

Tetris Version Up

프로젝트 제안서

팀장	2015110792	정치외교학과	융합소프트웨어	김정현
팀원	2019112093	정보통신공학과	데이터사이언스	박상준
팀원	2019112101	정보통신공학과	데이터사이언스	추예진

Index

1. Outline

- 1.1 Why Tetris?
- 1.2 SWOT
- 1.3 Goal
- 1.4 Base Source

2. Development Environment

3. Objectives

- 3.1 Main Targets
- 3.2 Detailed Targets

4. Team

- 4.1 Timetable
- 4.2 Team Rule

5. Final

- 5.1 Outcome
- 5.2 Expectation

1. Outline

1.1 Why Tetris?

게임이란 무엇인가?

모든 게임은 근본적으로 재미를 추구한다. 모든 RPG 장르의 모티브가 된 전설적인 Ultima Online의 개발자 Raphael Koster의 재미이론에 따르면, 게임의 rule을 이해하고 숙련도를 쌓는 과정은 학습과정이다. 게임에서 얻는 재미는 근본적으로 ‘학습’이 주는 쾌감에 있다.



Tetris와 기존 프로젝트의 문제점

일반적으로 학습에서 얻는 쾌감은 학습의 난이도가 매우 낮을 때 즉 단조로움이 느껴질 때 쉽게 사라진다. 그러나 학습에서 얻는 쾌감이 학습의 난이도와 정비례하는 것은 아니다. 이해하기 어려운 복잡한 rule과 숙련도를 쌓는데 많은 시간이 걸리는 학습의 재미는 반감된다. 학습의 쾌감을 최대화 하는 좋은 게임이 되려면 적절한 학습난이도를 갖춰야 한다. 게임 개발자는 플레이어에게 너무 쉬워서 지루하지도 않은 너무 복잡해서 거리감이 들지도 않는 도전할 만한 과제를 제시해야 되는 셈이다.

이러한 시각에서 Tetris와 이를 이용한 기존 프로젝트를 살펴봤을 때, 거의 모든 사람이 Tetris의 Rule을 선형적으로 학습했다는 점은 Tetris의 훌륭한 장점이자 치명적인 단점으로 작용한다. 따라서 Tetris를 조금 더 좋은 게임, 흥미로운 게임으로 Version Up 하기 위해서는 플레이어가 흥미를 느끼도록 도전할 만한 학습거리를 제시해야 한다.

따라서 Tetris의 rule이 보편적으로 이해되고 있다는 장점을 기반으로 단조로움을 극복할 수 있는 적절한 학습 과제를 제시해야 한다. 이를 위해서는 기존의 rule을 기반으로 변화를 준 새로운 mode 를 추가하여 Tetris가 가진 장점은 살리고 단점을 보완하는 방식으로 진행하여야 한다.

기존 Tetris Version Up 프로젝트들은 단순히 UI를 변경하거나 AI와의 대전모드 등의 변경에 불과했다. 근본적인 rule의 수정이 없었기 때문에 Tetris가 가진 익숙함을 낮설고 흥미로운 학습거리로 변경하여 제시하는데 부족하다.

Why Tetris?

앞서 서술된 바와 같이 Tetris는 훌륭한 장점과 단점을 동시에 가지고 있다. 따라서 기존 Open Source를 이용하여 Version Up 될 가능성이 충분히 존재한다. Open Source를 이용한 개발이 백지에서 창조하지 않고 기존의 창작물을 기반으로 발전시킨다는 관점에서, 기존 Tetris의 Opens Source 작업물 들을 비판적인 관점에서 접근하고 이를 Version Up 하는 프로젝트는 이번 수업의 주제에 매우 적합하다.

1.2 SWOT

Strength	Weakness
<ul style="list-style-type: none"> · 기능별로 Module화가 잘되어 있기 때문에 개발이 용이하다 · Tetris의 Rule을 대부분의 사용자가 이미 알고 있다 · 팀원들에게 가장 익숙한 python을 통해 개발되어있다 	<ul style="list-style-type: none"> · Tetris의 Rule에 대한 변경이 거의 없거나 단조롭다 · UI 변경과 PVP모드 같이 근본적인 Rule의 변화의 부재는 사용자들의 흥미를 저하시킨다 · 팀원들에게 익숙하지 않은 pygame 라이브러리를 통해 개발되었다
Opportunity	Threat
<ul style="list-style-type: none"> · 새로운 Rule을 제시하는 Mode 추가를 통해 적절한 학습거리를 제시함으로써 사용자들에게 새로운 흥미를 유발할 가능성이 있다 · 팀원들에게도 Tetris의 Rule은 이미 익숙하기 때문에 빠른 시간안에 효율적으로 자유로운 아이디어를 생각해 낼 수 있다 	<ul style="list-style-type: none"> · 팀원들이 제시한 새로운 Rule이 사용자들의 흥미를 이끌 만한 학습의 기회를 제공하지 못할 수 있다. (너무 난해하거나 단조로움으로 인해) · 팀원들의 숙련도 부족으로 아이디어를 구체적인 결과물로 이끌어 내는데 까지 기술적인 문제를 겪을 수 있다.

