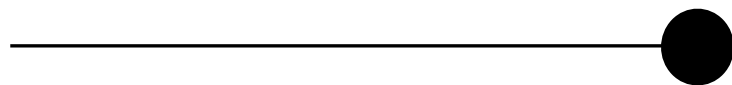


Java Mini Project

2차원 배열 - 오목게임 프로그램

목 차

001 요구사항분석

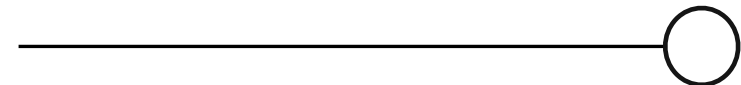


01 프로젝트 주제 설명

02 요구사항분석

03 기술요구사항

002 설계 및 작업



01 코드 작성

02 코드 실행테스트

001 요구사항분석

01 프로젝트 주제 설명

02 요구사항분석

03 기술요구사항

주제 I

오목 게임 프로그램

선정 이유 및 내용 I

행과 열을 각각 선택하여 바둑돌을 놓을 수 있는 규칙을
2차원 배열에 접목시켜 활용해 볼 수 있기 때문에 오목 게임을 선정하였다.

행과 열을 선택하여 두 명의 플레이어가 각각 흑돌과 백돌을 번갈아 가며 두다가,
가로, 세로, 대각선 상관 없이 먼저 오목을 두는 자가 이기는 게임이다.

프로젝트 요구사항 I

1. 오목판 배열을 생성

- (a) 기본 크기는 19x19 배열이나 크기 변경 가능하게 설정

2. 플레이어 설정 및 흑돌 둘 순서 결정

- (a) 2인용 게임으로 설정
- (b) 플레이어 이름은 각각 중복되지 않게 설정

3. 돌 배치 규칙

- (a) 본인 차례에 (x, y) 좌표를 입력하면 돌을 놓을 수 있도록 설정
- (b) 이미 올린 돌 위에 올리지 못하도록 설정

4. 승패 결정

- (a) 가로, 세로, 대각선 중 먼저 오목을 채우는 사람이 승리(게임 끝)
- (b) 육목 이상일 경우에는 오목으로 인정하지 않음

항목명	분류	기술 명	설명
게임 초기 설정	오목판	크기	가로 * 세로 = 기본 19 * 19 (행과 열 이름 제외) But, 사이즈를 변수로 지정해 <u>유동적으로 변경 가능하도록</u>
		생성	<u>좌표를 설정할 수 있도록</u> 오목판 바깥에 알파벳 대문자와 소문자로 행, 열 구분 <u>돌을 놓지 않은 빈 공간은 ". "</u> 으로 표시
	플레이어	이름	이름의 길이는 <u>최소 한 글자 이상</u> (공백 제외) 플레이어 1의 이름과 플레이어 2의 이름이 같으면 안 됨(<u>중복 금지</u>)
		중복 체크	플레이어 2의 이름을 입력 받을 때, 플레이어 1의 이름과 <u>동일하지 않도록</u> 설정
		순서 정하기	각 플레이어 별로 숫자를 랜덤 추출하여 <u>더 큰 숫자인 사람이 흑돌(선공)</u>

항목명	분류	기술 명	설명
게임 진행	돌 배치	배치	콘솔에 <u>행의 위치(알파벳 대문자)</u> 와 <u>열의 위치(알파벳 소문자)</u> 를 입력 Ex) Aa
		오류	<u>이미 배치한 돌 위에 다시 배치할 경우</u> , 배치 오류 메시지 출력 및 재입력
	승패 판단	승리 조건	가로, 세로, 대각선 중 <u>먼저 오목을 두는 자가 승리</u> 단, <u>육목 이상일 경우에는 오목으로 인정하지 않음</u> (승리 x)
		가로	<u>한 행에 같은 색의 돌이 5개가 놓여 있을 경우</u> 승리 판정
		세로	<u>한 열에 같은 색의 돌이 5개가 놓여 있을 경우</u> 승리 판정
		대각선	각 대각선 방향 별로(ex. 오른쪽 위 → 왼쪽 아래 대각선) <u>행과 열을 비교하여,</u> <u>같은 색의 돌이 5개가 놓여 있을 경우</u> 승리 판정
		안내 메시지	승리했을 경우 <u>흑돌인지 백돌인지 구분 후</u> , 플레이어의 이름과 함께 축하 메시지 출력

002 설계 및 작업

01 코드 작성

02 코드 실행 테스트

코드설명

오목판 배열 생성 I

1. 오목판 한 변의 크기를 저장할 변수 size 선언
2. 오목판 배열을 저장할 2차원 배열 field 선언. 오목판의 크기는 size변수에 대응 됨
3. 오목판의 빈 공간을 표현할 변수 space 선언

```
int size = 20; // 바둑판 가로세로(19*19)
```

```
String[][] field = new String[size][size];
```

```
String space = ". "; // 바둑판 빈 공간
```

코드설명

오목판 배열 생성 I

4. 오목판 배열에 빈 공간을 표현할 변수 space을 대입

```
for (int i = 0; i < size; i++) {  
    for (int j = 0; j < size; j++) {  
        field[i][j] = space;  
    }  
}
```

코드설명

오목판 배열 생성 I

5. 오목판 배열에 행 번호(알파벳 대문자), 열 번호(알파벳 소문자) 대입

```
for (int i = 0; i < size - 1; i++) {  
    field[i + 1][0] = (char) ('A' + i) + " ";  
}  
for (int i = 0; i < size - 1; i++) {  
    field[0][i + 1] = (char) ('a' + i) + " ";  
}  
field[0][0] = " ";
```

