

2021-1 OSSP

# 최종보고서

8조 3인분

<https://github.com/CSID-DGU/2021-1-OSSPC-3People-8>

2015110510 통계학과 김태환

2015112583 산업시스템공학과 박재민

2018111711 의생명공학과 배유진

## 목차

- 프로젝트 소개
- 일정 진행
- 세부 기능
- 역할 분담 & git 협업
- 최종 평가 및 소감

## 프로젝트 소개

### 기존 프로젝트



### A simple alien shooting game built in python3 using pygame

URL: <https://github.com/jpritcha3-14/shooting-game>

w/a/s/d키로 우주선(ship)을 움직여 적(alien)을 격추하는 간단한 슈팅게임으로, Space키로 미사일을 발사하고, b키로 폭탄(bomb)을 사용할 수 있음.

### 구성

shooting\_game.py (460 lines): 게임을 실행하는 메인 코드

load.py (38 lines): 게임 내의 이미지와 음악을 로드하는 함수를 선언하는 코드

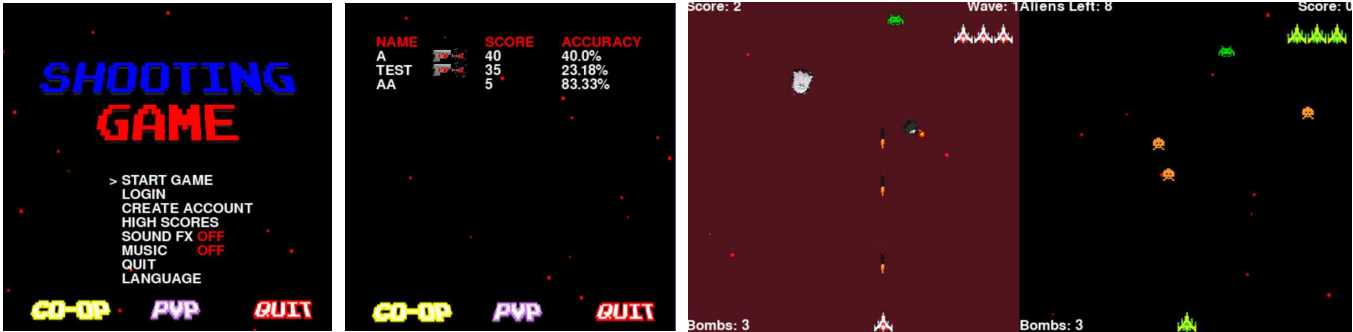
sprites.py (291 lines): 게임 내에서 사용되는 객체를 정의하는 코드

database.py (61 lines): 하이스코어를 저장하는 hiScore.db를 생성하고, db에서 인게임으로 점수를 불러오는 코드

### 라이선스

MIT License

개선된 프로젝트



Shooting-game by 3People

URL: <https://github.com/CSID-DGU/2021-1-OSSPC-3People-8>

기존 프로젝트에 2가지 게임 모드와 서버 기능, 업적 기능 등을 추가해 개선한 프로젝트  
기존 프로젝트에서 부족했던 경쟁요소를 추가하여 게임의 성취감을 높임

구성

- main.py(38 lines): 각종 모드와 화면 크기를 관리하기 위한 코드로, 게임 플레이를 위해 실행해야 하는 코드
- shooting\_game.py (1407 lines): 게임의 기본 모드(싱글 모드)를 실행하는 코드
- mode\_one.py(1530 lines): 협동모드를 실행하는 코드
- mode\_two.py(958 lines): PVP모드를 실행하는 코드
- load.py (38 lines): 게임 내의 이미지와 음악을 로드하는 함수를 선언하는 코드
- sprites.py (624 lines): 게임 내에서 사용되는 객체를 정의하는 코드
- database.py (61 lines): 하이스코어를 저장하는 hiScore.db를 생성하고, db에서 인게임으로 점수를 불러오는 코드
- server\_code.txt: 서버에 업로드한 코드를 텍스트파일로 저장해 놓은 파일(AWS EC2에서 구동 중)

라이선스

MIT License

Requirements

- OS: Ubuntu 20.04
- Python >= 3.6
- pygame == 2.0.1
- grequests

실행 방법

- pip3 install pygame
- pip3 install grequests
- git clone <https://github.com/CSID-DGU/2021-1-OSSPC-3People-8.git>
- python3 main.py

## 일정 진행

### 초기 진행 계획

	1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차
⇒ 메인 코드 분석 및 발표 준비(모두)								
⇒ UI개선, 마우스로 메뉴선택 (박재민) 아이템 추가(배유진) 창 크기 조절(김태환)								
⇒ 코인, 라이프 시스템(박재민) 적에 따라 score다르게 (배유진)								
⇒ 서버 기능, 령킹 저장 시스템(김태환)								
⇒ 미사일 공격하는 적 추가 (박재민) 구체적인 업적 시스템 완성(배유진)								
⇒ 메뉴 화면 추가(박재민) 아이템 추가(배유진) 업적 시스템 추가(김태환)								
⇒ PvP모드, 협동 모드(모두)								
⇒ 기능 마무리 및 최종 발표준비(모두)								

