

# SW 개발 계획서

프로젝트명: 파이썬을 이용한 폭탄 피하기 게임

팀명: 메소드 (9 조)

참여 개발자			
학과	학번	이름	역할
컴퓨터전자시스템공학부	201703353	조성준	전반적인 코드 작성
컴퓨터전자시스템공학부	201902763	이전희	전반적인 코드 작성
컴퓨터공학부	202101313	민휘원	계획서와 보고서 작성, 발표
컴퓨터공학부	202102395	이명진	게임 기본값 설정, ppt 제작, 발표
컴퓨터공학부	202103811	황혜진	팀장, 회의록 작성

2022-1 학기 오픈소스 SW 및 실습과목

2022. 05. 04

## 1. 프로젝트 개요

### 1) 주제 선정 동기

- 팀원들이 공통적으로 다룰 수 있는 파이썬을 이용한 프로젝트 실행
- 파이썬을 이용하여 게임을 제작하는 과정에서 GUI프로그래밍에 대해 이해하고, 개인의 개발 능력을 향상 시키기 위해서 게임 프로젝트로 선정
- 폭탄피하기, 체스, 무한의 계단, 갤러그 등의 게임 중에서 폭탄 피하기 게임의 경우, 단순한 알고리즘으로 동작하기 때문에 모든 팀원들이 골고루 참여할 수 있어 해당 게임 선정

## 2. 목표 및 내용

### 1) 최종 목표

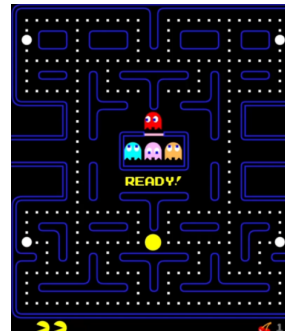
“폭탄 피하기 게임”

- 게임 설계
- 게임 구현
- 캐릭터, 목숨, 폭탄 등 이미지 선정
- pygame라이브러리 분석
- 캐릭터, 폭탄 움직임 구현

### 2) 프로젝트 계획

- 게임 설계

: 모두가 쉽게 이해할 수 있는 게임을 만들기 위해 한 번에 이해할 수 있는 규칙과 간단한 움직임을 설계하는 것이 필요하므로 이와 관련된 자료를 조사해 통해 설계할 예정 (하단의 그림 참고)



- 게임 구현

: 캐릭터는 양옆으로 움직이며 위에서 무작위로 내려오는 폭탄을 피하는 구조이다. 총 3번의 기회가 있으며 폭탄을 3번 피하지 못할 경우 게임은 끝난다. 게임을 하면서 점수는 누적하여 계산한다. 사용자가 게임 화면의 상단에서 이를 실시간으로 확인할 수 있게 구현한다.

