



# OSSPC 중간보고



8조 3인분



2015110510 통계학과 김태환  
2015112583 산업시스템공학과 박재민  
2018111711 의생명공학과 배유진





01



게임 외 기능

02

인게임

03

서버

04

향후 계획



# 게임 외 기능



MENU

# 인터페이스



★

변경 전



★

변경 후



★

★

- 파란색 글씨가 검은 배경에서 잘 보이지 않는 부분 수정

- 마우스로 메뉴 선택 불가능한 메뉴에서, 마우스로 클릭 가능한 버튼 추가

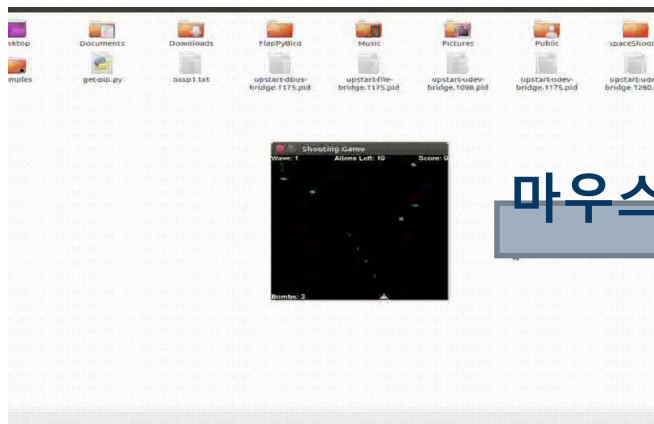
MENU

## 창 크기 조절

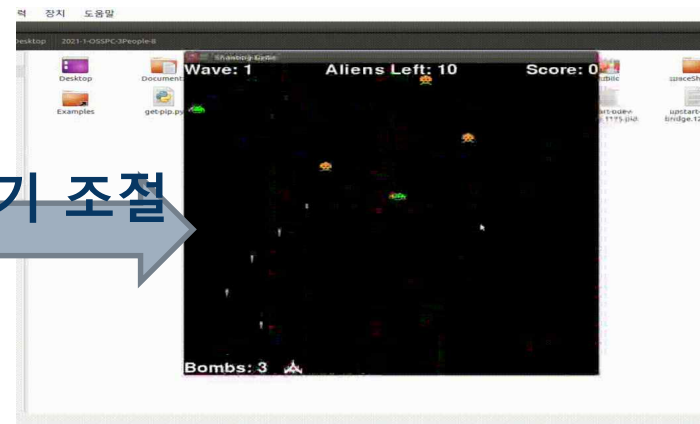


★

★



마우스로 크기 조절



★ - 기존 500x500으로 고정된 위치에서, shooting\_game.py와 sprites.py의 상위 모듈이 있는 main.py를 생성하여 모든 값이 화면 크기에 비례하도록 수정

- 메뉴 선택 화면에서 마우스로 창 크기를 조절 가능, 화면 크기에 비례하여 모든 이미지가 조절됨

MENU

## 로그인



### 추가된 기능



- 개인 기록 저장, 경쟁 요소를 추가하기 위해 서버 기능 추가
- 로그인 화면, 계정 생성 화면 추가
- 로그인 후에는 최고기록, 업적, 언어 설정을 불러옴

\* \*

MENU

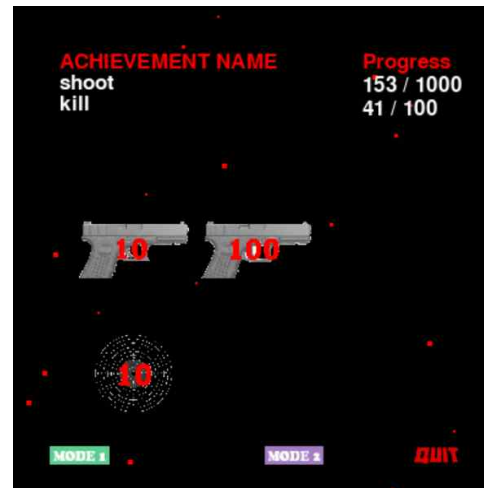
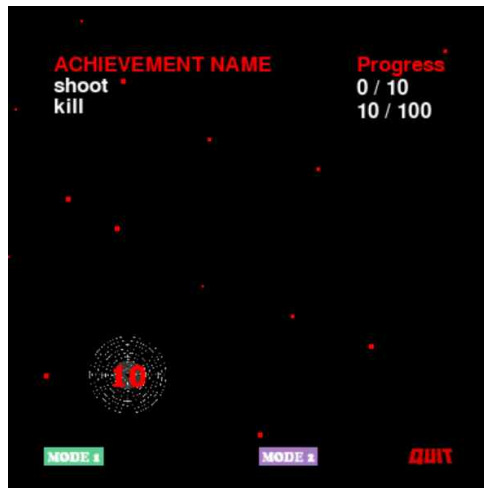
업적



