

**소스코드분석 1차과제**

**MinesFinder 분석/개선하기**

**11조**

201613998 소프트웨어공학과 설경원

201818218 소프트웨어공학과 한별



**목차**

**01 팀원소개** : page.3

**02 개선아이디어** :page.3~4

1. 개선 아이디어 정리 및 난이도5

**03 분석 및 개선 다이어그램** : page.5

1. 기존 코드 분석 class Diagram5

2. 개선 후 분석 class Diagram 6

**04 각 과제 내용에 대한 구현 방법 설명 및 Code Snippet : page.7~27**

1. 전체 화면 capture7

2. 각 개선 아이디어 정리.13

**05 느낀 점 : page.28**

**06 참고 자료 : page.29**

**○ 팀원 소개**

**- 팀원 소개**

**11조.**

교수님이 질문/답변 발표중에 이름이 연예인 같다고 하셨던 조입니다.

저희에 대한 설명과 하고 싶은말은 건 **느낀점**에서 적도록 하겠습니다.

**201613998 소프트웨어공학과 설경원**

**201818218 소프트웨어공학과 한별**

**○ 개선 아이디어**

**- 개선 아이디어 정리 및 난이도**

**1. 사용자 설정 타임어택: 0.5점**

게임 시작 전 사용자가 게임시간을 설정 한 후에 설정한 시간이 초과하면 사용자에게 게임을 더 할것인지 그만할 것인지를 물어보는 기능을 넣었습니다.

**2. 배경음악, 효과음 삽입 : 1점**

청각적인 측면에서 아무 기능이 없는 프로그램에 배경음악과 버튼을 눌렀을 때와 지뢰가 터진상태(BUSTED) 되었을 때 효과음을 주었습니다.

**3. 키보드를 이용해 플레이할 수 있는 기능 추가 : 0.5점**

마우스 뿐만이 아닌 키보드를 이용해 플레이할 수 있는 기능을 추가하였습니다.

**4. 사용자가 맵을 설정한 2인용 모드 : 5점**

원래는 서버와 소켓을 이용한 네트워크 멀티플레이기능을 구현하려 했으나 오랫동안 구현하던 도중에 어려움을 느껴 사용자가 맵을 설정한 2인용 모드로 바뀌게 되었습니다.

**5. 지뢰 발견시 모든 지뢰 나타내기 : 1점**

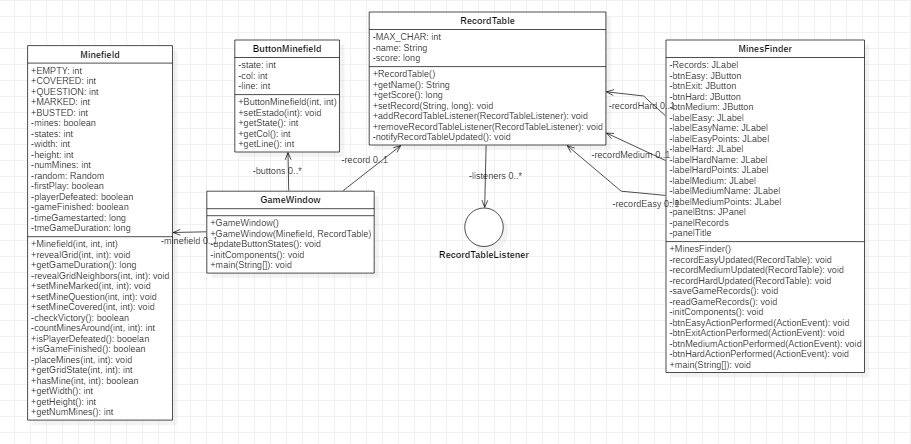
원래 개선 아이디어는 아니였지만 개선 아이디어 구현 도중 넣은 기능입니다. 지뢰를 찾아 게임을 패배하였을 시에 모든 지뢰를 나타내어주는 기능입니다.

**6. 게임 결과 복기 하기 : 2.5점**

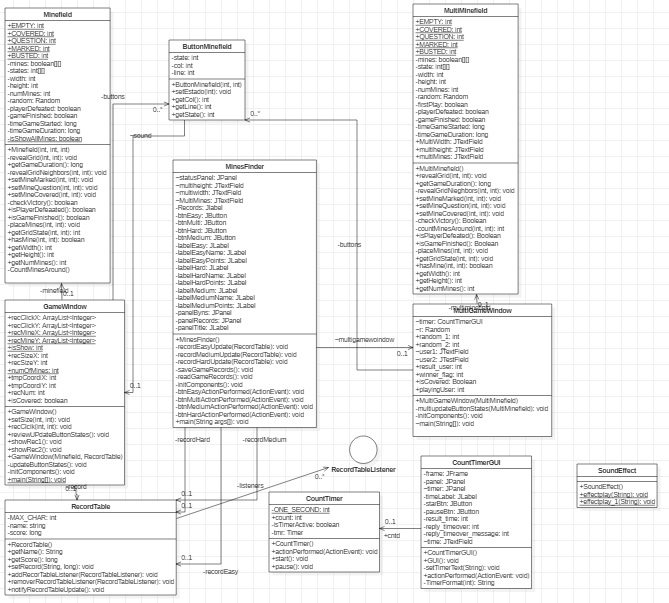
원래 개선 아이디어 (사용자 설정 ON/OFF, 플레이 결과 feedback 가능, **다른 사람들과 공유가능**) 에서 **다른 사람들과 공유가능 (저장 : 2.5점)** 기능을 제외한 게임 결과 복기 하기를 구현하였습니다.

