

YAMI TETRIS

최종 보고서

남남쩍쩍 조


<팀원>

2016112546 산업시스템공학과 조성우

2016112582 산업시스템공학과 김유탄

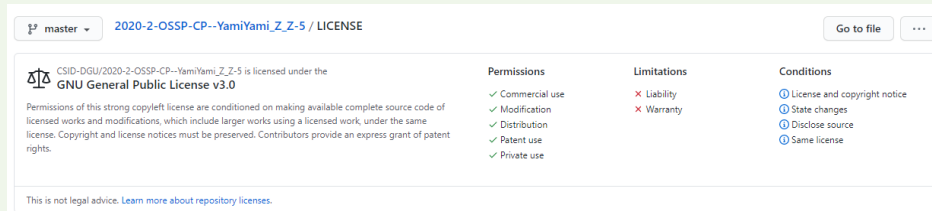
2018112474 산업시스템공학과 복유연

목차

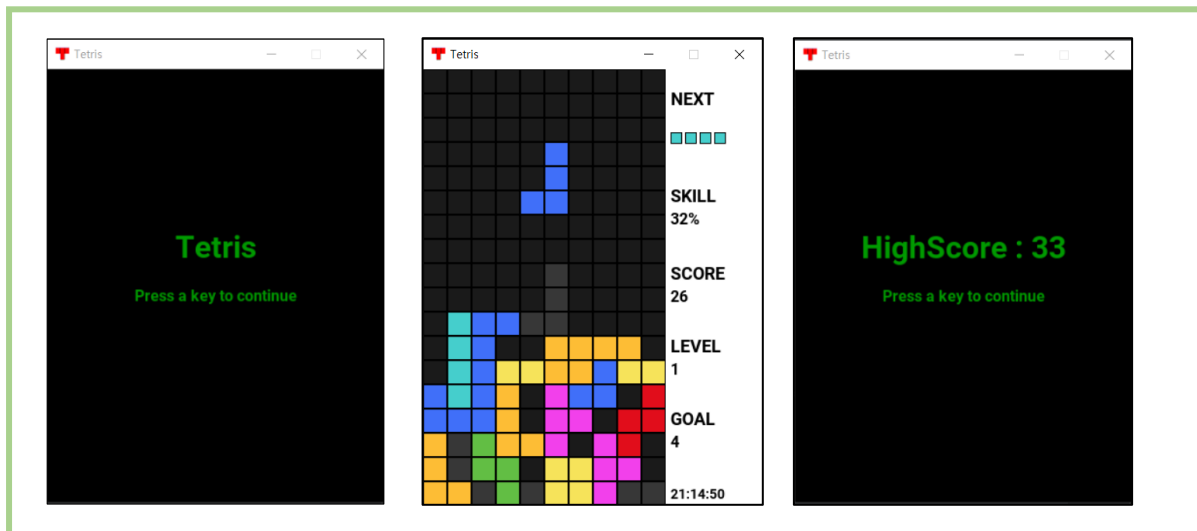
- 
1. Base Source
 2. 개발 환경
 3. Timeline 및 역할
 4. 세부 진행 사항
 5. Reference
 6. 최종 평가

