SPACE SHOOTER
슈팅게임 개발

안승진 (팀장)
정찬홍
임강현
최열호

목차

table of contents

1 프로젝트 기획

2 WBS 및 개발 일정

3 기능별 구현 사항

4 개발결과

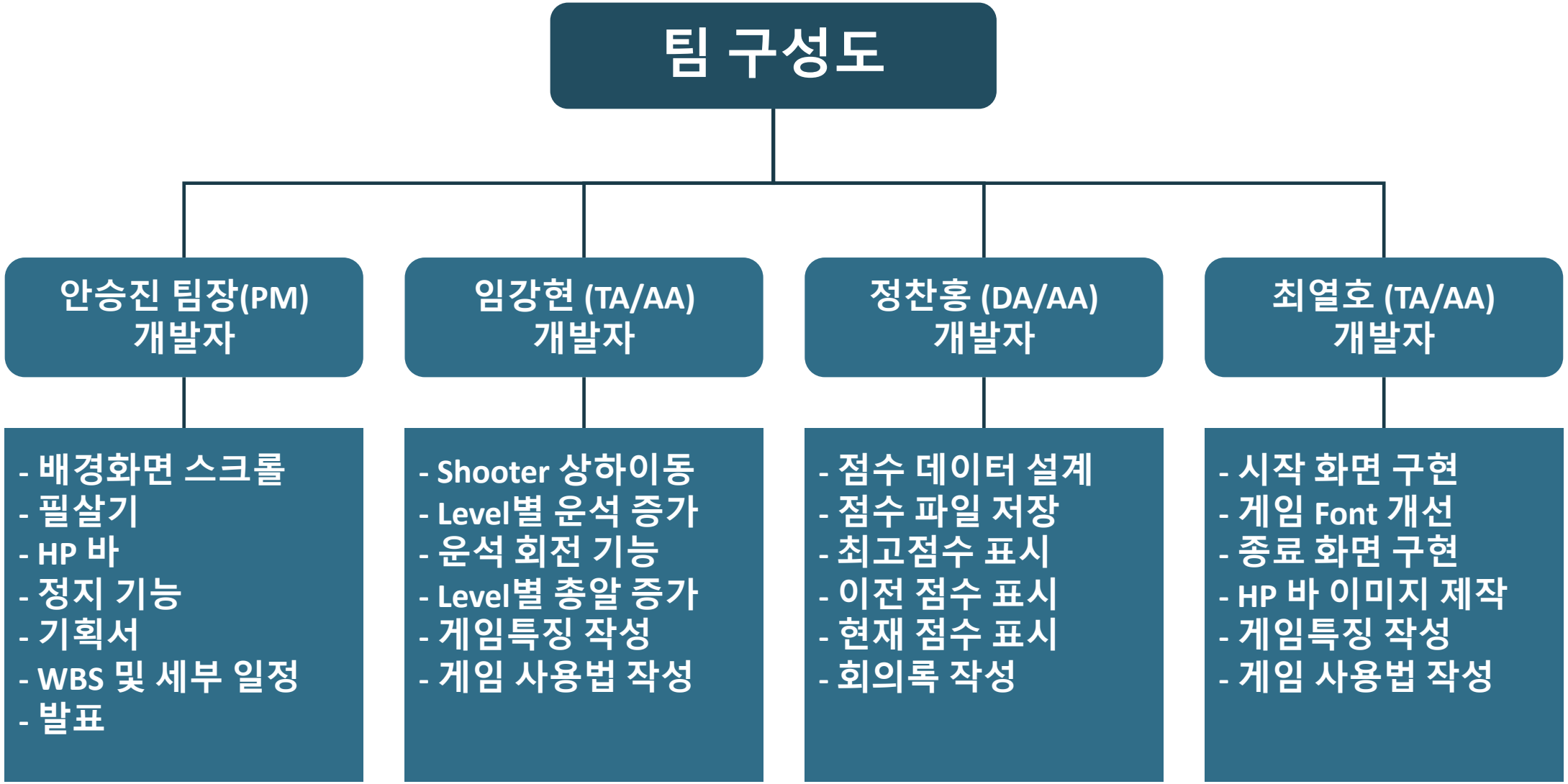
별첨 회의록

1

프로젝트 기획

Part 1

프로젝트 팀 구성



시놉시스

우주 이변으로 끊임 없이 우주에서 날아오는 운석을
지구 대기권에 진입하기 전에 쏴서 부숴버리는 게임으로
운석을 놓치거나 맞으면
플레이어의 HP가 10%씩 줄어들어 0%가 되면 게임이 종료한다.
게임이 진행되면서 빨라지는 운석과 늘어난 숫자를 대비하여
플레이어에게는 필살기를 포함한 강력한 무기를 제공한다.

