자바프로그래밍 파이널리포트

컴퓨터정보공학과

12080912

남건백

자바프로그래밍

오경진 교수님

2012.6.18

**1. 프로그램 개발 환경**

1.1 사용프로그램

- Eclipse (ver. Indigo)

모든 프로그래밍은 이클립스를 사용하여 코딩하였음

- Photoshop cs4

약간의 이미지 작업(게임 카드 이0미지)을 하기위해 사용하였음.

1.2 개발환경

- OS : Windows 7

- Java : jdk 1.7.0

Jre 1.6.0

**2. 프로젝트 구성**

2.1 프로젝트 이름 : javafinal

2.2 패키지 이름 : javafinal (단일 패키지로 구현하였음)

2.3 패키지 내 파일

**- Gameset.java**

Gameset class : Shape 클래스를 인스턴스화 하여 게임에 이용되는 레벨마다 별 초기 카드 상태를

저장하고 있는 클래스

현재 구현된 코드에는 0레밸 ~ 3래벨의 게임이 저장되어있음.(총4개의 래벨)

**- GuiTest.java**

GuiTest class : 프로젝트의 메인이 되는 클래스 MainForm class 를 실행하는 역할을 한다

실제적인 실행은 모두 MainFrom class 에서 일어난다.

**- Search.java**

Search class : 게임 내에서 동일한 카드가 연결되는지 확인하여 삭제하는 역할을 하는 클래스

**- Shape.java**

Shape class : 게임에 이용되는 카드의 배열의 type을 제공해주는 클래스

**- Timer.java**

Timer class : 게임에 표기되는 시간을 count을 표시해주는 클래스

**- MainForm.java**

실직적인 내부 코드가 모두 정의되어있는 클래스

**3. 구현한 알고리즘**

3.1 내부 ActionListener 클래스

총 3개의 ActionListener 클래스를 정의하였음

3.1.1 levelListener

게임 내에서 게임 레벨의 선택하는 JComboBox를 클릭하였을 때 콜백되는 클래스로써 레벨을 선택 할 경우 해당 레벨에 맞는 카드 배열이 화면에 표시되고 또한 화면 상단에 정보(레벨,타이머등)가 설정되고 게임이 실행되기 전이므로 카드가 lock되어있는 상태로 설정하는 기능을 구현하였음

3.1.2 btnListener

게임 내 버튼인 Start, Reset, Wait, Resume 버튼을 클릭하였을 때 콜백되는 클래스로써 해당 버튼의 기능을 구현하였음

주요기능:

Start button : 게임을 실행하기 위해 카드의 lock을 해제하고 타이머를 작동시킴.

Reset : 게임,타이머을 즉시 중단하고 선택되어있는 레벨의 게임으로 초기화하고 카드를 lock시킴.

Wait : 게임,타이머을 잠시 중단하고 화면을 가린다. 모든 카드를 lock시킴.

또한 Wait 버튼을 Resume버튼으로 변경시킴.

Resume : Wait버튼을 클릭했을 때 나타나는 버튼으로써 Wait를 통해 잠시 중단되었던 게임을 다시

이어서 할 수 있도록 카드를 unlock 시키고 타이머를 작동시킴.

3.1.3 cardListener

카드를 클릭하면 콜백되는 메서드로써 카드를 선택하여 카드의 위치를 바꾸고 연결되는 카드가 있는지 확인하는 메소드를 실행시킴.

카드를 처음 선택하면 해당 카드의 번호가 저장되며 두번째 클릭하면 선택된 카드가 서로 인접한 카드인지 확인하고 인접한 카드이면 서로 위치를 Swap하게 됨.

서로 위치를 바꾼 경우에는 화면에 바뀐 카드위치를 표시함.

또한 카드가 3개이상 연결되어있을 경우 찾아서 삭제시키는 메소드를 실행함.

3.2 lock 메소드

boolean형의 변수를 인자로 받아서 true(lock)일 경우 게임 내 카드를 모두 잠그고 선택이 불가능한 카드로 화면을 재표시함.

False인 경우 실행중이던 게임의 상태로 다시 화면을 표시하는 역할을 함.

3.3 StateInit 메소드

게임 내에서 게임의 상태를 저장하고 있는 Shape type의 객체인 shapeState를 초기화 하는 메소드

3.4 Timer 클래스

Timer클래스는 게임중 표시되는 시간을 표시할때 사용하는 클래스로 runnable interface를 구현하여 Mainform 안에서 생성되어 ExecuterService객체를 통해 실행되게 됨

3.4.1 생성자

생성자는 2가지를 선언.

인자가 1개인 생성자는 결과값을 표시하기 위하여 JLabel객체 전달받아 레퍼런스를 유지하게 됨.