1. Base Source

- ✓ OSD_game
- ✓ License : GPL-3.0



- ✓ 주소 : https://github.com/hbseo/OSD_game
- ✓ 사용 언어 : python

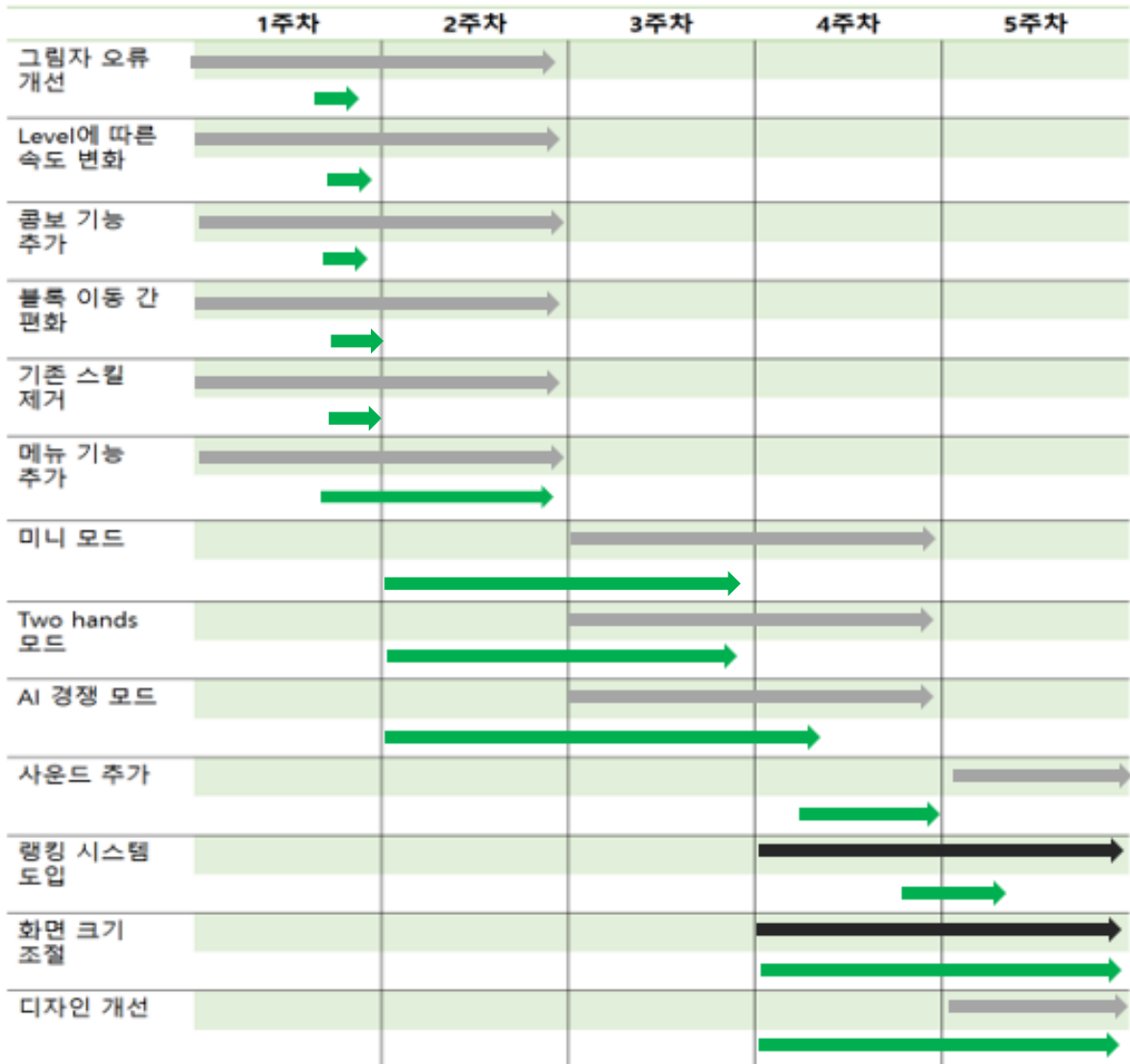


2. 개발환경

- ✓ 언어 : Python 3.6.12
- ✓ 편집기 : PyCharm
- ✓ OS : Linux(16.04.1)
- ✓ 추가 모듈
 pygame(2.0.0) , pygame_menu(3.3.0), pymysql(0.10.1)

