



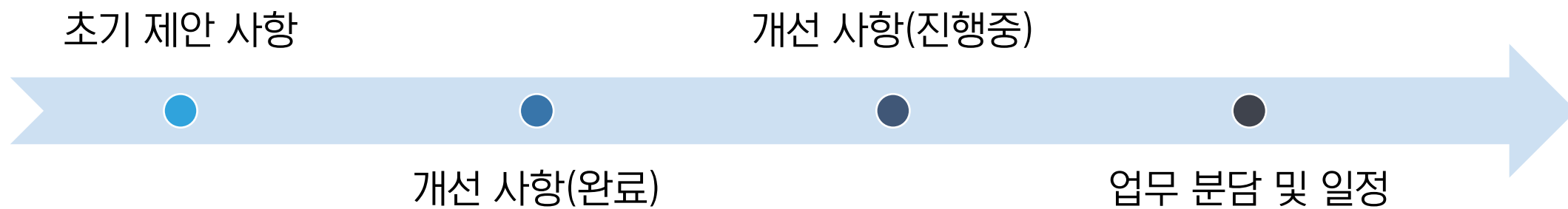
오픈소스 소프트웨어 프로젝트 중간보고서 발표

2020-2-OSSP-HotSource



2016112548 김정률
2016112587 김택원
2017112447 전영인

목차





초기제안사항

2020-2-OSSP-HotSource



초기 제안서 - 개선 사항



1. 게임 종료 후 오류 개선
2. 게임진행 오류 개선
3. 최고 점수 갱신
4. 시간 단축에 대한 추가 점수 부여
5. 게임 시작 전 준비 시간 부여
6. 쌓인 블록 구현
7. 음소거 기능



☐ 랭킹시스템 개선

- ☐ 로컬에서만 적용되는 랭킹시스템 X
- ☐ 어느 네트워크 환경에서도
동일한 랭킹 현황 갱신
⇒ DB 도입하여 해결

☐ 플레이 화면 여백 삭제

- ☐ 동시에 창크기에 따라 화면 Resizing
⇒ 현재 Resizing 중 오류 발생으로 수정 중

랭킹시스템 – DB연결

```

1 package com.ok.gamedb;
2
3 import java.sql.Connection;
4
5
6
7
8 public class MyDB {
9     static Connection conn = null;
10    static PreparedStatement pstmt = null;
11    static Statement stmt = null;
12    static ResultSet rs = null;
13
14    // RDS SETTING
15    static String url = "DB URL";
16    static String DB_ID = "ID";
17    static String DB_PW = "PASSWORD";
18    static String TABLE = "TABLE";
19 }

```

3. Ranking System with DataBase

ID : taekwontest, MY SCORE : 240
 1등! ID : yin, SCORE : 752
 2등! ID : jungreul, SCORE : 570
 3등! ID : taekwon2, SCORE : 260
 4등! ID : taekwon, SCORE : 240
 5등! ID : taekwontest, SCORE : 240

Result Grid		Filter Rows:	Edit:
id	score		
▶ jungreul	570		
taekwon	240		
taekwon2	260		
taekwontest	240		
yin	752		
• RULES	RULES		

Amazon Relational Database Service(RDS)

- Scoreboards that were only updated locally were synchronized anywhere by using the DataBase.