**○ 분석 및 개선 다이어그램**

**- 기존 코드 분석 Class Diagram**

****

**- 개선 후 분석 Class Diagram**

****

**○ 각 과제 내용에 대한 구현 방법 설명 및 Code Snippet (코드 정보)**

**- 1. 전체 화면 Capture**

|  |
| --- |
|  |
| **1.1-1사용자 설정 타임어택- 타임어택을 위해 게임시간을 설정하는 화면 캡쳐** |

|  |
| --- |
|  |
| **1.1-2 사용자 설정 타임어택- 타임어택을 위해 게임시간을 설정하는 화면 캡쳐** |

|  |
| --- |
|  |
| **1.2 사용자 설정 타임어택- 시간을 초과한 후 게임 지속 여부를 물어보는 다이얼로그 캡쳐** |

|  |
| --- |
|  |
| **4.1-1 사용자가 맵을 설정한 2인용 모드- 맵 설정하기** |

|  |
| --- |
|  |
| **4.1-2 사용자가 맵을 설정한 2인용 모드- 유저 이름 설정하기** |

|  |
| --- |
|  |
| **4.1-3 사용자가 맵을 설정한 2인용 모드- 선공 유저 알려주기(순서 정하기)** |

|  |
| --- |
|  |
| **4.2 사용자가 맵을 설정한 2인용 모드- 이긴 유저 알려주기** |

|  |
| --- |
|  |
| **5 지뢰 발견시 모든 지뢰 나타내기** |

|  |
| --- |
|  |
| **6 .1게임 결과 복기하기- 게임 복기 여부를 선택한다** |

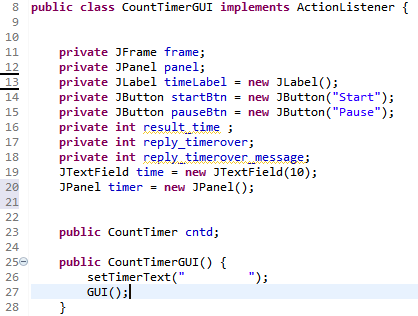
|  |
| --- |
|  |
| **6.2-1 게임 결과 복기하기-게임 복기를 시작한다** |

|  |
| --- |
|  |
| **6-2.2 게임 복기하기-게임 복기중** |

|  |
| --- |
|  |
| **6-3 게임 복기하기- 게임 복기 완료** |

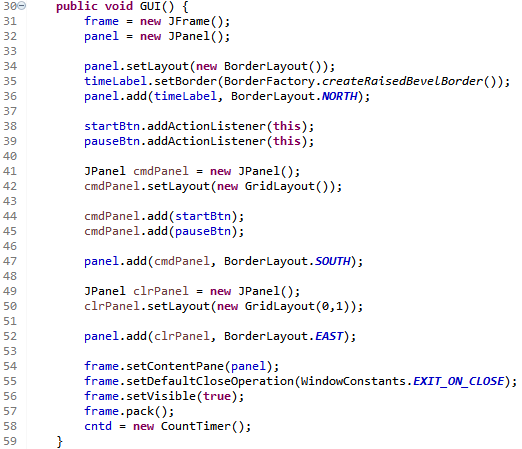
**- 2. 각 개선 아이디어 정리**

**1. 사용자 설정 타임어택 : 0.5점**

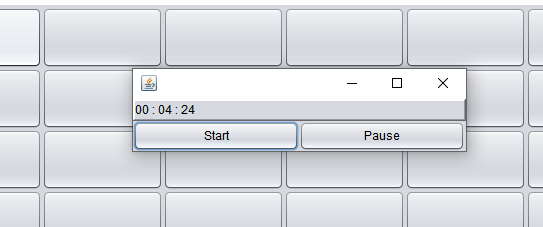
****

CountTimerGUI라는 클래스에서 GUI를 따로만들어 타이머를 구현해보았습니다.

구성요소로 startbtn(시작), pauseBtn(일시정지) 가 있고, 후에 시간 초과와 시간 초과 메시지를 담을 변수를 선언해주었습니다. JTextField로 선언된 time 변수는 사용자가 시간을 정하기 위해 변수를 선언하였는데 단위는 초단위 입니다.

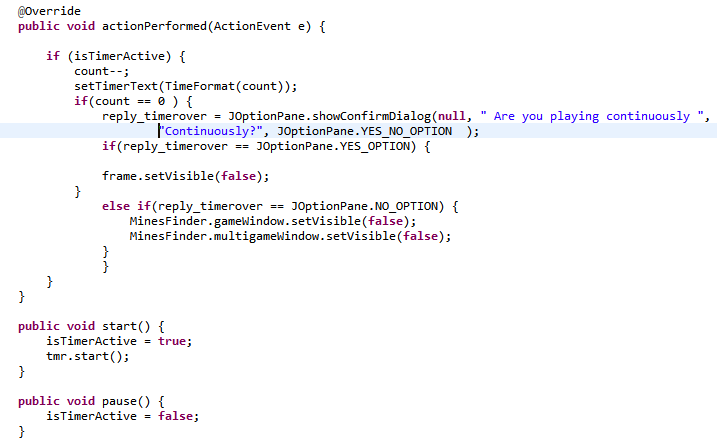


위의 코드는 Timer의 GUI부분인데 Layout설정과 startbtn, pausebtn 구현의 코드입니다. 아래는 코드 결과 TimerGUI() 입니다.