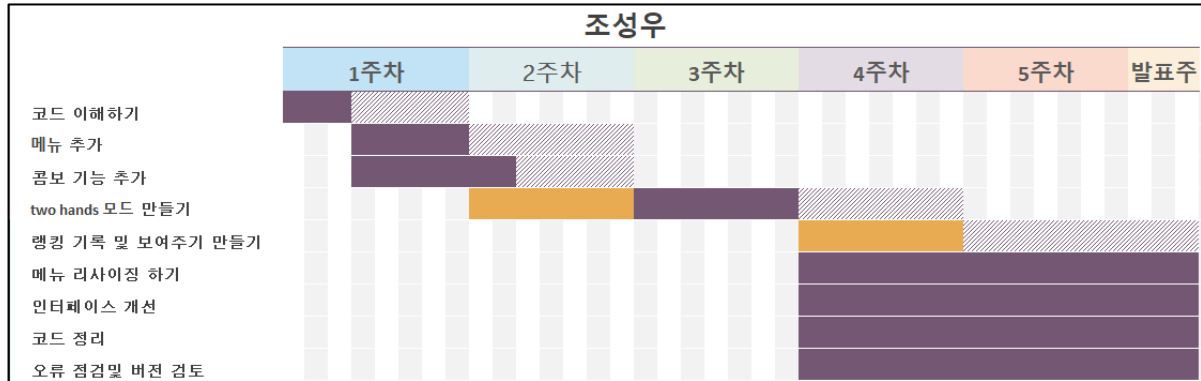
3. Timeline 및 역할

- 프로젝트 진행 타임라인 예정 :  추가 일정 :  실제 일정 : 

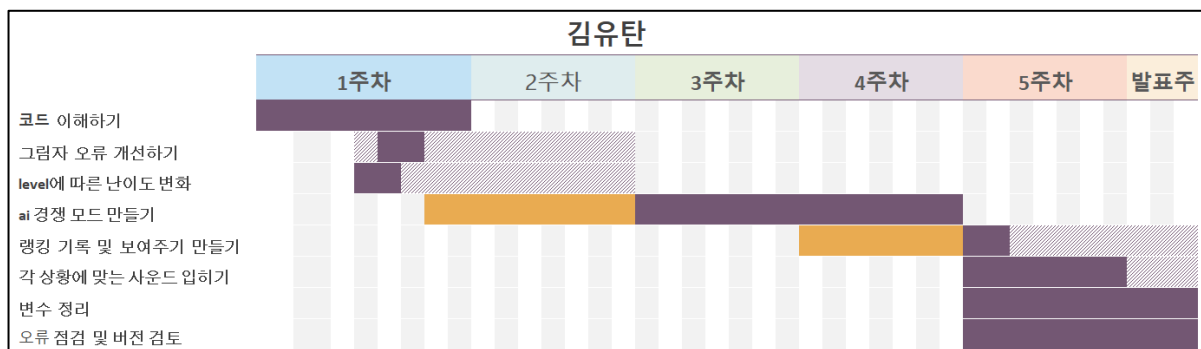


- 개인별 역할 분담 및 타임라인

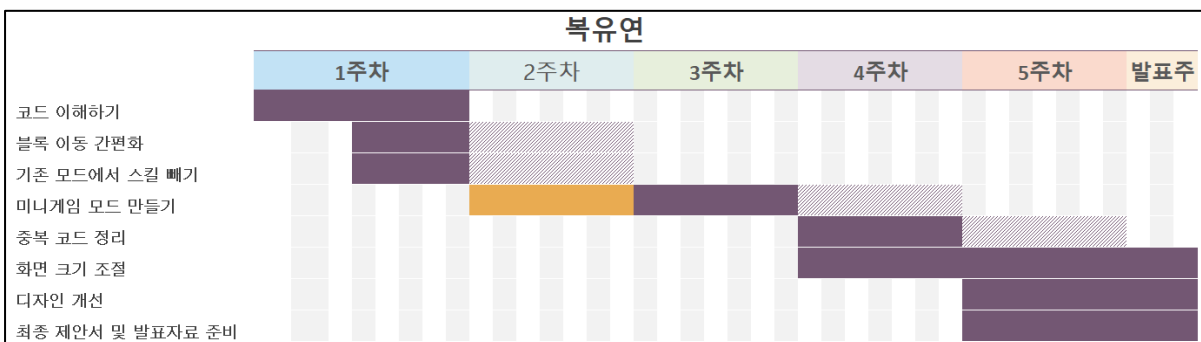
조성우



김유탄



복유연



기존 계획

- ✓ 그림자 오류 개선 / 블록 이동 간편화 / 콤보 기능 / 사운드 추가 / 디자인 개선
- ✓ Level에 따른 속도 변화 / 기존 스킬 제거 / 메뉴 기능 추가
- ✓ 시경쟁 모드 / 미니 모드 / Two hands 모드

