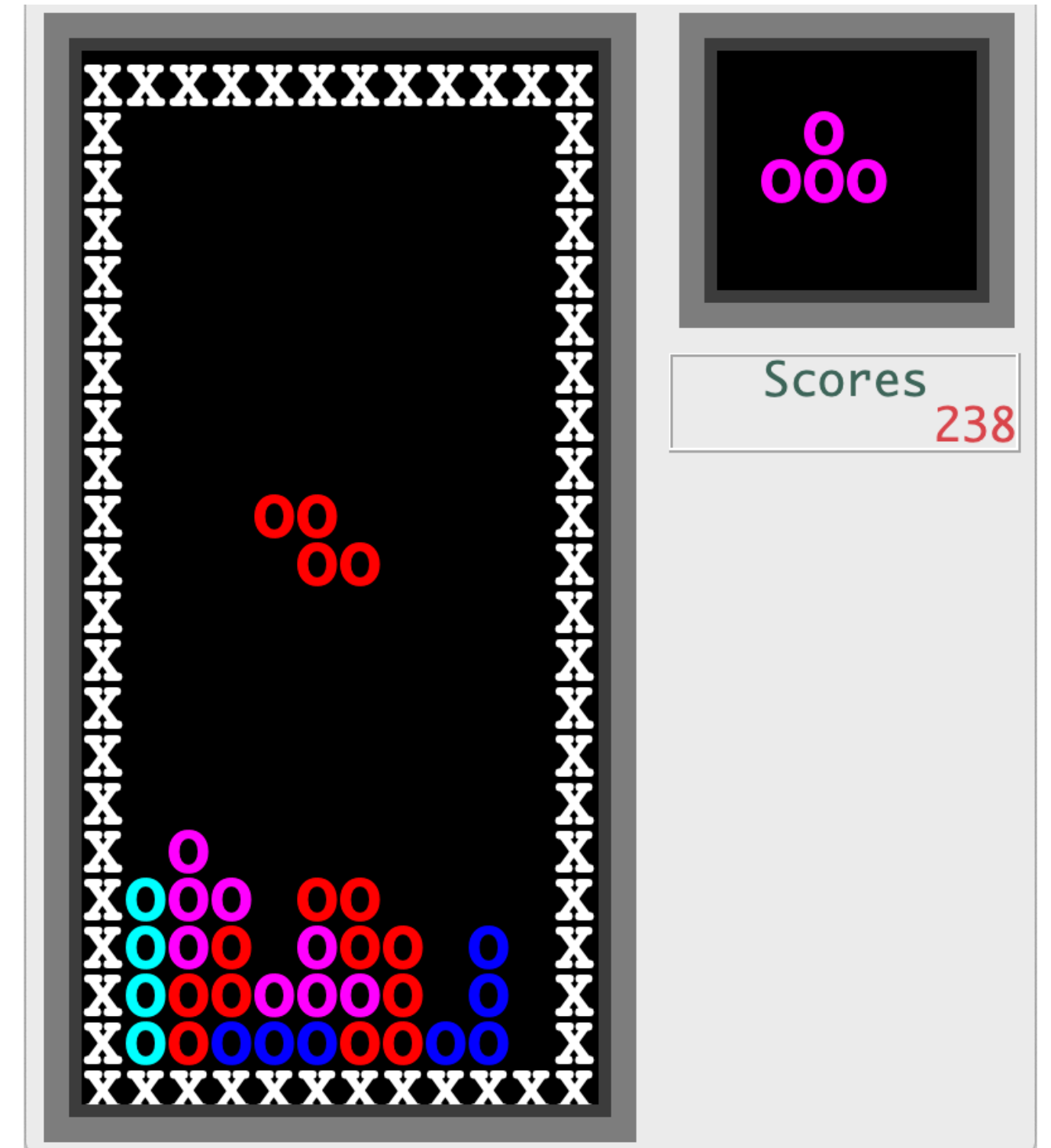


Requirements 1

Software Engineering Team Project
Spring 2022, SeoulTech

텍스트 기반 테트리스 게임 개발

- 팀프로젝트의 기본적인 목표는 텍스트 기반의 테트리스 게임을 개발하는 것입니다.
- 요구사항은 총 3차에 걸쳐 제시될 것이고, 각 차에 제시되는 요구사항으로 게임을 확장해 나가게 될 것입니다.
 - 따라서 확장될 것을 염두에 두고 프로젝트를 진행해야 합니다.
- 1차 요구사항은 기본적인 테트리스 게임을 구현하는 것을 목표로 합니다.
 - 오른쪽은 단순한 테트리스 게임의 예시입니다.
- 주어진 요구사항이 충족되는 경우 화면 구성 및 기타 세부사항은 각 팀별로 자유롭게 논의하여 정해도 됩니다.



Major Functional Requirements

- 1차로 개발할 테트리스 게임의 주요 기능적 요구사항(Functional Requirement)은 다음과 같음.
 - 시작 메뉴
 - 기본 테트리스 게임
 - 종료 화면
- 게임 실행 후 종료까지 거치는 기본적인 기능들이며, 자세한 요구사항은 이어지는 슬라이드에서 설명됨.

Major Non-functional Requirements

- 주요 비기능적 요구사항(Non-Functional Requirement)은 다음과 같음.
- 작성된 게임은 제시된 환경에서 적절한 성능으로 실행가능해야 함.
자바 1.8 이상 환경 , 메모리나 이런건 일단 개발하고 생각
OS 생각 좀 해야될듯
- 프로젝트 진행에 따라 발생하는 산출물 - 코드, ^{노션}관련 문서, 논의사항 등 - 을 추적 관리하고 그 내역을 제시할 수 있어야 함.
