

Java Programming (0409)

Final Project Report

Department of Computer Science and Engineering, Konkuk University
Yulwon Rhee (202211342)

담당 교수: 지정희 교수님
제출일: 2022년 12월 9일 금요일

1 문제 정의 및 분석

본 프로젝트는 Java와 Swing을 이용하여 빙고 게임 데스크탑 애플리케이션을 제작하는 것을 목표로 한다.

1.1 문제 정의

빙고판 생성: 사용자에게 N 빙고판의 크기를 입력받은 후, 단어장 파일에서 N^2 개의 랜덤한 단어를 골라 빙고판을 두 개 생성한다.

턴 진행: 빙고 게임은 유저와 컴퓨터가 번갈아가며 빙고판의 단어를 선택하는 식으로 진행된다.

단어 선택: 빙고판의 단어가 선택된 경우, 선택된 단어와 그 단어의 뜻을 출력한다.

단어 체크: 유저와 컴퓨터의 빙고판에 선택된 단어가 있는지 확인하고, 체크한다.

컴퓨터의 단어 선택: 컴퓨터가 단어를 선택할 때에는 승리를 위한 알고리즘을 설계하여 그에 따라 선택하도록 한다.

빙고 체크: 한 턴이 진행된 후마다 빙고 수를 체크한다. 빙고의 수가 1개 이상 많은 경우 승리하도록 한다.

승률 저장: 사용자와 컴퓨터의 승률을 파일로 저장하여, 다음 실행 시에도 불러오도록 한다.

UI 개선: Swing을 이용하여 UI를 개선한다.

2 주요 소스코드 설명

메소드들은 여러 클래스에 나뉘어 존재한다.

파일 이름	역할
BingoGame.java	프로그램의 시작점. 상황에 맞는 Frame을 띄워준다.
Game.java	통계를 위한 게임 정보를 관리하는 Class.
User.java	플레이어와 컴퓨터의 게임 중 데이터를 관리하는 Class.
Word.java	단어장에서 단어를 불러오고, 이를 관리하는 Class.
StatisticUtil.java	통계를 관리하는 Class.
StartFrame.java	게임을 시작하기 전 필요한 정보를 수집하는 Frame.
GameFrame.java	게임 화면 Frame. 게임과 관련된 동작이 여기서 이루어진다.
StatisticDialog.java	통계를 자세히 보여주는 Dialog.

2.1 BingoGame.java

public static void main(String[] args): 프로그램의 시작점으로써, 시작 시 저장된 통계 파일을 읽어와 StartFrame을 호출한다. 이후, StartFrame에서 단어장 파일, N 의 값과 AI 수준을 받아와 GameFrame을 호출한다. 게임이 끝나면, 게임 시작 시간, N 의 값, AI 수준과 승패 여부를 받아와 통계 리스트에 저장하고, 통계 파일에 추가한다.

Source Code

```

1  package ywrhee.project;
2
3  import com.formdev.flatlaf.FlatDarkLaf;
4
5  import javax.swing.*;
6  import java.awt.*;
7  import java.util.ArrayList;
8
9  public class BingoGame {
10
11     public static void main(String[] args) {
12         System.setProperty("apple.laf.useScreenMenuBar", "true");
13         System.setProperty("apple.awt.application.name", "202211342 이율원");
14         System.setProperty("apple.awt.application.appearance", "system");
15         FlatDarkLaf.setup();
16
17         try {
18             UIManager.setLookAndFeel(new FlatDarkLaf());
19             UIManager.getLookAndFeelDefaults().put("defaultFont", new Font("Pretendard",
20                                     ↳ Font.PLAIN, 14));
21         } catch (Exception e) {
22             e.printStackTrace();
23         }
24     }
25 }
```

```
22     }
23
24     StatisticUtil.init();
25
26     while (true) {
27         StartFrame startFrame = new StartFrame("BINGO GAME",
28             ↪ StatisticUtil.getStatisticList());
29
30         ArrayList<Word> wordList = Word.getWordList(startFrame.getSelectedFile());
31         int N = startFrame.getN();
32         int difficulty = startFrame.getDifficulty();
33
34         User user = new User(wordList, N);
35         User computer = new User(wordList, N);
36
37         Game game = new Game(N, difficulty);
38         GameFrame gameFrame = new GameFrame("BINGO GAME", user, computer, N, difficulty);
39         game.setWinLoseInfo(gameFrame.getWinLoseInfo());
40
41         StatisticUtil.addToStatisticList(game);
42         StatisticUtil.writeStatistic(game);
43     }
44 }
45
46
47 }
```

2.2 Game.java

승패 여부, N 의 값, AI 수준 및 게임 플레이 시간 등 게임 데이터를 관리하는 Class이다. 게임에 필요한 상수가 **static**으로 선언되어 있고, 통계 파일에서 데이터를 파싱해 Game 클래스로 바꿔주는 **static** 메소드가 포함되어 있다.

public String getCsvInfo(): 게임 데이터를 CSV(Comma-Separated Values) 형태로 변환해 텍스트 형태로 저장하기 쉽게 반환한다.

public Object[] getTableRow(int winLoseInfo): 게임 데이터를 JTable의 Row에 바로 넣을 수 있도록 가공하여 반환한다.

public static Game parseGameInfo(String gameInfoAsCsv): CSV 형태로 변환된 게임 데이터를 파싱하여 다시 Game 객체로 되돌려 반환한다. 인스턴스를 생성하지 않아도 바로 쓸 수 있게 **static** 메소드로 만들었다.

Source Code

```

1 package ywrhee.project;
2
3 import java.time.LocalDateTime;
4 import java.time.format.DateTimeFormatter;
5 import java.util.StringTokenizer;
6
7 public class Game {
8     static final String[] RESULT_TEXT = {"승리", "패배", "무승부"};
9     static final int VICTORY = 0;
10    static final int DEFEAT = 1;
11    static final int DRAW = 2;
12    static final int CONTINUE = 3;
13
14    static final String[] DIFFICULTY_TEXT = {"쉬움", "보통", "어려움"};
15    static final int EASY = 0;
16    static final int NORMAL = 1;
17    static final int HARD = 2;
18
19
20    private int winLoseInfo;
21    private int N;
22    private int difficulty;
23    private String playDateTime;
24
25    public Game(int N, int difficulty) {
26        this.N = N;
27        this.difficulty = difficulty;
28
29        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
30
31        this.playDateTime = LocalDateTime.now().format(formatter);
32    }
33
34    public Game(int winLoseInfo, int N, int difficulty, String playDateTime) {
35        this.winLoseInfo = winLoseInfo;
36        this.N = N;
37        this.difficulty = difficulty;
38        this.playDateTime = playDateTime;

```

```
39     }
40
41     /**
42     * 승패 정보, 빙고판 크기, AI 수준, 플레이 시간 순으로 CSV 만들어 반환
43     * @return CSV Formatted data
44     */
45     public String getCsvInfo() {
46         return winLoseInfo + ", "
47             + N + ", "
48             + difficulty + ", "
49             + playDateTime + "\n";
50     }
51
52     /**
53     * @return Convert Game to Object[]
54     */
55     public Object[] getTableRow() {
56         return new Object[] {RESULT_TEXT[winLoseInfo], N, DIFFICULTY_TEXT[difficulty],
57             ↪ playDateTime};
58     }
59
60     public void setWinLoseInfo(int winLoseInfo) {
61         this.winLoseInfo = winLoseInfo;
62     }
63
64     public int getWinLoseInfo() {
65         return winLoseInfo;
66     }
67
68     /**
69     * @param gameInfoAsCsv Game Information written in CSV format
70     * @return Parsed CSV Data as Game
71     */
72     public static Game parseGameInfo(String gameInfoAsCsv) {
73         StringTokenizer stringTokenizer = new StringTokenizer(gameInfoAsCsv, ",");
74         int winLoseInfo = Integer.parseInt(stringTokenizer.nextToken().trim());
75         int N = Integer.parseInt(stringTokenizer.nextToken().trim());
76         int difficulty = Integer.parseInt(stringTokenizer.nextToken().trim());
77         String playDateTime = stringTokenizer.nextToken();
78
79         return new Game(winLoseInfo, N, difficulty, playDateTime);
80     }
81 }
```

2.3 User.java

유저 및 컴퓨터의 빙고판, 빙고 갯수, 단어 목록 등을 묶어 편하게 관리하기 위한 Class이다.

public User(ArrayList<Word> fullWordList, **int** N): User 클래스의 생성자로써, fullWordList 단어장에서 받아온 전체 단어 리스트와 N 을 파라미터로 받아 N^2 크기의 빙고판을 만든다. Collections.shuffle() 메소드를 이용하여 fullWordList를 랜덤하게 섞어 맨 앞 N^2 개의 단어를 선택하는 방식으로 해당 기능을 구현했다.

public void constructBingoPanel(): Swing의 GridLayout을 사용하여 빙고판 레이아웃을 만든다. 빙고판을 다시 그릴 때마다 호출된다.

public boolean isSelectable(): 플레이어의 빙고판에 선택할 수 있는 단어가 남았는지 여부를 반환한다.

public void updateBingoCount(): 플레이어의 빙고판에서 빙고의 갯수를 계산한다.

public Word selectWordToWin(**int** difficulty, ArrayList<Word> comparableList): 컴퓨터가 사용하는 메소드로써, 이기기 위해 선택해야 할 단어를 반환한다. 컴퓨터가 단어를 선택하는 알고리즘은 다음과 같다.

