

오픈소스 소프트웨어 프로젝트

테트리스 프로젝트 제안서

남남찹찹 조

<팀원>

2016112546 산업시스템공학과 조성우

2016112582 산업시스템공학과 김유탄

2018112474 산업시스템공학과 복유연

목차

1. Base Source

- I. Base Source 기본 정보
- II. Base Source 에 대한 SWOT 분석

2. 개발 환경

3. 목표

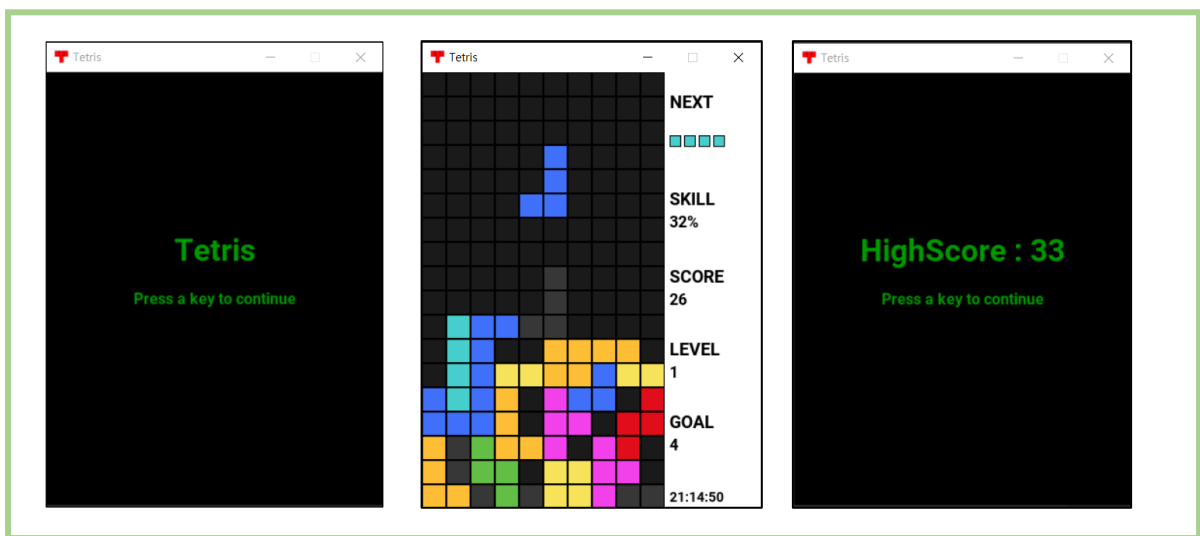
- I. 개선/추가 사항 및 기대효과
 - 오류 개선
 - 인터페이스 향상
 - 기존 모드 개선
 - 새로운 모드 추가

4. 프로젝트 일정 및 역할 분담

- I. 프로젝트 일정
- II. 역할 분담
- III. System Structure

BASE SOURCE – 기본 정보

- ✓ OSD_game
- ✓ License : GPL-3.0
- ✓ 주소 : https://github.com/hbseo/OSD_game
- ✓ 사용 언어 : python
- ✓ 추가 모듈 : pygame



BASE SOURCE – Base Source에 대한 SWOT 분석

- ✓ **Strength**
 - 기능별로 코드가 직관적으로 나누어져 있음
 - 테트리스에 대한 기본적인 기능요소들을 포함하고 있음
 - 팀에게 가장 익숙한 python을 기반으로 함
- ✓ **Weakness**
 - 효과음, 배경음, 메뉴 등에 대한 인터페이스가 단순함
 - 일관된 게임 모드로 인해 차별성, 흥미 요소 부족
 - 게임 내에 그림자와 관련된 오류가 존재함

✓ Opportunity

- 기본적인 기능들을 포함하기 때문에, 새로운 개선 사항에 대한 집중 가능
- 참고할 수 있는 다양한 테트리스 관련 소스들이 존재함
- 이미 유명한 게임이기 때문에, 게임에 대한 사용자의 이해도가 높은 편임

✓ Threat

- 이미 수많은 버전의 테트리스 게임이 존재하기 때문에, 흥미를 끌기 어려움
- 서로 얽혀 있는 코드가 많은 만큼, 기능을 추가할 때 문제 발생 가능성이 높음



