



**KYUNG HEE UNIVERSITY**

---

**[소프트웨어융합개론]**

**Term-Project: 최종보고서**

---

과목	소프트웨어융합개론
전공	글로벌문화기술 융합전공
학번/이름	2020103305 이지수
	2020103310 전수림
	2020103318 한룰

# 목차

<b>1. 프로젝트 내용과 목표</b>	<b>3~4</b>
1) 주제	3
2) 주제 선정 동기	3
3) 게임 타이틀	3
4) 게임 시나리오, 방식, 융합요소	3~4
<b>2. 프로젝트 진행 과정</b>	<b>4~8</b>
1) 주차 별 진행 과정	4
2) 역할 분장 (담당 코드)	5
3) 소통(협력) 방법, 도구, 과정	5~8
<b>3. 프로젝트 결과</b>	<b>8~10</b>
1) 파일 별 정리 기준	8
2) 클래스 및 함수 설명	8~9
3) 사진 삽입 및 간단한 설명	9~10
<b>4. 프로젝트 후 느낀 점</b>	<b>11~12</b>
<b>5. 참고 서적 및 자료</b>	<b>12~13</b>
<b>6. 이미지 및 사운드 Reference</b>	<b>13~15</b>

## 1. 프로젝트의 내용과 목표

1) 주제: Python을 활용한 RPG game 개발하기, IT 기술과 글로벌(일본) 문화 융합시키기

2) 주제 선정 동기

- 게임 개발을 통해 웹/파이썬 프로그래밍 과목에서 학습했던 Python의 문법을 복습하고, 활용 능력을 기른다.
- 일본어학과 강의를 수강하며 배운 일본 문화와 IT 기술을 융합하여 글로벌 문화기술 융합 전공자로서의 역량을 높인다.
- 평소 즐기던 게임을 팀원들과 함께 개발해보며 함께 실무적인 능력을 쌓고, 공동체 속 원활한 커뮤니케이션 능력을 기른다.

3) 게임 타이틀: 야마다상의 마네키네코의 탈환작전 (Rescue Manekineko)

4) 게임 시나리오, 방식, 융합요소:

- 시나리오

게임의 주인공은 라멘 가게의 아르바이트생인 야마다상이다. 어느 날 야마다상이 잠시 졸고 있던 사이에 가게에 있던 마네키네코를 옆 가게에서 보낸 몬스터로부터 도난당한다. 행운을 가져다주는 마네키네코가 사라지자 가게를 찾아오는 손님의 발길이 뚝 끊긴다. 근심에 빠진 라멘 가게 사장님을 위해 야마다상은 마네키네코를 되찾아오기로 하고 여정을 떠난다. 야마다상은 가는 길마다 등장하는 몬스터들을 해치우고 결국 마네키네코를 되찾게 된다. 이후 라멘 가게는 다시 손님들로 북적이기 시작하고, 사장님도 야마다상도 웃음을 되찾는다.

- 게임 방식

플레이어가 3개의 목숨을 가지고 3개의 스테이지에 걸쳐, 등장하는 몬스터들을 공격하여 모두 처리하는 방식이다. 플레이어는 방향키와 스페이스 바로 양 옆 이동, 점프, 공격을 할 수 있다.

- 일본 문화 융합 요소

• 마네키네코:

일본에서 마네키네코는 길조를 부르는 물건의 일종으로, 주로 상가 등에 장식해서 번창을 기원한다.

게임 시나리오에서 알 수 있듯이, 플레이어가 몬스터와 싸우는 이유는 잃어버린 라멘 집의 마네키네 코를 찾기 위해서이다.

- 몬스터(닌자 보이 & 닌자 걸):

닌자는 과거 일본에서 침투 전술과 암살을 일삼았던 특수 전투 집단이다. 몬스터로 닌자들을 등장시켰다.

- 후지산:

일본의 혼슈에 있는 후지산을 배경으로 한 첫 번째 스테이지에서 몬스터 '닌자 보이'가 걷기, 뛰기, 공격하기 형태로 등장한다.

- 오사카성:

일본의 오사카에 있는 오사카 성을 배경으로 한 두 번째 스테이지에서 몬스터 '닌자 걸'이 걷기, 공격하기, 날아다니기 형태로 등장한다.

- 일본 유명 관광지 종합을 배경으로 한 세 번째 스테이지에서 몬스터 '닌자 맨'이 총을 들고 다니는 부하들과 자이언트 닌자가 함께 등장한다.

