**소스코드 분석**

**1차 과제 보고서**

2019.10.21. (월)

소프트웨어공학과

201811218 김효진

201310501 박대성

양천가

****

**목 차**

**1. 팀원 소개 4**

**1.1 팀원 소개 4**

**2. 개선 아이디어 4**

**2.1 개선 아이디어 정리 및 난이도 4**

**3. 분석 및 개선 다이어그램 5**

**3.1 기존 코드 분석 Class Diagram 5**

**3.2 아이디어 개선 후(리팩토링 전) 코드 분석 Class Diagram 6**

**3.2 리팩토링 후 코드 분석 Class Diagram 7**

**4. 각 과제 내용에 대한 구현방법 설명 및 Code Snippet 8**

**4.1 전체 화면 Capture 8**

**4.2 각 개선 아이디어 정리 13**

**4.2.1 아이디어 1 - 게임 설명 기능 추가 13**

**4.2.2 아이디어 2 - 이미 밝혀진 버튼을 다시 한번 누르면 그 버튼 주변에 밝혀지지 않은 버튼들을 어둡게 표시해주는 기능 14**

**4.2.3 아이디어 3 - 게임 도중에 경과시간과 남은 지뢰 개수를 보여주는 상태 바 추가 16**

**4.2.4 아이디어 4 - 버튼 개수나 지뢰 개수를 사용자가 설정하는 기능 18**

**4.2.5 아이디어 5 - 지뢰가 터지면 효과음이 나오는 기능 19**

**4.2.6 아이디어 6 - 게임 머니 기능을 추가하여 게임 성공 시 받을 수 있는 기능 20**

**4.2.7 아이디어 7 - 게임머니를 사용하여 지뢰가 심어지지 않은 곳을 랜덤으로 알려주는 힌트 기능 추가 21**

**4.2.8 아이디어 8 - 자신의 기록을 따로 볼 수 있는 메뉴 추가 22**

**4.2.9 아이디어 9 - 사용자가 언어를 설정할 수 있는 기능 추가 23**

**4.2.10 아이디어 10 - 글로벌 랭킹 기능 추가 25**

**4.2.11 아이디어 11 – 로그인 기능 추가 29**

**4.2.12 아이디어 12 – 게임 끝났을 때 모든 지뢰 밝혀주는 기능 추가 30**

**4.2.13 아이디어 13 – 게임 중 다시하기 기능 추가 32**

**5. 리팩토링(Refactoring) 34**

**5.1 리팩토링한 내용 정리 34**

**5.1.1 Long Method-gamesetting 34**

**5.1.2 Long Method-initCompoent 35**

**5.1.3 Long Method-ActionListener 36**

**5.1.4 Long Method-난이도 별 레코드 36**

**5.1.5 Large Class-action 37**

**5.1.6 Large Class-text 38**

**5.1.7 Duplicated Code-font, color, dimension, cursor 40**

**5.1.8 Duplicated Code-btn 모아 놓기 40**

**5.1.9 Data Clumps-language 41**

**5.1.10 Visibility 41**

**5.2 리팩토링 후 stan4j 실행 결과 42**

**5.3 리팩토링 후 Sequence Diagram 43**

**6. 느낀점 51**

**1. 팀원 소개**

**1.1 팀원 소개**

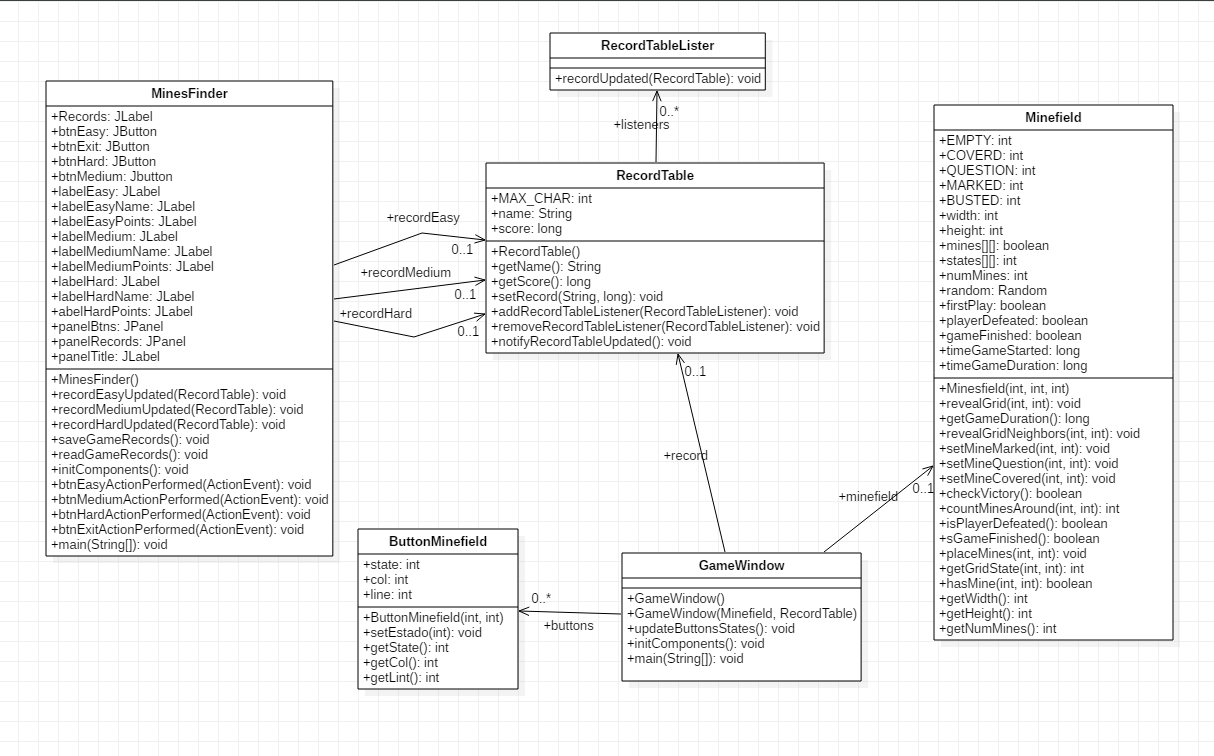
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 팀  구  성 | 성 명 | 김효진 | 박대성 | 양천가 |
| 학 번 | 201811218 | 201310501 |  |
| 연락처  사 진 | EMB00006bf04793 | **EMB00006bf04794** |  |

**2. 개선 아이디어**

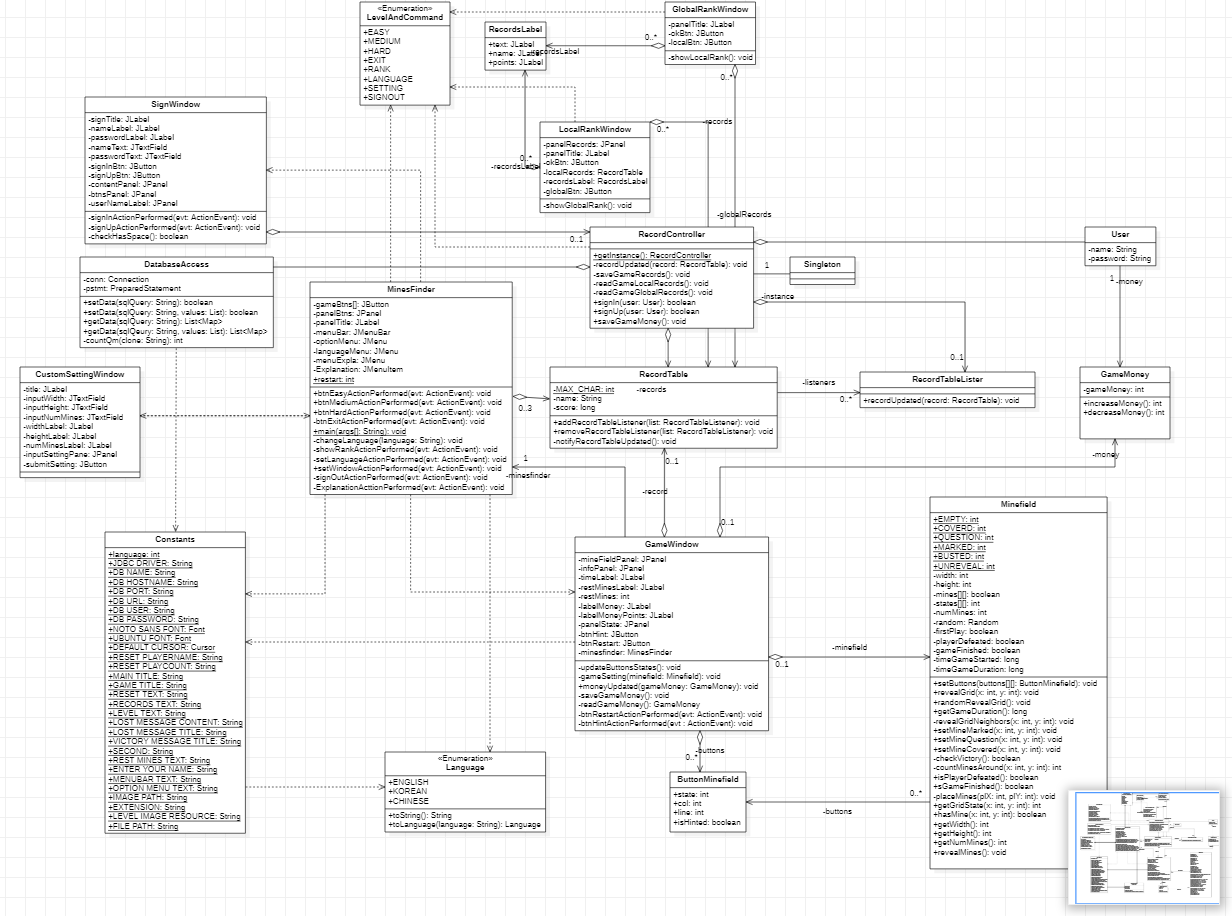
**2.1 개선 아이디어 정리 및 난이도**

|  |  |  |  |
| --- | --- | --- | --- |
| 아이디어 | | 난이도 | 완료 |
| 1. 게임 설명에 대한 메뉴를 추가한다. | | 0.5 | O |
| 2. 이미 밝혀진 버튼을 다시 한번 누르면 그 버튼 주변에 아직 밝혀지지 않은 버튼들을 어둡게 표시해준다. | | 0.5 | X |
| 3. 게임 도중에 경과시간과 남은 지뢰 개수를 보여주는 상태 바를 만든다. | | 1 | O |
| 4. 버튼 개수나 지뢰개수를 사용자가 설정할 수 있도록 만든다. | | 1 | O |
| 5. 지뢰가 터지면 효과음이 나올 수 있도록 만든다. | | 1 | O |
| 6. 게임 머니 기능을 추가하여 게임 성공 시 받을 수 있게 한다.(단, 충전기능은 없다.) | | 1 | O |
| 7. 힌트 기능을 추가하여 게임 머니로 힌트를 쓸 수 있게 한다.  (힌트 버튼을 누르면 랜덤으로 지뢰가 심어지지 않은 곳을 알려준다.) | | 1 | O |
| 8. 자신의 기록을 따로 볼 수 있는 메뉴를 만든다. | | 2.5 | O |
| 9. 사용자가 언어를 설정할 수 있도록 한다. | | 2.5 | O |
| 10. 글로벌 랭킹 기능을 추가한다. | | 5 | O |
| 11. 로그인 기능을 추가한다. | | 2.5 | O |
| 12. 게임이 끝나면 모든 지뢰를 보여주는 기능을 추가한다. | | 1 | O |
| 13. 게임 중에 다시 시작할 수 있는 기능을 추가한다. | | 1 | O |
| 총 난이도 점수 | 구현 점수 | | |
| 20.5점 | 20점 | | |