난이도	선택 알고리즘
쉬움	컴퓨터의 빙고판에서 선택되지 않은 단어 중 하나를 랜덤으로 뽑아 반환한다.
중간	가로, 세로, 대각선 중 가장 빙고에 가까운 줄에서 선택되지 않은 단어를 골라 반환한다.
어려움	만약 N 이 홀수이고, 가장 처음 선택되는 단어일 경우 빙고판 한가운데를 먼저 선택해 반환한다. N 이 홀수가 아니거나 가장 처음 선택되는 단어가 아니라면 가로, 세로, 대각선 중 가장 빙고에 가까운 줄에서 선택되지 않은 단어를 반환하되, 1/2의 확률로 사용자의 빙고판에 없는 단어를 우선적으로 골라 반환한다. 만약 사용자의 빙고판에 있는 단어더라도, 그 단어만 선택하면 빙고가 완성될 경우, 그 단어를 반환한다.

Source Code

```

1 package ywrhee.project;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.util.*;
6 import java.util.List;
7
8 public class User {
9     private Word[][] bingoBoard;
10    private int bingoCount = 0;
11    private ArrayList<Word> wordList;
12    private JLabel[][] labels;
13    private JPanel bingoPanel;
14    private int N;
15
16    public User(ArrayList<Word> fullWordList, int N) {

```

```

17     this.N = N;
18     bingoBoard = new Word[N][N];
19     labels = new JLabel[N][N];
20
21     Collections.shuffle(fullWordList);
22     wordList = new ArrayList<>(fullWordList.stream().limit(N * N).toList());
23
24     for (int i = 0, k = 0; i < N; i++) {
25         for (int j = 0; j < N; j++, k++) {
26             bingoBoard[i][j] = wordList.get(k);
27         }
28     }
29
30 }
31
32
33 /**
34  * Make a bingo board with Word bingoBoard[][]
35  */
36 public void constructBingoPanel() {
37     bingoPanel = new JPanel();
38     bingoPanel.setLayout(new GridLayout(N, N, -1, -1));
39     for (int i = 0; i < N; i++) {
40         for (int j = 0; j < N; j++) {
41             labels[i][j] = new JLabel(bingoBoard[i][j].getEnglish(), SwingConstants.CENTER);
42             labels[i][j].setBorder(BorderFactory.createLineBorder(Color.decode("#777777")));
43             labels[i][j].setPreferredSize(new Dimension(120, 120));
44
45             if (bingoBoard[i][j].isChecked()) {
46                 labels[i][j].setOpaque(true);
47                 labels[i][j].setBackground(Color.decode("#3a4d43")); // #036b3f 1f5540
48                 labels[i][j].setForeground(Color.decode("#dddddd"));
49             }
50
51             bingoPanel.add(labels[i][j]);
52         }
53     }
54 }
55
56 public JPanel getBingoPanel() {
57     return bingoPanel;
58 }
59
60 public ArrayList<Word> getWordList() {
61     return wordList;
62 }
63
64 public boolean isSelectable() {
65     return getWordList().stream().filter(it -> !it.isChecked()).findAny().orElse(null) !=
        ↪ null;
66 }
67
68 /**
69  * Calculate the bingo count of bingo board and update to the bingoCount
70  */
71 public void updateBingoCount() {
72     bingoCount = 0;
73
74     int sum1 = 0, sum2 = 0;
75     for (int i = 0; i < N; i++) {
76         int sumV = 0, sumH = 0;

```

```

77         for (int j = 0; j < N; j++) {
78             sumV += bingoBoard[i][j].isChecked() ? 1 : 0;
79             sumH += bingoBoard[j][i].isChecked() ? 1 : 0;
80         }
81         if (sumV == N) bingoCount++;
82         if (sumH == N) bingoCount++;
83         sum1 += bingoBoard[i][i].isChecked() ? 1 : 0;
84         sum2 += bingoBoard[N - i - 1][i].isChecked() ? 1 : 0;
85     }
86
87     if (sum1 == N) bingoCount++;
88     if (sum2 == N) bingoCount++;
89 }
90
91 public int getBingoCount() {
92     return bingoCount;
93 }
94
95 /**
96  * Choose the Word to win according to difficulty of AI
97  * @param difficulty AI's difficulty
98  * @param comparisonList Words in the comparisonList has a 50% chance of not being selected
99  * @return Word chosen by AI
100 */
101 public Word selectWordToWin(int difficulty, ArrayList<Word> comparisonList) {
102     // Returns Random Word in wordList which is not checked when Difficulty == Easy
103     if (difficulty == Game.EASY) {
104         Collections.shuffle(wordList);
105         return wordList.stream().filter(it -> !it.isChecked()).toList().get(0);
106     }
107
108     if (difficulty == Game.HARD && N % 2 == 1 &&
109         ↪ wordList.stream().filter(Word::isChecked).toList().isEmpty()) {
110         return bingoBoard[N / 2][N / 2];
111     }
112
113     int[] hWeights = new int[N];
114     Arrays.fill(hWeights, 0);
115     int[] vWeights = new int[N];
116     Arrays.fill(vWeights, 0);
117     int d1Weights = 0, d2Weights = 0;
118     Arrays.stream(bingoBoard).forEach(it -> Arrays.stream(it).forEach(it2 ->
119         ↪ it2.setWeight(0)));
120
121     for (int i = 0; i < N; i++) {
122         for (int j = 0; j < N; j++) {
123             hWeights[i] += bingoBoard[i][j].isChecked() ? 1 : 0;
124             vWeights[j] += bingoBoard[j][i].isChecked() ? 1 : 0;
125         }
126         d1Weights += bingoBoard[i][i].isChecked() ? 1 : 0;
127         d2Weights += bingoBoard[N - i - 1][i].isChecked() ? 1 : 0;
128     }
129
130     for (int i = 0; i < N; i++) {
131         for (int j = 0; j < N; j++) {
132             bingoBoard[i][j].setWeight(Integer.max(bingoBoard[i][j].getWeight(),
133                 ↪ hWeights[i]));
134             bingoBoard[j][i].setWeight(Integer.max(bingoBoard[j][i].getWeight(),
135                 ↪ vWeights[j]));
136         }
137     }

```



```

133         bingoBoard[i][i].setWeight(Integer.max(bingoBoard[i][i].getWeight(), d1Weights));
134         bingoBoard[N - i - 1][i].setWeight(Integer.max(bingoBoard[N - i - 1][i].getWeight(),
135             ↪ d2Weights));
136     }
137     List<Word> bestSelections = wordList.stream()
138         .filter(it -> !it.isChecked())
139         ↪ .sorted(Comparator.comparingInt(Word::getWeight).reversed())
140         .toList();
141
142     // Returns first item in bestSelection when Difficulty == Normal
143     if (difficulty == Game.NORMAL)
144         return bestSelections.get(0);
145
146     // Returns first item in bestSelection which is not in comparisonList when Difficulty ==
147     ↪ Hard
148     if (bestSelections.stream().anyMatch(it -> it.getWeight() == N - 1))
149         return bestSelections.stream().filter(it -> it.getWeight() == N -
150             ↪ 1).toList().get(0);
151
152     if (Math.random() < 0.5)
153         return bestSelections.get(0);
154     return bestSelections.stream().filter(it ->
155         ↪ !comparisonList.contains(it)).findFirst().orElse(bestSelections.get(0));

```

2.4 Word.java

단어의 영어, 한국어 뜻, 빙고판에서 선택 여부, 이 단어를 선택했을 경우 빙고가 될 확률을 묶어 편하게 관리하기 위한 Class이다. 단어장 파일에서 단어를 읽어오는 **static** 메소드가 포함되어 있다.

public static ArrayList<Word> **getWordList**(File file): 단어장 파일에서 영단어와 한국어 뜻을 읽어와 Word 클래스 형태로 바꾸어 리스트에 넣어 반환한다. 인스턴스를 생성하지 않아도 바로 쓸 수 있게 **static** 메소드로 만들었다.

