SW 프로그래밍

프로젝트

|  |  |
| --- | --- |
| **프로젝트 명** | JAVA Swing를 이용한 TETRIS |
| **팀 명** | 타조 |
| **문서 제목** | 완료 보고서 |

|  |  |
| --- | --- |
| **버전** | 1.0 |
| **날짜** | 2022-06-08 |

|  |  |
| --- | --- |
| **팀원** | 송 재근(팀장) |
| 이 성준 |
| 박 명현 |
| 김 다영 |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**목차**

1. 개요3

1.1 프로젝트 개요3

1. 목표4

2.1 목표4

1. 개발 내용6

3.1 개발 내용6

1. 시스템 구조 설계도8

4.1 시스템 구조도8

1. 제한 요소 및 해결 방안9

5.1 하드웨어9

5.2 소프트웨어9

1. 참고 문헌10
2. 부록11

7.1 사용자 메뉴얼11

7.2 tajoTETRIS SOURCE16

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**1 개요**

**1.1 프로젝트 개요**

본 프로젝트는 그 동안 배운 JAVA Eclipse를 활용하여 오락실 등에서 볼 수 있는 Tetris 프로그램 개발을 목표로 한다. 또한 이 프로젝트는 Semi-Project로써 상업적인 이용이나 배포가 목적이 아니며, 팀원의 프로젝트 경험과 코딩 실력 성장을 위함이다.

별도의 기능 없이 본연의 상(↑), 하(↓), 좌(←), 우(→) 키와 Space Bar 키 만을 이용하여 조작하는 게임이다. 테트리미노가 무작위로 생성되면, 앞서 명시된 키를 이용하여 테트리미노를 조작하여 Tetris 판에 쌓는다. 이 때 판의 가로 줄이 가득 차게 되면 해당 줄은 소거가 되게 된다. 테트리미노를 쌓는 도중 굳어진 테트리미노가 천장에 닿게 되면 게임이 종료된다.

게임 화면에는 Tetris 판과 이동하는 테트리미노, 쌓인 테트리미노들이 보이고, 화면 우측에는 조작 설명과 함께 다음 생성될 테트리미노가 출력된다.

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**2 목표**

**2.1 목표**

본 프로젝트는 사용자가 기존 Tetris와 이질감 없이 게임을 플레이할 수 있고 제작자가 프로젝트를 경험해보고, 그동안 배운 프로그래밍 실력을 유감없이 발휘하고자 한다.

**2.1.1 화면 출력 위치 세부 목표**

* 프로그램을 구동하거나 다른 화면으로 이동할 때 사용자 디스플레이의 정중앙에 출력하는 기능을 제공한다.

**2.1.2 Main화면 세부 목표**

* 화면 상단에 자바 API인 Graphics클래스의 사각형 그림 출력 함수(fill3DRect)를 이용하여 ‘T E T R I S’라는 문자를 출력함으로 Tetris 게임임을 인지할 수 있도록 한다.
* 게임을 시작할 수 있는 게임 시작 버튼과 게임을 종료할 수 있는 게임 종료 버튼을 제공한다.
* 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하게 되어 포인터의 위치를 알 수 있는 기능을 제공한다.
* 버튼 클릭 시 버튼의 색이 변하게 되어 어떤 버튼을 클릭했는지 알 수 있는 기능을 제공한다.

**2.1.3 Ingame화면 세부 목표**

* 화면 좌측에 Tetris 배열 판을 출력한다.
* 화면 우측에 조작법을 제공하여 사용자의 혼란을 줄여준다.
* 다음 테트리미노 예고를 출력하여 다음 턴을 생각할 수 있는 기회를 제공한다.

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**2.1.4 Pause화면 세부 목표**

* 게임을 재 시작할 수 있는 재 시작 버튼과 메인 화면으로 돌아갈 수 있는 메인 화면 버튼을 제공한다.
* 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하게 되어 포인터의 위치를 알 수 있는 기능을 제공한다.
* 버튼 클릭 시 버튼의 색이 변하게 되어 어떤 버튼을 클릭했는지 알 수 있는 기능을 제공한다.

**2.1.5 End화면 세부 목표**

* 화면 상단에 자바 API인 Graphics클래스의 사각형 그림 출력 함수(fill3DRect)를 이용하여 ‘ㅜ\_ㅜ’라는 문자를 출력함으로 Tetris 게임이 끝남을 인지할 수 있고 엔딩에 대한 애도를 표현한다.
* 게임을 메인 화면으로 돌아갈 수 있는 메인 화면 버튼과 게임을 종료할 수 있는 게임 종료 버튼을 제공한다.
* 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하게 되어 포인터의 위치를 알 수 있는 기능을 제공한다.
* 버튼 클릭 시 버튼의 색이 변하게 되어 어떤 버튼을 클릭했는지 알 수 있는 기능을 제공한다.