★

추가된 기능

★



- 하이스코어만 지원하던 기존 게임에서,  
미사일을 발사한 횟수(shoot)와 적을  
무찌른 횟수(kill)을 저장하여 로그인 후에  
업적(ACHIEVEMENTS)에서 확인  
가능하도록 함

- 10, 100, 1000회 마다 뱃치가 추가됨

★

★



MENU

## 최고기록



### 변경 전

NAME	SCORE	ACCURACY
A	40	0.4
AA	4	0.4444
A	3	0.375
TEST	26	0.6667
A	0	0.0

### 변경 후

NAME	SCORE	ACCURACY
A	40	40.0%
TEST	26	66.67%
AA	4	44.44%

- 로컬에서만 기록되던 최고기록을 서버에 등록하여 다른 사람과 경쟁 가능하도록 함

- 아이디마다 최고 점수 하나만 나타나도록 하고, 정확도를 %로 변경

- 각 아이디 옆에 개인의 업적 아이콘이 나타나도록 함

5주차



이름		점수	정확도
A		40	40.0%
TEST		35	23.18%
AA		5	83.33%



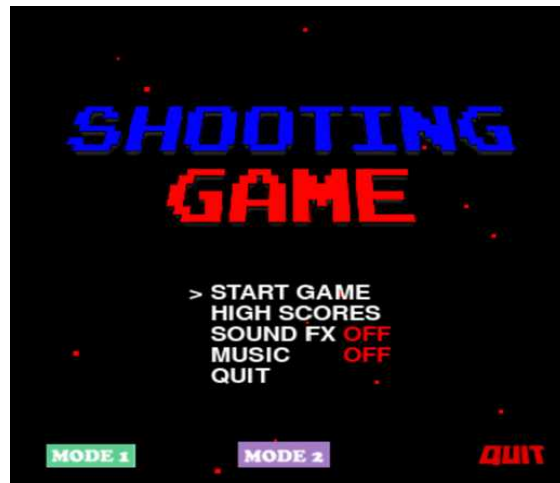
