

Java プログラムによるゲーム制作

白根 健治 大森 翔太
竹下 領一

1.研究概要

Java を通じてプログラム言語の理解を深めるとともに、作品を発表できるまで技術の向上を目指して、研究期間 8 ヶ月の期間で作品を制作する。その過程でブロック崩しを作ることを目指し、1 月の発表に向けて研究と制作する。

2.研究計画

- 5 月:開発環境 Eclipse を導入
- 6 月:個人で Java の研究・学習
- 7 月:中間発表に向けての資料作成
- 8 月:オープンスクールでの成果発表
- 9 月:作品制作
- 10 月:文化祭に出展する作品の資料作成
- 11 月:文化祭にて作品発表
- 12 月:応用プログラムの研究
- 1 月:発表

3.研究内容

最初の 2 ヶ月は Java の簡単なプログラム(四則計算や文字表示など)について勉強し、プログラムの制作に移る。私たちはブロック崩しを制作することにした。ブロック崩しのプログラム制作にはインターネットや本を参考に基礎の形となるプログラムを構成し、基礎が出来れば応用を加えていく。

(1) 開発環境「Eclipse」と Java プログラム

私たちが使用した Eclipse というソフトはオープンソースの統合ソフトウェア開発環境の一つであり、Java 開発者を中心に急速に普及しており、ソフトウェア開発の共通プラットフォームの標準になると予想されている。IBM 社が 1999 年から進めていた開発ツール研究プロジェクトの研

究成果をソフトウェアの形にしたもので、2001 年 11 月にオープンソース化され、誰でも無償で入手・改変・再配布できるようになった。

Eclipse は単なる開発ツールではなく、開発ツールの「共通プラットフォーム」と呼ばれる。これは、機能をプラグインの形であとから自由に追加できるためで、UML 関連ツールやテストツール、各種のプログラム言語などが用意されている。その数は数百種類に及ぶ。

大手ソフトウェアベンダーの中には、自社の開発ツール製品に Eclipse を組み込み、Eclipse に追加する形で自社独自部分を提供するという形の製品をリリースするところも現れている。以上が Eclipse の説明である。

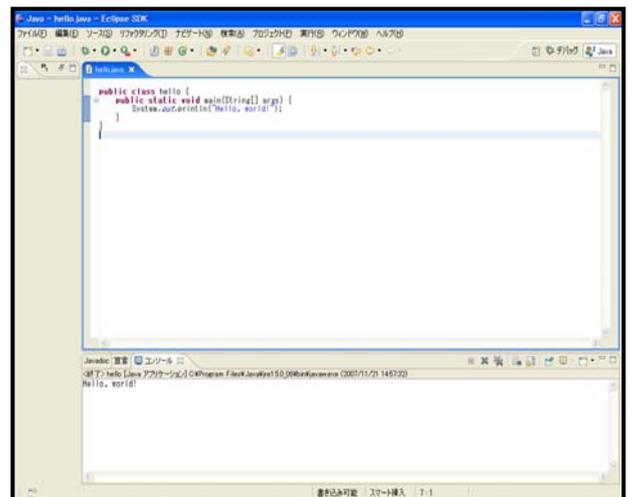


図 1 Eclipse での Java プログラム作成の様子

(2) プログラムについて

ア.Java の基本的なプログラム

```
class G基本動作 {
    public static void main(String iuiui[]) {
        System.out.println("九九の表");
        for (int i=1; i<=9; i++) {
            for (int j=1; j<=9; j++) {
                System.out.print(i*j + " ");
            }
            System.out.println();
        }
    }
}
```

図 2 九九の段のプログラム

図 2 は私たちが課題研究の最初の頃にテストで作った九九の段を表示プログラムである。簡単に説明すると、繰り返し関数「for 文」を使い、1～9までの数字を横に表示する。更にそれを「for 文」で式「i * j」を括り、横 9 縦 9 の数字の構造に「i * j」の結果が表示されて、九九の段が完成する仕組みになっている。

イ.グラフィック関連のプログラム

```
// アプレットの背景色を灰色に設定。  
this.setBackground(Color.lightGray);  
}  
  
public void paint(Graphics g) {  
    // 円の描画。  
    g.setColor(Color.red);  
    g.drawOval(30,30,60,60);  
    g.fillOval(130,30,60,60);  
  
    // 四角形の描画。  
    g.setColor(Color.green);  
    g.drawRect(30,100,60,60);  
    g.fillRect(130,100,60,60);  
}
```

図 3 グラフィック関連のプログラム

図 3 のプログラムは○と□、それぞれの色を決めて表示するプログラムである。例えば、このプログラムならば、背景(Background)を灰色(lightGray)に設定し、円を赤(red)に四角を緑(green)設定にする。次は g.drawRect と g.fillRect という関数を使用し位置と大きさを指定する。プログラムを実行した結果を図 4 に示す。