## 2. 프로젝트 진행 과정

### 1) 주차 별 진행 과정

주차	키워드	계획
1	Case Study 공유	프로젝트 관련 case study 공유 및 전체적인 컨셉 잡기
2	툴 설치	개발 환경 구축 (pygame 설치 및 깃허브 연동)
3	개발 및 피드백	기본 초기 설정, 이미지 로딩, 방향키 설정
4	개발 및 피드백	등장 캐릭터 이동 범위 설정, 적 등장, 주인공 공격
5	개발 및 피드백	적 여러 명 등장, health bar 설정
6	개발 및 피드백	등장 캐릭터 충돌 범위 설정, 사운드 삽입
7	개발 및 피드백	적 여러 명 등장, 난이도 설정, 무적 설정, 목숨 표현
8	개발 및 피드백	최적화 및 단순화, 스테이지 연결시키기, 오프닝과 엔딩,
9	코드 리뷰	각종 디버깅, 시나리오 제작
10	최종 점검	최종 디버깅 및 데모 영상 제작

## 2) 역할 분장 (담당 코드)

이지수	플레이어의 몬스터 공격 적용 플레이어와 몬스터 충돌 범위 설정, 몬스터 여러 명 등장시키기 스테이지 시작 전 이미지 제작 및 설정 시나리오 이미지 제작 및 설정 플레이어 목숨 이미지 제작 및 설정 코드 단순화 최적화
전수림	전체적인 키 설정 및 캐릭터 이동 몬스터 수명(health bar) 적용 사운드 삽입 설정 몬스터 충돌 시 플레이어 무적 설정 스테이지 조합 게임 오프닝, 엔딩 설정 코드 단순화 최적화
한률	이미지 파일, 크기, 편집 이미지 로딩 및 게임 초기 설정 몬스터 여러 명 등장시키기 다양한 종류의 몬스터 등장시키기 플레이어 위치 고정 설정 단계별 게임 난이도 조절

## 3) 소통(협력) 방법, 도구, 과정

### - 도구 및 방법

- 커뮤니케이션 도구: 카카오톡, Zoom, GitHub, Google 공유 드라이브

카카오톡을 통해 간단하거나 급한 내용을 매일 빠짐없이 공유했으며, zoom으로 매주 2회 이상 온라인

회의를 진행했다. 또한, 매번 회의 내용은 각자 정리하여 GitHub의 Project Plan과 Code plan에 업로드 하여 팀원 모두가 프로젝트의 진행 과정을 파악할 수 있도록 했다.

프로젝트 시작 초반부(1~2주차)에는 필요한 이미지 자료를 조사하여 Google 공유 드라이브에 업로드하였고, 코드를 개발하면서 각자 필요한 자료들을 해당 드라이브로부터 내려 받는 형식으로 진행했다. 팀원 간 코드 및 프로젝트 과제물 공유는 Visual Studio Code와 GitHub를 연동하여 진행했다. 각자의 코드 파일을 push하면 서로 확인 후 수정하고, 이것을 다시 공유하는 과정을 거치며 팀프로젝트를 진행했다.

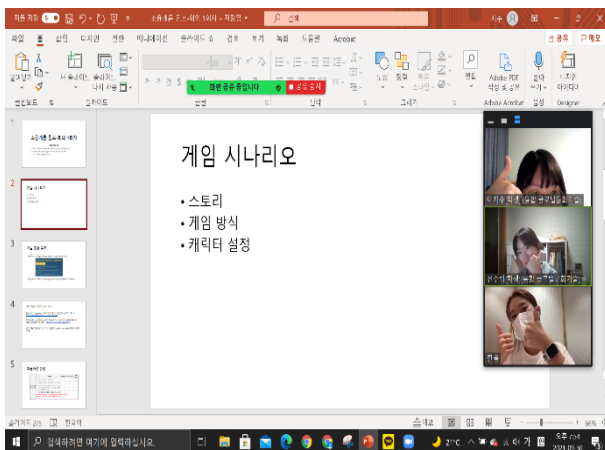
- 협업을 위한 GitHub Repository 주소

[https://github.com/maria1746-s0oramy-hanryoul1/TermProject\\_RPGgame](https://github.com/maria1746-s0oramy-hanryoul1/TermProject_RPGgame)

