

프로젝트 최종보고서

과제 수행원 현황						
수행 학기	<input type="checkbox"/> 2019년 1월~6월 <input checked="" type="checkbox"/> 2019년 3월~6월 <input type="checkbox"/> 2018년 9월~2019년 6월					
프로젝트명	오픈소스를 활용한 테트리스 게임 개발					
팀명	Millions					
	학과	학번	성명	성별	연락처	E-mail
팀장	경제학과	2015110871	김주영	남	01050291192	rlawndud1104@naver.com
팀원	화학과	2016110453	길효진	여	01056544089	gywls1345@naver.com
	수학과	2015110433	배정수	남	01036513639	bjs001123@naver.com
지도교수	교과목명	오픈소스소프트웨어프로젝트				
	소속	<input type="checkbox"/> 컴퓨터공학전공 <input type="checkbox"/> 정보통신공학전공 <input type="checkbox"/> 멀티미디어공학전공 <input checked="" type="checkbox"/> 융합소프트웨어연계전공				
	성명	김동호				

과제 일반 현황					
작품(과제)명					
특허.실용신안					
포상여부	상격	기관	행사명	수상일시	부상내역

	※ 포상실적은 해당사항이 있을시 필히 기재 요망. ※ 포상실적을 허위로 기재시 신청인은 포상대상에서 제외됨 ※ 타기관에서 이미 수혜받은 정부포상 과제는 포상대상에서 제외됨				
<div style="text-align: center;">보고서</div>					
작품명 (프로젝트명)	Tetris Of the player, By the player, For the player				
# Key Words	#키설정	#음악	#소켓통신	#오픈소스	#Player 중심
1.개발동기/목적/필요성및 개발 목표	<p>* 개발동기 및 목적: 오픈소스를 활용하여 개발하는 과제에서 팀원들이 모두 사용 가능한 Java 언어를 선택했습니다. 친숙하고 대중적인 테트리스 게임 소스를 사용해서 개발의 접근성을 높이고자 했습니다.</p> <p>* 개발목표: 플레이어를 중심으로 둔 게임 개발</p> <ol style="list-style-type: none"> 1. 배경음악, 효과음 삽입 2. 사용자의 키 설정 기능 추가 3. 게임 종료시 이벤트 추가 4. 상대 플레이어의 화면을 받아와서 출력하여 볼 수 있도록 변경 5. Github 활용 능력 향상 6. 리눅스 사용 경험 향상 				
2.최종 결과물 소개	<ol style="list-style-type: none"> 1. 배경음악, 효과음 삽입 <ul style="list-style-type: none"> - 배경음악과 효과음(Left,Right,Up,Down,Shift,Space,LineClear, MousePressed, MouseEntered, GameOver)을 추가하였습니다. - 효과음은 팀원들이 직접 녹음하였습니다. - 영상 첨부합니다. 2. 사용자 키 설정 기능 <ul style="list-style-type: none"> - 윈도우 메뉴에서 키 설정을 할 수 있습니다. - 이전에 입력한 키와 중복된 키를 입력되는 상황을 방지하는 기능을 추가하였습니다. 3.게임 종료 시 “Game Over” 팝업 이미지와 효과음 추가 본래의 코드에서는 게임이 종료되었다는 느낌을 받을 수 없었습니다. 게임이 끝나면 게임 종료 팝업 이미지와 효과음이 추가되어 극적인 반전을 주었습니다. 4. 상대방 플레이 화면 받아오기 기존의 멀티플레이 모드에서는 본인의 게임 화면만 보여 지기 때문에 대결의 긴장감을 				