AWS RDS 인스턴스 생성

외부에서도 접근할 수 있는 DB 구축

기 존 : Text 파일 사용하여 랭킹시스템 구현 ⇒ 각자 랭킹 현황이 다름

개선 후 : AWS DB 사용하여 랭킹시스템 구현 ⇒ 동일한 랭킹 현황 갱신

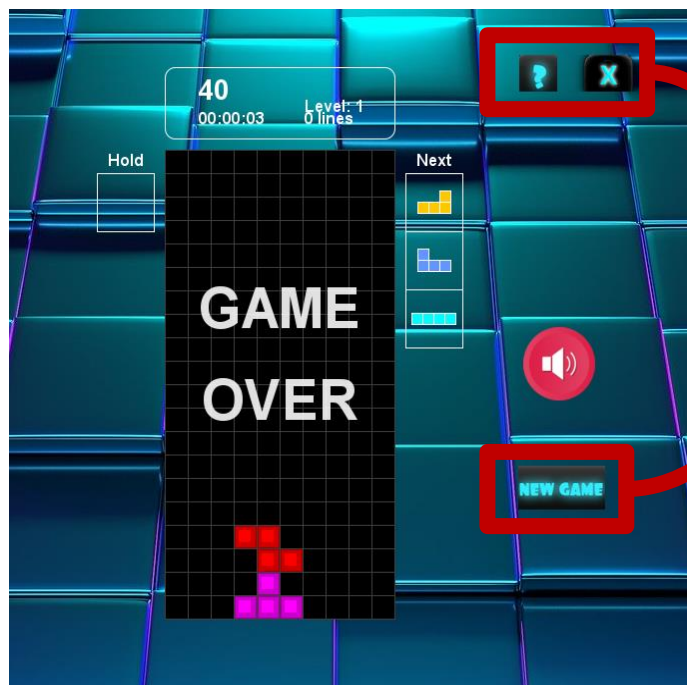


개선 사항(완료)

2020-2-OSSP-HotSource



게임 종료 후 오류 개선

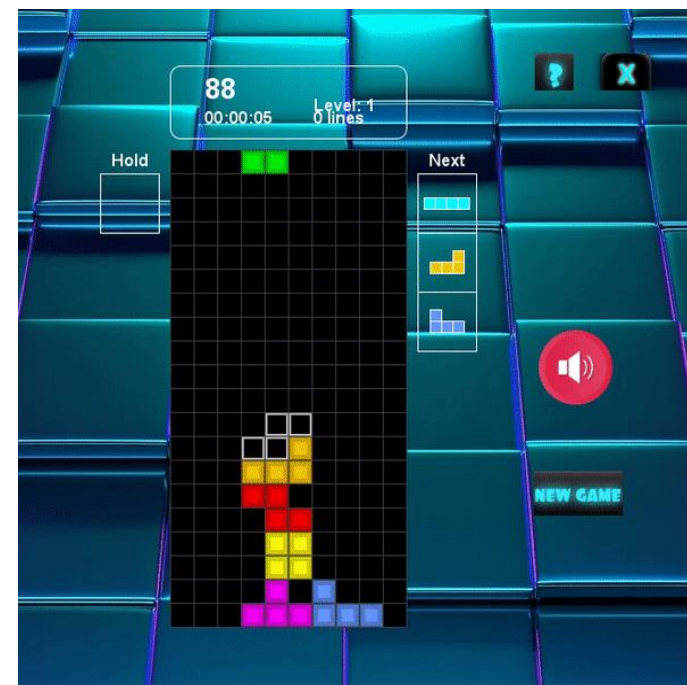


기존 실행화면

게임이 종료 된 후
마우스를 사용할 수 없음

```
// Add Click actionEvent
keyButton.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        launchKeyDialog();
    }
});
newButton.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        bgm_sound.stop();
        launchNewGameDialog();
        bgm_sound.play();
    }
});
homeButton.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        bgm_sound.stop();
        game.die();
        frame.dispose();
        main = new TMain();
    }
});
});
```