Part 1

데이터 및 추진방향

데이터

- **폰트**
  8bit 닌텐도 게임스러운 분위기 폰트 적용

- **배경이미지**
  근거리/ 원거리 화면 입체감

- **적**
  여러가지 운석 모양이 회전

- **필살기, 충돌 , 발사 사운드**
  액션에 맞는 사운드 적용

- **우주선 이미지**
  게임 배경에 걸맞는 디자인 적용

- **게임 난이도**
  점수에 따라 난이도 증가 (LEVEL)

추진 방향

- 게임 중에는 리얼한 비디오 및 오디오를 사용한다.

- 사용자에게 몰입감과 중독성을 제공하는 게임.

- 레트로한 아케이드 디자인으로 게임 분위기 조성

2

WBS 및 개발 일정

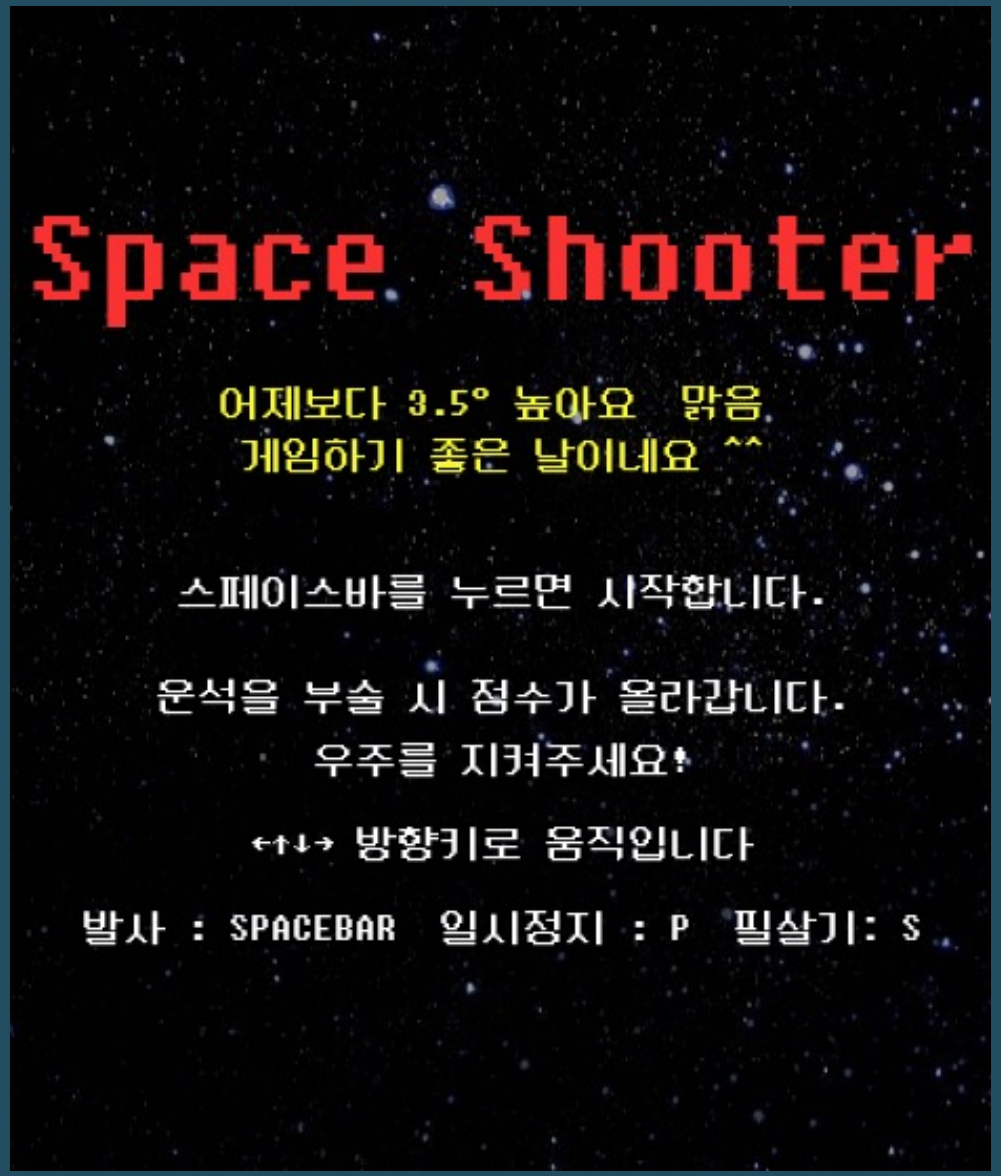
Part 2

WBS 및 개발 일정

		이름	기간	시작	종료	자원 이름
2		디일경 계획 수립/업무 분장	1 day?	23. 11. 30 ..	23. 11. 30 ..	안승진
3		배경 스크롤 구현	1 day?	23. 11. 30 ..	23. 11. 30 ..	안승진
4		필살기 구현	1 day?	23. 12. 1 ...	23. 12. 1 ...	안승진
5		1차 소스 통합	1 day?	23. 12. 4 ...	23. 12. 4 ...	안승진;임강현;정찬홍;최열호
6		HP 기능 구현	1 day?	23. 12. 4 ...	23. 12. 4 ...	안승진
7		2차 소스 통합	0.875 ..	23. 12. 5 ...	23. 12. 5 ...	안승진;임강현;정찬홍;최열호
8		경지 기능 구현	1 day?	23. 12. 5 ...	23. 12. 5 ...	안승진
9		3차통합/신뢰성 시험	1.25 d...	23. 12. 6 ...	23. 12. 7 ...	안승진;임강현;정찬홍;최열호
10		개발 완료 보고서 및 산출물 완료	2.25 d...	23. 12. 5 ...	23. 12. 7 ...	안승진;임강현;정찬홍;최열호
11		점수 저장 기능 구현	2 days?	23. 12. 1 ...	23. 12. 4 ...	정찬홍
12		점수 읽기 기능 구현	3 days?	23. 12. 4 ...	23. 12. 6 ...	정찬홍
13		시작 메뉴 구현	2 days?	23. 12. 1 ...	23. 12. 4 ...	최열호
14		종료 메뉴 구현	3 days?	23. 12. 4 ...	23. 12. 6 ...	최열호
15		슛터 상하 이동 구현	2 days?	23. 12. 1 ...	23. 12. 4 ...	임강현
16		레벨별 운석 개수 증가 구현	1 day?	23. 12. 5 ...	23. 12. 5 ...	임강현
17		레벨별 총알 개수 증가 구현	2 days?	23. 12. 5 ...	23. 12. 6 ...	임강현

3

기능별 구현 사항



3.1

시작 화면

화면 이벤트 구분

- 타이틀 화면과 게임 화면 이벤트 구분
  - runGame()함수 바탕으로 코드 작성
  - onGame = False일시 타이틀
  - onGame = True일시 게임화면 이벤트 처리

```
def runGame():
    global gamePad, clock, background, fighter, missile, explosion,
```

```
onGame = False
while not onGame:
    for event in pygame.event.get():
        if event.type in [pygame.QUIT]:  # 게임 프로그램 종료