	<p>느낄 수가 없었습니다. 상대방의 화면을 받아와서 경쟁의식을 자극하도록 하였고 한 쪽이 게임오버가 되면 양 쪽에 게임오버 창이 띄워지며 게임이 함께 종료될 수 있도록 하였습니다.</p> <p>5. 시작 화면 추가 프로그램을 실행시키면 바로 메인 게임으로 진입하지 않고 시작 화면이 보여집니다. START와 QUIT버튼이 있어서 START버튼을 누르면 메인 화면으로 진입하여 게임을 할 수 있고, QUIT 버튼을 누르면 프로그램이 종료됩니다.</p> <p>6. 메인 게임의 이미지 변경 메인 게임 화면의 바탕 그림을 변경하였습니다. 기존의 밋밋하고 딱딱했던 화면과 비교하여 밝은 화면으로 표현하였습니다.</p> <p>7. 효과음 / 배경음악을 끄고 켤 수 있는 체크박스 추가 기존의 체크박스(격자표시, 고스트블록 표시) 옆에 효과음과 배경음악을 끄고 켤 수 있는 체크박스를 추가하였습니다. 현란한 효과음과 배경음악으로 귀가 피곤해진 사용자에게 편안한 게임 환경을 제공해줍니다.</p> <p>8. 내 점수 표시 기존에 본인의 점수를 확인 할 수 없었던 진행 방식이 게임을 하는 사용자의 흥미를 떨어트릴 수 있다고 판단하여 확장시킨 공간에 점수 표시 기능을 추가했습니다. 점수는 한 줄에 100점씩 추가되도록 했습니다. 나아가, 한 번에 여러 줄을 제거할 시 추가 점수가 부여됩니다.</p> <p>9. 게임 배경음악 선택 옵션 게임 배경음악을 3개로 추가하여 사용자의 선택 폭을 넓혔습니다. 자신이 듣고 싶은 BGM의 번호를 누르면 해당 배경음악으로 변경할 수 있습니다.</p> <p>10. 블록의 높이에 따른 배경음악 변경 블록이 쌓일수록 배경음악이 변경됩니다. 8째 줄에 쌓였을 때, 15줄에 쌓였을 때 분위기를 전환하기 위해 자동으로 음악이 바뀝니다.</p>
<p>3.프로젝트 추진 내용</p>	<p>프로젝트 진행과정</p> <ul style="list-style-type: none"> - 1주차(4월15일 ~ 21일): 효과음, 배경음악 체크박스, 윈도우 상단에 키 조작 메뉴를 새로 만들었습니다. 설정하기 버튼을 추가했지만 기능은 없습니다. 상대방의 게임 화면을 받아올 공간을 늘렸습니다. 게임이 종료될 때 “Gave Over” 화면이 뜨도록 만들었습니다. - 2주차(4월22일 ~ 28일) : 2주차는 중간고사 기간이라 쉬었습니다.

