

[OSSP]1 차 Progress Report OS

2019.11.14(목)

1.프로젝트 개요

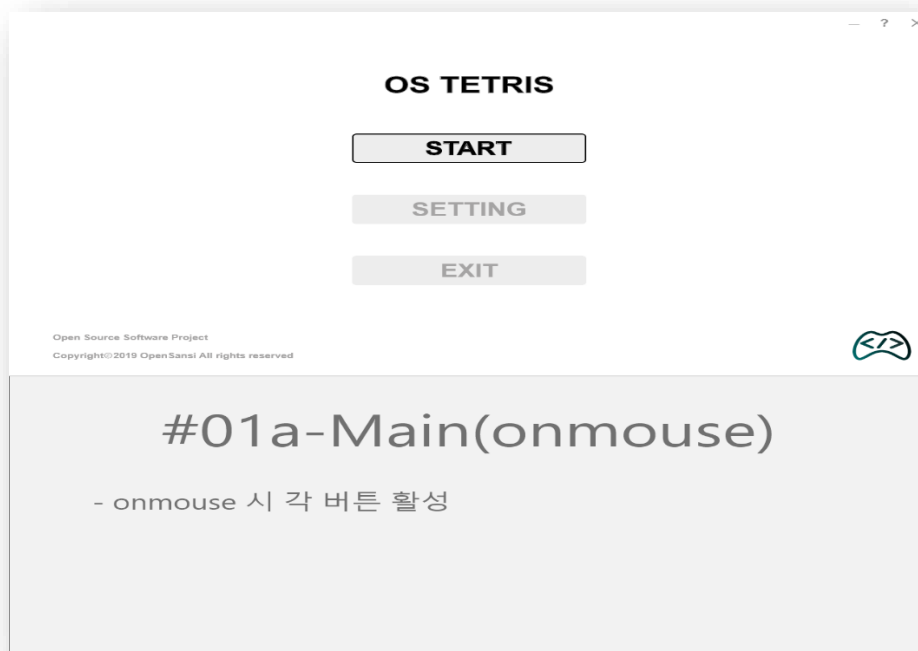
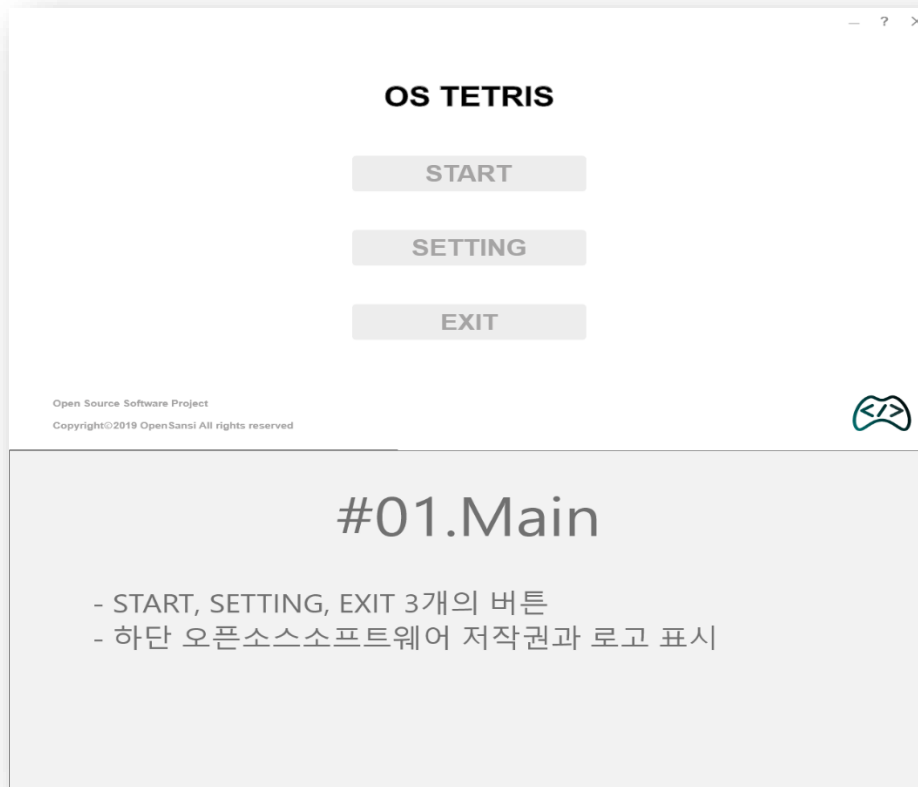
팀명	OpenSansi (OS)
멤버	김선후, 이민지, 남경현
주제	오픈소스를 활용한 테트리스 기능 개발

2.피드백

1. 프로젝트에 대한 구체적인 설계가 필요 (특히, 타이머와 레벨 등의 기능에 대한)
2. 게임 해상도 조절 기능에 대한 검토
- 3.
- 4.
- 5.

