

# Term project

## 컴퓨터의 개념 및 실습

박희웅

[hee188@snu.ac.kr](mailto:hee188@snu.ac.kr)

서울대학교 산업공학과

# 테트리스 인공지능

- 프로젝트 목표

- 현재 상황을 인지하고 판단하여 적절한 결정을 내리는 인공지능을 테트리스 게임 내에서 구현해본다

- 게임 룰

- 일반 테트리스와 대부분 같은 대신, del 키를 이용해 내려오는 피스를 없앨 수 있다
- 1줄 10점, 2줄 30점, 3줄 50점, 4줄 70점
- 피스 하나 delete 할 때마다 -1점
- 시간 제한 3분
  - $\text{Board.TIMELIMIT} = 180 * 1000$
- 피스 제한 300개
  - $\text{Board.NUMPIECELIMIT} = 300$



# 게임 로직

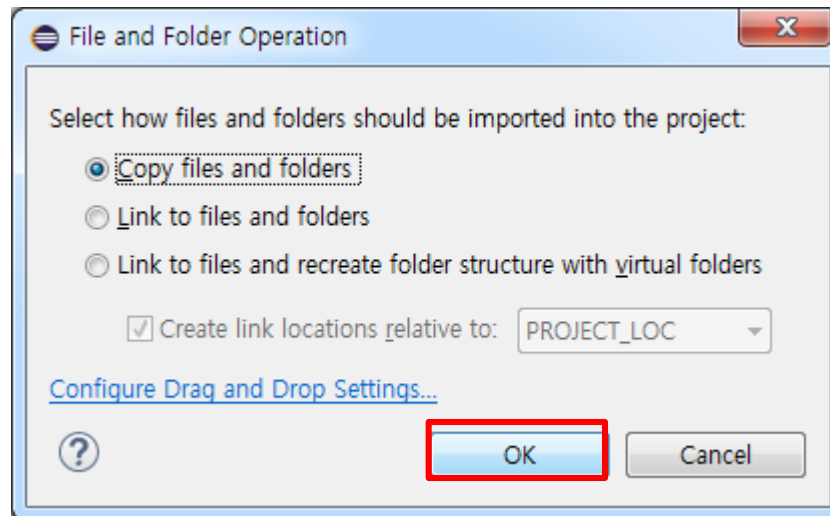
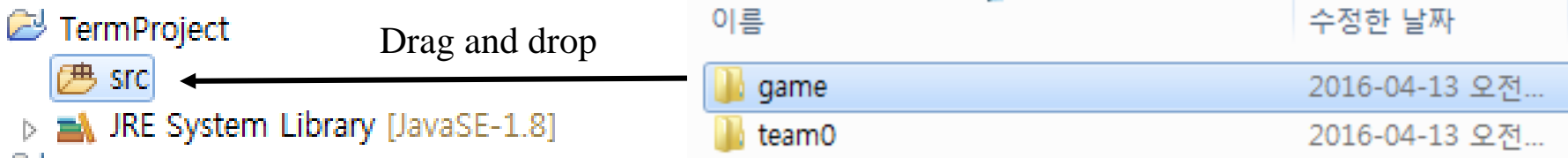
- 게임 속도
  - 400ms마다 한칸 내려오거나 새로운 피스 생성
  - 조작키 한번 누를 때 인공지능 스레드에 100ms 딜레이 발생
- 조작키 (Board.ControlKeys)
  - LEFT: 왼쪽으로 한칸
  - RIGHT: 오른쪽으로 한칸
  - UP: 시계반대방향으로 회전
  - DOWN: 시계방향으로 회전
  - D: 피스 한줄 내리기
  - SPACE: 피스 땅으로 내리기
  - DEL: 내려오는 피스 delete

```
public enum ControlKeys {  
  
    LEFT(KeyEvent.VK_LEFT),  
    RIGHT(KeyEvent.VK_RIGHT),  
    UP(KeyEvent.VK_UP),  
    DOWN(KeyEvent.VK_DOWN),  
    D(KeyEvent.VK_D),  
    SPACE(KeyEvent.VK_SPACE),  
    DEL(KeyEvent.VK_DELETE);  
  
    int keyCode;  
  
    ControlKeys(int keyCode) {  
        this.keyCode = keyCode;  
    }  
  
    int getKeyCode() {  
        return keyCode;  
    }  
};
```



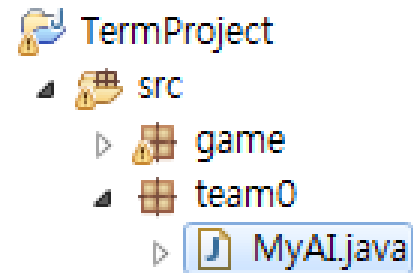
# 코드 작성

- game package
  - 주어지는 부분으로 게임 로직 담당. 코드 제출 범위 아님
  - 이클립스 프로젝트 내에 패키지를 생성하고 주어진 소스코드를 복사. Drag and drop 으로도 가능



# 코드 작성

- team package
  - 인공지능을 구현하는 부분으로 제출해야 하는 범위
  - 팀 이름을 갖는 패키지 생성하고 MyAI.java에 MyAI class 작성



- MyAI class
  - 현재 테트리스 게임 상황을 판단하여 조작 명령을 내리는 게임 인공지능을 구현하는 부분
  - game.AI class 상속하여 작성
  - protected ControlKeys command()
    - 오버라이드하여 현재 상황에 적절한 키 커맨드 리턴
  - public MyAI()
    - 기본 생성자는 반드시 public으로 작성



