

# PROJECT REPORT



과목명	임베디드 SW
학 과	소프트웨어학과
학 번	2019125051
이 름	임 세 현

# 목 차

## 1.게임 소개

(YouTube 링크)

## 2.플레이 방법

## 3.소스코드 구조 및 동작 설명

## 4.개발 이력

## 5.주요 이슈 및 해결책

## 6.배운점

## 1. 게임 소개

게임명 : “뽕야뽕야”

YouTube 링크

<https://youtu.be/6VktgFDY8Bk>

모바일 게임 ‘탕탕특공대’에서 아이디어를 얻어 제작하게 되었다.

게임 제목은 ‘뽕야뽕야’라고 지었는데, 이는 사용자의 캐릭터가 끊임없이 총알을 발사하는 데서 떠올렸는데, 기억하기 쉽고 이 게임이 어떤 게임인지 사용자가 이름만 봐도 곧바로 예측할 수 있을 것이라 생각해서 확정 짓게 되었다.

게임은 단순하다. 사방에서 무작위로 튀어나오는 적들을 총으로 쏘거나 스킬을 활용하여 처치해서 일정 점수를 쌓으면 이길 수 있다.

## 2. 플레이 방법

게임을 시작하면 게임을 바로 시작할지, 게임 방법을 확인할지 버튼을 통해 고를 수 있다. A버튼을 누르면 바로 시작, B버튼을 누르면 게임방법을 확인한다.

게임을 시작하면 화면 가장자리 무작위의 위치에서 몬스터들이 나온다. 사용자는 조이스틱을 이용하여 본인의 캐릭터를 움직이면서, 캐릭터가 바라보는 방향으로 발사되는 총알을 이용하여 몬스터를 해치워야 한다. 중간중간 보스 몬스터가 나오는데, 이들은 체력이 더 많기 때문에 많이 때려야 한다. 특정 조건을 충족하면 버튼을 통해 스킬을 사용할 수 있다. A버튼을 누르면 작은 몬스터들을 한번에 죽일 수 있고, B버튼을 누르면 사용자의 캐릭터가 일시적으로 무적이 되어 몬스터와 부딪혀도 데미지를 입지 않고 적을 죽일 수 있다. 위급한 상황에 스킬을 알맞게 활용하여 보스를 10마리 죽여 10점을 달성하면 게임을 승리하게 된다.

게임을 승리하거나 아쉽게 패배했을 때는 A버튼을 누르면 게임을 재시작 할 수 있다.

### 3. 소스코드 구조 및 동작 설명

#### • Main.py 구조

- 필요한 모듈을 Import하고, 이를 통해 필요한 객체들을 생성한다.
- Import한 모듈을 통해 필요한 객체들을 생성한다
- Main()은 크게 시작화면, 게임, 종료화면으로 나뉜다.
- 시작화면에서는 사용자의 선택에 따라 게임 방법을 읽거나 바로 시작할 수 있다.
- 시작화면이 종료되면, 게임 중 필요한 요소들을 담아낼 리스트들과 변수들을 선언한 뒤 게임이 시작된다.
- 게임이 시작되면 조이스틱의 방향에 따라 command가 결정되고, A와 B버튼이 눌리는지 확인한다.
- 버튼이 눌렸다면 스킬 효과가 발동된다.
- 이어서 enemy와 bullet, boss, meat가 조건에 맞게 생성된다.
- 총알이 적과 맞았는지를 확인하여 각각의 객체마다 상태를 최신화 해준다.
- 사용자가 고기를 먹었는지도 확인한 후 마지막으로 사용자의 상태도 최신화 해 준다.
- 모든 작업이 끝나면 화면에 그림을 그려준다.
- 그림은 배경 -> user, enemy, boss, bullet, meat, popup -> 스킬 게이지 바, 스킬 이펙트 순서로 그린다.
- 게임에 승리하거나 패배할 시, 게임종료 화면을 그린다.
- 승패에 따라 다른 문구와 함께 사용자의 체력, 점수가 표시된다.
- A버튼을 누르면 게임을 재시작 할 수 있다.

• Character.py

1) Character class를 통해 user 객체를 만든다.

2)

Appearance : 일반 모습, hit(분홍색), strong(무지개 색) 3가지가 있다. 초기값은 일반 모습이다.

State : alive, hit, strong, dead 4가지 상태가 있다.

Health, speed : 체력과 스피드를 나타낸다.

Moving\_gauge : 스킬B를 발동시키는 조건이다.

Position, center : 객체의 위치정보와 중앙값을 나타낸다.

Hit : 일정 시간 hit state임을 확인하기 위한 변수이다.