****

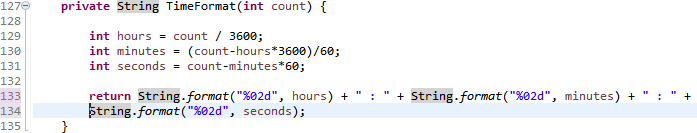
****

위의 코드는 start와 pause 버튼을 actionformed를 통해 활성화 시킨 코드와 사용자가 시간을 설정하는 코드입니다. JTextField로 사용자에게 받은 String 타입의 time을 Integer.parseInt로 Integer타입으로 변환시켜주는 모습입니다.

****

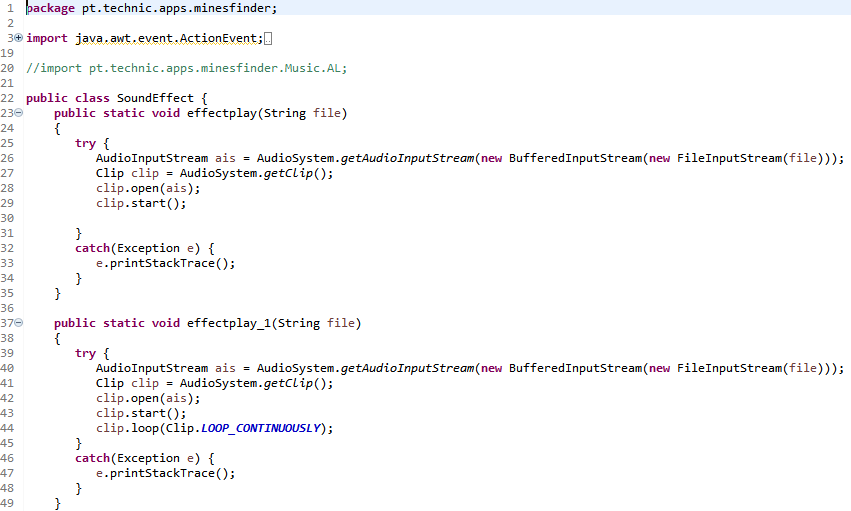
위의 코드는 사용자가 시간을 설정해주면 count-- 를통해 시간이 감소되고 count가 0이되었을때 사용자에게 계속 게임을 진행할 것인지 종료할 것인지 물어보고 Yes면 계속 진행, No면 종료하게 되는 코드입니다.

그리고 start()와 pause()의 역할을 isTimerActive 코드를 통해 나타내어 준 모습입니다.

****

마지막 시간을 시,분,초에 맞게 나타내어주는 코드입니다. 초를 입력받아 3600으로 나눈 시와 시를 빼고 60으로 나눈 분, 마지막 초까지 코드로 나타내주었습니다.

**2. 배경음악, 효과음 삽입 : 1점**

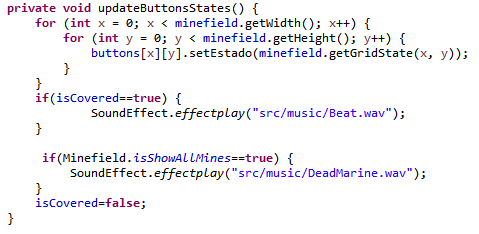


효과음과 배경음악은 SoundEffect라는 클래스에서 기능을 구현하였습니다. 위의 코드가 그 내용인데 effectplay는 효과음을 effectplay\_1에서는 배경음악을 나타내기 위해 메소드를 구별하여 구현하였습니다. 우선 효과음은 Clip이라는 java에서 제공하는 객체를 선언하여 AudioSystem.getclip 이라는 함수를 써서 받아오고 파일의 경로명을 지정해주어 open()과 start()를 이용하여 나오게 하였습니다.

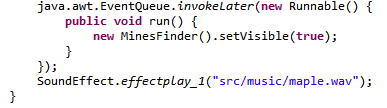
****

위의 코드와 같이 ButtonMinefield 클래스에서 COVERED상태에서와 BUSTED상태에서 효과음을 주도록 표현했습니다. 문제를 해결하는 과정에서 2가지 문제가 있었는데 하나는 COVERED에서 나는 소리가 연속해서 나는 것이었고, 하나는 mp3파일은 되지 않는 것이었습니다.

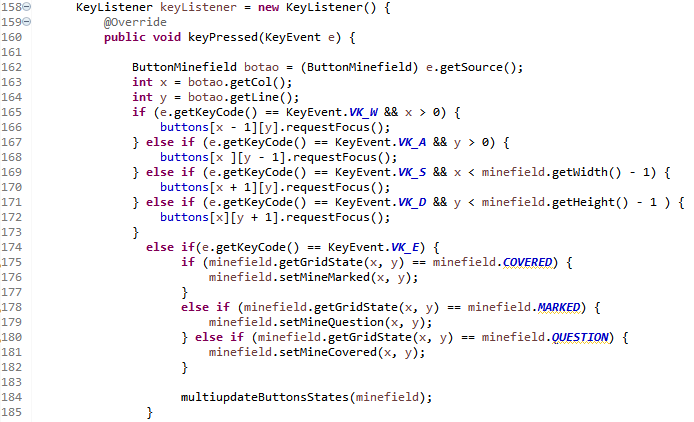
연속해서 소리가 나는 것은 아래의 코드처럼 GameWindow에서 updateButtonsStates()메소드를 수정하여 바꿔주었고, mp3파일은 wav형식으로 변환해서 해결할 수 있었습니다.