            pygame.quit()
            sys.exit()

        if event.type in [pygame.KEYDOWN]:
            if event.key == pygame.K_SPACE:
                onGame = True
```

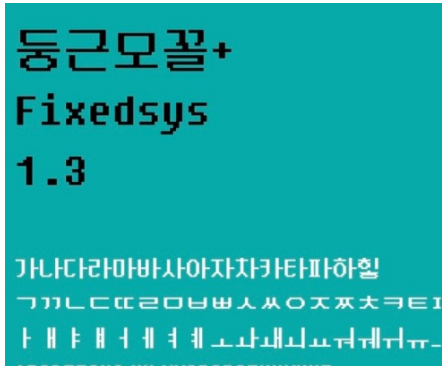
- 화면 업데이트 딜레이 설정

```
pygame.display.update()
clock.tick(60)
```

- 어울리는 레트로 글꼴 적용

```
font = pygame.font.Font("PS/font.ttf", 70)  # 폰트와 크기 설정
font2 = pygame.font.Font("PS/font.ttf", 20)
```

```
welcome_text = font.render("Space Shooter", Tru
welcome_rect = welcome_text.get_rect(center=(pa
gamePad.blit(welcome_text, welcome_rect)
welcome_text = font2.render(weather, True, (250
welcome_rect = welcome_text.get_rect(center=(pa
gamePad.blit(welcome_text, welcome_rect)
```



게임 설명 및 조작법

- **게임 설명 및 조작법 표시**
  받은 글꼴을 바탕으로 설명 작성

```
press_text = font2.render("스페이스바를 누르면 시작합니다.",
press_rect = press_text.get_rect(center=(padWidth / 2, pa
gamePad.blit(press_text, press_rect)

instruction_text = font2.render(
    "운석을 부술 시 점수가 올라갑니다.",
    True,
    (255, 255, 255),
)
instruction_rect = instruction_text.get_rect(
    center=(padWidth / 2, padHeight / 2 + 50)
)
gamePad.blit(instruction_text, instruction_rect)

instruction_text2 = font2.render(
    "우주를 지켜주세요!",
    True,
    (255, 255, 255),
```

```
instruction_text3 = font2.render(
    "←↑↓→ 방향키로 움직입니다",
    True,
    (255, 255, 255),
)
```

```
instruction_rect3 = instruction_text3.get_rect(
    center=(padWidth / 2, padHeight / 2 + 120)
)
gamePad.blit(instruction_text3, instruction_rect3)

instruction_text4 = font2.render(
    "발사 : SPACEBAR  일시정지 : P  필살기: S",
    True,
    (255, 255, 255),
)
instruction_rect4 = instruction_text4.get_rect(
    center=(padWidth / 2, padHeight / 2 + 160)
)
gamePad.blit(instruction_text4, instruction_rect4)
```