MENU

언어



★

변경 전



★

변경 후



★ - 기존 영어로만 플레이 가능하던 게임을,  
언어(LANGUAGE)를 선택하면 영어와 한글  
인터페이스가 반전됨

★ - 로그인 후 언어를 선택한 상태로 게임을  
종료하면 언어 정보가 계정에 저장되어 이후 게임  
접속 시 반영됨

# 인게임



IN-GAME

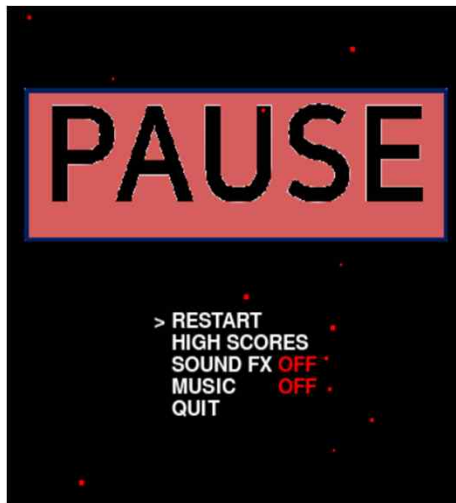
## 일시정지



★

★

### 추가된 기능



- 게임 중간에 P를 누르면  
일시정지를 할 수 있음

★

- 음향 설정 및 최고기록을 볼 수  
있으며, 로그인을 했을 때만 업적  
메뉴가 추가됨

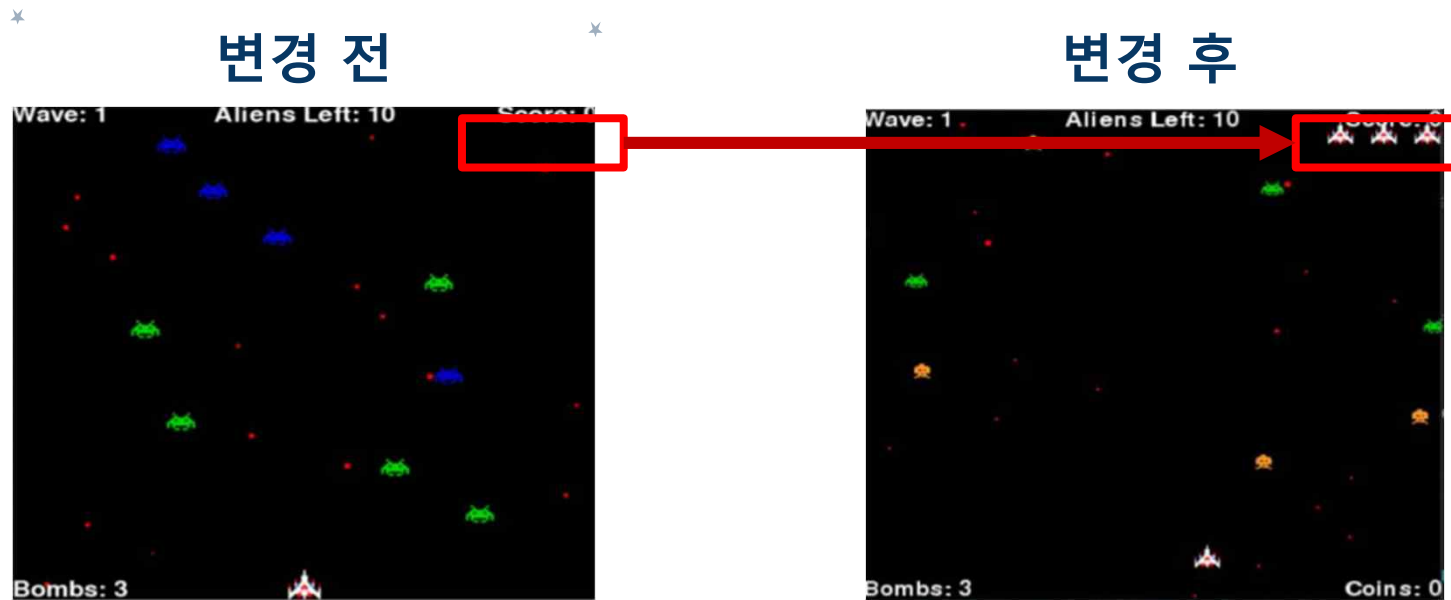
★

★

★

IN-GAME

## 라이프 구현



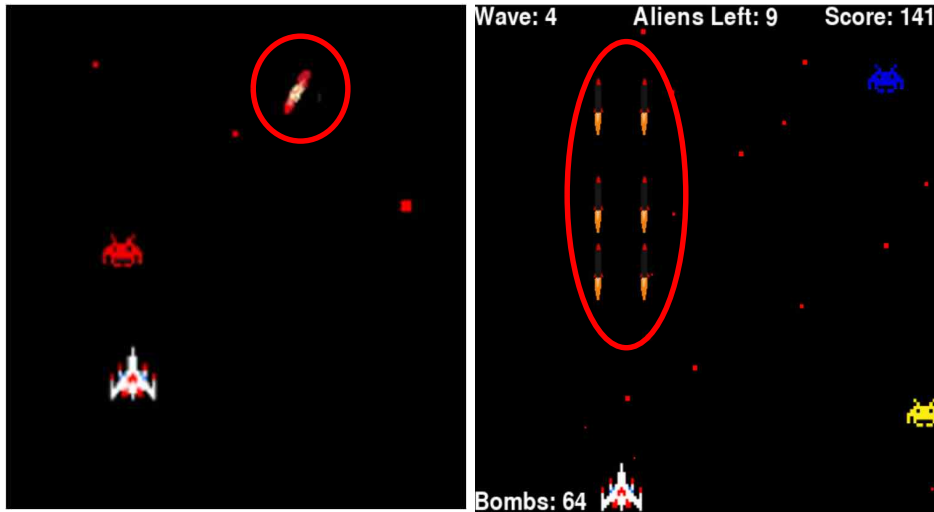
- ★ - 기존 한번만 맞으면 종료되던 게임에서, 라이프를 구현하여 게임 난이도를 완화함
- 적과 충돌할때마다 라이프가 하나씩 감소함

