

< 목차 >

1. 사용자 매뉴얼

- 가. 개요
- 나. 게임 시작 화면
- 다. 게임 실행 화면
- 라. 게임 결과 화면

2. 개발자 매뉴얼

- 가. Model
 - 1) MainView
 - 2) StartPanel
 - 3) HelpPanel
 - 4) GamePanel
 - 가) PuzzlePanel
 - 5) EndPanel
- 다. Controller
- 라. Starter

3. BGM 출처

4. 디자인 스케치

I. 사용자 매뉴얼

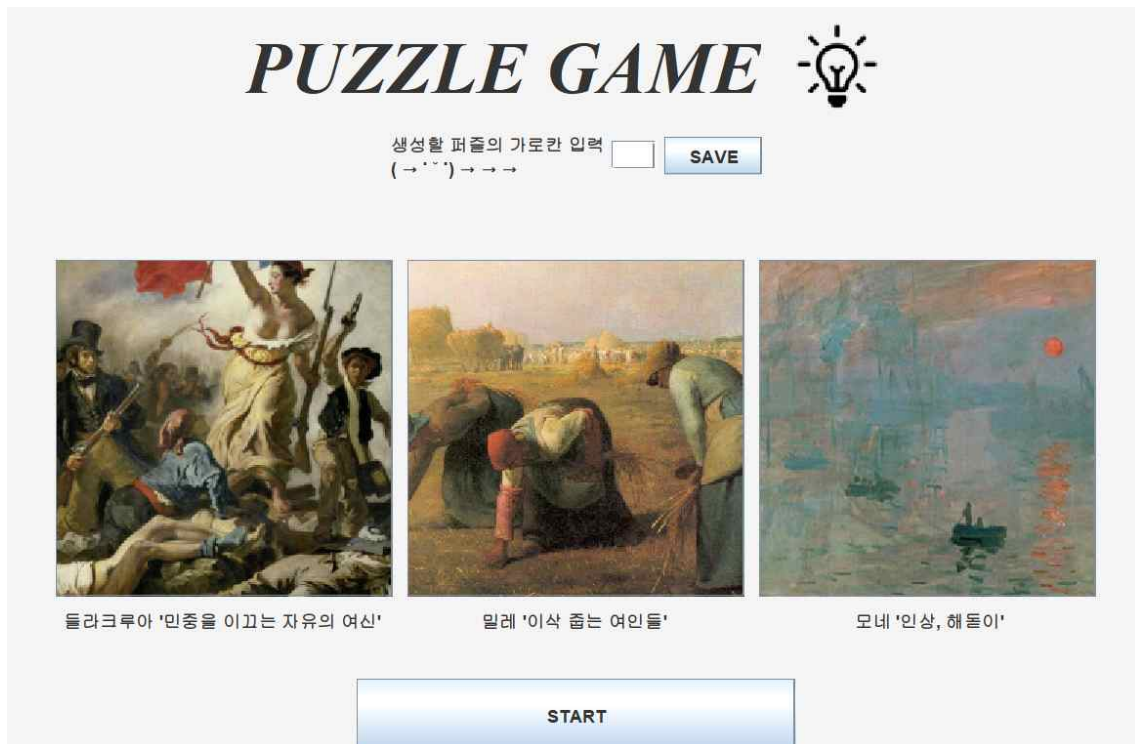
1. 개요

Puzzle Game이란 랜덤하게 조각나있는 그림들을 이동하여 원래의 그림을 맞추는 게임입니다. 이에 미술 작품이라는 요소를 추가하여 사용자가 직접 작품을 고를 수 있습니다.

프랑스 시민혁명 직후 지배계층이 원하는 화풍에서 벗어나 화가의 주관과 감정을 중요시하는 화풍인 낭만주의를 대표하는 외젠 들라크루아의 '민중을 이끄는 자유의 여신', 낭만주의 다음으로 등장한 서민적인 생활을 주 소재로 하는 사실주의 화풍 중 대중들에게 유명한 장 프랑수아 밀레의 '이삭 줍는 여인들', 그리고 마지막으로 시간에 따라 변하는 '빛'을 그리는 인상주의의 대표적인 화가인 클로드 모네의 '인상, 해돋이'라는 작품입니다.

단순히 퍼즐을 완성하는 것을 목표로 갖는 것이 아닌, 퍼즐을 맞추는 과정에서 작품에 대한 재미 또한 얻을 수 있을 것입니다.

