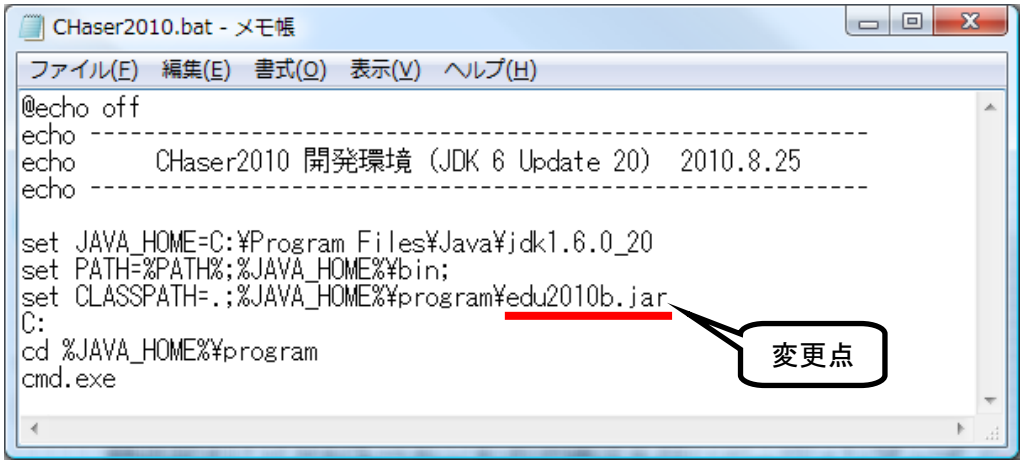
全国高校生プログラミングコンテスト

CHaser2010

ステップアップヒント３

14 競技部品（edu2010b.jar）へのバージョンアップについて

競技部品にバグがみつかったので修正を行い、a から b へバージョンアップしました。全情研のホームページから新しい競技部品をダウンロードしてファイルを置き換えてください。また、画面 25 を見ながらバッチファイルの変更も行ってください。



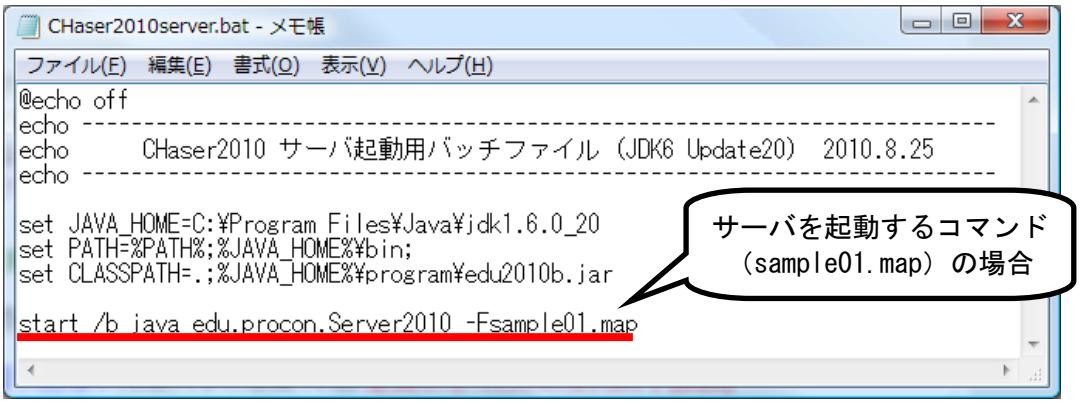
【画面 25】バッチファイルの変更点

15 競技サーバを起動するバッチファイルの作成

今まで競技サーバを起動するには、バッチファイル（CHaser2010.bat）をダブルクリックしてコマンドを打っていましたが、競技のたびにコマンドを打たなくても済むように「サーバを起動するバッチファイル」を作ります。

program フォルダの下に、CHaser2010server.bat を作り、メモ帳などで画面 26 のように入力します（CHaser2010.bat をコピーして利用すると簡単です）。

マップファイルを複数作った場合は、バッチファイルをコピーしてマップファイル名の部分を変更して保存しておくと便利です。



【画面 26】サーバ起動用バッチファイル

16 サンプルプログラム３（アイテム回収と相手への処理を追加したプログラム）

①プログラムの保存先

program フォルダの下に「sample2010_03」という名前のフォルダを作り、次のプログラムを入力します。プログラム名は「sample2010_03.java」です。

②プログラムの動作

・クール専用のプログラムで、基本の動きは sample2010_02 と同じです。
・相手が上下左右にいた場合、その方向にブロックをのせます（put）。
・アイテムが上下左右にあった場合、その方向に移動（walk）し、アイテムを回収します。
・old_mode という変数を作り、１つ前のモードを記憶させています。