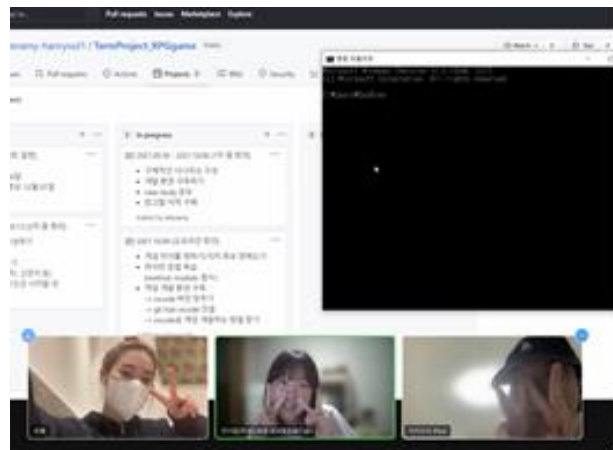
## - 소통 과정

- 온라인 회의

Zoom을 활용한 온라인 회의는 9월 30일을 시작으로, 11월 27일까지 총 15회 진행했다. 주 1~2회 온라인 회의를 진행하여 한 주간 각자의 코드 진행 상황을 나누며, 카카오톡으로 주고받은 피드백으로는 해결되지 않는 부분을 함께 해결해 나갔다. 매 온라인 회의를 시작하기에 앞서, 회의를 통해 정해야 하는 사항들을 미리 정리함으로써 회의시간을 효율적으로 사용하였고, 카카오톡을 통해서 미흡했던 전달 사항을 정리한 후 팀원들에게 다시 전달했다.



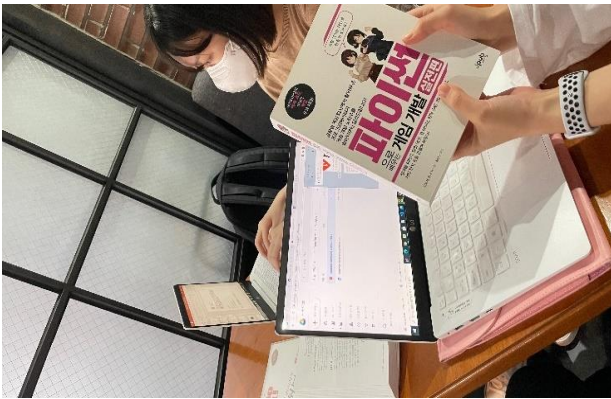
<9월 30일> 첫 온라인 회의



<10월 20일> 온라인 회의

- 오프라인 회의

코로나 사태를 감안하여, 오프라인 회의는 10월 8일부터 12월 2일까지 총 4회 진행했다. 오프라인 회의는 온라인 회의로는 해결하기 어려운 내용을 논의해야 할 때 이루어졌다. 오프라인 회의 초반에는 pygame에 대한 서적을 구매하고 게임에 로딩 할 최종 이미지 선택 등 오프라인으로 만나야만 하는 회의를 했다. 그 이후에는 개발 과정에서 담당한 코드의 오류가 해결되지 않아 집단지성이 필요하거나 구체적인 진행 방향에 긴 회의 필요한 경우 오프라인 회의를 진행했다.