코드추가

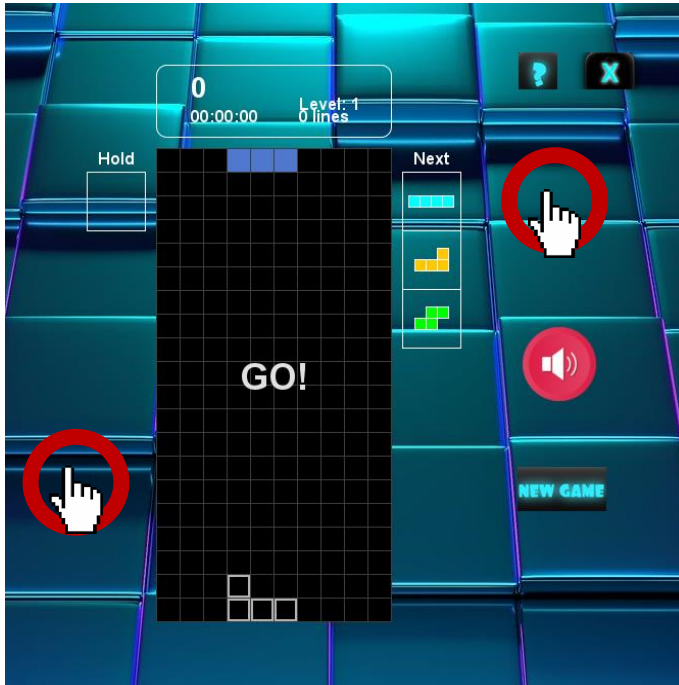


개선된 실행화면

ActionEvent가 제대로 작동

☑️ 정상적인 게임 진행 가능

게임 진행 오류 개선

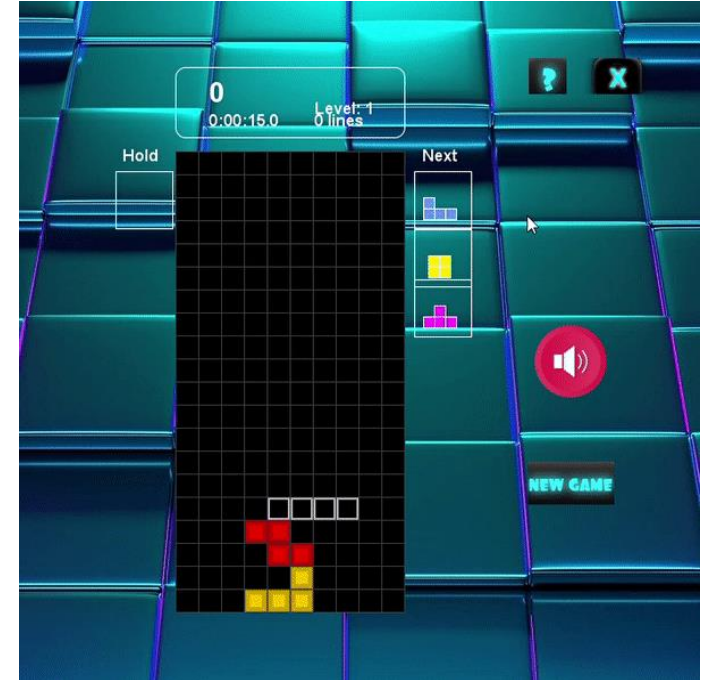


기존 실행화면

배경 클릭시 게임 진행 불가
키보드 먹통

```
background.setBounds(0, 0, 700, 720);
background.setBorderPainted(false);
background.setContentAreaFilled(false);
background.setFocusPainted(false);
background.setVisible(true);
background.setFocusable(false);
```

코드추가



개선된 실행화면

포커싱 오류 개선

☑️ 정상적인 게임 진행 가능

최고 점수 갱신

- 데이터베이스를 다루는 새로운 패키지를 구성
- MyDB에는 insert와 select에 사용되는 리소스들을 관리

```
// RDS SETTING
static String url = "jdbc:mysql://horsourcegamedb.rds.amazonaws.com/test?serverTimezo
static String DB_ID = "yin9931";
static String DB_PW = "yin9931";
static String TABLE = "`horsource_gamedb`.`gamedb`";
```