1.3 Goal

본 팀은 게임의 근본적인 재미가 학습의 쾌감이 있음으로 정의하고 이러한 시각으로 기존 Tetris와 이를 바탕으로 하는 Open Source를 분석한 바, Tetris가 가지고 있는 최고의 단점이자 장점인 ‘익숙함’을 해결하고 있지 못함을 발견했다.

이를 보완하기 위해서는 새로운 Mode 추가를 통해 사용자가 학습의 쾌감을 느낄 수 있는 새로운 Rule을 제시해야 한다. 따라서 본 팀은 회의를 통한 Brain Storming을 통해 단순히 UI이나 기존 Rule이 변경되지 않는 추가기능 보다는 ‘익숙함’과 ‘낯섦’이 공존할 수 있는 새로운 Mode를 추가하고자 한다. UI등은 게임플레이에 최적화 될 수 있도록 가능한 simple한 형식을 추구하고자 한다.

프로젝트를 통해 Linux 환경에 익숙해지고, Github을 이용하여 협업을 진행하며 협동성과 Tool 활용 능력을 증진시킨다. Open Source를 기반으로 기존 Tetris에 재미를 더하는 프로젝트를 진행하면서 Open Source에 대한 이해도와 Open Source를 이용한 개발능력 향상을 궁극적인 목표로 한다.

1.4 base source

pytris

<https://github.com/injekim/PYTRIS>

· License : MIT-License

· config

pytris.py : main함수 및 각종 기능

mino.py : 블록 모델

assets : 폰트, 이미지, 사운드

2. Development Environment

Language: python 3.7

OS: Ubuntu 16.04

Tool: Visual Studio Code

협업: Github

기록: Notion, Google 공유문서

회의 환경: Webex, 카카오톡, 오프라인

3. Objectives

3.1 main

추가할 Mode는 아래와 같다.

- Screen Rotate Mode
- Quarter Mode
- Blackout Mode

Screen Rotate Mode

- 사용자가 바라보는 방향에서 블록은 회전하지 않았으나 블록을 제외한 나머지 화면이 회전을 통해 블록이 회전한 것과 동일한 효과를 주는 동시에 사용자에게 익숙하지 않은 시각을 줌으로서 난이도를 올리는 Mode이다.
- 기존에는 배경이 따로 존재 하지않고 게임화면이 Screen 을 다 채우는 구조이다. 그러나 회전을 자연스럽게 하기 위해서는 게임화면 보다 더 넓은 배경화면을 삽입하고 그 안에 게임화면이 돌아가는 액자식 구성이어야 한다. 배경은 단색이 아닌 방향성을 인지 시킬 수 있는 화면으로 화면이 전환되는 효과를 플레이어가 감지 할 수 있도록 해야 한다.

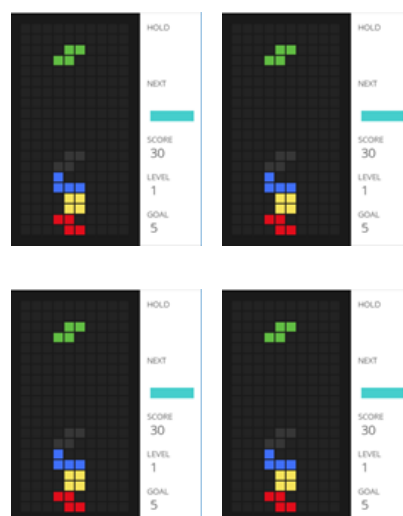
- * Rotate 함수를 사용해서 배경을 포함한 image 객체들을 모두 회전시킨다. pygame은 이미지의 회전만 지원하므로 한개의 객체를 통해서는 사용자에게 편리한 입력키를 구현하기 힘들다. 따라서 4개의 방향에 대해 플레이어가 바라보는 방향을 기준으로 블록이 움직이게 따로 객체를 만들어 적용시킬 필요가 있다.



화면을 보는 사용자의 입장에서 현재 떨어지는 초록색 블록은 회전하지 않았으나 블록을 제외한 배경이 회전하며 실질적으로 블록이 회전한 효과를 줄 수 있다.