3.금주 완료 항목

1) 프로젝트 디자인 정의 및 Mockup 스케치 완료



SETTING

Move Left	Left
Move Right	Right
Rotate Clockwise	Up
Rotate Counterclockwise	Z
Soft Drop	Down
Hard Drop	Space bar
Firm Drop	F
Hold Piece	Shift
Pause	P
Mute on/off	M

cancelApply

Open Source Software Project
Copyright©2019 OpenSansi All rights reserved

#02a-SETTING(onmouse)

- onmouse 시 조작키 변경 활성화

SETTING

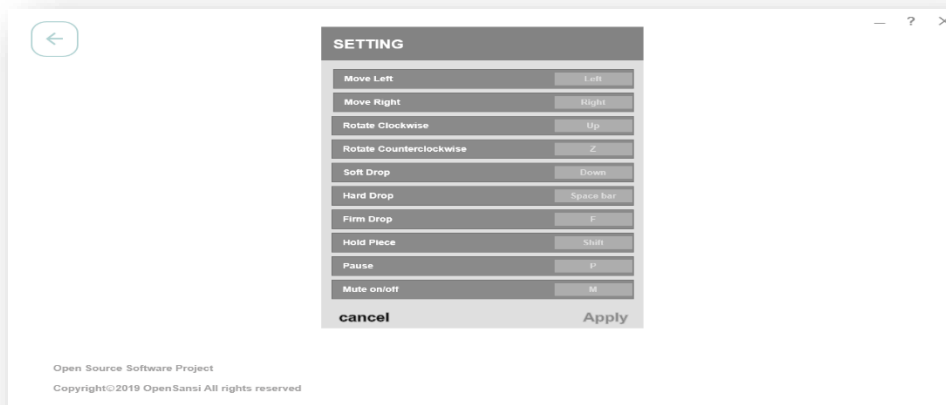
Move Left	
Move Right	Right
Rotate Clockwise	Up
Rotate Counterclockwise	Z
Soft Drop	Down
Hard Drop	Space bar
Firm Drop	F
Hold Piece	Shift
Pause	P
Mute on/off	M

cancelApply

Open Source Software Project
Copyright©2019 OpenSansi All rights reserved

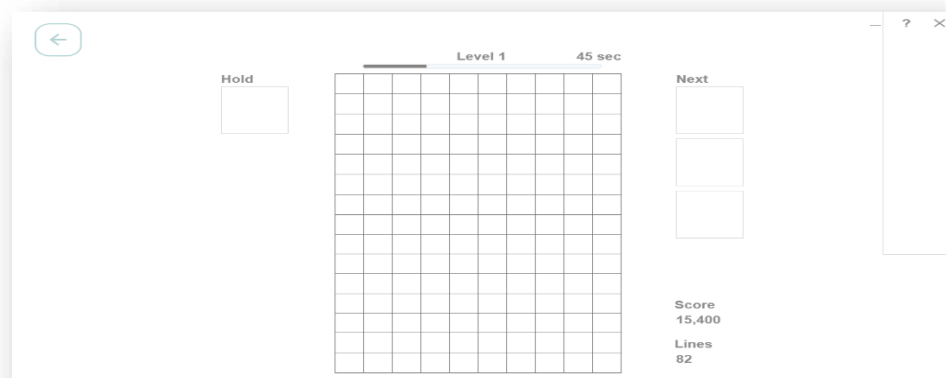
#02b-SETTING(input)

- onclick 시 변경될 키를 입력받음



#02-SETTING

- 기존 기능에 대한 키 셋팅, 음소거(Mute on/off) 추가
- 조작키 미변경 시 Apply 비활성
- 뒤로가기이나 배경, cancel 클릭 시 Main으로 이동



#03-Play

- Hold : 블록 보류
- Next : 다음 블록 표시 (3개)
- Score : 현재 점수 표시
- Level : 현재 레벨 표시 (Score에 따라 증가)

$$a(\text{필요값}). 500 * (\text{Level} + 0.1 * \text{Level}^3) (\text{score})$$
- Timer : 남은 시간 표시 (0이면 게임종료, 레벨업 시 초기화)

$$a(\text{초기값}). 20 * (\text{Level} + 0.05 * \text{Level}^2) (\text{sec})$$
- Lines : 제거한 라인 표시