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**3 개발 내용**

**3.1 개발 내용**

|  |  |
| --- | --- |
| **Contant Class** | * 조작키 움직임, 테트리미노가 움직이는 상태, Tetris 게임 판, 버튼, 테트리미노 낙하 속도, 게임 진행 상태, 사용할 색에 대한 변수를 미리 선언한다. |
| **Block Class** | * 테트리미노의 종류, 크기의 변수와 각 테트리미노의 좌표를 선언한다. |
| **ScreenUtil Class** | * 각 화면이 출력될 때 사용자 디스플레이의 정중앙에 위치하도록 하는 함수를 정의한다. |
| **TetrisManager Class** | * Tetris 게임 판을 출력하고 테트리미노 낙하 속도 조절과 줄 소거 시 색상 변화에 대한 함수를 정의한다. * 화면 우측에 게임 조작에 대한 함수를 정의한다. * 테트리미노가 더 이상 낙하하지 못하는 상황 발생 시 테트리미노가 굳는 함수를 정의한다. * 게임 판의 가로줄이 완성되면 해당 줄이 소거되는 함수를 정의한다. * 3개의 줄을 세는 함수를 정의하고 3 줄이 소거 되면 테트리미노 낙하 속도를 증가시키는 함수를 정의한다. * 낙하를 마치고 굳은 테트리미노가 게임 판의 천장 틀을 넘을 시 게임이 종료되고 그렇지 않을 시 게임이 진행중인 상태를 나타내는 함수를 정의한다. * 테트리미노가 회전할 때 게임 틀을 넘는 상황이 발생 시 회전이 가능할 때까지 테트리미노를 게임 틀 반대 방향으로 이동 후 회전 시키는 함수를 정의한다. |
| **MainMenuPopup Class** | * 프로그램 구동 시 처음으로 나오는 화면이다. * 화면 상단에 자바 API인 Graphics클래스의 사각형 그림 출력 함수(fill3DRect)를 이용하여 ‘T E T R I S’라는 문자를 출력하는 함수를 정의한다. * 게임을 시작할 수 있는 ‘START’ 버튼과 게임을 종료할 수 있는 ‘EXIT’ 버튼을 출력하는 함수를 정의한다. * 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하는 함수를 정의한다. * 버튼 클릭 시 버튼의 색이 변하게 되는 함수를 정의한다. |
| **TetrisView Class** | * 메인 화면에서 ‘START’ 버튼을 클릭 시 출력되는 화면이다. * 화면 출력 시 Tetris 음악을 출력하는 함수를 정의한다. * 게임 상태가 종료 상태일 시 ‘EndMenuPopup’을 출력하는 함수를 정의한다. * 일시정지 상태일 시 음악이 정지되는 함수를 정의한다. * 키보드 입력 시 발생하는 이벤트에 대한 함수를 정의한다. * 화면 좌측에 조작에 대한 설명과 다음 블록 출력에 대한 함수를 정의한다. |
| **PauseMenuPopup Class** | * Ingame화면에서 ESC를 입력하면 출력되는 화면이다. * 게임을 재 시작할 수 있는 ‘RESUME’ 버튼과 메인 화면으로 돌아갈 수 있는 ‘MAINMENU' 버튼을 출력하는 함수를 정의한다. * 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하는 함수를 정의한다. * 버튼 클릭 시 버튼의 색이 변하게 되는 함수를 정의한다. |
| **EndMenuPopup Class** | * Ingame화면에서 Tetris 게임 판의 상단 틀을 넘게 되면 출력되는 화면이다. * 화면 상단에 자바 API인 Graphics클래스의 사각형 그림 출력 함수(fill3DRect)를 이용하여 ‘ㅜ\_ㅜ’라는 문자를 출력하는 함수를 정의한다. * 게임을 메인 화면으로 돌아갈 수 있는 ‘MAINMENU’ 버튼과 게임을 종료할 수 있는 ‘EXIT’ 버튼을 출력하는 함수를 정의한다. * 버튼 제공 시 마우스 포인터가 버튼 위에 위치하게 되면 버튼 색이 변하는 함수를 정의한다. * 버튼 클릭 시 버튼의 색이 변하게 되는 함수를 정의한다. |
| **Main Class** | * 프로그램을 구동하기 위한 메인 클래스이다. * 메인 화면을 출력한다. |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**4 시스템 구조 설계도**

**4.1 시스템 구조도**

**Eclipse**

**Window**

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**5 제한 요소 및 해결 방안**

**5.1 하드웨어**

* Eclipse로 개발하였고, 구동 또한 Eclipse로 하기 때문에 Eclipse가 존재하는 한 실행이 가능하다.
* Eclipse로 구동되기 때문에 모바일에선 사용이 불가하다. 이는 추후에 애플리케이션 개발 실력을 갖춤으로써 해결이 가능하다.
* 사양이 낮은 PC에서는 동시에 여러 개 구동 시 지연 현상을 접할 수 있다. 현재 낮은 사양을 배려한 프로그래밍을 할 수 없기 때문에 사용자의 PC 사양을 높이는 수 밖에 없다.

**5.2 소프트웨어**

* 프로그램 소스에 한글로 주석을 달았기에 GitHub에서 Pull하게 되면 한글이 깨지는 경우를 발견할 수 있는데 인코딩을 UTF-8로 함으로써 깨지는 경우를 대비했다.

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**6 참고 문헌**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
| 1 | 서적 | 명품 JAVA Programming(개정 4판) | 생능출판사 | 2018 | 황기태, 김효수 |  |
| 2 | 웹 | Tetris | github.com  /bill176/tetris |  | Bill176 |  |
| 3 | 영상 | 유튜브 TPhamGameCoding 채널 | www.youtube  .com/c/Tpham  GameCoding | 2020 | Tpham  Game  Coding |  |
| 4 | 영상 | 유튜브 grasper coding 채널 | www.youtube  .com/watch?v  =boAJUSN8fOU | 2019 | Grasper  coding |  |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**7 부록**

**7.1 사용자 매뉴얼**

|  |  |
| --- | --- |
|  | * Eclipse에서 Tetris를 구동하는 화면입니다. * Run을 누르게 되면 게임을 실행할 수 있습니다. |
|  | * 구동 첫 화면입니다. * START 버튼과 EXIT 버튼이 존재하는데 각각 게임 시작과 나가기를 의미합니다. * 마우스 포인터를 버튼 위로 이동하면 버튼의 색이 노란색으로 변하는 것을 볼 수 있습니다. * 버튼을 클릭하면 버튼 색이 변하는 것을 볼 수 있습니다. * 위 두 사례로 마우스 포인터가 어떤 버튼 위에 있는지 확인할 수 있습니다. |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

|  |  |
| --- | --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | * START 버튼을 누른 후 Ingame 화면입니다. * 화면 전환 후 Tetirs 판에서 테트리미노가 낙하하게 되고 게임이 진행됩니다. * 우측에는 조작에 대한 설명과 다음 테트리미노 정보를 확인할 수 있습니다. |
|  | * 첫 테트리미노가 낙하하고 두 번째 테트리미노가 낙하하는 화면입니다. * 위 사진에서 나온 다음 테트리미노와 두 번째 출력된 테트리미노가 동일함을 볼 수 있습니다. |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

|  |  |
| --- | --- |
|  | * 한 줄 완성 시 줄이 소거될 때의 화면입니다. * 굳어진 블록들의 색과 흰색이 반복되며 반짝임 효과를 주면서 줄이 소거됩니다. |
|  | * ESC를 입력했을 때 나오는 일시정지 화면입니다. * 일시정지 화면으로 전환되면 음악과 테트리미노의 낙하가 정지됩니다. * RESUME 버튼과 MAINMENU 버튼이 존재하는데 각각 게임 재시작과 메인 화면으로 돌아가기를 의미합니다. * 메인 화면과 마찬가지로 마우스 포인터가 버튼 위에 존재하거나 버튼을 클릭하면 버튼의 색이 변하게 됩니다. |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

|  |  |
| --- | --- |
|  | * 굳어진 블록들이 천장에 닿았을 경우의 화면입니다. * 굳어진 블록들이 천장에 닿았을 경우 종료화면으로 전환됩니다. |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | * 종료 화면입니다. * MAINMENU 버튼과 EXIT 버튼이 존재하는데 각각 메인 화면으로 돌아가기와 나가기를 의미합니다. * 메인 화면과 마찬가지로 마우스 포인터가 버튼 위에 존재하거나 버튼을 클릭하면 버튼의 색이 변하게 됩니다. |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

|  |  |
| --- | --- |
|  | * 메인 화면과 종료 화면에서 EXIT 버튼을 누르기 직전의 화면입니다. * 사용자가 EXIT 버튼을 누르게 되면 프로그램은 종료되고 사용자의 운영체제 화면으로 전환됩니다. |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**7.2 tajoTETRIS SOURCE**

|  |  |
| --- | --- |
| **GitHub** | [tajoTETRIS SOURCE 바로가기](https://github.com/Bigmeo/-tajo/tree/main/SW%20%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/%EC%BD%94%EB%93%9C) |