**3. 분석 및 개선 다이어그램**

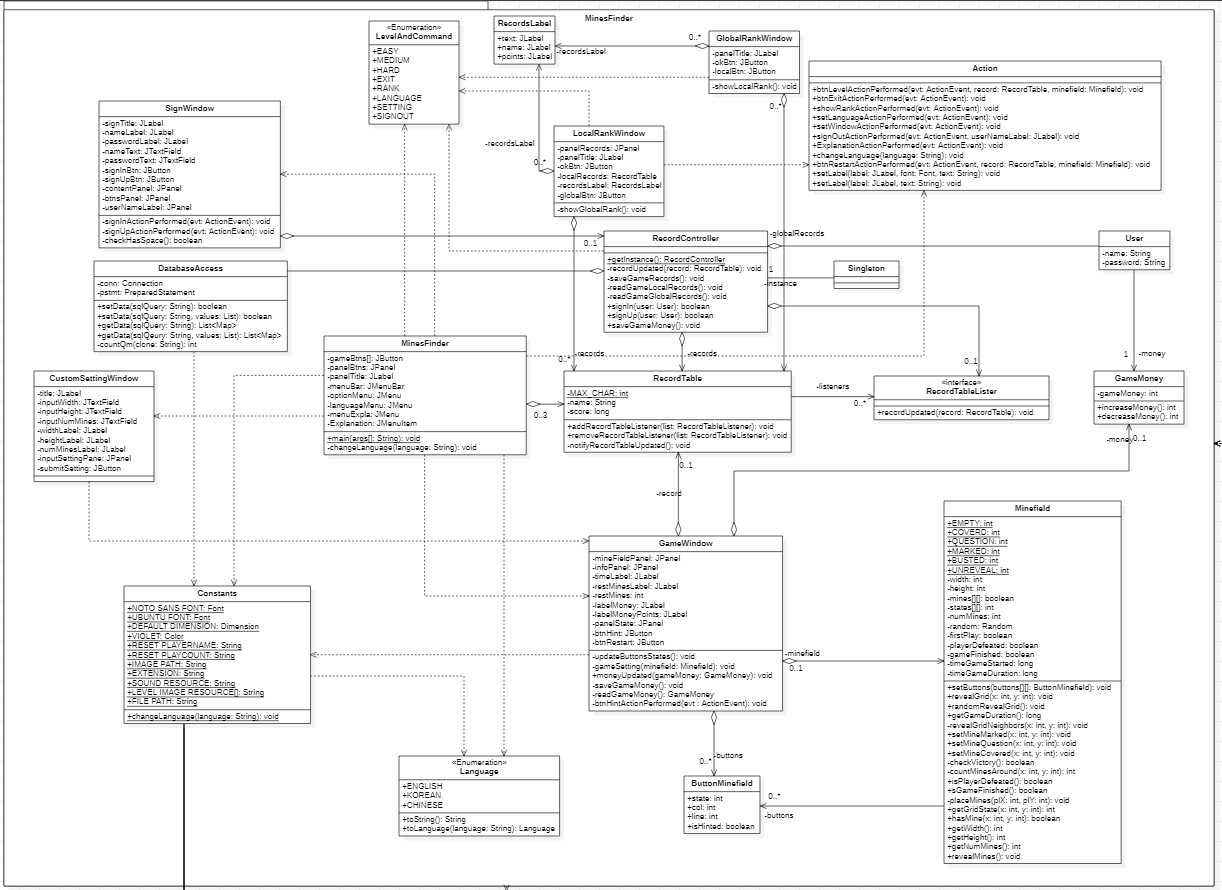
**3.1 기존 코드 분석 Class Diagram**

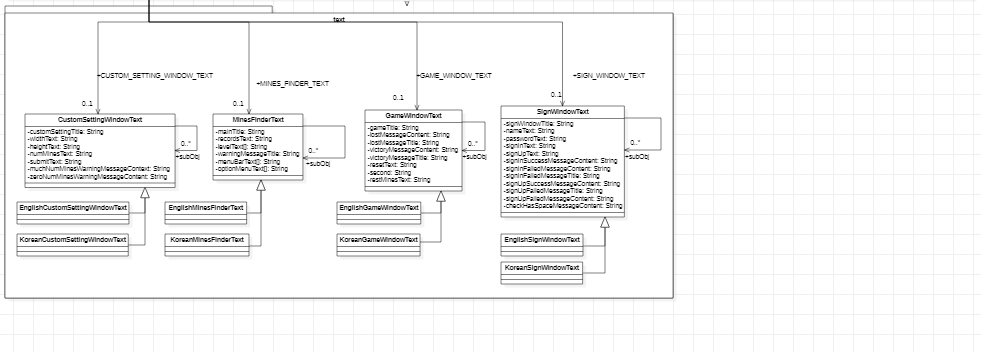
**3.2 개선 후 코드 분석 Class Diagram(리팩토링 전 ClassDiagram)**

****

**‘**

**3.3 리팩토링 후 코드 분석 Class Diagram**

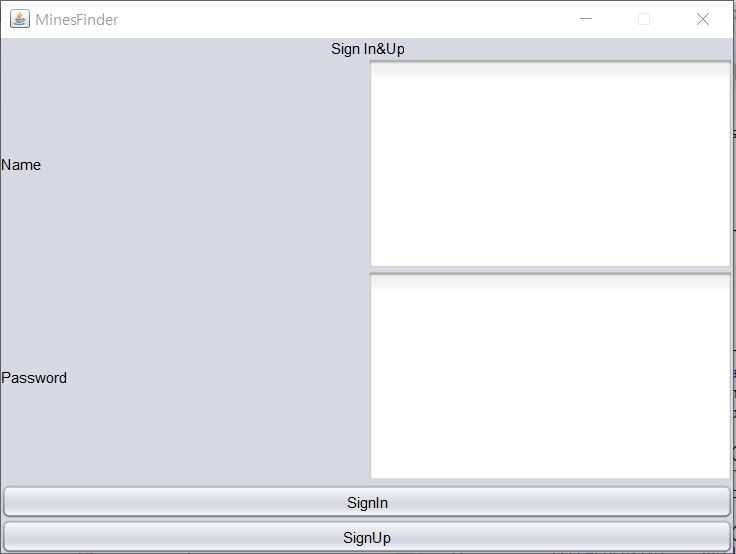


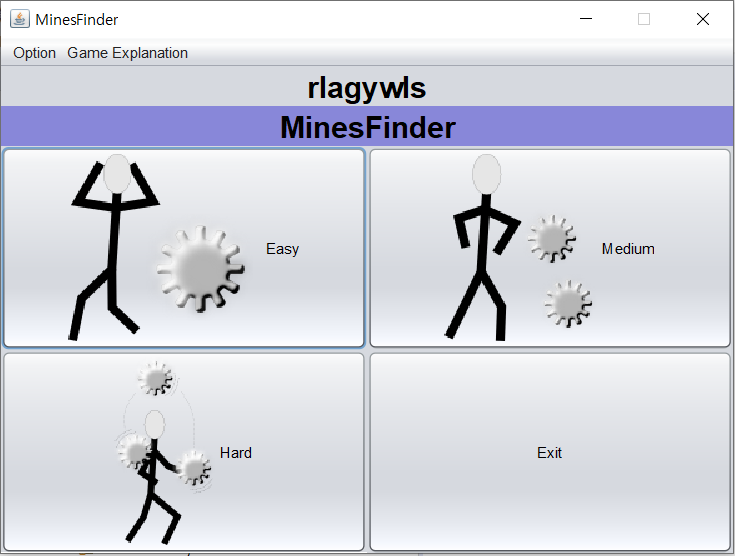
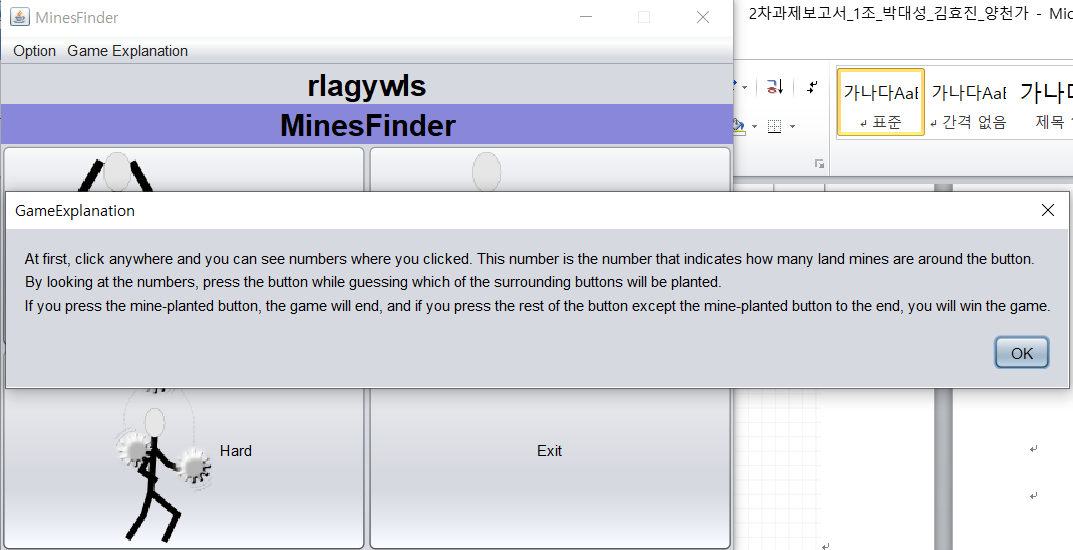