- 작성한 코드는 기능 및 성능을 충족하는 지 확인하기 위해 충분히 테스트되어야 함.
- 게임의 원활한 배포 및 실행을 위해 적절한 방법 및 도구가 사용되어야 함.
자바 8 , 인텔리제이

시작 메뉴

- 게임을 실행하면 처음으로 나오는 메뉴를 구현함.
- 메뉴 화면의 상단에는 게임의 이름이 표시되어야 함.
- 그 아래로 게임을 시작할 수 있는 메뉴와 설정 메뉴, 스코어보드, 게임 종료 메뉴(프로그램 종료)를 표시.
 - 각 메뉴를 선택하면 해당하는 기능을 수행.
- 키보드를 이용하여 메뉴 사이를 이동할 수 있어야 함.
 - 엔터(리턴) 키를 눌러 메뉴를 선택.
 - 다른 키를 사용하는 경우는 화면에 사용가능한 키를 표시.
- 메뉴가 차후 추가될 수 있도록 설계되어야 함.

테트리스 게임

- 시작 메뉴에서 게임 시작을 선택하였을 경우 테트리스 게임을 시작.
- 기본 테트리스 게임은 다음을 반복하는 식으로 진행됨.
 - 20줄, 10칸의 보드(board)가 존재.
 - 총 7가지의 테트로미노 - 이후는 블럭이라고 지칭 - 가 무작위로 등장.
 - 블럭을 쌓아 각 행을 채우면 해당 행이 삭제됨.
- 보다 자세한 게임 방식은 실제 테트리스 게임을 참고하여 이해할 것.
 - <https://ko.wikipedia.org/wiki/테트리스>

화면 구성

- 기본적인 화면 구성은 다음을 포함.
 - 블록이 쌓이는 보드(board), 다음 블록을 확인할 수 있는 부분, 점수를 확인할 수 있는 부분.
- 보드는 블록의 한 칸을 기준으로 하였을 때 20개의 줄과 10개의 칸을 가져야 함.
- 다음 블록 부분은 각 블록이 보드에 생성될 때마다 다음에 생성될 블록들을 표시.
- 점수 부분은 실시간으로 바뀌는 점수를 표시.