****

배경음악은 Minesfinder 클래스의 메인함수에서 다음과 같이 effectplay\_2를 선언하여 메인함수가 실행되자마자 나오게 하였고, LOOP\_CONTINUOUSLY를 사용하여 프로그램이 종료되기 전가지 무한정 재생되게 하였습니다. 한 가지 문제가 있는데 파일 경로명을 설정해주어 C:\\과제\\ , C드라이브에 과제라는 폴더에 Beat.wav , maple.wav 등과 같은 파일이 없으면 실행이 되지 않는데 해결하지 못하였습니다. 제출은 과제 폴더에 음악 파일을 넣어 제출하였습니다.

****

**3. 키보드를 이용해 플레이할 수 있는 기능 추가 : 0.5점**



클래스는 GameWindow와 MultiGameWindow에서 구현하였고 2인용 모드를 키보드를 통해 구현하였기 때문에 2인용 모드를 위해서는 필수적으로 해야될 개선사항이였습니다. 개선 전 원본 코드에서도 다음과 같이 키보드의 입력을 받는 public void KeyPressed(KeyEvent e)라는 메소드가 구현되어 있었는데,



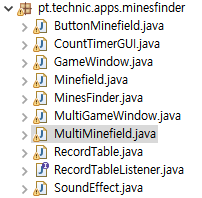
키보드에 상하좌우,M키를 눌렀을 때, 동작하지 않았고, 특히 왼쪽, 위쪽, M 키를 눌렀을 때에는 콘솔창에 다음과 같은 오류가 발생하였습니다.



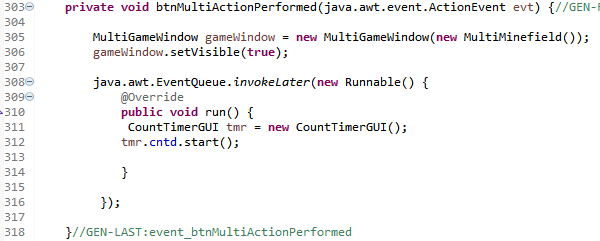
위 오류 문구는 인덱스가 배열의 크기를 벗어났다는 내용입니다. 이는 botao.getWidth(), botao.getHeight의 값이 button[x][y]의 인덱스를 가리키는 것이 부적절하기 때문에 발생하는 오류입니다.

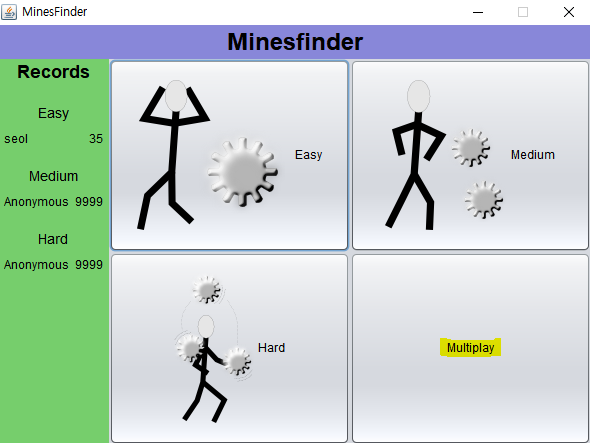
botao.getWidth(), botao.getHeight()는 ButtonMinefield의 크기를 나타내고, button[x][y]의 인덱스가 buttonMinefield botao에서의 버튼의 가로 위치, 세로 위치를 나타내므로, x, y 값에 botao.getWidth(), botao.getHeight()의 값을 대입하는 것이 아니라 botao.getCol, botao.getLine을 대입하여야 합니다.

button[0][0]이 가장 왼쪽상단의 버튼을 의미하고, button[x][y]는 위로부터 x+1번째 줄, 왼쪽으로부터 y+1번째 줄의 버튼을 의미합니다. 그러므로 위쪽으로 포커스를 이동하고 싶다면 x==0이라면 가장 위쪽 칸이므로 위쪽으로 이동할 수 없습니다. 그러므로 x>0이라는 조건이 있어야 합니다. 그러므로 button[x][y]에서 포커스를 위쪽으로 이동하도록 하기 위해서는 x>0이고 위쪽 버튼이 키보드 입력되었을 때, button[x-1][y].requestFocus()가 되도록 하면 됩니다. 같은 방식으로 코드를 수정해 키보드를 입력받을 수 있었습니다.

**4. 사용자가 맵을 설정한 2인용 모드 : 5점**

MultiMinefield와 MultiGameWindow라는 클래스를 새로 생성해 2인용 모드에서 사용하는 요소들을 따로 관리하였습니다. 왜냐하면 MultiMode는 사용자 설정 요소들이 많이 들어가기 때문입니다. 따라서 따로 관리하면 후에 개선할 때 편하게 관리할 수 있고, 한눈에 일반모드와 구별할 수 있습니다.

****

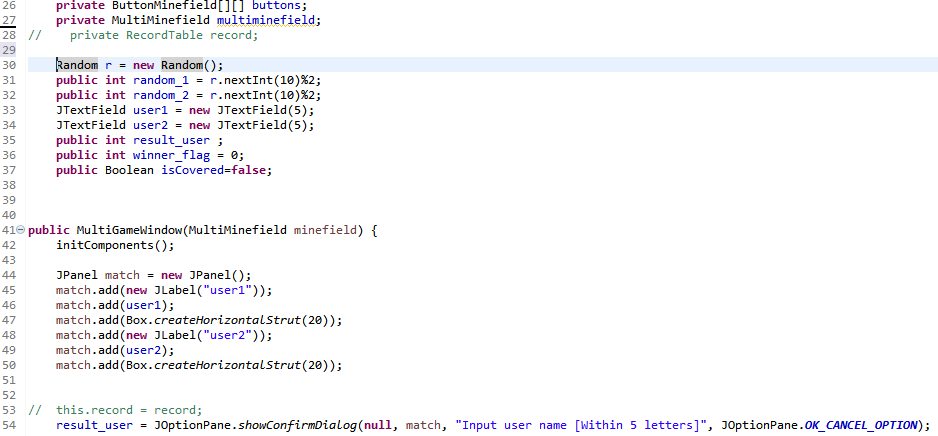
****주 main함수가 있는 MinesFinder 클래스에서는 multiplay버튼을 눌렀을 때 MultiGameWindow의 MultiMinefield와 위에서 구현한 사용자 설정 타이머가 구현되는걸 보실수 있습니다.