- DBSelect를 통해 유저의 데이터 중 상위 10개를 가져옴
- DBInsert를 통해 데이터를 RDS 서버로 전송

mysql 드라이버 로딩

RDS에 연결하기

SQL 쿼리 수행

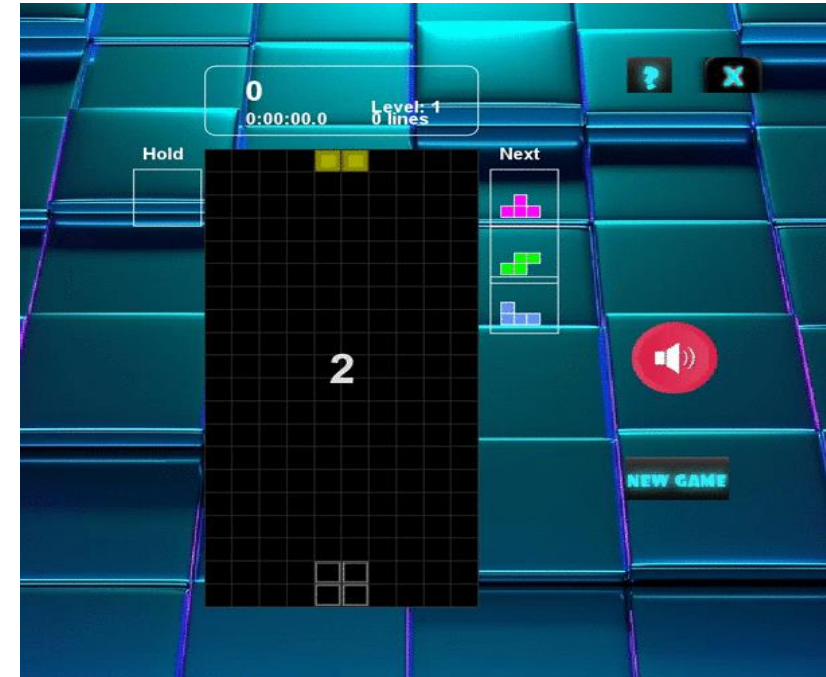
☑ 다른 플레이어들과 경쟁하면서 성취감을 느낄 수 있음

시간 단축에 대한 추가 점수 부여

```

30 public void firmDrop()
31 {
32     if(1 <= ty && ty < 5) {
33         score += (FirmDropPlusScore - (ty * DropDelayMinusScore));
34     }
35     else if(0 >= ty) {
36         score += FirmDropPlusScore;
37     }
38     int oldy = ty;
39     while (pieceLegal() == LEGAL)
40         ty++;
41     ty--;
42     if (oldy != ty)
43     {
44         lastMoveRotate = false;
45         delays = 0;
46     }
47 }
48

```



블록 빠르게 Drop 시 보너스 점수 부여

지연 시간에 따라 최대 20점, 최소 0점

☑ 이 유저에게 더 빠른 판단과 스피드를 요구하며 게임의 승부욕을 북돋아줄 수 있음.

게임 시작 전 준비시간 부여

카운트 다운 띄우기

게임 멈추기

```
// Tetris.java 232 line
private int countdown_number = 3;
```

```
// Tetris.java 564 line
protected void countdown() {
    countdown_number = 3;
```