2) Main Screen UI/UX 변경



그림 1. 기존 PAIS 테트리스 Main Screen UX/UI

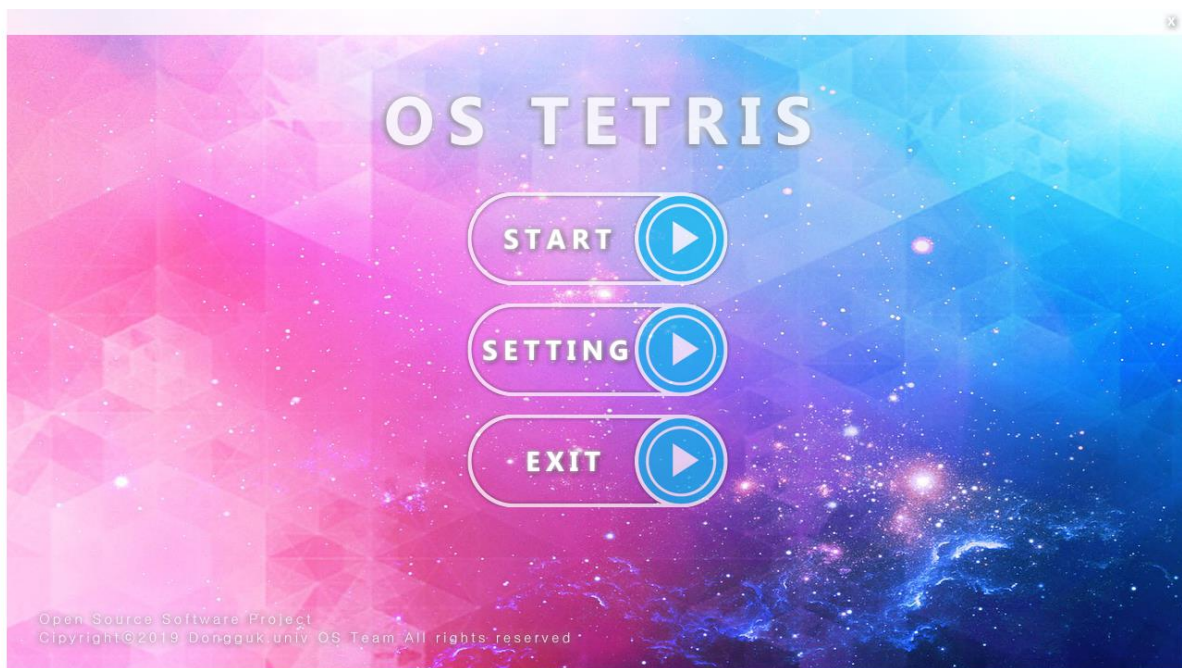


그림 2. OS 테트리스 Main Screen UX/UI

3) 배경음과 효과음 기능 탐색 및 1 차 기능 정의(Level, Timer)

code > rf03.java

```
1  //java sounds
2  //wav 파일만 재생
3
4  public static synchronized void playSound(final String url) {
5      new Thread(new Runnable() {
6          // The wrapper thread is unnecessary, unless it blocks on the
7          // clip finishing; see comments.
8          public void run() {
9              try {
10                 Clip clip = AudioSystem.getClip();
11                 AudioInputStream inputStream = AudioSystem.getAudioInputStream(
12                     Main.class.getResourceAsStream("/path/to/sounds/" + url));
13                 clip.open(inputStream);
14                 clip.start();
15             } catch (Exception e) {
16                 System.err.println(e.getMessage());
17             }
18         }
19     }).start();
20 }
```

