득하여 점수, 라이프에 반영하고 키보드를 통해 자신의 캐릭터를 움직이면서 플레이하는 게임을 만들게 되었다. 캐릭터, 배경, 오브젝트들을 설정할 때 익숙한 것들을 사용해 만들면서 최대한 재밌고 능률이 잘 나올 수 있게 하는 것을 기본적으로 기획하였다. 다행히도 구글링을 통해 pygame 을 이용한 여러가지 종류의 게임들의 코드들을 접해 볼 수 있었고 처음부터 새로운 코드를 짤 수도 있었다는 거에 비해 꽤 많이 시간을 아낄 수 있었다. 이를 통해 기존의 아이디어나 코드를 그대로 가져와서 쓰는 것이 아니라면 현재 수준에서는 다른 사람의 코드를 잘 참고하여 우리의 아이디어를 표출 하는 것도 중요한 덕목 중에 하나라고 생각하게 되었다. 본격적으로 프로젝트를 시작하기위해서 게임 컨셉, 코드, 협력 방법을 정해야 했다. 앞서 말했던 게임 컨셉에서 오브젝트는 밤하늘에 떠있는 것들을 차용 하였다. 별을 피하고 배경을 평화로운 밤하늘 이미지를 통해 전체적인 게임의 분위기를 설정하였다. 그 후 기본적인 틀을 토대로 추가기능으로 스킬과 사운드를 추가하고 별 이외에 오브젝트를 추가하고 이때 실제로 오브젝트는 하트를 추가하여 먹으면 라이프를 회복하는 식으로 기획하였다. 또 메인 게임 페이지 이외에 선택 화면을 생성하여 난이도와 캐릭터 배경화면을 선택 할 수 있도록 하기로 기획하였다. 마지막으로 bgm 과 효과음도 추가하자는 의견이 나왔다.



위의 기획 단계에서 나온 간단한 구상도 이다.

구상도의 기본적인 틀을 바탕으로 추가기능을 생각하면서 게임을 만들어 나갔다.

## 요구사항

### 2\_1 게임 로직의 요구사항

별 피하기 게임에서 로직의 기능적으로 요구되는 것이 무엇인지 먼저 분석해볼 필요가 있었다.

- 가장 기본적인 캐릭터, 별, 하트를 생성해야 한다. ,
- 별, 하트의 움직이는 속도와 크기는 랜덤으로 한다.
- 캐릭터는 키보드 입력에 맞춰 이동한다.
- 점수와 스킬 게이지는 주기적으로 올라간다.

- 화면을 벗어나거나 캐릭터와 부딪힌 별,하트는 삭제하고 캐릭터는 화면을 벗어나지 않게 한다.
- 게임 페이지 이외에 옵션 선택 페이지를 만든다.(배경, 캐릭터, 난이도)
- 효과음과 배경음악을 설정한다.

## 2\_2 인터페이스

- 선택 화면 캐릭터, 배경, 난이도를 선택할 수 있는 창을 만든다.
- 메인 게임 화면을 만든다.
- 메인 게임 화면에는 게임 정보를 담은 텍스트 창을 띄우고 실시간으로 정보가 바뀔 수 있도록 한다.
- 선택 화면에도 선택할 때 필요한 정보를 텍스트로 보여준다.

## 3\_3 비기능적 요구사항

- pygame 을 이용한다.
- 사진이나 소리파일이 필요하다.

