객체지향 언어와 실습 02반

주종화 교수님

기말 프로젝트

2018112042

컴퓨터공학과

송승민

<전체적인 구성>

:기본 프레임(0)과, 오목 게임 프레임(1), 지뢰 찾기 게임 프레임(2), 순위/점수 프레임(3)로 나누어 구성함.

**0. 기본 프레임**

: 기본 프레임에서는 게임 선택창을 만듦. 버튼 배열로 버튼 4개를 만들어서, 오목 버튼을 누르면, 오목 게임 설정 패널(1-1)이 뜨도록 하고, 지뢰 찾기 버튼을 누르면, 지뢰 찾기 게임 설정 패널(2-1)이 뜨도록 함. 게임 순위/점수 버튼을 누르면, 순위/점수 패널(3)이 뜨고, 종료 버튼을 누르면, 프로세스가 종료되도록 만듦. 프로세스 종료는 System.*exit*(0); 을 이용하여 종료함

**1. 오목 게임 프레임**

: 오목 게임 패널에서는 게임 설정 패널(1-1)과 게임 프레임(1-2) 2개로 나누어 만듦.

게임 설정 패널(1-1)에서 모든 항목을 설정하면, “게임 시작” 버튼을 눌러서, 게임 패널(1-2)가 나타나도록 설정함.

*1-1. 게임 설정 패널*

: 동전의 앞 뒤 중 누가 앞 뒤를 할지 콤보박스로 지정한 다음에, 동전 앞의 사용자가 지정되면, 자동으로 동전 뒤의 사용자가 결정되도록 omP1Listener 클래스 구현함.

동전 뒤집기 버튼 클릭시, 동전 던지기 메소드가 구현되어, 선공할 사용자를 알려줌. 동전 던지기 메소드 구현시, 선공할 사용자가 starterL레이블에 작성되고, 선공하지 않은 사람의 인덱스를 faceN 정적 변수에 담는다. faceN은 MouseEventHandler클래스에서 승자를 알리는 팝업창에서 승자를 알리기 위해 user 문자열 객체 배열의 인덱스로 쓰인다.

콤보박스로 게임판 선택하여, MapSize 클래스의 SIZE 변수에 값 해당 값 대입. 이때, omP1Listener2 클래스 이용함. 콤보박스를 2개 사용하므로, ActionListener를 2개 생성함. MapSize 클래스에서는 Omok클래스에서 설정한 판의 종류를 대입하는데, Omok 객체를 따로 생성하지 않으므로, 정적 함수 getBoardNum()으로 값을 대입한다.

게임 시작 버튼을 누르면, 현재 게임 설정 창이 보이지 않고, 새로운 게임 프레임이 뜬다.

*1-2. 게임 프레임*

: 게임 프레임은 GUI 클래스로 구현함. 새로 프레임이 뜨기 때문에, 기존 프레임은 보이지 않게 만들었습니다. showPopUp함수를 이용하여, 게임 종료시, 몇 수 안에 이겼는지와 승자가 나타나도록 팝업창 뜨는 함수 구현.

DrawBoard클래스에서 전체적인 오목판을 그리고, 돌을 그리는 메소드 구현함. DrawBoard 클래스는 패널로 GUI 클래스에서 프레임에 추가되는 유일한 패널임. 생성자에 사용자 이름이 적힌 레이블을 추가하고, 한 수 물리기 버튼과 한 수마다 지정된 시간을 표시한다. 사용자 이름은 Omok클래스에서 정적 문자열 배열로 설정한 user를 이용함. timeThread 객체를 생성하여, 이 객체를 생성할때 같이 카운트 다운되도록 함.

Map 클래스에서 돌이 흑인지 백인지 알려주는 함수, 현재 위치 알려주는 함수, 흑과 백의 차례를 확인하는 함수를 만들었음. 승리를 확인하기 위해 조건문 구현하여 승리 확인함. 이후, 현재 위치를 제외한 8곳의 위치에 돌이 있는지 확인함.

MapSize클래스에서는 한 칸의 크기와 판의 종류를 지정하는 상수가 있고, 이 값들을 반환하는 함수가 있음.

MouseEventHandler클래스에서는 벽에 닿은 경우와 이미 돌이 있는 경우를 제외하고 돌이 그려져야함. 그리고 현재 위치도 받아야함. 여기서 돌을 한번 놓고 나면, 상대 차례로 바뀌면서, 다시 카운트 다운이 실행되어야 한다. 따라서 돌을 놓기 전까지 실행된 timeThread를 stop하고, 대신 DrawBoard 객체에 있던 timeThread 객체에 새로운 timeThread를 생성하여, 다시 카운트 다운되도록 함. 이겼을 경우, 승자와 총 수가 나타나도록 함.