<10월 8일> 첫 오프라인 회의

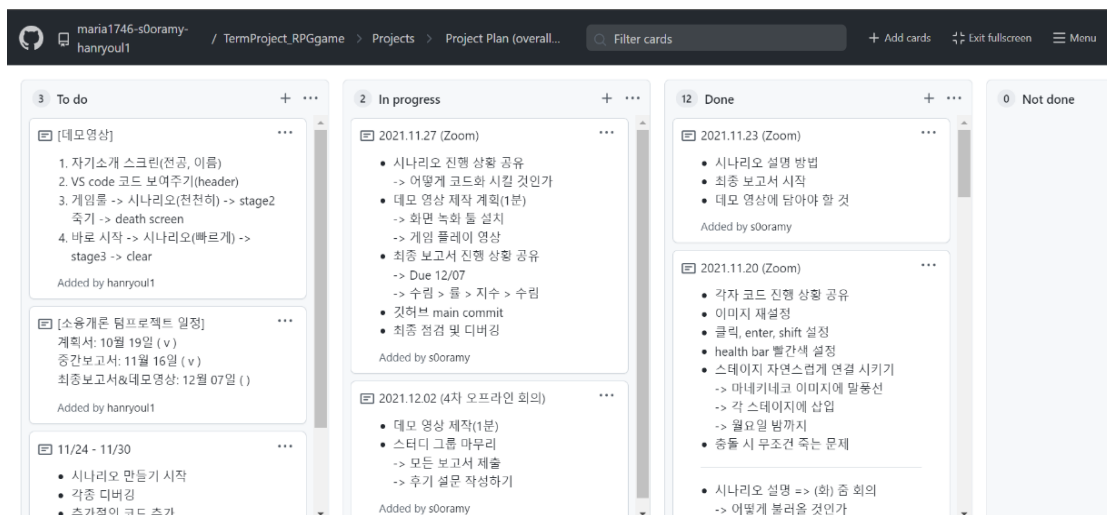


<11월 10일> 오프라인 회의

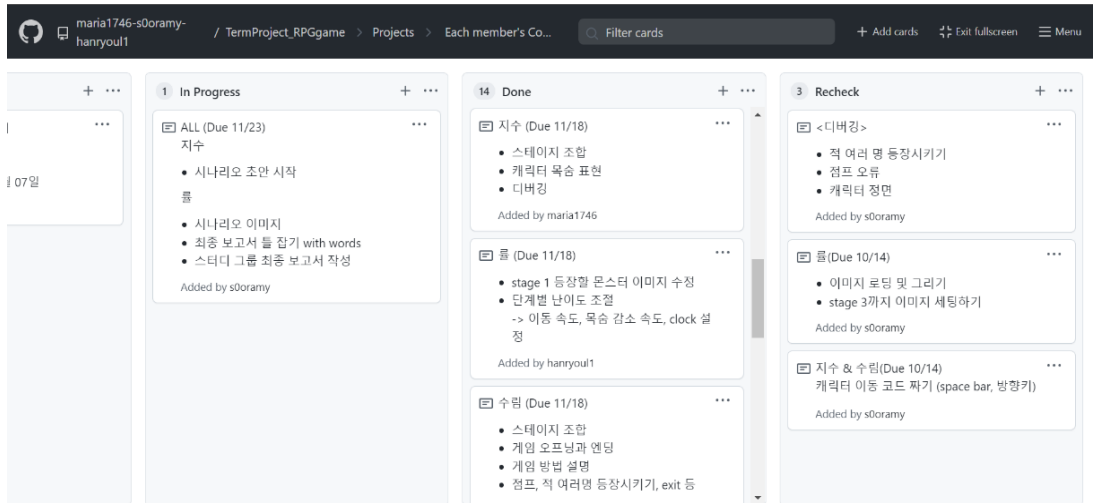
### • Github Project 활용

Github은 각자 수정하고 작성한 코드에 대한 푸시 및 풀 뿐만 아니라 Project를 이용하여 'Project Plan (overall version)'과 'Each member's Code Plan'이라는 프로젝트를 두 가지 생성하여 활용했다. 'Project Plan (overall version)'에는 해당 주차에 이루어지는 Zoom 회의에서 상의해야 할 내용에 관해 서술했으며, 전반적인 프로젝트 진행사항을 업로드하여 팀원 간 원활히 공유했다.

'Each member's Code Plan'는 일정 기간까지 개인이 완성시켜야 하는 코드의 내용을 직접 올리는 용도로 활용했다. 이를 통해 매번 서로 물어보지 않아도, 누가 어떤 코드를 작성했는지 알 수 있게 되었다. 결과적으로 텀프로젝트 진행 중, Github의 Project 기능을 활용하여 원활한 팀 작업이 이루어졌다.



< Project Plan (overall version) >



< Each member's Code Plan >

### 3. 프로젝트 결과

#### 1) 파일 별 정리 기준

게임에 대한 코드를 기능에 따라 5개의 파일로 나누었다.

- **main.py**: 이 코드 파일은 게임을 실행하는 파일로, main loop이 존재한다. 또한 header.py, stage1.py, stage2.py, stage3.py 파일을 모두 import한다. 파일 내에는 이 밖에 스타트 메뉴, 플레이 방법 설명, 미션 성공과 실패 화면, 시나리오 등을 실행하기 위한 각각의 함수를 작성했다.

- **header.py**: 게임에 필요한 캐릭터들(주인공, 다양한 모습의 닌자들)을 모두 class()로 만들어 필요한 스테이지에서 호출할 수 있도록 했다. 주인공이 닌자를 공격하기 위해 필요한 코드도 Attack class()로 만들어 header.py 파일 내에 추가했다.

- **stage1,2,3.py**: 각 스테이지별로 필요한 이미지와 사운드를 로드 했으며, 스테이지 별로 난이도 조절을 위한 코드를 작성했다. 각 stage.py 파일에서는 스테이지별로 필요한 이미지를 불러오고, 주인공과 닌자가 충돌할 경우, 닌자가 공격을 당했을 경우 등과 같은 상황에 필요한 코드를 작성했다.

#### 2) 클래스 및 함수 설명

코드를 최적화하기 위해서 1,2,3 단계에서 필요한 주인공, 닌자, 공격 등을 class()로 만들어 header.py 내에 작성했다. 각 스테이지별로 필요한 class()를 불러오는 방식으로 stage.py 파일을 작성했다. 다음은 header.py 파일 내에 작성한 주요한 class()들에 대한 간단한 설명이다.