③サンプルプログラム３（sample2010_03.java）

```
1  /*******************************************************
2    sample2010_03.java
3  *******************************************************/
4  public class sample2010_03 {
5      public static void main(String[] args) {
6          int[] value;
7          value = new int[10];
8
9          int mode = 1,      // 現在のモード
10             old_mode = 1; // 前のモード
11
12         /********** 競技サーバに接続する **********/
13         edu.procon.Connect2010 target;
14         target = new edu.procon.Connect2010("サンプ３", Integer.parseInt(args[0]));
15
16         while (true) {
17             value = target.getReady( );
18             if (value[0] == 0) break;
19
20             /********** 周囲（上下左右）に相手がいるかチェック **********/
21             if(value[2]==1 || value[4]==1 || value[6]==1 || value[8]==1){
22                 mode = 90;        // mode を 90 に変更する
23             }
24
25             /********** 周囲（上下左右）にアイテムがあるかチェック **********/
26             if(value[2]==3 || value[4]==3 || value[6]==3 || value[8]==3){
27                 old_mode = mode; // 現在のモードを old_mode に保存する
28                 mode = 20;        // mode を 20 に変更する
29             }
30
31             /********** mode の値で分岐する **********/
32             switch (mode) {
33                 case 1:  // ブロック（壁）にぶつかるまで下に移動する
34                     if(value[8] != 2){                // 下が壁でなければ、
35                         value = target.walkDown( );  // 右に移動する
36                     }
37                     else{
38                         value = target.walkRight( ); // 下が壁ならば、右に移動し、
39                         mode = 2;                     // mode を 2 に変更する
```

```
40                      }
41                      break;
42                  case 2:  // ブロック（壁）にぶつかるまで右に移動する
43                      if(value[6] != 2){
44                          value = target.walkRight( );
45                      }
46                      else{
47                          value = target.walkUp( );
48                          mode = 3;
49                      }
50                      break;
51                  case 3:  // ブロック（壁）にぶつかるまで上に移動する
52                      if(value[2] != 2){
53                          value = target.walkUp( );
54                      }
55                      else{
56                          value = target.walkLeft( );
57                          mode = 4;
58                      }
59                      break;
60                  case 4:  // ブロック（壁）にぶつかるまで左に移動する
61                      if(value[4] != 2){
62                          value = target.walkLeft( );
63                      }
64                      else{
65                          value = target.walkDown( );
66                          mode = 1;
67                      }
68                      break;
69                  case 20: // 周囲にアイテムがあったら、取りに行く
70                      // 上にアイテムがあったら、上に移動する
71                      if(value[2] == 3)       value = target.walkUp( );
72                      // 左にアイテムがあったら、左に移動する
73                      else if(value[4] == 3) value = target.walkLeft( );
74                      // 右にアイテムがあったら、右に移動する
75                      else if(value[6] == 3) value = target.walkRight( );
76                      // 下にアイテムがあったら、下に移動する
77                      else                    value = target.walkDown( );
78                      mode = old_mode; // モードを元に戻す
79                      break;
80                  case 90: // 周囲に相手がいたら、ブロックをのせる
81                      // 上に相手がいたら、上にブロックをのせる
82                      if(value[2] == 1)       value = target.putUp();
83                      // 左に相手がいたら、左にブロックをのせる
84                      else if(value[4] == 1) value = target.putLeft();
85                      // 右に相手がいたら、右にブロックをのせる
86                      else if(value[6] == 1) value = target.putRight();
87                      // 下に相手がいたら、下にブロックをのせる
88                      else                    value = target.putDown();
89                      break;
90              }
91              /********** 制御情報が 0 だったら終了する **********/
92              if(value[0] == 0) break;
93          }
94
95          /********** 競技サーバから切断する **********/
```

```
96          target.exit( );
97      }
98  }
```

⑤プログラム説明

・変数の宣言

```
10      int mode = 1,      // 現在のモード
11          old_mode = 1; // 前のモード
```

