

Java Programming (0409)

Final Project Report

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

담당 교수: 지정희 교수님
제출일: 2022년 12월 5일 월요일

1 문제 정의 및 분석

본 프로젝트에서는 Java와 Swing을 이용하여 빙고 게임 데스크탑 애플리케이션을 제작한다.

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 진행한다.

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.

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-Seperated 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.

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 생성자로서, fullWordList 단어장에서 받아온 전체 단어 리스트와 N을 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. 만약 AI 수준이 쉬움일 경우, 빙고판에서 선택되지 않은 단어 중 랜덤으로 하나를 뽑아 반환한다. AI 수준이 보통일 경우, 가로, 세로, 대각선 중 가장 빙고에 가까운 줄에서 선택되지 않은 단어를 골라 반환한다. AI 수준이 어려움일 경우, 만약 N이 홀수이고 가장 처음 선택되는 단어일 경우 빙고판 한가운데를 먼저 선택해 반환한다. N이 홀수가 아니거나 가장 처음 선택하는 단어가 아닐 경우, 가로, 세로, 대각선 중 가장 빙고에 가까운 줄에서 선택되지 않은 단어를 반환하되, 1/2의 확률로 사용자의 빙고판에 없는 단어를 우선적으로 반환하도록 한다. 사용자의 빙고판에 있더라도 그 단어만 선택하면 빙고가 완성되는 경우, 그 단어를 반환하도록 한다.

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.

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 **FileWriter**로 이어쓴 후, 버퍼를 **Flush**한다.

public static void closeStatisticWriterStream(): **FileWriter**의 **InputStream**을 **close**한다.

2.6 StartFrame.java

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

public StartFrame(String title, ArrayList<Game> statisticList): **StartFrame**의 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 **StatisticDialog**가 호출되도록 한다.

public void actionPerformed(ActionEvent e): **ActionListener**의 **actionPerformed**를 **Override**한 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 불가할 경우 경고 **Dialog**를 띄운다. 만약 정상적으로 게임이 실행 가능하면 메인 스레드에서 **GameFrame**을 호출할 수 있도록 **this.notify()**해주고 윈도우를 **dispose()**한다.

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

2.7 GameFrame.java

게임 화면을 출력하는 **Frame**이다.

public GameFrame(String title, User user, User computer, int N, int difficulty): **GameFrame**의 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 입력받은 값을 접근할 수 없도록 **this.wait()**을 사용한다.

private void initUI(): 게임 플레이 UI를 구성한다.

public void actionPerformed(ActionEvent e): **ActionListener**의 **actionPerformed**를 **Override**한 메소드로, 단어 입력 버튼이 눌렸거나 단어를 입력받는 텍스트 필드에서 **Enter** 키가 눌렸을 때 호출된다.

2.8 StatisticDialog.java

3 실행 결과

4 느낀점 및 토의 사항

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 수행 기간

2022.11.28(월) - 2022.11.30(수)