Source Code

```

1  package ywrhee.project;
2
3  import java.io.File;
4  import java.io.FileNotFoundException;
5  import java.util.ArrayList;
6  import java.util.Scanner;
7
8  public class Word {
9      private boolean isChecked = false;
10     private String english;
11     private String korean;
12
13     private int weight;
14
15     public Word(String english, String korean) {
16         this.english = english;
17         this.korean = korean;
18     }
19
20     public String getEnglish() {
21         return english;
22     }
23
24     public boolean isChecked() {
25         return isChecked;
26     }
27
28     public void setChecked(boolean isChecked) {
29         this.isChecked = isChecked;
30     }
31
32     public int getWeight() {
33         return weight;
34     }
35
36     public void setWeight(int weight) {
37         this.weight = weight;
38     }
39
40     @Override
41     public String toString() {
42         return english + "(" + korean + ")";
43     }
44
45     public static ArrayList<Word> getWordList(File file) {
46         ArrayList<Word> wordList = new ArrayList<>();
47         try (Scanner fileScanner = new Scanner(file)){

```

```
48         while(fileScanner.hasNextLine()) {
49             String str = fileScanner.nextLine();
50             String[] wordInfo = str.split("\\t");
51             String english = wordInfo[0].trim();
52             String korean = wordInfo[1].trim();
53             wordList.add(new Word(english, korean));
54         }
55
56     } catch (FileNotFoundException e) {
57         e.printStackTrace();
58     }
59
60     return wordList;
61 }
62 }
```

2.5 StatisticUtil.java

통계를 편리하게 관리하기 위한 유틸리티 Class이다. 인스턴스를 생성하지 않아도 바로 쓸 수 있게 모두 **static** 메소드로 만들었다.

public static void init(): CSV 형식으로 쓰여진 통계 파일에서 통계를 파싱해 statisticList에 저장하고, 통계 정보를 기록할 수 있는 FileWriter를 초기화한다.

public static void addToStatisticList(Game game): 게임 데이터를 statisticList에 저장한다.

public static void writeStatistic(Game game): CSV 형식으로 변환된 게임 데이터를 FileWriter로 이어쓴 후, 버퍼를 Flush한다.

public static void closeStatisticWriterStream(): FileWriter의 InputStream을 close한다.

Source Code

```

1 package ywrhee.project;
2
3 import java.io.File;
4 import java.io.FileWriter;
5 import java.io.IOException;
6 import java.util.ArrayList;
7 import java.util.Scanner;
8
9 public class StatisticUtil {
10
11     private static ArrayList<Game> statisticList;
12     private static FileWriter statisticWriter;
13
14     /**
15      * Get saved statistic information from statistics.txt and save it to statisticList
16      */
17     public static void init() {
18         statisticList = new ArrayList<>();
19         File statisticFile;
20         try {
21             statisticFile = new File(System.getProperty("user.dir") + "/statistics.txt");
22
23             if (!statisticFile.exists()) statisticFile.createNewFile();
24
25             statisticWriter = new FileWriter(statisticFile, true);
26
27             Scanner statisticScanner = new Scanner(statisticFile);
28             while(statisticScanner.hasNextLine()) {
29                 String str = statisticScanner.nextLine();
30                 statisticList.add(Game.parseGameInfo(str));
31             }
32
33         } catch (IOException e) {
34             e.printStackTrace();
35         }
36     }
37
38     public static ArrayList<Game> getStatisticList() {

```

```
39     return statisticList;
40 }
41
42 public static void addToStatisticList(Game game) {
43     statisticList.add(game);
44 }
45
46 /**
47  * Save Game information into the file as CSV formatted string
48  * @param game Game object to save in the file
49  */
50 public static void writeStatistic(Game game) {
51     try {
52         statisticWriter.write(game.getCsvInfo());
53         statisticWriter.flush();
54     } catch (IOException e) {
55         e.printStackTrace();
56     }
57 }
58
59 public static void closeStatisticWriterStream() {
60     try {
61         statisticWriter.close();
62     } catch (IOException e) {
63         e.printStackTrace();
64     }
65 }
66 }
```

2.6 StartFrame.java

게임을 시작하기 전 필요한 정보인 단어장 파일, AI 수준, N 을 입력받고, 저장된 통계 정보를 출력하는 Frame이다.

public StartFrame(String title, ArrayList<Game> statisticList): StartFrame의 생성자로서, 윈도우의 타이틀과 통계 목록을 파라미터로 받아 클래스 내에서 사용 가능하도록 한다. 게임을 시작하기 위해 필요한 정보가 입력될 때까지 메인 스레드에서 입력 받은 값을 접근할 수 없도록 **this.wait()**을 사용한다.

private void initUI(): 게임에 필요한 정보를 수집할 UI를 구성한다. 생성자를 통해 넘겨받은 통계 정보를 이용해 화면에 승률을 표시하고, 클릭 시 StatisticDialog가 호출되도록 한다.

public void actionPerformed(ActionEvent e): ActionListener의 actionPerformed를 Override한 메소드로, 시작하기 버튼이 눌렸거나 N 을 입력받는 텍스트 필드에서 Enter 키가 눌렸을 때 호출된다. 선택된 단어장 파일의 존재 여부와, N 값의 유효성(단어장의 단어 갯수가 N^2 보다 큰지, $3 \leq N \leq 10$ 인지)을 체크하여 정상적인 실행이 불가능할 경우 경고 Dialog를 띄운다. 만약 정상적으로 게임이 실행 가능하면 메인 스레드에서 GameFrame을 호출할 수 있도록 **this.notify()** 해주고 윈도우를 dispose()한다.

private int countLine(File file): 파라미터로 전달받은 단어장 파일의 단어 갯수를 반환한다.

Source Code

```

1 package ywrhee.project;
2
3 import javax.swing.*;
4 import javax.swing.filechooser.FileNameExtensionFilter;
5 import java.awt.*;
6 import java.awt.event.*;
7 import java.io.File;
8 import java.io.FileNotFoundException;
9 import java.io.IOException;
10 import java.nio.file.Files;
11 import java.util.ArrayList;
12
13 public class StartFrame extends JFrame implements ActionListener {
14
15     private Container frame = this.getContentPane();
16     private JPanel upperPanel;
17     private JLabel title, name, nLabel;
18     private JTextField nTextField, filePathTextField;
19     private JButton goButton;
20     private JRadioButton[] difficultyRadio = new JRadioButton[3];
21     private ButtonGroup difficultyGroup = new ButtonGroup();
22     private File selectedFile = new File("src/main/java/ywrhee/project/wordList.txt");
23     private ArrayList<Game> statisticList;
24
25     private int N = 0;
26
27     public StartFrame(String title, ArrayList<Game> statisticList) {
28         super(title);
29         this.statisticList = statisticList;

```

```

30
31     this.setSize(400, 600);
32     this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
33     this.setLocationRelativeTo(null);
34     this.setResizable(false);
35
36     initUI();
37
38     this.setVisible(true);
39
40     synchronized(this){
41         try {
42             this.wait();
43         } catch (InterruptedException e){
44             e.printStackTrace();
45         }
46     }
47 }
48
49 private void initUI() {
50     upperPanel = new JPanel();
51     upperPanel.setLayout(new BorderLayout());
52     upperPanel.setBorder(BorderFactory.createEmptyBorder(60, 0, 0, 0));
53
54     title = new JLabel("빙고 게임", SwingConstants.CENTER);
55     title.setFont(new Font("GangwonEduPower", Font.BOLD, 38));
56     upperPanel.add(title, BorderLayout.NORTH);
57
58     name = new JLabel("컴퓨터공학부 202211342 이율원", SwingConstants.CENTER);
59     name.setBorder(BorderFactory.createEmptyBorder(8, 0, 0, 0));
60     upperPanel.add(name, BorderLayout.SOUTH);
61
62     frame.add(upperPanel, BorderLayout.NORTH);
63
64     int gamePlayCount = statisticList.size();
65     int victoryCount = (int) statisticList.stream().filter(it -> it.getWinLoseInfo() ==
        ↳ Game.VICTORY).count();
66     int drawCount = (int) statisticList.stream().filter(it -> it.getWinLoseInfo() ==
        ↳ Game.DRAW).count();
67     int defeatCount = (int) statisticList.stream().filter(it -> it.getWinLoseInfo() ==
        ↳ Game.DEFEAT).count();
68     double winRatio = (double) victoryCount / (victoryCount + defeatCount) * 100;
69     winRatio = (double) Math.round(winRatio * 100) / 100;
70
71     String winningInfoStr = gamePlayCount + "전 " + victoryCount + "승 " + drawCount + "무 " +
        ↳ defeatCount + "패 | 승률 " + winRatio + "%";
72
73     JLabel winningInfo = new JLabel(winningInfoStr, SwingConstants.CENTER);
74     winningInfo.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
75     winningInfo.addMouseListener(new MouseAdapter() {
76         @Override
77         public void mouseClicked(MouseEvent e) {
78             new StatisticDialog(StartFrame.this, "통계", statisticList, winningInfoStr);
79         }
80
81         @Override
82         public void mouseEntered(MouseEvent e) {
83             winningInfo.setText("<html><u>" + winningInfoStr + "</u></html>");
84         }
85
86         @Override