# 코드 구현

-코드 리뷰

# 기본 변수 선언

```
import pygame, random, sys
from pygame.locals import *
from Object import Object

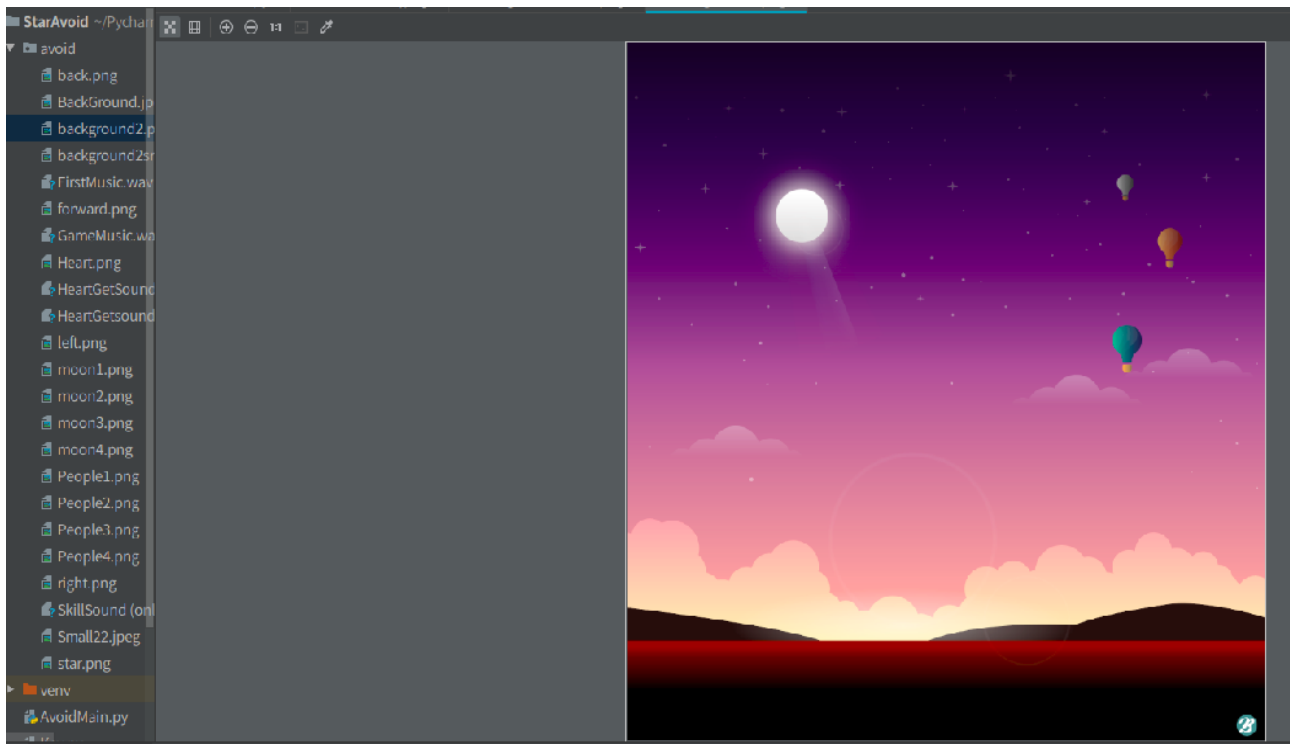
WINDOWWIDTH = 700
WINDOWHEIGHT = 800
TEXTCOLOR = (255, 255, 150)
temp = pygame.image.load("/home/user/PycharmProjects/StarAvoid/avoid/BackGround.jpeg")
BACKGROUND = pygame.image.load("/home/user/PycharmProjects/StarAvoid/avoid/BackGround.jpeg")
BACKGROUND2 = pygame.image.load("/home/user/PycharmProjects/StarAvoid/avoid/background2.png")
BACKGROUNDSMALL = pygame.image.load("/home/user/PycharmProjects/StarAvoid/avoid/Small22.jpeg")
BACKGROUND2SMALL = pygame.image.load("/home/user/PycharmProjects/StarAvoid/avoid/background2small.png")
FPS = 40
STARMINSIZE = 10
STARMAXSIZE = 40
STARMINSPD = 1
STARMAXSPD = 8
ADDNEWSTARRATE = 8
ADDNEWSTARRATELEVEL = [5, 12, 24]
ADDNEWHEARTRATE = 300
PLAYERMOVERATE = 5
FULL_HITPOINT = 4
FULL_Skill = 200
FULL_SkillLevel = [200, 400, 600]
```

우선 처음에 pygame, random, sys 등 모듈들을 import 하여 사용할 있게 합니다.

그리고 게임을 실행 했을 때 꼭 필요한 변수들을 선언해 주는데 WINDOWWIDTH는 게임 실행 창의 넓이 이고, WINDOWHEIGHT 는 높이 입니다.

TEXTCOLOR 는 화면의 띄어주는 text 글의 색깔을 설정한 것입니다.

그다음 temp 와 BACKGROUND 들은 pygame 모듈을 사용 하여 이미지를 가져와 변수에 저장하는 것인데



이렇게 파이썬 avoid 라는 디렉토리 안에 사진을 넣어두고 그 사진의 경로를 얻어 사진의 이미지를 저장하였습니다.

그리고 FPS는 주기적으로 발생하는 시스템을 설정하는 기준 시간입니다. (Clock Trick을 사용하였음)

이후 STARMINSIZE 등의 변수들은 위와 같이 각각의 필요한 값들을 선언한 것입니다.

#### # 게임 종료 함수

```
def terminate():
    pygame.quit()
    sys.exit()
```

만약 terminate 함수를 호출할 경우 게임을

종료하는 함수를 구현하였습니다.

#### # 텍스트 표시하기 함수

```
def drawText(text, font, surface, x, y):
    textobj = font.render(text, 1, TEXTCOLOR)
    textrect = textobj.get_rect()
    textrect.topleft = (x, y)
    surface.blit(textobj, textrect)
```

drawText 함수를 호출할 때 텍스트 정보들을 받아 그에 맞는 값들을 설정해 게임화면에 띄어 주는 함수를 구현하였습니다.

# 시작전 키입력 함수

```
def waitStart():
    global BACKGROUND
    global BACKGROUND2
    global temp
    global ADDNEWSTARRATE
    global ADDNEWSTARRATELEVEL
    global FULL_Skill
    global FULL_SkillLevel
    global player1
    global player2
    global playerImageSelect
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                terminate()
            if event.type == KEYDOWN:
                if event.key == K_ESCAPE: # ESC 키를 눌러 종료한다
                    terminate()
                elif event.key == pygame.K_1:
                    BACKGROUND = temp
                elif event.key == pygame.K_2:
                    BACKGROUND = BACKGROUND2
                elif event.key == pygame.K_e:
                    ADDNEWSTARRATE = ADDNEWSTARRATELEVEL[2]
                    FULL_Skill = FULL_SkillLevel[0]
                elif event.key == pygame.K_n:
                    ADDNEWSTARRATE = ADDNEWSTARRATELEVEL[1]
                    FULL_Skill = FULL_SkillLevel[1]
                elif event.key == pygame.K_h:
                    ADDNEWSTARRATE = ADDNEWSTARRATELEVEL[0]
                    FULL_Skill = FULL_SkillLevel[2]
                elif event.key == pygame.K_l:
                    playerImageSelect = player1
                elif event.key == pygame.K_r:
                    playerImageSelect = player2
        return
```

우선 전역변수를 호출하여 만약 게임창에서 키입력을 받았을 때 그 키 입력에 따른 각각의 결과 값들을 표현하였습니다. 예를 들어 키보드로 R을 입력했을 경우 pygame.K\_r:에 반응하여 실행 값을 실행한다.