- 3주차(4월29일 ~ 5월6일) : Music 이라는 기능을 만들어서 노래가 재생되도록 했습니다. 하지만 음악이 겹치는 문제가 발생하였습니다. “Gave Over”화면의 이미지를 다른 것으로 변경했습니다.
- 4주차(5월6일 ~ 12일) : Left, Right, Up, Down, Shift(블록저장), Space(블록 드랍) 버튼 입력, 그리고 한 줄이 클리어 되었을 때 효과음이 발생되도록 했습니다. 3주차에 음악이 겹치는 상황을 보완하여 하나만 재생되도록 하였고, 나가기 버튼을 누르면 음악이 꺼지도록 했습니다. “Gave Over”창과 함께 효과음이 나도록 구현했습니다. ‘시작하기’, ‘나가기’ 버튼에 공간에 마우스 커서가 들어갈 때와 버튼을 누르면 효과음이 발생하도록 했습니다. 키 설정 버튼을 누르면 설정창이 뜨도록 개발했습니다. “Button” 기능을 추가하여 키 설정을 변경했습니다. 그러나 이 기능이 원래 버튼 동작 과정에 적용이 잘 안되어서 어려움을 겪고 있습니다.
- 5주차(5월13일 ~ 19일) : 효과음에 들어갈 목소리 녹음을 마치고 게임에 적용하였습니다. 4주차에 문제가 있었던 키 설정 기능을 보완하여 완전하게 구현 가능하도록 개발했습니다. 시작 화면을 추가하여 ‘시작하기’와 ‘종료하기’를 선택할 수 있도록 하였습니다. 게임 색상을 변경했습니다.
- 6주차(5월20일 ~ 26일) : 키 설정 기능에서 동일한 버튼으로 두 가지(예를 들면, A라는 키보드로 Left와 Right 키 둘 다 설정하는 상황)이상의 중복 설정을 방지하도록 보완하였습니다. 중복된 키를 입력하면 “000와 중복된 키입니다.”라는 경고창이 뜨도록 했습니다. 효과음과 배경음악 체크박스에 실질적인 기능을 추가하여서 체크박스를 누르면 음악을 끄고 켤 수 있도록 개발했습니다. 그러나 배경음악에서는 체크박스에 마우스 커서를 가져다 올려놓기만 해도 재시작 하는 문제가 발생하여 보완이 필요합니다. 녹음한 mp3 파일들의 음량 크기가 다 달랐는데 mp3gain 이라는 응용프로그램을 이용해서 음량크기를 일정하게 맞추었습니다.
- 7주차(5월27일 ~ 6월2일) : 상대 플레이어 게임이 종료되었을 때 “Gave Over” 팝업, 음악 종료, 게임 종료가 안되었었는데 이를 해결하였습니다. 상대 플레이어의 체크박스가 기능을 하지 못했었는데 해결하였습니다. 상대방의 블록을 받아와서 내 화면에 그리도록 개발하였습니다.
- 8주차(6월3일 ~ 6월9일) : 게임 종료 이미지를 제작하여 변경했습니다. 나의 점수를 표시하고 라인을 클리어할 때마다 100점이 추가되도록 했습니다. 게임의 배경 화면을 변경하였습니다. 배경음악을 변경, 추가하고 사용자가 취사선택할 수 있도록 기능을 추가했습니다.
- 9주차(6월 10일 ~ 6월 12일) : 콤보 점수를 추가했습니다. 한 번에 여러 줄을 제거하면 기본 점수 + 추가 점수를 받게 됩니다. 블록 높이에 따라 음악이 변환됩니다. 블록이 8째 줄까지 쌓인 상황, 15째 줄까지 쌓인 상황이면 음악이 자동으로 전환됩니다. 단 사용자가 BGM 버튼을 눌렀을 경우는 해당 기능이

작동하지 않습니다.

2. 프로젝트 구현과정

① 배경음악, 효과음 삽입

1) 개념 설계

음악 기능을 추가하기 위해서 Thread 를 이용하기로 했습니다. 코드의 효율성을 위해 'Music' 클래스를 만들어서 다른 클래스의 어느 위치에서도 쉽게 객체로서 뮤직 메소드를 사용할 수 있도록 했습니다.

2) 상세 설계

(1) Music 클래스 생성하여 객체로서 어디서든 편하게 사용할 수 있도록 합니다.

(2) Thread를 사용하여 게임과 별도로 음악이 끄고 켜질 수 있도록 합니다.

(3) 간접경로의 음악 mp3 파일을 불러오도록 합니다.

3) 구현이론

(1) Music' 클래스를 새로 만들고, JLayer 라이브러리를 참조하였습니다.

(2) Public Music(String name, Boolean isloop) 메소드를 통해 name의 이름을 가진 음악 파일을 불러오고 isloop가 true라면 무한이 스레드가 돌도록 했습니다.

(3) Public void close() 메소드를 만들었고, 이 메소드를 호출하면 isloop가 false가 되고 thread를 interrup하여 스레드 동작이 그만두도록 했습니다.

(4) 다른 클래스에서 Music GameMusic = new Music("name", true); 과 같이 새로운 객체를 생성하고, GameMusic.start()를 명령어를 통해 해당 음악 스레드가 동작하도록 했습니다.

(5) 게임의 메인 thread 가 시작되는 메소드에 삽입하여 게임이 시작될 때마다 새로 시작하도록 구현했습니다.