인자가 2개인 생성자는 위의 역할뿐만 아니라 타이머가 실행될 초기 시간(초)를 전달받아 그 시간부터 count 하게 됨.

3.4.2 run 메소드

run메소드는 Timer를 Thread.sleep(1000)메소드를 통해 1초마다 자신을 다시 부름으로써 stop변수가

0이 아닐때까지 계속하여 time변수를 count하여 Timer를 구현함.

mainForm에서 쓰레드의 stop변수를 1로 만듦으로써 쓰레드를 멈추는 방법을 사용하였음.

3.5 Search 클래스

Search 클래스는 게임내 규칙인 동일한 카드가 연속으로 3개 이상 나열되어있을 경우 해당 카드들을 삭제하는 규칙을 구현하기 위한 클래스로써 게임내 모든 카드를 조사하여 카드가 연결된 상태를 조사하고 삭제함.

3.5.1 생성자

생성자에서는 인자로 전달받은 레밸을 저장하고 게임의 상태를 저장하고 있는 변수를 레퍼런스한다.

그밖에 다른 변수를 초기화함.

3.5.2 run 메소드

메소드는 자신과 같은 카드를 각 방향(up,down,left,right)의 4가지 함수를 통해 조사한다.

각 방향의 Search메소드는 재귀함수로써 만약 본체와 같은 카드가 Shape 배열의 바운스 안에서

연속에서 존재하는 경우 그 개수 조사하여 리턴함

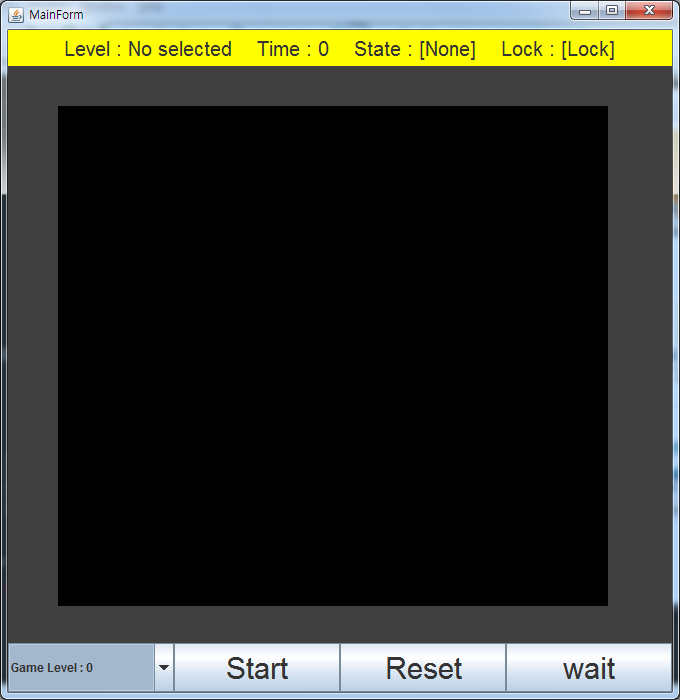
이 리턴 되는 개수를 통해 같은 카드가 가로 혹은 세로로 3개 이상 연결되어 있을경우 그 개수 만큼 카드를 삭제함.

3.5.3 각 방향 Search 메소드(upSearch,downSearch,leftSearch,rightSearch)