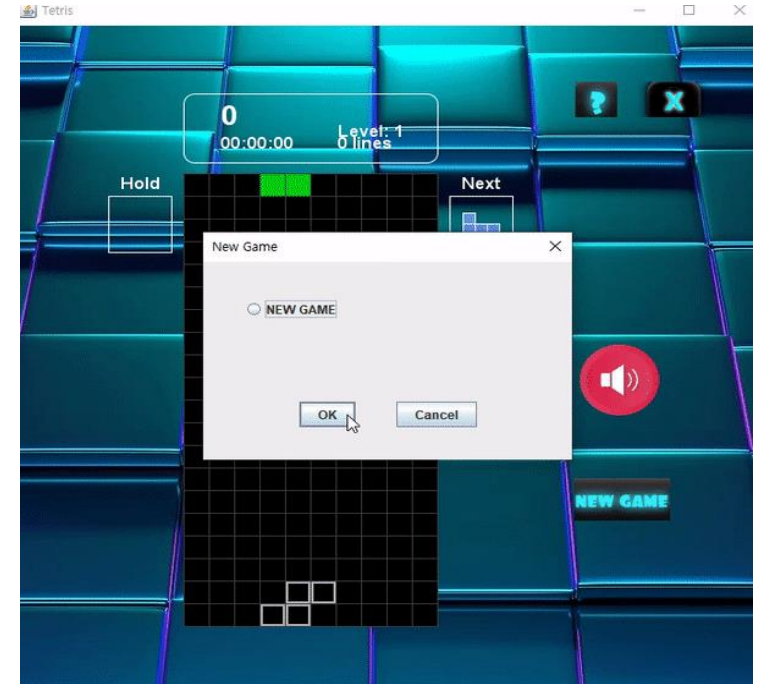
```
// 게임 멈추기
paused = true; // stop game
Timer paused_timer = new Timer();
TimerTask paused_task = new TimerTask() {

    @Override
    public void run() {
        paused = false; // game start!
    };
    paused_timer.schedule(paused_task, delay*(getcount()+1)); //run after 1200*4ms
```

```
// Tetris.java 1108 line
// 카운트 다운 도중 게임을 종료하는 경우를 고려하여 dead==false인지 확인
// countdown number>0일 때는 countdown number를 띄우기

else if (countdown_number>0 && dead==false) {
    g.setColor(new Color(0, 0, 0, 80));
    g.fillRect(x, y, FIELD_W, FIELD_H);
    g.setColor(C_NOTICE);
    drawCentered(g, countdown_number+"", x + FIELD_W / 2, y + 5 + FIELD_H / 2);
}

// 카운트 다운 도중 게임을 종료하는 경우를 고려하여 dead==false인지 확인
// countdown number==0일 때는 GO!값을 띄우기
else if (countdown_number==0 && dead==false) {
    g.setColor(new Color(0, 0, 0, 80));
    g.fillRect(x, y, FIELD_W, FIELD_H);
    g.setColor(C_NOTICE);
    drawCentered(g, "GO!", x + FIELD_W / 2, y + 5 + FIELD_H / 2);
}
```



자바의 Timer, TimerTask객체를 이용
게임 구역의 중심에 3, 2, 1 또는 go! 띄우기

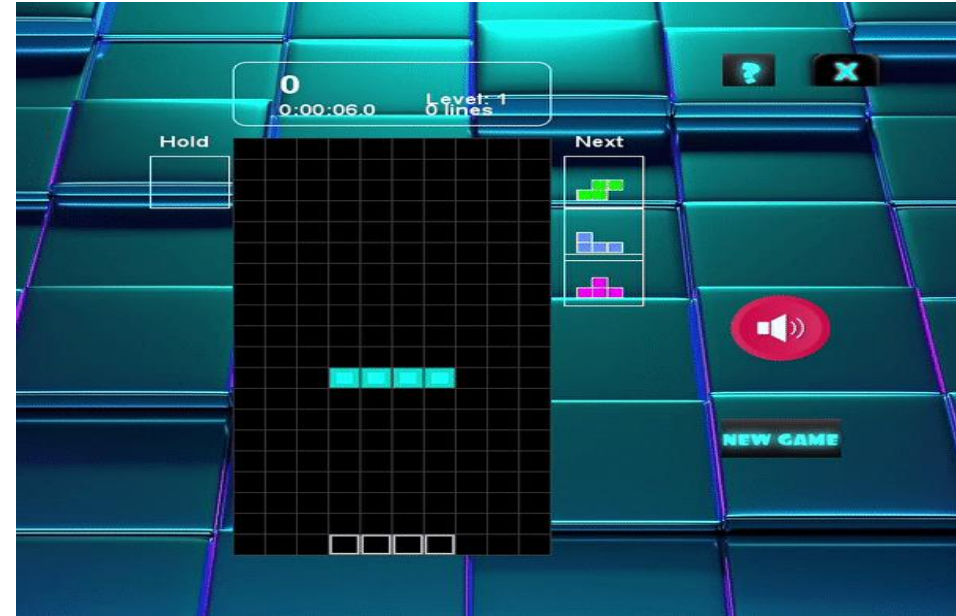
☑ 플레이어가 집중력 있게 게임에 임할 수 있는 환경을 제공

음소거 기능 추가

```
public void play(){
    clip.setFramePosition(0);
    clip.start();
}

public void stop() {
    clip.stop();
}

private void mute() {
    if(soundplay) {
        bgm.stop();
        soundplay = false;
    }else {
        bgm.play();
        soundplay = true;
    }
}
```