**Constant Class**

package constant;

import java.awt.Color;

public interface Constant {

public enum GameStatus {

PLAYING, END, PAUSE

}

public interface Direction {

public static final int SIZE = 4;

public static final int UP = 0;

public static final int RIGHT = 1;

public static final int DOWN = 2;

public static final int LEFT = 3;

}

public enum ProcessType {

DIRECTION, DIRECT\_DOWN, AUTO

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public interface KeyCode {

public static final int UP = 38;

public static final int LEFT = 37;

public static final int RIGHT = 39;

public static final int DOWN = 40;

public static final int SPACE\_BAR = 32;

public static final int ESC = 27;

}

public enum BoardType {

EMPTY, MOVING\_BLOCK, FIXED\_BLOCK, LEFT\_WALL, RIGHT\_WALL, BOTTOM\_WALL, TOP\_WALL, LEFT\_TOP\_EDGE, RIGHT\_TOP\_EDGE, LEFT\_BOTTOM\_EDGE, RIGHT\_BOTTOM\_EDGE

}

public interface MainMenu {

public static final int START = 1;

public static final int EXIT = 2;

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public interface PauseMenu {

public static final int RESUME = 1;

public static final int MAIN\_MENU = 2;

}

public interface EndMenu {

public static final int MAIN\_MENU = 1;

public static final int EXIT = 2;

}

public static final int MAX\_SPEED\_LEVEL = 10;

public static final int MIN\_SPEED\_LEVEL = 1;

public static final Color[] COLORS = { Color.RED, Color.ORANGE,

Color.YELLOW, Color.GREEN, Color.BLUE, Color.decode("#4B0082"),

Color.decode("#800080") };

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**Block Class**

package block;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Point;

import java.util.Random;

import constant.Constant;

public class Block {

private static final int BLOCK\_EXAMPLES\_SIZE = 7;

private static final int POSITIONS\_SIZE = 4;

private static final Point BLOCK\_EXAMPLES[][][] = {

{

{ new Point(0, 5), new Point(0, 6), new Point(0, 7),

new Point(0, 8) },

{ new Point(-1, 6), new Point(0, 6), new Point(1, 6),

new Point(2, 6) },

{ new Point(0, 5), new Point(0, 6), new Point(0, 7),

new Point(0, 8) },

{ new Point(-1, 6), new Point(0, 6), new Point(1, 6),

new Point(2, 6) } },

{

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

{ new Point(0, 8), new Point(1, 6), new Point(1, 7),

new Point(1, 8) },

{ new Point(-1, 7), new Point(0, 7), new Point(1, 7),

new Point(1, 8) },

{ new Point(0, 6), new Point(0, 7), new Point(0, 8),

new Point(1, 6) },

{ new Point(-1, 6), new Point(-1, 7), new Point(0, 7),

new Point(1, 7) } },

{

{ new Point(0, 7), new Point(0, 8), new Point(1, 6),

new Point(1, 7) },

{ new Point(-1, 6), new Point(0, 6), new Point(0, 7),

new Point(1, 7) },

{ new Point(0, 7), new Point(0, 8), new Point(1, 6),

new Point(1, 7) },

{ new Point(-1, 6), new Point(0, 6), new Point(0, 7),

new Point(1, 7) } },

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

{

{ new Point(0, 6), new Point(0, 7), new Point(1, 7),

new Point(1, 8) },

{ new Point(-1, 8), new Point(0, 8), new Point(0, 7),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(1, 7),

new Point(1, 8) },

{ new Point(-1, 8), new Point(0, 8), new Point(0, 7),

new Point(1, 7) } },

{

{ new Point(0, 7), new Point(1, 6), new Point(1, 7),

new Point(1, 8) },

{ new Point(-1, 7), new Point(0, 7), new Point(0, 8),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(0, 8),

new Point(1, 7) },

{ new Point(-1, 7), new Point(0, 6), new Point(0, 7),

new Point(1, 7) } },

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

{

{ new Point(0, 6), new Point(1, 6), new Point(1, 7),

new Point(1, 8) },

{ new Point(-1, 8), new Point(-1, 7), new Point(0, 7),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(0, 8),

new Point(1, 8) },

{ new Point(-1, 7), new Point(0, 7), new Point(1, 7),

new Point(1, 6) } },

{

{ new Point(0, 6), new Point(0, 7), new Point(1, 6),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(1, 6),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(1, 6),

new Point(1, 7) },

{ new Point(0, 6), new Point(0, 7), new Point(1, 6),

new Point(1, 7) } } };

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private Point mPositions[][];

private int mCurrent;

private int mNext;

private int mDirection;

private int mColor;

public Block() {

mPositions = new Point[POSITIONS\_SIZE][POSITIONS\_SIZE];

mCurrent = 0;

mNext = 0;

mDirection = Constant.Direction.UP;

mColor = 0;

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public Block(Block block) {

if (block == null) {

mCurrent = new Random().nextInt(BLOCK\_EXAMPLES\_SIZE);

} else {

mCurrent = block.getNext();

}

mPositions = new Point[POSITIONS\_SIZE][POSITIONS\_SIZE];

for (int i = 0; i < POSITIONS\_SIZE; i++) {

for (int j = 0; j < POSITIONS\_SIZE; j++) {

mPositions[i][j] = new Point(BLOCK\_EXAMPLES[mCurrent][i][j]);

}

}

mNext = new Random().nextInt(BLOCK\_EXAMPLES\_SIZE);

mDirection = Constant.Direction.UP;

mColor = new Random().nextInt(Constant.COLORS.length);

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public void copyOf(Block src) {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

for (int j = 0; j < POSITIONS\_SIZE; j++) {

mPositions[i][j] = new Point(src.getPositions()[i][j]);