코드설명

오목판 배열 생성 I

6. 콘솔창에 오목판 배열 출력

```
// 출력부
for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) {
        System.out.print(field[i][j]);
    }
    System.out.println();
}
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 1) I

1. 플레이어 1의 이름을 입력 받아 저장하는 변수 stName 선언
2. 입력 받은 값의 공백의 개수를 저장하는 변수 spCnt 선언

```
String stName = ""; // 플레이어 1의 이름 입력받는 변수 stName 선언
```

```
int spCnt = 0;
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 1) I

3. 플레이어 1의 이름을 입력하지 않거나, 플레이어 1의 이름에 공백만 입력한 경우까지 반복하도록 조건 및 증감식 작성
4. 입력 받은 값을 변수 stName에 저장

```
for (; stName.length() == 0 || (spCnt == stName.length());) {  
    spCnt = 0;  
    System.out.print("플레이어 1의 이름을 입력해주세요. >> ");  
    stName = in.nextLine();  
}
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 1) I

- 5. 변수 stName의 값의 모든 인덱스 번호를 반환하도록 조건 및 증감식 작성
- 6. 변수 stName의 값에 있는 공백의 개수 판단

```
for (int i = 0; i < stName.length(); i++) {  
    if ((int) (stName.charAt(i)) == 32) {  
        spCnt++;  
    }  
}
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 1) I

7. 조건에 맞지 않을 경우 재입력

```
if (stName.length() == 0) {  
    System.out.println("이름은 최소 1글자 이상이어야 합니다.");  
} else if (spCnt == stName.length()) {  
    System.out.println("이름을 공백으로 설정 할 수 없습니다.");  
}
```


코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 2) I

1. 플레이어 2의 이름을 입력 받아 저장하는 변수 ndName 선언

```
String ndName = "";
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 2) I

2. 플레이어 2의 이름을 입력하지 않거나 플레이어 2의 이름에 공백만 입력하거나 두 플레이어의 이름이 중복된 경우까지 반복하도록 조건 및 증감식 작성
3. 입력 받은 값을 변수 ndName에 저장

```
for ( ; ndName.length() == 0 || ( spCnt == ndName.length() ) ||  
ndName.equals(stName);  
    spCnt = 0;  
    System.out.print("플레이어 2의 이름을 입력해주세요. >> ");  
    ndName = in.nextLine();
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 2) I

- 4. 변수 ndName의 값의 모든 인덱스 번호를 반환하도록 조건 및 증감식 작성
- 5. 변수 ndName의 값에 있는 공백의 개수 판단

```
for (int i = 0; i < ndName.length(); i++) {  
    if ((int) (ndName.charAt(i)) == 32) {  
        spCnt++;  
    }  
}
```

코드설명

2명의 플레이어 이름 각각 입력 받기 (플레이어 2) I

6. 조건에 맞지 않은 입력 값을 받을 경우 조건을 명시하는 문자열 출력

```
if (ndName.length() == 0) {  
    System.out.println("이름은 최소 1글자 이상이어야 합니다.");  
} else if ( spCnt == ndName.length() ) {  
    System.out.println("이름을 공백으로 설정 할 수 없습니다.");  
} else if (ndName.equals(stName)) {  
    System.out.println("플레이어 1과 이름이 동일합니다. 다시 입력해주세요.");  
}
```

코드설명

선공 정하기 (흑돌 정하기) I

1. 흑돌과 백돌에 대한 플레이어 이름 저장하는 변수 blName, wtName을 선언함

```
String blName;  
String wtName;  
System.out.println("흑돌로 시작할 사람을 정하겠습니다.");  
System.out.println("더 큰 수를 뽑은 사람이 흑돌입니다.");
```

코드설명

선공 정하기 (흑돌 정하기) I

2. 무한 반복이 발생하도록 조건 및 증감식 작성
3. 각 플레이어에 대한 난수를 저장하는 변수 stNum, ndNum을 선언하고, 0~9의 난수를 반환하여 대입

```
for ( ;; ) {  
    int stNum = r.nextInt(10);  
    int ndNum = r.nextInt(10);
```

코드설명

선공 정하기 (흑돌 정하기) I

4. 더 큰 수가 반환된 변수에 따라 흑돌을 잡을 플레이어를 결정
5. stNum의 값이 ndNum보다 작은 경우 ndName의 값을 blName에 대입하고
플레이어 2가 흑돌임을 알려주는 문자열 출력

```
if (stNum > ndNum) {
    System.out.println(stName + " 님이 뽑으신 숫자는 " + stNum + "입니다.");
    System.out.println(ndName + " 님이 뽑으신 숫자는 " + ndNum + "입니다.");
    System.out.println("");
    System.out.println(stName + " 님이 흑돌입니다.");
    blName = stName; wtName = ndName;
    System.out.println("----- 게임 시작 -----");
    break;
} else if (stNum < ndNum) {
    System.out.println(stName + " 님이 뽑으신 숫자는 " + stNum + "입니다.");
    System.out.println(ndName + " 님이 뽑으신 숫자는 " + ndNum + "입니다.");
    System.out.println("");
    System.out.println(ndName + " 님이 흑돌입니다.");
    blName = ndName;
    wtName = stName; System.out.println("");
    System.out.println("----- 게임 시작 -----");
    break;
}
```

