技術レポート

島根大学 渡邉直樹

3 月 26 日

1 概要

これは Java アップレット上でリバーシを再現したプログラムである。クリックした時に反転できる場合は反転し、そうでない場合は何もしない。特徴として、コマが置ける場所は色が異なっていてわかりやすくなっている。また、サイズは自由に変更できる。

2 動機

私はリバーシを作ろうと考えた理由は Java の練習をするためです。リバーシを制作する中で、条件分岐や繰り返し処理を多く使用すると考え、リバーシを選択しました。また、盤面上でのクリック等、操作がシンプルで Java アップレットでの再現ができそうだったのでリバーシを選択しました。

3 説明

3.1 Main クラス

主にアップレットの操作とクリック時の処理を記す。

paint(Graphics):void

Applet クラスからオーバーライドしたものであり、描写全般を行う。主に、下部の勝敗等を示すテキストを表示する。ボードやコマの表示は、別のメソッドを呼び出すことにより行う。

mouseClickedMotion(MouseEvent):void

MouseListener インターフェイスからオーバーライドしたものであり、マウスをクリックした時の処理を行う。クリックした位置がボード上である場合、そのマスに置けるかどうかを判定する。置ける場合はそのマスに置き、ターンを渡す。そして、相手のターンで置けなかった場合は、もう一度自分のターンになる。その時点で、自分がコマを置けない場合は終了する。

最後に再描画する。

Reversi クラス

主にリバーシにおける操作による処理を記す。

int[][] board

現在の盤面を示した配列である。値が 0 の場合は何も置いていない、1 の場合は黒のコマ、2 の場合は白のコマを置く。

enablePlace(boolean):boolean

置けるかどうか判定するマスを格納したリスト checkPlace に含まれるマスのみを判定する。もともと、すべてのマスを判定していたが、処理効率の向上のため、リスト管理とした。

すべて置ける場合は true を返す。

enablePlace(int, int, boolean):boolean

引数で指定されたマスに対して置けるかどうか判定する。

1. 距離 d の地点について調べる。
2. もし、何もおいていない場合はその方向の判定は行わない。黒の場合はその方向の距離を記憶する。
3. もし白の場合は、特に何もしない。
4. すべての方向を調べたら、距離 d+1 の地点について 1 を繰り返す。

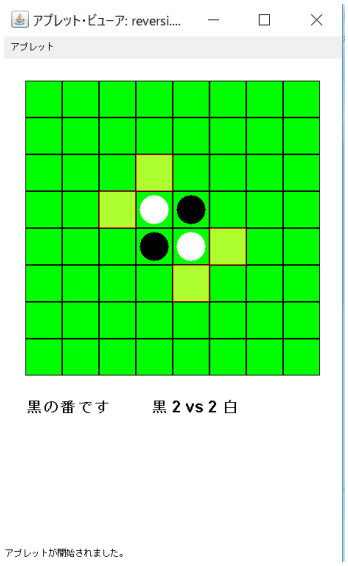
reverseDisk(int, int):void

指定されたマスに対して、enablePlace メソッドで記憶された方向と距離がわかるので、board[][] を用いて、各方向に対して、コマを反転させていく。

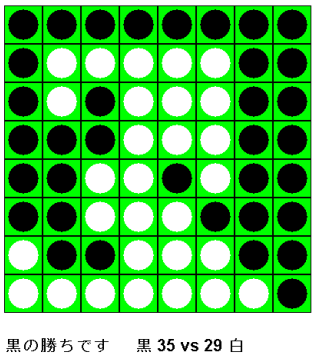
getWinner():String

勝敗を示す文字列を返す。countDisks[] により、コマの数が多いほうが勝利する。

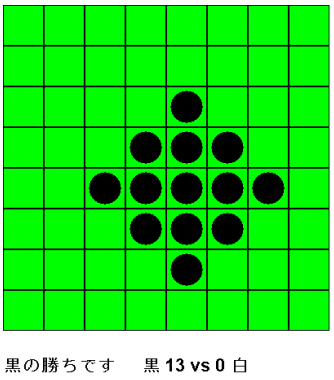
4　参考画像



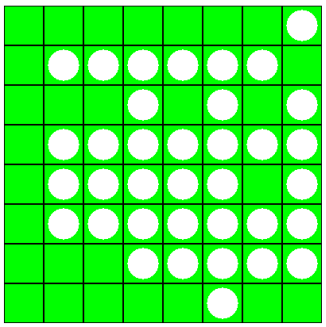
[1] 初期画面



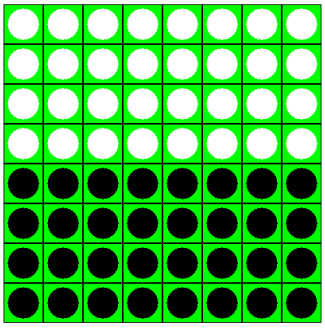
[2] 黒が勝利



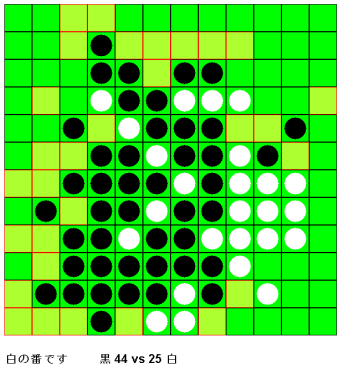
[3] 黒が勝利



[4] 白が勝利



[5] 引き分け



[6] サイズ変更

5　感想

実際に Java アップレットでリバーシを作成してみて、for 文の範囲を設定する際、方向に応じて設定する範囲が異なり、難しく感じた。処理内容をどちらのクラスに書くべきかを決めるのが大変だった。しかし、完成した際には、最初に作ったものよりも段々と効率化していて、よりよいものができたのでよかった。特に、リストを用いたことで置けるかどうかを判定する処理が格段に速くなったことに喜びを感じた。

6　リンク

`https://github.com/Ebikatsu/Reversi.git`