**4. 각 과제 내용에 대한 구현 방법 설명 및 Code Snippet**

**4.1 전체 화면 Capture**

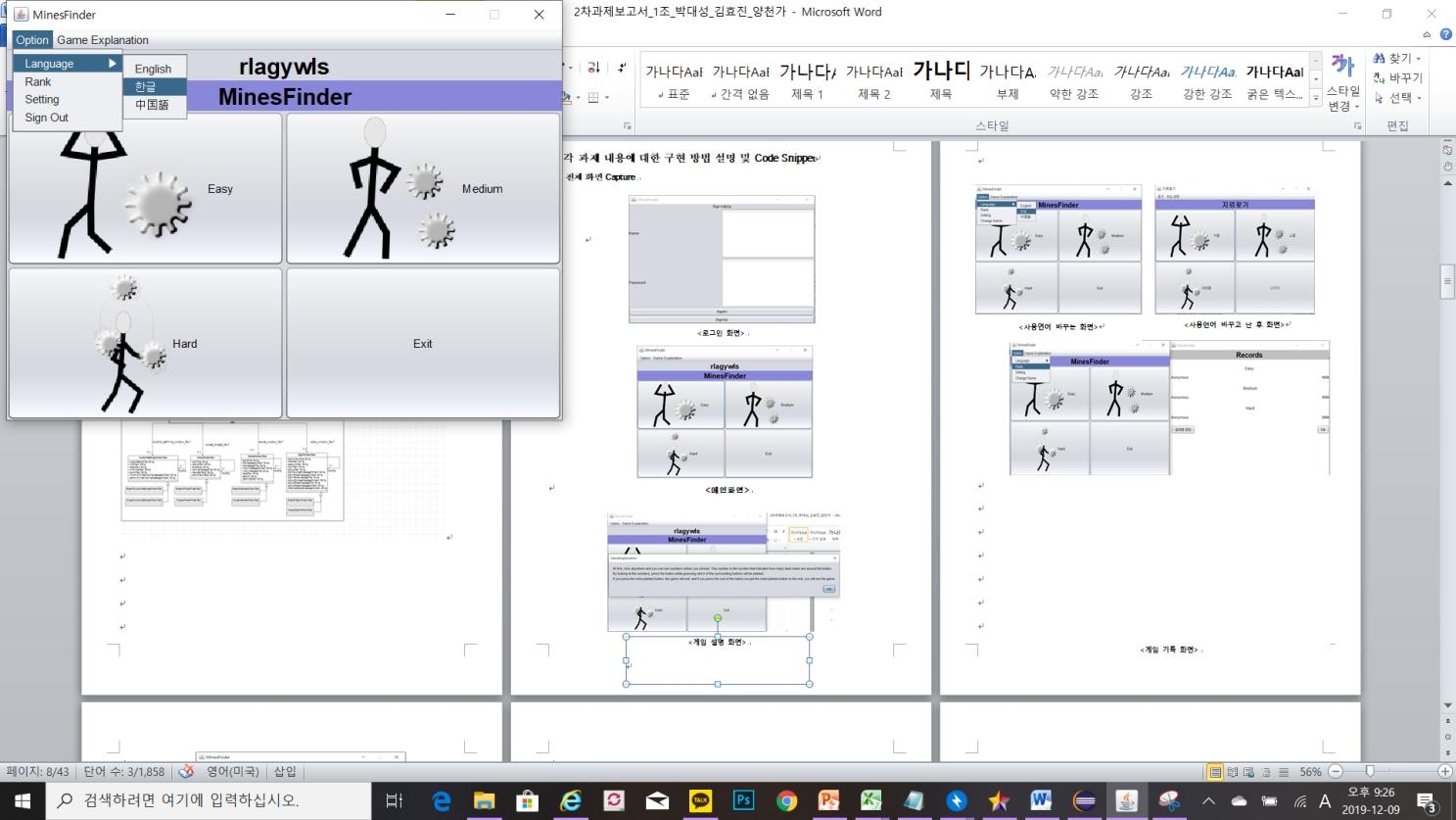
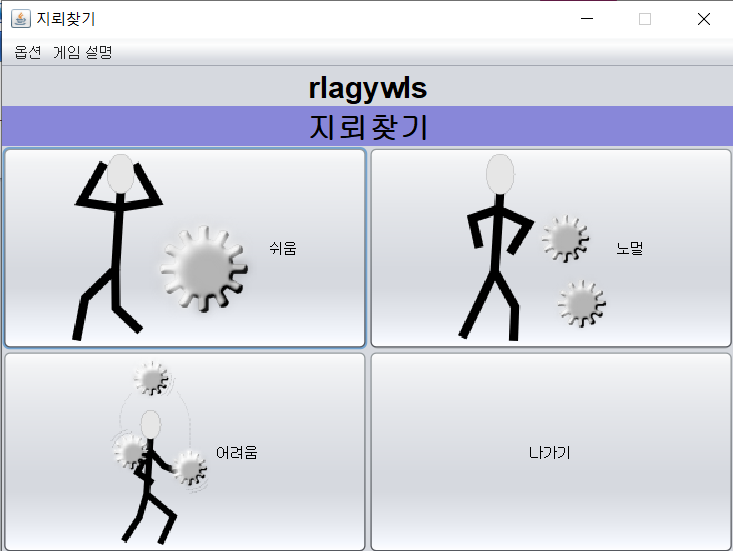
<로그인 화면>