# 게임 윈도우, 마우스를 설정한다.

```
pygame.init()
mainClock = pygame.time.Clock()
windowSurface = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.mouse.set_visible(False)
```

우선 파이게임을 만들고(pygame.init()) 기준 시간을 설정해줄 Clock 을 선언했습니다. 그리고 변수로 설정한 넓이와 높이를 갖는 윈도우를 만들었습니다.

# 사운드 설정

```
SkillSound = pygame.mixer.Sound('/home/user/PycharmProjects/StarAvoid/avoid/SkillSound (online-audio-converter.com).wav')
HeartGetSound = pygame.mixer.Sound("/home/user/PycharmProjects/StarAvoid/avoid/HeartGetsound.wav")
GameMusic = pygame.mixer.Sound("/home/user/PycharmProjects/StarAvoid/avoid/GameMusic.wav")
FirstMusic = pygame.mixer.Sound("/home/user/PycharmProjects/StarAvoid/avoid/FirstMusic.wav")
```

pygame.mixer.Sound 메소드(파이썬 에서 wav 음성파일을 받아 객체에 저장하는메소드)를 호출하여 각각 상황에 맞는 사운드를 설정하였습니다.

#### # 이미지 설정

```
playerImage = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/People1.png')
playerRect = playerImage.get_rect()
playerImage1 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/People1.png')
playerImage2 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/People2.png')
playerImage3 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/People3.png')
playerImage4 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/People4.png')
playerImage5 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/forward.png')
playerImage6 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/back.png')
playerImage7 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/left.png')
playerImage8 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/right.png')
starImage = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/star.png')
moonImage1 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/moon1.png')
moonImage2 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/moon2.png')
moonImage3 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/moon3.png')
moonImage4 = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/moon4.png')
heartImage = pygame.image.load('/home/user/PycharmProjects/StarAvoid/avoid/Heart.png')
player1 = [playerImage1, playerImage2, playerImage3, playerImage4]
player2 = [playerImage5, playerImage6, playerImage7, playerImage8]
playerImageSelect = player1
```

pygame.image.load 모듈을 이용하여 파이썬 파일 안에 있는 사진 파일의 경로를 얻어와 각각의 변수에 이미지 정보를 넣어주는 함수입니다.

player1, player2는 정의한 캐릭터 이미지를 앞,뒤, 오른쪽, 왼쪽 순으로 리스트에 정렬한 것이고, playerImageSelect은 선택한 플레이어를 표현하기 위한 변수입니다.

#### # 게임 'Start' 화면 보여주기

```
FirstMusic.play()
windowSurface.fill([0,0,0])
windowSurface.blit(BACKGROUNDSMALL, pygame.Rect(100,375,0,0))
windowSurface.blit(BACKGROUND2SMALL, pygame.Rect(400,400,0,0))
drawText('Avoid Star !!', font, windowSurface, (WINDOWWIDTH / 3), (WINDOWHEIGHT / 3))
drawText('Press 1 or 2 choice BackGround', font, windowSurface, (WINDOWWIDTH / 8), (WINDOWHEIGHT / 3) + 50)
pygame.display.update()
waitStart()
topScore = 0
```

우선 처음에 FirstMusic.play()로 아까 지정한 음악을 재생하고 메인 window를 검은색으로 지정합니다. 그리고 BACKGROUND 이미지를 가져와 widowSurface.blit() 함수를 사용하여 지정된 화면 위치에 이미지를 띄웁니다.

그리고 drawText 함수를 호출하여 게임을 시작하기 위해 'Press 1 or 2' 라는 텍스트를 마찬가지로 지정된 위치에 띄웁니다. 그리고 pygame.display.upate() 모듈을 사용하여



화면에 변경된 값을 적용해 줍니다. 그리고 waitStart() 함수를 호출하여 키보드로 입력하는 값이 1이거나 2일 경우 그에 맞는 배경화면을 설정합니다

# 난이도 선택 화면

```
windowSurface.fill([0,0,0])
drawText('Select Level E or N or H', font, windowSurface, (WINDOWWIDTH / 5), (WINDOWHEIGHT / 3))
pygame.display.update()
waitStart()
```

만약 처음 화면에서 입력 값을 얻었다면 바로 다시 화면을 호출하는데 아까와 마찬가지로 windowSurface를 검은색으로 채우고 “Select Level” 텍스트를 지정된 위치에 띄어줍니다. 그리고 또 다시 waitStart() 함수를 불러 입력 값을 얻습니다.