}

}

mCurrent = src.getCurrent();

mNext = src.getNext();

mDirection = src.getDirection();

}

public void move(int direction) {

switch (direction) {

case Constant.Direction.LEFT:

moveToLeft();

break;

case Constant.Direction.RIGHT:

moveToRight();

break;

case Constant.Direction.DOWN:

moveToDown();

break;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

case Constant.Direction.UP:

rotateRight();

break;

}

}

public void printNext(Graphics graphics, int x, int y) {

graphics.drawString("[Next block]", x, y);

y += 30;

graphics.setColor(Color.LIGHT\_GRAY);

switch (mNext) {

case 0:

graphics.fill3DRect(x, y, 20, 20, true);

graphics.fill3DRect(x + 20, y, 20, 20, true);

graphics.fill3DRect(x + 40, y, 20, 20, true);

graphics.fill3DRect(x + 60, y, 20, 20, true);

break;

case 1:

graphics.fill3DRect(x + 40, y, 20, 20, true);

graphics.fill3DRect(x, y + 20, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

graphics.fill3DRect(x + 40, y + 20, 20, 20, true);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

break;

case 2:

graphics.fill3DRect(x + 20, y, 20, 20, true);

graphics.fill3DRect(x + 40, y, 20, 20, true);

graphics.fill3DRect(x, y + 20, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

break;

case 3:

graphics.fill3DRect(x, y, 20, 20, true);

graphics.fill3DRect(x + 20, y, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

graphics.fill3DRect(x + 40, y + 20, 20, 20, true);

break;

case 4:

graphics.fill3DRect(x + 20, y, 20, 20, true);

graphics.fill3DRect(x, y + 20, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

graphics.fill3DRect(x + 40, y + 20, 20, 20, true);

break;

case 5:

graphics.fill3DRect(x, y, 20, 20, true);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

graphics.fill3DRect(x, y + 20, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

graphics.fill3DRect(x + 40, y + 20, 20, 20, true);

break;

case 6:

graphics.fill3DRect(x, y, 20, 20, true);

graphics.fill3DRect(x + 20, y, 20, 20, true);

graphics.fill3DRect(x, y + 20, 20, 20, true);

graphics.fill3DRect(x + 20, y + 20, 20, 20, true);

break;

}

graphics.setColor(Color.BLACK);

}

public void setPositions(Point[][] positions) {

mPositions = positions;

}

public Point[][] getPositions() {

return mPositions;

}

public void setCurrent(int current) {

mCurrent = current;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

public int getCurrent() {

return mCurrent;

}

public void setNext(int next) {

mNext = next;

}

public int getNext() {

return mNext;

}

public void setDirection(int direction) {

mDirection = direction;

}

public int getDirection() {

return mDirection;

}

public void setColor(int color) {

mColor = color;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

public int getColor() {

return mColor;

}

private void moveToDown() {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

for (int j = 0; j < POSITIONS\_SIZE; j++) {

mPositions[i][j].x++;

}

}

}

private void moveToLeft() {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

for (int j = 0; j < POSITIONS\_SIZE; j++) {

mPositions[i][j].y--;

}

}

}

private void moveToRight() {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

for (int j = 0; j < POSITIONS\_SIZE; j++) {

mPositions[i][j].y++;

}

}

}

private void rotateRight() {

mDirection = (mDirection + 1) % Constant.Direction.SIZE;

}

}

**ScreenUtil Class**

package util;

import java.awt.Dimension;

import java.awt.Point;

import java.awt.Toolkit;

import java.awt.Window;

public class ScreenUtil {

public static Point getCenterPosition(Window window) {

Dimension wholeScreenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension currentWindowSize = window.getSize();

int left = (wholeScreenSize.width / 2) - (currentWindowSize.width / 2);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

int top = (wholeScreenSize.height / 2) - (currentWindowSize.height / 2);

return new Point(left, top);

}

}

**TetrisManager Class**

package controller;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.Arrays;

import javax.swing.Timer;

import constant.Constant;

import constant.Constant.BoardType;

import constant.Constant.GameStatus;

import block.Block;

public class TetrisManager {

private static final int POSITIONS\_SIZE = 4;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private static final int BOARD\_ROW\_SIZE = 24;

private static final int BOARD\_COL\_SIZE = 14;

private static final int INITIAL\_SPEED = 300;

private static final int SPEED\_LEVEL\_OFFSET = 40;

private static final int LEVEL\_UP\_CONDITION = 3;

private static final int LINES\_TO\_DELETE\_HIGHLIGHTING\_MILLISECOND = 10;

private Constant.BoardType mBoard[][];

private Block mBlock;

private int mDeletedLineCount;

private int mSpeedLevel;

private int mColorIndex = 0;

public TetrisManager(int speedLevel) {

mBoard = new Constant.BoardType[BOARD\_ROW\_SIZE][BOARD\_COL\_SIZE];

for (int i = 0; i < BOARD\_ROW\_SIZE; i++) {

Arrays.fill(mBoard[i], BoardType.EMPTY);

}

clearBoard();

mBlock = new Block(null);

mDeletedLineCount = 0;

mSpeedLevel = speedLevel;

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public Constant.BoardType checkValidPosition(int direction) {

Block temp = new Block();

temp.copyOf(mBlock);

temp.move(direction);

for (int i = 0; i < POSITIONS\_SIZE; i++) {

int x = temp.getPositions()[temp.getDirection()][i].x;

int y = temp.getPositions()[temp.getDirection()][i].y;

//

if (x <= 0) {

return Constant.BoardType.TOP\_WALL;

}

if (!(mBoard[x][y] == Constant.BoardType.EMPTY || mBoard[x][y] == Constant.BoardType.MOVING\_BLOCK)) {

return mBoard[x][y];

}

}

return Constant.BoardType.EMPTY;

}

public void changeBoardByDirection(int direction) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

int tempDirection = Constant.Direction.DOWN;

Constant.BoardType tempCheckResult = Constant.BoardType.EMPTY;

clearBoard();

Constant.BoardType checkResult = checkValidPosition(direction);

if (checkResult == Constant.BoardType.EMPTY) {

mBlock.move(direction);

} else {

if (direction == Constant.Direction.UP

&& checkResult != Constant.BoardType.FIXED\_BLOCK) {

if (checkResult == Constant.BoardType.TOP\_WALL) {

tempDirection = Constant.Direction.DOWN;

tempCheckResult = Constant.BoardType.TOP\_WALL;