② 사용자 키 설정기능

1) 개념설계

기존 작품에는 블록을 움직이기 위해 가상 키보드(VK)를 불러와 고정된 키로 처리를 했지만 사용자가 원하는 키로 변환하기 위해선 고정된 키(가상 키보드)가 아닌 변경 가능한 변수로 설정해야 한다고 생각하였습니다. 또한 원하는 키를 입력할 창이 필요하다 생각하여 window package에 button class를 만들고 JFrame을 이용하여 창을 구현하였습니다.

2) 상세설계

(1) 기존 테트리스 상단 bar에 키 설정 가능한 menu를 삽입합니다.

(본 기능은 마우스와 키보드를 입력받아 처리할 수 있게 구현.)

(2) 총 6개의 기능(left, right, rotation, down, drop, hold)을 변경할 수 있는 창이 필요합니다.

(3) 사용자가 입력한 키를 원하는 기능에 넣을 수 있는 입력 창이 필요합니다.

(4) 입력한 키에 해당하는 코드값을 변수에 저장하고 get함수를 이용하여 해당 기능을 원하는 키에 맞춰서 실현합니다.

3) 구현 이론

(1) window package에 button class를 만들고 JFrame을 이용하여 창을 구현합니다. 또한 마우스와 키보드를 입력받고 처리할 수 있게 MouseListener와 KeyListener를 상속받습니다.

(2) 원하는 기능의 키를 변경할 수 있도록 각 기능에 해당하는 버튼을 JButton을 이용하여 생성하빈다.. 이때 각 버튼을 눌렀을 때 각기 다른 Text와 기능이 실행되어야하므로 각 버튼에 따른 pk값을 지정합니다.

(3) 버튼을 눌러 입력창이 팝업 되면 사용자가 하나의 키를 눌렀을 때 창이 종료되면서 입력 코드값을 변수에 저장합니다. TetrisBoard class에 그 기능에 맞는 부분을 get함수로 코드값을 불러옵니다.

4) 기존 작품과의 차이점

기존 작품에는 키를 설정할 수 있는 기능이 없었고 어떤 키로 플레이를 하는지 직접 눌러봐야 알 수 있었습니다. 이에 불편함을 느껴 키를 설정할 수 있는 기능을 추가하였고 이를 통하여 어떤 기능이 있는지 알려주는 환경을 만들었습니다.

③ 게임 종료 시 팝업 이미지와 효과음 추가

1) 개념 설계

게임 종료 시 반전 효과를 주기 위해 화면 전환과 효과음이 필요하다고 생각했습니다. 화면 전환은 팝업창이 뜨도록 생각했고, 효과음은 Music 클래스를 이용하여 구현하기로 했습니다. 팝업 이미지 파일과 와 음악 mp3 파일은 소스 폴더에 저장하고 간접 경로로 불러올 수 있도록 했습니다.

2) 상세 설계

(1) “Game Over” 팝업창과 효과음 코드를 작성합니다.

(2) 게임이 종료되는 조건문을 찾아서 팝업 메소드와 효과음 메소드를 삽입합니다.

(3) 게임이 종료되면 게임을 진행중일 때의 음악 thread는 종료되고 새로운 게임 종료 thread 가 동작해야합니다.

3) 구현이론

(1) Javax.swing.ImageIcon 과 javax.swing.JOptionPane 패키지를 import하여 활용합니다.

(2) Popupicon 이라는 객체를 생성하여 간접 경로(자바 파일 내)의 png파일을

호출하여 객체에 저장합니다.

(3) ShowMessageDialog메소드를 활용하여 이미지가 저장된 popupicon을 호출하여 팝업에 이미지가 나오도록 구현합니다. 만약 GameMusic(게임 실행 중의 음악Thread)가 살아있다면 해당 thread를 종료합니다.

(4) GamEndSound라는 새로운 Music 객체를 생성하여 동작합니다

(5) 게임이 종료되는 조건문을 통해 Public void gameEndCallBack(){ 메소드가 실행됩니다. 이 메소드 안에 위의 코드들을 삽입하여 구현합니다.