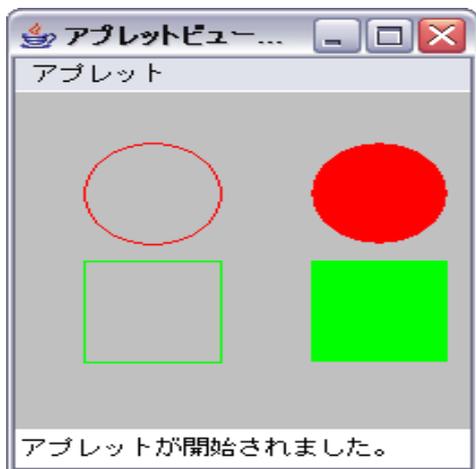


図 4 プログラムの実行結果

(3) GIF アニメーションソフト「Giam」について

Giam とは古溝 剛 氏が制作した、GIF アニメーション作成ソフトで、フリーソフトなので無料で配布されている。このソフトは使いやすく、比較的簡単に GIF アニメーションが制作が出来るので初心者にもお勧めである。今回は応用でボールに GIF アニメーションをつけてみようと思いこのソフトを使用した。ボールサイズで合計 18 コマを繰り返す GIF アニメーションを制作した。

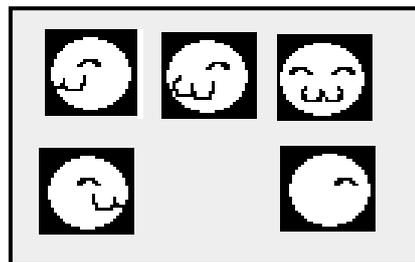


図 5 制作した GIF アニメーション

(4) ブロック崩しの制作について

まず、最初に Java での課題を決める際に私たちは誰にでもわかるポピュラーなゲームを作ることを考えた。そのなかでも Web 上に多くの資料があり、動作が理解しやすそうなブロック崩しゲームを選んだ。最初に画面の処理を表示させるために画面枠の表示から取り掛かった。画像処理のプログラム等はインターネット上に多くの資料が掲載されていて、それを参考にしたので効率良く学ぶことが出来た。次に、ブロックの表示とラケットの表示について学んだ。画像処理は先ほどの画面枠とほぼ同等なのですぐに来た。同時に配色のプログラムを取り込み、そのグラフィックをマウスを利用して動作させるプログラムをラケットに当てることができた。

その後 ブロック、ラケット、背景、枠などに色を付け、それぞれの部位に当たり判定を追加した。次はボールの表示と移動処理を追加した。ボールの移動は難しい計算式を用いた。最後に当たり判定の設定だった。この研究で一番難しい部分

だった。ボールの当たり判定はラケットに当たる入射角により反射する角度が変化するため非常に取り組みにくい部分だった。壁(枠)やブロック自体にも反射の為の当たり判定をつけて大体のブロック崩しとなる形を作った。その後付加機能を付けたり改良に取り掛かったが、プログラムの構成が悪く、大きく変身しなければならない場合もあった。

```
public void paint(Graphics g) {
    // ダブルバッファ使用時には、オフスクリーン側に先に描画する
    offg.setColor(Color.black);
    offg.fillRect(0, 0, 500, 500); // 画面を消去する

    // offg.setColor(Color.gray);
    // 枠の影
    offg.fillRect(0+3,0+3,fieldW,border);
    offg.fillRect(0+3,0+3,border,fieldH);
    offg.fillRect(0+3,fieldH-border+3,fieldW,border);
    offg.fillRect(fieldW-border+3,0+3,border,fieldH);
    // ボールに影をつける
    offg.fillOval(bx+3,by+3,bs,bs);
    // ラケットに影をつける
    offg.fillRect(rx+3,ry+3,rw,10);

    // 赤枠
    offg.setColor(Color.red);
    offg.fillRect(0,0,fieldW,border);
    offg.fillRect(0,0,border,fieldH);
    offg.fillRect(0,fieldH-border,fieldW,border);
    offg.fillRect(fieldW-border,0,border,fieldH);
    // ボール
    offg.drawImage(img, bx, by, this);
    // ラケット
    offg.drawImage(img2,rx,ry,this);//111
    //offg.setColor(Color.white);
    offg.fillRect(rx,ry,rw,10);

    // ブロック
```

図6 ブロック崩しプログラムの一部

3.研究のまとめ

Javaの研究を通してJavaの理解を深め、自分自身でプログラムを作れるまでに学習する。授業では受けられないような内容のできる課題研究は自分のやってみたい研究できる時間であり、また技術者としての能力を向上する場でもあると考えていた。そして、課題を決める際におなじことをやってみたい同志を集い、Java班を結成した。Javaを選んだ理由は様々だったが、大体は昨年(又は一昨年)の先輩の発表を見て選んだ人だった。

課題研究の初日。まずはそれぞれ個人で簡単なプログラムの研究を開始した。皆、最初は四則計算、文字表示などの簡単なプログラムを組んでいたが、後々ゲームやアプリなどのプログラムを作り始めるようになった。その時期では、知らない変数や難しいな計算式に悩まされた。全てのプログラムにいた事だが難解な変数などはゲーム