추가 사항

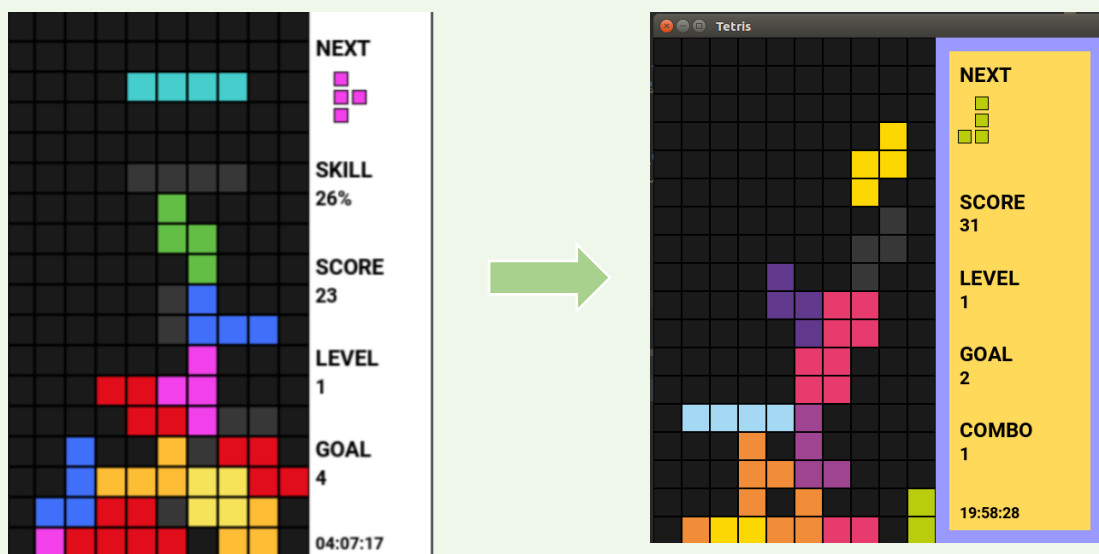
- ✓ 서버를 통한 랭킹 시스템 도입
- ✓ 화면크기 조절
- ✓ 하드 코딩 변경
- ✓ 중복 코드 및 파일 제거

4. 세부 진행 사항

- 오류 개선

A. 그림자 오류 개선

- ✓ 기존의 정상적으로 작동하지 않던 그림자가 제대로 생성 되도록 코드 변경
- ✓ 프로젝트를 진행하면서 생기는 여러 오류들 개선하면서 프로젝트 진행



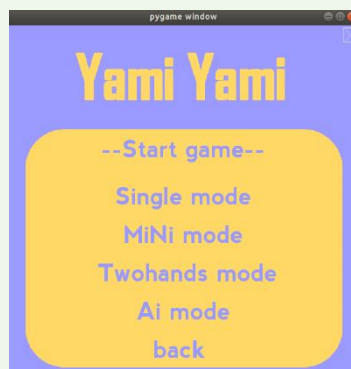
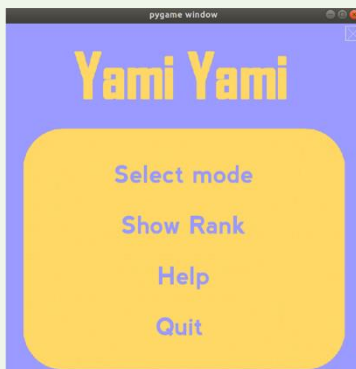
- 인터페이스 향상

B. 각 상황에 맞는 사운드 추가

- ✓ 기존의 저작권 문제로 정상적으로 재생되지 않던 사운드 제거 및 새로운 사운드를 wav파일로 추가
- ✓ 메뉴 버튼 클릭 사운드, 블럭 제거, 레벨업, 블록 정착 사운드 추가
- ✓ 시의 배경 음악은 경쟁 분위기 조성을 위해 다른 배경음악 선택
- ✓ 콤보 사운드는 넣었을 때 게임을 방해하는 느낌이 생겨서 제거

C. 메뉴 기능 추가

- ✓ Pygame_menu 모듈을 활용한 메뉴 만들기.
- ✓ 게임 시작시 메인 메뉴 화면 만들기.
- ✓ 게임 선택 및 각 랭크 확인 메뉴 만들기.
- ✓ 게임 종료 후 아이디 기록 창 만들기
- ✓ 랭크 입력 메뉴 만들기
- ✓ Help 메뉴 만들기

