

OSSP

프로젝트 제안서

과목명 | OSSP

학과 | 컴퓨터공학과

팀 명 | NHL(11 조)

팀 원 | 2016112171 이주영

2016112127 신지혜

2016112186 고노윤

담당 교수님 | 김동호 교수님

제출일 | 2018.04.15(일)

1. Research Problem Statement or Question

이번 프로젝트를 통해 우리 조가 진행할 연구의 주제는 작년 4 조가 진행한 테트리스를 분석하고 좀 더 발전시킨 형태로 구현하는 것이다. 우리는 기존 코드를 보고 5 가지의 문제를 생각했다.



[게임 진행 중]

[게임 종료]

1) 난이도 조절

기존의 코드는 별도의 난이도가 없다.

2) 재시작 여부

게임 오버 시 점수 출력 후 재 시작 여부 확인없이 바로 종료된다.

3) 디스플레이

별도의 경쟁 상대없이 사용자 모드의 화면만 보여주며 진행된다.

4) 고스트

블록이 내려오며 고정될 위치를 보여주지않는다.

2. Project Goal and Objectives

※ 1) ~ 6) 에 있는 소 제목은 Goal 을 나타내고 그 아래 자세한 설명이 Objectives 를 나타낸다.

1) 경쟁 상대 구현 (인공지능)

- 대결 형태의 포맷 구성: 사용자와 컴퓨터가 대결할 수 있는 형태의 테트리스 게임을 구현한다. 두 유저가 포함된 화면을 구성하고, 각각 같은 화면에서 독립적으로 실행될 수 있도록 한다. 한 유저의 게임이 종료되면 상대 유저가 이기는 것으로 게임이 진행되고, 각각의 Next block 은 독립적으로 랜덤하게 나오게 한다.
- 상대방(컴퓨터)의 인공지능 기능 구현: 상대방 유저 즉, 컴퓨터는 인공지능으로 구현한다. 인공지능의 목표는 최적의 장소를 찾아 그 자리에 위치할 수 있도록 하는 것으로 설정한다. 최적을 찾을 때는 가능한 모든 위치 중 최대 접촉 면적을 찾아 그 위치를 가장 좋은 곳으로 본다. 이 때, 레벨마다 이 기준을 달리하여 사용자의 수준에 맞출 수 있도록 한다.

2) Level 시스템

- 상대방(컴퓨터)의 난이도 선택: 상, 하 난이도를 만들어 유저의 수준에 맞게 선택하여 게임을 진행할 수 있도록 한다. 각각의 난이도에서는 기준을 적절히 조절해야 한다.
- 난이도 조절의 기준 선정: 앞에서 언급한 접촉 면적의 수 또는 속도에 변화를 두어 난이도를 조절하는 방법을 이용한다.

3) 고스트 구현

- 떨어질 정확한 위치를 시각화: 기존의 방식은 떨어지는 위치에 표시를 해주는데, 여기에서 더 나아가 떨어지는 블록의 종류에 따라 어떻게 떨어질지 구체적으로 시각화 해준다.

4) 디자인 그래픽 향상

- 사용자와 인공지능의 화면 구현: 기존에 사용자 것만 있었던 테트리스 화면에서 인공지능 컴퓨터의 화면을 동일한 형태로 구현해 동시에 보여준다.
- Intro page 추가: 게임의 시작 화면을 별도로 구현한다.
- 가독성 증가 등 그 외 시각적인 효과들

5) 게임 종료 후 재시작 기능 (quit, restart)

- 다시 실행할 수 있는 기능 추가: 기존에 한번만 실행하고 종료되는 프로젝트에 다시 시작하는 선택지를 추가한다.

6) 전체 시스템 향상

3. Significance of the Project and Background

최근 오픈소스에 대한 관심과 사용이 증가함에 따라 분산 버전 관리 프로그램인 Git 과 그 Git 데이터를 온라인에 저장해주는 Github 에 대한 활용 능력이 요구되고 있다. 또한 개인의 역량뿐 아니라 여럿이서 협력하며 프로젝트를 진행할 수 있는 능력도 필요하기 때문에 이번 프로젝트를 잘 수행하는 것이 굉장히 중요하다.