- 3주차까지 기존 일정대로 진행
- 2주차 구현 내용: 메뉴 UI 개선, 마우스 클릭 버튼 생성(다른 모드로 이동하는 버튼), 더블미사일 아이템 추가, 창 크기 조절, 게임 내 이미지 사이즈 조절
- 3주차 구현 내용: 게임 일시정지 및 라이프 시스템 구현, 서버 기능(로그인, 하이스코어) 구현, 적마다 score 다르게 변경

### 4주차: 진행 계획 조정

	4주차	5주차	6주차	7주차
김태환	서버 기능 마무리	업적시스템(2)		기능마무리 & 디버그
박재민	코인시스템	PVP모드		
배유진	업적시스템(1)	협동모드		

- 4주차 회의에서 진행 계획을 조정하기로 결정(개발 도중 버그 발생으로 일정 지연, 주간보고 피드백 반영, 기획 변경)
- PvP모드, 협동모드 개발 일정을 앞당기고, 4주차 진행 일정을 일부 수정함
- 코인 시스템이 비교적 많은 개발 시간이 소요되는 것으로 판단(웨이브 중 일시정지, 게임 중 코인 아이템 드랍, 코인 메뉴 구현, 코인으로 산 아이템 적용 등), 4주차에 추가적으로 구현하기로 결정


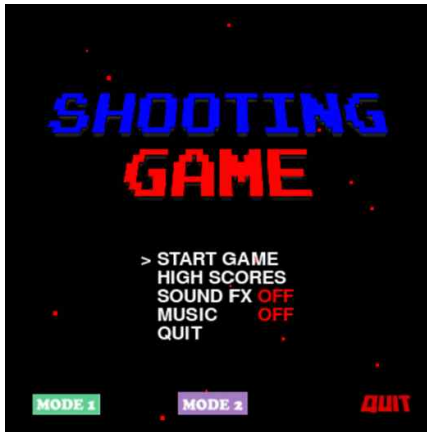
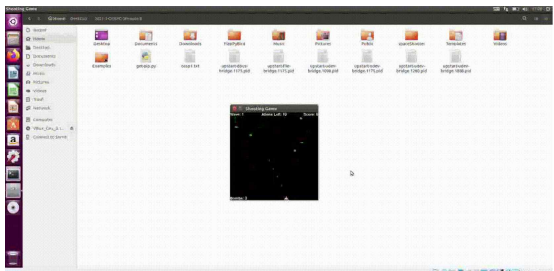
- 서버 최소 기능 구현이 3주차에 완료되어 서버 보안 기능(https 인증, 비밀번호 솔팅 해싱) 추가 시도 -> https 인증 오류로 서버 기능 추가를 포기하고 업적 시스템을 기존 일정보다 앞당겨 구현
- 업적 시스템을 만들고, 업적 페이지를 생성함, 추가로 언어 전환(한글화) 기능 구현

주차별 구현 기능 (최종)

	2주차	3주차	4주차	5주차	6주차	7주차
UI	메뉴 UI 개선	로그인 메뉴	언어 선택 기능 추가	하이스코어에서 개인 업적 확인	PvP, 협동모드 UI 수정	협동모드, PvP모드 기록 확인
	마우스 클릭 가능한 버튼		업적 메뉴 구현	언어 변경 구현 완료		
	창 크기 조절					
인게임	추가 아이템 (더블미사일)	라이프 시스템	코인 아이템 드랍	1인 모드 전체적인 코드 수정	게임 화면 두개 동시 생성(PvP)	협동모드 서로 적용되는 쉴드
	게임 이미지 사이즈 조절	일시정지 기능	웨이브 중간 코인 아이템 구매	전체적인 버그 수정	비행기 객체 두개 동시 생성	PvP 적 방해 아이템
		적마다 스코어 다르게			PvP모드 마우스로 게임 화면 넘기는 기능	
서버		로그인, 최고 기록 api	업적, 언어 api	하이스코어 업적 api	협동모드, PvP 관련 기능 api	모든 부분 디버깅 및 마무리

- 6주차부터 시간이 오래 소요될 것으로 예상되는 협동 모드와 PVP 모드를 만드는데 집중함
- 발표가 진행되는 7주차 수요일 이전에 디버깅이 완료될 수 있도록 마지막으로 일정을 재조정