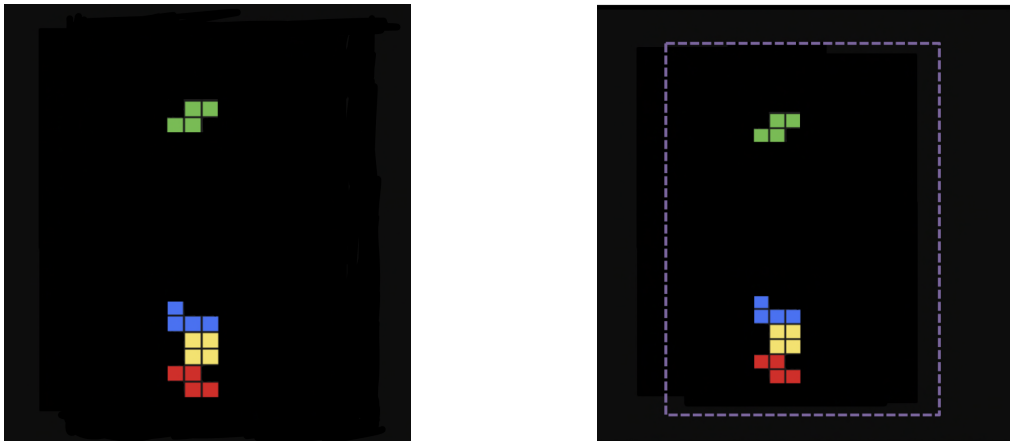
Quarter Mode

- 네 개의 Tetris를 동시에 플레이하는 Mode이다.
- Screen에서 네 개의 Tetris 화면이 동시에 진행된다. 특정 키를 선정해 pygame module의 Key Down이 일어날 때, 순서에 맞게 다음 Tetris 화면 조작이 가능하게 한다. 동시에 여러 게임을 진행하는 모드이기에 블록이 떨어지는 속도가 난이도에 큰 영향을 준다. 적절한 난이도를 위해 기존 게임의 블록이 떨어지는 속도보다 느리게 시작해서 게임이 진행됨에 따라 순발력을 요하는 수준으로 상승시킨다. 구체적인 속도에 대해서는 Mode 구현 후 Demo test를 통해 설정하도록 한다.
- * 지정된 키를 통해서만 다음 테트리스 조작으로 넘어갈 수 있도록 구현한다. 사용자의 혼란을 방지하기 위해 해당 Tetris가 조작되고 있음을 알려주는 GUI를 추가할 필요가 있다.



Blackout Mode

- 기존 테트리스와 rule은 동일하나 블록을 제외한 모든 screen의 색을 검정색으로 통일한다. 블록을 제외한 나머지 부분을 검정색으로 보이게 함을 통해 사용자는 블록의 위치와 게임이 진행되는 창을 범위를 알아보기 힘들며 이를 통해 기존 rule과 동일한 rule을 사용하여 체감 난이도를 올릴 수 있다.
- * 블록 외에 격자 무늬로 표시된 게임 플레이 범위와 스코어보드, 배경 등을 모두 검정색 배경으로 변환한다.



위 그림의 왼쪽이 실제로 사용자가 보게될 화면이며 오른쪽 사진에서 알 수 있듯이 실제로는 테트리스가 플레이되고 있는 부분(보라색으로 표시된 사각형 내부)이 존재하나 이를 보이지 않게 함을 통해 사용자에게 블록의 위치를 혼동시켜 난이도를 높인다.

3.2 detail

인터페이스

- 인터페이스는 게임구동의 최적화를 위해 가볍게 유지하도록 한다.
- 유저 인터페이스는 가능한 심플하게 구현한다.
- 게임 구동을 위해 시각적인 효과가 필요한 경우가 아닌 이상 단색을 지향한다
- 폰트 또한 가장 정적이고 단조로운 폰트를 지향한다
- 효과음이 아닌 배경음악은 지양하도록 한다.

4. Team

4.1 Timetable

	April			May				June		
	4/11-4/17	4/18-4/24	4/25-5/1	5/2-5/8	5/9-5/15	5/16-5/22	5/23-5/29	5/30-6/5	6/6-6/12	6/13-6/19
코드이해										
Rotate Mode										
Quatrefoil Mode										
Blackout Mode										
UI										
Demo 및 오류 수정										
발표자료 작성 및 발										

- 3가지 모드 구현을 각각 하나씩 맡아 독립적인 개발 환경에서 기능 개발을 구현하는 것이 작업의 집중도를 높일 수 있을 것으로 판단된다.
- 발표는 모든 조원이 균등하게 시간을 분배하여야 되므로 발표전에 발표 순서를 정하고 정해진 형식을 공유하고 각자 파트를 합쳐 취합하는 것으로 한다.
- 모드 구현 외에 인터페이스 개선 부분은 팀원들이 세부적으로 나누어 함께 작업한다. 따라서 개인의 Gant Chart는 팀 전체와 동일하므로 따로 명시하지 않는다.