코드설명

돌 배치하기 I

1. 게임을 진행하는 동안 반복될 for문 사용
2. 순서를 확인하기 위한 변수 i선언
3. 흑돌을 표현할 변수 black 선언
4. 백돌을 표현할 변수 white 선언
5. 현재 차례의 돌을 저장할 변수 stone 선언
6. 좌표값을 입력 받을 변수 userIn 선언
7. 입력 받은 행 번호를 저장할 변수 rowNum 선언
8. 입력 받은 열 번호를 저장할 변수 colNum 선언

코드설명

```
for (int i = 0;; i++) {  
    String userIn;  
  
    String black = "● ";  
  
    String white = "○ ";  
  
    char stone;  
  
    ...  
  
    int rowNum = userIn.charAt(0) - 'A' + 1;  
  
    int colNum = userIn.charAt(1) - 'a' + 1;
```

코드설명

돌 배치하기 I

9. 콘솔에 현재 차례인 사용자의 이름과 안내 메시지 출력

```
if (i % 2 == 0) {  
    System.out.println(blName + " 님 흑돌의 위치를 정하세요. 예시: Aa");  
    stone = black.charAt(0);  
} else {  
    System.out.println(wtName + " 님 백돌의 위치를 정하세요. 예시: Aa");  
    stone = white.charAt(0);  
}  
System.out.println(">>");
```

코드설명

돌 배치하기 I

10. 바둑알을 놓을 좌표를 입력 받고, 유효한 좌표를 입력 받았는지 검사
- (a) 두문자 이상의 문자를 입력 받았는지 검사
 - (b) 입력 받은 좌표가 오목판 내에 포함되는지 검사
 - (c) 입력 받은 좌표가 빈 공간인지 검사

```
for (userIn = in.nextLine(); userIn.length() <= 1;) {  
  
    userIn = in.nextLine(); // 길이 안맞으면 무한 반복  
  
}
```

코드설명

돌 배치하기 I

10. 바둑알을 놓을 좌표를 입력 받고, 유효한 좌표를 입력 받았는지 검사
- (a) 두문자 이상의 문자를 입력 받았는지 검사
 - (b) 입력 받은 좌표가 오목판 내에 포함되는지 검사
 - (c) 입력 받은 좌표가 빈 공간인지 검사

```
if (rowNum >= field.length || rowNum < 0 || colNum >= field[0].length || colNum < 0) {  
    // 범위 벗어난 인덱스  
  
        JOptionPane.showMessageDialog(null, "정확한 좌표를 입력하세요.",  
                                        "ERROR_MESSAGE",  
JOptionPane.ERROR_MESSAGE);  
  
        i--;  
  
        continue;  
  
}
```

코드설명

돌 배치하기 I

10. 바둑알을 놓을 좌표를 입력 받고, 유효한 좌표를 입력 받았는지 검사
- (a) 두문자 이상의 문자를 입력 받았는지 검사
 - (b) 입력 받은 좌표가 오목판 내에 포함되는지 검사
 - (c) 입력 받은 좌표가 빈 공간인지 검사

```
if (field[rowNum][colNum] != space) { // 이미 돌 있음
    System.out.println("해당 위치에 이미 돌이 있습니다.");
    JOptionPane.showMessageDialog(null, "해당 위치에 이미 돌이 있습니다.",
        "ERROR_MESSAGE",JOptionPane.ERROR_MESSAGE);

    i--;
    continue;
}
```

코드설명

돌 배치하기 I

11. 입력 받은 좌표에 해당하는 오목판 배열에 현재 차례의 바둑알 대입

```
field[rowNum][colNum] = stone + " ";

// 출력부
for (int j = 0; j < size; j++) {
    for (int k = 0; k < size; k++) {
        System.out.print(field[j][k]);
    }
    System.out.println();
}
```

코드설명

승패 결정 I

1. 돌이 연속적으로 있는 길이를 저장할 변수 cnt 선언.
2. 오목을 판단하는 자료형 boolean의 변수 fiveGo 선언. → true일 경우 오목으로 판단.

```
int cnt = 0;  
boolean fiveGo = false;
```

코드설명

승패 결정 – 가로가 오목일 때 I

1. 돌을 새롭게 놓은 행에서 모든 열의 인덱스 번호를 반환하도록 반복문의 조건 및 증감식 작성.
2. 반복문 종료 후 변수 cnt 초기화.

```
for (int ii = 1; ii < size; ii++) {  
    ...  
}  
cnt = 0;
```


코드설명

승패 결정 – 가로가 오목일 때 I

3. 해당 행에서 stone값을 찾을 경우 cnt를 1 증가 대입.
 - a. 변수 cnt 값이 변수 target 값(5)과 같고,
변수 ii 값이 행의 마지막 열의 인덱스 번호일 경우 오목으로 판단.
4. 해당 행에서 stone이 아닌 값을 찾고 그때 변수 cnt 값이 변수 target 값과 같은 경우 오목으로 판단.
5. 이 외의 경우에는 변수 cnt 값 초기화

코드설명

```
if (field[rowNum][ii].charAt(0) == stone) {  
    cnt++;  
    if (cnt == target && ii == size - 1) {  
        fiveGo = true;  
        break;  
    }  
} else if (cnt == target) {  
    fiveGo = true;  
    break;  
  
} else {  
    cnt = 0;  
}
```

코드설명

승패 결정 – 세로가 오목일 때 I

1. 오목이 달성되지 않은 경우에만 다른 기준으로 판단하도록 if문 조건 작성
2. if문 종료 후 변수 cnt 초기화.

```
if (fiveGo == false) {  
  ...  
}  
cnt = 0;
```

코드설명

승패 결정 – 세로가 오목일 때 I

3. 돌을 새롭게 놓은 열에서 모든 행의 인덱스 번호를 반환하도록 반복문의 조건 및 증감식 작성.

```
for (int ii = 1; ii < size; ii++) {  
    ...  
}
```

코드설명

승패 결정 – 세로가 오목일 때 I

4. 해당 열에서 stone 값을 찾은 경우 변수 cnt 값 1 증가 대입.
 - a. 변수 cnt 값이 변수 target 값과 같고, 변수 i의 값이 해당 열의 마지막 행 번호일 경우 오목으로 판단.
5. 해당 열에서 stone이 아닌 값을 찾고 변수 cnt 값이 변수 target 값과 같은 경우 오목으로 판단.
6. 이 외의 경우에는 변수 cnt 값을 초기화.