} else if (checkResult == Constant.BoardType.RIGHT\_WALL) {

tempDirection = Constant.Direction.LEFT;

tempCheckResult = Constant.BoardType.RIGHT\_WALL;

} else if (checkResult == Constant.BoardType.LEFT\_WALL) {

tempDirection = Constant.Direction.RIGHT;

tempCheckResult = Constant.BoardType.LEFT\_WALL;

}

do {

mBlock.move(tempDirection);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

} while (checkValidPosition(direction) == tempCheckResult);

mBlock.move(direction);

}

}

changeBoardByStatus(Constant.BoardType.MOVING\_BLOCK);

}

public void changeBoardByAuto() {

changeBoardByDirection(Constant.Direction.DOWN);

}

public void processDirectDown() {

while (!isReachedToBottom()) {

changeBoardByDirection(Constant.Direction.DOWN);

}

}

public void processDeletingLines(Graphics g) {

Color highlightingColors[] = { Color.GRAY, Color.WHITE };

ArrayList<Integer> indexes = new ArrayList<Integer>();

searchLineIndexesToDelete(indexes);

if (indexes.size() > 0) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

Timer timer = new Timer(LINES\_TO\_DELETE\_HIGHLIGHTING\_MILLISECOND,

new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

highlightLinesToDelete(g,

highlightingColors[mColorIndex], indexes);

mColorIndex = 1 - mColorIndex;

}

});

timer.start();

try {

Thread.sleep(LINES\_TO\_DELETE\_HIGHLIGHTING\_MILLISECOND \* 40);

} catch (InterruptedException e1) {

}

timer.stop();

deleteLines(indexes);

for (int i = mSpeedLevel; i <= mDeletedLineCount

/ LEVEL\_UP\_CONDITION; i++) {

upSpeedLevel();

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

}

public boolean isReachedToBottom() {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

int x = mBlock.getPositions()[mBlock.getDirection()][i].x;

int y = mBlock.getPositions()[mBlock.getDirection()][i].y;

if (mBoard[x + 1][y] != Constant.BoardType.EMPTY

&& mBoard[x + 1][y] != Constant.BoardType.MOVING\_BLOCK) {

return true;

}

}

return false;

}

public Constant.GameStatus processReachedCase() {

changeBoardByStatus(Constant.BoardType.FIXED\_BLOCK);

mBlock = new Block(mBlock);

if (isReachedToBottom()) {

return GameStatus.END;

} else {

return GameStatus.PLAYING;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

}

public void sleep() {

try {

Thread.sleep(getDownMilliSecond());

} catch (InterruptedException e) {

e.printStackTrace();

}

}

public void print(Graphics graphics) {

int x;

int y = 60;

for (int i = 0; i < BOARD\_ROW\_SIZE; i++) {

x = 30;

for (int j = 0; j < BOARD\_COL\_SIZE; j++) {

switch (mBoard[i][j]) {

case LEFT\_TOP\_EDGE:

case RIGHT\_TOP\_EDGE:

case LEFT\_BOTTOM\_EDGE:

case RIGHT\_BOTTOM\_EDGE:

case LEFT\_WALL:

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

case RIGHT\_WALL:

case TOP\_WALL:

case BOTTOM\_WALL:

graphics.fill3DRect(x, y, 25, 25, true);

break;

case EMPTY:

break;

case MOVING\_BLOCK:

graphics.setColor(Constant.COLORS[mBlock.getColor()]);

graphics.fill3DRect(x, y, 25, 25, true);

graphics.setColor(Color.BLACK);

break;

case FIXED\_BLOCK:

graphics.setColor(Color.GRAY);

graphics.fill3DRect(x, y, 25, 25, true);

graphics.setColor(Color.BLACK);

break;

}

x += 25;

}

y += 25;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

x = 460;

y = 150;

Font font = graphics.getFont();

graphics.setFont(new Font(font.getName(), Font.BOLD, 20));

y += 80;

graphics.drawString("[Key Description]", x, y);

y += 30;

graphics.drawString("←", x, y);

x = 560;

graphics.drawString(": 왼쪽 이동", x, y);

x = 460;

y += 30;

graphics.drawString("→", x, y);

x = 560;

graphics.drawString(": 오른쪽 이동", x, y);

x = 460;

y += 30;

graphics.drawString("↓", x, y);

x = 560;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

graphics.drawString(": 아래 이동", x, y);

x = 460;

y += 30;

graphics.drawString("↑", x, y);

x = 560;

graphics.drawString(": 블럭 회전", x, y);

x = 460;

y += 30;

graphics.drawString("SpaceBar", x, y);

x = 560;

graphics.drawString(": 블럭 낙하", x, y);

x = 460;

y += 30;

graphics.drawString("ESC", x, y);

x = 560;

graphics.drawString(": pause", x, y);

x = 460;

mBlock.printNext(graphics, x, y + 80);

}

public void setBoard(Constant.BoardType[][] board) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

mBoard = board;

}

public Constant.BoardType[][] getBoard() {

return mBoard;

}

public void setBlock(Block block) {

mBlock = block;

}

public Block getBlock() {

return mBlock;

}

public void setDeletedLineCount(int deletedLineCount) {

mDeletedLineCount = deletedLineCount;

}

public int getDeletedLineCount() {

return mDeletedLineCount;

}

public void setSpeedLevel(int speedLevel) {

mSpeedLevel = speedLevel;

}

public int getSpeedLevel() {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

return mSpeedLevel;

}

public long getDownMilliSecond() {

long milliSecond = INITIAL\_SPEED;

for (int i = Constant.MIN\_SPEED\_LEVEL; i < mSpeedLevel; i++) {

if (i < Constant.MAX\_SPEED\_LEVEL / 2) {

milliSecond -= SPEED\_LEVEL\_OFFSET;

} else {

milliSecond -= (SPEED\_LEVEL\_OFFSET / 5);

}

}

return milliSecond;

}

private void clearBoard() {

for (int i = 0; i < BOARD\_ROW\_SIZE; i++) {

mBoard[i][0] = Constant.BoardType.LEFT\_WALL;

mBoard[i][BOARD\_COL\_SIZE - 1] = Constant.BoardType.RIGHT\_WALL;

}

for (int i = 0; i < BOARD\_COL\_SIZE; i++) {

mBoard[0][i] = Constant.BoardType.TOP\_WALL;

mBoard[BOARD\_ROW\_SIZE - 1][i] = Constant.BoardType.BOTTOM\_WALL;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

for (int i = 1; i < BOARD\_ROW\_SIZE - 1; i++) {

for (int j = 1; j < BOARD\_COL\_SIZE - 1; j++) {

if (mBoard[i][j] != Constant.BoardType.FIXED\_BLOCK) {

mBoard[i][j] = Constant.BoardType.EMPTY;