<메인화면>

**<게임 설명 화면>**

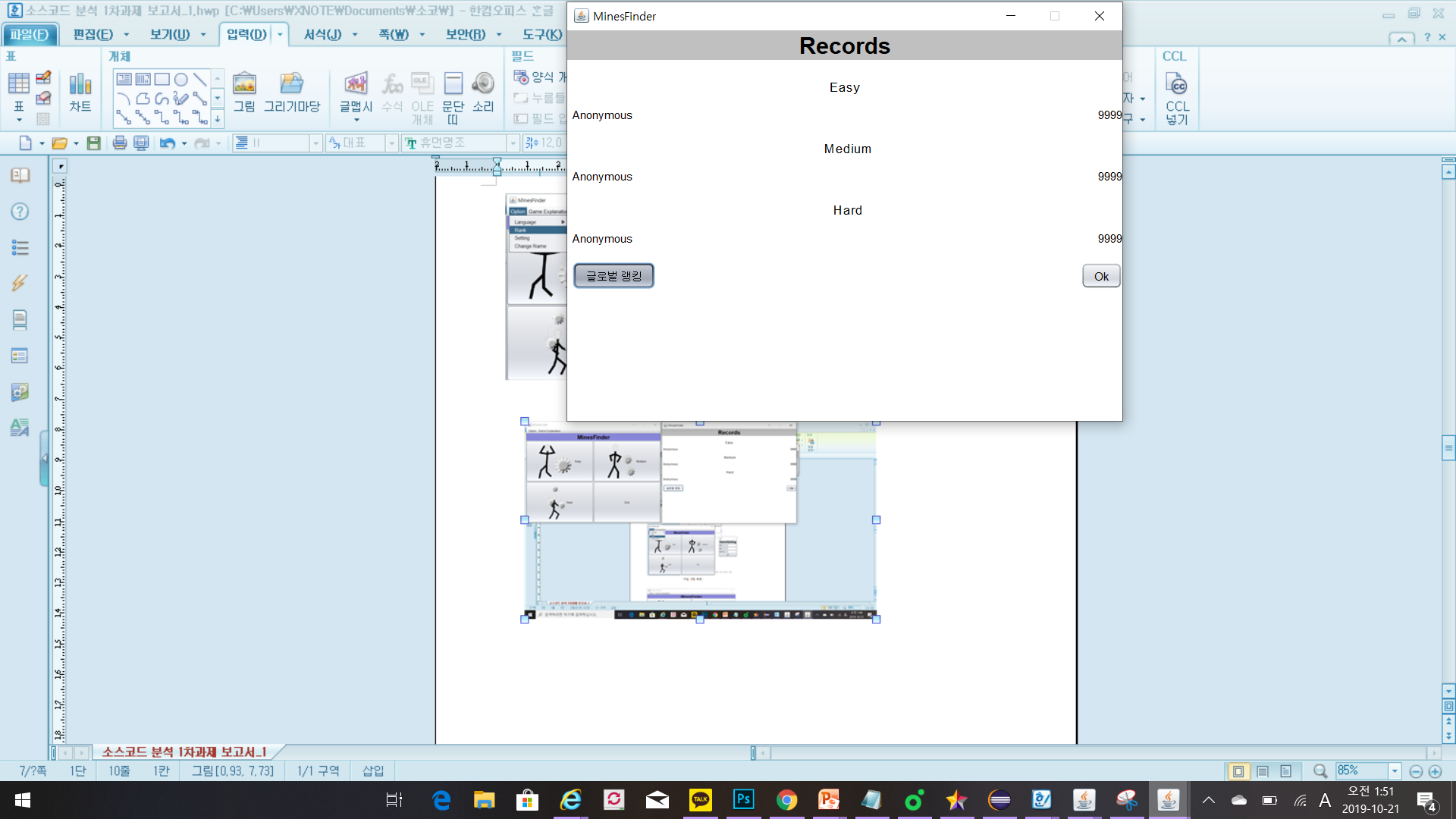


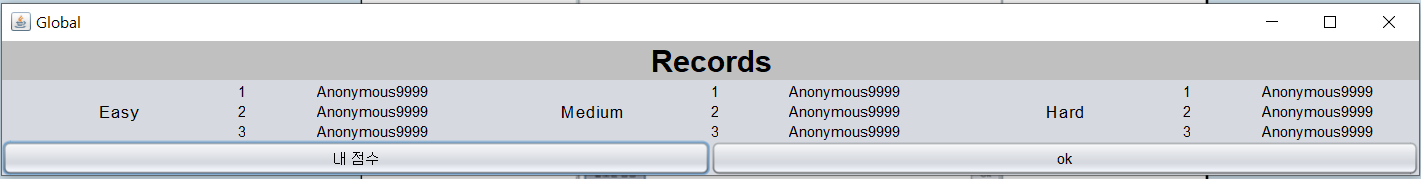
<사용언어 바꾸고 난 후 화면>

<사용연어 바꾸는 화면>

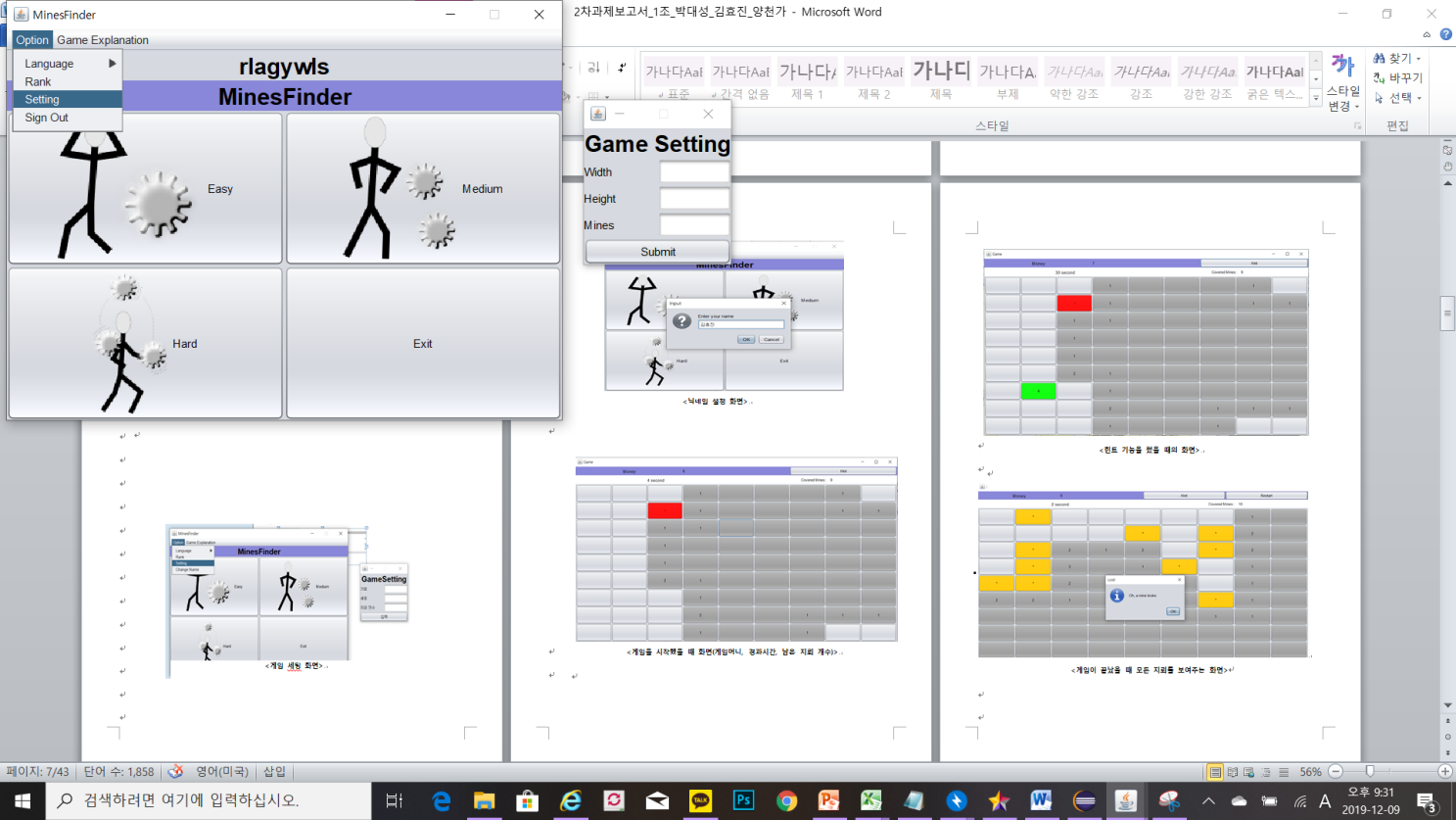
****

**<게임 기록 화면>**

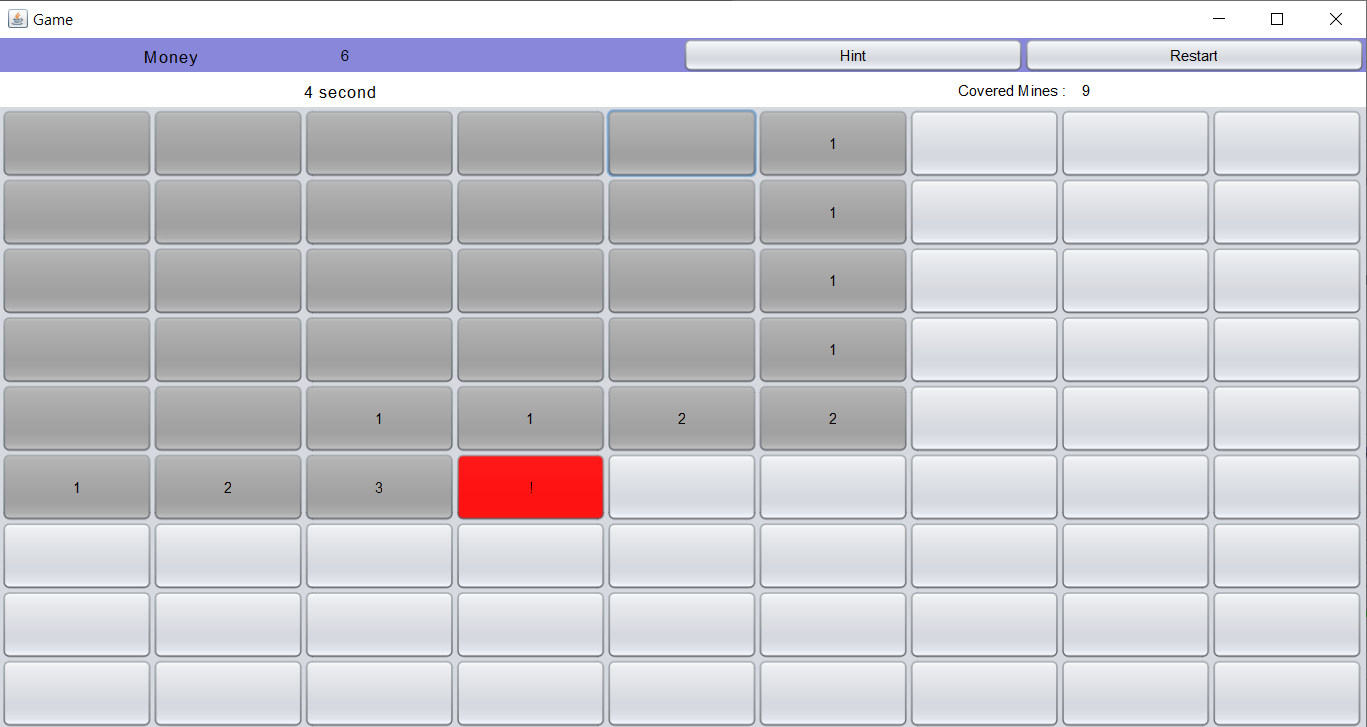
****

****

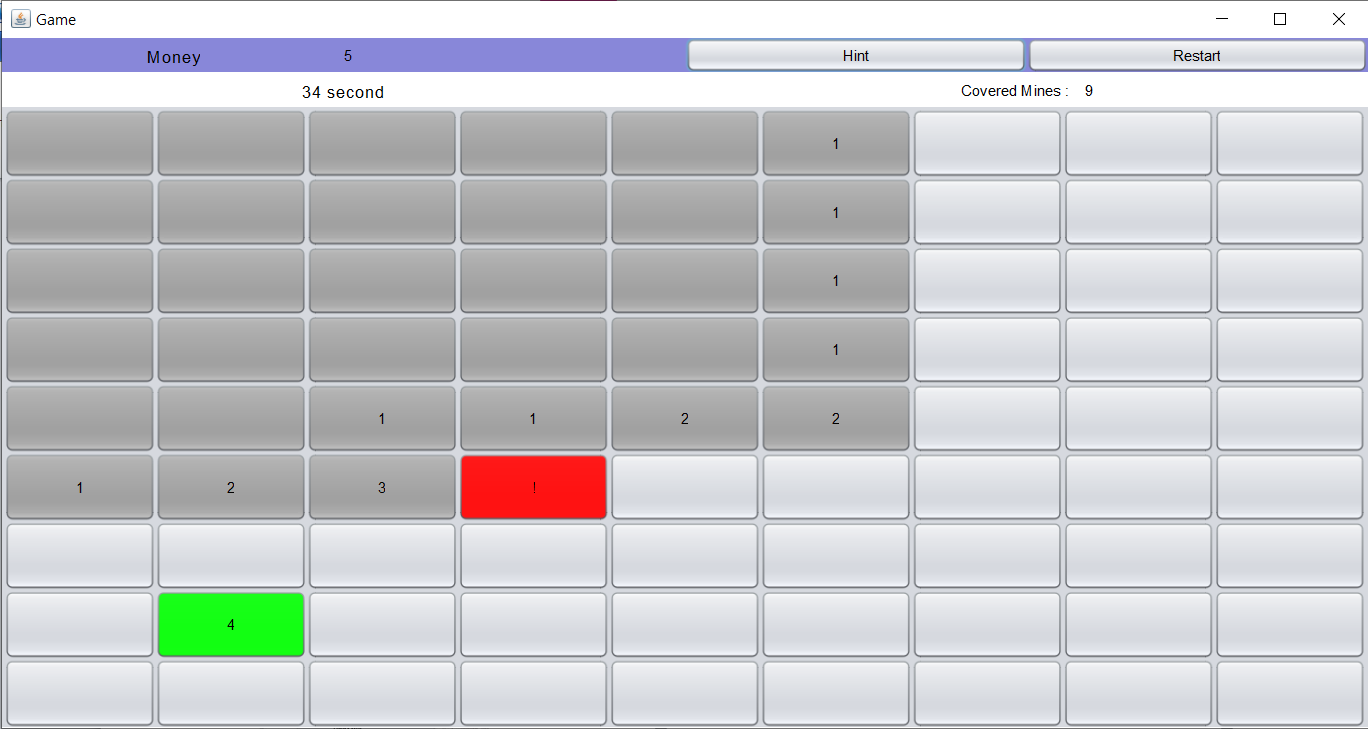
**<글로벌 랭킹 화면>**

****

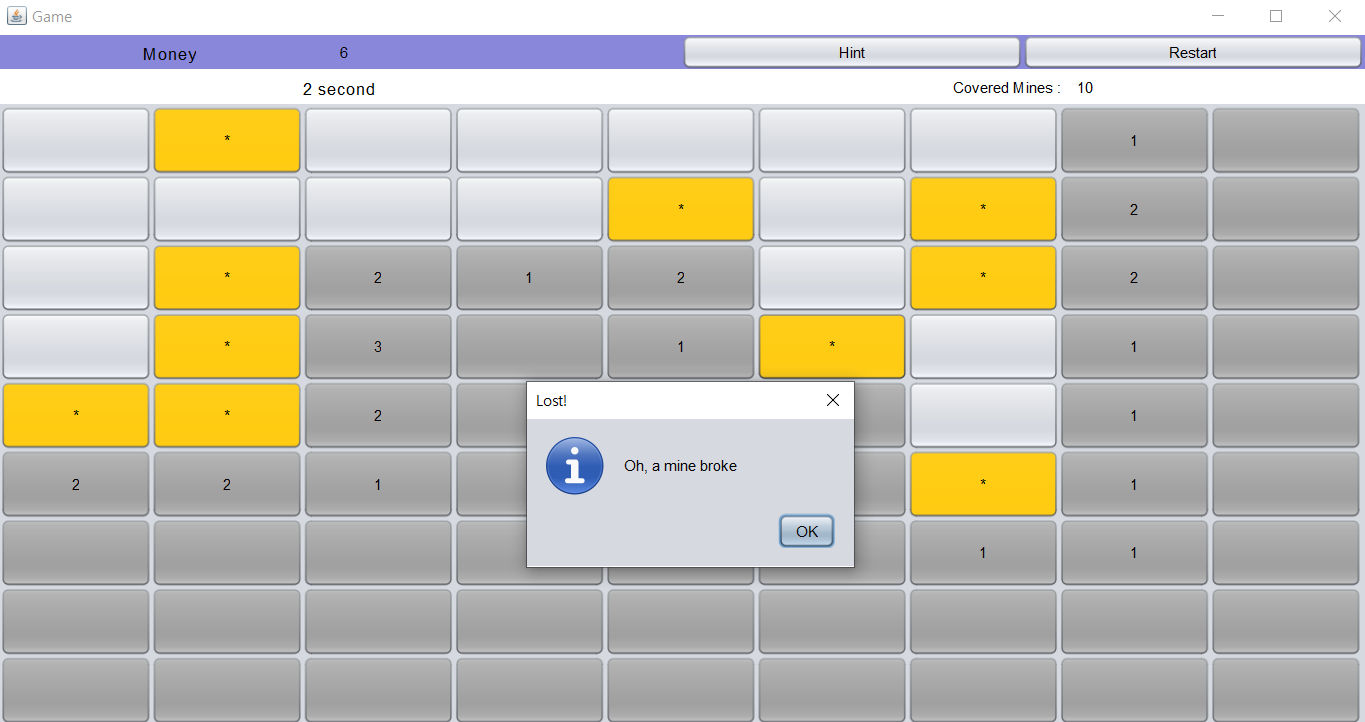
**<게임 세팅 화면>**



**<게임을 시작했을 때 화면(게임머니, 경과시간, 남은 지뢰 개수)>**



**<힌트 기능을 썼을 때의 화면>**



<게임이 끝났을 때 모든 지뢰를 보여주는 화면>

**4.2 각 개선 아이디어 정리**

**4.2.1 아이디어 1 - 게임 설명에 대한 메뉴 추가 (난이도 0.5)**

|  |  |
| --- | --- |
| 개선 코드 | 구현 화면 |
| EMB00006bf047a8EMB00006bf047a9EMB00006bf047aaEMB00006bf047ab | EMB00006bf047acEMB00006bf0479e |
| 개선 설명 | |
| 메뉴바에 게임설명 라벨을 넣고 게임설명을 누르면 addActionListener(new java.awt.eventActionListener)와 ExplanationActionLister(java.awt.event.ActionEvent evt)를 통해 게임에 대해 설명해주는 윈도우 창이  새로 생기도록 만들었다. | |