게임 조작

- 보드에 현재 블록이 생성되면, 이 블록은 아무 조작이 없어도 1초에 1칸씩 밑으로 떨어져야 함.
 - 일정 수 이상의 블록이 생성되거나, 줄이 삭제되면 떨어지는 속도가 증가해야 함.
- 현재 블록을 키보드 입력에 따라 다음과 같이 조작 가능해야 함.
 - 좌, 우, 아래로 한 칸씩 이동시킬 수 있어야 함.
 - 시계방향으로 90도씩 회전시킬 수 있어야 함.
 - 한 번에 끝까지 밑으로 떨어뜨릴 수 있어야 함.
- 반복된 키 입력이 무시되지 않고 블록이 즉시, 정확하게 반응해야 함.
- 게임 도중 게임을 잠시 중단/재개할 수 있는 키가 존재해야 함.
- 게임 진행/중지 상태에서 게임을 종료할 수 있는 방법이 존재해야 함 (alt-f4 등 OS를 이용한 방법 제외).

블럭 종류

- 참조 코드에 제시된 7가지 모양의 블럭을 구현.
 - <https://github.com/Jindae/SeoulTech-SE-Tetris-Ref>
- 다음에 나타날 블럭은 동일한 확률로 무작위로 결정됨.
- 각각의 블럭은 모두 다른 색상을 가져야 함.
 - 유사한 모양이나 좌우가 바뀐 블럭들은 구분이 용이하도록 대비되는 색상이나 무늬를 지녀야 함.

색맹 모드

- 전 세계 남성의 8%, 여성의 0.5%가 색각이상(color blind)에 해당.
 - <https://www.colourblindawareness.org/colour-blindness/>
- 테트리스 게임은 각 블럭이 모양으로 구분가능하나, 꺾인 방향 등이 색으로 표현되는 것을 인지하면 더 유리하게 되어있음.
- 따라서 색각이상자가 불리함을 겪지 않도록 색약 모드를 추가할 필요가 있음.
- 설정에서 색맹모드를 선택하면 적,녹생맥 및 청황색맹 또한 블럭 구분이 쉽도록 바꿔주는 기능을 구현.
- 색각이상과 관련한 디자인에 대한 정보는 다음의 링크들을 참조.
 - 색각이상자를 위한 접근성 가이드: <https://nuli.navercorp.com/community/article/1132656>
 - 색각이상자를 위한 유니버설디자인 연구: <http://aodr.org/xml/00201/00201.pdf>
 - 적록 색각 이상자를 위한 색변환 기법: <https://www.koreascience.or.kr/article/JAKO201310837317613.pdf>

점수 계산

- 기본적으로 블럭이 1칸 떨어질 때마다 한 단위의 점수를 획득.
 - 자동/수동 조작에 무관하게 획득함.
- 바뀐 점수는 실시간으로 게임 화면에 표시되어야 함.
- 블럭이 떨어지는 속도가 초기 속도보다 빨라진 경우 추가적인 점수를 획득.
- 그 외 다양한 방식의 점수를 가산/감산하는 방식을 추가할 수 있음.
 - 적어도 1개 이상의 직접 제시되지 않은 방식을 구현할 것.

줄 여러개 지우면 추가점수, 하드드롭일때 남은 칸 만큼 단위 점수 곱

설정 화면

- 설정 화면에서는 다음과 같은 기능을 제공해야 함.
 - 테트리스 게임 화면의 크기를 조절 - 최소 3가지 이상 미리 정의된 크기.
 - 게임 조작을 위해 사용될 키를 설정.
 - 스코어 보드의 기록을 초기화.
 - 색맹 모드 켜고 끄기.
 - 모든 설정으로 기본 설정으로 되돌리기.
- 바뀐 설정은 저장되었다가 다음 게임 실행시 동일한 설정을 불러와 사용할 수 있어야 함.
- 그 밖의 필요한 설정을 추가하는 것도 가능.