}

}

}

mBoard[0][0] = Constant.BoardType.LEFT\_TOP\_EDGE;

mBoard[0][BOARD\_COL\_SIZE - 1] = Constant.BoardType.RIGHT\_TOP\_EDGE;

mBoard[BOARD\_ROW\_SIZE - 1][0] = Constant.BoardType.LEFT\_BOTTOM\_EDGE;

mBoard[BOARD\_ROW\_SIZE - 1][BOARD\_COL\_SIZE - 1] = Constant.BoardType.RIGHT\_BOTTOM\_EDGE;

}

private void changeBoardByStatus(Constant.BoardType status) {

for (int i = 0; i < POSITIONS\_SIZE; i++) {

int x = mBlock.getPositions()[mBlock.getDirection()][i].x;

int y = mBlock.getPositions()[mBlock.getDirection()][i].y;

mBoard[x][y] = status;

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private void upSpeedLevel() {

if (mSpeedLevel < Constant.MAX\_SPEED\_LEVEL) {

mSpeedLevel++;

}

}

private void searchLineIndexesToDelete(ArrayList<Integer> indexes) {

indexes.clear();

for (int i = 1; i < BOARD\_ROW\_SIZE - 1; i++) {

boolean toDelete = true;

for (int j = 1; j < BOARD\_COL\_SIZE - 1; j++) {

if (mBoard[i][j] != Constant.BoardType.FIXED\_BLOCK) {

toDelete = false;

break;

}

}

if (toDelete) {

indexes.add(i);

}

}

}

private void deleteLines(ArrayList<Integer> indexes) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

int k = BOARD\_ROW\_SIZE - 2;

Constant.BoardType[][] temp = new Constant.BoardType[BOARD\_ROW\_SIZE][BOARD\_COL\_SIZE];

for (int i = 0; i < BOARD\_ROW\_SIZE; i++) {

Arrays.fill(temp[i], Constant.BoardType.EMPTY);

}

for (int i = BOARD\_ROW\_SIZE - 2; i > 0; i--) {

boolean toDelete = false;

for (int j = 0; j < indexes.size(); j++) {

if (i == indexes.get(j)) {

toDelete = true;

break;

}

}

if (!toDelete) {

for (int j = 0; j < BOARD\_COL\_SIZE; j++) {

temp[k][j] = mBoard[i][j];

}

k--;

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

for (int i = 1; i < BOARD\_ROW\_SIZE - 1; i++) {

for (int j = 1; j < BOARD\_COL\_SIZE - 1; j++) {

mBoard[i][j] = temp[i][j];

}

}

mDeletedLineCount += indexes.size();

}

private void highlightLinesToDelete(Graphics g, Color color,

ArrayList<Integer> indexes) {

g.setColor(color);

int x = 55;

int y = 60 + indexes.get(0) \* 25;

g.fill3DRect(x, y, 25 \* (BOARD\_COL\_SIZE - 2), 25 \* indexes.size(), true);

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**MainMenuPopup Class**

package view;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.WindowConstants;

import util.ScreenUtil;

public class MainMenuPopup extends JDialog {

private JButton mjbStart;

private JButton mjbExit;

public MainMenuPopup() {

initWholeSetting();

initMembers();

setEvents();

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private void initWholeSetting() {

setTitle("TETRIS - MAIN MENU");

setModal(true);

setLayout(null);

setSize(500, 400);

setLocation(ScreenUtil.getCenterPosition(this));

setResizable(false);

setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

}

private void initMembers() {

MyMouseListener listener = new MyMouseListener();

mjbStart = new JButton("S T A R T");

mjbStart.setBounds(150, 230, 200, 30);

mjbStart.setBackground(Color.WHITE);

mjbStart.addMouseListener(listener);

add(mjbStart);

mjbExit = new JButton("E X I T");

mjbExit.setBounds(150, 260, 200, 30);

mjbExit.setBackground(Color.WHITE);

mjbExit.addMouseListener(listener);

add(mjbExit);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

class MyMouseListener implements MouseListener{

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

JButton mjbStart = (JButton)e.getSource();

JButton mjbExit = (JButton)e.getSource();

mjbStart.setBackground(Color.YELLOW);

mjbExit.setBackground(Color.YELLOW);

}

@Override

public void mouseExited(MouseEvent e) {

JButton mjbStart = (JButton)e.getSource();

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

JButton mjbExit = (JButton)e.getSource();

mjbStart.setBackground(Color.WHITE);

mjbExit.setBackground(Color.WHITE);

}

}

private void setEvents() {

mjbStart.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

new TetrisView().setVisible(true);

}

});

mjbExit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

}

});

}

@Override

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public void paint(Graphics g) {

super.paint(g);

g.setColor(Color.RED);

g.fill3DRect(40, 70, 20, 20, true);

g.fill3DRect(60, 70, 20, 20, true);

g.fill3DRect(80, 70, 20, 20, true);

g.fill3DRect(60, 90, 20, 20, true);

g.fill3DRect(60, 110, 20, 20, true);

g.fill3DRect(60, 130, 20, 20, true);

g.fill3DRect(60, 150, 20, 20, true);

g.setColor(Color.ORANGE);

g.fill3DRect(120, 70, 20, 20, true);

g.fill3DRect(140, 70, 20, 20, true);

g.fill3DRect(160, 70, 20, 20, true);

g.fill3DRect(120, 90, 20, 20, true);

g.fill3DRect(120, 110, 20, 20, true);

g.fill3DRect(140, 110, 20, 20, true);

g.fill3DRect(160, 110, 20, 20, true);

g.fill3DRect(120, 130, 20, 20, true);

g.fill3DRect(120, 150, 20, 20, true);

g.fill3DRect(140, 150, 20, 20, true);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

g.fill3DRect(160, 150, 20, 20, true);

g.setColor(Color.YELLOW);

g.fill3DRect(200, 70, 20, 20, true);

g.fill3DRect(220, 70, 20, 20, true);

g.fill3DRect(240, 70, 20, 20, true);

g.fill3DRect(220, 90, 20, 20, true);

g.fill3DRect(220, 110, 20, 20, true);

g.fill3DRect(220, 130, 20, 20, true);

g.fill3DRect(220, 150, 20, 20, true);

g.setColor(Color.GREEN);

g.fill3DRect(280, 70, 20, 20, true);

g.fill3DRect(300, 70, 20, 20, true);