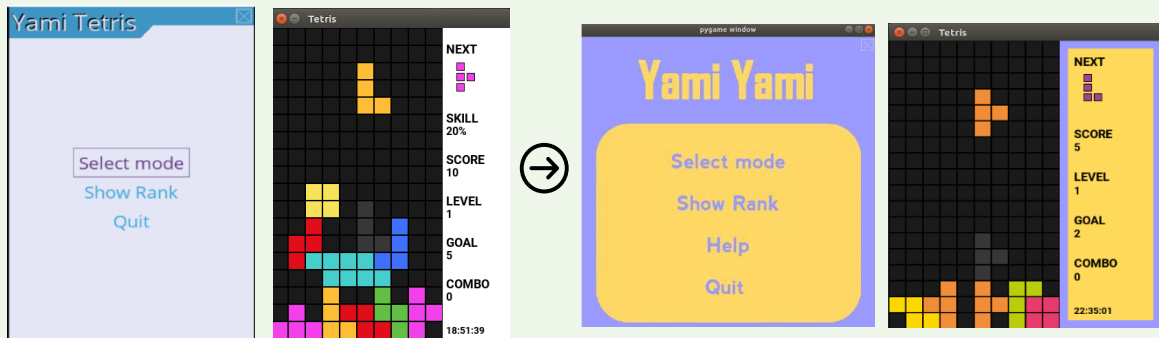


D. 블록 이동 간편화

- ✓ 키보드를 꼭 눌렀을 때 연속적으로 이동하도록 연속 키입력 구현

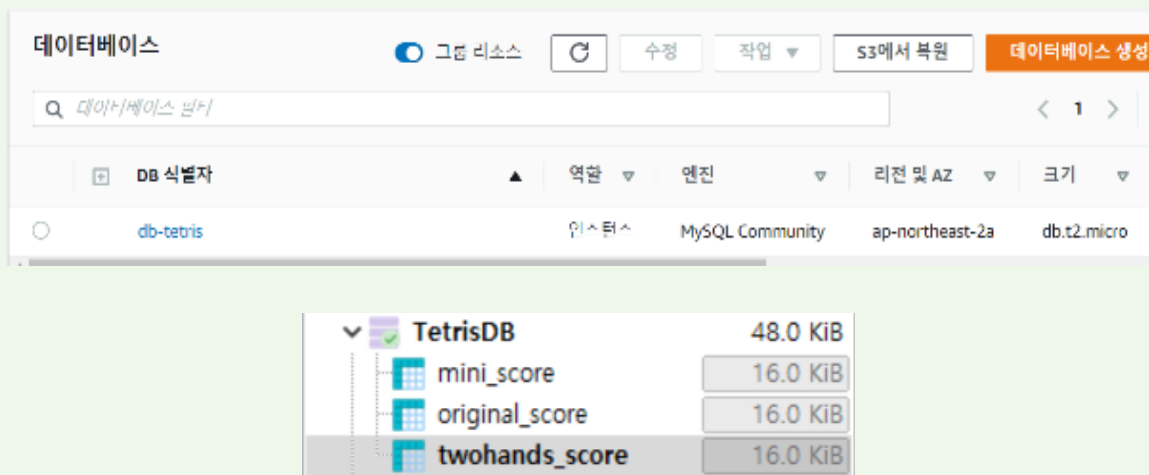
E. 디자인 개선

- ✓ 메뉴 화면 디자인 변경
- ✓ 게임 화면 메뉴와 디자인 통일
- ✓ 게임 종료 시 점수 화면 구현



F. 서버를 통한 랭크 제도 추가

- ✓ Amazon RDS에서 MySQL 데이터 서버 만들기
- ✓ 파이썬 내에서 데이터 서버 불러 오기 구현
- ✓ 게임 끝나면 랭크 기록하면 자동으로 데이터 서버에 기록 하기
- ✓ 기본, 미니, 양손 모드에 대한 점수는 데이터 베이스에 기록되고, 메뉴의 랭크 보기에서 전체 랭킹 5위까지 확인 가능