2. 게임 시작 화면



● 도움말

상단의 전구 모양을 클릭하면 도움말 창으로 전환됩니다.

도움말 창에는 퍼즐 게임에 대한 설명과 게임 실행 방법이 명시되어 있습니다.

도움말 창의 좌측 상단 화살표를 클릭하면 다시 게임 초기 화면으로 되 돌아옵니다.



Help

< PUZZLE GAME 이란? >

랜덤하게 조각나있는 그림들을 이동하여 원래의 그림을 맞추는 게임입니다.

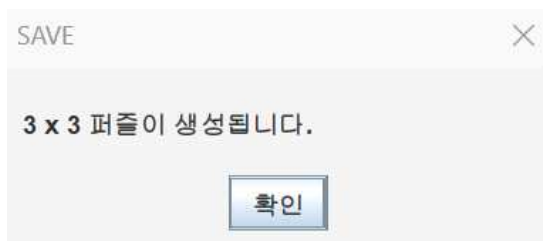
< 게임 방법 >

1. 퍼즐의 가로칸을 저장해주세요.
2. 퍼즐의 그림을 선택해주세요.
3. START 버튼을 누르세요.
4. 빈 퍼즐의 주변을 클릭하면 퍼즐이 움직입니다.

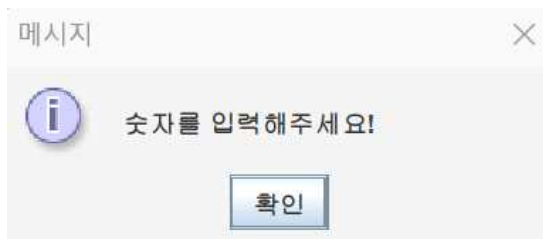


● 게임 시작 방법

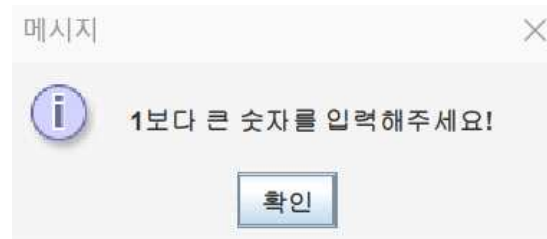
① 퍼즐의 가로칸을 입력한 후 'SAVE' 버튼을 클릭하면 입력한 값이 저장됩니다.



↑ 올바른 숫자를 입력한 경우



↑ 숫자가 아닌 문자열 등을 입력한 경우



↑ 1보다 작은 숫자를 입력한 경우

② 세 개의 그림 중 생성하고 싶은 퍼즐의 그림을 클릭하면 그림이 선택됩니다.

그림 생성



<민중을 이끄는 자유의 여신>
낭만주의 화가 외젠 들라크루아가 프랑스 7월 혁명을 기념하여 그린 그림

"나는 현대적인 주제, 즉 바리케이트전(戰)을 그리기 시작했습니다.
나는 조국의 승리를 위해 직접 나서지는 못했지만
그래도 조국을 위해 이 그림을 그리고 싶습니다."

확인

그림 생성



<이삭 줍는 사람들>
사실주의 화가 장 프랑수아 밀레가 농촌의 생활을 그린 그림

"다른 사람을 감동시키려면 먼저 자신이 감동하지 않으면 안된다.
그렇지 못하면 아무리 정교한 작품이라도 결코 생명력을 갖지 못한다."

확인

그림 생성



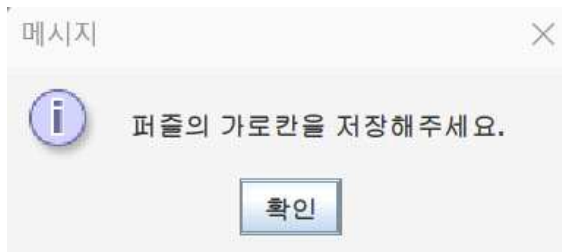
<인상, 해돋이>
인상주의 화가 클로드 모네가 프랑스 항구의 아침 풍경을 그린 그림

"우리가 실제로 보는 것은 반사된 빛이므로,
진짜 눈에 보이는 그대로 그리려면 반사된 빛을 그리면 된다."