```

```

87         public void mouseExited(MouseEvent e) {
88             winningInfo.setText(winningInfoStr);
89         }
90     });
91     frame.add(winningInfo, BorderLayout.CENTER);
92
93     JPanel optionPanel = new JPanel();
94     optionPanel.setBorder(BorderFactory.createEmptyBorder(0, 65, 50, 65));
95     optionPanel.setLayout(new GridBagLayout());
96     GridBagConstraints gridBagConstraints = new GridBagConstraints();
97     gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
98     gridBagConstraints.gridwidth = 2;
99     gridBagConstraints.gridx = 0;
100    gridBagConstraints.gridy = 0;
101    gridBagConstraints.weightx = 1;
102
103    JLabel filePathLabel = new JLabel("단어장 파일");
104    filePathLabel.setBorder(BorderFactory.createEmptyBorder(0, 0, 8, 0));
105    optionPanel.add(filePathLabel, gridBagConstraints);
106
107    filePathTextField = new JTextField(selectedFile.getName());
108    filePathTextField.setEditable(false);
109    filePathTextField.setMargin(new Insets(10, 10, 10, 10));
110    gridBagConstraints.gridwidth = 1;
111    gridBagConstraints.gridy++;
112    optionPanel.add(filePathTextField, gridBagConstraints);
113
114    JButton browseButton = new JButton("...");
115    browseButton.setMargin(new Insets(10, 10, 10, 10));
116    browseButton.addActionListener(e -> {
117        JFileChooser jFileChooser = new JFileChooser();
118        FileNameExtensionFilter filter = new FileNameExtensionFilter("텍스트 파일", "txt");
119        jFileChooser.setFileFilter(filter);
120        if (jFileChooser.showOpenDialog(null) == 0) {
121            filePathTextField.setText(jFileChooser.getSelectedFile().getName());
122            selectedFile = jFileChooser.getSelectedFile();
123        }
124    });
125    gridBagConstraints.gridx++;
126    gridBagConstraints.weightx = 0;
127    optionPanel.add(browseButton, gridBagConstraints);
128    gridBagConstraints.gridx--;
129    gridBagConstraints.gridwidth = 2;
130    gridBagConstraints.weightx = 1;
131
132    // Difficulty Label
133    JLabel difficultyLabel = new JLabel("AI 수준");
134    difficultyLabel.setBorder(BorderFactory.createEmptyBorder(16, 0, 8, 0));
135    gridBagConstraints.gridy++;
136    optionPanel.add(difficultyLabel, gridBagConstraints);
137
138    // Difficulty Radio Buttons
139    JPanel difficultyPanel = new JPanel();
140    difficultyPanel.setLayout(new GridLayout(1, 3));
141    for (int i = 0; i < 3; i++) {
142        difficultyRadio[i] = new JRadioButton(Game.DIFFICULTY_TEXT[i]);
143        difficultyGroup.add(difficultyRadio[i]);
144        difficultyPanel.add(difficultyRadio[i]);
145    }
146    difficultyRadio[Game.EASY].setSelected(true);
147    gridBagConstraints.gridy++;

```



```

148     optionPanel.add(difficultyPanel, gridBagConstraints);
149
150     // N Label
151     nLabel = new JLabel("빙고판 크기");
152     nLabel.setBorder(BorderFactory.createEmptyBorder(16, 0, 8, 0));
153     gridBagConstraints.gridy++;
154     optionPanel.add(nLabel, gridBagConstraints);
155
156     // N Input Text Field
157     nTextField = new JTextField();
158     nTextField.setMargin(new Insets(10, 10, 10, 10));
159     nTextField.addActionListener(this);
160     gridBagConstraints.gridy++;
161     optionPanel.add(nTextField, gridBagConstraints);
162
163     // Space Between Text Field and Button
164     gridBagConstraints.insets = new Insets(35, 0, 0, 0);
165
166     // Start Button
167     goButton = new JButton("시작하기 →");
168     goButton.setMargin(new Insets(10, 10, 10, 10));
169     goButton.setFont(goButton.getFont());
170     goButton.addActionListener(this);
171     gridBagConstraints.gridy++;
172     optionPanel.add(goButton, gridBagConstraints);
173
174     frame.add(optionPanel, BorderLayout.SOUTH);
175 }
176
177 @Override
178 public void actionPerformed(ActionEvent e) {
179     if (e.getSource() == goButton || e.getSource() == nTextField) {
180         try {
181             if (!selectedFile.exists()) throw new FileNotFoundException();
182
183             N = Integer.parseInt(nTextField.getText());
184             nTextField.setText("");
185             if (N < 3 || N > 10) {
186                 N = 0;
187                 throw new NumberFormatException();
188             }
189
190             int numberOfWords = countLine(selectedFile);
191             if (N * N > numberOfWords) {
192                 String dialogMessage = "단어의 수가 N2보다 작습니다.\n단어가 " + (N * N) + "개 이상인
193                     → 단어장을 사용하거나, N을 " + (int) Math.sqrt(numberOfWords) + "이하의 값으로 입력
194                     → 해주세요.";
195                 JOptionPane.showMessageDialog(this, dialogMessage, "오류",
196                     JOptionPane.ERROR_MESSAGE);
197                 return;
198             }
199
200             synchronized(this){
201                 this.notify();
202             }
203
204             this.setVisible(false);
205             this.dispose();
206         } catch (NumberFormatException ex) {
207             JOptionPane.showMessageDialog(this, "N을 3과 10 사이의 정수 값으로 입력해주세요", "경고",
208                 JOptionPane.WARNING_MESSAGE);
209         }
210     }
211 }

```

```

205         } catch (FileNotFoundException ex) {
206             JOptionPane.showMessageDialog(this, "파일을 찾을 수 없습니다. 다른 파일을 선택해주세요.", "
                ↳ 경고", JOptionPane.WARNING_MESSAGE);
207         }
208     }
209 }
210
211 private int countLine(File file) {
212     int lines = 0;
213     try {
214         lines = (int) Files.lines(file.toPath()).count();
215     } catch (IOException e) {
216         e.printStackTrace();
217     }
218
219     return lines;
220 }
221
222 public int getN() {
223     return N;
224 }
225
226 public File getSelectedFile() {
227     return selectedFile;
228 }
229
230 public int getDifficulty() {
231     for (int i = 0; i < 3; i++) {
232         if (difficultyRadio[i].isSelected()) return i;
233     }
234
235     return -1;
236 }
237 }

```

2.7 GameFrame.java

게임 화면을 출력하는 Frame이다.

public GameFrame(String title, User user, User computer, **int** N, **int** difficulty):

GameFrame의 생성자로서, 윈도우 타이틀, 유저 및 컴퓨터의 플레이 데이터, N 값과 AI 수준을 파라미터로 입력받아 Class 내에서 사용 가능하도록 한다. 게임 데이터가 완성될 때까지 메인 스레드에서 GameFrame의 데이터를 접근할 수 없도록 **this.wait()**을 사용한다.

private void initUI(): 게임 플레이 UI를 구성한다.

public void actionPerformed(ActionEvent e): ActionListener의 actionPerformed를 Override한 메소드로, 단어 입력 버튼이 눌렸거나 단어를 입력받는 텍스트 필드에서 Enter키가 눌렸을 때 호출된다. 만약 입력받은 단어가 컴퓨터 또는 유저의 빙고판에 존재할 경우, 체크해주고 빙고판을 다시 그리도록 updateBingoBoard()를 호출하고, 선택된 단어의 뜻을 출력한다. 이후 computerTurn()이 호출되어 컴퓨터의 턴으로 넘어간다.

private void computerTurn(): 컴퓨터의 빙고판에 선택 가능한 단어가 남아있을 경우, User 클래스의 selectWordToWin() 메소드를 이용하여 단어를 선택한다. 바로 선택하여 유저 턴으로 넘어가는 모습이 어색하지 않도록 1000ms의 딜레이 이후에 컴퓨터가 선택한 단어가 출력되고, 빙고판을 다시 그리도록 updateBingoBoard()를 호출한다. 이후 승패 여부를 가리도록 checkWinner() 메소드를 호출한다.

private void checkWinner(): 한 턴이 끝난 후 호출되는 메소드로써, 유저와 컴퓨터의 빙고판을 체크하여 빙고의 갯수가 한 개이상 많은 플레이어를 승리로 처리한다. 빙고판이 모두 열렸지만 승패가 결정되지 않은 경우 무승부로 처리한다. 이후 결정된 승패 정보를 Dialog로 출력하고, 게임 화면을 종료하도록 closeGameFrame()을 호출한다.

private void updateBingoBoard(): 유저와 컴퓨터의 빙고판에 선택된 단어를 표시하여 다시 그린다.

private void closeGameFrame(): 메인 스레드에서 GameFrame의 데이터를 접근할 수 있도록 **this.notify()**하고 윈도우를 dispose()한다.