코드설명

```
if (field[ii][colNum].charAt(0) == stone) {  
    cnt++;  
    if (cnt == target && ii == size - 1) {  
        fiveGo = true;  
        break;  
    }  
} else if (cnt == target) {  
    fiveGo = true;  
    break;  
} else {  
    cnt = 0;  
}
```

코드설명

승패 결정 – 대각선(우상단 -> 좌하단) I

1. 하나의 대각선(우상단 -> 좌하단)의 좌표들은 행 번호와 열 번호의 합이 동일함
2. 행 번호와 열 번호의 합을 저장할 변수 kk 선언
3. 오목판 배열을 순회하며 좌표와 kk를 비교하기 위해 2중 for문 사용

```
int kk = rowNum + colNum;
for (int ii = 1; ii < size; ii++) {
    for (int jj = 1; jj < size; jj++) {
        if (ii + jj == kk) {
            ...
        }
    }
}
```

코드설명

승패 결정 - 대각선(우상단 -> 좌하단) I

4. 대각선을 순회하며 오목판 배열값을 확인
5. 직전에 착수한 돌이 오목을 이루었는지 검사
6. 오목을 이루었으면 승리 처리

```
if (field[ii][jj].charAt(0) == stone) {  
    cnt++;  
    if (cnt == target && (ii == size - 1 || jj == 1)) {  
        fiveGo = true;  
        break;  
    }  
} else if (cnt == target) {  
    fiveGo = true;  
    break;  
} else {  
    cnt = 0;  
}
```


코드설명

승패 결정 – 대각선(좌상단 -> 우하단) I

1. 오목이 달성되지 않은 경우에만 다른 기준으로 판단하도록 if문 선언
 - (a) 시작열을 저장할 변수 z 선언
 - (b) 시작행을 저장할 변수 jj 선언
2. 행 번호가 열 번호보다 클 경우를 판단하도록 조건식 작성
 - (a) 변수 jj에 행번호에서 열번호를 제외한 값만큼 순회할 값 대입
 - (b) 변수 z에 1 대입
3. 행 번호가 열 번호보다 작을 경우를 판단하도록 조건식 작성
 - (a) 변수 z에 열번호에서 행번호를 제외한 값만큼 순회할 값 대입
 - (b) 변수 jj에 1 대입

코드설명

```
if (fiveGo == false) {  
    int z; // 시작 열  
    int jj; // 시작 행  
    if (rowNum > colNum) {  
        jj = rowNum - colNum + 1;  
        z = 1;  
    } else {  
        z = colNum - rowNum + 1;  
        jj = 1;  
    }  
}
```

코드설명

승패 결정 - 대각선(좌상단 -> 우하단) I

4. 직전에 착수한 돌이 오목을 이루었는지 검사

5. 오목을 이루었으면 승리 처리

```
if (jj <= size - target && z <= size - target) {
    for (; z < size && jj < size;) {
        if (field[jj][z].charAt(0) == stone) {
            cnt++;
            if (cnt == target && (jj == size - 1 || z == size - 1))
                fiveGo = true;
            break;
        }
        } else if (cnt == target) {
            fiveGo = true;
            break;
        } else {
            cnt = 0;
        }
        z++;
        jj++;
    }
}
```

코드설명

오목을 달성하였을 경우 I

1. 오목이 달성되었을 경우에만 오목 달성 안내 메시지를 출력하도록 if문 조건 작성
(a) 달성하였을 경우 반복문 종료

```
if (fiveGo == true) {  
  ...  
      break;  
}
```

코드설명

오목을 달성하였을 경우 I

2. 흑돌이 오목을 달성한 경우를 판별하기 위한 if문 조건 작성
 - (a) 흑돌에 대한 플레이어 이름을 저장한 변수 blName과 승리 안내 메시지를 콘솔창과 팝업창으로 출력
3. 백돌이 오목을 달성한 경우를 판별하기 위한 if문 조건 작성
 - (a) 백돌에 대한 플레이어 이름을 저장한 변수 wtName과 승리 안내 메시지를 콘솔창과 팝업창으로 출력

```
System.out.println("★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★");  
if (i % 2 == 0) {  
    System.out.println(blName + " 승리.");  
    JOptionPane.showMessageDialog(null, blName + " 님 승리!!!", "Game Over",  
        JOptionPane.INFORMATION_MESSAGE);  
} else {  
    System.out.println(wtName + " 승리.");  
    JOptionPane.showMessageDialog(null, wtName + " 님 승리!!!", "Game Over",  
        JOptionPane.INFORMATION_MESSAGE);  
}
```