게임 화면에 음소거 버튼 추가
버튼을 누를 시 음소거 가능
버튼을 다시 누를시 BGM 재생

☑ 다양한 플레이 환경을 제공



개선사항(진행 중)

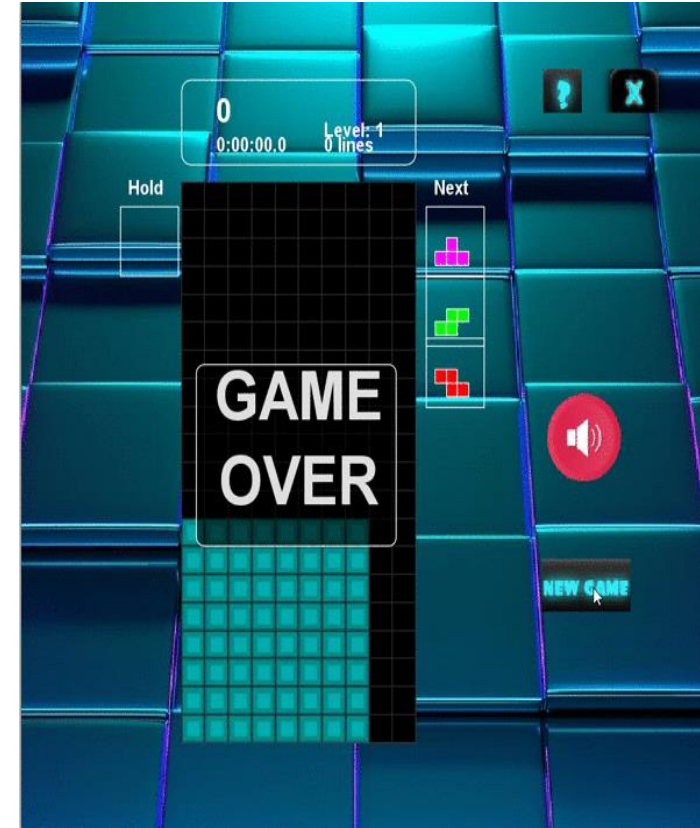
2020-2-OSSP-HotSource



쌓여 있는 블록 구현

```
//게임 시작하자마자 쌓여있는 블록 생성
for(int i=0; i<8; i++) {
    for(int j=12; j<20; j++) {
        board[i][j] = 1;
    }
}
```