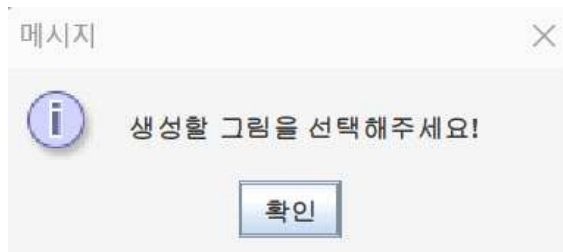
확인

↑ 각 작품을 클릭했을 때 보이는 창

③ 위 두 과정을 모두 완료한 후 START 버튼을 클릭합니다.



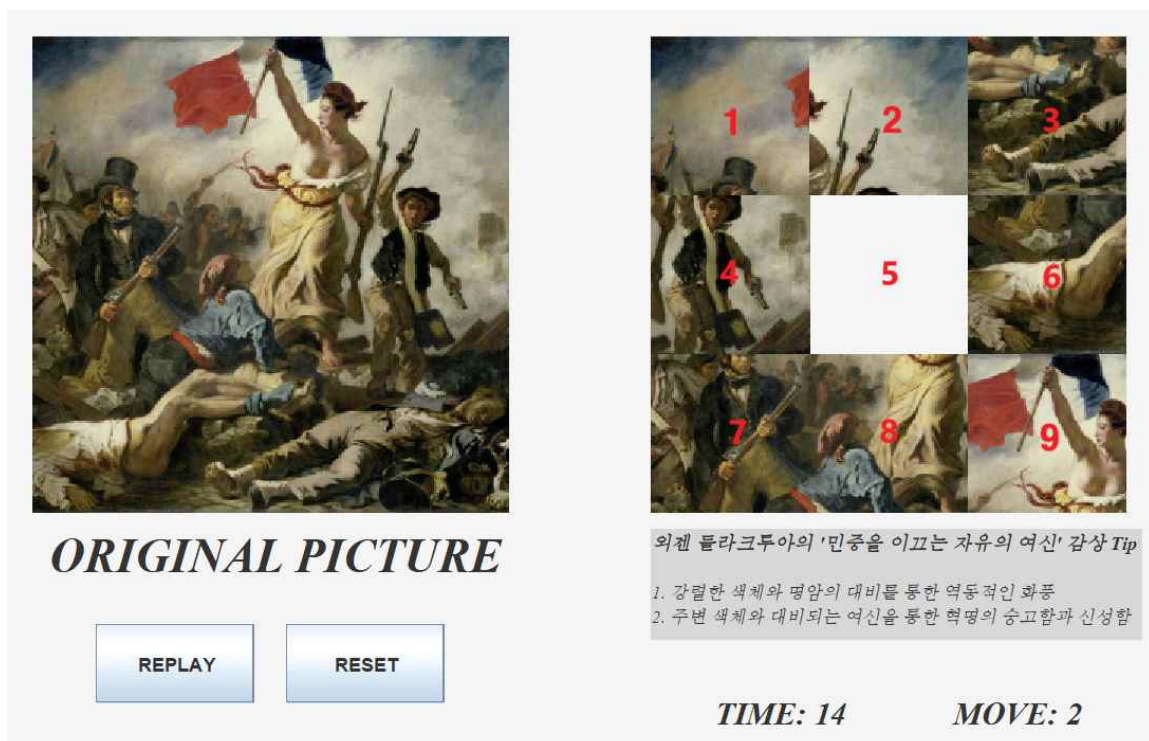
↑ 가로칸을 저장하지 않고 클릭한 경우



↑ 그림을 선택하지 않고 클릭한 경우

④ 화면이 게임 실행 화면으로 전환됩니다.

3. 게임 실행 화면



● 그림 퍼즐

좌측의 원본 그림을 보면서 퍼즐의 빈 칸의 주변을 클릭하여 퍼즐을 이동시켜 원래의 그림대로 만드는 게임입니다.

상단의 그림처럼 1-9 까지의 가상의 숫자가 있다고 가정할 때, 빈칸은 5이므로 그 상하좌우에 해당하는 2, 4, 6, 8만 클릭하여 이동시킬 수 있습니다.

● 감상 Tip

그림 퍼즐 하단에 위치한 감상 Tip은 세 작품 모두 존재합니다. Tip을 읽고 작품을 보게 된다면 게임을 더 재밌게 즐길 수 있습니다.

● TIME & MOVE

‘TIME’은 게임이 시작된 이후로부터 경과한 시간을 보여주고, ‘MOVE’는 이동횟수를 보여줍니다.