- **class Player:** 플레이어 클래스 내에는 다음과 같은 함수를 작성했다.

<code>def __init__(self,x,y,width,height)</code>	플레이어 초기화를 위한 함수
<code>def draw(self, screen)</code>	화면 내 플레이어 이동을 위한 함수
<code>def hit(self)</code>	몬스터와 충돌할 경우 플레이어를 3초 동안 무적으로 만들어주는 함수
<code>def real_hit(self)</code>	몬스터와 충돌할 경우 플레이어가 목숨을 1개 잃는 함수

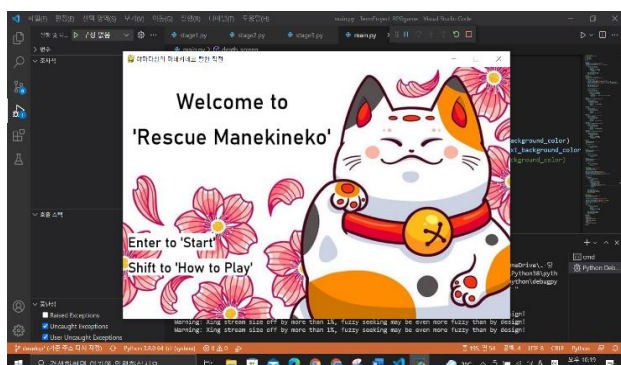
- **class Attack:** 어택 클래스 내에는 다음과 같은 함수를 작성했다.

<code>def __init__(self,x,y,radius,color,facing)</code>	플레이어가 몬스터를 공격하기 위해 필요한 총알을 초기화하기 위한 함수
<code>def draw(self,screen)</code>	화면 내에 플레이어가 발사한 총알을 출력하기 위한 함수

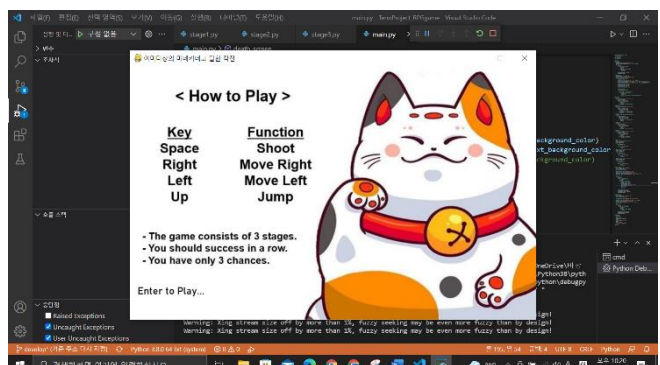
- **class Monster:** 몬스터 클래스 내에는 다음과 같은 함수를 작성했다. (몬스터 클래스의 경우 각 단계별로 등장하는 몬스터마다 난이도 조절을 위해 코드에 약간의 차이가 있다)

<code>def __init__(self, x, y, width, height, end)</code>	몬스터 초기화를 위한 함수
<code>def draw(self,screen)</code>	화면 내 몬스터 이동을 위한 함수
<code>def move(self)</code>	몬스터 이동 범위 설정을 위한 함수
<code>def hit(self)</code>	몬스터가 플레이어에게 공격받을 경우 몬스터의 hp가 감소하는 함수