## 세부 기능

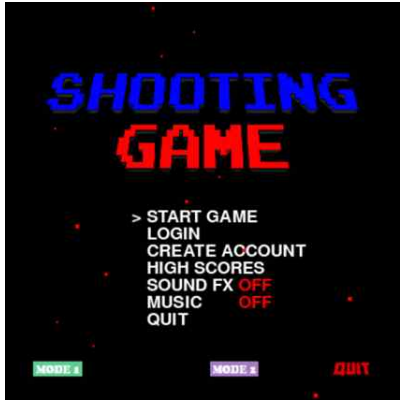
항목	변경 전	변경 후
2주차		
인터페이스	<ul style="list-style-type: none"> <li>- 파란색 글씨가 검은 배경에서 잘 보이지 않음</li> <li>- 로고와 메인메뉴만 구현되어 있으며, 마우스로 메뉴 선택 불가능</li> </ul> 	<ul style="list-style-type: none"> <li>- 흰색 글씨로 바꾸어 가독성이 향상됨</li> <li>- 메뉴 화면 밑에 마우스로 선택 가능한 MODE1, MODE2, QUIT 버튼을 추가, 버튼에 마우스를 올리면 버튼 크기가 변하도록 설정</li> </ul> 
창 크기	<ul style="list-style-type: none"> <li>- 500*500 사이즈를 기준으로 위치가 상수 값으로 고정되어 있으며, 창 크기를 조절할 수 없음</li> </ul> <pre># size stars background = pygame.Surface((500, 2000)) background = background.convert() background.fill((0, 0, 0)) backgroundLoc = 1500 finalStars = deque() for y in range(0, 1500, 30):     size = random.randint(2, 5)     x = random.randint(0, 500 - size)</pre>	<ul style="list-style-type: none"> <li>- 모든 값이 화면 크기에 비례하도록 수정하고, shooting_game.py와 sprites.py의 상위 모듈이 있는 main.py를 생성</li> <li>- 메뉴 선택 화면에서 마우스로 창 크기를 조절 가능하며, 화면 크기에 비례하여 모든 이미지가 조절되기 때문에 기능적인 변화 없이 화면 사이즈를 늘릴 수 있음</li> </ul> 

		
<p><b>게임 중 PAUSE</b></p>	<p>- 게임 중에 일시정지 기능이 없고 한 번 게임을 실행하면 게임이 끝날 때 까지 나갈 수 없음</p>	<p>- 키보드로 p버튼을 눌렀을 때 게임이 정지되며 pause화면이 나옴</p>  <p>- 게임이 실행되고 있는 도중 즉, ship객체가 alive가 true인 상황에서 pause버튼, 키보드p버튼,을 누르면 새로운 while문을 생성해 pause 화면이 나올 수 있도록 코드 작성</p>

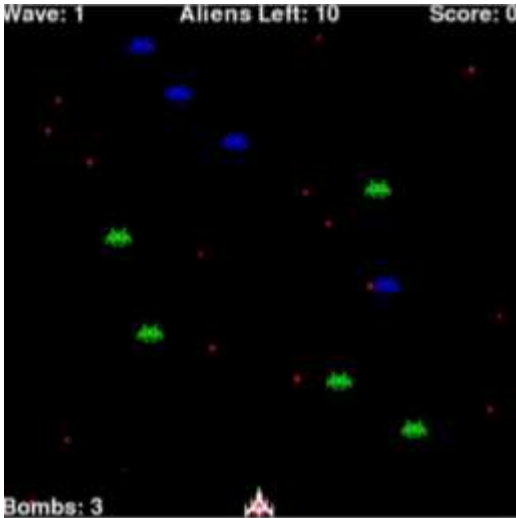



		<pre># pause 구현부분 elif (event.type == pygame.KEYDOWN and event.key == pygame.K_p):     inMenu = True     if id != '':         menuDict = {1: restartPos, 2: hiScorePos, 3: fxPos, 4: musicPos, 5: achievementPos}     else:         menuDict = {1: restartPos, 2: hiScorePos, 3: fxPos, 4: musicPos, 5: quitPos}     selectPos = selectText.get_rect(topright=restartPos.topleft)     while inMenu:         clock.tick(clocktime)         screen.blit(             background, (0, 0), area=pygame.Rect(                 0, backgroundLoc, scr_size, scr_size))         backgroundLoc += speed         if backgroundLoc - speed &lt;= speed:             backgroundLoc = scr_size*3      ###     for event in pygame.event.get():         if (event.type == pygame.QUIT):             pygame.quit()             sys.exit()         elif (event.type == pygame.KEYDOWN               and event.key == pygame.K_RETURN): # K_RETURN은 enter누르면             if showHiScores:                 showHiScores = False             elif showAchievement:                 showAchievement = False             elif selection == 1:                 inMenu = False             break</pre>
--	--	---