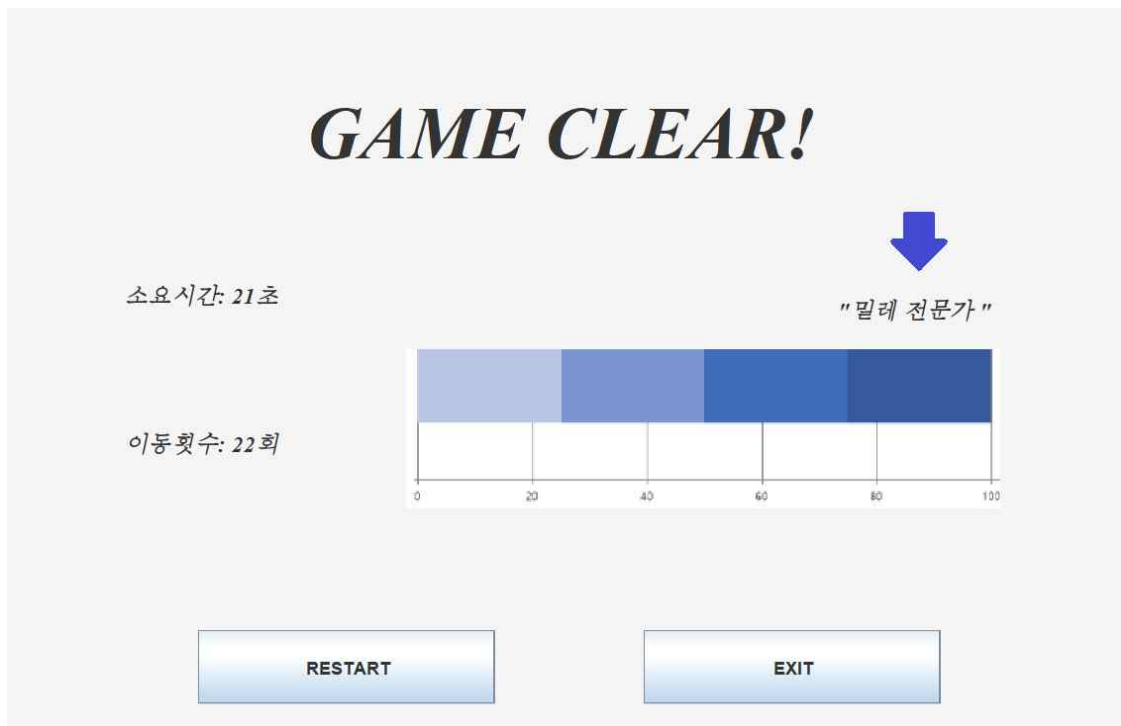
- **REPLAY 버튼**

‘REPLAY’ 버튼을 클릭하면 실행하던 게임을 중단하고 게임 시작 화면으로 돌아갑니다. 게임 시작 화면에서는 동일하게 퍼즐의 가로칸 수와 그림을 선택할 수 있습니다.

- **RESET 버튼**

‘RESET’ 버튼을 클릭하면 퍼즐을 다시 섞고 재배치합니다. 경과시간과 이동횟수 또한 초기화됩니다.

4. 게임 결과 화면



- **소요시간 & 이동횟수**

퍼즐이 완성될 때까지의 시간과 이동횟수를 보여줍니다.

- **점수**

	초보	중수	고수	전문가
가로칸 ≤ 3	2분 초과	2분 이내	1분 이내	30초 이내
가로칸 > 3	3분 초과	3분 이내	2분 이내	1분 이내

점수는 수치화가 아닌 칭호 개념으로 측정됩니다.

가로칸 수가 2칸 혹은 3칸인 경우에는 30초 이내에 완성하면 전문가, 1분 이내에 완성하면 고수, 2분 이내에 완성하면 중수, 2분을 초과하면 초보로 구분합니다.

가로칸 수가 4칸 이상부터는 1분 이내에 완성하면 전문가, 2분 이내에 완성하면 고수, 3분 이내에 완성하면 중수, 3분이 초과된 경우 초보로 구분됩니다.

- **RESTART 버튼**

‘RESTART’ 버튼을 클릭하면 게임 시작 화면으로 돌아가서 다시 게임을 실행할 수 있습니다. 원래 실행했던 게임의 가로칸 수와 그림이 초기값으로 설정되어있습니다.

- **EXIT 버튼**

‘EXIT’ 버튼을 클릭하면 게임이 종료되고 게임창이 닫힙니다.