G. 화면 크기 조절 기능 추가

- ✓ 파이게임 메뉴를 리사이징 가능하게 변경
- ✓ 리사이징 이벤트 발생시에 창 크기를 받아 다른 변수들 조정
- ✓ 메뉴 배경, 폰트사이즈, 버튼, 여백 등 창 크기에 따른 리사이징 완료
- ✓ 각 모드별 스크린 비율에 맞게 블록사이즈, 글씨 크기, 보드판 내용물 모두 변경되도록 리사이징 완료

- 기존 모드 개선 및 모드 추가 하기

H. Level에 따른 난이도 변화

- ✓ user와 AI의 이벤트 발생 주기를 level에 따라 점점 작게 만들어서 게임의 진행속도가 빨라지고, 이에 따라 난이도가 증가 하도록 설정.

```
#레벨별 스피드 조절
def level_speed(self):
    if self.level < Var.max_level:
        pygame.time.set_timer(pygame.USEREVENT, (Var.user_start_speed - Var.user_per_speed * self.level))
        pygame.time.set_timer(Var.ai_event, (Var.AI_start_speed - Var.AI_per_speed * self.level))
    else:
        pygame.time.set_timer(pygame.USEREVENT, (Var.user_start_speed - Var.user_per_speed * self.level))
        pygame.time.set_timer(Var.ai_event1, (Var.AI_start_speed - Var.AI_per_speed * self.level))
```

I. 기존 모드 스킬 제거

J. Combo 기능 추가

- ✓ 한줄을 깨면 콤보 1 증가
- ✓ 최대 콤보 설정
- ✓ 콤보가 증가하고 10간 새로운 줄이 삭제 안되면 초기화

```
def combo_null_start(self):
    for i in range(self.combo_max):
        if self.combo==i:
            self.timer_list[i]=threading.Timer(10, self.combo_null)
            self.timer_list[i].start()
            for j in range(self.combo_max):
                if i != j:
                    self.timer_list[j].cancel()
```

