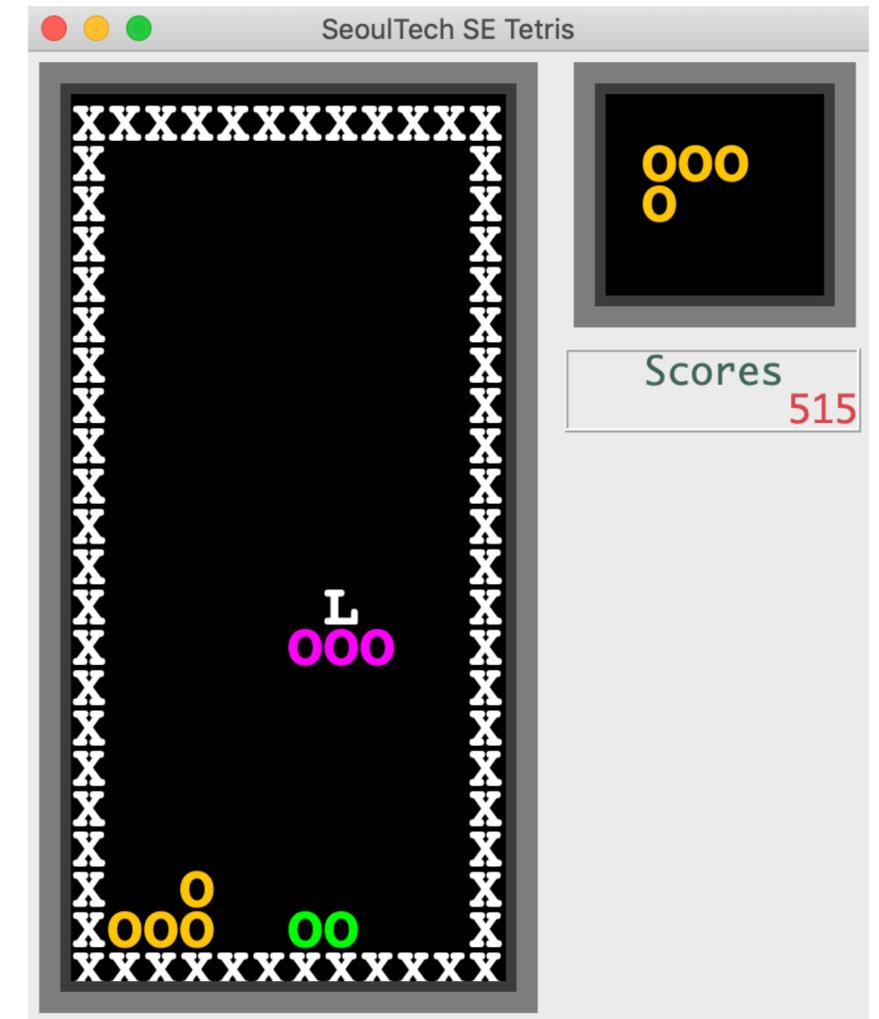


Requirements 2

Software Engineering Team Project
Spring 2024, SeoulTech

아이템 모드 개발

- 2차 요구사항의 핵심 내용은 게임에 아이템 모드를 추가하는 것과 일반 모드를 일부 수정하는 것입니다.
- 직접적으로 제시되는 요구사항 외에도, 이를 적절히 구현하고 높은 품질의 게임을 만들기 위해 같이 수정되어야 할 부분은 직접 파악하여 처리해야 합니다.
 - 또한 요구되는 비기능적 요구사항 또한 충분히 만족시킬 수 있도록 구현 및 테스트를 진행합니다.
- 명확히 언급되지 않은 부분에 대해서는 요구사항과 어긋나지 않는 한 자유롭게 구현하여도 좋습니다.
 - e.g.) 게임 화면 구성, UI설계, 추가적인 게임 정보 표시 등.



기존 방식의 변경사항

- 기존의 일반 게임 모드에 난이도 설정을 추가합니다.
 - 설정에서 easy, normal, hard의 3가지 난이도를 선택할 수 있습니다.
 - easy 난이도에서는 I형 블럭(OOOO)이 20% 더 자주 등장하고, hard 난이도에서는 20% 덜 등장합니다.
 - 줄 삭제에 따라 증가하는 블럭이 떨어지는 속도 또한 easy에서는 20% 덜 증가하고, hard에서는 20% 더 증가합니다.
8% 12%
 - 난이도에 따른 추가 규칙이나 점수 부여 등은 자유롭게 결정하세요.
 - 스코어 보드에서는 어떤 난이도에서 점수를 얻은 것인지 구분하여 확인할 수 있어야 합니다.
- 삭제되는 줄을 잘 구분할 수 있도록 삭제되는 애니메이션을 추가합니다.
 - e.g.) 삭제되는 줄이 하얀색으로 한 번 반짝이고 삭제 등.