날씨정보

- 날씨 정보 크롤링
  시작화면 진입 시 Naver 날씨 정보 Crawling 하여 보여줌

```
def get_weather():
    url = "https://search.naver.com/search.naver?where=nexearch&sm=top_sug.pre&fbm=0&acr=1&acq=%EB%82%98&qdt=0

    res = req.urlopen(url)
    soup = BeautifulSoup(res, "html.parser")
    # print(soup)
    weather_condition = soup.select_one(
        "#main_pack > section.sc_new.cs_weather_new._cs_weather > div > div:nth-child(1) > div.content_wrap > 
    )
    return weather_condition.get_text()
```

- 시작메뉴에서 호출하여 날씨정보 표시 할 때 사용되는 get weather()

3.2

플레이 화면

플레이어 이동 관련

- 플레이어 상하 이동
  - 방향키에 따라 플레이어 이동

```
elif event.key == pygame.K_UP:
    fighterY -= 5  # Move up (backward)
elif event.key == pygame.K_DOWN:
    fighterY += 5  # Move down (forward)
```

- 전투기 위치 업데이트 및 가로 & 세로 경계 처리

```
# 전투기 위치 재조정
x += fighterX
y += fighterY
# 가로 경계 처리
if x < 0:
    x = 0
elif x > padWidth - fighterWidth:
    x = padWidth - fighterWidth

# 세로 경계 처리
if y < 0:
    y = 0
elif y > padHeight - fighterHeight:
    y = padHeight - fighterHeight
```

Rock관련 개수 및 회전 기능

- 레벨 증가시 Rock 개수와 스피드 증가
  - Rock과 똑같이 Rock2 , Rock3 생성 후 적용

```
if level == 3:
    if rockY2 > padHeight:
        # 새로운 운석 (랜덤)
        rock2 = pygame.image.load(random.choice(rockImage))
        rockSize2 = rock2.get_rect().size
        rockWidth2 = rockSize[0]
        rockHeight2 = rockSize[1]
        rockX2 = random.randrange(0, padWidth - rockWidth2)
        rockY2 = -30
        rockPassed += 1
        shooter_life -= 1
    if isShot2:
        # 운석 폭발
        drawObject(explosion, rockX2, rockY2)  # 운석 폭발 그리기
        rock2 = pygame.image.load(random.choice(rockImage))
        rockSize2 = rock2.get_rect().size
        rockWidth2 = rockSize2[0]
        rockHeight2 = rockSize2[1]
        rockX2 = random.randrange(0, padWidth - rockWidth2)
        rockY2 = -30
        isShot2 = False
        # 운석 맞추면 속도 증가
        rockSpeed += 0.15
        if rockSpeed >= 10:
            rockSpeed = 10
```

- Rock 로테이션
  - Rotation_angle 변수 생성후 숫자 증가

```
rotation_angle = random.randint(1, 20)
```

```
ag = random.randint(1, 10)
rotation_angle = (rotation_angle + ag) % 360
rotation_angle1 = -((rotation_angle + ag) % 360)
```

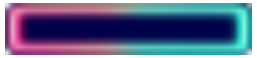
  - Rock을 pygame.transform.rotate 써서 변환 후 그려주기

```
rotated_rock2 = pygame.transform.rotate(
    rock2, rotation_angle
)  # Adjust the rotation angle as needed
rotated_rock2_rect = rotated_rock2.get_rect(
    center=(rockX2 + rockWidth2 / 2, rockY2 + rockHeight2 / 2)
)
gamePad.blit(rotated_rock2, rotated_rock2_rect.topleft)
```

플레이어 HP UI

- HP 이미지 로딩

```
hpbar = pygame.image.load("PS/hp_bar.png")
hpbar_rect = laser.get_rect(topleft=(0, 0))
```

- HP (Health Power) bar 기능 
  운석이 플레이어와 충돌하거나 통과한 경우 10%씩 HP 감소

- HP를 플레이어의 수명에 맞추어 HP 이미지 내부에 Rectangle 채움

```
hpbar_rect.x = int(x) - 3
hpbar_rect.y = int(y + fighterHeight + 1)

drawObject(hpbar, hpbar_rect.x, hpbar_rect.y)

pygame.draw.rect(
    gamePad,
    (100, 230, 100),
    [x + 1, y + fighterHeight + 3, float(shooter_life / 10) * fighterWidth, 5],
    5,
)
```

필살기

- Laser 이미지 로딩

```
laser = pygame.image.load("PS/laser.jpg")
laser_rect = laser.get_rect(topleft=(0, padHeight))
```

- Laser의 y좌표를 위로 올리면서 운석과의 충돌 체크

  필살기 (laser sweeper) : 화면 하단에서 출발 하여 상단으로 올라가면서 운석 충돌 파괴

```
if shoot_laser:  # collision
    drawObject(laser, 0, laser_rect.y)
    special.play()
    laser_rect.y -= 3
    if laser_rect.y < rockY:
        isShot = True
        shotCount += 1
    if level == 2 or level == 3:
        if laser_rect.y < rockY2:
            isShot2 = True
            shotCount += 1
        if laser_rect.y < rockY3:
            isShot3 = True
            shotCount += 1
    if laser_rect.y < 0:
        shoot_laser = False
        laser_rect.y = padHeight
        special.stop()
```