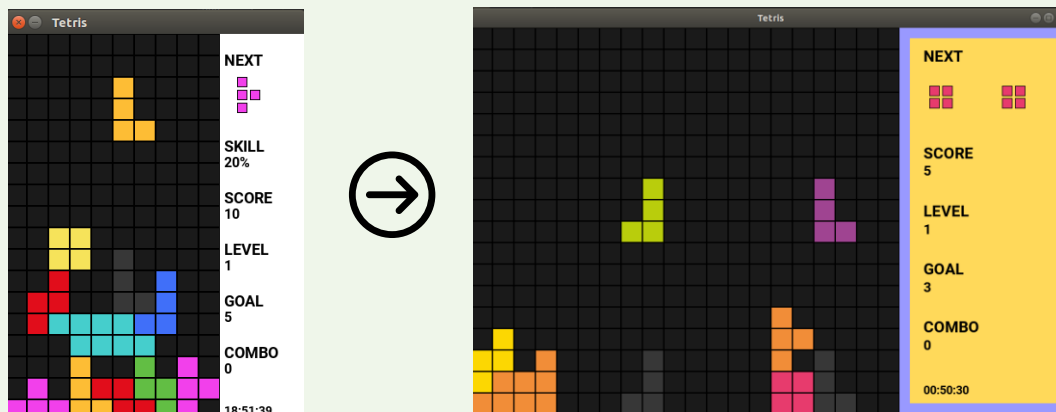
K. 미니 게임 모드 추가

- ✓ 가로가 5칸, 세로가 15칸인 미니 게임 모드
- ✓ 작동 방식은 기존 모드와 같음
- ✓ 보드가 작아져 일반 모드보다 난이도가 높음



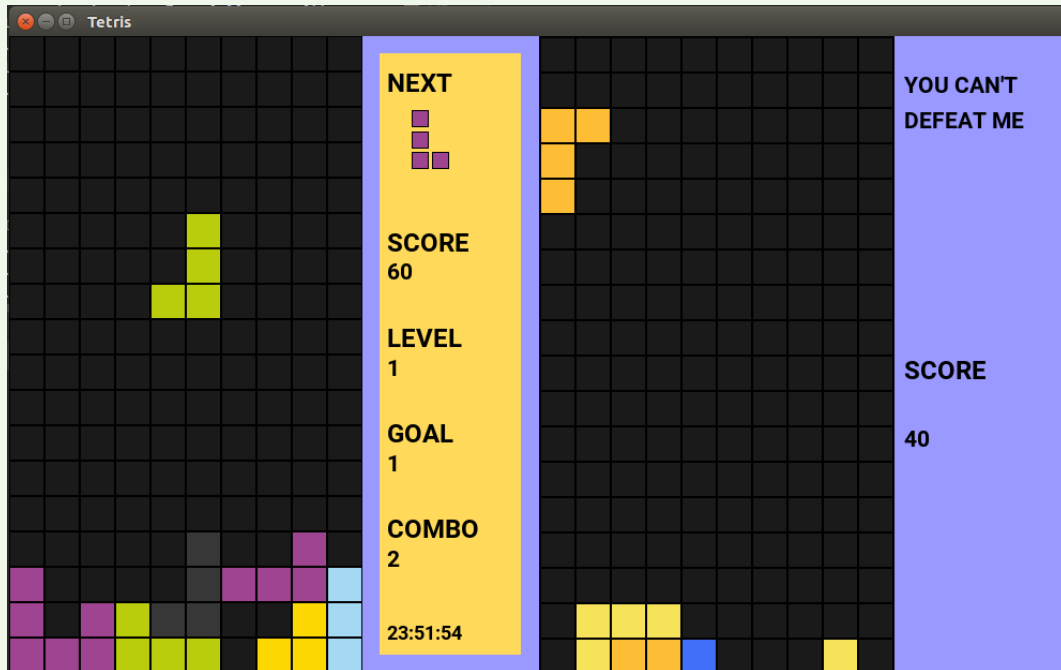
L. Two Hands 모드 추가

- ✓ 두명에서 혹은 양손으로 플레이 가능한 심화 게임 모드
- ✓ 좌측 블록은 모양 변환 = w , 방향 = asd, 빠르게 내리기 = e 로 조정
- ✓ 우측 블록은 모양 변환 = ↑, 방향 = ←→ ↓, 빠르게 내리기 = space 로 조정



M. AI 모드 추가

- ✓ 시와 경쟁하는 모드를 추가하여 혼자 게임을 진행하더라도 지속적으로 흥미를 유발함
- ✓ 유전 알고리즘을 사용하여 학습된 weight를 통해 화면의 우측에서 자동으로 테트리스가 진행됨
- ✓ Weight는 테트리스 보드의 밑에서부터 빈공간의 수, 한번에 지워지는 줄의 수 등 여러가지 변수들을 통해서 점수를 측정하고, 측정된 점수가 높은 경우의 weight를 통해 새로운 경우를 만들기를 반복함(자식 생성)
이를 통해서 최종적으로 점수를 가장 많이 획득하는 경우의 weight로 테트리스를 자동으로 진행하게 됨



추가 사항

- ✓ 하드 코딩 변경
 - 프로젝트를 진행하면서 생긴 중복 코드 정리
 - 기존에 있던 숫자 하드코딩들 코드 정리
 - 변수 파일 만들어 따로 변수 관리하기
- ✓ 중복 코드 및 파일 제거
 - 각자 나눠서 각 모드를 만들어서 생기는 중복 클래스 및 파일 정리
 - 중복되는 코드들 하나의 코드로 사용하기

5. Reference

AI 모드 : <https://github.com/YangtaoGe518/Tetris-AI>

사운드 : <https://www.bensound.com/royalty-free-music>

메뉴 :

https://pygame-menu.readthedocs.io/en/latest/_source/add_widgets.html

6. 프로젝트 평가

프로젝트 초반에는 Github으로 협업을 진행하는 방식에 대해서 모르는 부분이 많았고, 효율적인 협업을 진행하는 것에 대해 어려움을 많이 겪었지만, 각자의 fork저장소를 만들어 branch에서 작업을 하면서 개인의 github 사용 역량을 향상 시킬 수 있었고, 팀 저장소에 merge를 하는 과정에서 발생하는 conflict를 다루면서 팀으로 프로젝트를 진행할 때 어떠한 것을 주의 해야하는지 판단하고, 해결하는 능력이 향상되었다.