먼저 MultiMinefield부터 설명 드리자면

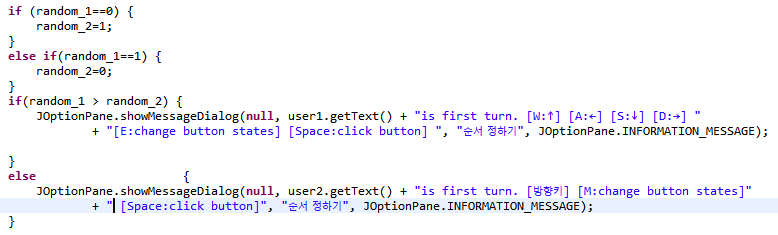


원래의 Minefield클래스는 높이, 너비, 지뢰갯수를 정해주지만 Multi의 사용자 모드라는 특성을 살려 JPanel을 받아 JTextField 3개(높이, 너비, 지뢰갯수)를 받아 사용자가 설정할수 있게 개선해보았습니다. JTextField의 크기를 2로 설정해줌으로써 사용자에게도 “within two digits” 알려주며 너무 커지지않음을 막았습니다.

다음은 MultiGameWindow 입니다.



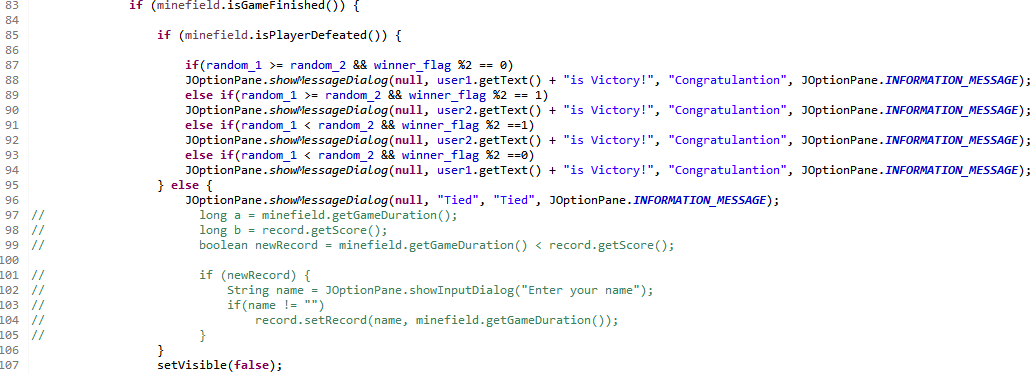
원래의 GameWindow는 Record를 저장하지만 랭킹을 구현하지 않아 승자만 바로 알려주는 것이 맞다 생각하여 record값을 주석처리하고 생성자에서도 record값을 받지 않았습니다. 변수들을 보시면 Random값과 JTextField user, result\_user, winner\_flag가 추가되었는데 일단 user는 사용자가 이름(닉네임)을 입력할 수 있게 하기위에 JTextField를 사용하였고, result\_user에 입력받는 모습을 보실 수 있습니다. 이름은 5자이내로 정하였고 showconfirmDialog와 OK\_CANCEL\_OPTION을 사용해 입력받을 수 있는 알림창을 구현하였습니다.

****

random값을 통해 순서를 정하는 단계입니다. 사용자1은 W,A,S,D를 통해 방향키, E키를 통해 마우스의 오른쪽 기능인 버튼을 MARKED, QUESTION 상태로 바꿀수 있으며, 사용자 2는 방향키와 M키를 통해 상태를 바꿀 수 있습니다. 보시면 Spacebar가 두 사용자의 버튼 클릭기능인데 후에 winner\_flag를 변화시킬 키 입니다..

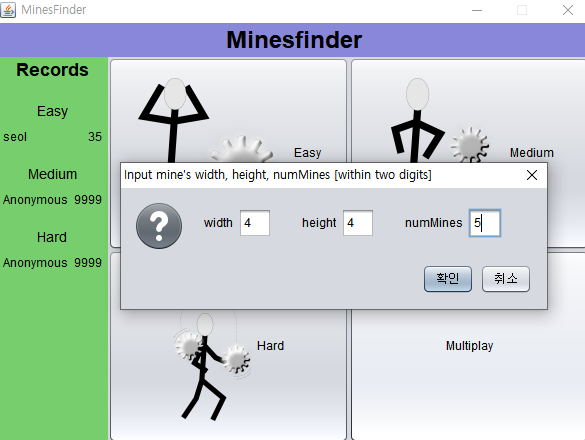


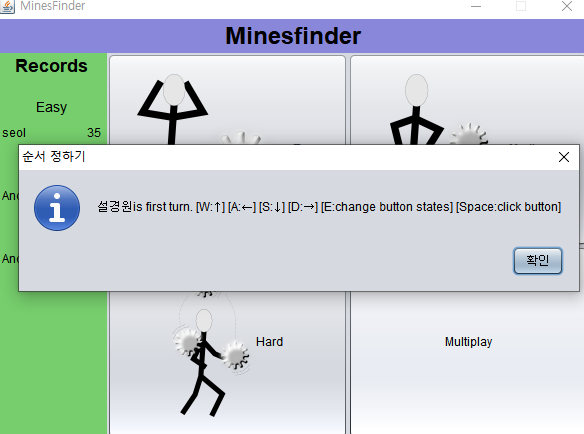
winner\_flag는 SPACE가 눌렸을때 하나씩 증가하게 되는데 누가 선인지 알게되면 승리자를 알 수 있게 되는 방식입니다.