4.2 Team rule

Github Link

<https://github.com/CSID-DGU/2021-OSSPC-CHUPARKIM-5>

자료 관리

모든 자료는 Github를 통해 공유하도록 한다. 참고할 만한 자료는 구글 공유문서를 통해 업로드 날짜, 공유 목적을 포함한 짧은 메모와 함께 업로드한다. 구글 공유문서에 명시한 후 Github Reference Folder에 업로드 하도록 한다.

개발환경

- Ubuntu 16.04
- python 3.7 이상

Github Rule

- Commit message
 - 제목과 본문을 빈 행으로 구분한다
 - 제목을 50글자 내로 제한
 - 제목 첫글자는 대문자로 작성

제목 끝에 마침표 넣지 않기

제목은 명령문으로 사용하며 과거형을 사용하지 않는다

본문의 각 행은 72글자 내로 제한

어떻게 보다는 무엇과 왜를 설명한다

Reference <https://blog.ull.im/engineering/2019/03/10/logs-on-git.html>

- Branch

· Master 배포 Release(Prod) 버전의 소스가 들어있는 branch 기본적으로 github 저장소를 생성하면 있는 branch이다. 배포이력을 관리하기 위한 용도로 사용한다.

· Develop 개발버전의 소스가 들어있는 branch 일반적으로 Master branch에 병합하기 전 최종 개발버전의 소스가 들어있다. 다음 Release될 버전의 소스라고 생각하면 된다.

· Feature 기능을 개발중인 branch 개발자들이 기능개발을 위하여 생성/이용 하는 branch이다. 개발이 완료되면 develop와 병합하여 다른 사람들과 공유한다.

· Hotfix Master branch의 오류사항을 수정하는 branch Feature > Develop > Master의 병합 순이 아니라 Master에서 급하게 수정해야하는 경우에 사용한다. Master에서 직접 branch를 분기하여 생성하며 수정 후 Develop가 아닌 Master에 병합하여 배포한다.

Reference <https://www.kyungyeon.dev/posts/13>

Code

변수명, 함수명은 자유롭게 정한다. 하지만 새롭게 선언한 경우 반드시 주석처리를 통해 해당 변수나 함수의 목적을 밝히도록 한다.

Merge

Merge하기 위해서는 항상 pull request를 거친 후 merge한다. Pull request가 있는 경우 해당 Issue를 보고 반드시 피드백을 남긴다.

회의 시간

회의는 매주 수요일 수업시간과 수업시간 이후 그리고 회의 결과에서 도출된 일요일 2시에 정기적으로 진행한다. 회의 시작 후 10분간은 현재 작업상황에 대해 개략적으로 이야기하고 시작한다. 회의는 특별한 Issue가 없어도 반드시 참석하는 것으로 하고, 회의 후에는 매번 스케줄을 re-mapping 한다. 그리고 이를 반드시 구글 공유문서에 기록한다.

5. Final

5.1 Outcome

본 프로젝트는 기존코드에 대해 UI 변화나 Tetris의 Rule에 변화가 없는 진부한 수정보다는, 좋은 게임이란 무엇인가? 라는 측면에서 Tetris의 Rule을 수정하여 보다 나은 게임이 될 수 있는 결과를 창출하고자 한다.

- Screen Rotate Mode

- Quarter Mode

- Blackout Mode

따라서 회의 결과 Tetris의 장점을 살리되, 새로운 학습 과제를 제시하여 게임의 흥미를 유발 할 수 있도록 도출된 위의 새로운 세가지 Mode를 구현하는데 충실히 한다.

개발과정에서 필수적으로 구현해야 하는 Menu 선택과 같은 UI 부분은 최소한 simple하고 직관적으로 만들어 게임의 근본적인 목적에 부합하는 결과를 도출한다.

5.2 Expectation

본 프로젝트는 단순히 수업과제를 진행하기 위해서 설계되지 않았다. 수업에서 배운 공학적 설계 방식을 충분히 고려하여 기존 Open Source에 대한 문제 제기부터 시작하여, 이를 분석하고 Brain Storming을 거쳐 아이디어를 취합하는 과정을 거쳤다. 따라서 본 프로젝트에 대한 기대는 다음과 같다.

- 궁극적으로 처음에 프로젝트 선정 초기에 Tetris에 제기한 문제 : ‘학습요소가 적어 게임의 흥미가 떨어진다’ 를 해결하여 Tetris를 익숙하고도 흥미로운 게임으로 발전시킨다.
- Open Source 프로젝트를 계획하고, 진행하고, 완성함으로써 단순히 수업을 듣는 학생이 아니라 개발자로서 Open Source의 세계에 첫 발을 디딘다.
- 아이디어를 구상하고 이를 구현하는 과정을 거침으로써 팀원 전원의 커리어가 향상될 것으로 기대된다.