**4.2.2 아이디어 2 - 이미 밝혀진 버튼을 다시 한번 누르면 그 버튼 주변에**

**밝혀지지 않은 버튼들을 어둡게 표시해주는 기능 (난이도 0.5)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047adEMB00006bf047afEMB00006bf047ae |

|  |
| --- |
| 코드 설명 |
| revealGrid함수에서 unRevealMarked함수를 호출하게 하여 이미 밝혀진 버튼을 또 누르면 unRevealMarked함수에서 누른 버튼 주변에 밝혀지지 않은 버튼만 어두운 회색으로 나타나도록 구현 해봤는데. 위에 구현 화면에서 보는 것처럼 누른 버튼 주변에 13이 뜨게 된다. 13은 MARKED,COVERED처럼 final 상수 13으로 설정해서 나타나게 된 것이다. 에러는 뜨지 않는다. |

|  |
| --- |
| 구현 화면 |
| EMB00006bf047b0 |

**4.2.3 아이디어 3 - 게임 도중에 경과시간과 남은 지뢰 개수를 보여주는 상태바 추가 (난이도 1)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047b3  EMB00006bf047b6  EMB00006bf047bbEMB00006bf047bc |

|  |
| --- |
| 코드 설명 |
| 게임 윈도우에 상태바 패널을 만들고 남은 지뢰개수와 경과시간 라벨을 넣은 다음 mousePressed함수에서 사용자가 지뢰가 있는 곳을 표시할 때마다 지뢰개수를 1씩 감소시켜 나타내주고 run함수를 통해  경과시간을 나타낼 수 있게 했다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.4 아이디어 4 - 버튼 개수나 지뢰개수를 사용자가 설정하는 기능**

**(난이도 1)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047bd |

|  |
| --- |
| 코드 설명 |
| 게임 윈도우 클래스처럼 지뢰 개수와 버튼 개수를 세팅해주는 customSettingWindow클래스를 따로 만들어 사용자의 입력을 받아 지뢰와 버튼 개수를 설정할 수 있도록 하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.5 아이디어 5 - 지뢰가 터지면 효과음이 나오는 기능(난이도 1)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047be  EMB00006bf047bf |

|  |
| --- |
| 코드 설명 |
| 외부에서 효과음 wav파일을 가져와서 리소스 한 다음 GameWindow클래스의 actionPerformed함수에서  지뢰가 심어진 버튼을 누르면 게임이 종료됨과 동시에 효과음이 사용자가 들을 수 있도록 하였다. |

**4.2.6 아이디어 6 - 게임 머니 기능을 추가하여 게임 성공 시 받을 수 있는 기능**

**(난이도 1)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047c0EMB00006bf047c1EMB00006bf047c2 |

|  |
| --- |
| 코드 설명 |
| GameMoney 클래스를 생성하고 게임머니를 감소, 증가 , get하는 함수를 만든다. GameWindow클래스의 actionPerformed함수에서 게임을 승리하면 게임 머니를 증가시키고 업데이트를 시키는 함수인 moneyUpdated함수를 넣는다. 그리고 moneyUpdated함수에선 saveGameMoney함수를 통해 증가된 게임 머니를 저장할 수 있도록 하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.7 아이디어 7 - 게임머니를 사용하여 지뢰가 심어지지 않은 곳을 랜덤으로 알려 주는 힌트 기능 추가 (난이도 1)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047c3EMB00006bf047c4 |

|  |
| --- |
| 코드 설명 |
| GameWindow 클래스에서 힌트 버튼을 누르면 actionPerformed함수를 통해 btnHintActionPerformed함수를 실행되고 이 함수에서 게임머니가 1개 이상일 때 밝혀지지 않고 지뢰가 심어져 있지 않은 버튼들 중 하나를 랜덤으로 보여주고 GameMoney클래스의 decreaseMoney함수를 불러와 게임 머니를 1만큼 감소시키고 저장할 수 있도록 하였다. 힌트로 밝혀진 버튼은 setState함수에서 초록색으로 표시되도록 하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.8 아이디어 8 - 자신의 기록을 따로 볼 수 있는 메뉴 추가 (난이도 2.5)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047c5 |

|  |
| --- |
| 코드 설명 |
| 따로 LocaRankWindow클래스를 만들어서 메인화면에서 Rank버튼을 누르면 새로운 창이 나타나고 그 창에서 자신의 기록을 따로 볼 수 있도록 만들었다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.9 아이디어 9 - 사용자가 언어를 설정할 수 있는 기능 추가(난이도 2.5)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047c6  EMB00006bf047c7EMB00006bf047c8 |

|  |
| --- |
| 코드 설명 |
| 따로 Language 목록을 만들고 메인화면에서 Language 메뉴를 통해 사용언어를 바꾸는 버튼을 누르면 changeLanguage함수를 통해 사용언어가 바뀔 수 있도록 하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.10 아이디어 10 - 글로벌 랭킹 기능 추가 (난이도 5)**

|  |
| --- |
| 개선 코드 |
| EMB00006bf047caEMB00006bf047c9 |

|  |
| --- |
| EMB00006bf047ccEMB00006bf047cb |

|  |
| --- |
| EMB00006bf047cdEMB00006bf047ce |

|  |
| --- |
| 코드 설명 |
| DataAccess클래스에서 list<map>으로 data를 세팅하거나 가져오는 함수를 구현하고 RecordController함수에서는 saveGameRedcords,readGameLocalRecords,ReadGameGlobalRecords함수를 통해 기록을 파일에 저장하고 자신의 기록을 읽어서 getData함수를 통해 가져온 글로벌 랭킹 기록들을 자신의 기록과 비교하여 랭킹에 등록할 수 있도록 하였다. |

|  |
| --- |
| 구현 화면 |
| EMB00006bf047a0  EMB00006bf047a3 |

**4.2.11 아이디어 11 – 로그인 기능 추가(난이도 2.5)**

|  |
| --- |
| 개선 코드 |
|  |

|  |
| --- |
| 코드 설명 |
| signIn함수에서 query문으로 입력받은 user정보를 가져와서 RecordController객체의 user변수에 저장한다.  signUp함수에서 query문으로 입력받은 user가 이미 존재하는지 검사하고 없으면 입력받은 정보를 데이터베이스에 저장한다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.12 아이디어 12 – 게임 종료 시 모든 지뢰 보여주는 기능 추가(난이도 1)**

|  |
| --- |
| 개선 코드 |
|  |

|  |
| --- |
| 코드 설명 |
| MinesField클래스에서 모든 지뢰를 표시해주는 함수(revealMines)을 만들고 GameWindow에서 게임이 끝나면 실행되는 if문에서 revealMines()함수를 호출하여 게임 종료 시 모든 지뢰를 보여줄 수 있도록 구현하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**4.2.13 아이디어 13 – 게임 중 다시 시작하는 기능 추가(난이도 1)**

|  |
| --- |
| 개선 코드 |
|  |

|  |
| --- |
| 코드 설명 |
| GameWindow클래스에 restart버튼을 추가하고 restart버튼을 처리하는 함수에서 minefield를 새로 만들 수 있도록 하여 다시 시작할 수 있는 기능을 구현하였다. |

|  |
| --- |
| 구현 화면 |
|  |

**5. 리팩토링(Refactoring)**

**5.1 리팩토링한 내용 정리**

**리팩토링은 stan4j에서 Composition과 Violations부분을 참고하면서 Long Method, Large Class, Duplicated Code, visibility을 고려하여 진행하였다.**

**5.1.1 Long Method-gamesetting**

**Stan4j를 실행해보니 Violations에서 함수가 길다는 부분이 많았다. 그래서 함수를 부분적으로 나누어**

**함수의 길이를 줄였다. 아래 그림은 GameWindow 클래스 gamesetting 함수의 길이를 줄인 부분이다.**

****

mouseAction(), keyAction(), action()함수를 따로 만들어 주고

.

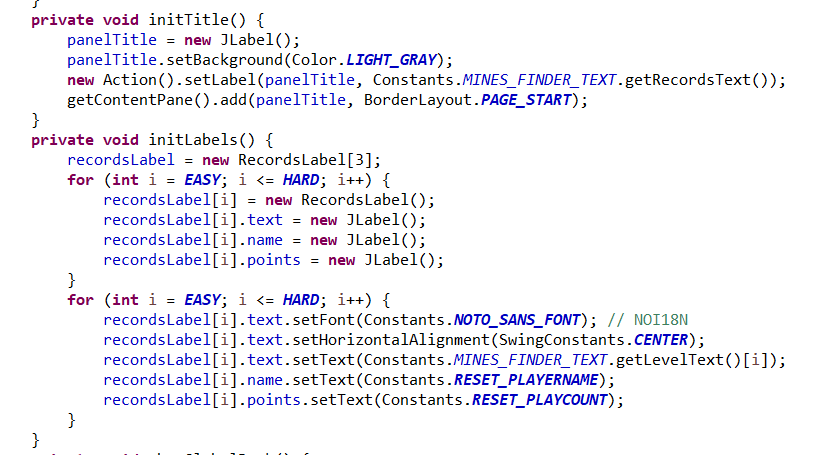


gamesetting()함수에서는

매개변수 e로 전달받아 이 함수들을 실행할 수 있도록 하였다

**5.1.2 Long Method-initComponents**

**위와 같은 내용을 MinesFinder클래스에 initComponents에도 적용했다.**

****

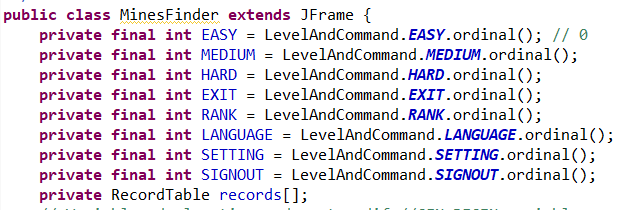
initTitle(), initLabels()함수를 따로 만들어 준 뒤

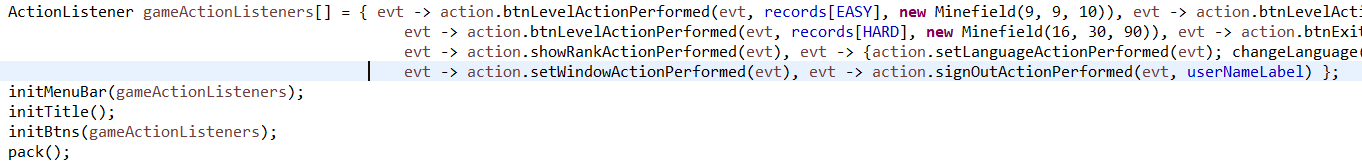


initComponents()에서는 해당 함수를 호출하도록 해주어 함수의 길이를 줄였다.

**5.1.3 Long Method-ActionListener**

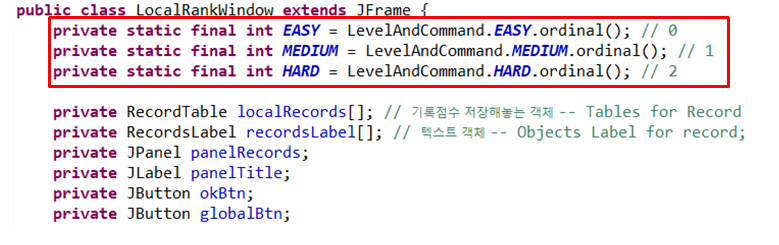
**ordinal()함수로 각 변수에 순서를 매겨 준 다음 ActionListener배열에서 evt매개 변수로 해당 함수들을 차례로 실행할 수 있도록 하여 함수의 길이를 줄였다.**

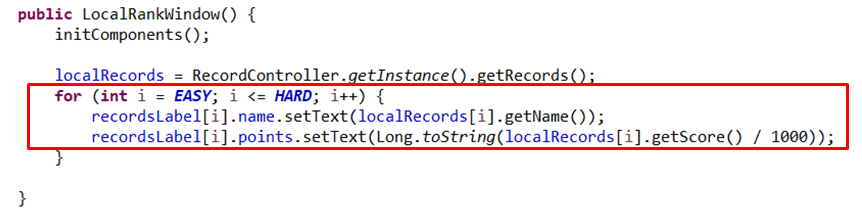
****

****

**5.1.4 Long Method-난이도 별 레코드**

**recordTable을 만들어 내는 곳에도 ordinal()함수로 각 변수에 순서를 매겨 준 다음 for문으로 해당 record들을 차례로 불러올 수 있도록 하여 중복 코드를 제거하고 함수의 길이를 줄였다.**





**5.1.5 Large Class-action**



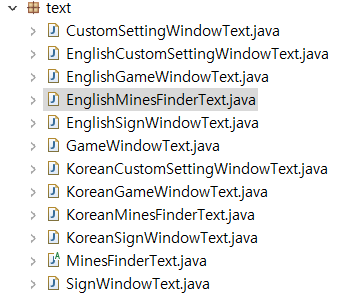
버튼을 처리하는(ActionPerfomred) 함수들을 Action클래스를 만들어 따로 모아놓고



필요한 곳에서 객체 참조로 간단히 써서 클래스를 줄이도록 하였다.