# 캐릭터 선택 화면

```
windowSurface.fill([0,0,0])
drawText('Select Character L or R', font, windowSurface, (WINDOWWIDTH / 5), (WINDOWHEIGHT / 3))
pygame.display.update()
windowSurface.blit(playerImage1, pygame.Rect(200,400,0,0))
windowSurface.blit(playerImage5, pygame.Rect(400,390,0,0))
pygame.display.update()
waitStart()
FirstMusic.stop()
```

또 다시 입력 값을 얻었다면 아까와 같이 마찬가지로 다른 화면을 만들고 텍스트를 띄워 줍니다. 그리고 고를 수 있는 플레이어 사진을 화면에 띄우고 다시 waitStart() 함수를 호출하여 입력 값을 얻어 냅니다. 마지막으로 처음에 재생한 음악을 멈춥니다.

# 게임이 시작 되었을 때 기본적인 변수 선언

```
while True:
    GameMusic.play()
    star = []
    heart = []
    score = 0
    playerRect.topleft = (WINDOWWIDTH / 2, WINDOWHEIGHT - 50)
    moveLeft = moveRight = moveUp = moveDown = False
    game = False
    starAddCounter = 0
    heartAddCounter = 7
    hitPoint = FULL_HITPOINT
    SkillTimer = 0
```

이제 모든 입력 값을 받은 뒤에는 GameMusic을 재생하고 나중에 별과 하트 등 나중에 필요한 기본적인 변수들을 선언합니다.

여기서 playerRect.topleft는 아까 지정한 플레이어의 위치를 화면의 가운데 맨 아래로 이동시키는 것입니다.

# 게임이 시작되었을 때

```
while True:
    score += 1
    if SkillTimer < FULL_Skill:
        SkillTimer += 1
    for event in pygame.event.get():
        if event.type == QUIT:
            terminate()
        if event.type == KEYDOWN:
            if event.key == ord('q'):
                if SkillTimer == FULL_Skill:
                    SkillSound.play()
                    for b in star[:]:
                        star.remove(b)
                    SkillTimer = 0
            else:
                if event.key == K_LEFT or event.key == ord('a'):
                    moveRight = False
                    moveLeft = True
                    playerImage = playerImageSelect[2]
                if event.key == K_RIGHT or event.key == ord('d'):
                    moveLeft = False
                    moveRight = True
                    playerImage = playerImageSelect[3]
                if event.key == K_UP or event.key == ord('w'):
                    moveDown = False
                    moveUp = True
                    playerImage = playerImageSelect[1]
                if event.key == K_DOWN or event.key == ord('s'):
                    moveUp = False
                    moveDown = True
                    playerImage = playerImageSelect[0]

        if event.type == KEYUP:
            if event.key == K_ESCAPE:
                terminate()
            if event.key == K_LEFT or event.key == ord('a'):
                moveLeft = False
            if event.key == K_RIGHT or event.key == ord('d'):
                moveRight = False
            if event.key == K_UP or event.key == ord('w'):
                moveUp = False
            if event.key == K_DOWN or event.key == ord('s'):
                moveDown = False
```

이제 본격적으로 게임이 시작되고 처음에 점수를 계속해서 1을 더하며 올립니다. 그리고 스킬 게이지 또한 한계 값이 아닐 때 까지 계속해서 1을 더하며 올립니다.

그 다음 for 문을 사용하여 얻어지는 값들에 대해 계속해서 작업을 시작하는데 만약 얻어 진 값이 'Esc'키 일 경우 게임을 종료하고, 'q' 일 경우에는 게이지가 한계점에 도달

했을 때 스킬을 사용했을 때 사운드를 재생하고 star 안의 모든 항목을 지우는데 이것은 생성된 모든 별을 지우는 코드입니다. 마지막으로 만약 'w', 's', 'd', 'a', 방향 키일 경우 캐릭터의 각각의 위치에 관한 것들은 False, True 로 선언해 주면서 캐릭터가 이동의 유무를 판단합니다.

만약 키 입력에서 입력이 되지 않을 경우에는 모두 False로 선언하여 캐릭터가 움직이지 않도록 합니다.

# 별과 하트를 추가하기

```
if not game:
    starAddCounter += 1
    heartAddCounter += 1
if starAddCounter == ADDNEWSTARRATE:
    starAddCounter = 0
    starSize = random.randint(STARMINSIZE, STARMAXSIZE)
    newstar = {'rect': pygame.Rect(random.randint(0, WINDOWWIDTH - starSize), 0 - starSize, starSize,
                                    starSize),
               'speed': random.randint(STARMINSPEED, STARMAXSPEED),
               'surface': pygame.transform.scale(starImage, (starSize, starSize)),
               }
    star.append(newstar)
if heartAddCounter == ADDNEWHEARTRATE:
    heartAddCounter = 0
    heartSize = (40)
    newheart = {'rect': pygame.Rect(random.randint(0, WINDOWWIDTH - heartSize), 0 - heartSize, heartSize,
                                    heartSize),
                'speed': (8),
                'surface': pygame.transform.scale(heartImage, (heartSize, heartSize)),
                }
    heart.append(newheart)
```

계속해서 별과 하트가 생성될 한계점에 도달 할 때까지 1씩 더하다가 도달하면 별과 하트를 생성하는데 별은 사이즈가 지정된 범위 내에서 정해지도록 랜덤 값을 통해 얻어내고, 별의 이미지를 띄울 위치('rect'), 속도('speed'), 크기('surface')를 각각 설정하여 사전 형태인 상태로 star 리스트에 추가합니다

하트는 별과 다른 점은 속도, 크기가 일정한 것이고 나머지는 똑같이 구현합니다.