3) 함수

- Move(self, command)

Command에 따라 캐릭터를 옮긴 뒤 center값을 갱신해 준다. 화면 밖으로는 움직일 수 없다.

- Live\_check(self)

캐릭터의 health를 확인해서 0이라면 state를 'dead'로 변경한다.

- Hitcheck(self)

캐릭터의 hit값에 따라 state를 설정하고 hit값도 변경해 준다. Hit가 1에서 0이 되었다면 다시 alive로 state를 바꾸어 준다.

- Set\_image(self)

캐릭터의 state값에 따라 어떤 이미지를 띄울 것인지 결정한다. strong이라면 무지개 그림, hit라면 분홍그림, 그렇지 않다면 일반 그림으로 설정한다.

• Enemy.py, Boss.py

1) 각 class를 통해 적들을 생산한다.

2) State : alive, dead 2가지가 있다.

3) 함수

- Move(self, usercenter)

User.center를 인자로 받아 user가 있는 방향으로 움직이고, 만약 체력이 0이라면 state를 'dead'로 바꿔준다.

- Check(self)

health를 확인해서 0이라면 state를 'dead'로 변경한다.

- Kiss(self, user)

본인과 user의 position을 overlap함수를 통해 서로 겹치는지 확인한다. 겹쳤을 때 본인의 체력을 0으로 하고, user의 state가 strong이 아니라면 user의 체력을 깎는다.

- Overlap(self, this\_position, object\_position)

본인의 상, 하, 좌, 우에 대상과 맞닿은 부분이 있으면 true를 반환한다.

- Set\_image(self)(Boss class에만 존재)

보스 본인의 체력에 따라 이미지를 바꿔준다.

- Popup.py

- 1)

Popup class로 각종 변동값이 있을 경우 이를 띄워주는 객체를 만든다.

Spawn\_position, 변동값의 종류, 변동값을 인자로 받아 객체가 만들어진다.

예를 들어, 보스 몬스터가 처치되면 그가 있던 자리에 스킬A 게이지와 KILL점수가 올라간다는 의미로 'A +5', 'kill +1' 2개의 popup객체가 생성되고 show()를 통해 보여진다.

- 2)

Duration : duration > 0인 동안 popup이 유지된다.

State : write로 초기값을 갖는다.

- 3) 함수 Show(self, draw, font)

Duration > 0인 동안 kind와 그 변동값을 출력해 주고 duration값을 줄여나간다. 이 값이 0이 되면 state가 erase로 바뀌어 지워진다.

- Meat.py

- 1) Meat class로 체력을 회복할 매개체를 만들어낸다.

- 2) 랜덤으로 좌표를 받아 만들어지며, 초기 state는 cooked이다.

- 3) 함수 Eat, overlap가 있으며, 각각 enemy의 kiss, overlap과 유사하다. Eat()은 user의 health를 1 증가시킨다.

- Bullet.py

- 1) Bullet class로 총알을 담는 객체를 만들어낸다.

- 2)

Damage : 총알의 데미지를 나타낸다.

Direction : command에 따라 총알 발사의 방향이 달라지며, 아무 방향도 입력되지 않으면 toward를 통해 최근에 바라본 방향으로 총알의 진행방향이 결정된다.

State : shot으로 시작한다.

- 3) 함수

- Move(self)

Direction 방향으로 움직인다

- Collision\_check(self, list), overlap(self, this\_position, object\_position)

총알이 다른 enemy, boss 리스트에 있는 객체들과 겹치는지 확인하여 데미지를 입힌다.

- Out\_check(self)

총알이 화면 밖으로 나가면 state를 out으로 변경한다.

#### • Img

Img폴더는 다음과 같이 이루어져 있다.

- 배경 2장
- 보스몬스터의 체력별 그림 6장
- 첫 화면에 나올 그림 1장
- 메인 캐릭터 상태별 그림 3장
- Bullet, enemy 각 1장씩

#### • Font

대구광역시의 무료 폰트 '달서힐링미디어'폰트를 사용하였다.

#### • How To Play

게임을 시작하고 B버튼을 누르면 게임방법이 나오게 되어 있다.

```
≡ howToPlay.txt
1  조이스틱으로 이동, 총알 자동 발사!
2  물려오는 적들을 피하며 처치하세요!
3  적과 부딪히면 체력이 깎여요πθπ
4  큰 몬스터는 체력도 많고 강합니다!
5  갈색 고기를 먹으면 체력 회복!
6
7  몬스터 처치 시 스킬A게이지가 상승!
8  스킬A -> 작은 적 ALL KILL!!!
9  정지상태일 때 스킬B 게이지가 상승!
10 스킬B -> 일시적으로 무적이 됩니다!
11 스킬은 한 번씩만 사용이 가능해요!
```

#### 4. 개발 이력

- 프로젝트 시작 ~ 11.22

프로젝트를 시작하기에 앞서 필요한 그림을 그리고, 조교님의 예제코드를 보며 어떤 식으로 코드를 짜야 하는지 방향을 잡았다.