**5.1.6 Large Class-text**

**Text를 다루는 내용들을 따로 패키지에 모아 놓고 쓸 수 있도록 만들었다. MinesFinder, GameWindow,CustomSettingWindowText, SignWindowText에서 text를 다루는 데 그 중 MinesFinder에 대한 내용만 보여 주도록 하겠다.**

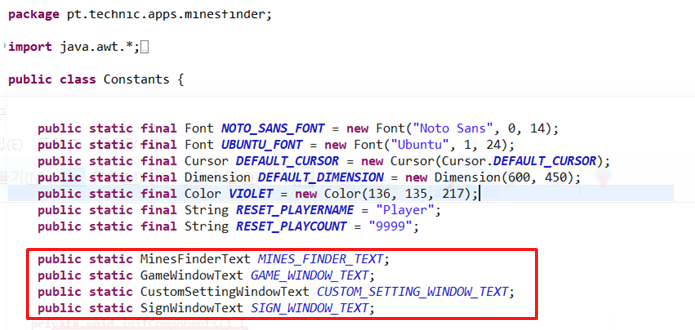
****

text패키지에 있는 MinesFinderText클래스에 text와 관련된 함수들 추상적선언

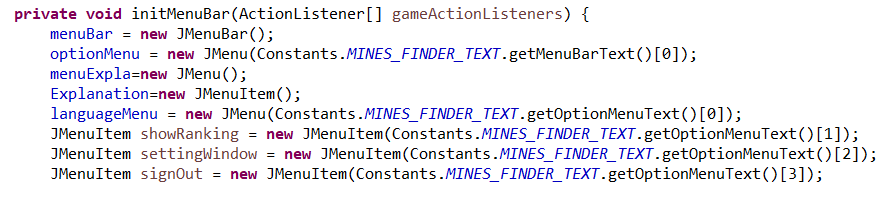
text패키지안에 있는 text 클래스

****

text 패키지에 MinesFinderText를 상속받는 클래스에서 실제로 쓰이는 내용 구현

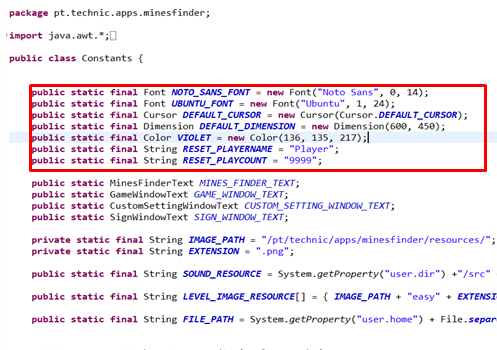


기존 패키지에 Constants클래스에서 text클래스 선언

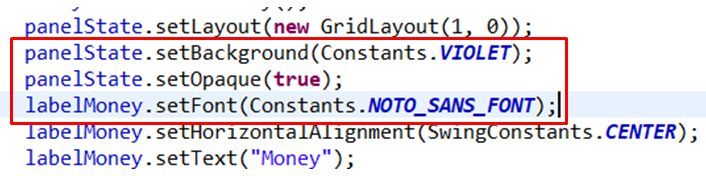
****

MINES\_FINDER\_TEXT로 모든 text내용 간단하게 구현

**5.1.7 Duplicated Code-font, color,dimension,cursor**

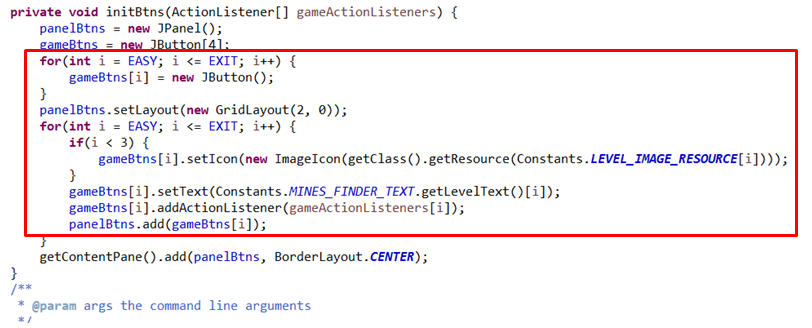


Constants클래스에서 자주 쓰이는 UI를 위해 Font, Color, Dimention, Cursor, 등을 따로 final로 설정하여 중복코드를 제거 하였다.



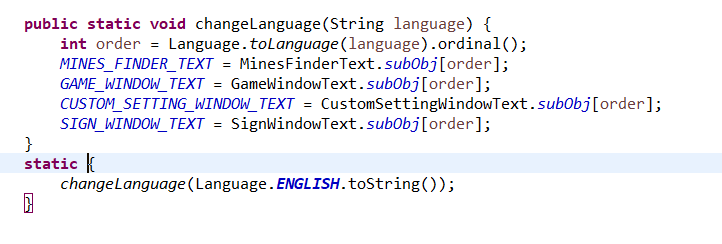
**5.1.8 Duplicated Code-btn 모아 놓기**

**Easy, Medium, Hard 버튼을 처리하는 함수를 각자 따로 불러 오지 않고 반복문을 통해서 불러올 수 있도록 하여 중복코드를 제거하고 함수의 길이도 줄였다.**



**5.1.9 Data Clumps-language**

**개선 기능 중 사용언어를 바꿀 수 있는 기능이 있는데 window에 보이는 언어를 모두 바꿔야 하다 보니 멤버 변수가 매우 많아졌다. 이를 해결하기 위해 Contants클래스에서 원래는 배열로 일일이 언어를 바꿨던 것을 changeLanuage()함수를 통해 객체를 생성하여 사용언어를 바꿀 수 있도록 하였다. 각 Tetxt클래스들은 위에서 언급했듯이 text패키지에 속해 있고 그 안에 text내용이 들어있다.**

****

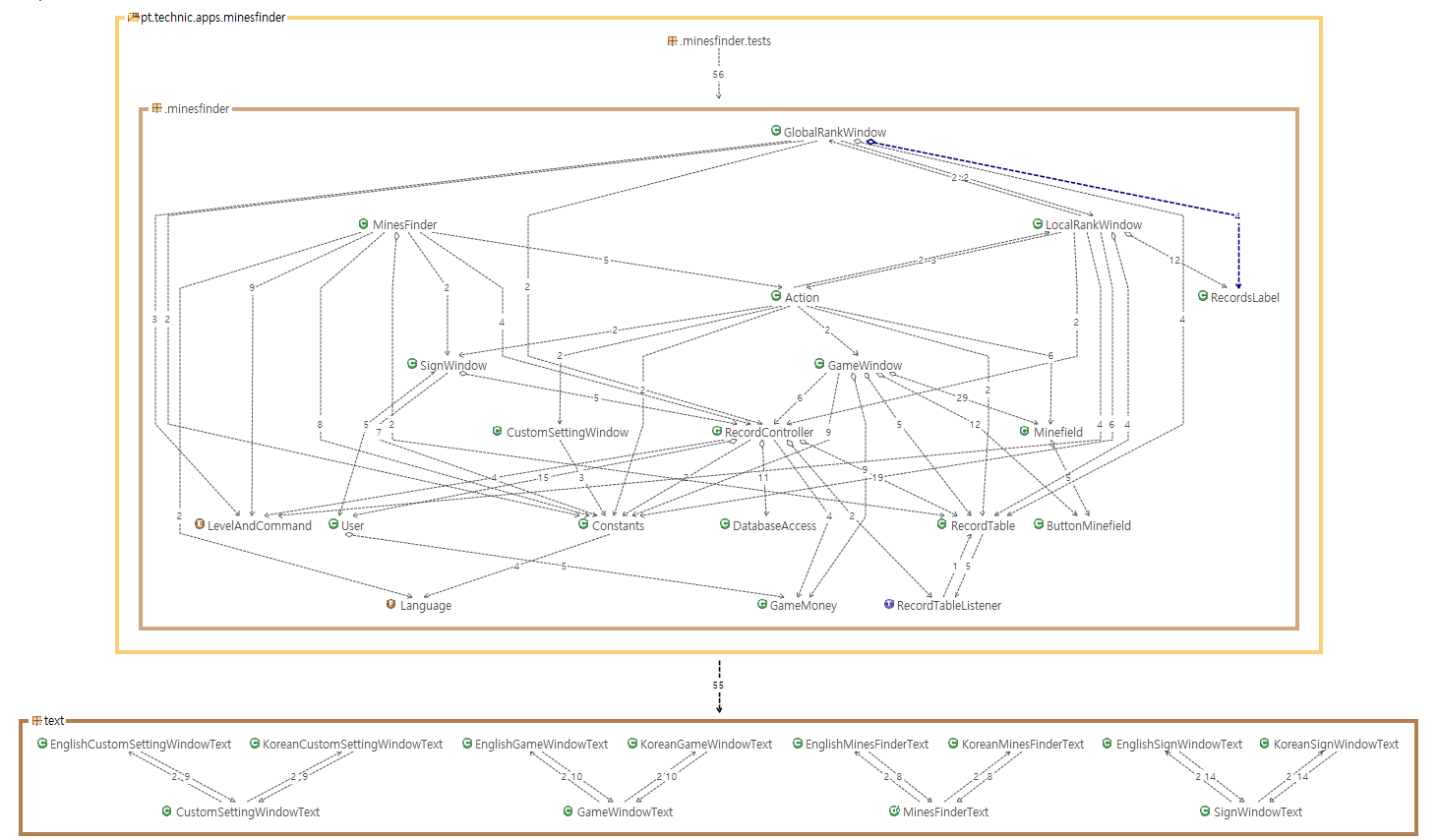
**5.1.10 Visibility**

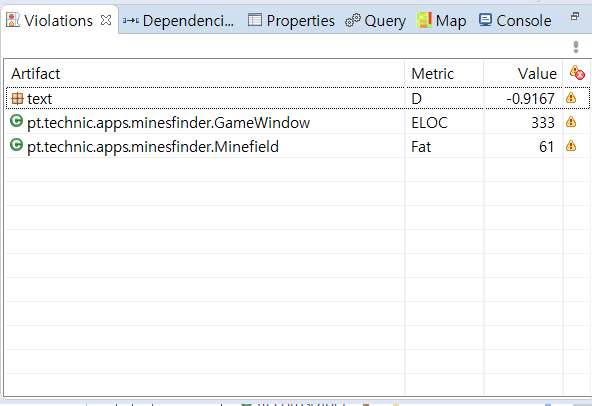
**text패키지에는 text함수를 추상적으로 선언하는 MinesFinderText, GameWindowText, CustomSettingWindowText, SignWindowText, 클래스가 있고 이 클래스들을 각각 상속받아 실제로 text함수를 구현하는 함수들이 있다. 이 함수는 상속받는 함수들 사이에서만 쓰이기 때문에 가시성을 protected를 지정하였다.**