Part 3.2.5

배경 스크롤

• 근거리/원거리 배경 로드(alpha값 설정

```
background = pygame.image.load("PS/space_bg.png")  # 배경 그림
bg_y = 0
background_near = pygame.image.load("PS/near_bg.jpeg")  # 배경 그림
background_near.set_alpha(160)
bgnear_y = 0
```

•근거리/원거리 배경 간 개별 속도 조절

```
drawObject(background, 0, bg_y - padHeight)
drawObject(background, 0, bg_y)
bg_y += 2
if bg_y > padHeight:
    bg_y = 0
drawObject(background_near, 0, bgnear_y - padHeight)
drawObject(background_near, 0, bgnear_y)
bgnear_y += 4
if bgnear_y > padHeight:
    bgnear_y = 0
```

배경 스크롤 기능 – Parallax Scroll 구현

: 개체 별 조작을 통해 시차를 이용한 입체감 구현

Part 3.2.6

레벨 시스템에 따른 기능 구현사항

- 레벨 시스템 개발
  - 초기 변수 설정

```
level = 1
lv = True
lv2 = True
```

  - 레벨 보여주는 function

```
def show_level(level):
    font = pygame.font.Font("PS/font.ttf", 20)
    level_text = font.render("레벨:" + str(level), True, (255, 255, 255))
    gamePad.blit(level_text, (10, 30))  # Adjusted position
```

  - 점수 따라 레벨 증가

```
if lv == True:
    if shotCount >= 10:
        level += 1
        rockSpeed += 1
        lvup.play()
        lv = False
```

```
if lv2 == True:
    if shotCount >= 40:
        level += 1
        rockSpeed += 1
        lvup.play()
        lv2 = False

show_level(level)
```

- 레벨 증가시 미사일 모형과 개수 증가
  - 미사일 위치 조정 후 missileXY.append

```
if level == 2:
    missileSound.play()  # 미사일 사운드 재생
    missileX = x + fighterWidth / 2
    missileY = y - fighterHeight
    missileXY.append([missileX, missileY])
    missileX = x
    missileY = y - fighterHeight
    missileXY.append([missileX, missileY])
```

- 다양한 미사일 모형 그려주기

```
missile = pygame.image.load("PS/m2.png")  # 미사일 그림
missile1 = pygame.image.load("PS/m1.png")
missile2 = pygame.image.load("PS/m3.png")
```

```
if len(missileXY) != 0:
    if level == 1:
        for bx, by in missileXY:
            drawObject(missile2, bx, by)
    if level == 2:
        for bx, by in missileXY:
            drawObject(missile, bx, by)
```

3.3

일지 정지 화면

일시정지(Pause)

- **Pasue 기능**
  P키를 누르면 isPause를 설정하여 True시 looping하면서 pygame.time.delay를 이용 pygame sleep
- **pygame.time.delay() 호출**

```
while isPause:
    for event in pygame.event.get():
        # if event.type in [pygame.QUIT]:  # 게임 프로그램 종료
        #     pygame.quit()
        #     sys.exit()

        if event.type in [pygame.KEYDOWN]:
            if event.key == pygame.K_p:
                isPause = False
    pygame.time.delay(100)
    press_text = font.render("Game Paused", True, (255, 255, 0))
    press_rect = press_text.get_rect(center=(padWidth / 2, padHeight / 2))
    gamePad.blit(press_text, press_rect)
    press_text = font2.render("계속 플레이 하려면 p를 누르세요", True, (255, 255, 255))
    press_rect = press_text.get_rect(center=(padWidth / 2, padHeight / 2 + 60))
    gamePad.blit(press_text, press_rect)
    pygame.display.update()
```

3.4

엔딩 화면

게임 오버화면 이벤트 처리

- **게임 오버 화면 음악 처리**
  게임 오버 사운드 on 배경 음악 정지

```
# 전투기가 운석과 충돌했을 때 메시지 출력
def crash():
    global gamePad, shooter_life
    shooter_life = 10
    pygame.mixer.music.stop()  # 배경 음악 정지
    gameOverSound.play()  # 게임 오버 사운드 재생
```

```
# 게임 오버 메시지 보이기
def gameOver():
    pygame.mixer.music.stop()  # 배경 음악 정지
    gameOverSound.play()  # 게임 오버 사운드 재생
```

- **게임 오버 화면 이벤트 처리**
  ENTER키를 누르면 다시 시작
  Q를 누르면 게임 종료
  G를 누르면 스코어 그래프 (그래프는 그래프 설명 칸에서 다시)