확률에 따른 블록 생성

- 요구사항 구현을 위해서는 각 블록의 생성확률을 다르게 처리해야 합니다.
- I형 블록이 20% 더 등장하게 하려면, 다른 블록의 등장확률 가중치가 10이라고 할 때 I형 블록은 12를 가져야 합니다.
- 이를 구현하기 위해 Roulette Wheel Selection을 사용할 수 있습니다.
 - 이해 및 효율적인 구현을 위해 다음의 링크와 논문을 참조하세요.
 - 논문은 e-class에 rws.pdf로 업로드 되어있습니다.
 - https://en.wikipedia.org/wiki/Fitness_proportionate_selection
 - Lipowski, A., Lipowska, D.: Roulette-wheel selection via stochastic acceptance. Physica A: Statistical Mechanics and its Applications, 2012

확률에 따른 블럭 생성

- 실제 확률에 따라 블럭이 생성되고 있는지 확인하기 위해 테스트를 추가합니다.
 - 최소 1,000번 이상 선택을 반복하여 설정된 확률과 큰 차이없이 블럭의 분포가 나타나는지 확인하는 단위 테스트(unit test)를 구현합니다.
 - 오차범위 +-5% 이내에서 블럭이 선택되고 있는지 확인할 수 있는 테스트를 작성하세요.
 - 실제 게임에서 블럭을 생성하는 코드가 이 테스트를 통과해야 합니다.
- 예를 들어 총 4개의 블럭이 있고, 그 중 한 블럭B의 선택 확률이 다른 것들의 2배라면,
 - 블럭 B가 선택될 확률은 40%이고, 나머지 블럭은 20%여야 합니다.
- 총 1,000번의 선택을 반복했다고 하였을 때, 블럭 B는 약 400번 정도, 나머지는 약 200번 정도 선택되어야 합니다.
 - 코드에 문제가 없는데 오차가 크다면 반복횟수를 늘려 보세요.

아이템 모드 추가

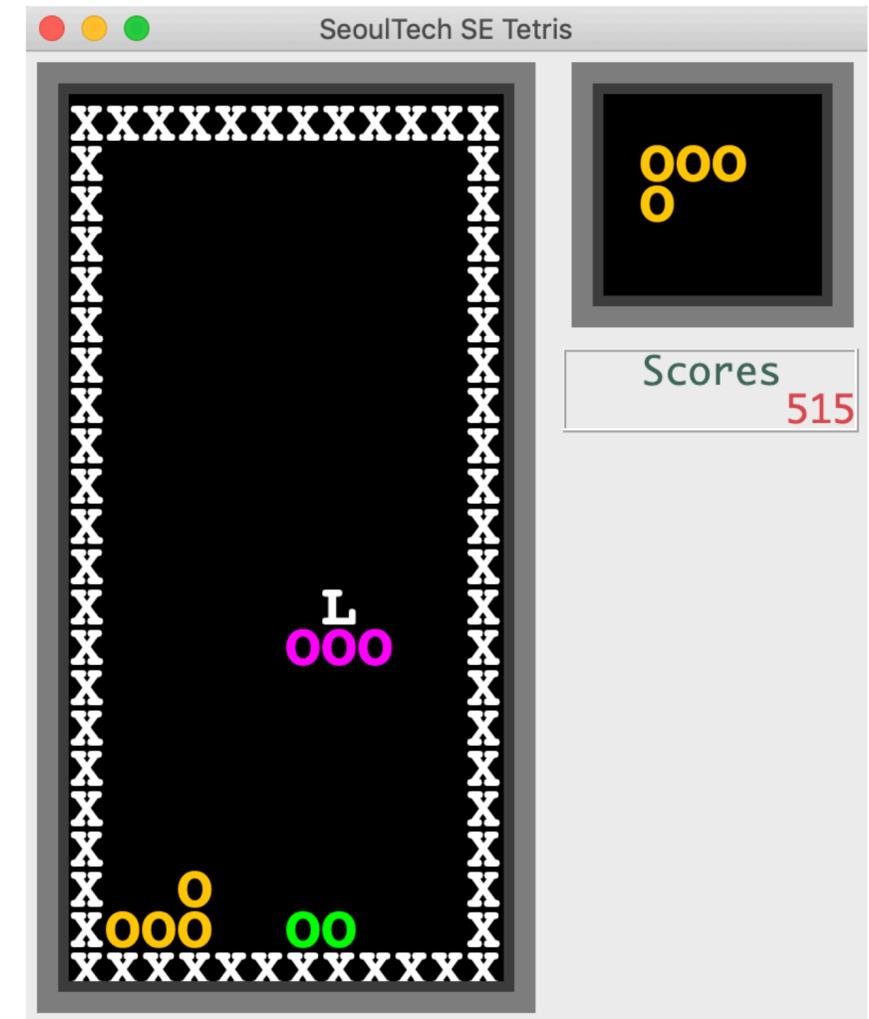
- 아이템 모드를 추가하기 위해서 시작 메뉴를 수정합니다.
 - 기존과 같은 게임을 시작할 수 있는 메뉴에 더하여,
 - 아이템 모드를 시작할 수 있는 메뉴가 추가되어야 합니다.
- 시작 메뉴에서 어떤 모드를 선택하느냐에 따라 게임 형태를 달리합니다.
- 스코어 보드에서도 기존 방식과 아이템 모드의 점수를 구분하여 보여줍니다.