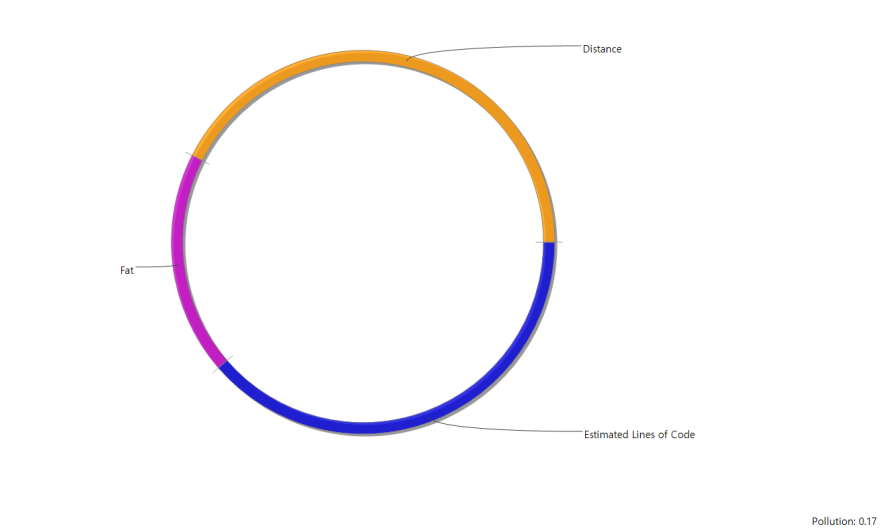


**5.2 리팩토링 후 stan4j 실행 결과**

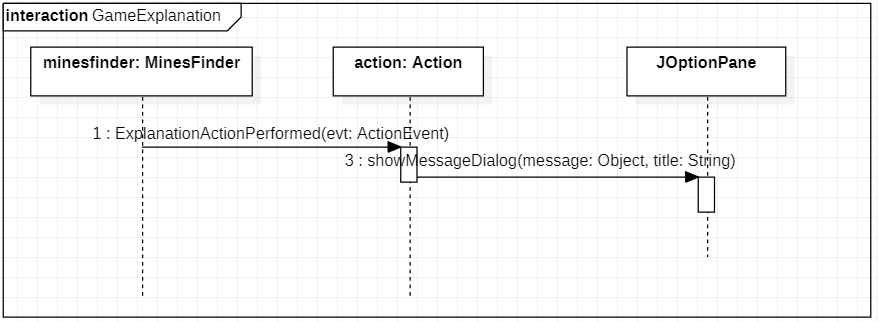
**Stan4j를 처음 실행하였을 때는 Composition도 되게 복잡하였고 Violations부분도 함수가 길다는 내용이 10개 정도 되었는데 리팩토링을 한 후 에는 Composition을 보면 대부분 클래스들 사이에 선들이 아래로만 뻗어나가는 것을 볼 수 있다. Violations에서도 2가지 항목만 남고 나머지는 제거된 것을 볼 수 있다. Pollutiond은 0.96에서 0.17로 변했다.**