# 코드 작성

- AI class로부터 상속 받는 메서드

```
protected int getCurX() { return board.curX; }
protected int getCurY() { return board.curY; }
protected Shape getCurPiece() { return board.curPiece; }
protected Tetrominoes shapeAt(int x, int y) { return board.shapeAt(x, y); }

protected long getStartTime() { return board.startTime; }
protected int getnDropped() { return board.nDropped; }
protected int getScore() { return board.score; }

// need to be overridden
protected ControlKeys command() {
    ControlKeys[] values = ControlKeys.values();
    return values[(int)(Math.random() * values.length)];
}
```

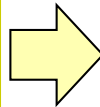


# 테스트

- 작성한 MyAI 테스트

- Board class의 board 생성자 내에서 AI 객체가 아닌 MyAI 객체 생성 (import 구문 필요)

```
public class Board extends JPanel
implements ActionListener {
    ...
    public Board(GameFrame game) {
        ...
        ai = new AI();
        ai.setBoard(this);
    }
    ...
}
```



```
import team0.MyAI;

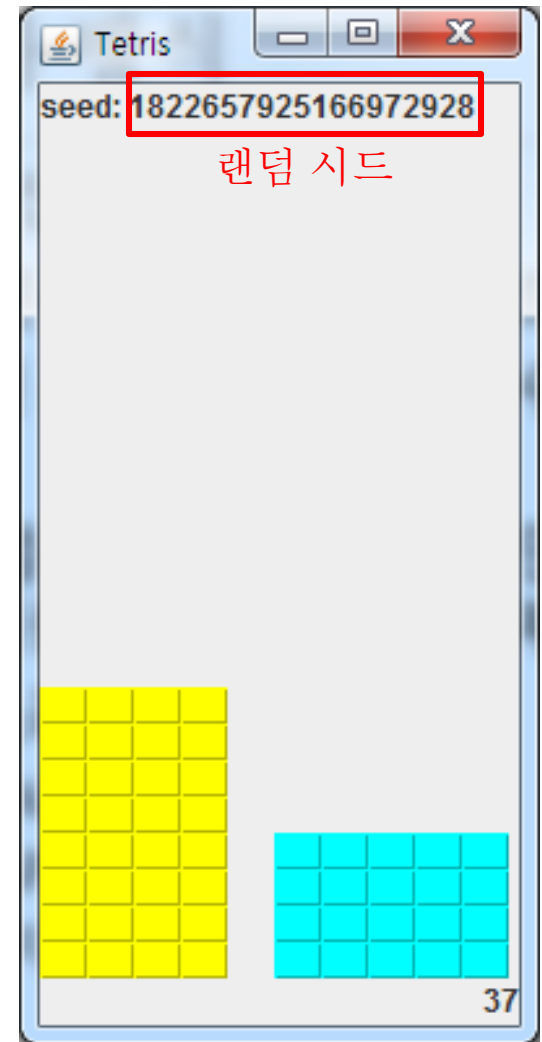
public class Board extends JPanel
implements ActionListener {
    ...
    public Board(GameFrame game) {
        ...
        ai = new MyAI();
        ai.setBoard(this);
    }
    ...
}
```



# 성능 평가 방식

- 점수 합산

- 각 팀은 자신있는 랜덤시드 번호 제출
  - Board.seed
- 제출된 시드들로 게임을 수행하여 평균 점수를 성능 점수로 이용
- 평균 점수가 베이스라인 인공지능의 평균 점수 이하인 경우, 베이스라인 인공지능의 평균 점수로 부여
- 베이스라인 인공지능
  - SQUARE를 왼쪽 4줄로
  - ISHAPE를 오른쪽 6줄로
  - 나머지는 모두 delete





# 팀 프로젝트 진행

- 진행 방식
  - 전반부에는 개인 별로 아이디어 도출 및 컨셉 실험
  - 후반부에 최대 3인으로 팀 구성하여 최종 결과물 산출
- 타임라인
  - 4월 14일: 팀 프로젝트 공지
  - 5월 10일 0시: etl에 개인 보고서(2장) 제출
  - 5월 10일 수업: 개인 보고서 하드카피 제출
  - 5월 12일: 팀 편성 공지. 팀 과제 제출물과 형식 공지
  - 6월 7일 0시: etl에 팀 별 결과물(보고서 포함) 제출
  - 6월 7일, 9일 수업: 팀 별 발표



# 성적 평가

- 과목 성적 평가의 30% 부분을 다음 항목으로 채점

- 성능 평가 15%

- 절대 점수 7.5%

$$\frac{s_j}{\max_i(s_i)} \times 7.5, s_j: \text{팀 } j \text{의 성능 점수}$$

- 순위 점수 7.5%

$$\sum_{i \neq j} (I(s_i < s_j) + \frac{I(s_i = s_j)}{2}) \times \frac{7.5}{n-1}, n: \text{팀 수}$$

- 상호 평가 10%

- 절대 점수 5%

$$\frac{v_j}{\max_i(v_i)} \times 5.0, v_j: \text{팀 } j \text{의 상호 평가 점수}$$

- 순위 점수 5%

$$\sum_{i \neq j} (I(v_i < v_j) + \frac{I(v_i = v_j)}{2}) \times \frac{5.0}{n-1}$$

- 보고서 평가 5%

- 개인보고서를 더 높은 비율로 반영