이 주제를 선택한 가장 큰 이유는 이미 Github 에 많은 테트리스 게임의 오픈소스들이 공개되어있어 기회가 된다면 한 번 다뤄보고 싶었기 때문이다. 그리고 코드를 분석해보니 팀원들에게 가장 익숙한 언어인 C 언어를 사용하고 내용적인 측면에서 조금 더 추가해 보고싶은 기능들이 있다는 점에서 이 프로젝트를 선택하게 되었다.

4. Experimental or Project Design

1) 작업 계획 설계 및 설명

- 오픈 소스를 이용하여 진행하는 프로젝트라는 점을 고려하여 Github 의 오픈 소스를 참고하도록 한다.

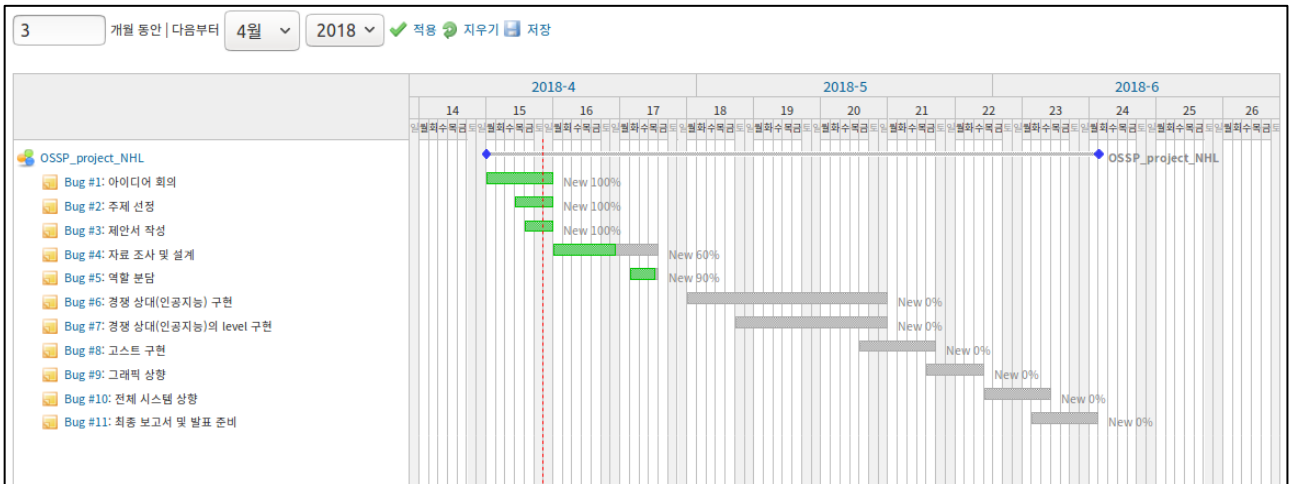
- 두 유저를 기본으로 하는 프로젝트이고 C 를 기반으로 프로그래밍하기 때문에 두 개 이상의 스레드 또는 프로세스 구현해야 한다. 이를 위해 fork() 등의 방법을 이용하여 독립적으로 두 유저를 운영 할 수 있도록 한다.

2) 구성원 및 역할 분담

NHL 팀의 구성원은 이주영, 고노윤, 신지혜로 총 3 명이며 각각의 역할 분담은 다음과 같다.

- 이주영: 인공지능 테트리스 및 Level 시스템 구현, 고스트 구현, 결과 보고서 작성
- 고노윤: 인공지능 테트리스 및 Level 시스템 구현, 시작 화면, 결과 화면, 듀얼 화면 구현, 최종 ppt 제작
- 신지혜: 인공지능 테트리스 및 Level 시스템 구현, 고스트 구현, 결과 보고서 작성