각 방향의 Search메소드는 재귀함수로써 만약 본체와 같은 카드가 Shape 배열의 바운스 안에서

연속에서 존재하는 경우 그 개수 조사하여 리턴함

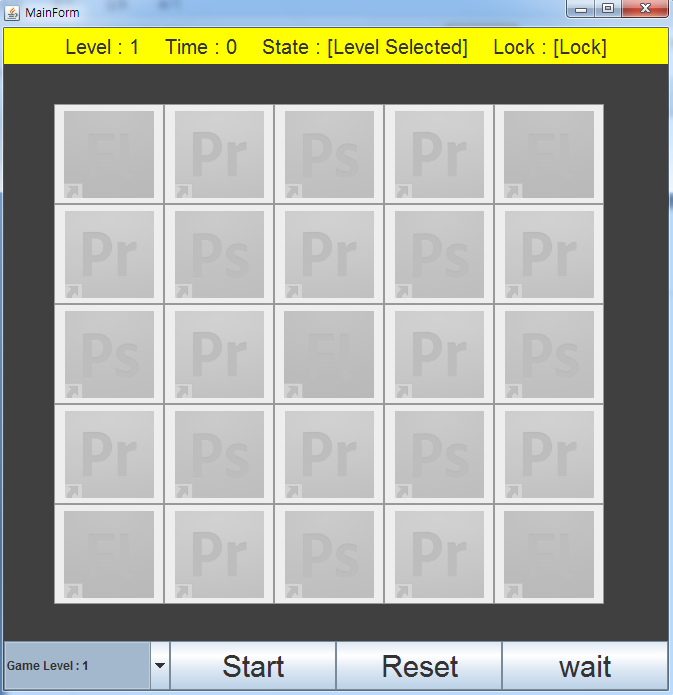
**4.코드 실행화면**

****

4.1 실행 초기화면

래밸의 선택하지 않으면 게임이 start되지 않음

화면 상단에 게임의 상태를 표시해줌

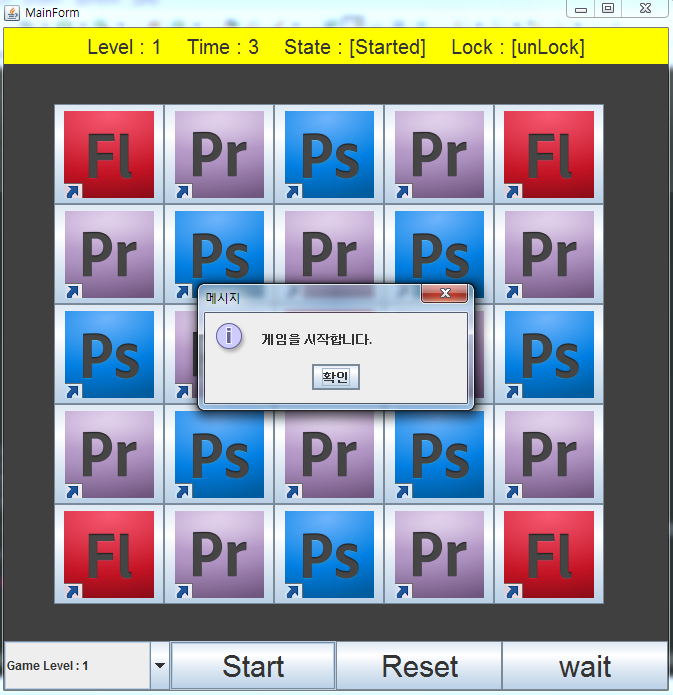


4.2 게임 레벨 선택한 화면

화면 중간에 카드들이 레벨에 맞게 배치됨

아직 게임을 Start하지 않았기 때문에 카드가 Lock되어있고

Lock상태가 화면 상단에 표시됨

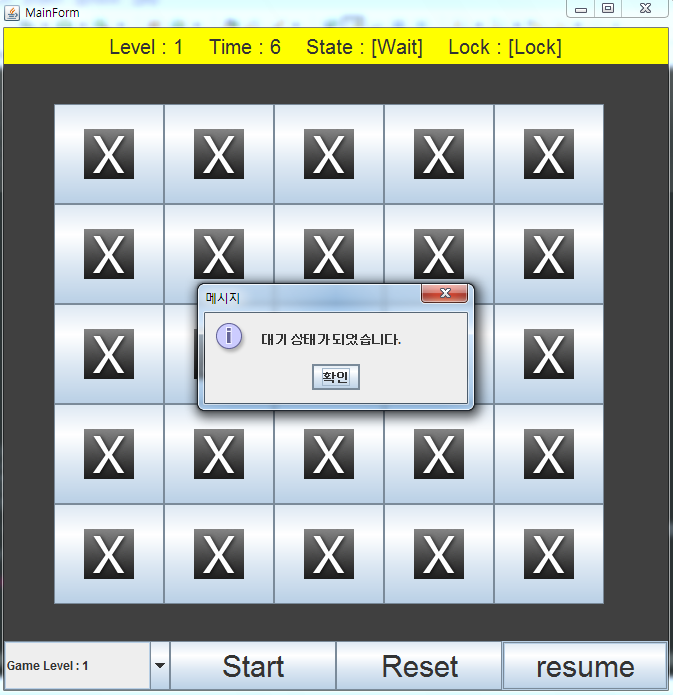


4.3 게임을 시작한 화면

Start버튼을 누르게 되면 Dialog창이 나타나면서 게임이 실행됨

타이머가 작동되는 것을 확인할 수 있음

카드가 unlock되고 그 상태가 화면 상단에 표시됨



4.4. Wait 상태

Wait 버튼을 누르게 되면 나타나는 상태

Dialog창이 대기상태가 되었다고 알려줌

모든 카드를 볼 수 없도록 가리고 카드가 lock상태가 됨

타이머가 중단됨

Wait버튼이 Resume버튼으로 변하게 됨

Resume을 누르게 되면 이전에 게임을 진행중이던 상태로 돌아가게 됨