- 캐릭터, 목숨, 폭탄 등 이미지 선정

: 캐릭터는 1개로 선정

: 목숨은 하트 모양으로 선정

: 폭탄은 내려오는 이미지와 캐릭터와 부딪혔을 때 나타나는 이미지 다르게 선정

- pygame라이브러리 분석

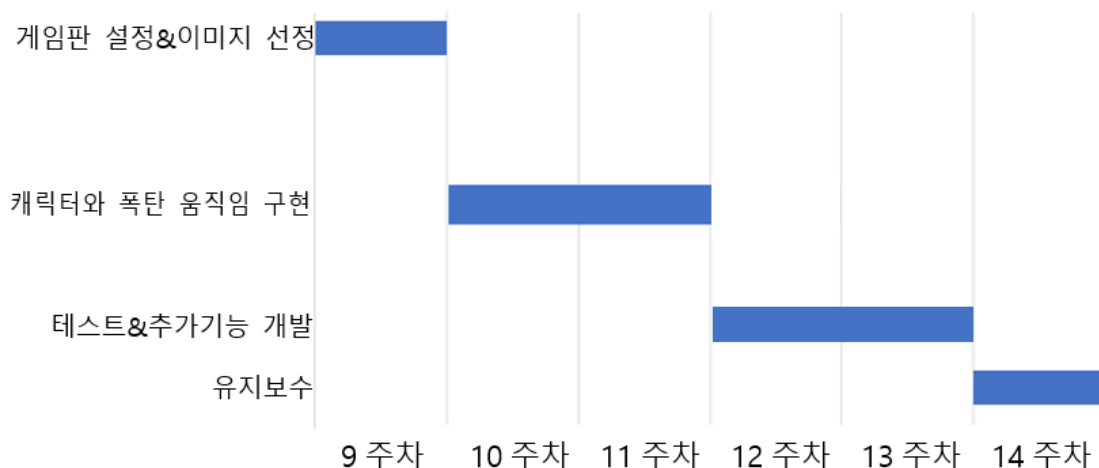
: 게임판 크기 설정 및 이미지 삽입 방법

- 캐릭터, 폭탄 움직임 구현

: 캐릭터는 키보드 방향키 ←, →를 이용해 움직임 구현

: 폭탄은 상단에서부터, 개수와 속도는 무작위로 설정되어 내려오게 구현

### 3. 프로젝트 수행 일정



# SW 개발 최종 보고서

프로젝트명: 파이썬을 : 이용한 폭탄 피하기 게임  
팀명 매소드 : (9 조)

참여 개발자			
학과	학번	이름	역할
컴퓨터전자시스템공학부	201703353	조성준	전반적인 코드 작성
컴퓨터전자시스템공학부	201902763	이전희	전반적인 코드 작성, 이미지 선정
컴퓨터공학부	202101313	민휘원	계획서와 보고서 작성
컴퓨터공학부	202102395	이명진	코딩 보조, ppt 제작, 발표
컴퓨터공학부	202103811	황혜진	팀장, 회의록 작성, 코딩 보조

2022-1 학기 오픈소스 SW 및 실습과목

2022. 05. 04

# 1. 프로젝트 개요

## 1) 프로젝트 소개

팀원 모두가 공통적으로 다룰 수 있는 파이썬을 이용하여 GUI프로그래밍에 대해 이해하고, 개인의 개발 능력을 향상시키며 GIT HUB를 통한 협업을 통해 협동심을 기르며 최종적으로 모두가 쉽게 즐길 수 있는 게임을 구현

## 2. 목표 및 내용

### 1) 최종 목표

“ 폭탄 피하기 게임 ”

- 게임 설계
- 게임 구현
- 캐릭터, 목숨, 폭탄, 아이템 등 이미지 선정
- pygame라이브러리 분석
- 캐릭터, 폭탄 움직임 구현

### 2) 프로젝트의 내용

- 떨어지는 폭탄을 캐릭터는 상하좌우 4방향으로 움직여 피하는 게임 구현
- 게임 진행 시 시간과 아이템에 따라 점수 부여
- 떨어지는 폭탄을 세 번 맞게 되면 게임 종료
- 최종 게임 점수는 누적하여 순위는 매김

### 3) 주요 코드 및 설명

- 폭탄의 움직임 설정

```
def createBomb(width, height):
    posX = random.randint(0, width) # 떨어질 X좌표
    posY = random.randint(50, height) # 떨어질 Y좌표
    rect = pygame.Rect(bomb_image.get_rect())
    rect.top = posY - 200 # 200 위부터 시작
    rect.left = posX
    speed = random.randint(5, 10) # 속도

def bomb_MoveEffect(bomb):
    bomb['rect'].top += bomb['speed']
    sc = (200 - ((bomb['y'] - bomb['rect'].top) if bomb['rect'].top < bomb['y'] else 200))
    bomb['scale'] = (sc, sc)
    bomb['rect'].left = bomb['x'] + ((50 - sc) / 2)
    bomb['shadow'].top = bomb['y'] + ((50 - sc) / 2)
    bomb['shadow'].left = bomb['x'] + ((50 - sc) / 2)
    bomb['rotate'] = (bomb['rotate'] + 1) % 3
    return bomb
```