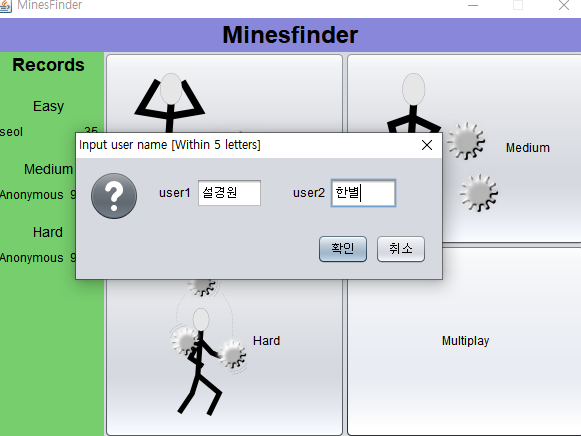


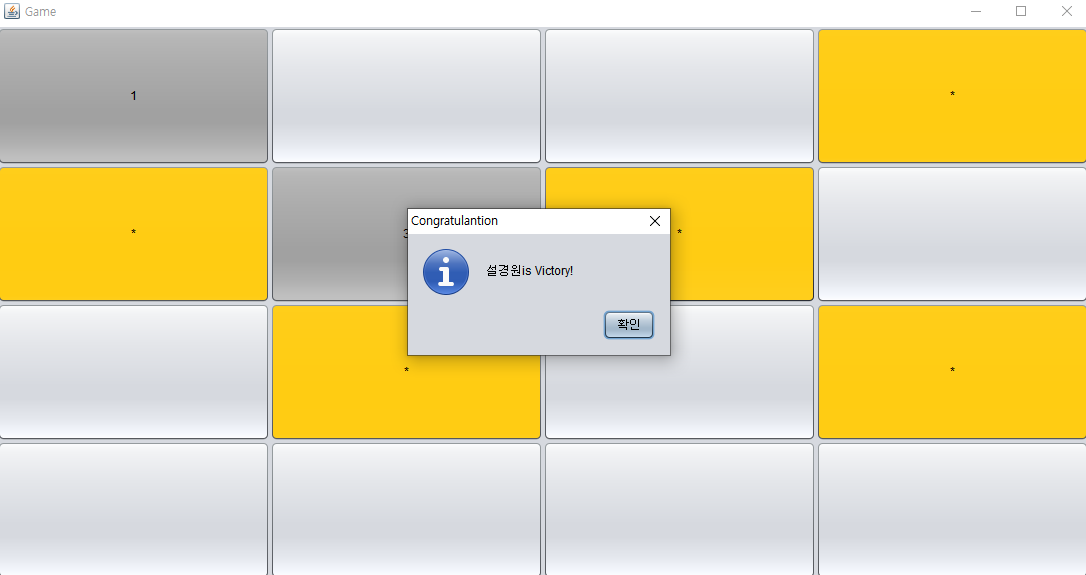
.위의 그림과 같이 random값으로 선을 알고 winner\_flag의 숫자만 알게되면 2인용모드에서는 winner\_flag를 2로 나눈 나머지의 값과 선이 누군지의 조합으로 누군가가 지뢰를 밟았을때(isPlayerDefeated()) 승리자를 알 수 있습니다. 아무도 지뢰를 밟지않고 게임이 승리조건이 갖추어지면 무승부로 설정하였습니다. 아래의 recordtable에 기록되는 부분은 승리자만 알려주는 방법으로 대체하여 주석처리해주었습니다.

아래는 순서대로 사용자 설정 맵정하기, 선 알려주기, 이름 입력하기, 선(설경원)인 상태에서 두번(선, 후) 클릭 후에 후(한별)에서 지뢰를 밟아 승리자(선)를 표시해주는 결과 창입니다.

