

# Java 言語によるリバーシの制作

中山 翔太 杉谷 幸之輔

## 1. まえがき

今回、Java 言語を使って何かを作ることで Java 言語に触れ、少しでも Java 言語について知りたいと思い、リバーシの制作を行った。

## 2. 原 理

リバーシのゲームを作るためには次のような 5 つの処理が必要である。

### (1) ボードの管理

まず基本となるボードに配置するデータを管理するためのものが必要である。

リバーシの状態はコマのないとき、白のコマが置いてあるとき、黒のコマが置いてあるときの 3 パターンなのでこの 3 パターンのうちのどの状態かと示す値を配列で管理する必要がある。

### (2) ゲーム開始、コマの交換

ゲームの機能としてはゲームの開始、コマ(白か黒か)の変更の処理が必要。

### (3) タッチしてコマを配置するための処理

コマを押すマス目をタッチし、その周囲の相手のコマを裏返す処理が必要となる。

ここでは上下左右斜めの八方向を調べ、どこまで裏返すか調べなければならない。

### (4) パスのチェック処理

置く場所がないとき、パスをすることになる。これは空いているすべてのマス目について、置くことが出来ないことを確認しなければならない。

### (5) ゲーム終了の処理

すべてのマス目が埋まればゲーム終了となる。また、両方のプレイヤーが続けてパスをした場合もどちらも置く場所がないということだからゲーム終了となる。

## 3. 研究内容

### (1) Reversi クラス

レイアウト作成や初期化を行う他、ゲームの開始などの操作をするオプションメニューの作成処理などを用意するクラスである。アクティビティごとに存在するオプションメニューの表示を行うプログラムを図 1 に示す。また、ゲーム開始のプログラムの処理を行うプログラムを図 2 に示す。

```
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);
```

図 1 オプションメニューの自動追加

```
view = (ReversiView) this.findViewById(R.id.ReversiView01);  
  
view.start();  
return false;
```

図 2 スタートメソッドの呼び出し

### (2) ReversiView クラス

実際に画面に配置されて表示されるものを用意するクラスである。ボードの大きさ、位置についてのプログラムを図 3 に示す。

```
private int board_x=41;  
private int board_y=149;  
private float board_w=400;  
private float board_h=400;  
private int score_x=60;  
private int score_y=83;  
  
float w=reversi.disp_w;  
float h=reversi.disp_h;  
float dw=w/480f;  
float dh=h/854f;
```

```

board_w*=dw;↓
board_y*=dh;↓
board_w*=dw;↓
board_h*=dh;↓
score_x*=dw;↓
score_y*=dh;↓

```

図3 ボードのサイズ、位置

### (3) Reversiboard クラス

ボードの情報を管理し、ボードを操作するための機能をまとめたクラスである。ボードのデータは Reversioobject で「data」という二元配列として管理されているので、二重の for を使い、この data から順に値を取り出し、その値に応じて描写をしている。(図4)

```

switch (object.data[i][j]){↓
  case 0:↓
    break;↓
  case 1:↓
    black.setBounds(r);↓
    black.draw(c);↓
    break;↓
  case 2:↓
    white.setBounds(r);↓
    white.draw(c);↓
    break;↓
}↓

```

図4 ボードの描写

コマの処理を行っているのは、「setPiece」で指定した場所にコマを配置するもので、「reversiPiece」で指定した場所にコマを置いたときに相手の裏返す処理を行う。この一連の処理でコマを置くことが出来る。(図5)

```

object.setPiece(c,r,side);↓
object.reversePieces(object.data, c, r, side);↓
object.countData();↓

```

図5 コマの処理

### (4) Reversioobject クラス

Reversiboard と Reversioobject はまとめて Reversiboard としてもよいが、分量が多くなるため Reversioobject クラスに分けた。このクラスではボードのデータとそれを処理するための機能を作成するクラスである。相手のコマを裏返す計算を行っている。上下左右斜めの八方向にマス目を移動して状態を調べていかなければならないため、二重の繰り返し処理を使い、次のような手順を行う。

- ①縦移動と横移動に、繰り返しで取り出した値を足して、隣のマス目の位置を得る。
- ②それが現在と同じ位置と同じなら、「取れない」と判断し、次のチェックへ進む。
- ③縦または横の位置がゼロ未満か8より大きかったら「取れない」と判断し、次の方向のチェックへ進む。
- ④その位置の値が自分のコマなら、「そこまで取れた」と判断し、次の方向のチェックへ進む。
- ⑤どれにも引っかからなかった場合は、まだ相手のコマが続いていると考え、裏返せるコマを1増やし、また「①」に戻る。

これだけの処理を行うことにより相手のコマの裏返す量を計算することが出来る。

```

int[] dt={0,-1,1};↓
for (int dx:dt){↓
  for (int dy:dt){↓
    int xn = x1;↓
    int yn = y1;↓
    boolean f = false;↓
    int nn = 0;↓

```

図6 相手側のコマを裏返す処理

## 4. まとめ

ゲームのプログラムにおいてたった一つのミスでも実行したときにとっても重大な欠陥となるということが改めて分かった。

Java 言語は C 言語とおおまかな概念は同じだが、使える文字などがとても多くまだまだ理解で

きていない部分もあった。

## 5. あとがき

今回のこの課題研究では多くある言語の中でも Java 言語を使ったが、ここで私が学んだのはどんなものをどんな機能を使って何の言語を使うにしても、それぞれ自分の頑張り次第で楽しくもなり、そしてつまらなくもなるということだ。

私は Java 言語についていろいろなことを学ぶことが出来、課題研究前より Java 言語という言語について知ることは出来たが、それ以上に自分なりに努力し学び、完成させるというその過程を実行するという経験し得ることが出来た。

この経験は私のこれからの人生のなかでも役に立つことは多いだろう。この経験を私はこれからの自分の生活に活かしていこうと思う。

中山 翔太

今回、適当に軽い気持ちでこの課題研究を選んだが、適当に選ぶのは間違いだったと思うくらい難しかった。しかし、プログラムの入力作業など自分でしっかりできたので、よかったと思う。

杉谷 幸之輔

## 6. 参考文献

15歳から始める Android ゲームプログラミング  
Java からはじめよう Android プログラミング