### 3주차

로그인	<p>- 로컬에서만 게임을 플레이하고, sqlite 데이터베이스를 이용해 개인의 하이스코어를 저장하는 것만 가능</p>	<p>- 서버를 이용해 계정을 생성하고, 로그인과 로그아웃 기능을 추가하여 플레이 정보를 서버와 주고 받을 수 있게 함</p> <p>- 계정을 생성하고 플레이기록을 서버에 저장하여 다른 사용자와 점수를 경쟁할 수 있음</p>
	<p>database.py</p> <pre>@staticmethod def setScore(hiScores, entry):     conn = sqlite3.connect(Database.path)     c = conn.cursor()     if len(hiScores) == Database.numScores:         lowScoreName = hiScores[-1][0]         lowScore = hiScores[-1][1]         c.execute("DELETE FROM scores WHERE (name = ? AND score = ?)",                   (lowScoreName, lowScore))         c.execute("INSERT INTO scores VALUES (?,?,?)", entry)         conn.commit()         conn.close()</pre> <p>shooting_game.py</p> <pre>elif (event.type == pygame.KEYDOWN       and event.key == pygame.K_RETURN       and len(name) &gt; 0):     Database.setScore(hiScores, (name, score, accuracy))     return True</pre>	<p>- 로그인 전 화면</p>  <p>The image shows the main menu of a game titled 'SHOOTING GAME' in large, stylized blue and red letters. Below the title, there is a list of options: '&gt; START GAME', 'LOGIN', 'CREATE ACCOUNT', 'HIGH SCORES', 'SOUND FX OFF', 'MUSIC OFF', and 'QUIT'. At the bottom, there are three buttons: 'MODE 1' (green), 'MODE 2' (purple), and 'QUIT' (red).</p> <p>- 계정 생성 화면</p>

		 <p>- 다른 사용자와 점수 경쟁</p> 
<p>마우스로 버튼을 클릭하여 모드 변경</p>	<p>- 마우스를 사용하여 메뉴를 선택할 수 없음</p>	<p>- 메뉴화면에 마우스로 선택할 수 있는 모드선택 버튼과 quit버튼 생성</p>  <p>- 버튼 구현 클래스 작성</p>

		<pre> class Button:     def __init__(self, gameDisplay, img_in, x, y, width, height, img_act, x_act, y_act, action):         self.lvl_size = 0         mouse = pygame.mouse.get_pos()         click = pygame.mouse.get_pressed()         if x + width &gt; mouse[0] &gt; x and y + height &gt; mouse[1] &gt; y:             gameDisplay.blit(img_act, (x_act, y_act))             if click[0] and action == 'quitgame':                 pygame.quit()                 sys.exit()             elif click[0] and action == 'mode_one':                 self.lvl_size = -1             elif click[0] and action == 'mode_two':                 self.lvl_size = -2         else:             gameDisplay.blit(img_in, (x, y))     def draw(self, screen):         # blit the text.         screen.blit(self.txt_surface, (self.rect.x+5, self.rect.y+5))         # blit the rect.         pygame.draw.rect(screen, self.color, self.rect, 2) </pre> <p>- 메인 메뉴 아래에 버튼을 생성하는 코드</p> <pre> #버튼 구현 button1Pos = 0.88 #mode1 button2Pos = 0.52 #mode2 button3Pos = 0.82 #quit  modeButton_one = Button(screen, modeImg_one, round(scr_size*button1Pos), round(scr_size* modeButton_two = Button(screen, modeImg_two, round(scr_size*button2Pos), round(scr_size* quitButton = Button(screen, quitImg, round(scr_size*button3Pos), round(scr_size*0.9), ro  if modeButton_one.lvl_size == -1 :     return scr_size, 1, id if modeButton_two.lvl_size == -2 :     return scr_size, 1.6, id  pygame.display.flip() #이미지 및 버튼 구현 </pre>
Life 기능	<p>- 적과 한번 충돌 시 바로 게임 오버</p> 	<p>- life기능을 추가하여 적과 충돌할 때 마다 오른쪽 위의 비행기 아이콘이 하나씩 사라짐, 비행기 아이콘이 다 사라지면 게임 종료됨.</p> 
서버 기능	<p>- 서버를 사용하지 않음</p>	<p>- ubuntu에 pem키를 400권한으로 저장하여, root 권한으로 EC2 접속</p> <p>- api만을 이용한 단순한 기능 구현, 파이썬 웹프레임워크인 fastapi 이용</p> <p>- fastapi 공식문서를 참고하여, python3.8 이미지에 fastapi, uvicorn을 빌드하는 Dockerfile을 만들고, docker-compose.yml 파일을 작성하여 컨테이너를 빌드함</p>

