

Visual Studio C#

による大富豪の制作

麻田 空良 平田 雄人

1. まえがき

二年生の課題研究で大富豪を制作しようと思っていたが知識が足りなく、断念したので、三年生の課題研究でも継続して制作することとなった。

2. 原 理

大富豪は Visual Studio の C# を使用し制作した。今回制作した大富豪は 4 人制で、その内 3 人はプログラムで動く AI となっている。カードはボタンを使用しどのカードが押されたかが分かるようにした。実行画面を下の図に示す。

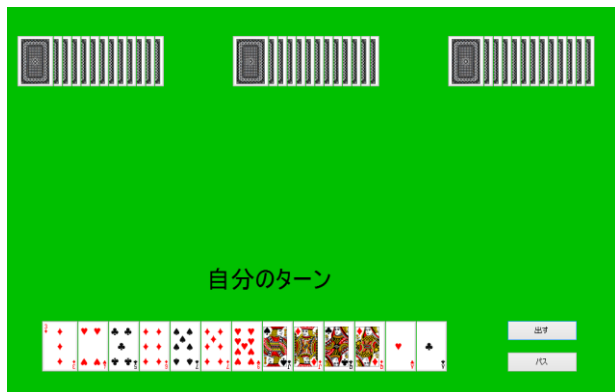


図 1 実行画面

3. 研究内容

(1) 手札の作成

まず最初にボタンの配列を 52 個作成した。それにトランプの画像を入れ手札として表示するプログラムを作成した。ボタンの数が多かったので for 文と SetBounds を使いプログラムでボタンの配置をした。

(2) 手札のシャッフル

52 個あるボタンの配列の Tag に button[1] には 1 を button[2] には 2 というように 1 ~ 52 の数値を入れていく。for 文を使い 1 ~ 52 までを

ループさせその中で、乱数で出た値の配列の要素と for 文の回数の値の配列を入れ替えるようにした。

```
Random ransu = new Random();
int random = ransu.Next(0, 51);
int[] number = new int[52];
ransu = new Random();
for (int i = 0; i < number.Length-1; i++)
{
    number[i] = i + 1;
}

for (int i = 0; i < number.Length-1; i++)
{
    random = ransu.Next(0, 51);
    int temp = number[i];
    number[i] = number[random];
    number[random] = temp;
}
```

図 2 乱数を使用したシャッフル

このボタンにはまだ画像が表示されていないので画像を表示させるプログラムを作成した。

まずあらかじめ、Image 型の配列を画像の数だけ用意しておき、その配列に一つ一つ画像を入れていく。そして Button[1] には Image[1] の画像、というように画像を入れていく。

```
for (int m = 0; m <= 3; m++) //プレイヤー数
{
    for (int i = 0; i < 13; i++) //カード数
    {
        Card[m, i].Tag = number[i+(13*m)]; //ボタンのTagに乱数を入れる
        Controls.Add(Card[m, i]); //ボタン配置
        if (m == 0 && i < 13) //プレイヤー1の時の手札設定
        {
            int z = (int)Card[m, i].Tag;
            Card[0, i].Image = image[z];
        }
        if (m >= 1) //AIの手札設定
        {
            Card[m, i].Image = Image.FromFile("D:\\png\\z01.png");
            int z = (int)Card[m, i].Tag;
            Card[m, i].Image = image[z];
        }
    }
    Card[m, i].Click += new EventHandler(GetCard);
}
}
```

図 3 ボタンの配列に画像をいれるプログラム

(3) 手札の並び替え

各プレイヤーには乱数によりシャッフルされた手札があるが並び替えをされていないので、手札の数字が昇順になるようにするプログラムを作成した。並び替えのプログラムには、基本選択

法を使用した。ボタンにはトランプの数字と対応した値を **Tag** に入れているため **Tag** の値を比較し大小を調べた。

```
private void narabikae() //手札の並び替え
{
    int work;
    int x=0,y=0;
    for (int p = 0; p < 4; p++)
    {
        for (int i = 0; i < 12; i++)
        {
            for (int j = i + 1; j < 13; j++)
            {
                int a, o;
                a = atai(p, i);
                o = atai(p, j);
                if (a >= o)
                {
                    x = (int)Card[p, i].Tag;
                    y = (int)Card[p, j].Tag;
                    Card[p, i].Image = image[y];
                    Card[p, j].Image = image[x];
                    work = (int)Card[p, i].Tag;
                    Card[p, i].Tag = Card[p, j].Tag;
                    Card[p, j].Tag = work;
                }
            }
        }
    }
}
```

図4 手札の並び替えをするプログラム

このプログラムでは、ボタンの **Tag** と画像だけを入れ替えるようにした。

(4) カードが一枚の時の比較

次に、カードを場に出すため、場に出ているカードとの比較を行うプログラムを作成した。場のカードの数値を変数 **ba** にいれるようにした。場にカードが何もなければ **ba** に -1 を入れ、次にカードが出された場合には、そのカードの数値を **ba** に入れるようにした。既にカードが出ている場合は、**ba** に入っている数値と出そうとしている数値を比較し、出すカードの方が数値が大きかった場合、**ba** に出すカードの数値を入れるようにした。

```
private int rank(int i)
{
    int dasu,hante;
    dasu = 0;
    dasu = atai(0, i);
    if (ba < dasu)//場にある数値と出すカードとの比較
    {
        hante = 1;
        ba = dasu;
        label12.Text = ba.ToString();
    }
    else
        hante = 0;
    return hante;
}
```