✓ 선정 이유

기존의 강점을 기반으로, 기회요소들을 통해 약점과 위험 요소에 대한 해결을 충분히 할 수 있다 판단하고 주제로 선정

개발 환경

- ✓ 언어 : Python
(python version은 사용하는 여러 모듈과 호환되는 가상환경을 만들어 사용 예정)
- ✓ 코드 편집기 : PyCharm
- ✓ OS : Linux / Ubuntu

1. VirtualBox 활용

- ✓ aws를 사용하지 않아도, 기존의 컴퓨터의 성능을 나눠서 쓰더라도 프로젝트가 진행 가능하기 때문

2. Git 사용 방식 - fork

1. Main remote 저장소를 fork하기
2. Fork한 저장소를 local로 가져와 수정 후 add, commit, push 하고 pull request 수행
3. 팀원끼리 Main remote 저장소로의 merge 여부에 대해 논의
4. Merge에 대해 모두 동의할 경우 merge 승인하기

✓ 주의사항

1. fork했던 저장소들을 합치는 과정에서, conflict가 발생할 수도 있기 때문에 이에 대한 주의 필요
2. Main remote 저장소는 합의된 경우가 아니라면 직접 사용하지 않기.

3. Git 사용 방식 - branch

1. fork한 저장소에 자신의 역할을 수행할 branch 생성
2. 생성된 branch에서 작업 수행
3. 작업 완료시 master branch에 merge하기

✓ Rule

1. master branch를 보호하기 위해 master branch에서 최대한 작업을 줄이고 새로운 branch에서 작업한 뒤에 merge하는 방식 사용
2. 여러 경우의 수를 둔 다음, branch를 만들고 테스트 해보기

목표 - 개선/추가 사항 및 기대효과

1. 오류 개선

A. 그림자 오류 개선

- Base Source에서 상황에 따라 그림자가 이상하게 발생하는 현상 개선하기
- ➔ 게임 중 그림자 오류로 인한 혼란 및 불편함 제거

2. 인터페이스 향상

B. 각 상황에 맞는 사운드 추가

- 레벨, 콤보, 게임모드 등 다양한 환경에 맞는 사운드 추가
- ➔ 게임을 진행하는 동안 몰입감과 긴장감을 높여줌

C. 메뉴 추가

- 새롭게 추가하는 여러 기능에 맞는 메뉴를 만들어 낸다.
- ➔ 게임 모드 선택, 랭킹 확인 등 추가 기능에 대한 접근 편의성 향상

D. 블록 이동 간편화

- 블록을 움직일 때, 좌우로 이동할 때, 꺾 누르고 있으면 빠르게 이동하도록 하기
- ➔ 게임 진행 중 편리함을 제공하고 답답함을 없애줌

E. 디자인 개선

- 단조로운 디자인에서 벗어나, 눈을 즐겁게 해주는 디자인으로 개선
- ➔ 게임적인 요소 외에서도 사람들이 재미를 느끼도록 해줌

F. 랭크 제도 추가하기

- 각 모드의 플레이 점수를 기록하고, 다른 사람들의 기록과 비교 가능하게 함
- ➔ 경쟁구도를 만들어 게임에 더 몰입하게 함

3. 기존 모드 개선

G. 기존 모드 콤보 기능 추가

- 블록을 일정 시간 안에 연속으로 파괴하면 콤보가 쌓이고 추가점수를 주는 시스템
- ➔ 콤보를 빨리 쌓기 위해 몰입하게 하여 박진감 있는 플레이를 이끌어냄

H. 기존 모드 Level에 따른 난이도 변화

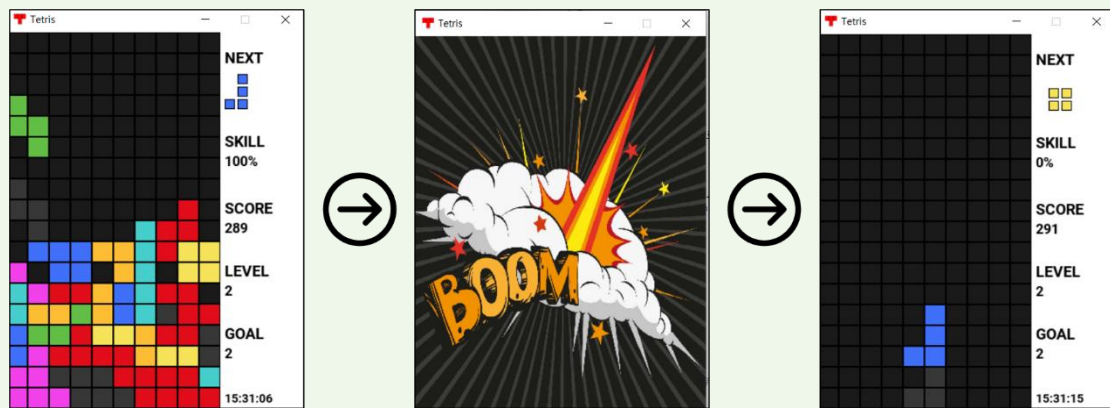
- level이 올라감에 따라 블록이 내려오는 속도를 빠르게 하여, 난이도를 조절
- ➔ 게임이 진행될수록 박진감 있게 하며 쉬운 난이도로 인한 무한 플레이를 방지