IN-GAME

## 아이템 추가



### 추가된 기능



- 더블미사일 아이템이 드랍되게 함

- 더블 미사일 아이템을 먹거나,  
상점에서 구매하면 더블미사일이  
일정시간 발사됨

IN-GAME

## 상점



### 추가된 기능



- 아이템과 같이 떨어지는 코인을 얻어서 코인을 획득할 수 있음
- 웨이브 종료마다 i키를 누르면 상점에 들어갈 수 있음

IN-GAME

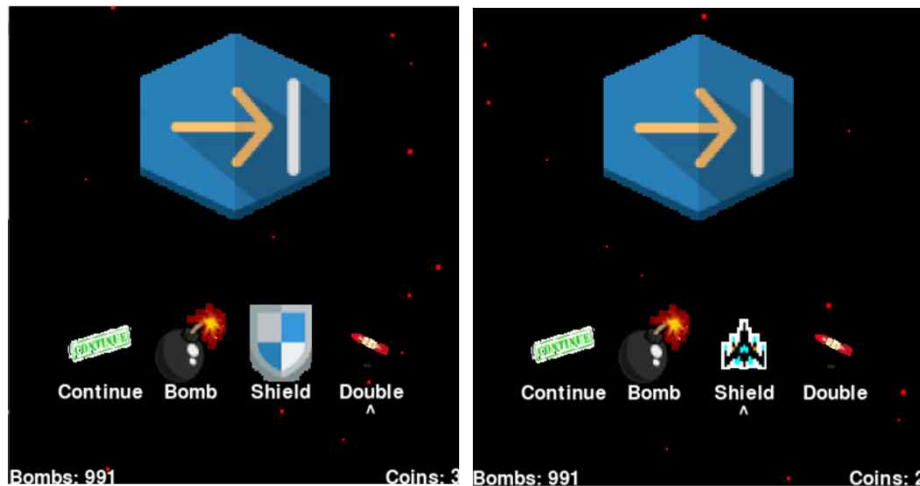
상점



★

★

## 추가된 기능



- 폭탄 구매 시 폭탄 개수가 증가하고, 일회성 아이템인 쉴드, 더블미사일은 웨이브당 한번만 구매 가능하며, 구매 시 아이콘이 바뀜

★

★

★

★

서버





IN-GAME

## 서버 구현



```
ubuntu@ubuntu-VirtualBox:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-VirtualBox:/home/ubuntu# ssh -i /home/ubuntu/
Documents/security/awstestkth.pem ubuntu@15.164.204.93
```

```
ubuntu@ip-172-31-15-56:~$ cd fastapi/
ubuntu@ip-172-31-15-56:~/fastapi$ ls
Dockerfile_fastapi  app  docker-compose.yml
ubuntu@ip-172-31-15-56:~/fastapi$ docker-compose up -d
```

```
app = FastAPI()

@app.get("/")
async def hello():
    return "osspc-2021-1-8"

@app.post("/create_account/")
async def create_id(account: Account):
    if account.id not in id_list:
        id_list.append(account.id)
        pw_list.append(account.password)
        language_list.append("ENG")
        achievement_list.append([account.id, 0, 0])
        return "Account created"
    else:
        return "Exist same ID"
```

- 단순한 api 기능만을 구현하기 위해, 비교적 단순하고 공식문서가 잘 정리된 fastapi(python 웹 프레임워크)로 서버 개발

- app/main.py에 api 작성, dockerfile\_fastapi 작성 후 docker-compose를 이용하여 fastapi 서버 컨테이너 빌드

IN-GAME

## 비동기 예외 처리



★

★

- 비동기 처리를 지원하는  
grequests 라이브러리 사용

```
data = {"id": id, "language": "KOR"}  
req = grequests.post(url + '/record_language/', json=data)  
res = grequests.map([req])  
if req.response == None : return scr_size, level_size, '', language
```

★

- Json, grequests 라이브러리를  
이용하여 api post 요청을  
받아들임

★

- 요청 실패 시 로그아웃된 메인  
메뉴로 돌아가도록 설정

★

★

★

# 향후 계획



PLAN

## PvP 모드, 협동 모드 추가



★

★

6주차	7주차
PvP, 협동모드 UI 수정	협동모드, PvP모드 기록 확인
게임 화면 두개 동시 생성 (PvP)	협동모드 서로 적용되는 쉴드
비행기 객체 두개 동시 생성 ★	PvP 적 방해 아이템
PvP모드 마우스로 게임 화면 넘기는 기능	모든 부분 디버깅 및 마무리
협동모드, PvP 관련 기능 api	

★

- 작업량이 가장 많을 것으로 예상되는 기능으로, 팀원 전부가 2주 동안 기능을 구현할 것으로 예상됨

- 최종 발표 전에 모든 디버깅을 완료할 예정

★

★

감사합니다