## 인터페이스

- 게임을 영어로만 플레이 할 수 있음

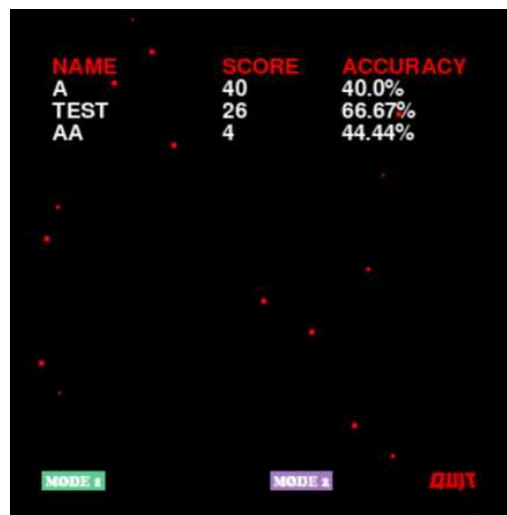
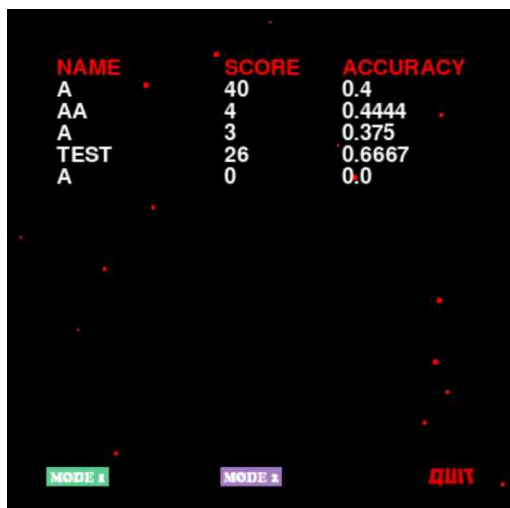
```
startText = font.render('START GAME', 1, BLUE)
startPos = startText.get_rect(midtop=titleRect.inflate(0, 100).midbottom)
hiScoreText = font.render('HIGH SCORES', 1, BLUE)
hiScorePos = hiScoreText.get_rect(topleft=startPos.bottomleft)
fxText = font.render('SOUND FX', 1, BLUE)
fxPos = fxText.get_rect(topleft=hiScorePos.bottomleft)
fxOnText = font.render('ON', 1, RED)
fxOffText = font.render('OFF', 1, RED)
fxOnPos = fxOnText.get_rect(topleft=fxPos.topright)
fxOffPos = fxOffText.get_rect(topleft=fxPos.topright)
musicText = font.render('MUSIC', 1, BLUE)
musicPos = fxText.get_rect(topleft=fxPos.bottomleft)
musicOnText = font.render('ON', 1, RED)
musicOffText = font.render('OFF', 1, RED)
musicOnPos = musicOnText.get_rect(topleft=musicPos.topright)
musicOffPos = musicOffText.get_rect(topleft=musicPos.topright)
quitText = font.render('QUIT', 1, BLUE)
quitPos = quitText.get_rect(topleft=musicPos.bottomleft)
selectText = font.render('*', 1, BLUE)
selectPos = selectText.get_rect(topleft=startPos.topleft)
```

- 언어(LANGUAGE)를 선택하면 영어와 한글 인터페이스가 반전됨, 로그인 후 언어를 선택한 상태로 게임을 종료하면 언어 정보가 계정에 저장되어 이후 게임 접속 시 반영됨



- 중복된 아이디도 하이스코어 랭킹에 표시됨
- 정확도(Accuracy)가 소수 형태로 표기

- 정확도(Accuracy)표기를 퍼센테이지(%)로 변경하고, 중복되는 아이디는 최고 점수만 표시되도록 함



## 코인 시스템

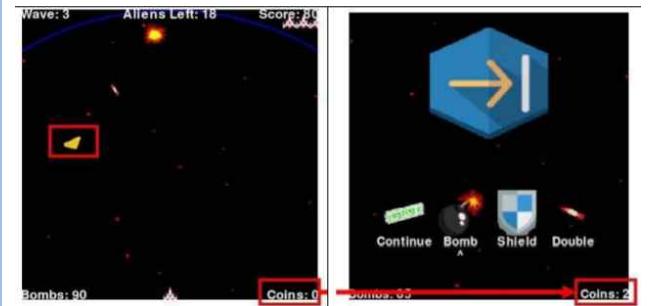
- 아이템은 게임 중에 드랍되는 것만 사용할 수 있으며, 별도로 구매할 수 있는 시스템이 없음

- 아이템과 같이 랜덤으로 드랍되는 코인을 추가하여 코인을 모아 shop에서 아이템을 구매할 수 있도록 함

- 코인 이미지

	
Coin * 1	Coin * 2

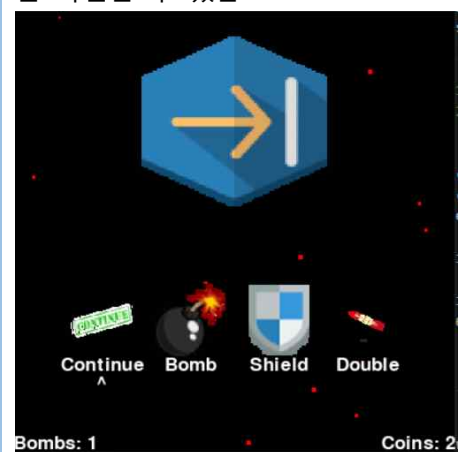
- 코인 드랍과 코인 획득





- 스테이지 별로 적들이 10, 20, 40으로 증가하며 등장하고, 해당 스테이지 적들을 모두 처치했을 때 다음 wave로 넘어가기 전에 3초의 간격이 있음