Source Code

```

1 package ywrhee.project;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.util.ArrayList;
8 import java.util.HashSet;
9
10 public class GameFrame extends JFrame implements ActionListener {
11
12     private JPanel containerPanel, bottomPanel;
13     private JTextField userInput;
14     private JTextArea logArea;
15     private JButton inputButton;

```

```

16     private JLabel turnLabel;
17     private User user, computer;
18     private HashSet<Word> wordsInBingoBoard = new HashSet<>();
19     private int difficulty, winLoseInfo;
20
21     public GameFrame(String title, User user, User computer, int N, int difficulty) {
22         super(title);
23         this.user = user;
24         this.computer = computer;
25         this.difficulty = difficulty;
26         wordsInBingoBoard.addAll(user.getWordList());
27         wordsInBingoBoard.addAll(computer.getWordList());
28
29         this.setSize(N * 120 * 2 + 400, N * 120 + 200);
30         this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
31         this.setLocationRelativeTo(null);
32         this.setResizable(false);
33
34         initUI();
35
36         this.add(containerPanel);
37         this.setVisible(true);
38
39         synchronized(this){
40             try {
41                 this.wait();
42             } catch (InterruptedException e){
43                 e.printStackTrace();
44             }
45         }
46     }
47
48     private void initUI() {
49         containerPanel = new JPanel();
50         containerPanel.setLayout(new BorderLayout());
51         containerPanel.setBorder(BorderFactory.createEmptyBorder(25, 25, 25, 25));
52
53         turnLabel = new JLabel("<html><b>유저 차례</b> | 유저 빙고: " + user.getBingoCount() + "개,
54             ↪ 컴퓨터 빙고: " + computer.getBingoCount() + "개</html>");
55         turnLabel.setFont(turnLabel.getFont().deriveFont(20f));
56         turnLabel.setBorder(BorderFactory.createEmptyBorder(0, 0, 25, 0));
57         containerPanel.add(turnLabel, BorderLayout.NORTH);
58
59         user.constructBingoPanel();
60         containerPanel.add(user.getBingoPanel(), BorderLayout.WEST);
61
62         computer.constructBingoPanel();
63         containerPanel.add(computer.getBingoPanel(), BorderLayout.EAST);
64
65         logArea = new JTextArea(10, 10);
66         logArea.setEnabled(false);
67         logArea.setDisabledTextColor(Color.decode("#777777"));
68         logArea.setAutoscrolls(false);
69         logArea.setLineWrap(true);
70         logArea.setWrapStyleWord(true);
71         containerPanel.add(new JScrollPane(logArea), BorderLayout.CENTER);
72
73         bottomPanel = new JPanel();
74         bottomPanel.setBorder(BorderFactory.createEmptyBorder(25, 0, 0, 0));
75
76         userInput = new JTextField("", 12);

```

```

76     userInput.setMargin(new Insets(10, 10, 10, 10));
77     userInput.addActionListener(this);
78     bottomPanel.add(userInput);
79
80     inputButton = new JButton("→");
81     inputButton.setMargin(new Insets(10, 10, 10, 10));
82     inputButton.addActionListener(this);
83     bottomPanel.add(inputButton);
84
85     containerPanel.add(bottomPanel, BorderLayout.SOUTH);
86
87 }
88
89 @Override
90 public void actionPerformed(ActionEvent e) {
91     if (e.getSource() == inputButton || e.getSource() == userInput) {
92         ArrayList<Word> filteredList = new ArrayList<>(wordsInBingoBoard.stream().filter(it
          ↪ -> it.getEnglish().equals(userInput.getText().trim())).toList());
93
94         if (!filteredList.isEmpty()) {
95             filteredList.get(0).setChecked(true);
96             logArea.setText((logArea.getText() + "\n" + filteredList.get(0) + "이(가) 선택되었
          ↪ 습니다.").trim());
97         }
98
99         updateBingoBoard();
100
101         userInput.setText("");
102         userInput.setEditable(false);
103         userInput.removeActionListener(this);
104         inputButton.setEnabled(false);
105
106         computerTurn();
107     }
108 }
109
110 private void computerTurn() {
111     turnLabel.setText("<html><b>컴퓨터 차례</b> | 유저 빙고: " + user.getBingoCount() + "개, 컴퓨
          ↪ 터 빙고: " + computer.getBingoCount() + "개</html>");
112     Word selectedWord = null;
113     if (!computer.getWordList().stream().filter(it -> !it.isChecked()).toList().isEmpty()) {
114         selectedWord = computer.selectWordToWin(difficulty, user.getWordList());
115         selectedWord.setChecked(true);
116     }
117
118     Word finalSelectedWord = selectedWord;
119
120     Timer timer = new Timer(1000, new AbstractAction() {
121         @Override
122         public void actionPerformed(ActionEvent ae) {
123             if (finalSelectedWord != null)
124                 logArea.setText(logArea.getText() + "\n컴퓨터가 " + finalSelectedWord + "를 선택
          ↪ 하였습니다.");
125             updateBingoBoard();
126
127             checkWinner();
128
129             turnLabel.setText("<html><b>유저 차례</b> | 유저 빙고: " + user.getBingoCount() + "
          ↪ 개, 컴퓨터 빙고: " + computer.getBingoCount() + "개</html>");
130
131             inputButton.setEnabled(true);

```

```

132         userInput.addActionListener(GameFrame.this);
133         userInput.setEditable(true);
134     }
135 });
136 timer.setRepeats(false);
137 timer.start();
138 }
139
140 private void checkWinner() {
141     user.updateBingoCount();
142     computer.updateBingoCount();
143     turnLabel.setText("<html><b>컴퓨터 차례</b> | 유저 빙고: " + user.getBingoCount() + "개, 컴퓨
    ↪ 터 빙고: " + computer.getBingoCount() + "개</html>");
144
145     if (user.getBingoCount() > computer.getBingoCount()) {
146         winLoseInfo = Game.VICTORY;
147         JOptionPane.showMessageDialog(GameFrame.this, "당신이 이겼습니다.", "게임 종료",
    ↪ JOptionPane.INFORMATION_MESSAGE);
148         closeGameFrame();
149     } else if (user.getBingoCount() < computer.getBingoCount()) {
150         winLoseInfo = Game.DEFEAT;
151         JOptionPane.showMessageDialog(GameFrame.this, "컴퓨터가 이겼습니다.", "게임 종료",
    ↪ JOptionPane.ERROR_MESSAGE);
152         closeGameFrame();
153     } else if (!user.isSelectable() || !computer.isSelectable()) {
154         winLoseInfo = Game.DRAW;
155         JOptionPane.showMessageDialog(GameFrame.this, "무승부입니다.", "게임 종료",
    ↪ JOptionPane.WARNING_MESSAGE);
156         closeGameFrame();
157     }
158 }
159
160 /**
161  * Redraw bingo board
162  */
163 private void updateBingoBoard() {
164     containerPanel.remove(user.getBingoPanel());
165     user.constructBingoPanel();
166     containerPanel.add(user.getBingoPanel(), BorderLayout.WEST);
167
168     containerPanel.remove(computer.getBingoPanel());
169     computer.constructBingoPanel();
170     containerPanel.add(computer.getBingoPanel(), BorderLayout.EAST);
171
172     GameFrame.this.revalidate();
173     GameFrame.this.repaint();
174 }
175
176 /**
177  * Called when the window closes
178  */
179 private void closeGameFrame() {
180     synchronized(this){
181         this.notify();
182     }
183
184     this.setVisible(false);
185     this.dispose();
186 }
187
188 public int getWinLoseInfo() {

```

```
189         return winLoseInfo;
190     }
191
192 }
```

2.8 StatisticDialog.java

통계 정보를 테이블로 출력하는 Dialog이다.

public StatisticDialog(StartFrame owner, String title, ArrayList<Game> statisticList, String statisticInfo): StatisticDialog의 생성자로써, 부모 Frame, 윈도우 타이틀, 통계 데이터를 파라미터로 입력받아 Class 내에서 사용 가능하도록 한다.

private void initUI(): 통계 화면 UI를 구성한다. 통계 데이터를 JTable에 표시하고, 승패 여부에 따라 Row의 배경 색을 바꾼다.