④ 상대방의 게임 화면 받아와서 그리기

1) 개념 설계

게임의 화면 넓이를 넓혀서 나의 화면과 똑같이 격자표시와 게임 보드를 그립니다. 현재 소켓통신을 이용해서 Message를 주고 받고 있습니다. 이 개념을 이용하여 상대방의 블록 데이터를 받아와서 화면에 그립니다.

2) 상세 설계

(1) 상대방의 보드와 블록을 그릴 공간을 확보합니다.

(2) 상대방의 블록이 표현될 보드와 격자를 그립니다.

(3) Network 패키지의 GameServer와 GameClient 클래스에서 메시지를 주고받는 메소드를 보고, 블록을 주고 받는 메소드를 구현합니다.

(4) 전송된 블록 데이터를 내 게임 화면에 그립니다..

3) 구현이론

(1) 패널(게임 전체화면) 값에 *2를 하여 너비를 늘립니다.

(2) 나의 보드(테트리스 블록이 표현되는 화면)와 격자를 그리는 코드에서 x좌표를 수정하여 상대방의 보드를 그립니다.

(3) GameClient 클래스

DataShip 클래스에 DRAW_BLOCK_SHAP과 DRAW_BLOCK_DEPOSIT 변수를 선언합니다. Shap 이라는 새로운 TetrisBlock 객체를 생성합니다. data라는 DataShip 객체 생성하여 DRAW_BLOCK_SHAP을 객체에 저장하고, 해당 데이터를 서버에 요청합니다.

(4) GameSever 클래스

클라이언트로부터 받은 블록과 요청한 인덱스 데이터를 player의 변수로 가져와서 그대로 broadcast하여 화면에 그립니다.

⑤ 시작화면 추가

1) 개념 설계

기존 작품에는 시작 화면 없이 바로 게임 화면으로 시작하여 화면의 다양성과 게임에 있어서 필수적이라고 생각하는 시작 화면을 구현하는 것을 구상했습니다. 시작 화면에는 상단바가 있어 상단바를 잡고 움직일 수 있어야 하고 상단바 우측에는 창을 닫을 수 있는 버튼을 추가합니다. 또한 '시작'과 '나가기'의 두개의 옵션을 주어 사용자가 게임에 응할 것인지 게임을 종료할 것인지 선택할 수 있습니다. 배경화면과 각 버튼들은 디자인하여 파일을 간접 경로로 불러서 이용합니다.

2) 상세 설계

(1) 새로운 창을 만들 새로운 클래스 생성합니다.

(2) 상단바, 상단바의 나가기버튼, 시작하기 버튼, 나가기 버튼 디자인 및 각각에 맞는 기능 부여합니다.

(3) 배경이미지를 간접 경로로 불러와 paint하도록 설정합니다.

(4) '시작하기' 버튼을 누르면 게임화면으로 넘어가도록 설정합니다.

3) 구현 이론

(1) 'TetrisOpening' 클래스 생성합니다.

(2) 사용할 아이콘과 JButton을 이용하여 버튼 정의합니다. 각 버튼마다 setBounds 함수를 통해 x좌표, y좌표, 너비, 높이를 지정합니다. 각 버튼의 기능에 맞게 mousePressed를 받아 구현합니다.

(3) 상단바는 mouseDragged가 되었을 때 x좌표를 마우스의 x좌표, y좌표를 마우스의 y좌표로 받아와 사용자가 원하는 대로 움직일 수 있도록 구현합니다.

(4) Paint 함수를 이용해 이미지를 그려주고 setVisible을 이용하여 보이기 설정을 맞춥니다.

(5) TetrisMain에서 시작하면 TetrisOpening을 불러오도록 하고 '시작하기' 버튼을 눌렀을 때 기존의 게임 메인화면으로 넘어가도록 구현합니다.

⑥ 메인게임의 배경 화면 변경

1) 개념 설계

배경으로 사용할 이미지를 준비합니다. 현재 화면을 채우고 있는 코드를 지우고, 이미지 파일을 배경화면으로 사용합니다.

2) 상세 설계

(1) 현재 배경화면을 채우고 있는 코드를 지웁니다.

(2) 게임 해상도(960 x 600)에 맞는 이미지를 준비합니다.