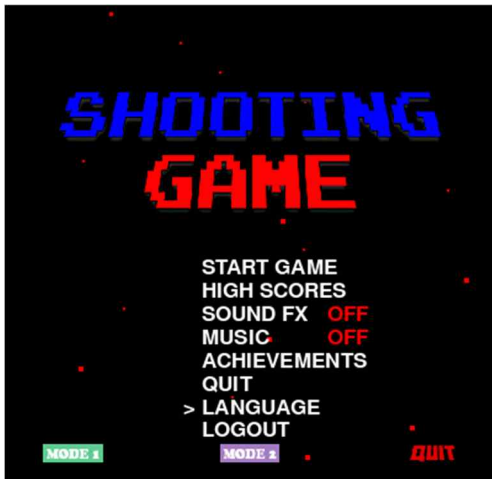
- Wave 이동 구간에서 카운트다운을 5초로 늘리고, 5초내에 키보드 i버튼을 누르면 shop으로 이동  
- 현재 보유하고 있는 코인이 있다면 아이템 구매 가능(아이템 가격은 1코인으로 동일)

- Bombs는 1이고 coins는 2인 상황에서 Bomb을 2개 구매했을 때 bombs는 3 coins는 0으로 감소한 것을 확인할 수 있음







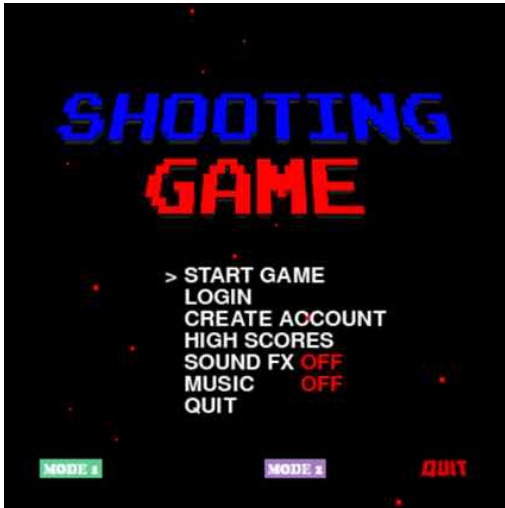

<p><b>업적</b></p>	<ul style="list-style-type: none"> <li>- 하이스코어(이름, 점수, 정확도)를 제외한 플레이정보가 저장되지 않음</li> </ul>	<ul style="list-style-type: none"> <li>- 미사일을 발사한 횟수(shoot)와 적을 무찌른 횟수를 저장하여 로그인 후에 업적(Achievements)에서 확인 가능하도록 함</li> <li>- 10, 100, 1000회 마다 배지가 추가됨</li> <li>- Kill 10회</li> </ul>  <ul style="list-style-type: none"> <li>- Shoot 151회, kill 41회</li> </ul> 
<p><b>서버 기능, 비동기 예외 처리</b></p>	<ul style="list-style-type: none"> <li>- 로그인, 하이스코어 관련 기능만 구현됨</li> <li>- 인터넷 연결이 안 되는 경우 게임이 종료됨</li> </ul>	<ul style="list-style-type: none"> <li>- 업적 관련 기능 추가 구현</li> <li>- 비동기를 지원하는 grequests 라이브러리를 이용하여, 요청이 느린 경우 대기함</li> <li>- 요청에 실패한 경우, 로그아웃된 메뉴 화면을 반환</li> </ul>


5주차		
<p>하이스코어 창에서 아이디별 업적 아이콘 추가</p>	<p>- 업적은 개인 업적 창에서 혼자만 볼 수 있었음</p>	<p>- hiscore 창에서도 개인의 업적을 볼 수 있는 아이콘 생성</p> 
<p>한글 미구현 부분 추가 구현</p>	<p>- 메인메뉴를 제외한 부분이 한글화 되지 않음</p>	<p>- 로그인, 업적, 하이스코어, 인게임에서 한글화 되도록 수정</p> 
<p>전체 코드 정리</p>	<p>- 반복되는 코드, 정리되지 않은 변수, 기타 버그 다수 존재</p>	<p>- 1인 모드를 기반으로 2인 모드를 개발하기 전에, 기존 모든 코드를 정리함</p> <p>- Resize로 변하는 모든 값을 size 클래스 변수로 모아선언</p> <p>- 반복되는 코드들을 함수로 구현</p>
<p>디버깅</p>	<p>- 언어 변환 시, 텍스트 길이가 달라서 텍스트가 겹치는 버그가 존재</p> 	<p>- 버그 수정, 버튼 위치 수정</p> <p>- 기타 메뉴 이동이 안되는 페이지, 아이템 오류 수정</p> 