=> 8x8 쌓인 블록



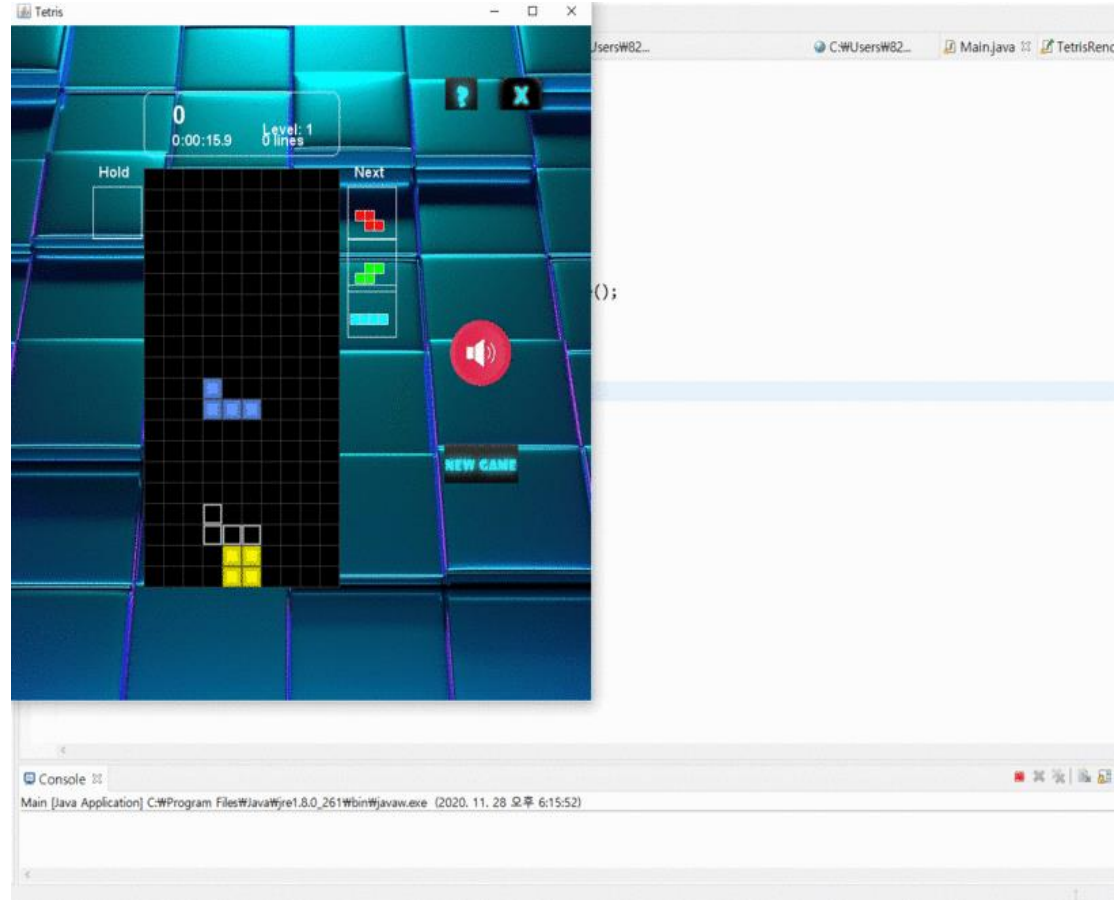
게임의 다양성 향상을 위한

쌓여 있는 블록 모드 기본 구현

Hard 모드 / Normal 모드 나눠서 구현 예정

FrameResizing

- 현재 진행 상황
 - 전체 창의 크기 조절 가능
- 진행 예정
 - 게임화면의 크기도 창의 크기에 비례하여 늘어 날 수 있도록 조정 예정
 - 플레이 화면 여백 삭제





업무 분담 및 일정

2020-2-OSSP-HotSource



HotSource 팀 진행 상황

진행 예정
↓

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE	PERIODS							PARTICIPANTS
						9	10	11	12	13	14	15	
프로젝트 주제 선정 및 제안서 작성	9	1	9	1	100%								김택원,김정률,전영인
게임 오류 사항 수정	10	2	10	2	100%								김택원,김정률,전영인
블록 모드 구현	12	1	13	2	50%								김택원
시작 전 카운트 구현	12	1	12	1	100%								전영인
기록 갱신	12	1	12	1	100%								김정률
시간 단축에 따른 추가 점수 부여	13	1	13	1	100%								김택원,전영인
배경 음악	13	1	13	1	100%								김정률
오류 사항 수정 및 해결 못한 기능 추가	14	1	14	1	100%								김택원,김정률,전영인
검토 및 피드백 / 디버깅	15	1	15	1	100%								김택원,김정률,전영인

차주 진행 사항 : 오류 사항 수정 및 해결 못한 기능 추가

- 쌓인 블록 모드 구현
- FrameResizing

역할 분담



김택원

AWS DB구축

게임 중 오류 사항 수정

보너스 점수 구현

쌓인 블록 모드

FrameResizing



김정률

음소거 기능 구현

게임 중 오류 사항 수정

주석 및 불필요 코드 정리

exe 실행 파일 제작

FrameResizing



전영인

시작 시 카운트 다운 구현

게임 중 오류 사항 수정

JDBC 활용한 랭킹 시스템 구축

불필요한 코드 정리

배경화면 UX/UI 개선



Thank You



⋮ Q&A