(3) Java.awt.Graphics 패키지를 이용하여 이미지를 삽입합니다.

3) 구현이론

- (1) 현재 java.awt.Graphics 패키지를 참조해서 protected void paintComponent(Graphics g) 메소드에서 게임 화면을 그리고 있습니다.
- (2) 배경 색을 채우는 코드를 찾아서 지웁니다.
- (3) javax.swing.ImageIcon 패키지를 참조하고, icon 1이라는 ImageIcon 객체를 생성하여 icon1에 배경 화면에 쓰일 이미지를 저장합니다.
- (4) 화면을 그려내는 g.drawImage(icon1.getImage(),0,0,null) 명령어를 이용하여 객체에 저장된 이미지를 화면에 그립니다.

⑦ 효과음과 배경음악 체크박스 추가

1) 개념 설계

본래 없던 배경음악과 효과음을 삽입함으로써 음악을 키고 끌 수 있는 설정이 필요합니다. 기존에 있던 그리드체크박스와 고스트 체크박스 우측에 배경음악과 효과음 체크박스를 생성하는 것을 구상했습니다. 마우스를 클릭함에 따라서 음악을 키고 끌 수 있도록 제어하는 코드를 구현합니다.

2) 상세 설계

- (1) 기존 테트리스 게임 화면 격자표시와 고스트모드 체크박스 옆에 배경음악/효과음 체크박스 생성합니다.
- (2) 체크박스를 클릭했을 시 체크가 된 상태라면 음악을 재생하고 체크가 안 되어진 상태라면 재생되었던 음악을 정지합니다.

3) 구현 이론

- (1) TetrisBoard 클래스에 새로운 JcheckBox를 usingEffect(효과음)/usingBGM(배경음) 으로 생성하고 초기 Boolean 값을 true로 설정합니다.
- (2) 체크박스의 크기, 색상, 글씨체를 설정해주고 setRequestFocusEnabled를 false로 주어 한번 체크박스를 클릭 했을 때 화면의 포커스가 체크박스로 마우스를 가져다 대기만 하거나 다른 버튼을 눌렀을 때 체크박스가 해제/선택 되지 않도록 해줍니다.
- (3) actionPerformed를 통해 효과음/배경음 체크박스가 isSelected == true 이면 usingEffect(효과음)/usingBGM(배경음)을 true로 동작하고, isSelected == false 이면 usingEffect(효과음)/usingBGM(배경음)가 false가 되도록 설정합니다.
- (4) 효과음과 배경음을 넣어준 메소드에 if 조건문을 삽입하여 usingEffect의 값에 따라 Music 객체의 상태를 조절할 수 있습니다.

⑧ 점수 표시 기능

1) 개념 설계

기존 프로그램에 점수를 추가하는 기능이 있었지만 이를 UI상에 표시를 하지 않아

이를 시각화하였습니다. 게임 화면에 My Score 글자를 표시하고 밑에 점수를 삽입하였습니다. 콤보 점수 기능을 추가하였습니다.

2) 상세 설계

- (1) 게임 화면에 My Score를 그립니다.
- (2) 초기 점수를 0으로 설정하고, My Score 밑에 그립니다.
- (3) 한 줄이 clear 될 때마다 100점이 추가 되도록 합니다.
- (4) 시작하기를 누르면 점수가 0으로 초기화됩니다.
- (5) 한 번에 2줄, 3줄, 4줄을 삭제하면 추가 점수가 부여됩니다.

3) 구현 이론

- (1) score 라는 int 타입의 변수값을 0으로 초기 설정합니다.
- (2) My score와 score 의 글씨모양과 색을 정의합니다.
- (3) java.awt.swing 의 Graphics 컴포넌트를 이용해서 My Score와 score를 화면에 그립니다.
- (4) 한 줄이 채워지면 사라지는 메소드 조건문 안에 score가 100점이 추가되는 코드를 넣습니다.
- (5) 블록을 놓을 때 깨진 줄(combo)을 check하면서 점수가 추가됩니다. 즉, 깨지는 줄만큼 pk를 지정해주면 한 블록을 놓을 때 pk에 따른 점수를 부여할 수 있게 됩니다.
- (5) 시작하기를 누르면 스코어가 0으로 초기화 됩니다.