Source Code

```

1  package ywrhee.project;
2
3  import javax.swing.*;
4  import javax.swing.table.DefaultTableCellRenderer;
5  import javax.swing.table.DefaultTableModel;
6  import java.awt.*;
7  import java.util.ArrayList;
8
9  public class StatisticDialog extends JDialog {
10     private Container dialog = this.getContentPane();
11     private StartFrame parent;
12
13     private JTable statisticTable;
14     private String columns[] = {"결과", "빙고판 크기", "AI 수준", "게임 시간"};
15
16     private ArrayList<Game> statisticList;
17     private String statisticInfo;
18
19     public StatisticDialog(StartFrame owner, String title, ArrayList<Game> statisticList, String
20     ↪ statisticInfo) {
21         super(owner, title, true);
22         parent = owner;
23         this.statisticList = statisticList;
24         this.statisticInfo = statisticInfo;
25
26         this.setSize(450, 600);
27         this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
28         this.setLocation(parent.getLocation().x + parent.getWidth(), parent.getLocation().y);
29
30         initUI();
31
32         this.setVisible(true);
33     }
34
35     private void initUI() {
36         JPanel upperPanel = new JPanel();
37         upperPanel.setLayout(new BorderLayout());
38         upperPanel.setBorder(BorderFactory.createEmptyBorder(60, 0, 0, 0));
39
40         JLabel title = new JLabel("통계", SwingConstants.CENTER);
41         title.setFont(new Font("GangwonEduPower", Font.BOLD, 38));
42
43         upperPanel.add(title, BorderLayout.NORTH);
44
45         JLabel winningInfo = new JLabel(statisticInfo, SwingConstants.CENTER);

```



```

45 winningInfo.setBorder(BorderFactory.createEmptyBorder(5, 0, 30, 0));
46 upperPanel.add(winningInfo, BorderLayout.SOUTH);
47
48 dialog.add(upperPanel, BorderLayout.NORTH);
49
50
51 DefaultTableModel tableModel = new DefaultTableModel(columns, 0) {
52     @Override
53     public boolean isCellEditable(int row, int column) {
54         return false;
55     }
56 };
57
58 statisticTable = new JTable(tableModel);
59
60 DefaultTableCellRenderer tableCellRenderer = new DefaultTableCellRenderer() {
61     @Override
62     public Component getTableCellRendererComponent(JTable table, Object value, boolean
63         ↳ isSelected, boolean hasFocus, int row, int column) {
64         final Component c = super.getTableCellRendererComponent(table, value,
65         ↳ isSelected, hasFocus, row, column);
66         switch (statisticList.get(row).getWinLoseInfo()) {
67             case Game.VICTORY -> c.setBackground(Color.decode("#3a4d43"));
68             case Game.DRAW -> c.setBackground(Color.decode("#4e4e3c"));
69             case Game.DEFEAT -> c.setBackground(Color.decode("#4e4040"));
70         }
71         return c;
72     }
73 };
74 tableCellRenderer.setHorizontalAlignment(JLabel.CENTER);
75
76 statisticTable.setDefaultRenderer(Object.class, tableCellRenderer);
77 statisticTable.setEnabled(false);
78
79 for (int i = 0; i < 3; i++)
80     statisticTable.getColumnModel().getColumn(i).setPreferredWidth(1);
81 statisticTable.setRowHeight(50);
82 statisticList.forEach(it -> tableModel.addRow(it.getTableRow()));
83
84 JScrollPane scrollPane = new JScrollPane(statisticTable);
85 dialog.add(scrollPane, BorderLayout.CENTER);
86
87 }

```

3 실행 결과

3.1 게임 전 화면

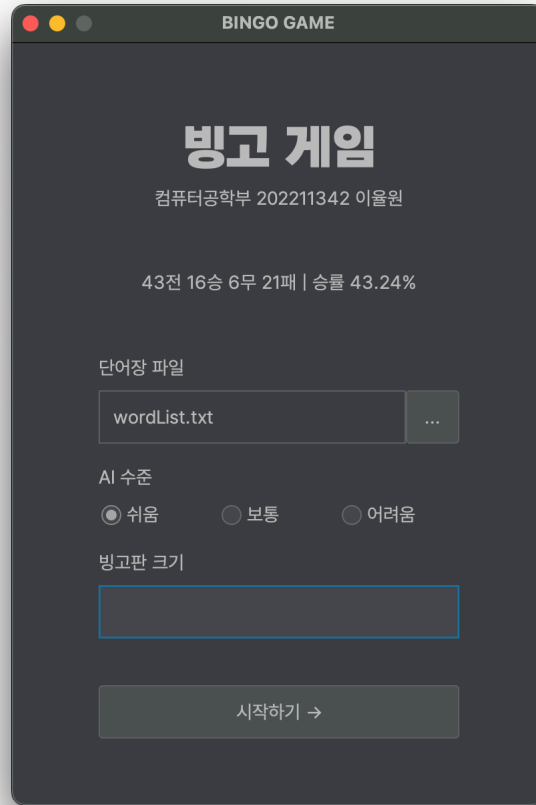
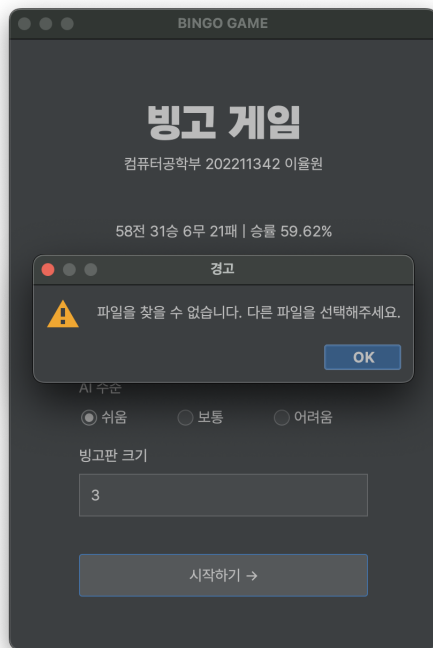


Fig. 1: 메인 화면

이 화면에서는 게임 플레이 전 필요한 정보를 수집하는 역할을 한다. 화면 상단의 승률 Label을 클릭해 전체 통계를 볼 수 있고, 단어장 파일은 '...'이 적힌 버튼을 눌러 선택할 수 있다. 정보를 다 선택한 이후 엔터를 누르거나 '시작하기 →' 버튼을 눌러 게임 화면에 진입할 수 있다. 만약 선택된 파일이 없는 파일이거나, 입력된 N 의 값이 $3 \leq N \leq 10$ 이 아닌 경우 경고창을 띄워주고, 단어장의 단어가 모자라 빙고판을 생성할 수 없는 경우, 입력받은 N 값으로 빙고판을 생성하기 위해 필요한 최소 단어 수와, 선택된 단어장으로 생성 가능한 최대 N 을 알려준다.



