

# 情報学群実験第 1 最終レポート

1260277 浅野友哉

2023 年 8 月 1 日

## 目次

1	概要	2
2	基本仕様・実現方法	2
2.1	仕様 1: ゲーム開始時の盤面の初期化 . . . . .	2
2.2	仕様 2: タイルを開く処理 . . . . .	2
2.3	仕様 3: 地雷を踏んだ際の処理 . . . . .	3
2.4	仕様 4: フラグの処理 . . . . .	3
2.5	仕様 5: 適切なダイアログの表示 . . . . .	3
3	追加仕様・実現方法	3
3.1	追加 1: 盤面の UI の変更 . . . . .	3
3.2	追加 2: ゲームの状態表示等文字表示 . . . . .	4
3.3	追加 3: 新しいゲームの作成 . . . . .	4
3.4	追加 4: ゲーム開始時の爆弾配置 . . . . .	4
4	まとめ・結論	4

# 1 概要

このレポートでは、情報学群実験第 1 で開発した Minesweeper (地雷ゲーム) プログラムについて報告する。本レポートでは、Minesweeper の基本仕様と追加仕様について説明している。

追加仕様では、ゲームの盤面 UI を改善し、昔の Windows 版 Minesweeper で使われていた文字色を再現し、背景色を変更する機能を追加している。さらに、ゲームの進行に応じて激励文を表示し、プレイヤーの楽しみを向上させている。また、新しいゲームを作成する際にタイルの数や地雷の数を指定できるように拡張し、柔軟なゲーム設定が可能となっている。

プログラムの開発には、再帰処理の実装が特に大変だったが、適切に実装することでスムーズなゲームプレイを実現している。また、拡張性を考慮しメソッドを随時分けた設計により、追加機能の実装が容易となっている。

報告書では、基本仕様・実現方法および追加仕様・実現方法について詳細に記述しており、得られた成果や課題に基づいた結論を示し、今後の展望を述べている。

## 2 基本仕様・実現方法

### 2.1 仕様 1: ゲーム開始時の盤面の初期化

ゲーム開始時に、盤面にランダムに地雷を設置する

雛形にあるように、盤面は二次元配列として実装されている。ゲーム開始時には、ランダムな場所に指定された数の地雷が配置される。そのあと同じ二次元配列内に、各タイルの周辺にいくつ地雷があるかを示す数字も計算される。タイル周辺の数字を計算する部分は、`countAdjacentBombs` メソッドとした。

追加仕様 4 の実装により変更がある

### 2.2 仕様 2: タイルを開く処理

パネルを左クリックした際、パネルを開く。

タイルの表示状態・フラグの状態の二次元配列にその時の状態を保存しながら実行させていった。条件分岐により、盤面の範囲・タイルの表示状態・フラグの状態をチェックさせた。周辺に地雷がない場合、再帰処理により隣接するタイルも自動的に開かれる。

openTile メソッド内で再帰処理させると、煩雑なので openAdjacentTiles メソッドに取り出した。openAdjacentTiles メソッドでも再帰処理による無限ループを防ぐために、openTile メソッドとほぼ同じ条件分岐により対応した。

## 2.3 仕様 3: 地雷を踏んだ際の処理

パネルを開いた時、そのパネルに地雷が隠されていれば全てのパネルを開く。

プレイヤーが地雷を踏んだ場合、ゲームオーバーとなる。gameEnd のフラグを True にして、ResultDialog を出して、全ての地雷の位置を表示させる。gameEnd のフラグが立っているので、どちらのクリックも聞かないようになる。

## 2.4 仕様 4: フラグの処理

開かれていないパネルを右クリックした際、そのパネルに旗を立てる/取り除く。

旗を立てられたパネルは、旗が取り除かれるまで左クリックで開けない。

ゲームの開始前、最初の左クリックが行われるまで右クリックできないようにした。フラグ用の二次元配列を反転させていった。

仕様 2 と 4 で使用した setTextToTile メソッドに引数を追加して、盤面表示である状態を変更させた。

## 2.5 仕様 5: 適切なダイアログの表示

ゲームクリアもしくはゲームオーバーになった際、適切なダイアログを表示する。

雛形にある ResultDialog をほぼそのまま使用した。追加仕様 3 で流用したため、ボタンの処理とサイズを変更した。

# 3 追加仕様・実現方法

## 3.1 追加 1: 盤面の UI の変更

文字についての変更

昔の windows に搭載されていた MineSweeper の文字色を再現した。setTextToTile メソッドで条件分岐させて、色を変更させるのがむずかしかった。ゲーム終了後フラグが立っていた爆弾の場所は、黒塗りの星。そうでない場所は中抜きの星で表した。

### 背景色の変更

初期の背景色を少し明るくし、ゲーム中に空いたタイルは、灰色に変更させた。ゲーム終了後すべてのタイルが表示されるとき、ゲーム中開いていないタイルは背景色を変更させないようにした。setTextToTile メソッド内での条件分岐で tilebgcolor メソッドを呼び出している。

## 3.2 追加 2: ゲームの状態表示等文字表示

盤面の下に激励文を表示させるようにした。ゲームの盤面によって文章を変更させた。雛形に文章表示用の label を追加した。

## 3.3 追加 3: 新しいゲームの作成

タイルの数・爆弾の数を指定して新しいゲームを開始できるようにした。タイルの総数-10 個の爆弾の数を指定できる。その時とエラーは仕様 5 で使ったダイアログを流用した。

タイルどちらかの数でも少なくすると、盤面の表示や追加した UI がおかしくなるのがなおせなかった。ボタンのイベントリスナーに処理を追加して、Main インスタンスを生成した。

## 3.4 追加 4: ゲーム開始時の爆弾配置

ゲーム開始時最初の左クリック時に、そのマスが 0 になるようにした。コンストラクタの時点で初期化でなく、firstClick フラグクリックを判定して初期化している。initTable と setBombs メソッド引数を追加して、クリック位置の爆弾設置判定をしている。

## 4 まとめ・結論

このレポートでは、情報学群実験第 1 で制作した Minesweeper のプログラムについて報告した。プログラムの作成には、特に再帰処理の実装や拡張性の考慮が重要であった。

プログラムの主な成果として、ゲームの盤面を開いた際に自動的に隣接するタイルも開く再帰処理を実装し、プレイヤーにスムーズな操作を提供することができました。また、UI の変更や激励文の表示など、追加仕様によってゲームのエンジョイメントを向上させ

ました。

一方で、課題としては、タイルの数や爆弾の数を指定する際に発生する UI のおかしい挙動があった。この課題に対しては、入力値のバリデーションや UI の調整などを検討し、改善を図る必要がある。もう少し時間をかけて調整を行えば改善できると考えられる。

今後の展望としては、さらなる拡張や改善が考えられる。例えば、難易度選択やタイム記録の追加、グラフィカルな改善などが挙げられる。また、コードの可読性や効率性の向上も重要な課題となりうる。

## 参考文献

- [1] Oracle and/or its affiliates, "java.awt (Java Platform SE 8)", <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/package-summary.html>, 7/23
- [2] 中山清喬, 国本大悟: スッキリわかる Java 入門 第3版インプレス 2019., 7/23
- [3] JavaDrive, "Java — 絶対値を取得する (Math.abs)", <https://www.javadrive.jp/start/math/index1.html>, 7/23
- [4] "Container (Java Platform SE 6)", <https://www.ugs.kochi-tech.ac.jp/mendori/internal/docs/ja/api/java/awt/Container.html>, 7/23