- 11.22 ~ 11.28

게임의 기본 틀을 제작하였다. 랜덤으로 몬스터가 나오고, 총알을 발사하여 몬스터를 처치하고 스킬A를 사용하는 단계까지 진행되었다.

- 11.29 ~ 12.05

스킬B와 체력 회복을 위한 고기, 스킬 게이지 바를 구현하였다. 게임오버 화면을 만들었다.

- 12.06

교수님께 그간 만들어 온 코드리뷰를 부탁드렸고, 어떤 변화가 있었을 때 이를 사용자에게 알려주는 '피드백' 요소를 추가할 것을 권유해 주셨다.

- 12.07 ~ 12.18

폰트 추가를 통해 글자의 크기를 조절하게 되었다.

보스의 체력마다 다른 그림을 보여주고, 몬스터와 부딪혔을 때 캐릭터의 모습이 일시적으로 변해 체력이 깎였다는 것을 사용자가 알 수 있게 했다.

점수, 체력 등이 변할 때마다 팝업을 띄워 값이 변하는 것을 사용자에게 알려주었다.

게임 시작 화면을 추가하고, 게임을 재시작 할 수 있게 변경하였다.

추가로 스킬을 썼을 때 화면이 일시적으로 하얘지거나 캐릭터의 색이 변하는 등 사용자의 명령에 대한 피드백을 주는 요소를 추가하는 데 집중했다.



## 5. 주요 이슈 및 해결책

- 관리 미흡

가방에서 음료수가 터져 같이 있던 디스플레이가 망가져 교수님께서 교환을 해주셨다.

- paste

Paste()함수의 존재를 몰라서 그림을 그려 놓고 사용을 못하다가 수업 시간에 우연히 조교님이 다른 학생에게 알려주시는 것을 듣고 알게 되었다.

- time 모듈 사용

초기 목적은 time 모듈을 활용하여 시간적인 요소들을 포함하려 했는데, 잘 구현이 되지 않았고, 앞서 디스플레이가 망가졌던 것과 그림을 못 그려 시간이 많이 지체되었던 관계로 while문이 한 번 돌 때마다 count를 1씩 증가시키는 것으로 대체하고 이를 이용했다. 대신 time모듈은 랜덤값을 추출하는 용도로 사용하였다.

- 스킬B 작동

스킬B를 누를 때마다 캐릭터와 몬스터들이 순간이동을 하는 듯 갑자기 위치가 바뀌었는데, 조건문 설정을 잘못 해서 B버튼이 눌렸을 때에는 그림을 그리지 않아서 스킵이 된 것이었다.

- 보스 몬스터와의 접촉

캐릭터가 보스 몬스터와 접촉했을 때 보스 몬스터가 죽고 캐릭터가 데미지를 입어야 하는데 부딪히는 게 확인이 안 될 때가 있었다. 이는 캐릭터의 move()함수가 좌표를 4칸씩 이동시키기 때문에 보스 몬스터의 좌표와 정확히 들어맞지 않을 때가 있었기 때문이었다. 이는 접촉을 확인하는 함수에 범위를 추가하여 해결하였다.

## 6. 배운 점

프로젝트를 진행하다 막히는 부분이 있을 때마다 구글링을 하면서, 원하는 답을 찾는 시간도 빨라지고 더 정확한 내용을 찾을 수 있게 되었다. 구글링도 실력이라는 말을 많이 들었는데, 이번 프로젝트를 하면서 가장 큰 수확 중 하나라고 생각한다.

코드를 더 보기 쉽게 짜는 법을 배웠다. 하나의 main함수에 다 적는 게 아니라, 각각의 클래스, 함수 등을 분리해서, 잘 찢다고는 생각하진 않지만 조금 더 찾거나 수정하기 편한 코드를 작성했다. 다른 수업에서 객체지향 프로그래밍에 있어서 copy and paste를 줄이는 것이 중요한 것 중 하나라고 배웠는데, 최대한 적용해보려 했으나 그렇지 않은 부분이 본인이 봐도 많이 있어 아쉽다.

게임을 만드는 데 있어서 사용자의 관점에서 어떻게 봐야 할지 안목을 기를 수 있었다. 스스로 생각해 낸 것도 있었지만, 주변 학우들과 교수님의 피드백 덕분에 더욱 다방면에서 바라보는 시선을 기를 수 있었고, 게임의 완성도를 조금이라도 더 높일 수 있었다.