(a) 선택된 파일이 없는 경우

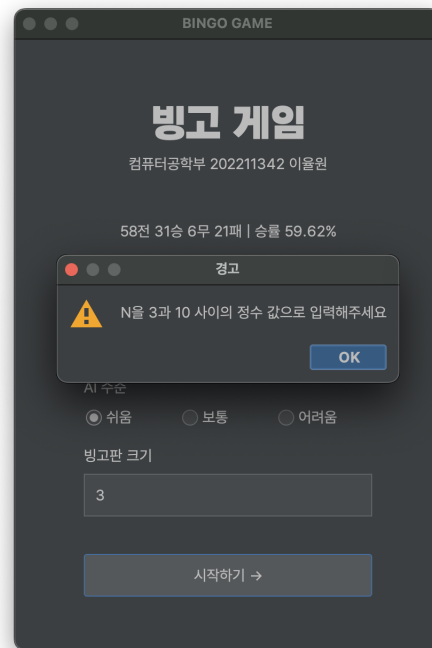
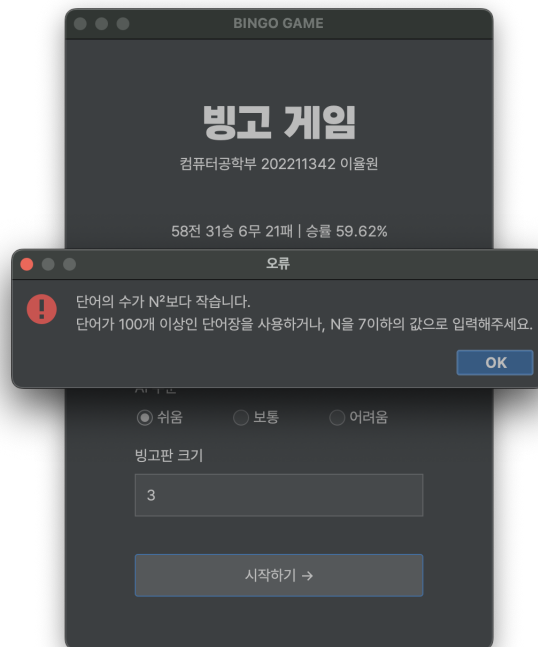
(b) $3 \leq N \leq 10$ 이 아닌 경우(c) $N = 10$ 일 때, 단어장의 단어가 모자라 빙고판을 생성할 수 없는 경우

Fig. 2: 사용자 입력 예외 처리

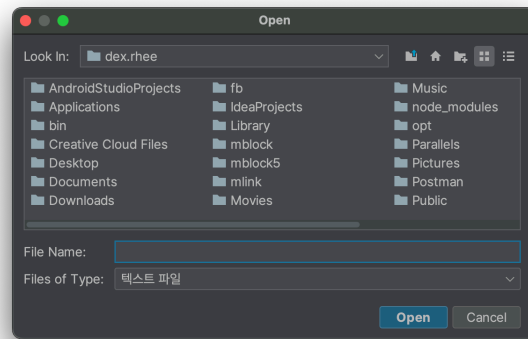


Fig. 3: 단어장 파일 선택 Dialog

단어장 파일을 선택할 수 있다. JFileChooser를 사용하여 구현하였다.



Fig. 4: 통계 Dialog

이 Dialog에서는 플레이한 모든 게임의 승패 여부, 빙고판 크기, AI 수준, 게임 시간을 볼 수 있다. Modal Dialog이기 때문에, 닫기 전까지 게임을 시작할 수 없다.

3.2 게임 중 화면

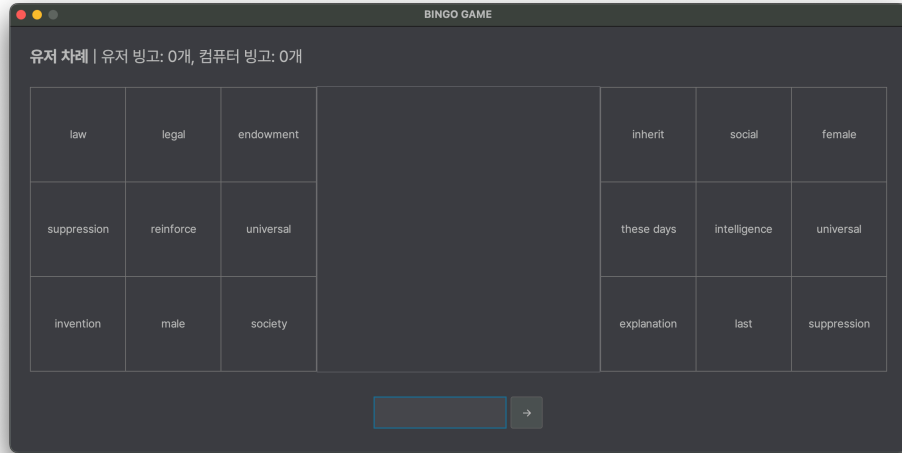


Fig. 5: $N = 3$ 일 때

N 값을 3으로 입력했을 때 구성된 게임 화면이다. 유저 차례일 경우 아래의 텍스트 필드 및 버튼이 Enable 되고, 원하는 단어를 입력 후 Enter키를 누르거나, ‘→’ 버튼을 눌러 선택할 수 있다. 컴퓨터 차례일 경우 아래의 텍스트 필드 및 버튼이 Disable 되고, 1초 후 컴퓨터가 선택한 단어가 화면에 표시된다.

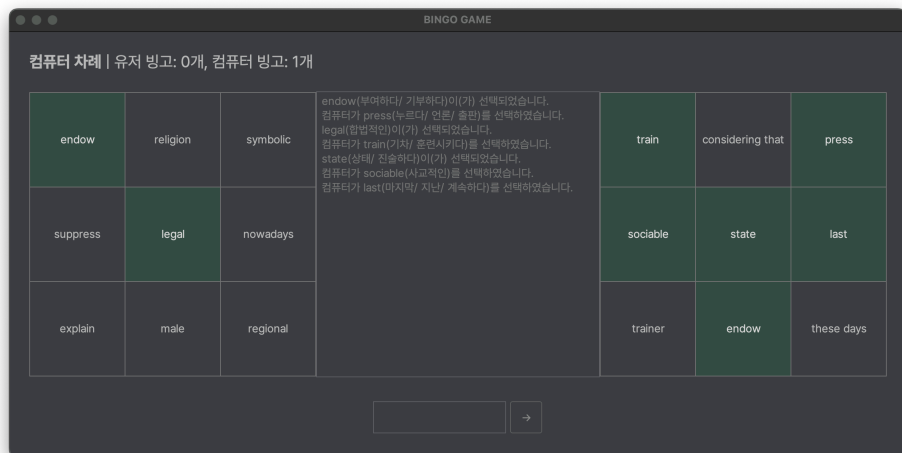
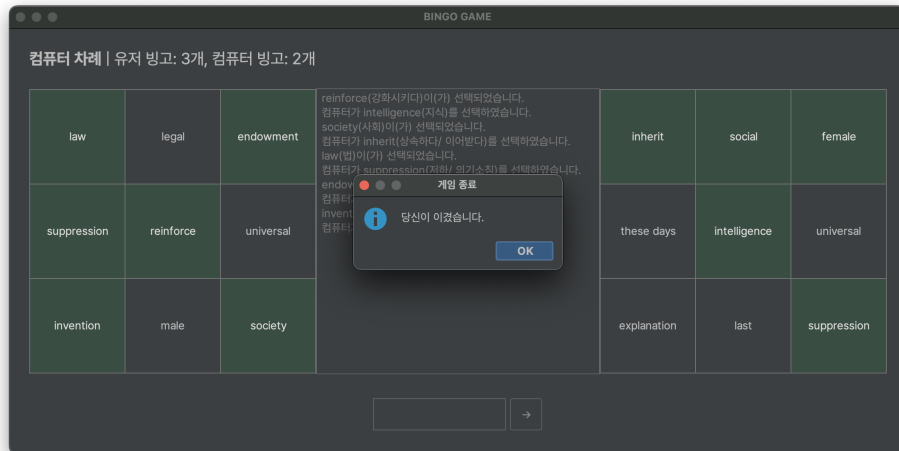
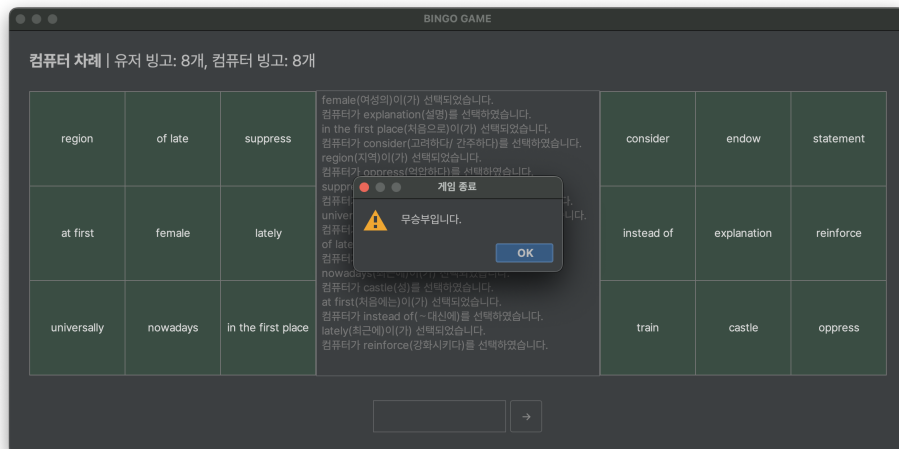