動作の種類を記憶させる mode を宣言します。old_mode はアイテム回収処理をした後に元のモードに戻るために必要です。

・チーム名

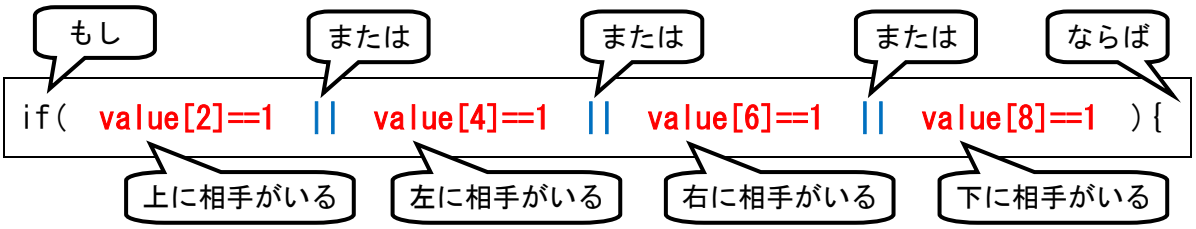
```
14      target = new edu.procon.Connect2010("サンプ３", Integer.parseInt(args[0]));
```

edu.procon.Connect2010 クラスを使って競技サーバに接続します。どのプログラムが対戦しているかわかるようにするため、チーム名を「**サンプ３**」にしました（変更しても構いません）。

・周囲（上下左右）に相手がいるかチェック

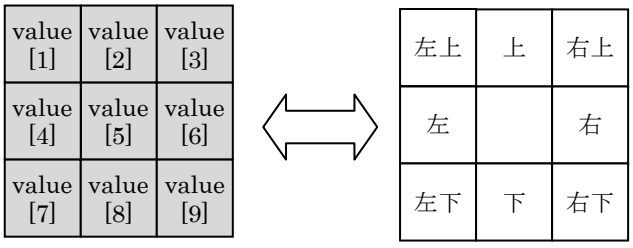
```
20      /********** 周囲（上下左右）に相手がいるかチェック **********/
21      if(value[2]==1 || value[4]==1 || value[6]==1 || value[8]==1){
22          mode = 90;        // mode を 90 に変更する
23      }
```

CHaser2010 は、相手にブロックをのせて（put）勝利することが一番の目的です。サーバから戻り値で相手は「1」なので、if 文を使って判断します。「value[2]==1」は上に相手がいるか？という意味で、「||」は論理和（OR）という意味で論理演算子といいます。



【図 6】if 文の意味

図 6 をまとめると「もし、上下左右に相手がいるならば」という意味になります。
その場合は、mode を 90 にします。



【図 7】サーバの戻り値の格納先

・mode=90 の処理

```
80      case 90: // 周囲に相手がいたら、ブロックをのせる
81          // 上に相手がいたら、上にブロックをのせる
82          if(value[2] == 1)       value = target.putUp();
83          // 左に相手がいたら、左にブロックをのせる
84          else if(value[4] == 1) value = target.putLeft();
85          // 右に相手がいたら、右にブロックをのせる
86          else if(value[6] == 1) value = target.putRight();
87          // 下に相手がいたら、下にブロックをのせる
88          else                    value = target.putDown();
89          break;
```

switch 文を使って mode の値で分岐処理を行います。mode=90 の場合、上下左右に相手がいるので、相手のいる方向にブロックをのせて（put）勝利となります。

・周囲（上下左右）にアイテムがあるかチェック

```
25      /********** 周囲（上下左右）にアイテムがあるかチェック **********/
26      if(value[2]==3 || value[4]==3 || value[6]==3 || value[8]==3){
27          old_mode = mode; // 現在のモードを old_mode に保存する
28          mode = 20;       // mode を 20 に変更する
29      }
```

相手の処理と同様に if 文を使用して周囲にアイテム（3）があるか判断します。アイテムを回収したあと元のモードに戻るため old_mode に現在のモードを格納し、mode を 20 にします。

・mode=20 の処理

```
69      case 20: // 周囲にアイテムがあったら、取りに行く
70          // 上にアイテムがあったら、上に移動する
71          if(value[2] == 3)       value = target.walkUp( );
72          // 左にアイテムがあったら、左に移動する
73          else if(value[4] == 3) value = target.walkLeft( );
74          // 右にアイテムがあったら、右に移動する
75          else if(value[6] == 3) value = target.walkRight( );
76          // 下にアイテムがあったら、下に移動する
77          else                    value = target.walkDown( );
78          mode = old_mode; // モードを元に戻す
79          break;
```

mode=20 の場合、上下左右にアイテムがあるので、その方向に移動（walk）します。アイテムを回収したあと、元のモードに戻るため old_mode の値を mode に格納します。

現在、使用しているモードは表 4 の通りです。

【表 4】モードの種類と動作

mode	動　作
1	ブロックにぶつかるまで上に移動する
2	ブロックにぶつかるまで右に移動する
3	ブロックにぶつかるまで下に移動する
4	ブロックにぶつかるまで左に移動する
20（新規）	周囲（上下左右）のアイテムを回収する（walk）
90（新規）	周囲（上下左右）の相手にブロックをのせる（put）