- 폭탄이 터질 때 효과

```
def getPos():
    res = []
    for exp in explosion:
        res.append({'x': exp['rect'].left, 'y': exp['rect'].top, 'scale': 50, 'hit': False})
    return res
```

```
def run(screen):
    global SCREEN_WIDTH
    global SCREEN_HEIGHT

    for bomb in bombs:
        bomb = bomb_MoveEffect(bomb)
        if bomb['rect'].top > bomb['y']:
            explosion.append(changeExplosion(bomb))
            bombs.remove(bomb)
            bombs.append(createBomb(SCREEN_WIDTH, SCREEN_HEIGHT))

    for bomb in bombs:
        screen.blit(transformImage(bomb_shadow, bomb['scale'], bomb_rotate[bomb['rotate']])), bomb['shadow'])
        screen.blit(transformImage(bomb_image, bomb['scale'], bomb_rotate[bomb['rotate']])), bomb['rect'])
    for exp in explosion:
        screen.blit(explosionImage(exp['cnt']), exp['rect'])
        exp['cnt'] += 1
        if exp['cnt'] >= 15:
            explosion.remove(exp)
```

## - 캐릭터의 움직임과 경계 설정

```
def run(screen):
    global SCREEN_WIDTH, SCREEN_HEIGHT, characterXpos, characterYpos, toX, toY, f, moveIndex, direct
    # 캐릭터 이동 & 프레임맞추기
    if toX != 0 or toY != 0:
        f += 1
        if f % 3 == 0:
            moveIndex += 1
        characterXpos += toX
        characterYpos += toY

    # 경계 설정-가로
    if characterXpos < 0:
        characterXpos = 0
    elif characterXpos > SCREEN_WIDTH - characterWidth:
        characterXpos = SCREEN_WIDTH - characterWidth

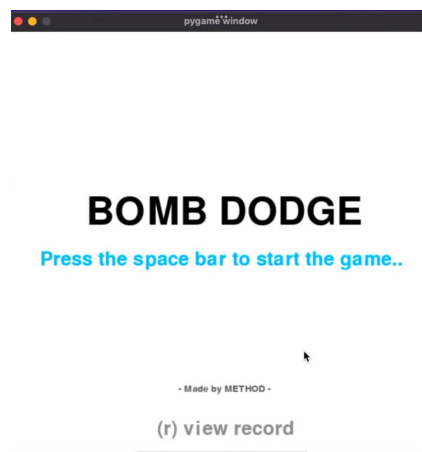
    # 경계 설정-세로
    if characterYpos < 50:
        characterYpos = 50
    elif characterYpos > SCREEN_HEIGHT - characterHeight:
        characterYpos = SCREEN_HEIGHT - characterHeight

    if direct == 1:
        screen.blit(character[moveIndex % len(character)], (characterXpos, characterYpos))
    else:
        screen.blit(pygame.transform.flip(character[moveIndex % len(character)], True, False),
                    (characterXpos, characterYpos))
```