Fig. 6: $N = 3$ 일 때 플레이 중 화면

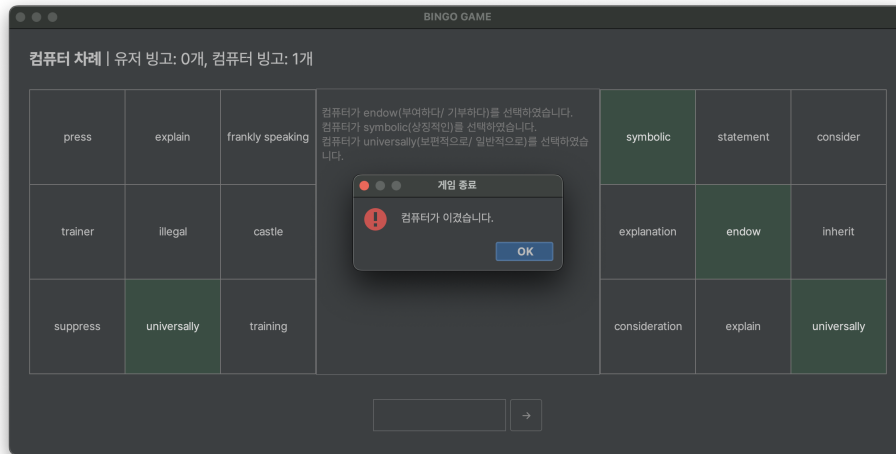
유저가 선택한 단어가 유저 또는 컴퓨터의 빙고판에 존재할 경우, 해당 단어가 있는 빙고칸의 배경이 초록색으로 바뀌고(이하 체크), 선택한 단어와 단어의 뜻이 출력된다. 컴퓨터가 단어를 선택한 경우, 유저의 빙고판에 있는 동일한 단어도 체크된다. 만약 유저가 빙고판에 없는 단어를 입력했을 경우, 바로 컴퓨터의 턴으로 넘어간다. 만약 유저가 입력한 단어가 유저의 빙고판에는 없지만 컴퓨터의 빙고판에는 존재할 경우, 컴퓨터의 빙고판에 있는 해당 단어가 체크된다. 한 턴이 끝날 때마다 빙고 여부를 체크하고, 빙고 개수가 하나 이상 많은 플레이어를 승리 처리한다.



(a) 이겼을 때

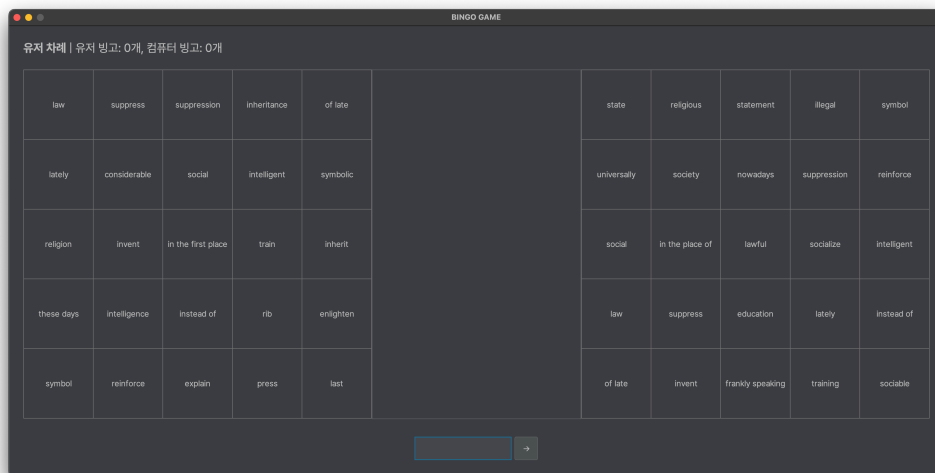


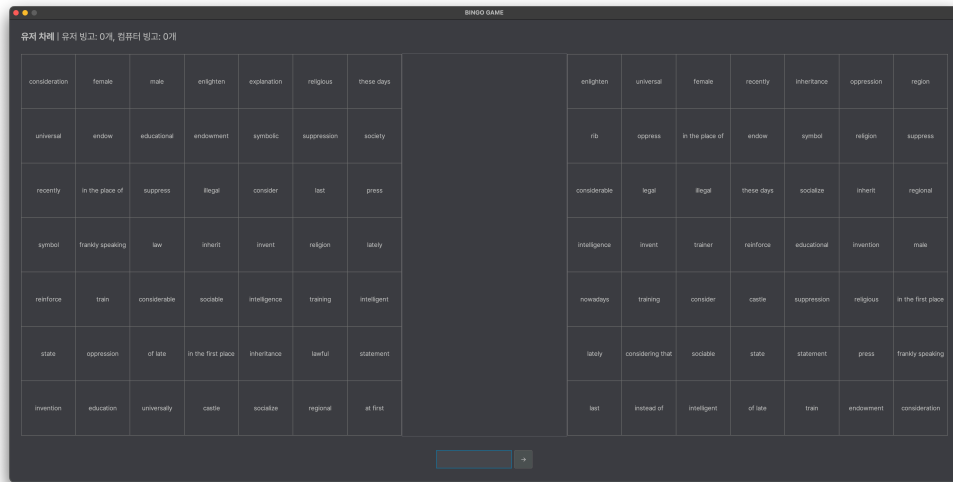
(b) 비겼을 때



(c) 족을 때

Fig. 7: 승패 처리 화면

(a) $N = 5$

(b) $N = 7$ Fig. 8: 여러 크기의 N 에 대한 빙고판

$3 \leq N \leq 10$ 사이의 모든 빙고판을 생성할 수 있다.

4 느낀점 및 토의 사항

빙고 게임의 특성 상 유저와 컴퓨터 두 플레이어의 동일한 종류의 데이터를 다루어야 할 경우가 많기 때문에, 소스코드의 중복되는 부분을 줄이고, 가독성을 높이고자 적지 않은 수의 Class와 그에 딸린 함수들로 쪼개어 코드를 작성했다. 이로 인해 유지보수의 관점에서 봐도 추후 수정과 기능 추가에 유리할 것으로 판단된다. 또한 통계 기능에서, 여러 기능을 하는 함수들을 묶어 유틸리티 Class로 만들어 보는 등 이번 프로젝트를 진행하며 다양한 시도를 해보았다.

Swing으로 프로그램을 만들어 본 것이 처음이라 레이아웃을 짜는 데 어려움을 겪었지만, 원하는대로 레이아웃을 짜기 위해 여러 Layout들을 사용해보고, Java Documentation에서 Swing이 제공하는 Component들의 메소드를 찾아보는 과정에서 큰 도움이 된 것 같다.

팀 프로젝트로 진행하여 협업하며 코드를 작성하는 편이 더 재미있겠다고 생각해 개인적으로 이번 프로젝트가 개인 프로젝트로 진행된 점이 아쉬웠지만, 그만큼 여러 다양한 시도들도 해볼 수 있어서 역량 향상에는 큰 도움이 된 것 같았다.

5 기타

5.1 개발 환경

이 프로젝트를 개발하는 과정에서, 운영체제는 macOS 13.1 Ventura를 사용하였고, IDE는 JetBrains사의 IntelliJ IDEA 2022.2.3 버전을 사용하였다.

JDK는 Temurin의 OpenJDK를 사용하였고, 버전 정보는 다음과 같다.

```
openjdk 17.0.5 2022-10-18
```

```
OpenJDK Runtime Environment Temurin-17.0.5+8 (build 17.0.5+8)
```

```
OpenJDK 64-Bit Server VM Temurin-17.0.5+8 (build 17.0.5+8, mixed mode, sharing)
```

5.2 참고 사항

Swing의 UI 개선을 위해 오픈소스 라이브러리 Flatlaf(<https://github.com/JFormDesigner/FlatLaf>)를 LookAndFeel로 적용하였다. 버튼 모양 등 수정하기 힘든 부분에서 UI의 개선을 이루고자 해당 라이브러리를 사용하였으며, 해당 라이브러리를 사용하지 않더라도 기능 면에서는 차이가 없기 때문에 최종 제출을 위한 소스코드에서는 주석 처리하였다. (BingoGame.jar의 12-22번 라인)

해당 라이브러리를 포함한 원본을 실행해보고자 한다면 첨부된 jar 파일을 더블 클릭하여 실행시키거나, 더블 클릭으로 실행되지 않을 경우 `java -jar` 명령어로 실행시키면 된다.

5.3 수행 기간

2022.11.28(월) - 2022.12.1(목)