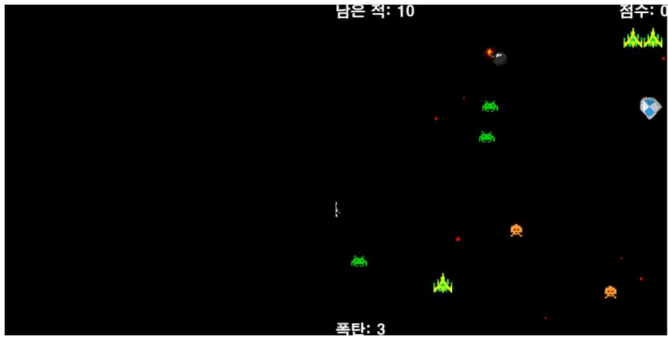



		
--	---	--

#### 6주차 ~ 7주차

버튼 수정	<ul style="list-style-type: none"> <li>- 버튼 이미지가 게임 컨셉에 맞지 않음</li> </ul>	<ul style="list-style-type: none"> <li>- 버튼 이미지를 게임에 어울리도록 수정함</li> <li>- 버튼을 눌렀을 때 다른 모드로 연결되도록 함 (CO-OP:협동모드, PVP: PVP 모드)</li> </ul>
		
협동 모드	<ul style="list-style-type: none"> <li>- 혼자서만 게임을 플레이 할 수 있음</li> </ul>	<ul style="list-style-type: none"> <li>- 협동 모드를 추가하여 하나의 화면에서 두 명의 플레이어가 함께 게임을 플레이 할 수 있음</li> <li>- 협동 모드에서는 두 명의 플레이어가 폭탄과 라이프, 코인 등을 공유하여 협력함</li> <li>- 방패 아이템 대신 팀방패(team shield)아이템이 드랍되어, 한 명이 방패를 얻는다면 다른 한 명에게도 방패가 적용됨</li> <li>- 협동 모드를 선택하면 화면 가로 비율이 큰 화면으로 변환되고, w/a/s/d키로 움직일 수 있는 ship2가 추가됨. Ship2는 q로 폭탄, v로 미사일을 사용할 수 있음</li> <li>- 협동모드도 싱글과 마찬가지로, 로그인을 두 번 하여 두명 로그인이 가능하고, 로그인 후 최고기록과 업적</li> </ul>

		<p>을 볼 수 있음</p>    
PVP 모드	- 혼자서만 게임을 플레이 할 수 있음	<p>- PVP모드를 추가하여 두 명의 플레이어가 각자의 화면에서, 더 높은 점수를 얻거나 더 오래 살아남는 서바이벌 경쟁을 함</p> <p>- 두 플레이어는 서로의 화면으로 넘어갈 수 없음</p> <p>- 게임 시작 전, L키를 이용하여 Player1과 Player2의 자리(입력하는 키보드 위치)를 바꿀 수 있고, 이는 화면 색깔로 구분됨</p> <p>- 상대 화면을 가리는 아이템을 추가하여, 드랍되는</p>

		<p>아이템을 획득 시 상대 화면을 2초간 가림 (비행기나 적은 그대로 움직이지만, 화면에 출력되지 않음)</p> <ul style="list-style-type: none"> <li>- 폭탄은 상대 화면을 넘어가고, 상대 화면에서 적을 죽여도 폭탄을 쓴 사람의 점수가 올라감</li> <li>- 빠른 진행을 위해 웨이브마다 게임 속도가 증가하고, 아이템이 많이 드랍됨. 끝 웨이브에 도달할 시에는 높은 점수가 승리함</li> </ul>   
디버깅		<ul style="list-style-type: none"> <li>- 게임 중간에 Pause 화면이 나타나는 버그 수정</li> <li>- Pause 화면이 깜빡거리는 버그 수정</li> <li>- 버튼 위치와 버튼 클릭 범위가 이미지 크기에 맞도록 수정</li> <li>- '적 남은 수' 등의 부자연스러운 번역 수정</li> </ul>