II. 개발자 매뉴얼

1. Model.java

- int n

사용자가 입력한 퍼즐의 가로칸 수

- String imagePath

사용자가 선택한 그림의 파일명을 리턴하는 함수

- void setN(int n)

StartPanel에서 사용자가 입력한 n을 저장하는 함수

- int getN()

n 값을 리턴하는 함수

- void setImagePath(String imagePath)

StartPanel에서 사용자가 선택한 그림의 파일명을 저장하는 함수

- String getImagePath()

파일명을 리턴하는 함수

2. View.java

MainView	StsrPanel: 게임 시작 화면	
	HelpPanel: 게임 도움말 화면	
	GamePanel: 게임 실행 화면	PuzzlePanel: 퍼즐 생성 Panel
	EndPanel: 게임 결과 화면	

1) MainView.java

- 패널 전환을 위한 CardLayout 설정

cardPanel을 Container parent로 사용하여 CardPanel을 설정

- setPreferredSize(new Dimension(800, 550));

화면 크기 (800 * 550) 설정

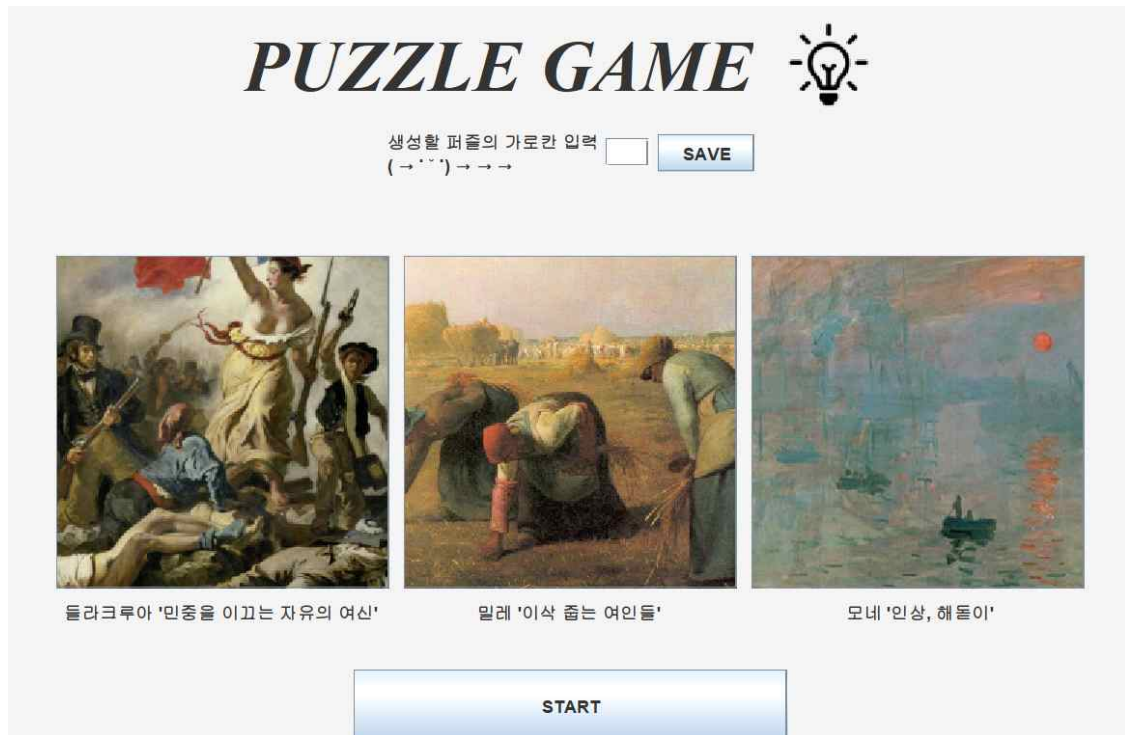
- addPanel(JPanel panel, String panelName)

MainView의 CardLayout에 Panel 추가

- changePanel(String panelName)

인자로 받은 이름을 가진 Panel로 전환

2) StartPanel.java



- BorderLayout() 설정

- BorderLayout.NORTH

introPanel	JLabel title: 중앙 상단의 게임명
	JButton helpButton: 클릭시 HelpPanel로 전환되는 도움말 Button