게임 종료

- 더 이상 블럭을 쌓을 수 없게 되면 게임이 종료되어야 함.
- 게임 종료시 스코어 보드를 표시함.
 - 현재 게임 점수가 스코어 보드에 기록되어야 하는 경우 이름을 입력 받는 화면을 표시.
 - 이름을 입력하고 나면 스코어 보드를 업데이트 하여 방금 입력한 이름과 점수를 강조하여 표시.
- 스코어 보드 처리가 끝나면 현재 게임을 종료하고 시작 메뉴로 돌아가거나 프로그램을 종료.

스코어 보드

- 게임의 역대 점수 순위를 보여주는 스코어 보드를 구현.
- 각 순위별로 이름 및 점수 등의 정보를 점수가 높은 순으로 정렬하여 보여줌.
- 전체 기록된 순위는 적어도 상위 10개 이상을 포함.
- 설정에서 초기화하기 전에는 프로그램을 종료하더라도 스코어 보드 기록이 유지되어야 함.
- 그 외의 사항은 자유롭게 설계 및 구현.

게임 실행 환경

- 게임은 Java 1.8버전 이상이 설치된 다음의 사양을 만족하는 PC에서 오류없이 실행 가능해야 함. == 자바 1.8 이상의 모든 버전에서 돼야되다는거? >> 도커라도 쓰자
- Windows 10 또는 macOS 10.14이상 - 하나만 대상으로 해도 무방.
- 1.2GHz이상 프로세서와 1GB RAM.
- 500MB이상의 사용공간.
- 위 사양을 만족하는 PC에서 게임의 플레이에 문제가 없을 정도의 성능이 제공되어야 함.

프로젝트 관리

- 프로젝트 진행시 발생하는 산출물은 적절하게 관리되어야 함.
 - 코드는 Version Control System (VCS)를 이용해 관리.
 - 프로젝트 진행시 발생한 이슈와 그 해결상황을 적절히 기록하고 관리.
 - 팀원별 업무 분담을 적절히 기록하고 관리.
- 중간 및 최종 결과물로 위의 사항을 모두 제시할 수 있어야 함.

작성된 코드의 검증

- 작성된 코드를 검증할 수 있는 단위 테스트(unit test)를 충분히 작성해야 함.
 - Line Coverage 기준 직접 작성한 코드의 60%이상이 테스트되어야 함.
- 비기능적 요구사항의 충족여부를 검증할 수 있는 테스트가 하나 이상 존재해야 함.
- 테스트들은 코드와 마찬가지로 적절히 관리되어야 함.
- 프로젝트 진행시 작성된 테스트를 손쉽게 실행하여 변경사항 발생시마다 검증을 수행할 수 있는 체계를 갖추어야 함.
- Unit test framework 사용 및 code coverage의 지속적인 관리 등.

게임의 실행 및 배포

- 작성된 코드 및 필요한 정보를 이용하여 손쉽게 게임의 배포용 패키지를 작성할 수 있는 체계를 갖춰야 함.
- 배포용 패키지는 **아이콘이 포함된 실행파일**을 가지고 있어야 하며, 이 실행파일을 더블클릭하는 등의 운영체제에서 지원하는 동작으로 게임을 실행할 수 있어야 함.
 - 추가적인 명령어 입력 및 설정 등이 없이 실행 가능해야 함.
 - e.g.) `java -cp xxx Tetris` 등의 명령을 명령 프롬프트에 입력하게 하는 방식은 불가.
- 대상이 되는 운영체제는 Windows나 macOS 중 하나로 정해도 무방함.
 - 각 팀이 보유한 기기를 이용할 것.
- Java 1.8버전 이상은 이미 설치되어 있고, 그 경로 또한 설정되어있다고 가정해도 무방함.
 - 실행 파일에서 `JAVA_HOME` 환경변수 사용 후 `JAVA_HOME`이 설정되지 않은 PC에서는 실행되지 않아도 됨.
 - 또는 JRE 자체를 패키지에 포함시키는 방법을 사용하는 것도 가능함.

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.