図5 カードの比較

(5) カードが二枚以上の時の比較

カードを一枚出すときと基本的には同じだが、選択されたカードが同じものなのかを調べる必要があった。選択されたもので、一番左のカードを一枚目とした。4枚出す場合には、一枚目と二枚目、二枚目と三枚目、三枚目と四枚目と調べ全て同じだった時に1を返すようにした。そして、2枚だった場合にはカードの数値に100を足し、**ba** に入れるようにした。3枚の時は200、4枚の時は300を足すようにした。

(6) カードを場に出す

カードの比較を行いカードが出せる場合は、出したカードの画像を読み取り、あらかじめ用意しておいたラベルにその画像を入れるようにした。しかし、このままではカードのボタンが残っているので出されたカードが二度使われないようにボタンを移動と、**Tag** の値の変更をした。

(7) AI の作成

私たちには AI に関する知識がなく、1から AI について学ぶ時間もなかったのでシンプルな AI を作成した。まず最初にカードが何枚場に出ているか調べる。そして、枚数に応じて左から出せるカードがあったら場に出すという簡単なプログラムにした。それを AI1 から AI3 までそれぞれ作成し、行動が終わったら次の AI が動くようになっている。

```
int dasu = 0;
for (int i = 0; i < 13; i++)
{
    dasu = atai(turn, i);
    if (dasu > ba && (int)Card[turn, i].Tag != 2000 && count != 1)
    {
        int z = (int)Card[turn, i].Tag;
        Card[turn, i].Image = image[z];
        label11.Image = Card[turn, i].Image;
        label11.Tag = Card[turn, i].Tag;
        Card[turn, i].SetBounds(1000, 1000, 10, 10);
        Card[turn, i].Tag = -2;
        if (turn == 1)
        {
            aione += 1;
            if (aione >= 13)
                syouhai(1);
        }
        if (turn == 2)
        {
            aitwo += 1;
            if (aitwo >= 13)
                syouhai(2);
        }
        if (turn == 3)
        {
            aithree += 1;
            if (aithree >= 13)
                syouhai(3);
        }
        ba = dasu;
        count = 1;
        limit = 0;
        if (dasu == 5)
            hatigiri(turn);
    }
}
if (count != 1)
    limit += 1;
```

図6 AI がカードを出すプログラム

(8) パスをした時のプログラムの作成

大富豪は4回パスすると場のカードがなくなり最後に出した人が好きなカードを出せるというルールとなっているので、パスをした時のプログラムを作成した。パスの回数をカウントする変数 `limit` を作成した。プレイヤーがパスのボタンを押したとき、`limit` を1足し、AI1が動くようにした。AIは出せるカードがなかった場合 `limit` を1足し、次のAIが動くようにした。カードが出された場合には、`limit` を0にした。`limit` が4になった場合、`ba` に-1を入れ、ラベルは画像をNULLにした。

```
private void limit_over()
{
    limit = 0;
    ba = -1;
    label1.Image = null;
    nimai.Image = null;
    sanmai.Image = null;
    yonmai.Image = null;
}
```

図7 パスを4回した時のプログラム

(9) 勝敗

枚数のカウント用の変数をプレイヤーとAI1～AI3の4個作り、カードを出すときに出した枚数だけ変数に足していくようにした。そして変数が13になったとき `syohai` という関数が動くようにした。`syohai` では、メッセージボックスで誰が勝ったのかを表示させた。

```
private void syohai(int p)
{
    if(p==0)
    {
        MessageBox.Show("プレイヤーの勝利");
    }
    if(p==1)
    {
        MessageBox.Show("AI1の勝利");
    }
    if (p == 2)
    {
        MessageBox.Show("AI2の勝利");
    }
    if (p == 3)
    {
        MessageBox.Show("AI3の勝利");
    }
}
```

図8 勝敗の表示

4. まとめ

大富豪には、8切りや10捨てなどたくさんのルールがあり、いくつか取り入れようと思ったがそのルール1つひとつすごく長いプログラムで作るのをあきらめたルールがあった。とくに7渡しや10捨てなどカードを選ぶルールは取り入れるのが難しかった。

AIは、左から出せるカードを出すという人工知能とは呼べないものだったので、もう少し頭のいいAIを作りたいかった。

5. あとがき

2年生の時もゲームを作成していて、そのことを生かせば大富豪を作れると思ったが大富豪のプログラムは非常に難しかった。最初の手札の表示や手札のシャッフルで行き詰まりあまり進まなかった。Visual studioは3年生になって少し授業で勉強していたのでよくわかった。2年生で制作したゲームも今回制作した大富豪もデザインがネット上に配信されているゲームなどに比べてすごくシンプルだった。良いゲームを作るには正しく動くだけでなくゲームのデザインも大切だということが分かった。(麻田)

C#で何かを作るのはこの大富豪が2回目で、知識が足りなく無駄が多いプログラムになってしまった。次何かを作るときは、無駄がなくエラーが起こらないプログラムを作成したいと思った。今回作成した大富豪はゲームとしてまだ面白くなく、まだ改良するところがあるので、ルールの追加などをして、完成を目指していきたいと思う。今回の大富豪の制作では、プログラムの知識と物を作る難しさを学ぶことができ、とてもいい体験になった。今回学んだことを今後の勉強などに生かしていきたいと思う。(平田)

6. 参考文献

無料素材倶楽部

<http://sozai.7gates.net/docs/trump/>

トランプの画像を使用させていただきました。