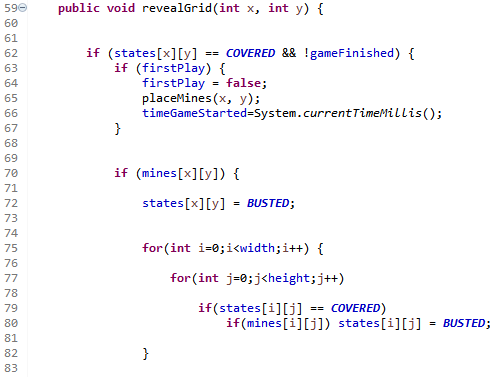




****

****

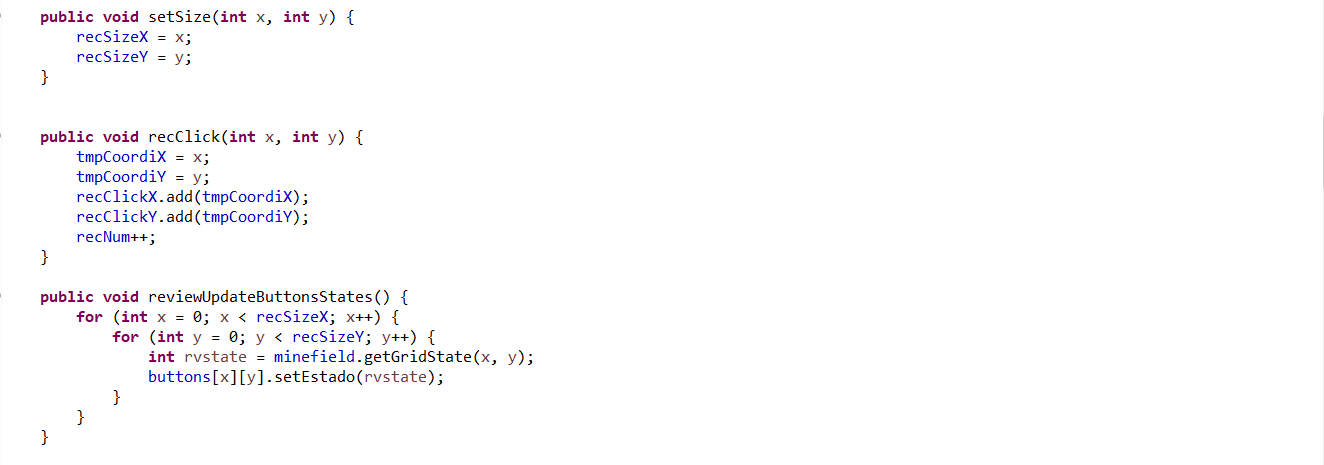
**5. 지뢰 발견시 모든 지뢰 나타내기 : 1점**

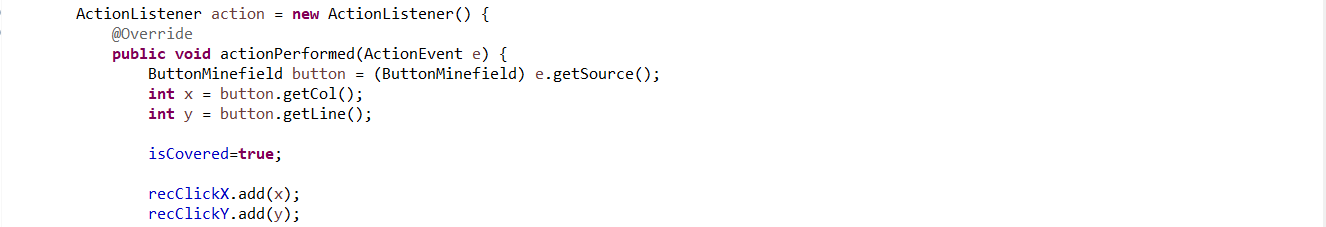


구현 한 코드중에 이해만 쉽게 했다면 가장 간단하게 구현할 수 있는 코드였습니다. 클래스는 Minefield와 MultiMinefield에서 구현하였고 버튼들의 상태를 확인해주는 revealGrid(int x, int y) 클래스에서 구현해주었습니다. 코드에 대해 간단하게 설명해보면 실제로 구현한 것은 코드의 75번째 라인에서 82번째 라인인데 선택한 버튼이 지뢰가 있다면(mines[x][y]) 선택한 버튼뿐만이 아닌 모든 지뢰(COVERED상태여서 보이지 않는)를 BUSTED로 바꿔주는 2중 for문을 사용해 간단하게 구현하였습니다.

**6. 게임 결과 복기 하기 : 2.5점**

게임 결과 복기하기는 게임에서 이기는 경우 이용자의 선택에 따라 지난 게임의 진행을 보여주는 기능입니다.

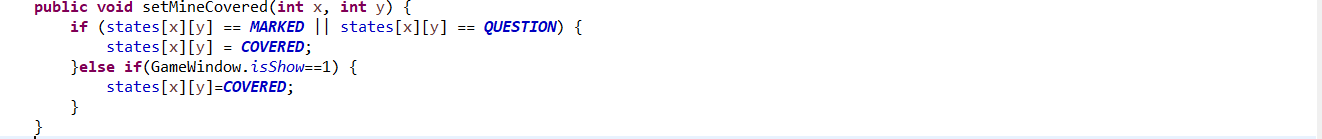
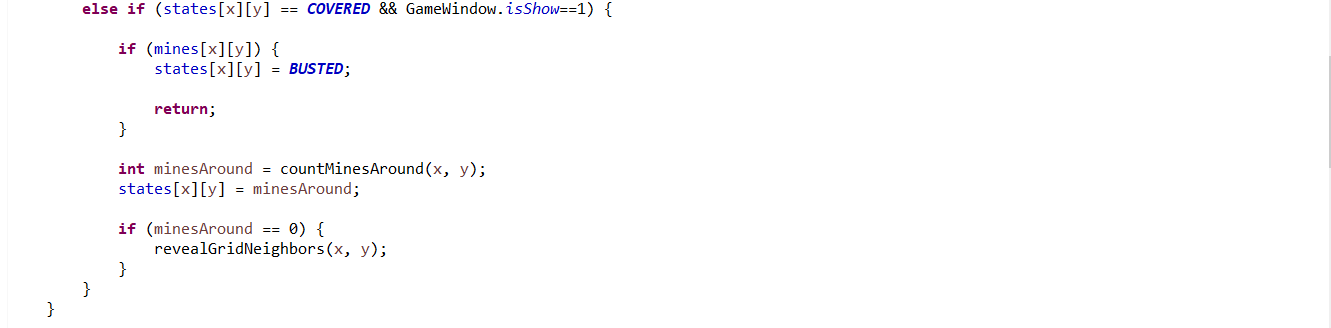


setSize는 게임 맵의 크기에 따라 button[][]배열의 크기가 달라질 수 있으므로 버튼의 가로개수, 세로개수를 각각 recSizeX, recSizeY에 저장하는 메소드입니다.