# 플레이어를 움직인다

```
if moveLeft and playerRect.left > 0:
    playerRect.move_ip(-1 * PLAYERMOVERATE, 0)
if moveRight and playerRect.right < WINDOWWIDTH:
    playerRect.move_ip(PLAYERMOVERATE, 0)
if moveUp and playerRect.top > 0:
    playerRect.move_ip(0, -1 * PLAYERMOVERATE)
if moveDown and playerRect.bottom < WINDOWHEIGHT:
    playerRect.move_ip(0, PLAYERMOVERATE)
```

아까 호출한 키 입력 값에서 True 인 값이 있고 캐릭터가 화면 가장자리에 있지 않을 경우

캐릭터의 위치를 설정한 속도만큼 움직이려는 방향으로 움직입니다.

# 별과 하트를 아래로 내리기

```
for b in star:
    if not game:
        b['rect'].move_ip(0, b['speed'])
```

```
for b in heart:
    if not game:
        b['rect'].move_ip(0, b['speed'])
```

별과 하트가 생성 되었을 때 star, heart 안에 있는 사전들을 불러와 각각 키에 맞는 값을 통해 위치를 위에서 아래로 “speed”의 속도로 내립니다.

# 바닥에 떨어진 별과 하트 삭제하기

```
for b in star[:]:
    if b['rect'].top > WINDOWHEIGHT:
        star.remove(b)
```

```
for b in heart[:]:
    if b['rect'].top > WINDOWHEIGHT:
        heart.remove(b)
```

별과 하트의 위치가 화면의 가장 아래에 도달 했을 때 그 별과 하트는 삭제합니다.

# 점수, 탑 스코어, 스킬 쿨 타임, HP 그리기

```
drawText('Score: %s' % (score), font, windowSurface, 10, 0)
drawText('Top Score: %s' % (topScore), font, windowSurface, 10, 40)
drawText('Skill gauge: %s' % (SkillTimer), font, windowSurface, 10, 80)
drawText('HP: %s' % (str(hitPoint)), font, windowSurface, 10, 120)
```

현재 얻어지는 score와 topScore에 있는 값들을 지정된 위치에 띄워 줍니다. 그리고 그 밑에는 SkillTimer 와 남은 체력을 보여주는 텍스트를 띄워 줍니다.

# HP에 따라 달의 모양 그리기

```
if hitPoint == 4:
    windowSurface.blit(moonImage4, (WINDOWWIDTH - 40, 40))
elif hitPoint == 3:
    windowSurface.blit(moonImage3, (WINDOWWIDTH - 40, 40))
elif hitPoint == 2:
    windowSurface.blit(moonImage2, (WINDOWWIDTH - 40, 40))
elif hitPoint == 1:
    windowSurface.blit(moonImage1, (WINDOWWIDTH - 40, 40))
```

각각의  
힌트

포인트가 되었을 때 처음에는 초승달 모양을 띄어 주었다가 값이 변할 경우 그에 맞는 이미지를 띄웁니다.

# 게임 화면에 별과 하트 그리기

```
for b in star:
    windowSurface.blit(b['surface'], b['rect'])
pygame.display.update()
for b in heart:
    windowSurface.blit(b['surface'], b['rect'])
pygame.display.update()
```

아까 리스트에 추가된

값들을 호출하여 화면에 이미지와 위치에 맞게 띄웁니다.

# 플레이어와 부딪힌 별과 하트가 있는지 검사하기

```
if Object.playerHasHitStar(playerRect, star):
    hitPoint -= 1
    if hitPoint == 0:
        pygame.display.update()
        if score > topScore:
            topScore = score
        break
elif Object.playerHasHitHeart(playerRect, heart):
    HeartGetSound.play()
    if hitPoint == 4:
        pass
    else:
        hitPoint += 1
mainClock.tick(FPS)
```

여기서 처음에

Object class 의 함수를 불러오는데

```

class Object():
    def __init__(self):
        pass

    def playerHasHitStar(playerRect, star):
        for b in star:
            if playerRect.colliderect(b['rect']):
                star.remove(b)
                return True
        return False

    def playerHasHitHeart(playerRect, heart):
        for b in heart:
            if playerRect.colliderect(b['rect']):
                heart.remove(b)
                return True
        return False

```

만약

playerRect.colliderect() 모듈을 이용하여 충돌 하였다면 그 별과 하트를 지우고 True를 리턴 아닌 경우는 False 를 리턴합니다.

즉 별이 피격되었을 경우에는 hitpoint를 -1 하는데 만약 0이 되면 게임을 종료하는 창을 호출하고, 현재 스코어가 저장한 topscore보다 클 경우에는 값을 바꿉니다.