### 3) 사진 삽입 및 간단한 설명



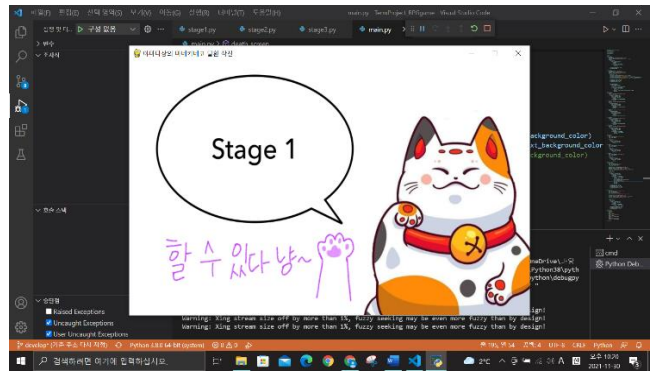
< 게임 시작 화면 >



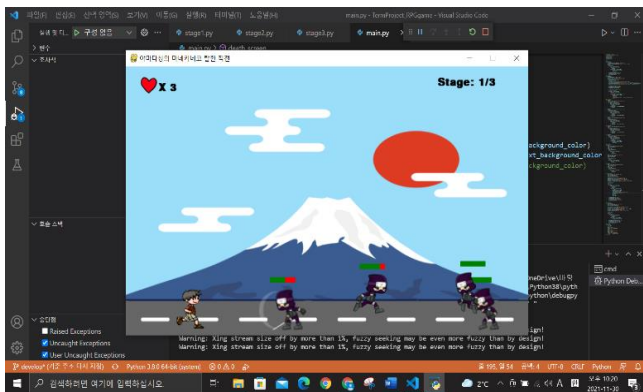
< 게임 방식 설명 화면 >



< 시나리오 1번째 화면 >



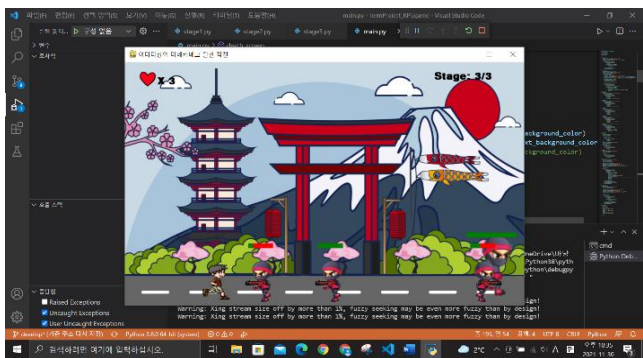
< 1단계 시작 전 화면 >



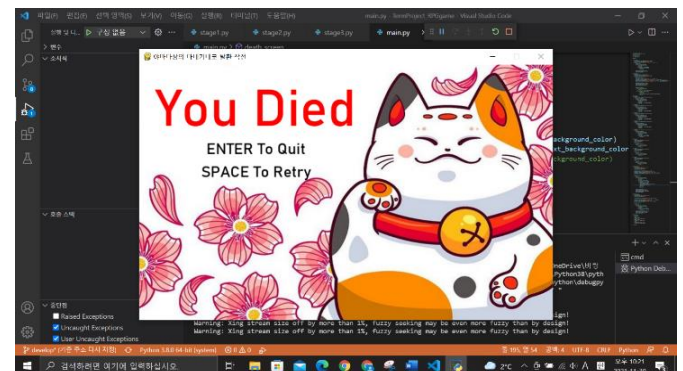
< 1단계 게임 화면 >



< 1단계 게임 화면 >



< 3단계 게임 화면 >



< 3개의 목숨을 모두 소진한 실패 화면 >



< 3단계 클리어 이후 게임 성공 화면 >

#### 4. 프로젝트 후 느낀 점

- 2020103305 이지수

: 텀프로젝트를 처음 시작할 때, 게임을 개발하면서 오류가 발생하지 않았으면 좋겠다고 생각했다. 하지만 프로젝트를 진행할수록, 게임 개발 과정에서 수많은 오류가 발생했으며, 이를 통해 개발 과정과 오류는 떼어 놓을 수 없는 관계라는 것을 알게 되었다. 이제는 게임 개발 과정에서 발생한 오류들을 해결하면서 코드를 논리적으로 생각할 수 있는 힘을 기를 수 있었기 때문에 더 성장할 수 있었다고 생각한다.

프로젝트를 진행하기 전 까지만 해도 나는 그저 게임을 즐기는 사람이었다. 하지만 프로젝트를 진행하면서 게임 하나를 만들기 위해서 큰 노력이 들어간다는 점을 느낄 수 있었다. 나 혼자서 게임 개발 프로젝트를 수행했다면, 제대로 프로젝트를 완성할 수 없을지도 모른다. 하지만, 팀원들과 함께 서로 응원하고, 이끌어 주며 프로젝트를 진행한 덕분에 좋은 결과물을 만들어낼 수 있었다고 생각한다. 또한, 전수림 학우가 제안한 방법으로 GitHub와 VSC를 연동한 덕분에 수월하게 협업을 진행할 수 있었다. 이 방법은 앞으로 다른 프로젝트를 진행하는 경우에도 유용하게 사용할 수 있을 것 같다고 느꼈다.

- 2020103310 전수림

: 이번 프로젝트를 통해 긴 코드를 다루면서 코드를 최적화하고 단순화하는 작업의 중요성을 깨닫게 되었다. 함수 및 class 설정, 명확한 변수 이름 설정을 통해 누구나 쉽게 읽고 이해하도록 작업했다. 이 작업 이후 코드 기능 또는 역할에 따라 파일을 분리하고 main에서 불러오도록 하였다. 특히, 파일 분리 이후에는 오류 검출, 수정, 추가적인 코드 삽입의 속도도 빨라지는 것을 느낄 수 있었다.