17 発展

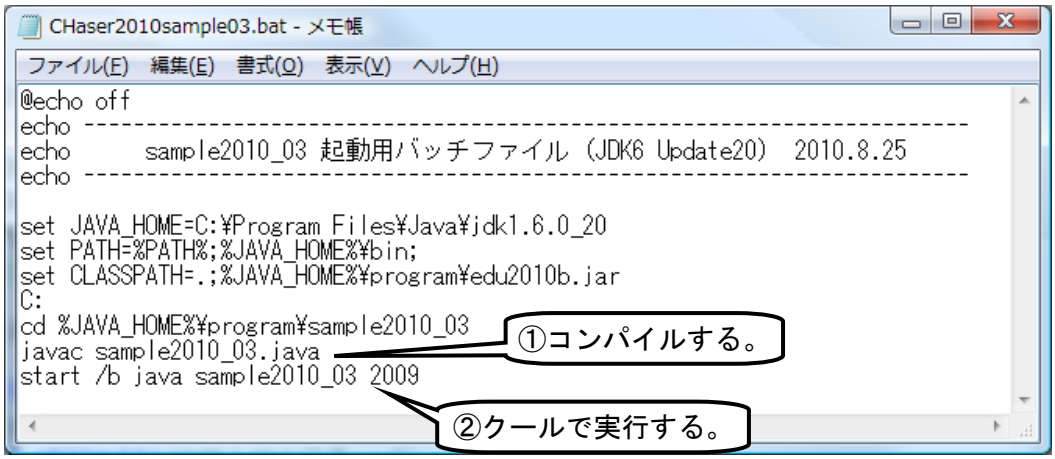
今回追加したプログラムは上下左右しか対応していないので、斜め方向（右上、右下、左上、左下）に対応させなければなりません。斜め方向は１つのターンでは処理できないので新たにモードを作るべきです。

特に、相手が斜め方向にいる場合、相手も自分がいることがわかるため慎重にメソッドを選択しなければなりません。また、プログラムがループしてしまわないよう注意してください。

18 サンプルプログラム３を起動するバッチファイルの作成

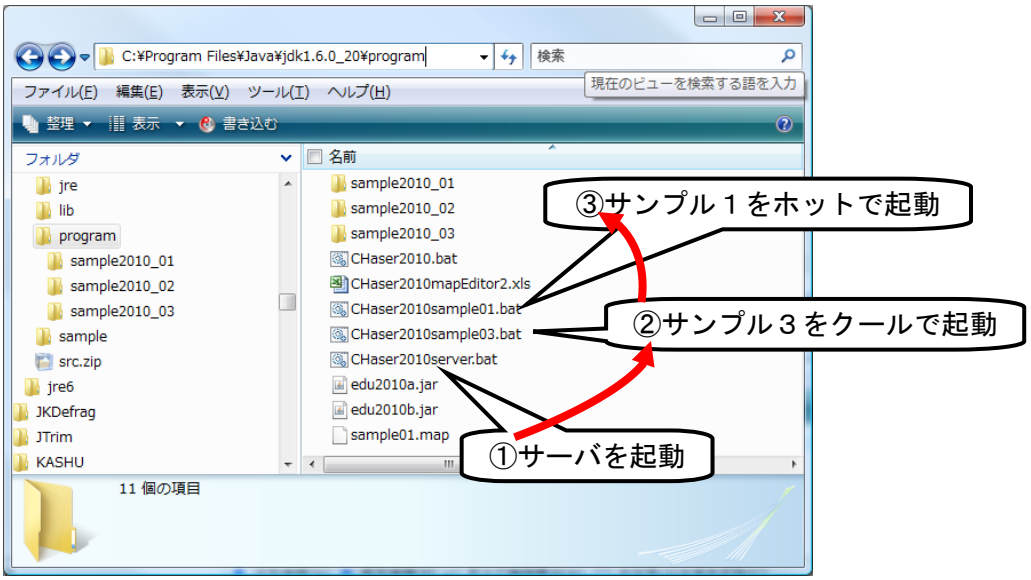
競技サーバを起動するバッチファイルと同様に「サンプルプログラム３を起動するバッチファイル」を作ります。

program フォルダの下に、CHaser2010sample03.bat を作り、メモ帳などで画面 27 のように入力します。



【画面 27】サンプルプログラム３起動用バッチファイル

このバッチファイルは sample2010_03.java をコンパイルしてから実行します。もし、エラーがある場合は実行せずに終了します。筆者の場合、sample2010_01.java をホットで起動するようにバッチファイルを作成して、バッチファイルをクリックするだけで対戦ができるようにしています。



【画面 28】バッチファイルを使用した対戦