I. 기존 모드 스킬 제거

- 게이지가 채워지면 모든 블록을 없애는 기존 스킬 제거

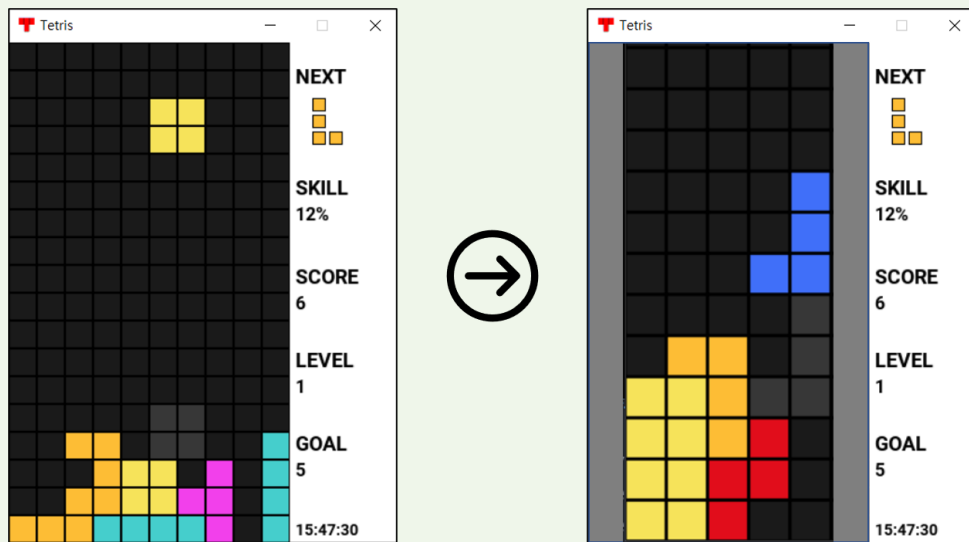
➔ 플레이 도중 균형이 깨지는 스킬로 인한 흥미 하락 및 무한 플레이 방지

(기존 스킬 사진)