코드설명

오목을 달성하였을 경우 I

4. 놓여져 있지 않은 모든 돌의 행과 열 각각의 인덱스 번호를 구하기 위한 이중 for문의 조건식 작성
5. 바둑판의 빈 공간과 4번에서 구한 바둑판 2차원 배열의 인덱스 번호의 값이 같을 경우를 구하는 if문의 조건 작성
 - (a) 같을 경우 빈 공간을 저장하는 변수 space에 “ ”(공백값) 대입
6. 게임을 종료하는 내용의 메시지를 콘솔창에 출력

```
for (int ii = 0; ii < size; ii++) {  
    for (int jj = 0; jj < size; jj++) {  
        if (field[ii][jj] == space) {  
            field[ii][jj] = " ";  
        }  
        System.out.print(field[ii][jj]);  
    }  
    System.out.println();  
}  
System.out.println("★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★");  
System.out.println("게임을 종료합니다.");
```

```

1 package j001;
2
3 import java.util.Random;
4
5 public class H001 {
6
7     public static void main(String[] args) {
8
9         int size = 20; // 바둑판 가로세로(19*19)
10         String[][] field = new String[size][size];
11         Scanner in = new Scanner(System.in);
12         Random r = new Random();
13         String space = " "; // 아무런 빈공간
14         JOptionPane na = new JOptionPane();
15         na.showMessageDialog(null, "보노 세외 시작");
16         // 기본 오목판 배열 생성
17         for (int i = 0; i < size; i++) {
18             for (int j = 0; j < size; j++) {
19                 field[i][j] = space;
20             }
21         }
22         for (int i = 0; i < size - 1; i++) {
23             field[i + 1][0] = (char) ('A' + i) + " ";
24         }
25         for (int i = 0; i < size - 1; i++) {
26             field[0][i + 1] = (char) ('B' + i) + " ";
27         }
28         field[0][0] = " ";
29
30         // 출력부
31         for (int i = 0; i < size; i++) {
32             for (int j = 0; j < size; j++) {
33                 System.out.print(field[i][j]);
34             }
35             System.out.println();
36         }
37
38         // 게임 플레이 성공 했는지 1 플레이어 했는지
39
40         String stName = ""; // 플레이어 1의 이름 입력받는 변수 stName 선언
41         int spCnt = 0;
42         for (; stName.length() == 0 || (spCnt == stName.length()); {
43
44             spCnt = 0;
45             System.out.print("플레이어 1의 이름을 입력해주세요. >> ");
46             stName = in.nextLine();
47             for (int i = 0; i < stName.length(); i++) {
48                 if ((int) (stName.charAt(i)) == 32) {
49                     spCnt++;
50                 }
51             }
52             if (stName.length() == 0) {
53                 System.out.println("이름은 최소 1글자 이상이어야 합니다.");
54             } else if (spCnt == stName.length()) {
55                 System.out.println("이름을 공백으로 설정 할 수 없습니다.");
56             }
57         }
58
59         String ndName = ""; // 플레이어 2의 이름 입력받는 변수 ndName 선언
60         // 플레이어 이름 중복 체크
61         for (; ndName.length() == 0 || (spCnt == ndName.length()) || ndName.equals(stName);) {
62             spCnt = 0;
63             System.out.print("플레이어 2의 이름을 입력해주세요. >> ");
64             ndName = in.nextLine();
65             // 플레이어 2의 이름을 입력받은 뒤, 플레이어 1의 이름과 중복된다면,
66             for (int i = 0; i < ndName.length(); i++) {
67                 if ((int) (ndName.charAt(i)) == 32) {
68                     spCnt++;
69                 }
70             }
71             if (ndName.length() == 0) {
72                 System.out.println("이름은 최소 1글자 이상이어야 합니다.");
73             } else if (spCnt == ndName.length()) {
74                 System.out.println("이름을 공백으로 설정 할 수 없습니다.");
75             } else if (ndName.equals(stName)) {
76                 System.out.println("플레이어 1과 이름이 중복됩니다. 다시 입력해주세요.");
77             } else {
78                 //
79             }
80         }
81
82         System.out.println();
83         String blName/* = null */; // 흑돌 플레이어 이름 저장할 변수
84         String wtName/* = null */; // 백돌 플레이어 이름 저장할 변수

```

```

87 System.out.println("복합로 처리할 사람을 선택했습니다.");
88 System.out.println("나눌 수를 줄일 수에 속합니다.");
89
90 for (/* for i = 0 */; /* stName != "" && ndName != "" */; ) {
91     /* if(stName != null && ndName != null) { */
92     int stNum = r.nextInt(10);
93     int ndNum = r.nextInt(10);
94     if (stNum > ndNum) {
95         System.out.println(stName + " 남은 번호의 숫자는 " + stNum + "입니다.");
96         System.out.println(ndName + " 남은 번호의 숫자는 " + ndNum + "입니다.");
97         System.out.println("");
98         System.out.println(stName + " 남은 속합니다.");
99         blName = stName;
100        wtName = ndName;
101        System.out.println("----- 게임 시작 -----");
102        break;
103    } else if (stNum < ndNum) {
104        System.out.println(stName + " 남은 번호의 숫자는 " + stNum + "입니다.");
105        System.out.println(ndName + " 남은 번호의 숫자는 " + ndNum + "입니다.");
106        System.out.println("");
107        System.out.println(ndName + " 남은 속합니다.");
108        blName = ndName;
109        wtName = stName;
110        System.out.println("");
111        System.out.println("----- 게임 시작 -----");
112        break;
113    } // else if (stNum == ndNum) {
114    //         i--;
115    //     }
116    // }
117 }
118
119 // 게임 플레이
120 for (int i = 0; i < 10; i++) {
121     String userIn;
122     String black = "X ";
123     String white = "O ";
124     char stone;
125     int target = 5;
126
127     if (i % 2 == 0) {
128         System.out.println(blName + " 남은 번호의 위치를 선택하세요. 예시: A5");
129         stone = black.charAt(0);
130     } else {
131         System.out.println(wtName + " 남은 번호의 위치를 선택하세요. 예시: A5");
132         stone = white.charAt(0);
133     }
134     System.out.println(">>");
135
136     userIn = r.nextLine(); userIn.length() <= 1;
137     userIn = r.nextLine(); // 입력 잘못하면 무한 반복
138
139     int rowNum = userIn.charAt(0) - 'A' + 1;
140     int colNum = userIn.charAt(1) - '1' + 1;
141     if (rowNum >= field.length || rowNum < 0 || colNum >= field[0].length || colNum < 0) { // 잘못 넣어서 입력
142         JOptionPane.showMessageDialog(null, "유효한 좌표를 입력하세요.", "ERROR MESSAGE", JOptionPane.ERROR_MESSAGE);
143         i--;
144         continue;
145     }
146     if (field[rowNum][colNum] != stone) { // 이미 넣은 칸
147         System.out.println("이미 넣은 칸입니다.");
148         JOptionPane.showMessageDialog(null, "이미 넣은 칸입니다.", "ERROR MESSAGE", JOptionPane.ERROR_MESSAGE);
149         i--;
150         continue;
151     }
152     field[rowNum][colNum] = stone + " ";
153
154     // 출력
155     for (int j = 0; j < size; j++) {
156         for (int k = 0; k < size; k++) {
157             System.out.print(field[j][k]);
158         }
159         System.out.println();
160     }
161
162     // 승리 조건
163     int cnt = 0;
164     boolean findGo = false;
165
166     // 1. 3개
167     for (int ii = 1; ii < size; ii++) {