아이템 모드 게임 방식

- 아이템 모드에서는 10개의 줄을 삭제할 때마다 아이템이 등장합니다.
- 오른쪽 화면처럼 블록의 일부에 아이템이 포함되는 경우도 있고,
 - 아예 블록 자체가 아이템인 경우도 있습니다.
- 10개의 줄이 삭제되면, 다음에 생성될 블록을 보여주는 부분에 새롭게 나타나는 블록은 반드시 아이템이거나 아이템을 포함해야 합니다.
- 아이템은 구분이 쉽도록 특징적인 모양을 하거나 기본 블록과 다른 문자를 사용해 표현합니다.
 - 각각의 아이템은 반드시 다른 문자를 사용해야 합니다.
- 요구사항에서 제시되는 2개의 아이템 외에도 각 팀별로 직접 구상한 아이템을 **최소 3개 이상** 추가합니다.

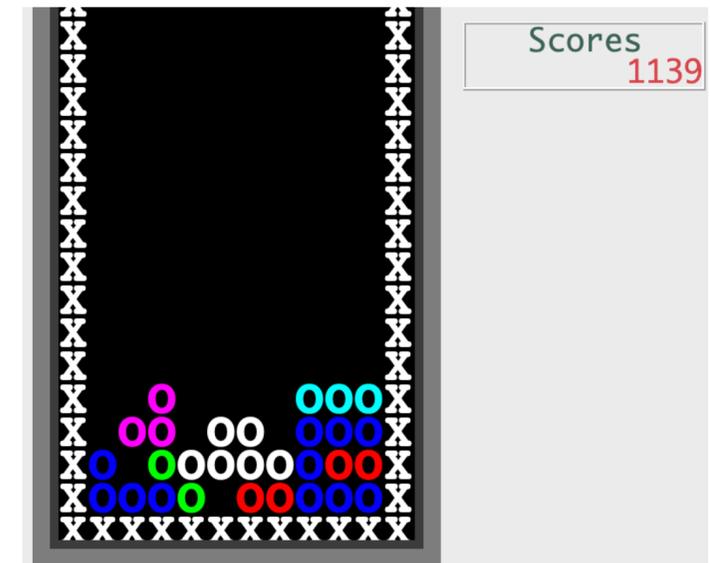
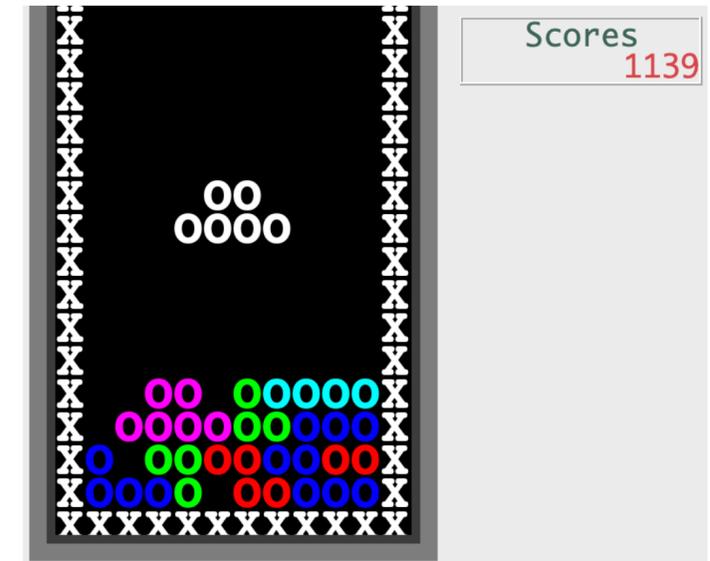
줄 삭제 아이템