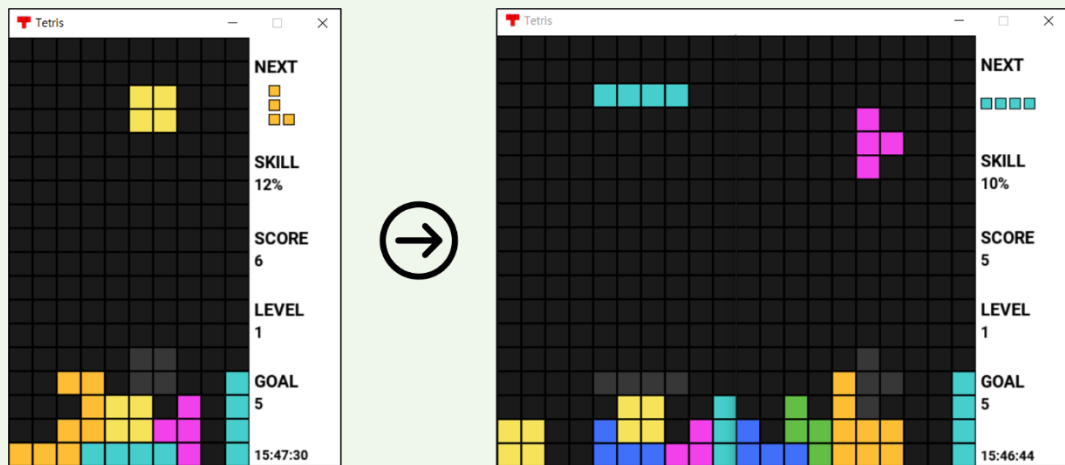
4. 새로운 모드 추가

J. 미니 게임 모드 추가



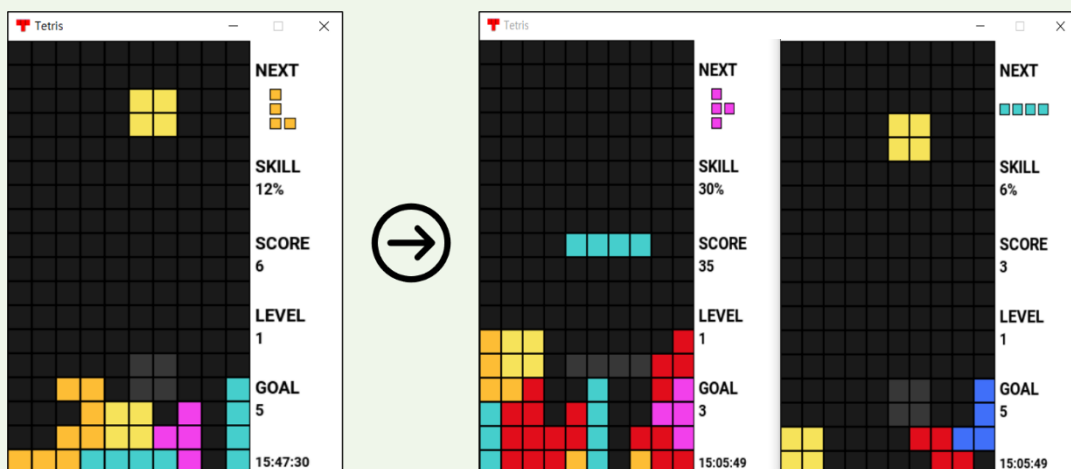
- 테트리스 board 칸 수를 5(~7)칸으로 줄이기
- 직접 플레이 해보면서 board 높이 조정 또는 블록 종류 변화 등을 통해 난이도 조절 예정

K. Two Hands 모드 추가



- 블록이 두 개씩 내려오고, 양손으로 블록을 조정하여 플레이하는 모드 추가
- 두 블록이 모두 왼쪽 끝부터 오른쪽 끝까지 이동 가능하며 서로 통과할 수 없음

