オセロゲームの制作

澤山　朋紀

１．研究概要

Microsoft 社の Visual Studio（Basic）と呼ばれる統合開発環境を使用し，オセロゲームを作成すると同時に Visual Studio(Basic)の環境下でのソフトウェアの開発方法の理解，オセロゲームのアルゴリズムの理解をする。

２．研究の具体的内容

（１）準備

オセロゲームを作るにあたって必要な開発環境の準備を行う。

Visual Studio（Basic）のインストール

ア．Visual Studio をパソコンにインスールし，開発環境の設定を行う。

イ．これらの流れの中で，Visual Studio（Basic）の開発環境に慣れ，基本的なプログラムの作成方法を習得する。

ウ．Visual Studio（Basic）でのプログラミングに慣れる。

エ．簡単な処理プログラムなどを作成し，プログラムの大まかな作り方を理解する。

（２）オセロゲームの制作

オセロゲームを作成するにあたって必要な処理方法等を調べ，理解する。

さらに通常のオセロに逆転のすることができるようになる機能をつける。

ア．プログラムの設計を行う。

オセロについてまとめているサイト等を参考にし，プログラムの概形を作る。

オセロのプログラムをメインプログラム１つとサブプログラム５つにブロック化する。

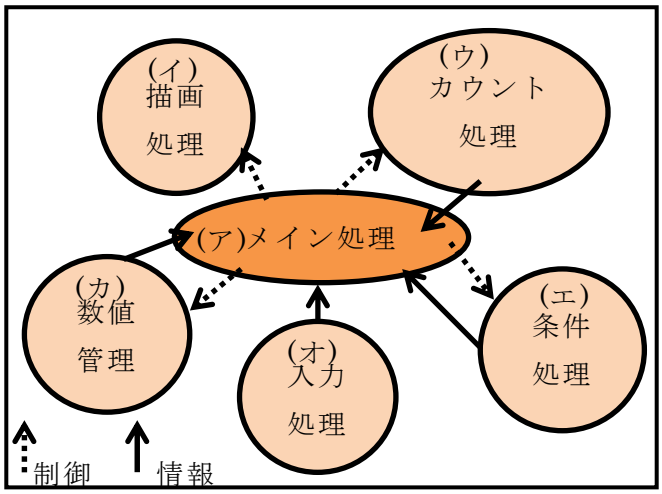

図１　ブロック化

また，６つのプログラムの処理内容を下に記す。

(ア)メイン処理

カウント処理，条件処理，入力処理，数値管理から情報を受け取り，描画処理，カウント処理，数値管理に命令を送る。

すべての入力情報をメイン処理で受け取り，その後に処理を行う。

(イ)描画処理

メイン処理から受けた命令をもとに石や盤の描画を行う。

プログラム上では，図２のような描画を行い，マウスカーソルに合わせて，グリッドをオレンジ色で囲い，見やすくしている。

また，次の順番を表示し，わかりやすくした。（図３）

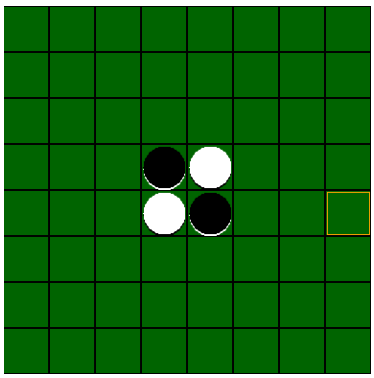

図２　オセロ盤

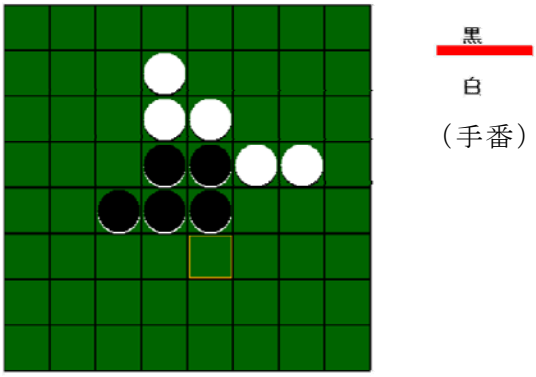

図３　オセロの手番

(ウ)カウント処理

マスに配列を設定し,白(１),黒(－１),空き(０)で管理することで，黒白と空きを把握している。図２を数値表示したものを図４に示す。

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	-1	1	0	0	0
0	0	0	1	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