```



```

169         if (field[rowNum][jj].charAt(0) == stone) {
170             cnt++;
171             if (cnt == target && ii == size - 1) {
172                 fiveGo = true;
173                 break;
174             }
175         } else if (cnt == target) {
176             fiveGo = true;
177             break;
178         } else {
179             cnt = 0;
180         }
181     }
182     cnt = 0;
183
184     // 세로
185     if (fiveGo == false) {
186         for (int ii = 1; ii < size; ii++) {
187             if (field[ii][colNum].charAt(0) == stone) {
188                 cnt++;
189                 if (cnt == target && ii == size - 1) {
190                     fiveGo = true;
191                     break;
192                 }
193             } else if (cnt == target) {
194                 fiveGo = true;
195                 break;
196             } else {
197                 cnt = 0;
198             }
199         }
200     }
201     cnt = 0;
202
203     // 오른쪽 위 -> 왼쪽 아래 대각선
204     if (fiveGo == false) {
205         int kk = rowNum + colNum;
206         for (int ii = 1; ii < size; ii++) {
207             for (int jj = 1; jj < size; jj++) {
208                 if (ii + jj == kk) {
209                     if (field[ii][jj].charAt(0) == stone) {
210                         cnt++;
211                         if (cnt == target && (ii == size - 1 || jj == size - 1)) {
212                             fiveGo = true;
213                             break;
214                         }
215                     } else if (cnt == target) {
216                         fiveGo = true;
217                         break;
218                     } else {
219                         cnt = 0;
220                     }
221                 }
222             }
223         }
224     }
225     cnt = 0;
226
227     // 왼쪽 위 -> 오른쪽 아래 대각선
228     if (fiveGo == false) {
229         int r; // 시작 행
230         int c; // 시작 열
231         if (rowNum > colNum) {
232             r = rowNum - colNum + 1;
233             c = 1;
234         } else {
235             r = colNum - rowNum + 1;
236             c = 1;
237         }
238         if (r <= size - target && c <= size - target) {
239             for (; c < size && r < size; c++) {
240                 if (field[r][c].charAt(0) == stone) {
241                     cnt++;
242                     if (cnt == target && (r == size - 1 || c == size - 1)) {
243                         fiveGo = true;
244                         break;
245                     }
246                 } else if (cnt == target) {
247                     fiveGo = true;
248                     break;
249                 } else {
250                     cnt = 0;
251                 }
252             }
253         }
254     }
255     cnt = 0;