- BorderLayout.CENTER

choosePanel : BorderLayout()	BorderLayout.NORTH setupPanel: 사용자 퍼즐 가로칸 입력	nLabel
		text: 가로칸 입력 Field
		nButton (ActionListener)
	BorderLayout.CENTER imagePanel: 사용자 퍼즐 그림 선택	delacroixButton (")
		milletButton (")
		MonetButton (")
	BorderLayout.SOUTH explainPanel: 퍼즐 그림명	label1
		label2
		label3

- BorderLayout.SOUTH

actionPanel	startButton: 조건 만족 후 클릭시 GamePanel로 화면 전환
-------------	---

※ 조건: 사용자의 올바른 가로칸 입력 + 그림 선택

- getN()

Controller에 사용자가 입력한 가로칸 수인 n을 정수로 변환하여 리턴하는 함수

- getHelpButton()

Controller 내에서 도움말 HelpPanel로 전환하는 ActionListener 부착

- getNButton()

Controller 내에서 입력한 n에 대한 메시지창을 띄우고 n을 저장하는 ActionListener 부착

- getDelacroixButton(), getMilletButton(), getMonetButton()


Controller 내에서 각 작품을 클릭하면 메시지를 띄우는 ActionListener 부착

3) HelpPanel.java



null	backButton: 좌측 상단의 버튼을 클릭하면 StartPanel로 전환
	title: 중앙의 패널명
	qLabel: 퍼즐 게임 질문 Label
	explainLabel: 퍼즐 게임 설명
	methodLabel: 게임 방법 Label
	label1, 2, 3, 4: 게임 방법 설명
	helpImage: 게임 시작 화면 실행 순서가 명시된 이미지


4) GamePanel.java



ORIGINAL PICTURE

REPLAY

RESET



외젠 들라크루아의 '민중을 이끄는 자유의 여신' 감상 Tip

1. 강렬한 색채와 명암의 대비를 통한 역동적인 화풍
2. 주변 색채와 대비되는 여신을 통한 혁명의 숭고함과 신성함

TIME: 6

MOVE: 0

null	helpPanel	originalPanel: ImageIcon으로 원본 이미지 삽입
		picturePanel: 원본 이미지 Label
		replayButton: 클릭시 StartPanel로 전환
		resetButton: ActionListener 부착
	① puzzlePanel	퍼즐 생성 Panel
	explainPanel	explainText1: 감상 Tip
		explainText2: Tip 세부 내용
	timeLabel	게임이 시작된 시간으로부터 경과한 시간
	moveLabel	퍼즐 이동 횟수 카운팅

- timer 생성

1초마다 실행되어 time 값을 하나 증가

puzzlePanel의 getCount()를 통해 이동횟수를 가져옴

setText()를 통해 TIME과 MOVE 값 변화 반영

solved() 메소드로 퍼즐이 완성됐는지 수시로 확인

- resetButton.addActionListener()

puzzlePanel 메소드 randomPuzzle()을 이용해서 퍼즐의 순서섞고, repaint()로 퍼즐 재조정
timer를 멈추고 0으로 초기화 시킨 후 재시작

- setExplainText()

사용자가 고른 작품에 맞는 감상 Tip으로 explainText1과 2의 내용을 초기화

- solved()

puzzlePanel의 solved() 함수가 true라면 timer를 멈춤

퍼즐이 완성됐을 때의 time과 count를 저장

Mainview, 파일명, 가로칸 수, 소요시간, 이동횟수를 인자로 가지는 EndPanel 객체 생성
mainView에 EndPanel 추가 후 전환

① PuzzlePanel.java

퍼즐 생성 메소드 설명

- saveImageSpace(String imagePath)

step 1. 이미지 배열과 숫자 배열 생성

step 2. 이미지 불러오기

step 3. 불러온 이미지를 잘라서 이미지 배열에 저장 + 숫자 배열에 순서 저장
(마지막은 빈칸 유의)

- randomPuzzle()

step 1. 반복문으로 0~숫자 배열의 값을 랜덤으로 돌려서 j에 저장

step 2. j와 숫자 배열의 값 교환

step 3. 배열의 마지막 값을 빈칸으로 저장

- paint(Graphics g)

step 1. 가로칸 수까지만 가로로 배치

step 2. 숫자 배열의 마지막을 제외한 곳에 숫자에 맞는 이미지 그리기

- handleClick(MouseEvent e)

step 1. 마우스가 클릭한 칸과 빈칸의 행과 열 찾기

step 2. 클릭한 곳이 빈칸의 주변에 해당하는지 확인하기

step 3. 사운드 효과("Move.wav") 넣기

step 4. 클릭한 칸과 빈칸의 숫자 배열 변경

step 5. 클릭한 칸을 빈칸으로 저장

step 6. repaint()