⑨ 배경음악 추가 선택 가능

1) 개념 설계

3개의 음악을 소스 파일에 저장합니다. 게임 화면에 BGM을 선택할 수 있는 버튼을 추가하여 사용자가 입맛대로 선택할 수 있도록 합니다.

2) 상세 설계

- (1) 3개의 mp3파일을 소스파일에 저장합니다.
- (2) 버튼 3개를 추가합니다.
- (3) 버튼을 누르면 해당 BGM을 호출하여 배경음악으로 설정됩니다.
- (4) 게임의 스레드가 작동중, 배경음악 체크박스가 체크되어있을 때만 음악이 실행되도록 합니다.
- (5) 현재 어떤 음악이 재생되고 있는지 나타냅니다.

3) 구현 이론

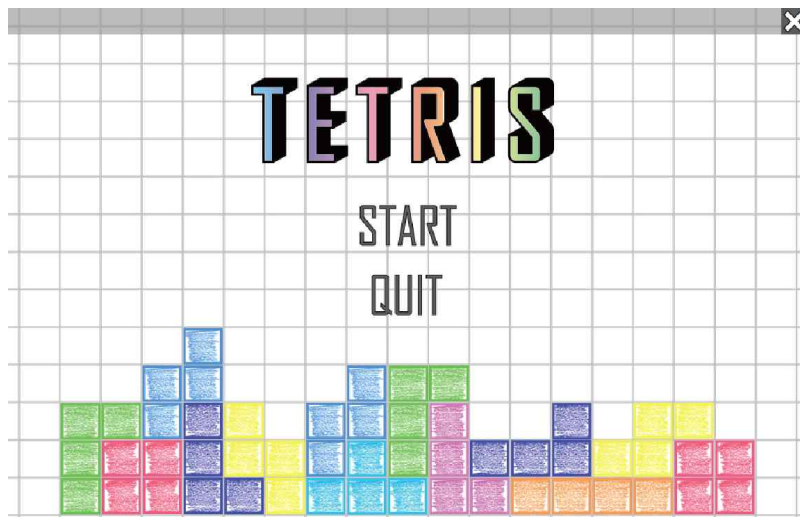
- (1) 3개의 mp3파일을 Music 패키지에 저장합니다.
- (2) JButton btnSound1, 2, 3을 추가하고 setbounds명령어로 절대위치를 조정하고, actionPerformed명령어로 실행 명령을 정의합니다.
- (3) SoundNumber라는 int 타입 변수를 만들어서 게임 시작시 초기값은 1로