L. AI 모드 추가

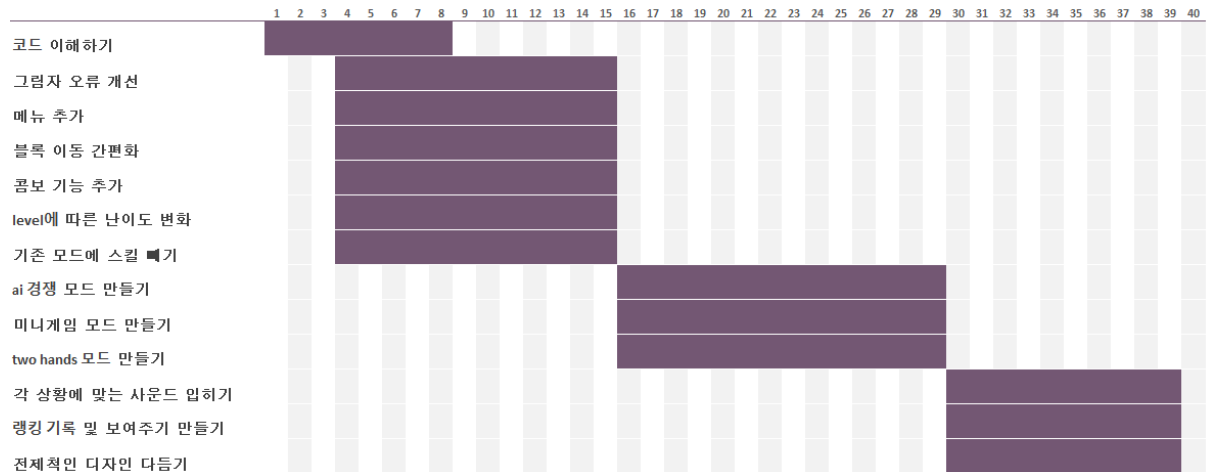


- 매우 느린 속도로 시작하여 점점 속도가 빨리지는 AI와 대결하는 모드
- 점수가 따라잡히면 패배하는 방식

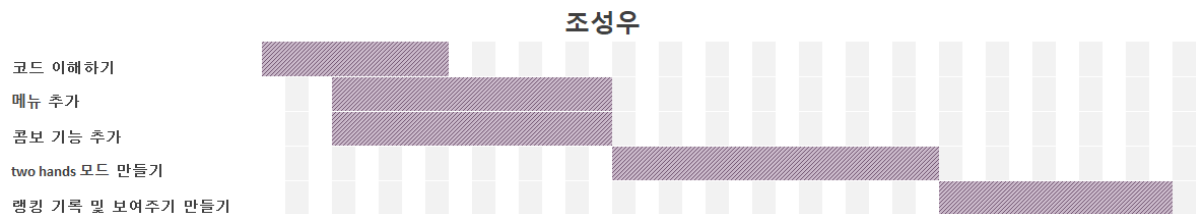
➔ J, K, L 여러 게임 모드를 추가하여 게임 플레이의 다양성 증가시키고 이에 따라 유저들의 흥미를 유발함

프로젝트 일정 및 역할 분담

- 프로젝트 일정 (프로젝트 시작일 : 11/1)

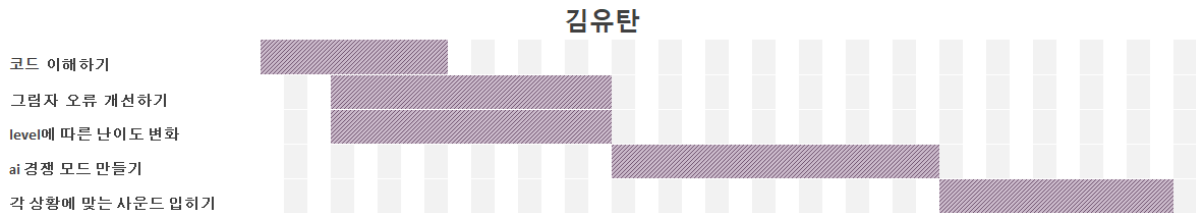


- 역할분담



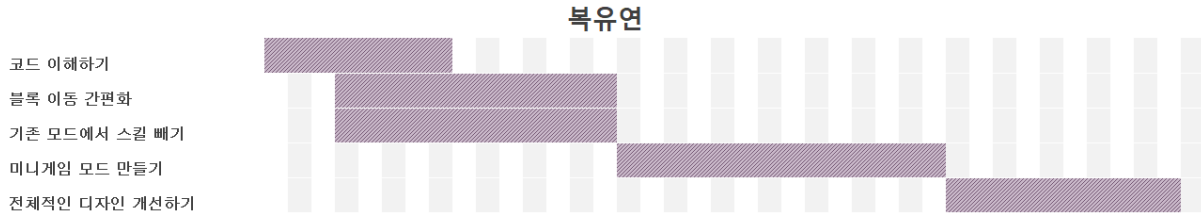
코드 이해하기/메뉴 구현/폼보 기능 추가/twohands 모드/랭킹 추가

- ✓ 기존 코드 이해하기
- ✓ 후에 만들 새로운 모드들에 맞게 메뉴 구현하기
- ✓ 기존 모드에 폼보 기능 추가하기
- ✓ 기존 모드를 응용한 양손으로 play 하는 two hands 모드 만들기
- ✓ 랭킹을 기록해주고, 기록된 랭킹을 보여줄 수 있는 인터페이스 추가하기



코드이해/오류 개선/난이도 설정/ai 모드/사운드

- ✓ 기존 코드에 대한 이해하기
- ✓ 기존 코드의 그림자 오류 개선하기
- ✓ 기존 모드에 level 별로 난이도(속도) 설정하기
- ✓ 기존 모드를 응용한 ai 경쟁 모드 만들기
- ✓ 각 모드들이 다 만들어지면 각 상황에 맞는 사운드 입히기



코드 이해/블록이동 간편화/기존 모드 스킬 없애기/미니게임 모드/디자인

- ✓ 기존 코드에 대한 이해하기
- ✓ 블록이동 간편화 시켜주기
- ✓ 기존 모드에서 스킬 없애주기
- ✓ 5칸짜리 미니게임 모드 만들기
- ✓ 전체적인 디자인 다듬기

System Structure