```

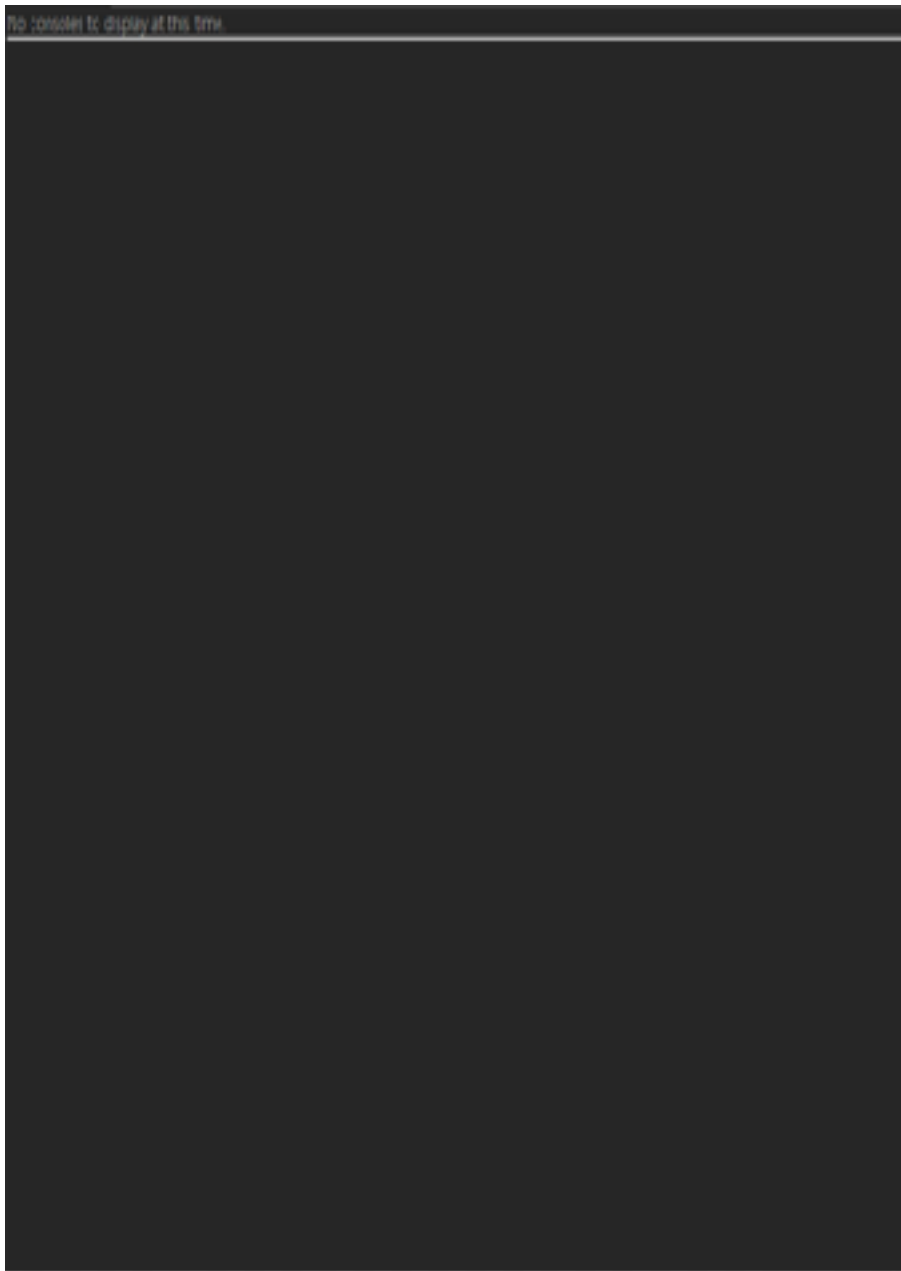
```

250         cnt = 0;
251     }
252     z++;
253     j111;
254 }
255 }
256 }
257
258 // 5가 연속 발견했을 때
259 if (fiveIn == true) {
260     System.out.println("*****");
261     if (i % 2 == 0) {
262
263         System.out.println(blName + " 승리.");
264         JOptionPane.showMessageDialog(null, blName + " 님 승리!!!.", "Game Over",
265             JOptionPane.INFORMATION_MESSAGE);
266     } else {
267         System.out.println(wtName + " 승리.");
268         JOptionPane.showMessageDialog(null, wtName + " 님 승리!!!.", "Game Over",
269             JOptionPane.INFORMATION_MESSAGE);
270     }
271     for (int ii = 0; ii < size; ii++) {
272         for (int jj = 0; jj < size; jj++) {
273             if (field[ii][jj] == space) {
274                 field[ii][jj] = " ";
275             }
276             System.out.print(field[ii][jj]);
277         }
278         System.out.println();
279     }
280     System.out.println("*****");
281     System.out.println("게임을 종료합니다.");
282
283     break;
284 }
285 }
286

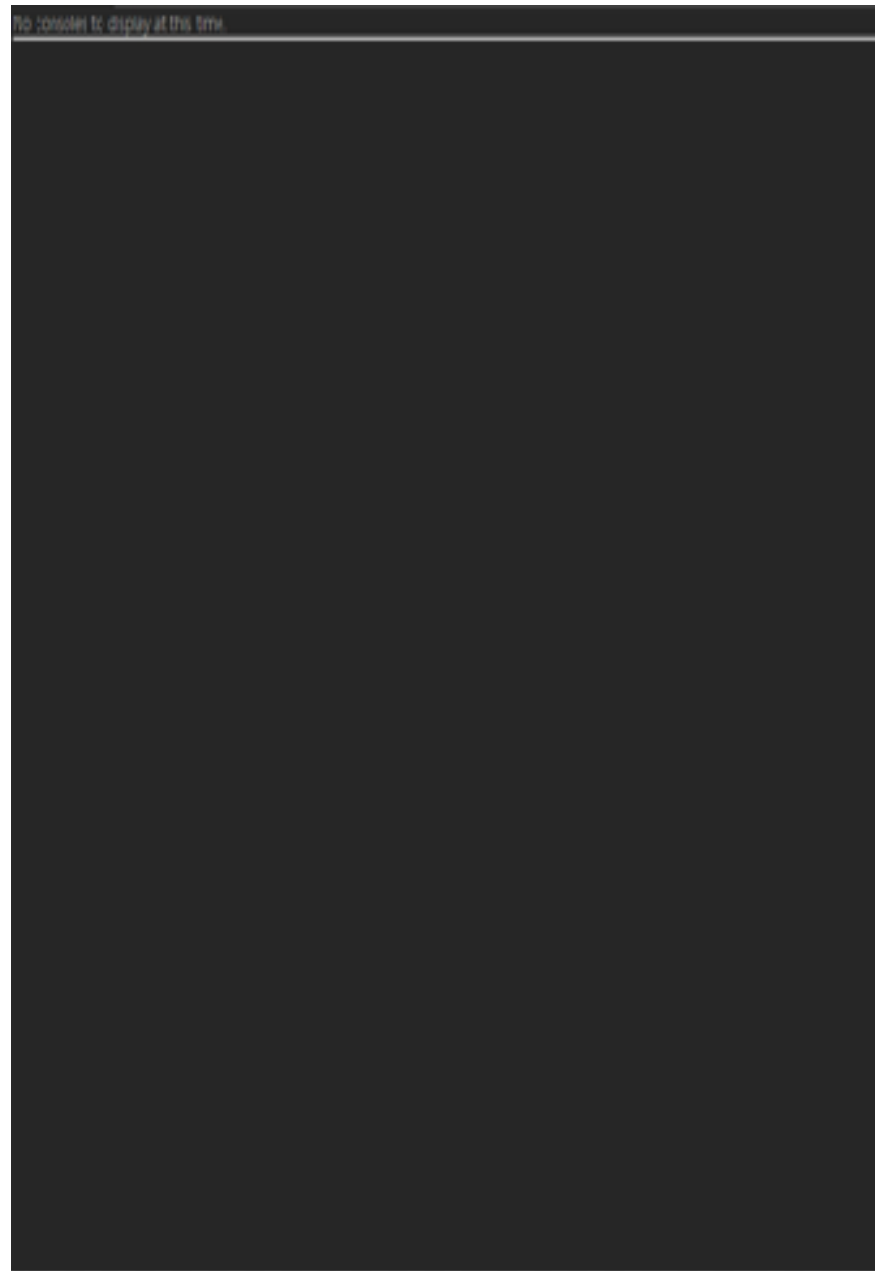
```