を制作する上で必要であるので、理解しておく必要があった。まずはプログラムを中間発表に向けてプログラムを制作し、一学期に発表した。今度は二学期の中間発表に向けて更なる研究し、ほぼ完成した作品を発表した。その時点で形がほぼ完成していたので、そこからは応用的な技術を使い、GIFアニメーションを取り入れるなど、更なる研究をし、現在それぞれの作品の完成に至る。

この研究を通して、実際にプログラミングする人の気持ちになり、どのようにしたら楽しんでもらえるか、その重要性を学んだ。例えば、プログラムを組み、デバッグをすることによってユーザーにわかりやすいような表現方法や出来る限り面白いゲームに仕立てるための方法を知ることができたのである。それだけでなく、友人に見せ合い等をして、お互いに作品の評価をし合い、制作することによってよりユーザーに易しいゲームに近づけることが出来た。

4.制作した作品

ア 白根作品

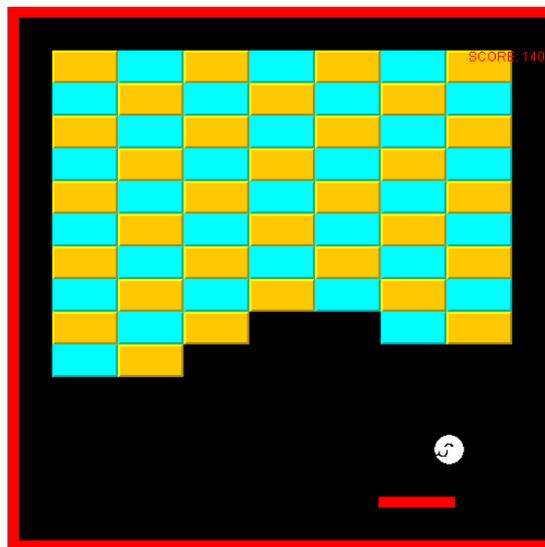


図7 白根 制作ブロック崩し

配色を見易いものにし、面白みをつける為にボールに自作のGIFアニメーションを加えた。ブロックの数を増やして出来る限り長い時間楽しめるように設定した。他にもラケットとボールのスピードの調整など、ユーザーの遊びやすい環境を目

指して制作した。

イ 大森作品

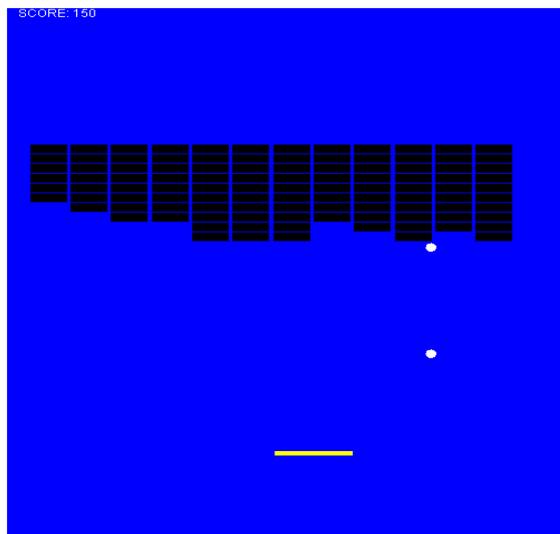


図8 大森 制作ブロック崩し

ブロック崩しにインベーダーの動作を加えてみたり、ボールの数を増やしたり、スコアも表示させるようにした。

ウ 竹下作品

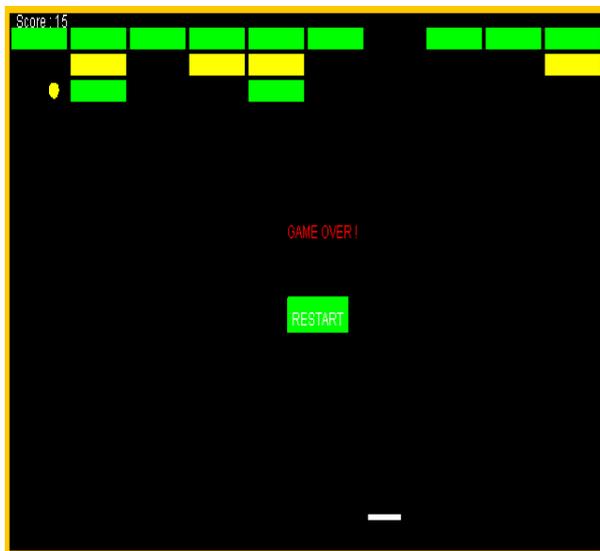


図9 竹下制作ブロック崩し

リセットボタンを作って GAME OVER や PERFECT になった時に表示出来るようにして、何度でもゲームが出来るようになっている。ブロックは三段に分かれており、二段目のブロックはボールが当たるとスピードを速くし、一段目

と三段目のブロックは速くなったボールが当たると元の速さに戻るようにプログラムを組んでいる。

5.参考文献

(1) Java プログラミング入門

<http://www.ee.fukui-nct.ac.jp/~yoneda/text/other/programming/index.html>

(2) Java によるゲーム解説

<http://www012.upp.so-net.ne.jp/flab/javatuto/javatuto.html>

(3) Giam の製作者のページ

<http://homepage3.nifty.com/furumizo/giamd.htm>