g.fill3DRect(280, 90, 20, 20, true);

g.fill3DRect(310, 90, 20, 20, true);

g.fill3DRect(280, 110, 20, 20, true);

g.fill3DRect(300, 110, 20, 20, true);

g.fill3DRect(280, 130, 20, 20, true);

g.fill3DRect(310, 130, 20, 20, true);

g.fill3DRect(280, 150, 20, 20, true);

g.fill3DRect(320, 150, 20, 20, true);

g.setColor(Color.BLUE);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

g.fill3DRect(360, 70, 20, 20, true);

g.fill3DRect(360, 90, 20, 20, true);

g.fill3DRect(360, 110, 20, 20, true);

g.fill3DRect(360, 130, 20, 20, true);

g.fill3DRect(360, 150, 20, 20, true);

g.setColor(Color.PINK);

g.fill3DRect(410, 70, 20, 20, true);

g.fill3DRect(430, 70, 20, 20, true);

g.fill3DRect(400, 90, 20, 20, true);

g.fill3DRect(400, 110, 20, 20, true);

g.fill3DRect(420, 110, 20, 20, true);

g.fill3DRect(440, 110, 20, 20, true);

g.fill3DRect(440, 130, 20, 20, true);

g.fill3DRect(430, 150, 20, 20, true);

g.fill3DRect(410, 150, 20, 20, true);

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**PauseMenuPopup Class**

package view;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDialog;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import util.ScreenUtil;

public class PauseMenuPopup extends JDialog {

private TetrisView tetrisView;

private JButton mjbResume;

private JButton mjbMainMenu;

public PauseMenuPopup(TetrisView tetrisView) {

initWholeSetting();

initMembers(tetrisView);

setEvents();

}

private void initWholeSetting() {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

setUndecorated(true);

setModal(true);

setLayout(null);

setSize(200, 60);

setLocation(ScreenUtil.getCenterPosition(this));

}

private void initMembers(TetrisView tetrisView) {

this.tetrisView = tetrisView;

MyMouseListener listener = new MyMouseListener();

mjbResume = new JButton("R E S U M E");

mjbResume.setBounds(0, 0, 200, 30);

mjbResume.setBackground(Color.WHITE);

mjbResume.addMouseListener(listener);

add(mjbResume);

mjbMainMenu = new JButton("M A I N M E N U");

mjbMainMenu.setBounds(0, 30, 200, 30);

mjbMainMenu.setBackground(Color.WHITE);

mjbMainMenu.addMouseListener(listener);

add(mjbMainMenu);

}

class MyMouseListener implements MouseListener{

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

JButton mjbResume = (JButton)e.getSource();

JButton mjbMainMenu = (JButton)e.getSource();

mjbResume.setBackground(Color.YELLOW);

mjbMainMenu.setBackground(Color.YELLOW);

}

@Override

public void mouseExited(MouseEvent e) {

JButton mjbResume = (JButton)e.getSource();

JButton mjbMainMenu = (JButton)e.getSource();

mjbResume.setBackground(Color.WHITE);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

mjbMainMenu.setBackground(Color.WHITE);

}

}

private void setEvents() {

mjbResume.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

}

});

mjbMainMenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

tetrisView.end();

new MainMenuPopup().setVisible(true);

}

});

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**EndMenuPopup Class**

package view;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.WindowConstants;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import util.ScreenUtil;

public class EndMenuPopup extends JDialog {

private JButton mjbMainMenu;

private JButton mjbExit;

public EndMenuPopup() {

initWholeSetting();

initMembers();

setEvents();

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private void initWholeSetting() {

setTitle("TETRIS - END MENU");

setModal(true);

setLayout(null);

setSize(500, 400);

setLocation(ScreenUtil.getCenterPosition (this));

setResizable(false);

setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

}

private void initMembers() {

MyMouseListener listener = new MyMouseListener();

mjbMainMenu = new JButton("M A I N M E N U");

mjbMainMenu.setBounds(150, 240, 200, 30);

mjbMainMenu.setBackground(Color.WHITE);

mjbMainMenu.addMouseListener(listener);

add(mjbMainMenu);

mjbExit = new JButton("E X I T");

mjbExit.setBounds(150, 270, 200, 30);

mjbExit.setBackground(Color.WHITE);

mjbExit.addMouseListener(listener);

add(mjbExit);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

class MyMouseListener implements MouseListener{

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

JButton mjbMainMenu = (JButton)e.getSource();

JButton mjbExit = (JButton)e.getSource();

mjbMainMenu.setBackground(Color.YELLOW);

mjbExit.setBackground(Color.YELLOW);

}

@Override

public void mouseExited(MouseEvent e) {

JButton mjbMainMenu = (JButton)e.getSource();

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

JButton mjbExit = (JButton)e.getSource();

mjbMainMenu.setBackground(Color.WHITE);

mjbExit.setBackground(Color.WHITE);

}

}

private void setEvents() {

mjbMainMenu.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

new MainMenuPopup().setVisible(true);

}

});

mjbExit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

}

});

}

@Override

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public void paint(Graphics g) {

super.paint(g);

g.fill3DRect(145, 90, 20, 20, true);

g.fill3DRect(165, 90, 20, 20, true);

g.fill3DRect(185, 90, 20, 20, true);

g.fill3DRect(165, 110, 20, 20, true);

g.fill3DRect(165, 130, 20, 20, true);

g.fill3DRect(165, 150, 20, 20, true);

g.fill3DRect(165, 170, 20, 20, true);

g.fill3DRect(225, 170, 20, 20, true);

g.fill3DRect(245, 170, 20, 20, true);

g.fill3DRect(265, 170, 20, 20, true);

g.fill3DRect(305, 90, 20, 20, true);

g.fill3DRect(325, 90, 20, 20, true);

g.fill3DRect(345, 90, 20, 20, true);

g.fill3DRect(325, 110, 20, 20, true);

g.fill3DRect(325, 130, 20, 20, true);

g.fill3DRect(325, 150, 20, 20, true);

g.fill3DRect(325, 170, 20, 20, true);