## 역할분담 & git 협업














### 역할분담

김태환(팀장): UI 수정, 창 크기 조절, 서버 기능(계정 생성, 로그인), 업적 기능, 협동모드 & PVP모드 도움

박재민: UI 수정, pause기능, 라이프기능, 코인 시스템, PVP모드

배유진: UI 수정, 아이템 추가, 언어 설정, 랭킹 수정, 협동모드

### git 협업

<input type="checkbox"/>	 screen switch	#19 by jaemin00955 was merged 12 hours ago
<input type="checkbox"/>	 협동 모드 추가, 버튼 수정	#18 by ebeuy was merged 5 days ago
<input type="checkbox"/>	 코인샵 완료	#17 by jaemin00955 was merged 13 days ago
<input type="checkbox"/>	 coin shop finish	#16 by jaemin00955 was closed 14 days ago
<input type="checkbox"/>	 coin shop	#15 by jaemin00955 was closed 15 days ago
<input type="checkbox"/>	 하이스코어에서 아이디 중복 제거	#14 by ebeuy was merged 15 days ago
<input type="checkbox"/>	 업적메뉴 로그인 후에만 보이도록 수정, accuracy 표기 퍼센트로 변경, 언어 변경 기능 추가	#13 by ebeuy was merged 16 days ago
<input type="checkbox"/>	 Coin Drop and Life function	#12 by jaemin00955 was merged 19 days ago
<input type="checkbox"/>	 업적 메뉴 추가	#11 by ebeuy was merged 23 days ago
<input type="checkbox"/>	 score differentiation	#10 by ebeuy was merged 24 days ago
<input type="checkbox"/>	 버튼 클릭으로 모드를 바꿀 수 있음	#9 by jaemin00955 was merged 27 days ago
<input type="checkbox"/>	 Doublemissile	#8 by ebeuy was merged on 4 May
<input type="checkbox"/>	 update img and pause and color	#7 by jaemin00955 was merged on 3 May

- 팀 저장소를 fork한 개인 저장소에서 팀원들이 각자 코드를 작성한 뒤 pull request를 보냄

- 팀장이 확인 후 merge하고, 충돌이 있는 코드들을 병합

- 팀원들이 branch를 사용하는데 익숙하지 않고, 원본 코드가 기능별로 잘 나누어져 있어 main branch로 작업

- 코드 간 충돌을 막기 위해 pull request 후에는 카톡방을 통해 공지하도록 함

- 매주 주말에는 개인 fork를 업데이트하고(fetch upstream), 로컬 저장소를 동기화(git pull)

## 최종 평가 및 소감

### 최종 평가

- 기존에 계획했던 기능들이 모두 구현되었으며, 주차별로 기능들을 더 추가하여 처음 계획보다 다양한 기능들이 추가되었음

- 게임 외 기능: UI 수정, 이미지 추가, 버튼 생성, 창 크기 조절, 언어 설정 추가, 서버 기능(계정 생성, 로그인, 로그 아웃, 업적) 추가, 최고 기록 랭킹 수정 → 사용자 편의성 향상, 경쟁 & 성취 요소 추가

- 인게임 기능: 일시정지 기능 추가, 라이프 기능 추가, 새로운 아이템 추가, 코인 시스템 추가, 협동 모드 & PVP 모드 추가 → 게임성 향상

- 그 외: 코드 내에 숫자 값 함수로 정리, 코드 기능별로 파일 분리 → 간결한 코드, 코드 수정 시 가독성 & 효율성 향상

## 소감

- 김태환: 누군가와 협업을 하면서 코드를 개선하는 경험, 특히 git을 통한 협업은 처음이여서 많이 어려웠다. 하지만 어려웠던 과정을 겪었기 때문에 협업 규칙의 중요성을 더욱 절실하게 느꼈고, 이번 기회로 협업 능력이 향상되었다고 생각한다. 또한 pygame을 비롯하여 Linux, AWS, Docker 등 새로운 것을 접해본 것도 큰 성과라고 생각한다. 다들 바쁜 일정속에서도 자료를 찾아보고, 편의성을 위해 고민하며 기능들을 추가해주어서 많은 기능과 완성도가 있는 결과물을 만들 수 있었다고 생각한다.
  
- 박재민: 오픈소스 프로젝트를 기반으로 새로운 오픈소스 프로젝트를 만든다는 것과 팀 단위의 협업 프로그래밍을 통해 이 수업의 목표를 경험해볼 수 있다는 것이 의미있는 시간이었던 것 같다. 그리고 개발자로서 가장 많이 활용할 git을 배울 수 있어서 좋았던 것 같다. window운영체제만 사용했던 지금까지와는 다르게 linux를 활용하여 프로젝트를 진행한 것 또한 추후에 많은 도움이 될 수 있을것이라고 생각한다.
  
- 배유진: 팀으로 프로젝트를 진행해 본건 처음인데, 좋은 팀원들을 만나서 잘 진행한 것 같다. 평소에 게임을 좋아해서 게임에 추가할 기능들을 논의하고 직접 구현해보는 과정이 재미있었다. 프로젝트 과정에서는 git과 python 능력을 기를 수 있었고, 수업 전반적으로는 Linux, Ubuntu, AWS와 docker, 오픈소스 라이선스 등 다양한 내용을 배울 수 있어서 힘들긴 했지만 의미 있는 수업이었다고 생각한다.