또한 게임이라는 분야는 다양한 분야의 작업이 이루어져야 한다는 것을 느꼈다. 단순히 게임 코드를 작성하는 것뿐만 아니라 이미지, 디자인, 시나리오, 사운드 등의 요소가 필요하다. 이는 게임 유저들이 더 몰입하여 흥미를 느낄 수 있는 게임이 되게 해준다. 직접 게임을 개발해 보면서, 게임 산업은 융합적인 분야이며 역할 분담이 중요하다는 것을 알게 되었다. 이번 프로젝트에서 팀원들과 적절하게 분업하고 회의를 하였기에 모두가 만족하는 게임을 만들 수 있었다고 생각한다.

- 2020103318 한률

: 이번 텀프로젝트의 주제로 파이게임을 개발하면서, 코드를 섬세하게 다루는 경험을 해보았다. 구체적으로, 게임의 각 스테이지마다 해당하는 캐릭터의 이미지를 적절한 때와 위치에 등장하게 하는 코드를 작성해보았다. 이때, 내가 지정하는 XY 좌표의 변수에 따라 캐릭터가 갑자기 사라져버리거나 게임이 진행되지 않는 모습들을 보면서 다양한 고민을 통해 해결하는 과정을 거칠 수 있었다. 또한, 게임에 적합한 디자인적 요소를 조사하고 결정하는 작업을 통해 사용자의 요구를 만족시키는 프로그램을 만들기 위해서는 디자인의 역할이 중요하다는 것을 느낄 수 있었다. 따라서, 단순히 잘 돌아가는 코드의 작성뿐만 아니라 사용자로 하여금 사용하고 싶게 만드는 프로그램의 개발자가 되기 위해 노력해야겠다.

사실 소프트웨어를 공부하면서 좋은 개발자가 되기에는 나의 상상력이 많이 부족하다는 생각을 많이 했었고, 이 때문에 개발에 대한 자신감이 떨어졌었다. 그런데, 막상 더 좋은 게임을 만들기 위해 다양한 아이디어를 떠올리다 보니 생각보다 나에게는 많은 잠재력이 있다는 것을 알게 되었고, 동시에 자신감 역시 회복할 수 있었다. 또한, 각자 개성을 가진 팀원들과 함께 프로젝트를 진행하며 잘 해결되지 않는 부분은 서로 피드백과 지원을 통해 원활하게 해결할 수 있었고, 프로그램 개발에서 협업 활동의 중요성을 느끼게 되었다. 이번 팀프로젝트는 자기 계발면에서도, 다양한 사람과의 협업면에서도 값진 경험을 안겨 준 활동이라고 생각한다.

## 5. 참고 서적 및 자료

게임 개발 과정에서 일부 코드는 구입한 서적, 유튜브, 다른 깃허브 등을 참고하여 활용하였다.

### 1) 서적

- 히로세 츠요시, 「파이썬으로 배우는 게임 개발 입문편」, 김연수, 제이펍, 2020

- 적 등장시키기 p. 92-97
- 실시간 처리 구현하기 p. 140~153
- Pygame key 입력 p. 244~248
- Key 입력에 따라 이미지 움직이기 p. 147~149
- 주인공 이동 부분 p. 275

- 히로세 츠요시, 「파이썬으로 배우는 게임 개발 실전편」, 김연수, 제이펍, 2020

- 캐릭터 이동하기 p. 68~73
- 플레이어 움직이기 p. 20~211
- 여러 스테이지 조합하기 p. 112, 150
- 엔딩 만들기 p. 98, 139

### 2) 기타

- 유튜브 Tech With Tim의 Pygame Tutorial 영상 (Pygame Tutorial #1~#10)

이 튜토리얼을 통해 pygame의 기초적인 활용 방법에 대해 학습하고, 이를 프로젝트에 활용

[https://www.youtube.com/watch?v=UdsNBIZsmII&list=PLW\\_M3kGkfUvEuO5AfcokphY-d1y3XAkV](https://www.youtube.com/watch?v=UdsNBIZsmII&list=PLW_M3kGkfUvEuO5AfcokphY-d1y3XAkV)

- 깃허브를 통해 다른 사람들의 소스 코드 참고

- 게임 오프닝과 엔딩 [https://github.com/fethidogan/Washing\\_Machine\\_Thrower.git](https://github.com/fethidogan/Washing_Machine_Thrower.git)
- 사운드 설정 <https://github.com/PrintedLove/Python-pygame-RPG-tutorial.git>