recClick(x,y)는 ArrayList recClickX, recClickY에 각각 마우스의 클릭 입력/ 키보드의 스페이스 입력이 이루어지는 버튼의 좌표값을 x, y 성분으로 분해하여 저장하는 메소드입니다.

reviewUpdateButtonState()는 GameWindow에 기존에 존재하던 UpdateButtonState를 변형한 것으로, UpdateButtonState와 달리 효과음 기능을 제외하였습니다.



showRec1()은 isGameFinished()==true이고 minefield.isPlayerDefeated()==false인 상황에서 복기 여부를 조사하는 dialog에서 Yes를 선택했을 때 실행되는 메소드로, minefield의 모든 버튼을 COVERED상태로 만들고, showRec2()를 실행합니다.

showRec2(int recSizeX, int recSizeY, int recNum, ArrayList<Integer> reclickX, ArrayList<Integer> reclickY)는 minefield의 버튼 개수, 클릭 또는 스페이스바로 입력받은 횟수와 그 좌표값을 이용하여 매 입력마다의 변화를 보여줍니다. 각 입력마다 보여주는 시간은 사람마다 적절한 시간이 다를 수 있기 때문에 매 입력마다 information message을 띄워 다음 입력을 보여주도록 하였습니다.

마지막으로, int isShow라는 변수를 두어 setMineCovered(), revealGrid()를 showRec1(),showRec2()에서 이용할 수 있도록 변경하였습니다**.**

**○ 느낀 점**

**201613998 소프트웨어공학과 설경원**

저는 대전에서 태어나 사실은 과에 큰 관심이 있었던 것은 아닌 상태로 전북대학교 소프트웨어공학과에 입학하게 되었습니다. 1학년때는 학고를 받을 정도로 공부도 안하고 편입이나 전과 생각도 많이 하였습니다. 그러다가 1사단(파주)에서 군복무하면서 휴가나오면 지도교수님이셨던 이지현교수님께 상담을 하곤했습니다. 휴가나오기전에 여쭤볼 질문들을 정리해서 적어놓고 상담하며 최대한 과에 흥미를 가지고자 많은 노력을 했던 것 같습니다. 군대에서의 결과물로는 정보처리기능사 자격증과 C언어로 손코딩한 노트2권과 나중에 실력이 향상되면 만들고 싶은 app을 정리해놓은 노트가 있습니다. 서론이 길었는데 제가 상담했던 내용중**(마지막에 참고자료로 첨부하였습니다)**에는 나중에 복학하면 다른사람과 코딩할수 있는 기회가 있는지도 있었습니다. 항상 혼자 손코딩했던 저에게는 정말 중요했습니다. 3년 넘게 공부를 했지만 실제로 제 바램과 가장 가까운 공부를 했던 건 이번 소스코드분석에서 다른사람과 하는 프로젝트였던 것 같습니다. 처음 해본 프로젝트라 간단한 로직 넣는 것도 정말 어렵고 공부를 그동안 헛되게 한 것 같았습니다. 하지만 같이 서로를 격려하고 하는 단계에서 조금은 돈독해진것도 같습니다. 가장 많이 느끼고 배우게 된 부분은 클래스가 많을 경우에 그들간의 관계파악의 중요성과 알고 싶은 내용을 어떠한 제시어로 검색을 하는지도 알게된 거 같고 앞으로 공부방향을 잡을 지와 개발자로써 생각한 것을 코드로 구현하는게 얼마나 어려운지를 느끼고 배우게 된 것 같습니다. 저는 군대에서 가지게 된 생각이지만 힘든 만큼 남는게 많이 있다고 생각합니다. 이번 소스코드분석 과제를 통해 많이 남았다고 생각하고 다음에 더 큰 프로젝트를 위한 한 단계 나아갔다고 생각하겠습니다. 느낀점을 길게 썼는데 어려운 시간 내서 읽어주셔 감사합니다. 이상입니다.

**201818218 소프트웨어공학과 한별**

소프트웨어공학과에 입학한 후에 java나 c언어, c++를 처음 배웠습니다. 수업시간에 교재를 통해 개념을 배우고 간단한 예제를 풀어보았지만 실제로 어떻게 하면 c언어나 java로 이루어졌다는 주변의 소프트웨어들을 만들 수 있는지, 책에 나와있는 예제 이상의 공부를 하려면 어떻게 해야 하는 것인지 궁금했습니다. github나 기타 인터넷 사이트들을 통해서 도움이 될 만한 참고자료를 찾아보거나 다른 사람의 소스코드를 보고 저한테 도움이 될 수 있는 부분을 참고하거나 예제로서 공부하는 방법에 대해서 알 수 있었습니다. 또, 지금까지 해왔던 예제 수준보다 복잡한 코드를 보고 수정해보는 것도, 다른 사람과 함께 소스 코드를 수정하는 것도 이번 과제가 처음이어서 이번 과제를 통해 새로운 경험을 많이 해 본 것 같습니다.

**○ 참고 자료**

사실 첨부하지 않아도 되지만 첨부한 이유는 나중에 혹시 진로나 코드 관련 궁금한 점이 있다면 지도 교수님은 아니시지만 교수님이 가능한 시간대에 미리 연락드려 상담해도 되는지 여쭤보기 위함인데 혹시 보시게되면 말씀해주시면 정말 감사하겠습니다!