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_RETURN:
                pygame.mixer.music.play(-1)  # 배경 음악 재생
                runGame()
            elif event.key == pygame.K_q:
                pygame.quit()
                sys.exit()
            elif event.key == pygame.K_g:
                draw_score()
```

Part 3.4.2

게임 오버화면 메시지 보이기

- 게임 오버 글씨 위치 조정 및 설정
  색상 조정 , 위치 조정 , 폰트 조정

```
def writeMessage(text):
    global gamePad
    textfont = pygame.font.Font("PS/font.ttf", 105)
    text = textfont.render(text, True, (255, 0, 0))
    textpos = text.get_rect()
    textpos.center = (padWidth / 2, padHeight / 2 - 150)
    gamePad.blit(text, textpos)
    pygame.display.update()
```

```
def writeMessage2(text):
    global gamePad
    textfont = pygame.font.Font("PS/font.ttf", 25)
    text = textfont.render(text, True, (0, 250, 250))
    textpos = text.get_rect()
    textpos.center = (padWidth / 2, padHeight / 2 - 40)
    gamePad.blit(text, textpos)
    pygame.display.update()
```

- 게임 오버 화면에 글씨 적용

```
def crash():
    global gamePad, shooter_life
    shooter_life = 10
    pygame.mixer.music.stop()  # 배경 음악 정지
    gameOverSound.play()  # 게임 오버 사운드 재생
    writeMessage("GAME OVER")
    writeMessage2("다시시작 : ENTER")
    writeMessage3("나가기 : Q")
    scores = shootCount  ######### scores ######## 점수를 scores 리스트에 추가
    show_end_screen(scores)  ######### scores ########
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    pygame.mixer.music.play(-1)  # 배경 음악 재생
                    runGame()
                elif event.key == pygame.K_q:
                    pygame.quit()
                    sys.exit()
                elif event.key == pygame.K_g:
                    draw_score()
```

게임 점수 저장

- 게임 점수를 File(“scores.txt)에 저장하는 함수

```
def save_scores(scores):
    # 현재 게임의 점수를 파일에 저장
    with open("scores.txt", "a") as file:
        file.write(f"{scores}\n")
```

게임 점수 파일 저장 및 최고/이전/현재 점수 표시
: 게임 종류 시 점수를 File에 저장하고 기록

- 파일에서 점수들을 읽어와서 최고점수와 이전 점수를 종료화면 표시 모듈에 전달

```
def read_high_scores():
    try:
        with open("scores.txt", "r") as file:
            all_scores = [int(line.strip()) for line in file.readlines()]
            global end_score, before_score
            end_score = all_scores[-1]
            if len(all_scores) > 1:
                before_score = all_scores[-2]
            else:
                before_score = 0
            return max(all_scores)
    except FileNotFoundError:
        return 0
```

점수 PYPLOT 그래프 연동

- 파일에 저장된 데이터를 읽어 오는 함수

```
def read_scores():
    try:
        with open("scores.txt", "r") as file:
            all_scores = [int(line.strip()) for line in file.readlines()]

            return all_scores
    except FileNotFoundError:
        return []
```

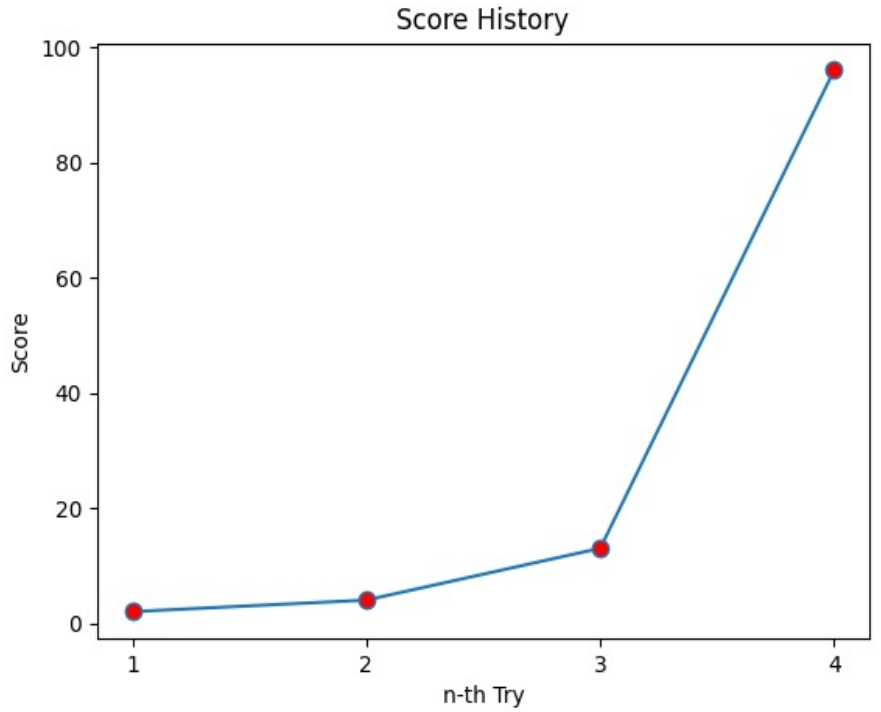
- 점수를 읽어와서 pyplot을 이용하여 그래프를 그려 주는 함수