code > rf01.java

```
1  //Java sounds
2  // 효과음 삽입 (단일재생)
3
4  try
5  {
6      AudioInputStream ais = AudioSystem.getAudioInputStream(new File("end.wav"));
7      Clip clip = AudioSystem.getClip();
8      clip.stop();
9      clip.open(ais);
10     clip.start();
11 }
12 catch (Exception ex)
13 {
14
15 }
```

```

code > rf02.java
1
2 //java sounds
3 //javax를 사용하여 bgm과 효과음 넣기
4
5 Clip bgmclip;
6 public void playBgm(File file, float vol, boolean repeat){
7     try{
8         //BGM은 임의의 시점에서 정지시킬 수 있어야 하므로 전역으로 전용 clip을 사용한다
9         bgmclip = (Clip)AudioSystem.getLine(new Line.Info(Clip.class));
10        bgmclip.open(AudioSystem.getAudioInputStream(file));
11        bgmclip.addLineListener(new LineListener() {
12            @Override
13            public void update(LineEvent event) {
14                // TODO Auto-generated method stub
15                System.out.println(" " + event.getType());
16                if(event.getType()==LineEvent.Type.STOP){
17                    bgmclip.close();
18                }
19            }
20        });
21        FloatControl volume = (FloatControl)bgmclip.getControl(FloatControl.Type.MASTER_GAIN);
22        volume.setValue(vol);
23        bgmclip.start();
24        if(repeat)
25            bgmclip.loop(bgmclip.LOOP_CONTINUOUSLY);
26    }catch(Exception e){
27        e.printStackTrace();
28    }
29 }
30 public void stopBgm(){
31
32     if(bgmclip!=null && bgmclip.isRunning()){
33         bgmclip.stop();
34         bgmclip.close();
35     }
36 }
37 public void playSound(File file, float vol, boolean repeat){
38     try{
39         final Clip clip = (Clip)AudioSystem.getLine(new Line.Info(Clip.class));
40         clip.open(AudioSystem.getAudioInputStream(file));
41         clip.addLineListener(new LineListener() {
42             @Override
43             public void update(LineEvent event) {
44                 // TODO Auto-generated method stub
45                 if(event.getType()==LineEvent.Type.STOP){
46                     //이 부분이 없으면 효과음이 메모리에 점점 쌓여서 언젠가 크래시된다
47                     clip.close();
48                 }
49             }
50         });
51         FloatControl volume = (FloatControl)clip.getControl(FloatControl.Type.MASTER_GAIN);
52         volume.setValue(vol);
53         clip.start();
54         if(repeat)
55             clip.loop(Clip.LOOP_CONTINUOUSLY);
56     }catch(Exception e){
57         e.printStackTrace();
58     }
59 }

```

```

code > rf04.java > Test > main(String[])
1 //java sounds
2 //반복재생
3
4 import java.io.File;
5 import javax.sound.sampled.AudioSystem;
6 import javax.sound.sampled.Clip;
7 public class Test {
8
9     Run | Debug
10    public static void main(String[] args) {
11
12        File Clap =new File("test.wav");
13        PlaySound(Clap);
14    }
15
16    static void PlaySound(File Sound)
17    {
18        try{
19            Clip clip = AudioSystem.getClip();
20            clip.open(AudioSystem.getAudioInputStream(Sound));
21            clip.start();
22
23            Thread.sleep(clip.getMicrosecondLength()/1000);
24        }
25        catch(Exception e)
26        {
27
28        }
29    }

```

4) 테트리스 Main board 코드 분석 완료 아래의 그림과 같이 각 메스도 및 중요 코드 라인마다 주석 처리를 해 놓았음.

```
public TetrisBoard(Tetris tetris, GameClient _client, boolean isSingle, int[] key_setting) {
    this.tetris = tetris;
    this.client = _client;
    this.isSingle = isSingle;
    if (isSingle == false) { // 멀티모드
    {
        systemMsg = new SystemMessageArea(21, 330, 100, 240);
        messageArea = new MessageArea(this, 500, 460, 360, 130);
        this.add(messageArea); // 컴포넌트 부착 -> 배열 뒤에 데이터 더하기
        this.add(systemMsg); // 컴포넌트 부착
    }
    } else {
        systemMsg = null;
        messageArea = null;
        this.add(comboSpeed);
    }
}

this.enemy = new EnemyBoard();
this.setBounds(0, 30, Main.SCREEN_WIDTH, Main.SCREEN_HEIGHT); // 위치 0, 30 너비 Main.SCREEN_WIDTH, 높이 Main.SCREEN_HEIGHT로 지정
this.setBackground(Color.WHITE);
this.addKeyListener(this);
this.addMouseListener(this);
this.setLayout(null); // 레이아웃설정
this.setFocusable(true); // 키 이벤트의 포커스를 받을 수 있는 컴포넌트가 여러개있을 때 우선적으로 입력받기 위해 설정
this.key_set = key_setting;
```

4.차주 진행 항목

차주 업무 분담

업무 상세	담당자	기한
난이도 조절을 통한 Score 에 대한 상세 정의 배경음과 효과음 샘플 선정 배경음과 효과음 기능 추가 및 테스트(github)	남경현	11 월 21 일(목)
테트리스 실행 스크린 UI/UX 변경 AI 기능 삭제 후 테트리스 싱글 모드의 한 페이지로 변경	김선후	11 월 21 일 (목)
콤보 기능 테트리스 실행 화면에 구현 Score 를 통한 레벨 기능 구현	이민지	1 월 21 일 (목)