이에 비해 하트가 피격되었을 경우에는 얻었을 때의 사운드를 재생, hitpoint를 +1 합니다. hitpoint가 4일 경우는 그냥 pass 합니다.

그리고 마지막 mainClock.tick(FPS)를 호출하여 정해진 기준시간(FPS)에 따라 함수를 반복 합니다.

# 게임을 멈추고 “Game over” 장면 보여주기

```

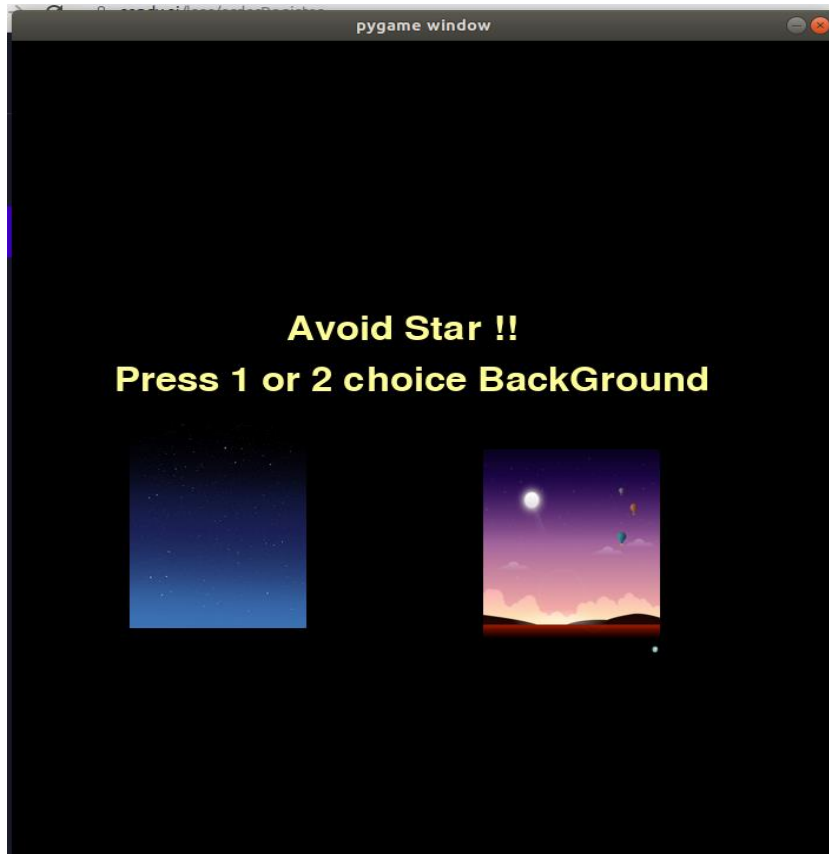
GameMusic.stop()
pygame.mixer.music.stop()
drawText('GAME OVER', font, windowSurface, (WINDOWWIDTH / 3), (WINDOWHEIGHT / 3))
drawText('Press a key to play again.', font, windowSurface, (WINDOWWIDTH / 3) - 80, (WINDOWHEIGHT / 3) + 50)
pygame.display.update()
waitStart()

```

우선 게임이 시작 될 때 재생한 음악을 종료 하고 “Game Over”, “Press a key” 등의 텍스트를 띄웁니다. 그리고 waitStart()를 통하여 다시 키 입력을 받는 것으로 코드를 마무리 합니다.