図４　オセロの配列

白(１)，黒(－１)の個数を調べ，変数に格納してゲーム終了時に最終結果を表示する。

(エ)条件処理

入力方法はマウスのクリックのみの入力となっているため，判定範囲に石がある場合には入力をスルーし，次の入力を待つ設定にしてある。

石が他の色の石に挟まれたときには１を返し，それ以外は０を返すことで，１の時にのみ反転処理を行い，処理速度を速めることができる。

また，次の番のプレイヤーが石を置けないときには１００を返し，順番をスキップする。

(オ)入力処理

マウス入力のみにし，誰でも操作をしやすくしている。その時に，ポインタのあるマスの範囲を強調するなどの処理を２描画処理で行っている。

(カ)数値管理

メイン処理～入力処理で出てきた各値の管理をし，メイン処理に呼ばれたときにデータを伝送する。

イ．追加機能のプログラムを作成

特定のマスに置くことで，逆転を狙えるチャンスがあるイベントを作るために必要な機能のサブプログラムを作成。

ゲーム開始時にランダムに１０を２つ配置し，そのマスを選択することで条件処理を行い，そのマスに石が置かれた際，空いているマスに一つ自分の石を置くことができる。

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	-1	1	0	10	0
0	0	0	1	-1	0	0	0
0	10	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

図４　ランダム配置処理

ウ．デバッグ

プログラムのエラーや，バグを修正する。

下図のように,エラーが表示されるので，そのエラー箇所を訂正し，エラーを消していく。

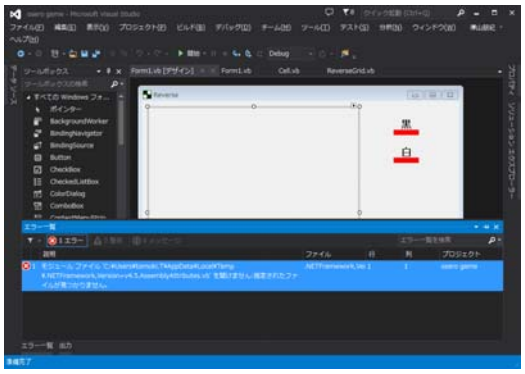

図５　デバッグ画面

３．研究のまとめ

オセロゲームを作るには，主にメイン処理プログラムと，５つのサブプログラムが必要になってくることがわかった。

マスを配列処理にし，その都度データを比較することで様々な機能をつけることができた。また，計算量を少なくすることができた。

プログラムは，念入りな設計と各機能わけてサブプログラムを作ることで，エラーやバグなどを減らすことができることがわかった。

また，プログラム自体が壊れても最小限の被害で抑えることができ，各機能ごとにわけることでサブプログラム上でミスを見つけることができる。

オセロゲームはバグなく完成し，図６ように，条件に当てはまったとき一つ自分の望む場所に石を置くことができるようになった。

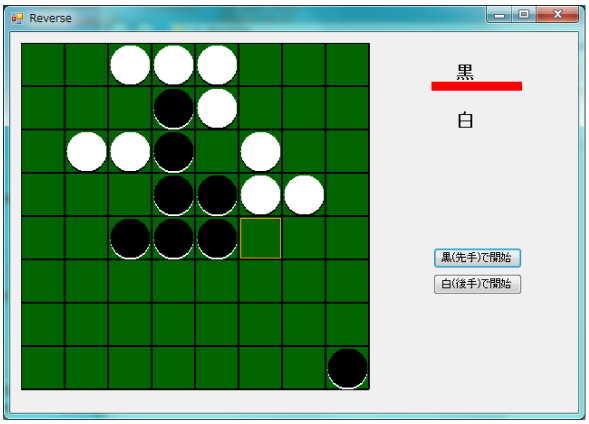

図６　完成図

４．感想

今回オセロゲームを作ろうと考えた理由は，以前からゲームソフトの作成に興味があり，課題研究で作る機会があったので，一から調べ，理解しながら作ってみようと思い，着手した。

オセロゲームを作る過程で，ただのオセロを作るだけでは面白くないと思い，オセロにアタックチャンスをつけようと試行錯誤した結果，Visual Studio(Basic)でのソフトウェア開発の難しさを痛感するとともに着実に知識が身についていくのがわかり，作っていてとても面白く感じた。また，自分が作ったソフトをほかの人に楽しんでもらえるように扱いやすくGUI（グラフィカルユーザインターフェイス）を調整し，それに合わせたプログラムを作成することがいかに難しいことかを理解することができた。

５．参考文献

・Othello Game
http://www.cc.kyoto-su.ac.jp/~yamada/ap/othello.html

・簡単オセロ制作
http://idehideout.fc2web.com/p/rev/00.html

・VB 実技４ オセロ - nifty
http://homepage1.nifty.com/rucio/main/dotnet/shokyu/standard52.htm

・オセロ・リバーシプログラミング講座～勝ち方・考え方～
http://uguisu.skr.jp/othello/

・Visual Studio (Basic) 2010 でオセロゲーム:水より柔弱
http://kanakatsu.cocolog-nifty.com/blog/2012/11/for-vbnet-2012-.html

VB でオセロゲーム(コンピューター同市の対戦) -Gizcollabo
http://www.gizcollabo.jp/vbtomo/log/archive/vbqanda_2233_1.html