**2. 지뢰 찾기 게임 프레임**

: 게임 설정 패널(2-1)과 본 게임 프레임(2-2)로 나누어 만들었음.

*2-1. 게임 설정 패널*

: 지뢰 개수와 게임 종류를 선택한다. Game Start 버튼 클릭 시, main 객체가 실행된다.

*2-2. 게임 프레임*

: 시간, 리셋, 폭탄 개수가 나타나는 패널과 지뢰밭이 나타나는 패널 2개로 나누어 작성함. level로 게임의 종류를 구분함. ButtonListener를 구현하여, 지뢰가 아닌 버튼 클릭시 실행되도록 함. MineListener를 구현하여 지뢰 클릭시에는 현재까지 진행시간을 저장하고, 메시지 팝업창 띄움. clickListener를 구현하여 깃발 표시를 하기 위한 이벤트를 구현. 2번 클릭시 물음표로 나타나도록 조건문 구현. Victory()함수를 이용하여 승리 나타냄. MyThread클래스를 구현하여 게임 시작부터 진행된 시간 나타나도록 함.

**3. 순위/점수 프레임**

: 사용자 이름과 점수와 순위는 Vector를 이용함.

3-1. 오목 게임 순위

3-2. 지뢰 게임 순위

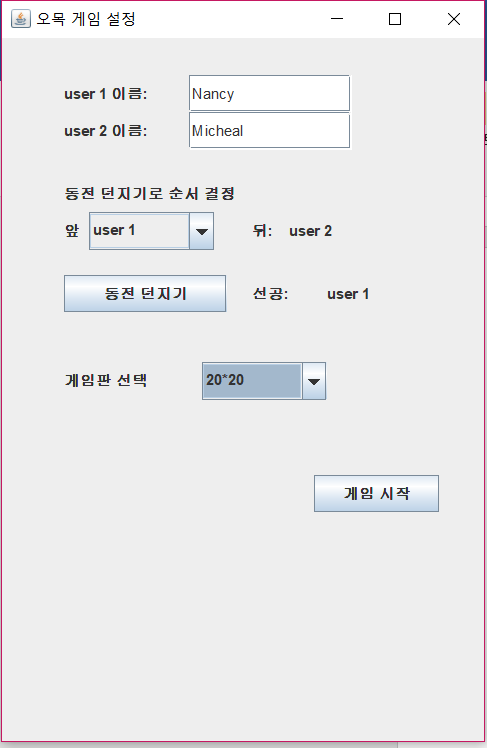
<기본 프레임>



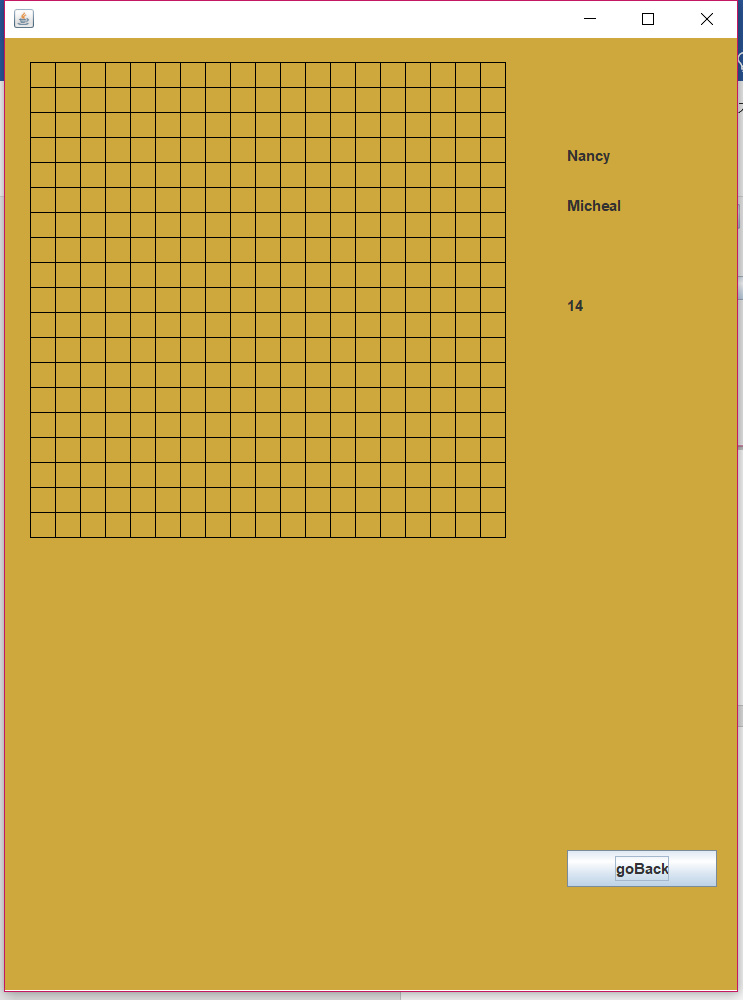
: 게임 선택 창

<오목 게임 프레임>

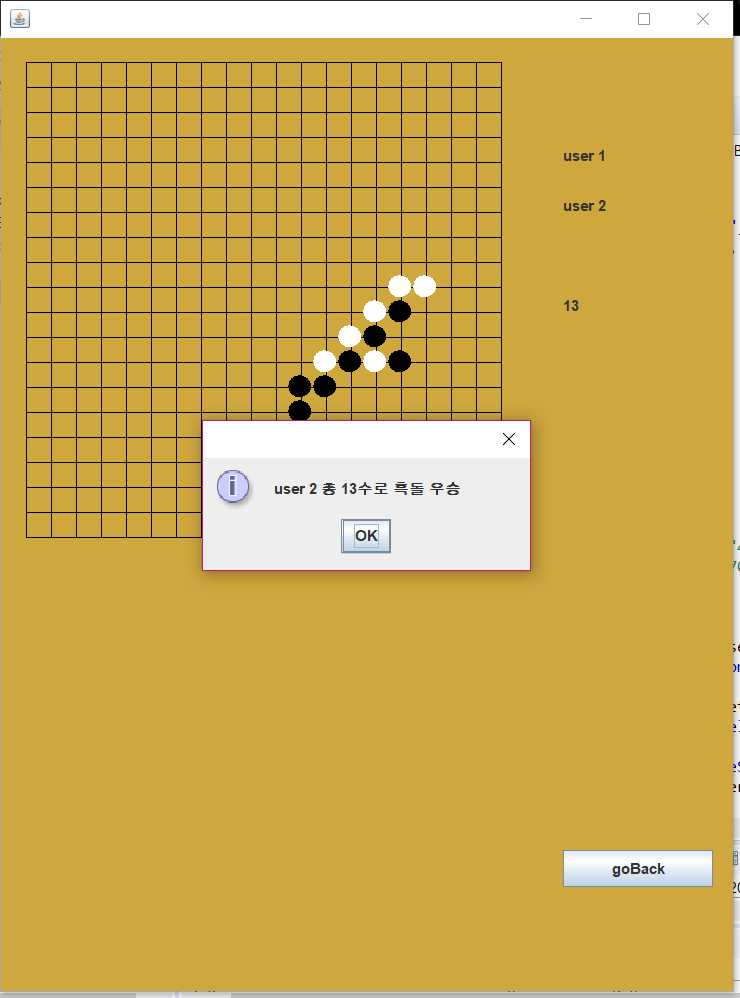
1-1. 오목 게임 패널



1-2. 오목 게임 프레임

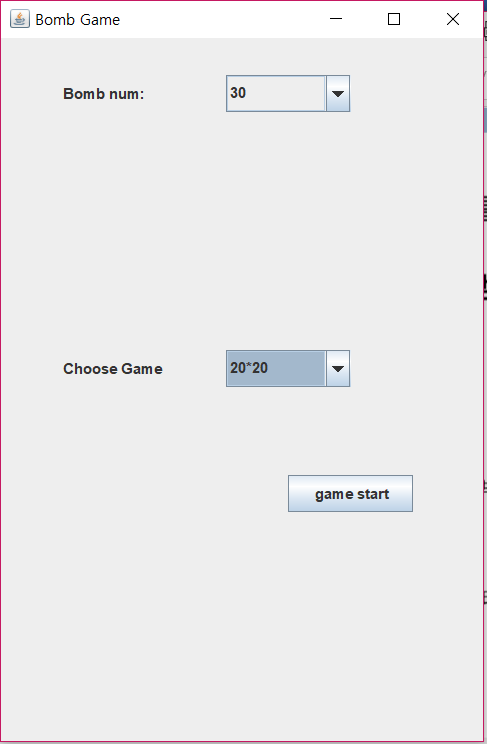


게임 진행 시 화면

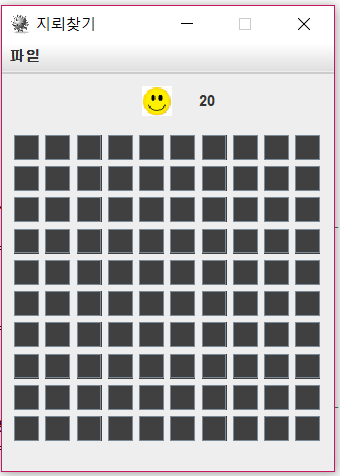


<지뢰 찾기 게임 프레임>

2-1. 게임 설정 패널



2-2. 지뢰 게임 프레임



게임 진행 화면