- 생성자 puzzlePanel(String imagePath, int n)

퍼즐 크기 (300 * 300) 설정

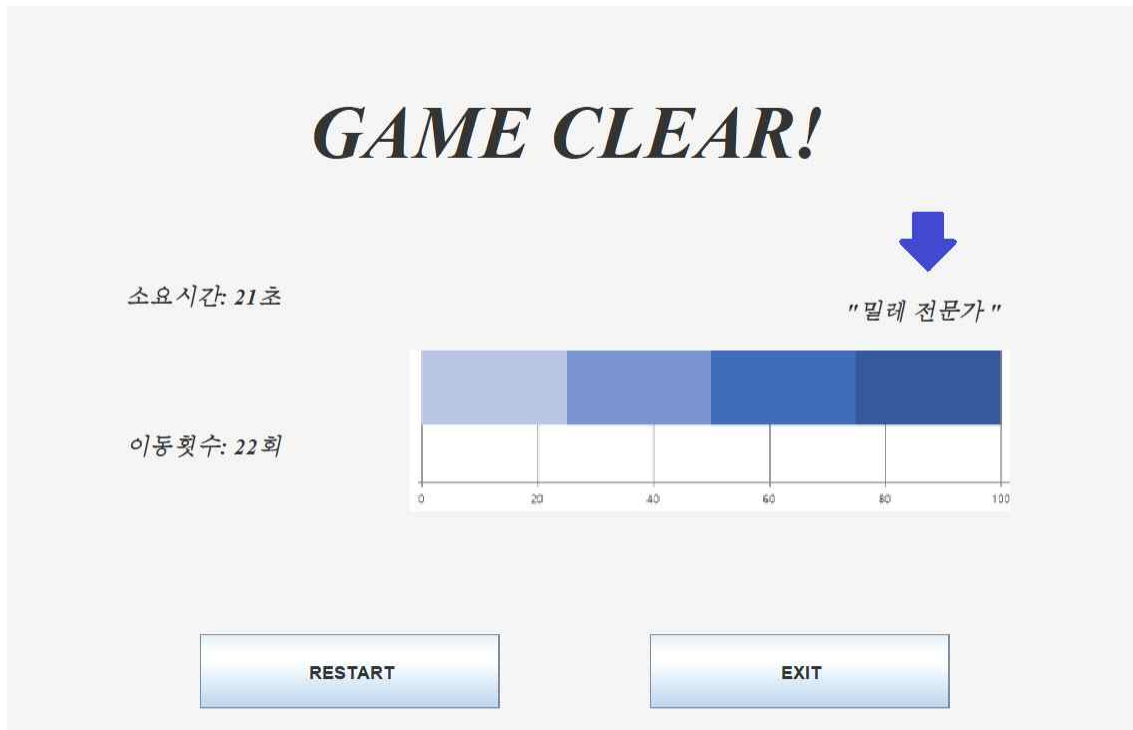
생성한 메소드 순서대로 호출

MouseEvent가 있을 때마다 solved()를 호출하여 퍼즐이 완성되었는지 확인

- solved()

반복문의 숫자와 현재 숫자 배열의 위치가 일치하는지 확인

5) EndPanel.java



- GridLayout(3,1) 설정

title: 중앙 상단 Label

resultPanel	timeLabel: 소요시간
	countLabel: 이동횟수
	scoreImage: 점수 이미지
	score: 점수에 따른 칭호명
lasPanel	restartButton: 클릭시 StartPanel로 전환
	exitButton: 클릭시 게임 종료하고 창 닫음

- 칭호명 기준

	초보	중수	고수	전문가
n <= 3	else	time <= 120	time <= 60	time <= 30
n > 3	else	time <= 120	time <= 120	time <= 60

3. Controller.java

- 생성자 Controller()

BGM 설정: clip 객체 생성 후 start() 함수 사용하여 "BGM.wav" 설정

startPanelListener();

helpPanelListener();

- startPanelListener();