● 02 코드 실행테스트

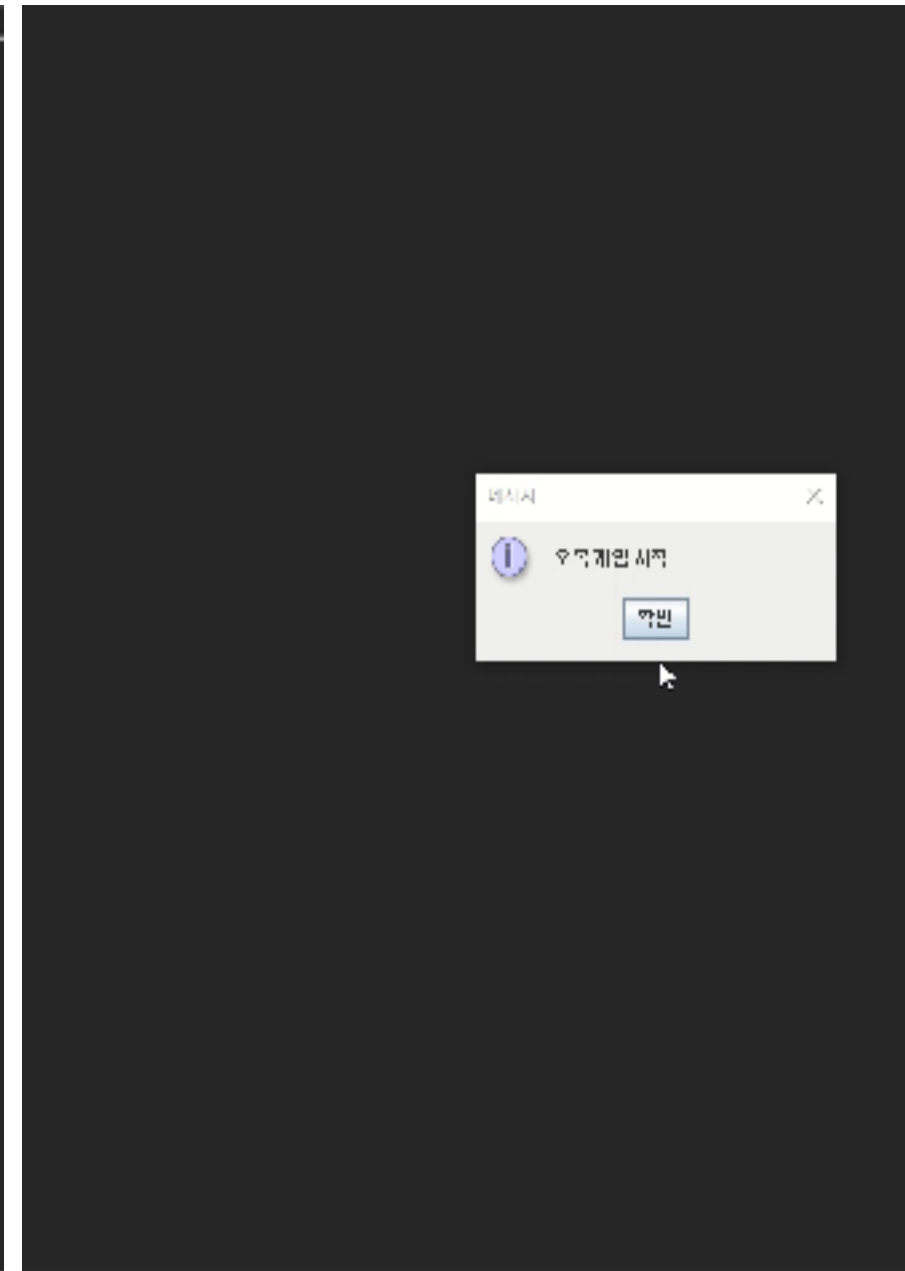
○ 02 - 02



세로 오목 & 같은 위치에 돌 놓기 오류



대각선 오목 & 잘못된 좌표에 돌 놓기 오류



오목 & 육목 시 오목 처리 불가 판단

Thank you
for
watching

2차원 배열 – 오목 게임 프로그램
램