5. Project Timeline

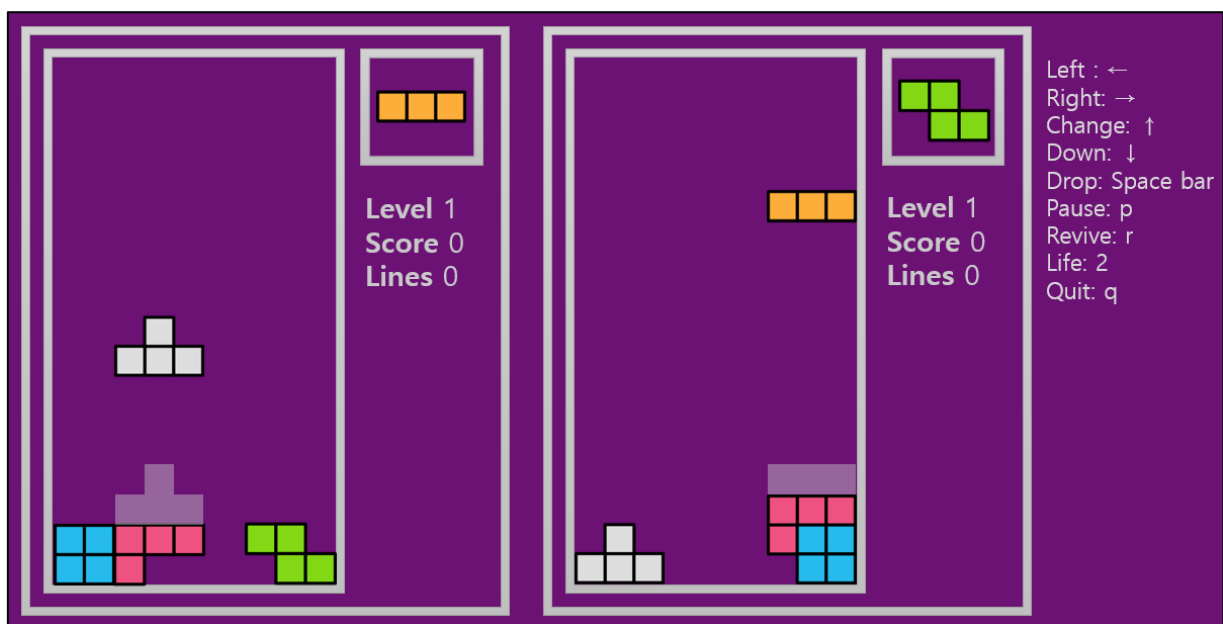


6. Anticipated Results

확장한 테트리스는 사용자 모드 하나로만 진행이 되고 별도의 난이도가 없는 기존 테트리스와 다르게 사용자 모드와 인공지능 모드 총 두 개의 모드가 동시에 진행이 되면서 경쟁하는 시스템이다. 인공지능의 모드는 상, 하로 조절이 가능하며 그 난이도는 사용자가 결정할 수 있다.

그리고 기존 테트리스는 블록이 고정될 위치를 보여주지 않고 좌우이동에 따라 어느 위치로 내려가는지 경로를 보여준다. 하지만 확장한 테트리스는 블록이 내려갈 경로만이 아닌 고정된 후의 블록 형태를 미리 보여준다. 또 기존에는 게임 오버 시 점수를 출력하고 바로 종료되지만 확장된 테트리스는 재 시작의 여부를 물어본 후 종료한다.

추가적으로 게임을 진행하는데 있어 가장 중요한 정보인 Level, Score, Lines 를 bold 체로 강조해주고 조작키에 대한 설명은 오른쪽에 따로 출력해준다.



[예상하는 결과 화면]

	기존 내용	확장할 내용
난이도 조절	블록의 속도가 항상 일정하며 별도의 난이도가 없다.	경쟁할 인공지능의 난이도를 상, 하로 조절 가능
디스플레이	사용자 모드 화면 하나만 보여준다.	하나의 사용자 모드, 다른 하나는 인공지능 모드로 듀얼 화면을 보여준다.
재 시작 여부	게임 오버 시 점수 출력 후 종료	게임 오버 시 점수 출력과 함께 재 시작 기능 추가
고스트 기능	블록이 고정될 위치를 미리 보여주지 않는다.	블록이 고정될 위치를 미리 보여준다.

[현재 코드와 확장할 내용 비교]

7. Project References

- CSID_DGU 에 있는 2017-2-OSSP-Dongguk_Banjeom-4 의 프로젝트 참고
- Git, Github 의 개념 관련 자료: https://blog.naver.com/azure0777?Redirect=Log&logNo=220812405201&proxyReferer=https%3A%2F%2Fsearch.naver.com%2Fsearch.naver%3Fwhere%3Dnext%26sm%3Dtop_hy%26fbm%3D1%26ie%3Dutf8%26query%3Dgit%2Bgithub