```
def draw_score():
    y = read_scores()
    # print(y)

    x = [i + 1 for i in range(len(y))]

    plt.plot(x, y, marker=".", markerfacecolor="red", markersize=15)
    plt.xticks(range(1, len(y) + 1))
    plt.xlabel("n-th Try")
    plt.ylabel("Score")
    plt.title("Score History")
    plt.show()
```

- 게임 점수 이력을 읽어 PYPLOT과 연동,그래프 출력



4

개발 결과

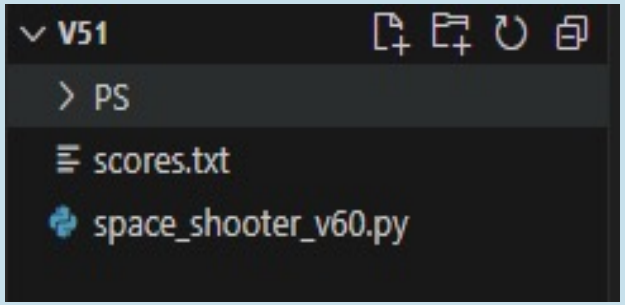
Part 4

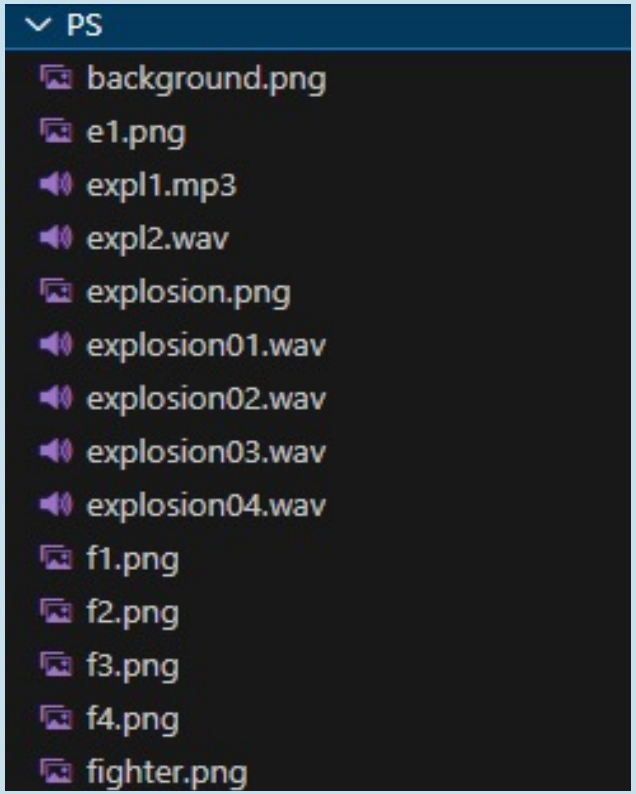
버전 이력

23/11/30	23/12/04	23/12/05	23/12/06	23/12/7
Version 0	Version 10	Version 30	Version 50	Version 60
주요 기능 •원본	주요 기능 •플레이어 상하 이동 •시작화면 •배경 스크롤 •필살기	주요기능 •HP 바 적용 •종료화면 •게임 레벨 적용	주요 기능 •게임 정지 •운석 회전 •폰트 추가 적용 •점수저장 및 표시	주요 기능 •점수 이력 그래프 표시 •날씨 정보 크롤링 •운석/플레이어 간 충돌 조건 개선
주요 문제점 스피커 미 연결 시 프로그램 실패 fail -> 3.5파이 이어폰 연결 후 개발		주요 문제점 레벨 변경 후 미사일 관련 리스트 remove시 crash 발생 -> 미사일리스트 remove전 미사일 개체 포함 여부 체크	주요 문제점 마지막 게임 점수 가 최고 점수 시 부정확한 최고 점수 표시 -> 마지막 점수를 점수 리스트에서 POP 하는 부분 제거	

Part 4.1

FILE TREE 및 실행환경





• Space_shooter_v60.py
소스를 작성한 메인 파일

• PS
게임 리소스 폴더

• Scores.txt
점수 저장 텍스트 파일

•실행을 위한 필요 모듈
- Pygame
- Matplotlib
- bs4

• PS 폴더 내부 파일 예시

Background.png = 배경화면

Expl1.mp3 = 폭파 사운드

F1.png = 플레이어 모델

사용 설명서

- 게임 시작
  SPACEBAR 를 눌러서 게임을 시작합니다

- 방향키
  ←↑↓→ 로 움직입니다.

- 발사
  SPACEBAR를 눌러 발사합니다.

- 필살기
  S를 눌러서 필살기를 사용합니다.

- 일시 정지
  P를 눌러 일시 정지 하고 P를 다시 눌러 재개합니다.

- 점수 이력 그래프 보기
  게임 오버화면에서 G를 눌러 점수 이력 그래프를 출력합니다.

- 게임 나가기
  게임 오버화면에서 Q를 눌러 게임을 종료합니다.

#별첨 회의록

회의록

회 의 록

회의일자	2023 . 11 . 30 (목)	참석자	안승진,임강현,정찬홍,최열호
회의장소	4층 접견실	작성자	정찬홍
제 목	PY-SHOOTING GAME 개발 기획 협의		

1. 슈팅게임 요구사항 협의
   - 스코어/레벨 기능협의
   - 배경화면 구현 협의
   - 시작표시 구현협의
   - 필살기 아이디어 발췌

2. 업무 분장 협의
   - 안승진 팀장 : 전체 업무 총괄 및 주요 모듈 구현
   - 정찬홍 : 점수 관리
   - 임강현 : 슈팅 관련 기능 총괄