## 6. 이미지 및 사운드 Reference

### 1) 게임 플레이 이미지

- Background

- 오사카성: "notefolio". 2021/10/15 접속.  
[https://notefolio.net/choco\\_sec/106205](https://notefolio.net/choco_sec/106205)
- 후지산: "urbanbrush". 2021/10/15 접속.  
<https://www.urbanbrush.net/web/wp-content/uploads/edd/2018/09/urbanbrush-20180927060821306154.png>
- 일본 관광지 종합: "Pinterset". 2021/10/15 접속.  
<https://www.pinterest.cl/pin/817333032359889028/>

- Main character

- 야마다상: "KindPNG". 2021/10/18 접속.  
[https://www.kindpng.com/imgv/iThmiTT\\_video-game-character-sprite-sheet-hd-png-download/](https://www.kindpng.com/imgv/iThmiTT_video-game-character-sprite-sheet-hd-png-download/)
- 마네키네코: "pngtree". 2021/10/09 접속.  
[https://kor.pngtree.com/freepng/cute-luck-sitting-happy-japanese-cartoon-lucky-cat\\_5742671.html](https://kor.pngtree.com/freepng/cute-luck-sitting-happy-japanese-cartoon-lucky-cat_5742671.html)

- Monster

- NINJA GIRL: "Game Art 2D". 2021/10/18 접속.  
<https://www.gameart2d.com/ninja-girl---free-sprites.html>
- NINJA ADVENTURE: "Game Art 2D". 2021/10/18 접속.  
<https://www.gameart2d.com/inja-adventure---free-sprites.html>
- NINJA: "gameartpartners". 2021/11/15 접속.  
<https://gameartpartners.com/downloads/ninja-character-set/>

## 2) 시나리오 이미지

### - 배경 및 자막

- 자막: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2014/10/blog-post\\_656.html](https://www.irasutoya.com/2014/10/blog-post_656.html)
- 배경: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2019/05/blog-post\\_64.html](https://www.irasutoya.com/2019/05/blog-post_64.html)

### - 인물

- 사장님 정면: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2015/12/blog-post\\_152.html](https://www.irasutoya.com/2015/12/blog-post_152.html)
- 기쁜 사장님: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2016/01/blog-post\\_96.html](https://www.irasutoya.com/2016/01/blog-post_96.html)
- 당황한 사장님: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2014/08/blog-post\\_201.html](https://www.irasutoya.com/2014/08/blog-post_201.html)
- 속상한 사장님: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2016/10/blog-post\\_82.html](https://www.irasutoya.com/2016/10/blog-post_82.html)

### - 기타

- 라멘 가게 간판: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2013/07/blogpost\\_7219.html](https://www.irasutoya.com/2013/07/blogpost_7219.html)
- 짬뽕 가게: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2020/09/blog-post\\_684.html](https://www.irasutoya.com/2020/09/blog-post_684.html)
- 빈 가게: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2020/09/blog-post\\_417.html](https://www.irasutoya.com/2020/09/blog-post_417.html)
- 낙엽: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2021/06/blog-post\\_21.html?m=1](https://www.irasutoya.com/2021/06/blog-post_21.html?m=1)
- 라면: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2016/07/blog-post\\_42.html](https://www.irasutoya.com/2016/07/blog-post_42.html)
- 포스기: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2013/06/blog-post\\_292.html](https://www.irasutoya.com/2013/06/blog-post_292.html)
- 카운터 측면: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2017/04/blog-post\\_542.html](https://www.irasutoya.com/2017/04/blog-post_542.html)
- 카운터 정면: "irasutoya". 2021.11.26 접속. [https://www.irasutoya.com/2017/04/blog-post\\_476.html](https://www.irasutoya.com/2017/04/blog-post_476.html)

## 3) 사운드

- background.wav: 2021/10/30 접속.

<https://github.com/PrintedLove/Python-pygame-RPG-tutorial/blob/main/Python-pygame-RPG-tutorial/sound/background.wav>

- bullet.mp3: 2021/10/30 접속.

<https://github.com/techwithtim/Pygame-Tutorials/blob/master/Game/bullet.mp3>

- hit.mp3: 2021/10/30 접속.

<https://github.com/techwithtim/Pygame-Tutorials/blob/master/Game/hit.mp3>

- death.mp3: 2021/11/03 접속.

[https://github.com/fethidogan/Washing\\_Machine\\_Thrower/blob/main/death\\_theme.mp3](https://github.com/fethidogan/Washing_Machine_Thrower/blob/main/death_theme.mp3)

- gameclear.wav: 2021/11/15 접속.

[https://drive.google.com/file/d/1LjZ0GAQ\\_H5uOLS3BHSnKPRXjf-MBsOq6/view](https://drive.google.com/file/d/1LjZ0GAQ_H5uOLS3BHSnKPRXjf-MBsOq6/view)