C++언어 기말 프로젝트

2017011 이준영

### 문제정의

# 오셀로 게임을 만들고 플레이 할 수 있게 하라.

* 이때, 오셀로 보드의 크기 N은 6~12사이의 동적으로 할당할 수 있게 한다.

# 중간 프로젝트 당시 사용하였던 코드를 확장하여 코드를 작성하라.

(해결방법 가. 장에 나열함. )

# 주어진 인터페이스를 사용하여 객체 지향적인 오셀로 코드를 작성하라.

(해결방법 나. 장에 나열함.)

1. BoardDraw : 말판 기본화면 구성클래스, 좌표에 해당하는 말 표현

2. UserPlay: 사용자가 좌표에 말을 두는 것을 처리, 사용자 입력 처리

3. ComPlay: 컴퓨터가 놓을 수 있는 자리 중 가중치가 가장 높은 자리 찾음

4. BoardCheck: 현재 보드의 점수를 계산

# 오셀로 게임의 기본 정의는 아래와 같다.

* 선 후공을 정한다.
* 일단, 기본 말을 배치한다. 동적으로 크기를 제공하게 되어 있으므로, 보드의 중간을 찾기 위해 보드 길이의 1/2 지점과 그 지점의 +1 지점에서 초기 돌을 배치한다. (4개 배치)
* 배치 이후, 다음과 같은 사이클을 진행한다.
* 1. 선공은 적군 말 앞에 배치한다. 이때, 배치했을 때 최소 1개 이상의 적군 말을 역전시킬 수 있어야 한다.
* 2. 직선, 각 대각선 방향으로 적군 말이 있는지 체크한다.
* 3. 만약, 적군말이 있다면, 그 방향으로 더 나아가서 아군말이 있는지 확인한다.
* 4. 만약 아군말이 그 방향으로 확인된다면, 그 방향으로 둘러싸인 말들을 모두 아군 말로 역전한다. // 그렇지 않다면, 후공으로 넘긴다.
* 5. 체크가 끝나면, 후공으로 넘긴다.
* 6. 후공도 이상의 과정을 반복한다.
* 7. 그런데 이때, 둘 중 한 명이 배치 할 수 없는 상황 (어디든 배치하여도 적군 말을 역전시킬 수 없을 때 등)이 온다면, 배치할 수 없는 측을 패스한다.
* 8. 그런데, 둘 다 둘 수 없다면, 게임을 끝낸다.
* 9. 만약, 보드가 다 차면, 게임을 끝낸다.
* 게임이 끝나면, 누가 가장 많은 말을 두고 있는지에 따라서 승부를 결정한다.

### 해결 방법

1. 중간 프로젝트와의 연관성 명시

|  |  |  |
| --- | --- | --- |
| 중간 | 기말 | 차이점 |
| Board :  Void print  setInput | Board :  PiecesCode\*\* getBoardInfo  setInput | * Print -> getBoardInfo :   중간 프로젝트 당시, 보드를 보여주는 model class인 board에 view역할인 print에서 역할을 추출하여 개선   * setInput은 최대한 유지함:   단, 두 게임의 규정 중 오셀로측이 역전 현상이 존재함을 감안하여 코드가 변경됨. 인터페이스는 동일. |
| Ware :  Bool computer | Pieces :  Bool computer | * Ware-> Pieces : 기물이라는 이름을 최대한 오셀로에 맞게 수정함. * 다만 의미론적으로 배치된 기물이 computer인지 아닌지를 확인해주는 클래스이므로, computer라는 변수는 수정하지 않음. * Board->Ware\*\* // Board-> Pieces\*\* 의 동일한 구조를 통하여 확장하였음. |

1. 주어진 인터페이스의 명시
2. 상세 객체의 설명과 구현 동작의 설명
3. `main`함수의 동작 구현
4. <반복 LOOP>
5. 컴퓨터의 선공인지 확인한다.
6. UserPlay를 생성(userPlay)하여 게임으을 초기화한다.
7. 게임이 끝나지 않았다면, (userPlay의 getEnded()가 false인 상태)

<반복 LOOP>

1. userPlay를 통하여 턴을 진행한다.

<반복종료 END LOOP>

1. userPlay에서 추출한 게임 결과를 가져온다.
2. 게임 결과를 보여준다.
3. 게임의 승자를 판단한다.
4. 게임의 승자는 게임 결과의 컴퓨터 점수와 유저 점수를 통하여 판단한다.
5. 보드를 보여준다.
6. 이후 게임을 삭제하고, 게임을 다시 시작할 지 묻는다.
7. 만약, n 을 입력한다면 break를 통해 반복을 나간다.

<반복종료 END LOOP>

1. 1cycle의 구현
2. 별첨

* (별첨 1)기물 역전의 판단 범위

이하의 그림과 같이, 배치 이후 대각선/직선으로 판단한다.

다만, 아래의 그림은 방향만 보여주는 것이며, x,y는 보드가 허락하는 한 I,j방향으로 증가한다고 가정한다.

텍스트, 사각형, 스크린샷, 직사각형이(가) 표시된 사진

자동 생성된 설명

* (별첨 2)기물 역전의 판단 알고리즘

기물역전은 (별첨 1)의 방향으로 아래 그림과 같은 알고리즘으로 판단한다.

텍스트, 라인, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

### 소스코드

### 실행결과