   - 최열호 : 시작 / 종료 화면 구현

3. Py game에 대한 이해도 학습 및 기능 업그레이드 아이디어 구상

특이 사항 및 계획	1. PyGame에 대한 업그레이드 구상 협의

회 의 록

회의일자	2023 . 12 . 01 (금)	참석자	안승진,임강현,정찬홍,최열호
회의장소	6층 강의실	작성자	정찬홍
제 목	PY-SHOOTING GAME 개발 구상 협의		

1. 슈팅게임 구상 발표 및 개발 준비 현황 협의
   - 스코어/레벨 최 상위 5위까지 종료 화면에 구현
   - 입체감 있는 배경화면 구현
   - Legend 시작 화면 참조
   - 특정 키로 스톤 전파 장치 필살기 구현

2. 개발 일정 협의
   - WBS 작성
   - 12/4 일 까지 기본 프로그램 작성
   - 기본 프로그램 기능 점검 및 기능별 단위 테스트 준비
   - 폰트 및 장비 시안 검색

특이 사항 및 계획	1. 스피커 미 장착시 게임 실행 불가 2. 각자 개발용 유선 이어폰 준비 3. "mixer, "sound" 등 음향 관련 주석 처리 후 실행

회의록

회 의 록

회의일자	2023 . 12 . 05(화) 14:00	참석자	안승진,임강현,정찬홍,최열호
회의장소	6층 강의실	작성자	정찬홍
제 목	프로그램 1차 통합 테스트 / 수정 및 업그레이드 방안 협의		

1. 프로그램 통합 테스트 및 수정

- 스코어 종료 화면에 구현

;--> 최상위 점수, 현재 점수, 바로 전 점수 보이기로 함

- 원근감으로 입체감 있는 배경화면 구현

;--> 배경 화면과 낙하 속도가 차이 나도록 변경

- Legend 시작 화면 참조

;-->화면 글자 위치 조정

- 특정 키로 스톤 전파 장치 필살기 구현

;--> 필살기에 걸맞을 음향 검색

2. 산출물 협의

- 개발시 산출물 고려 준비

특이 사항 및 계획	1. 팀별 스피커 지급

회 의 록

회의일자	2023 . 12 . 05(화) 14:00	참석자	안승진,임강현,정찬홍,최열호
회의장소	6층 강의실	작성자	정찬홍
제 목	프로그램 1차 통합 테스트 / 수정 및 업그레이드 방안 협의		

1. 프로그램 통합 테스트 및 수정

- 스코어 종료 화면에 구현

;--> 최상위 점수, 현재 점수, 바로 전 점수 보이기로 함

- 원근감으로 입체감 있는 배경화면 구현

;--> 배경 화면과 낙하 속도가 차이 나도록 변경

- Legend 시작 화면 참조

;-->화면 글자 위치 조정

- 특정 키로 스톤 전파 장치 필살기 구현

;--> 필살기에 걸맞을 음향 검색

2. 산출물 협의

- 개발시 산출물 고려 준비

특이 사항 및 계획	1. 팀별 스피커 지급

회의록

회 의 록

회의일자	2023 . 12 . 06(수) 14:00	참석자	안승진,임강현,정찬홍,최열호
회의장소	6층 강의실	작성자	정찬홍
제 목	프로그램 2차 통합 테스트 / 수정 보완 작업 및 산출물 협의		

1. 프로그램 기능 단위별 실행 및 업그래이드

- 스코어 최상위 점수, 현재 점수, 바로 전 점수 종료 화면에 구현

;--> 현재 점수상위일 시 재경기에 최 상위 점수로 보이지 않음(.pop() 변경)

- 시작 화면 ---> 게임 (필살기) ---> 종료, 통합 테스트

;--> 통합 Version 5.1 작성

2. 산출물 협의

- WBS 기준 단계별로 준비된 산출물 정리 방안 협의

;--> 파워포인트로 통합 정리

- 사용자 통합 시스템 매뉴얼 작성 구상

특이 사항 및 계획	