	<p>설정합니다.</p> <p>ActionPerformed 메소드에서 사용자가 btnSound1을 선택하면 SoundNumber = 1, btnSound2를 선택하면 SoundNumber = 2, btnSound3을 선택하면 SoundNumber = 3이 되도록 합니다. 또한, 버튼을 누르면 해당 BGM이 실행되도록 합니다..</p> <p>(4) 게임을 실행시키는 메소드에서 위와 같이 SoundNumber의 값에 따라 각기 다른 BGM을 실행시키도록 합니다.</p> <p>(5) 게임의 스레드 (isPlay)가 true일 때만 실행되도록 합니다.</p> <p>(6) 버튼 밑에 현재 재생중인 음악을 나타내고 SoundNumber의 수(1,2,3)에 따라 글씨가 바뀝니다.</p> <p>⑩ 일정 높이 이상일 경우 BGM 전환</p> <p>1) 개념 설계</p> <p>일정 높이 이상으로 블록이 차오르면 음악이 변경되어 분위기의 전환을 일으킵니다.. 단, 사용자가 버튼을 클릭해서 음악을 설정해놓았으면 높이 이상이 되어도 음악이 변경되지 않습니다.</p> <p>2) 상세 설계</p> <p>(1) 블록의 높이를 판단하는 메소드를 찾습니다.</p> <p>(2) 블록의 높이와 체크박스 체크여부, BGM버튼 클릭 여부 등의 조건문을 만듭니다.</p> <p>(3) 조건이 성립되면 이 전의 음악은 종료되고 새로운 음악이 재생됩니다.</p> <p>3) 구현 이론</p> <p>(1) 맵을 계속해서 확인하는 메소드를 찾습니다.</p> <p>(2) 게임보드의 x좌표와 y 좌표값을 이용해서 줄의 좌표값을 찾습니다.</p> <p>(3) 게임 시작 시 FixedSound = 0 이라고 초기 설정값을 선언합니다.</p> <p>(4) Player가 BGM 버튼을 누르면 FixedSound = 1 이 되도록 합니다.</p> <p>(5) 8번째 줄에 블록이 쌓였을 경우 (블록의 높이, BGM 체크박스의 체크 유무, SoundNumber == 1 (Player가 BGM버튼을 누르지 않았다면 기본적으로 1이 재생되고 있기 때문입니다.) , FixedSound = 0)이라는 조건문을 작성하여 조건이 성립되면 BGM2가 재생됩니다.</p> <p>(6) 15번째 줄에 블록이 쌓였을 경우도 위와 동일한 조건문 형식으로 코드를 작성합니다. 단 블록의 높이, SoundNumber == 2 (BGM버튼을 누르지 않았다면 8째 줄에서 2번 음악이 재생되기 때문이다) 의 조건이 다릅니다.</p>
<p>4.기대효과</p>	<p>상업의 목적으로 제작된 것이 아니기 때문에 상용화되어지고 있는 게임들에 비해 경제적인 창출은 미약할 것으로 보여 지지만, Java를 이용하여 게임을 제작하는 프로그래머들에게 소스를 공개함으로써 도움을 줄 수 있을 것으로 예상됩니다. 또한 테트리스 게임을 갈망하는 이들에게 고전적인 게임 방식으로 추억을 부르게 하면서도 현대적인 디자인을 가미하여 테트리스를 처음 접하는 이들에게도 흥미와 재미를 유발할 수 있을 것으로 전망됩니다. 뿐만 아니라 기존의 코드에서 향상된 멀티플레이 모드를 통해 게임의 흥미도가 극대화 되어 남녀노소 즐길 수 있을 것으로 기대됩니다.</p>

	<p>Java로 개발한 테트리스 게임에서 여러 기능들을 추가하여 오픈함으로써, 차후에 다른 사람들이 공부하고 참고할 때 도움이 될 것이라고 생각합니다. 테트리스 게임에만 국한된 것이 아니라, 다른 게임이나 프로그램을 할 때에도 우리 팀이 구현해놓은 클래스와 로직을 보고 문제 해결의 씨앗이 되어 큰 나무가 될 것으로 예상됩니다.</p>
5.역할분담	<p><김주영> 매주 Progress Report 작성, 매주 해야 할 목표와 작업량 설정, 팀원들의 역할 분배, 상대방의 화면 받아오기 기능 구현, 팝업 이미지와 효과음 구현, 음악 기능 구현, BGM 전환 담당, Github 관리, 효과음 녹음, 발표,</p> <p><길효진> 게임 시작 화면 디자인 및 구현, 메인 게임화면 배경화면 제작 및 블록 색상 변경, 음악 메소드 구현, 효과음과 배경음악 체크박스 기능 구현, BGM 전환 담당, 효과음 녹음 및 본인 목소리 사용, 발표용 PPT제작, 발표, Github 관리,</p> <p><배정수> 키 설정 기능 구현, 제안서 ppt 작성, 제안서 및 보고서 최종 수정, 발표, BGM선택 기능 구현, 점수 표시 기능 구현, Github 관리, 효과음 녹음, 콤보 점수 담당</p>
6.참고문헌	<p>김동호 교수님 수업 자료</p>
7.첨부	<p>1. 최종 결과물의 메인 화면</p>



2. 시작 화면



3. 종료 화면