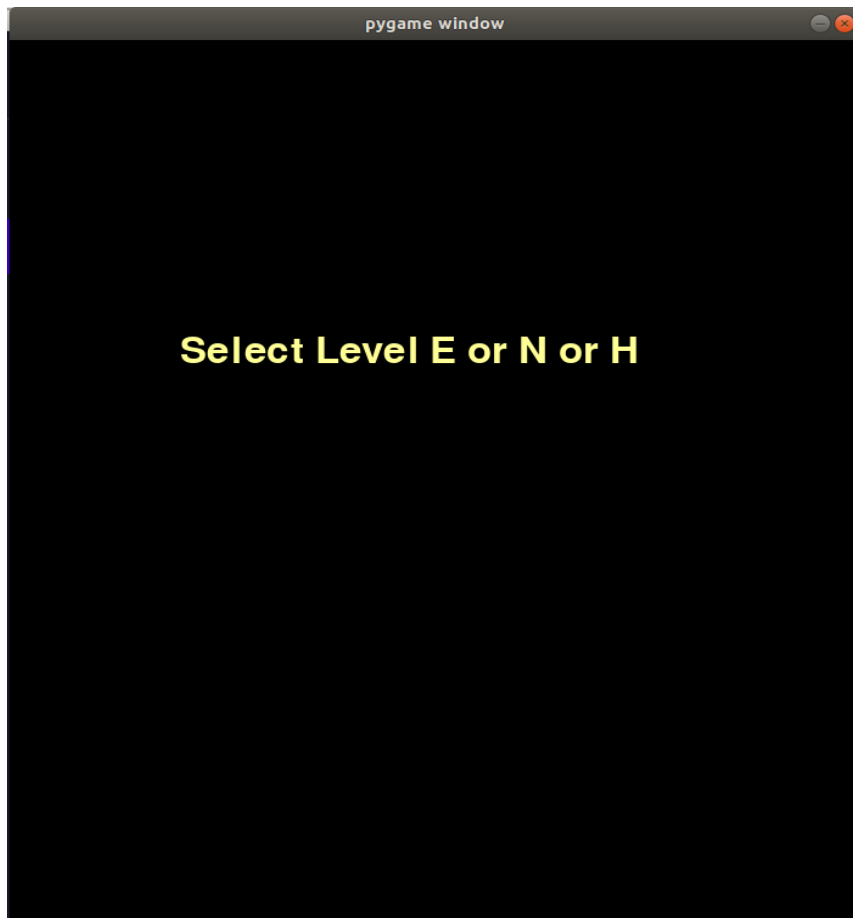
## 결과

보고서 외에 시연영상도 함께 제출하니 결과 내용과 함께 참고 해주시면 감사하겠습니다.



< 배경 화면 선택 창 >

게임 시작 시 나오는 화면으로 키보드 입력을 통해 배경화면을 선택할 수 있게 하였습니다.

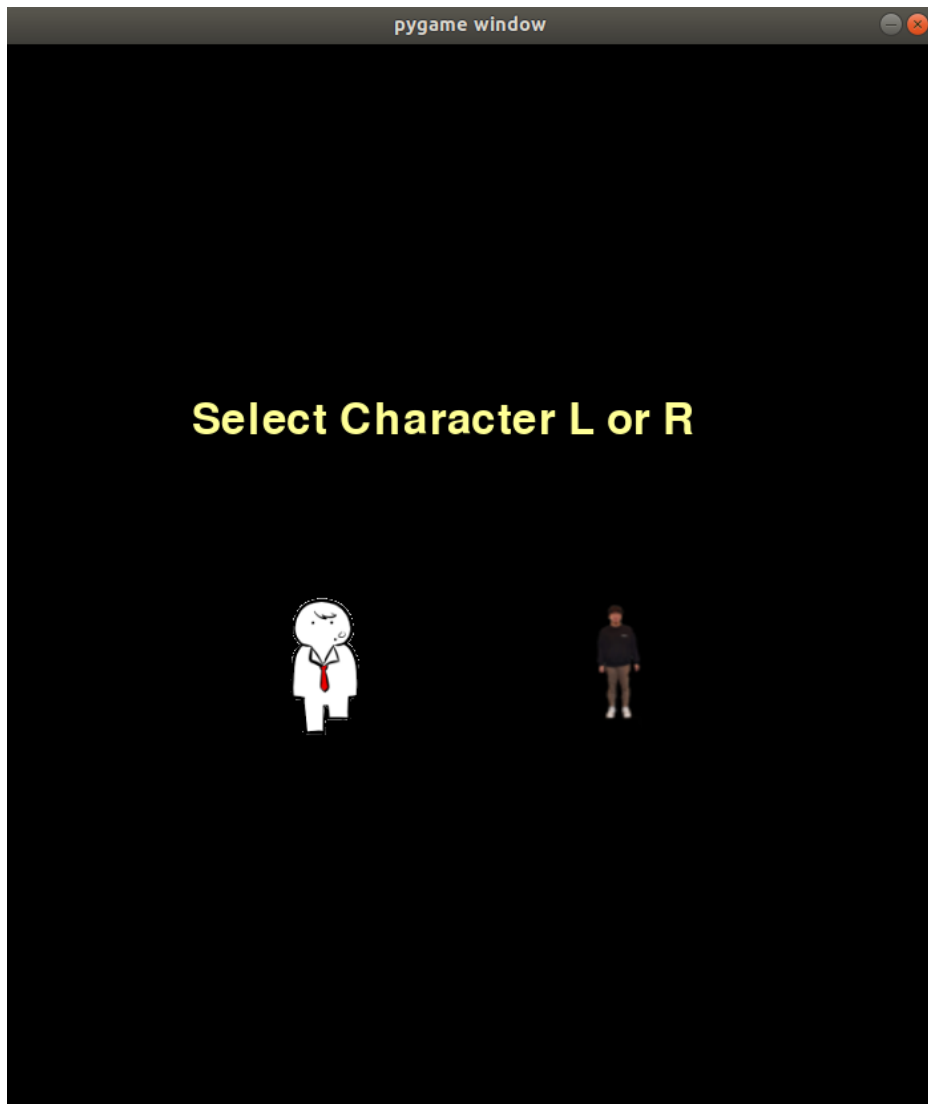


< 난이도 선택 창 >

배경화면을 선택하면 난이도 선택으로 넘어갑니다.