StartPanel에 있는 각 Button에 ActionListener 부착

getHelpButton()	HelpPanel을 MainView에 추가 후 전환
getNButton() : startPanel.getN()	n < 2: 1보다 큰 숫자를 입력하라는 메시지
	else: model.setN()에 n 저장
	NumberFormatException: 숫자를 입력하라는 메시지
getDelacroixButton()	그림의 파일명을 model.setN()에 저장 그림 관련 메시지
getMilletButton()	
getMonetButton()	
getStartButton()	모든 요건 만족: gamePanel을 MainView에 추가 후 전환
	model.getImagePath() == null: 그림을 선택하라는 메세지
	else: 퍼즐의 가로칸을 입력하라는 메시지

※ 모든 요건: (n > 1) && (model.getImagePath() != null)

- helpPanelListener();

getBackButton()의 ActionListener 부착: StartPanel로 전환

- clickSound();

clip 객체 생성 후 start() 함수 사용하여 “Click.wav” 설정

- start();

최종적으로 실행될 Controller의 메소드

초기 화면에 해당하는 StartPanel을 MainView에 추가하고 전환하여 초기화면 실행

4. Starter.java

- Controller controller = new Controller();

퍼즐 게임 실행을 위한 Controller 객체 생성

- controller.start();

Controller 내 start() 메소드를 실행하여 게임 시작 화면부터 실행

Ⅲ. BGM 출처

유튜브 단무 https://www.youtube.com/watch?v=TaVLl_elGtE

초기 설정 화면

```
< Model - 데이터 저장과 처리 >
- 사용자가 입력한 Board의 n x n
int n;
- 값을 전달받은 용수
void setN(int n) { this.n = n; }
int getN() { return n; }
```

< View - 사용자 입력, 데이터 전달, 출력 >

```
- StartPanel 생성 (Layout: BorderLayout)
JPanel (start) [ JLabel : 입력사항 (0~5) ) CENTER
                [ JTextField : (1) 칸 ) SOUTH
- JPanel (startButton)
JPanel (intro) [ JLabel : 'Puzzle Game' ) NORTH
                [ JLabel : '게임 설명' ) SOUTH
```

< Controller - Model, View 객체 실행 및 제어 >

```
View를 불러와서 → 입력받은 n을 → Model에 전달
→ Start 버튼을 누르면 → ActionListener 생성
StartPanel : 입력값 (false)
GamePanel : 요미 (true)
```

* Frame - BorderLayout Hg

```
→ 프레임 생성
public void changePanel() {
    cords.next(this.getContentPane());
    ↑
    getContentPane().add = new BorderLayout();
```

JAVA Project. Puzzle

< 게임 화면 >

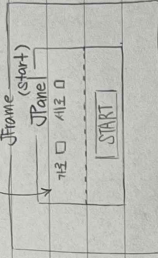
생성 영역에 따라 3x3, 4x4, 5x5가 선택될 수 있게 함

여러개를 입력할 것

mvc에 따라 클래스를 네 개 작성

시작 화면을 옮길 것

이동할 것, 스프레드를 작성하여 결과

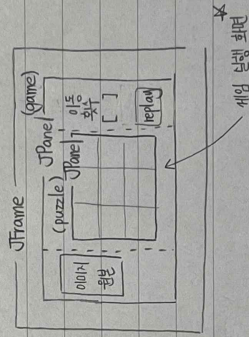


첫 메인 화면 (초기 설정 화면) Layout: BorderLayout

↳ 있어야 하는 것: (n x n)을 선택할 수 있는 칸 (JTextField)

start 버튼 (JButton)

↳ 추가할 기능: 이미지 선택 가능한 기능, 배경 음악 on/off, 게임 설명 + 지도



- 게임 실행 화면 (StartPanel을 settable (false)하여 사용)

Layout: BorderLayout

↳ 있어야 하는 것: 이미지 문명 (JPanel에 graphics)

파본 생략과 MouseListener (JPanel)

Layout: GridLayout (n x n)

이동할 것 (JLabel)

replay 버튼 (JButton) → 초기 화면으로

- 게임 종료 화면 (파본 관련된 것)

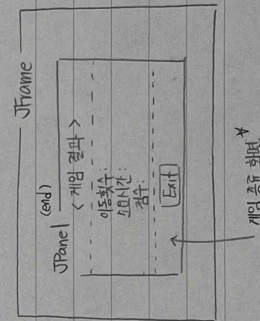
↳ 있어야 하는 것: 게임 결과 (JLabel)

이동할 것, 요미간 (JLabel)

잠수 (JLabel)

End 버튼 (JButton)

↳ 추가할 기능: 잠수? 잠수 text 깨는 저장 후 TextArea



게임 종료 화면