****

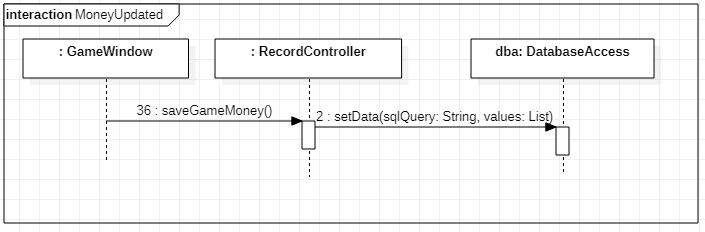
****

****

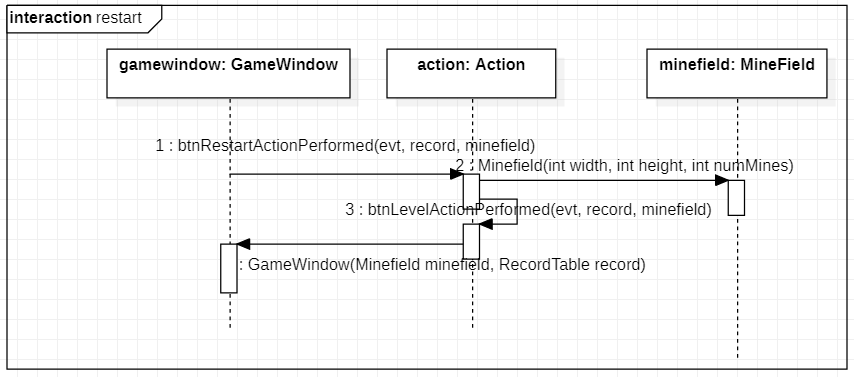
**5.3 리팩토링 후 Sequence Diagram**

****

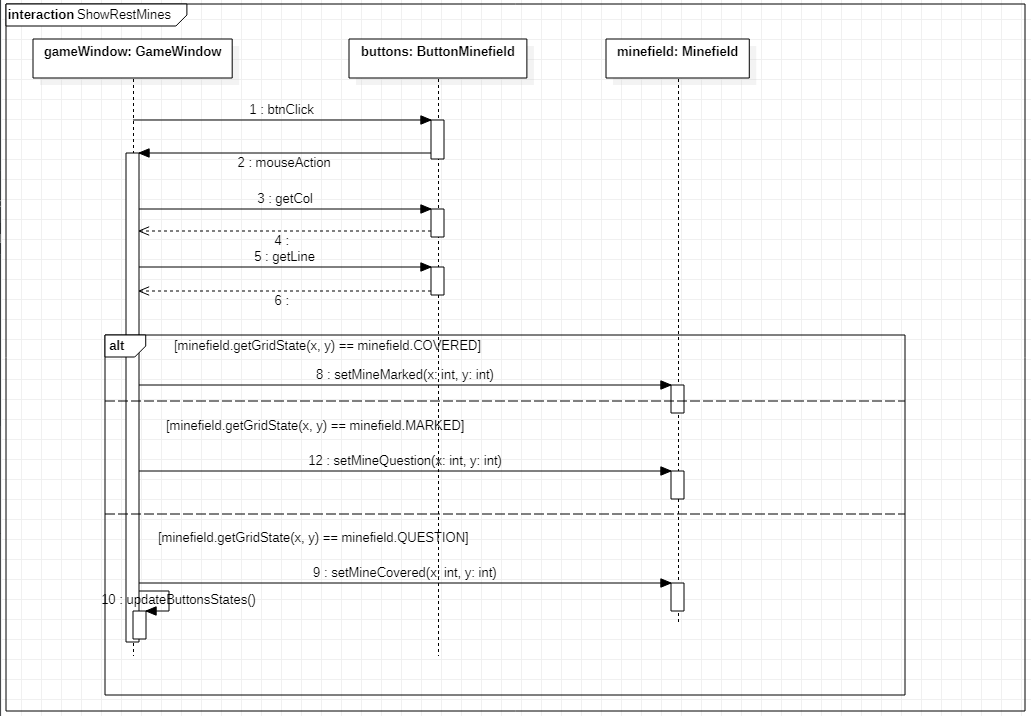
<게임 설명 기능>

****

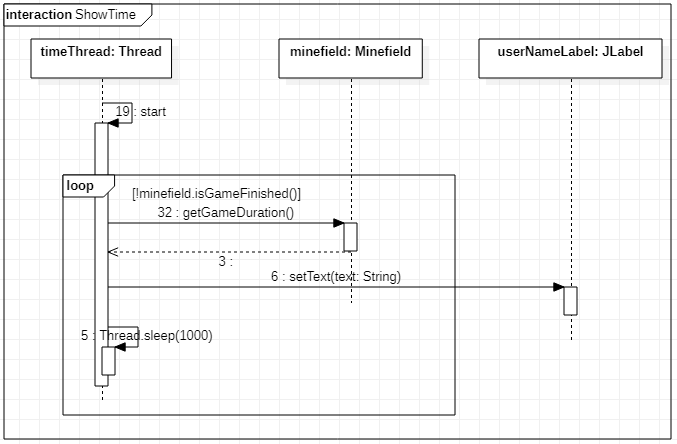
<게임 머니 기능>

****

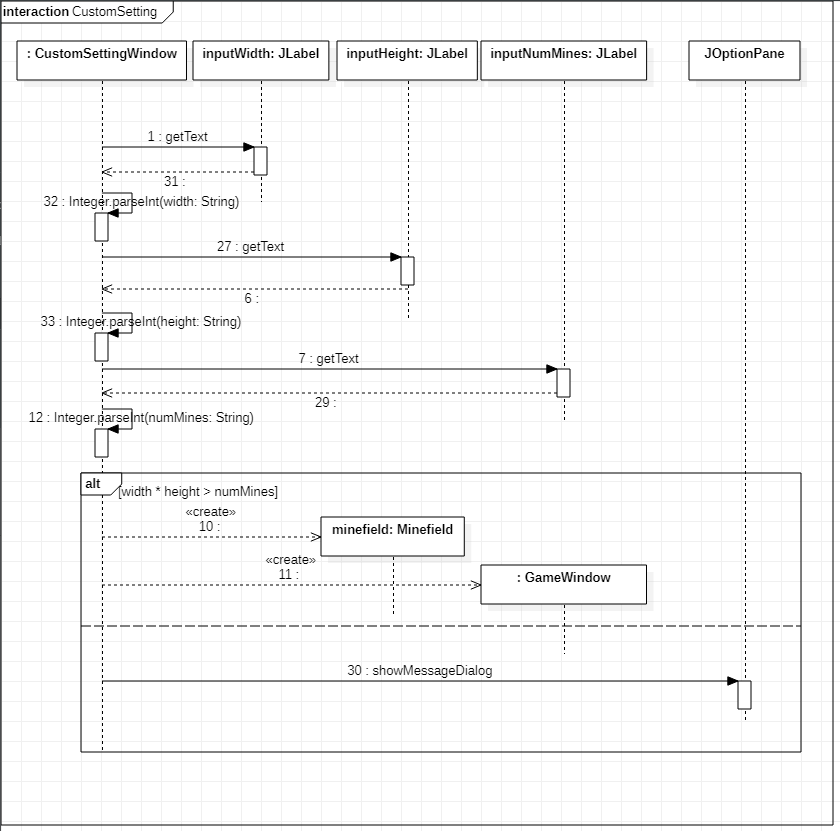
<게임 중 다시하기 기능>

****

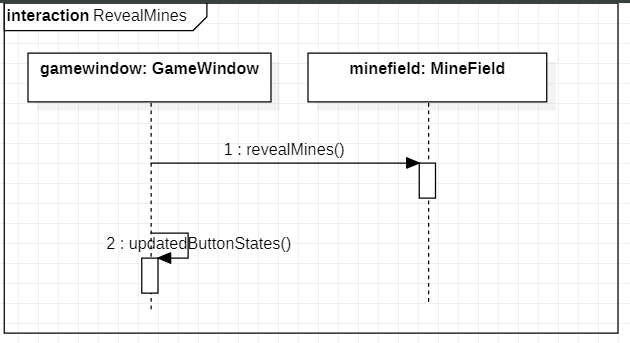
<남은 지뢰 개수 보여주기 기능>

****

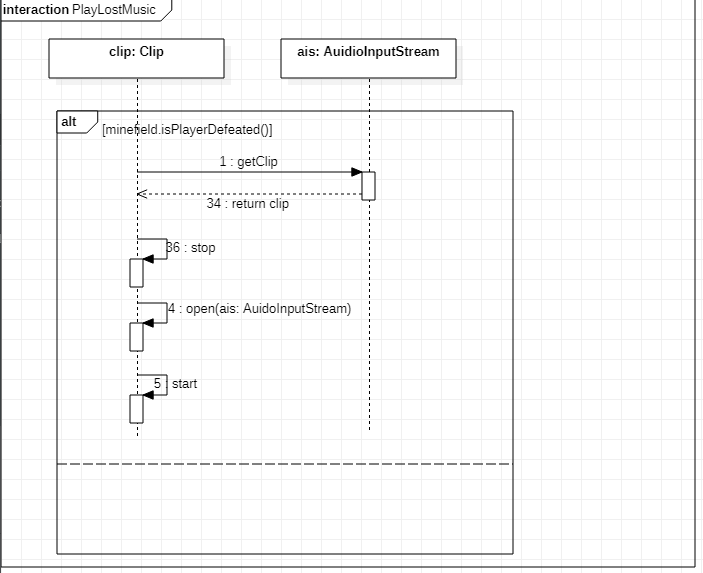
<경과 시간 보여주기 기능>

****

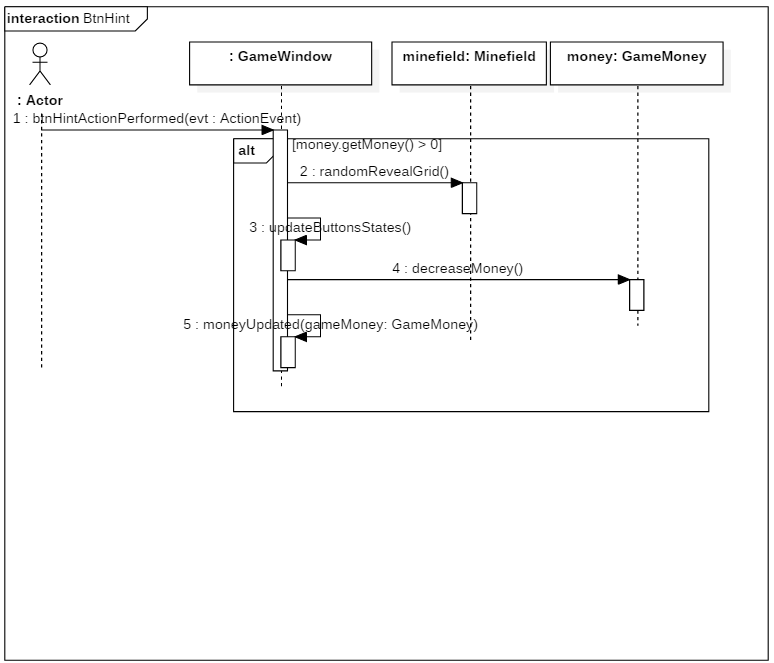
<사용자 설정 기능>

****

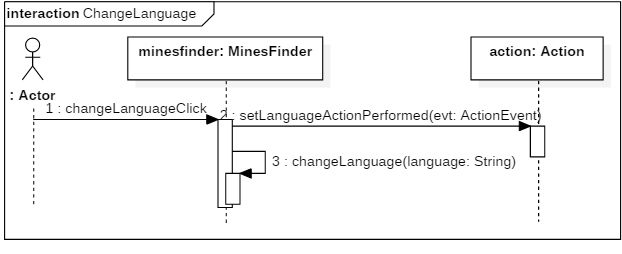
<종료 시 모든 지뢰 보여주기 기능>

****

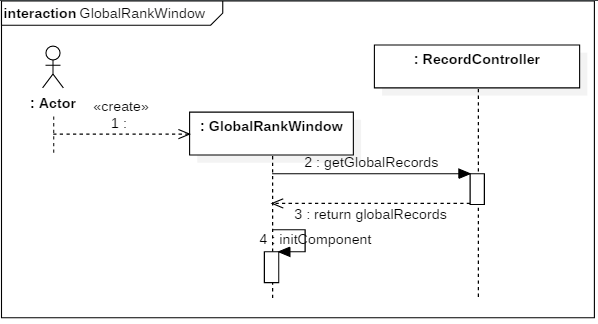
<게임 실패 시 효과음 기능>

****

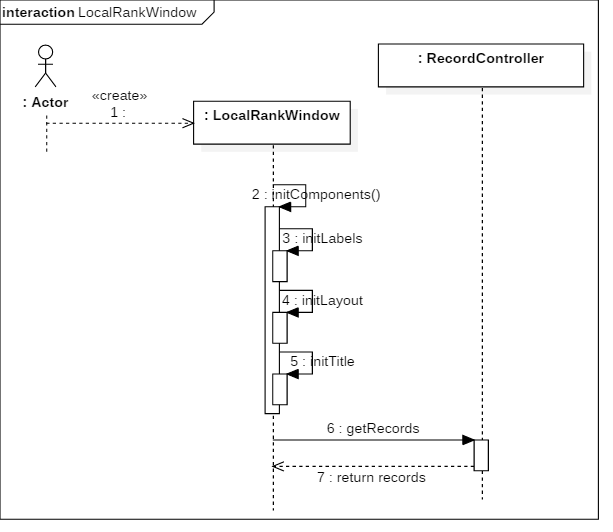
<힌트 기능>

****

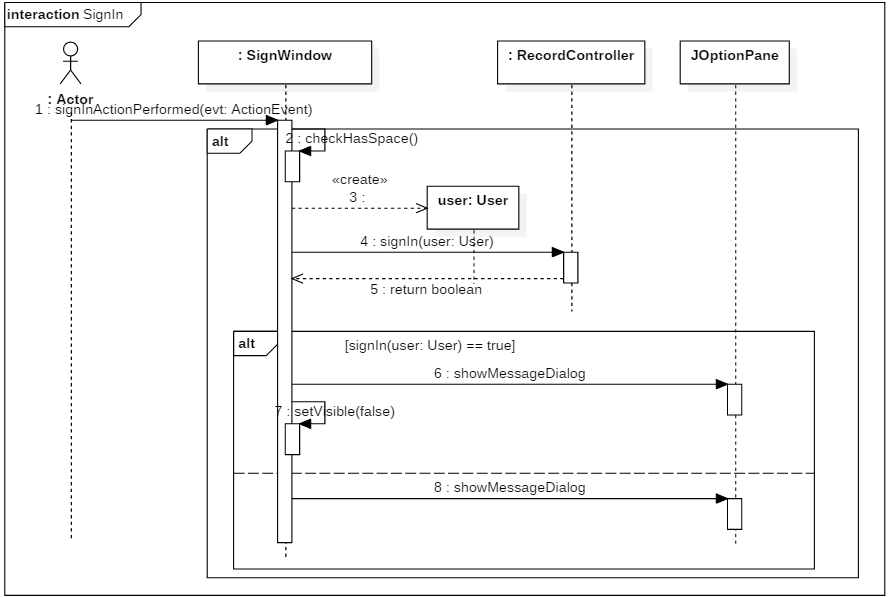
<사용언어 바꾸기 기능>

****

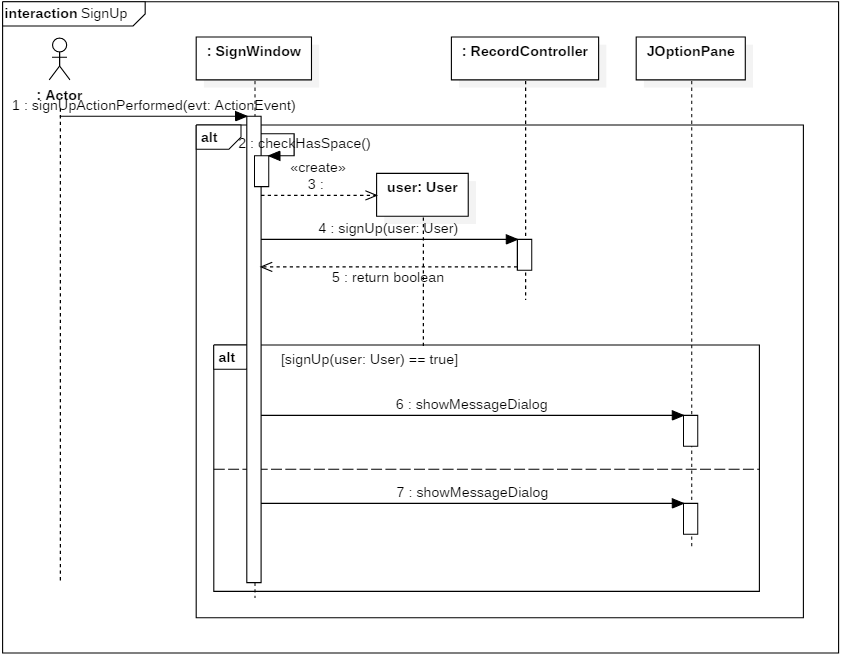
<글로벌 랭킹 기능>

****

<자신의 기록 따로 볼 수 있는 기능>

****

<로그인 기능>

****

<회원가입 기능>

**6. 느낀점**

**박대성 : 자바 swing을 좀 더 쓸 수 있게 돼서 좋았고 실제 프로그램에서 옵저버 패턴(Observer Pattern)이 쓰이는 방식에 대하여 잘 알 수 있었다. Swing이 디자인 패턴이 들어간 클래스가 많은데 디자인 패턴에 대하여 공부할 수 있게 되어 좋았다. 1차과제를 한 내용을 바탕으로 2차과제로 기능 추가 및 리팩토링을 하는데 내가 잘 만들었다고 생각한 코드에서 Stan4J로 분석해보니 Constants클래스에서 멤버변수가 많다고 했다. 내가 직접 이상을 못 느끼더라도 Stan4J를 통해서 보니 내 문제가 뭔지 알게 되었고, 더 유연한 프로그램을 만들 수 있었다. 앞으로도 정적분석툴을 많이 사용해야되겠다는 생각이 들었다.**

**김효진 : 지뢰 찾기 게임을 해본 적이 없어서 처음에 과제로 이 게임이 주어졌을 때 되게 낯설기도 했고 이렇게 큰 프로젝트를 하는 것도 처음이어서 이 과제를 주어진 시간 동안 잘 해결해낼 수 있을지 걱정이 됐었다. 심지어 자바에 대한 지식도 얕다 보니 소스코드를 분석할 때도 하나부터 열까지 인터넷을 보거나 책을 봐야만 이해를 할 수 있었다. 그래서 질문을 만들어 낼 때도 질문할 게 너무 많아서 그 중 하나를 고르기 어려웠다. 게임 개선 아이디어를 낼 때는 구현 생각은 뒤로 미뤄두고 개선되면 좋을 것 같은 아이디어를 생각해냈다. 그래서 인지 나중에 구현하기가 쉽지가 않았다. 결국 하나를 구현해내지 못해서 너무 아쉬웠다. 클래스 다이어그램을 만들면서 클래스 사이에 연관관계를 그리는 과정에서 일반화, 의존, 집합 관계 중 어떤 것으로 그려야 하는지 결정하는 게 어려웠다. 이론으로 배울 때는 쉬웠던 것 같은데 실제 프로그램에 적용하려고 하니까 고려사항이 많아져서 되게 헷갈렸던 것 같다. 2차 과제로 구현을 할 때는 이제 코드가 익숙해져서 그런지 구현이 생각보다 잘되었다. 문제는 리팩토링이었는데 지금껏 열심히 짠 코드를 바꿔야 한다는 게 마음이 아팠다. 괜히 건드렸다가 오류가 많이 생길 것 같아서 걱정도 되었다. 이렇게 막막해서 한 두 시간 멍하니 코드만 보다가 수업시간에 배웠던 내용을 하나씩 생각해보며 비교적 쉬운 것부터 해보니까 Stan4j에 있는 composition, violations에서 문제의 부분들이 하나 둘씩 없어지는 것을 볼 수 있었다. 그리고 교수님 말씀처럼 팀원과 밤을 새워가면서 개발을 하다 보니 묘한 뿌듯함을 느끼기도 했다. 비슷한 아이디어를 가진 다른 팀과도 소통하며 서로 몰랐던 것을 알려주고 배워가는 재미도 있었다. 결과적으로 자바와 개발에 대한 지식을 많이 알게 되어서 좋았다.**

**양천가 :**