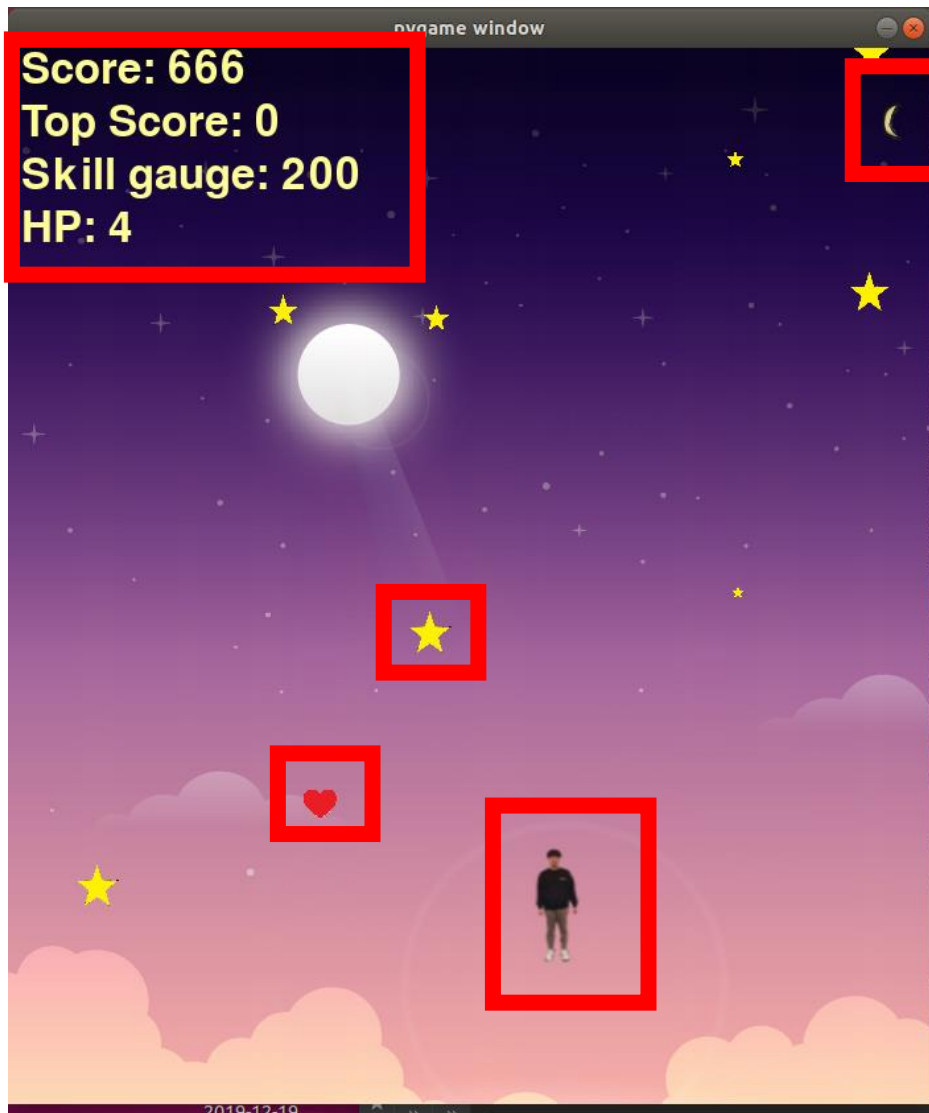
역시 키보드 입력을 통해 난이도를 선택할 수 있습니다.





< 캐릭터 선택 창 >

마지막은 캐릭터 선택 입니다.



< 게임 화면 >

모든 선택을 하면 선택한 정보를 입력 받아 게임 화면을 띄웁니다.

## 느낀 점

### < 박승현 >

먼저 소프트웨어 프로젝트 2, AD 프로젝트는 저희 둘에게는 큰 도전과제였습니다. 1 학기 때부터 대학교 공부는 생각만큼 호락호락하지 않았고 성적이 생각한만큼 나오지도 않아 큰 상실감에 빠지기도 했었습니다. 1 차 라이브코딩 때도 점수가 좋지 못했고 수업을 따라가는데 있어서 벅차기도 하였습니다. 하지만 친구들과 같이 공부하면서 점차 코딩이라는 활동과 학문에 재미가 붙기 시작했고 2 차 라이브코딩 테스트에서 만점, 중간고사, 기말고사를 거치면서 실력이 쭉쭉 늘어나는 제 모습을 보면서 너무 뿌듯하였습니다. 아직 많이 부족한 실력이지만 이정도까지 성장할 수 있었던 것은 주변 친구들, 교수님 등 여러 소중한 사람들의 덕분이라고 생각합니다. AD 프로젝트를 그저 과제가 아닌 재미있는 하나의 진짜 프로젝트라고 생각하고 활동 하다보니 어느새 한학기가 끝나 있었고 저희는 게임을 하나 완성해 있었습니다. 하나 아쉬운 점이 있다면 모듈화를 시키지 못한 점인데 그래도 아쉬움과 미련보다는 뿌듯함이 더 큰 활동이자 수업이자 한 학기였다고 생각합니다. 한 학기동안 정말 감사했습니다.

### < 박근우 >

처음에 모듈을 사용하는 것에 있어서 아직 알지 못하는 점이 많아 프로젝트를 진행하는 것이 어려움이 많았다. 하지만 이번 프로젝트에서 사용한 pygame 이라는 모듈을 학습, 응용한 프로그램을 제작하면서 Python 모듈 구조에 대해 많이 배울 수 있었다. 한가지 아쉬운 점은 우리의 프로젝트가 class 단위로 묶는 것을 좀 더 구조적으로 만들지 못한 것이었다. 그래서 프로젝트가 끝난 이후에도 게임의 완성도를 높힐 것이다.

감사합니다.