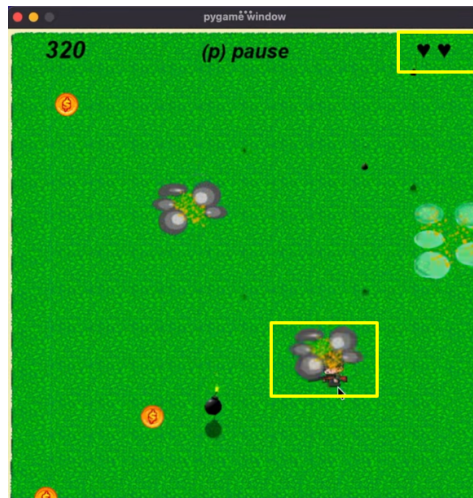
## 3. 결과 및 느낀 점

### 1) 결과 화면 및 설명

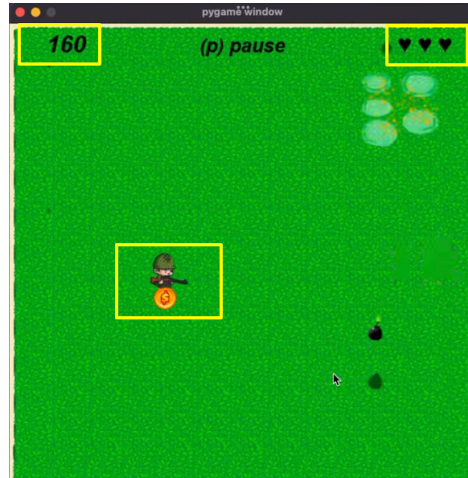
-게임 시작 화면



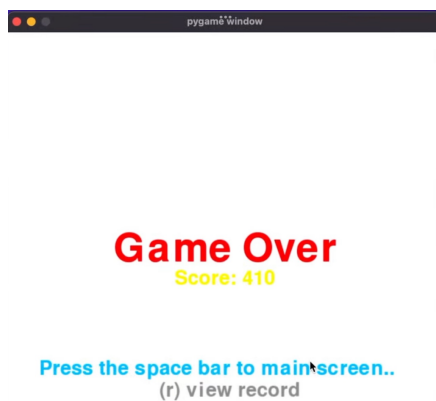
- 캐릭터가 아이템 먹을 때 점수누적



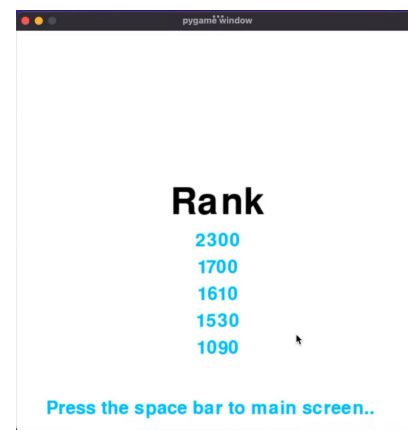
-캐릭터가 폭탄에 맞을 때 목숨 감소



- 게임 종료 화면



-게임 종료 후 점수 누적 화면



## 2) 느낀 점

조성준- 깃허브와 pygame라이브러리를 사용한 프로젝트는 처음이었지만 협업을 통해 어려운 점을 해결해 나갈 수 있었습니다. 깃허브는 실무에서 많이 사용한다고 알고있었다. 이번 프로젝트를 통해 깃허브 사용하는 방식에 익숙해져 좋은 경험이 되었다.

이전희- pygame을 처음 접하여 어려운 점도 많았으나 따로 공부하여 해결할 수 있었고, 각각의 기능들을 구현한 코드들을 세분화 하거나 연결 시키면서 코드가 어떻게 작동하는지에 대해 파악할 수 있었다.

민휘원- 팀원 모두가 이번 프로젝트를 통해 파이썬을 이용한 게임을 만들며 각자의 능력을 향상시켰으며 개개인이 맡은 역할을 훌륭하게 수행하였다. 또한 GIT HUB를 이용한 진행 과정은 개발 진행을 아주 빠르게 진전시키는 데에 효과적이었으며 더 높은 퀄리티의 게임을 만들기 위한 오류 수정 과정 속에서 팀원들과의 적극적인 소통을 통하여 이런 최종 결과물을 만들 수 있어서 뿌듯하였다.

이명진- 프로젝트와 협업을 처음 해보면서 vs code, 깃허브 사용도, 회의하는 과정도 어색했지만 새로운 경험이었다. 또한 협업 시 개인의 역량이 팀에 큰 영향을 준다는 점에서 내 실력을 키워야겠다는 자아성찰의 계기도 되었다. 팀원분들이 많은 부분을 도와주시고 같이 협력하면서 이번 프로젝트를 잘 끝낼 수 있었고 후에 협업에서도 이번기회를 살려 많은 도움이 될 것 같다.

황혜진- 협업을 통해 프로젝트를 진행해 보는 것이 처음이어서 미숙하기도 하고 어색하기도 했던 것 같다. 또한 pygame을 처음 다루어 보아 어려움을 겪기도 하였다. 그럼에도 팀원들이 잘 도와주었고 함께해서 프로젝트를 잘 마무리 할 수 있었던 것 같다. 앞으로 이 경험이 다른 협업 프로젝트를 진행하는데 좋은 바탕이 될 것 같다.