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**TetrisView Class**

package view;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.io.File;

import java.io.IOException;

import javax.sound.sampled.AudioSystem;

import javax.sound.sampled.Clip;

import javax.sound.sampled.LineUnavailableException;

import javax.sound.sampled.UnsupportedAudioFileException;

import javax.swing.JFrame;

import constant.Constant;

import controller.TetrisManager;

import util.ScreenUtil;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

public class TetrisView extends JFrame {

private final Object mMonitorObject = new Object(); // to pause <-> resume

private static final String TETRIS\_BACKGROUND\_MUSIC\_FILE\_NAME = ".\\res\\tetris\_background\_music.wav";

private static final int PROCESS\_REACHED\_CASE\_COUNT = 2;

private long mCurrentTimeMilliSecond;

private Clip mSoundClip;

private TetrisManager mTetrisManager;

private Constant.ProcessType mProcessType;

private int mDirection;

private Constant.GameStatus mGameStatus;

private boolean mIsKeyPressed;

private int mProcessReachedCaseCount;

public TetrisView() {

initWholeSetting();

initMembers();

setEvents();

}

public void start() {

mSoundClip.start();

mSoundClip.loop(Clip.LOOP\_CONTINUOUSLY);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

repaint();

}

public void process(Constant.ProcessType processType, int direction) {

if (processType == Constant.ProcessType.DIRECTION) {

mTetrisManager.changeBoardByDirection(direction);

} else if (processType == Constant.ProcessType.DIRECT\_DOWN) {

mTetrisManager.processDirectDown();

} else if (processType == Constant.ProcessType.AUTO) {

mTetrisManager.changeBoardByAuto();

}

if (mTetrisManager.isReachedToBottom()) {

if (processType == Constant.ProcessType.DIRECT\_DOWN) {

mProcessReachedCaseCount = 0;

if (mTetrisManager.processReachedCase() == Constant.GameStatus.END) {

end();

new EndMenuPopup().setVisible(true);

return;

}

} else {

if (mProcessReachedCaseCount == PROCESS\_REACHED\_CASE\_COUNT) {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

if (mTetrisManager.processReachedCase() == Constant.GameStatus.END) {

end();

new EndMenuPopup().setVisible(true);

return;

}

mProcessReachedCaseCount = 0;

} else {

mProcessReachedCaseCount++;

}

}

}

repaint();

mTetrisManager.processDeletingLines(getGraphics());

}

public void end() {

mGameStatus = Constant.GameStatus.END;

mSoundClip.stop();

dispose();

}

public void pause() {

mGameStatus = Constant.GameStatus.PAUSE;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

mSoundClip.stop();

new PauseMenuPopup(this).setVisible(true);

synchronized (mMonitorObject) {

mMonitorObject.notify();

}

if (mGameStatus != Constant.GameStatus.END) {

mGameStatus = Constant.GameStatus.PLAYING;

mSoundClip.start();

mSoundClip.loop(Clip.LOOP\_CONTINUOUSLY);

}

}

@Override

public void paint(Graphics g) {

Image buffer = createImage(getWidth(), getHeight());

Graphics graphics = buffer.getGraphics();

graphics.setColor(Color.black);

Font font = graphics.getFont();

graphics.setFont(new Font(font.getName(), Font.BOLD, 30));

mTetrisManager.print(graphics);

g.drawImage(buffer, 0, 0, this);

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

private void initWholeSetting() {

setTitle("TETRIS - ING...");

getContentPane().setLayout(null);

setSize(800, 700);

setLocation(ScreenUtil.getCenterPosition(this));

setResizable(false);

}

private void initMembers() {

try {

mSoundClip = AudioSystem.getClip();

mSoundClip.open(AudioSystem.getAudioInputStream(new File(

TETRIS\_BACKGROUND\_MUSIC\_FILE\_NAME)));

} catch (LineUnavailableException lue) {

lue.printStackTrace();

} catch (IOException ioe) {

ioe.printStackTrace();

} catch (UnsupportedAudioFileException uafe) {

uafe.printStackTrace();

}

mTetrisManager = new TetrisManager(Constant.MIN\_SPEED\_LEVEL);

new Thread(new Runnable() {

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

@Override

public void run() {

start();

mGameStatus = Constant.GameStatus.PLAYING;

mIsKeyPressed = false;

mCurrentTimeMilliSecond = System.currentTimeMillis();

while (mGameStatus != Constant.GameStatus.END) {

if (mGameStatus == Constant.GameStatus.PAUSE) {

synchronized (mMonitorObject) {

try {

mMonitorObject.wait();

} catch (InterruptedException e) {

}

}

}

mProcessType = Constant.ProcessType.AUTO;

mDirection = Constant.Direction.DOWN;

while (true) {

if (mIsKeyPressed) {

mIsKeyPressed = false;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

break;

}

if (!mIsKeyPressed

&& System.currentTimeMillis()

- mCurrentTimeMilliSecond > getDownMilliSecond()) {

mProcessType = Constant.ProcessType.AUTO;

mDirection = Constant.Direction.DOWN;

mCurrentTimeMilliSecond = System

.currentTimeMillis();

break;

}

try {

Thread.sleep(1);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

process(mProcessType, mDirection);

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

}

}

}).start();

}

private void setEvents() {

addKeyListener(new KeyListener() {

@Override

public void keyTyped(KeyEvent e) {

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == Constant.KeyCode.UP) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.DIRECTION;

mDirection = Constant.Direction.UP;

} else if (e.getKeyCode() == Constant.KeyCode.DOWN) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.DIRECTION;

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

mDirection = Constant.Direction.DOWN;

mCurrentTimeMilliSecond = System.currentTimeMillis();

} else if (e.getKeyCode() == Constant.KeyCode.LEFT) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.DIRECTION;

mDirection = Constant.Direction.LEFT;

} else if (e.getKeyCode() == Constant.KeyCode.RIGHT) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.DIRECTION;

mDirection = Constant.Direction.RIGHT;

} else if (e.getKeyCode() == Constant.KeyCode.SPACE\_BAR) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.DIRECT\_DOWN;

mCurrentTimeMilliSecond = System.currentTimeMillis();

} else if (e.getKeyCode() == Constant.KeyCode.ESC) {

mIsKeyPressed = true;

mProcessType = Constant.ProcessType.AUTO;

mCurrentTimeMilliSecond = System.currentTimeMillis();

pause();

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

});

addWindowListener(new WindowAdapter() {

@Override

public void windowClosing(WindowEvent e) {

end();

}

});

}

public long getDownMilliSecond() {

return mTetrisManager.getDownMilliSecond();

}

}

|  |  |  |
| --- | --- | --- |
| **IPP**  **SW Programing** | 완료 보고서 | |
| 프로젝트 명 | JAVA Swing을 이용한 TETRIS |
| 팀 명 | 타조 |
| 날짜 | 2022-06-08 |

**Main Class**

package main;

import view.MainMenuPopup;

public class Main {

public static void main(String[] args) {

new MainMenuPopup().setVisible(true);

}

}