- 줄 삭제(Line clear) 아이템은 기본 블록 내에 포함되며, 문자 'L'로 나타냅니다.
- 블록이 더 이상 움직이지 못하게 되면 (바닥이나 다른 블록에 닿으면),
 - 이 'L'이 위치한 줄을 해당 줄이 꽉 차 있지 않아도 삭제합니다.
- 이 아이템이 블록에 붙는 위치는 무작위로 정해져야 합니다.
 - e.g.) I형 블록의 경우 'OOOL', 'OOLO', 'OLOO', 'LOOO' 형태가 만들어 질 수 있고 이는 무작위로 결정.
- 아이템으로 인해 삭제되는 줄에 대해서도 기존의 방식대로 점수를 계산합니다.



무게추 아이템

- 무게추 아이템은 총 4칸의 너비를 지닌 아이템입니다.
 - ```
 OO
 OOOO
```
- 이 모양은 예시로, 너비가 4칸이면 어떤 형태나 문자를 사용해도 좋습니다.
- 해당 아이템은 떨어지면서 밑에 있는 모든 블럭들을 지우는 아이템입니다.
- 처음에는 자유롭게 좌우로 움직일 수 있습니다.
  - 하지만 일단 바닥이나 다른 블럭에 닿으면 더 이상 좌우로는 움직이지 못합니다.
- 기본적으로 이 아이템의 사용이나 지워지는 블럭 등에 대해서는 점수를 부여하지 않습니다.
  - 다만 원하는 경우 적절한 규칙에 따라 점수를 부여할 수 있습니다.



# 아이템 모드 관련 사항

- 아이템을 추가하는 것이 어렵지 않도록 소프트웨어 구조를 적절히 설계해야 합니다.
- 명시된 요구사항 외의 게임 규칙 등은 1차에서 제시된 요구사항과 어긋나지 않도록 적절히 설계합니다.
- e.g.) 아이템 표시 + 색맹 모드, 난이도 설정 저장, 게임 실행 환경 등.
- 팀별로 설계/구현한 아이템 등에 대해서는 직접 요구사항이나 명세서 - 아이템 형태, 기능 - 등을 정리해야 합니다.
- 이 내용은 평가시 평가하는 팀이 잘 구현되었나 확인할 수 있도록 제공해야 합니다.

# 프로젝트 관리

- 프로젝트의 진척 상황 및 할 일(Task) 배정 내역 등을 Trello 등의 도구에서 관리하여야 함.
  - 각 할 일 및 배정 내역을 토대로 팀원의 생산성이 적절히 확인될 수 있도록 함.
- 프로젝트에서 구현되는 코드 및 필요한 자료(resource)는 Git과 GitHub의 저장소를 이용하여 버전 관리가 이루어져야 함.
  - Branch 관리 및 Pull Request 처리 정책 등.
- 프로젝트 진행시 발생하는 이슈(Issue)에 대해서는 GitHub에 이슈로 등록하고 그 해결 내역을 추적가능하도록 해야함.

# 주의 사항

- 이 문서는 대략적인 요구사항을 설명하고 있습니다.
  - 추가적, 세부적 요구사항에 대해서는 팀별로 주어진 요구사항을 분석하여 결정해야 하며,
  - 모호한 부분이나 팀 내에서만 결정해서 문제가 될 것 같은 부분은 e-class에 문의하여 정해야 합니다.
- 각 슬라이드의 요구사항은 기능적/비기능적 요구사항이 섞여서 제시될 수 있습니다.
- 명시적으로 작성되어 있지 않으나, 주어진 요구사항을 충족시키기 위해 암시적으로 요구되는 사항 또한 평가 시 확인할 수 있습니다.
- 강의시간에 배우는/배우게 될 내용들을 잘 활용하여 프로젝트에 적용하시기 바랍니다.