

プログラミング応用
知的エージェントの設計

担当・鳥海不二夫

知的エージェントとは？

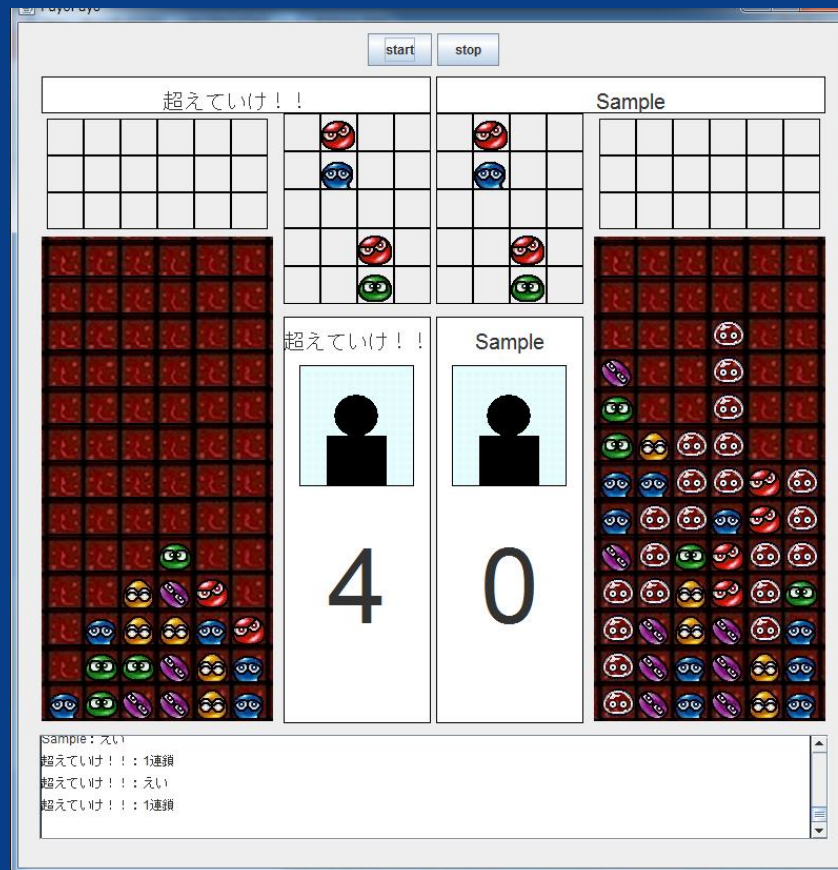
- 人工知能を有するソフトウェア
 - 人間に代わって作業を行うエージェント
 - 自律的に行動を行う
- 知的エージェントの例
 - 対話エージェント
 - WEBクローラ
 - ゲームエージェント(将棋・シミュレーションゲーム)

本講義で扱う知的エージェント

- 実世界での知的エージェント実装は困難
 - 世界には情報があふれている
 - フレーム問題
- 特定の環境下で自分で考え行動を行うエージェントを設計する
 - 環境情報の取得
 - 行動の決定
- 限られた環境を提供
 - 目的指向型
 - ある目的の達成力を競う

今回のターゲット

- ぶよぶよ



今回のターゲット

- ふよふよ
 - ふよふよの敵を作る
 - ふよふよの敵=知的エージェント
- なぜふよふよか？
 - ふよふよはNP完全問題
 - 牟田 秀俊 (2005), “ふよふよはNP完全”, *IEICE technical report. Theoretical foundations of Computing* **105** (72): 39-44
 - ゲーム性があるって面白い

ぷよぷよ的知的エージェント

- 環境
 - 6x13マスの世界
 - 対戦相手情報
 - 次に降ってくるぷよの色
 - おじゃまぷよの数
- 行動
 - どこに次のぷよを落とすのか

単純化された環境と行動

このプロジェクトの概要

- 第1週： ふよふよ作成環境の整備
 - 第2週： ふよふよエージェントの動作を確認
 - 第3～5週： エージェントの設計・作成
 - 第4週： プレ大会
 - 第6週： 大会
-
- 成績：大会の結果を加味します

今日の作業

- Eclipseのインストール
- ぶよぶよプロジェクトのインストール
- サンプルの実行

ぷよぷよプロジェクトの作成

- プロジェクトをダウンロード

- <http://syrinx.q.t.u-tokyo.ac.jp/tori/lecture/agent/>
 - Download -> [agent-ver1.5.zip](#)

- EclipseのWorkspaceに展開

- たとえば...
- C:\¥¥workspace¥PuyoPuyoProject.zip
- ↑解凍(lhasaとか)

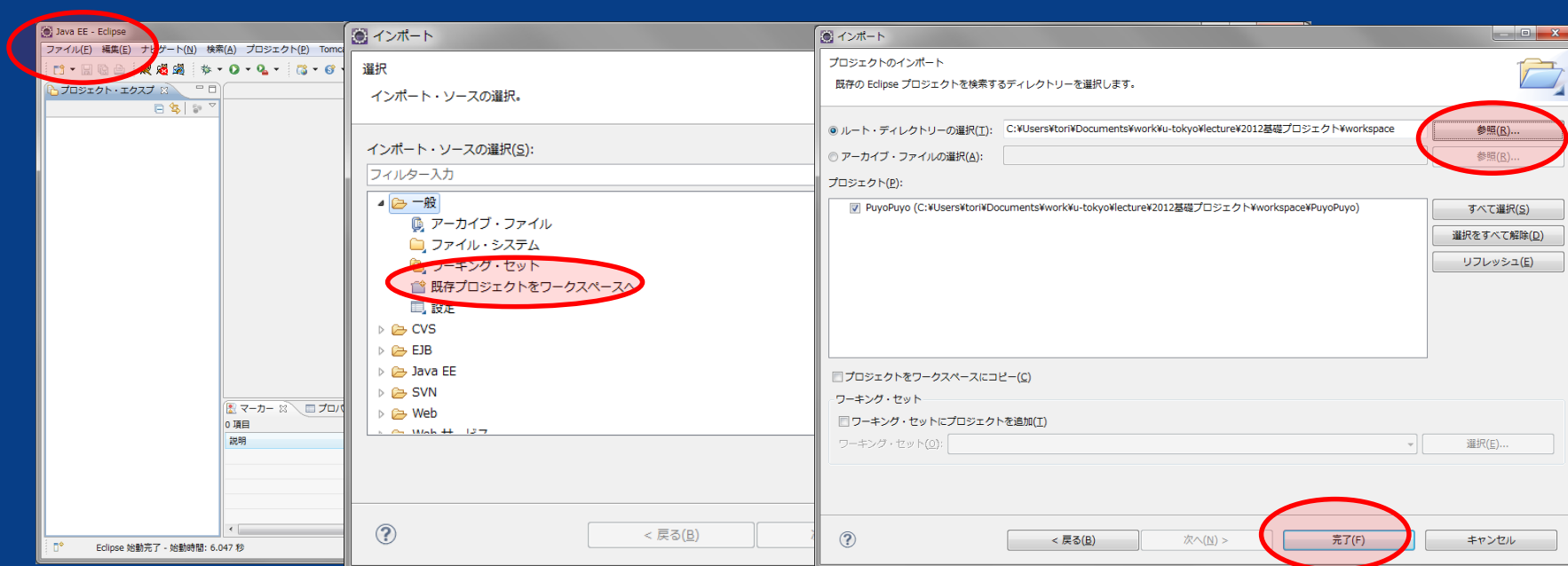
- WORKSPACE/PuyoPuyo/以下にぷよぷよプロジェクトが展開される



プロジェクトのインポート

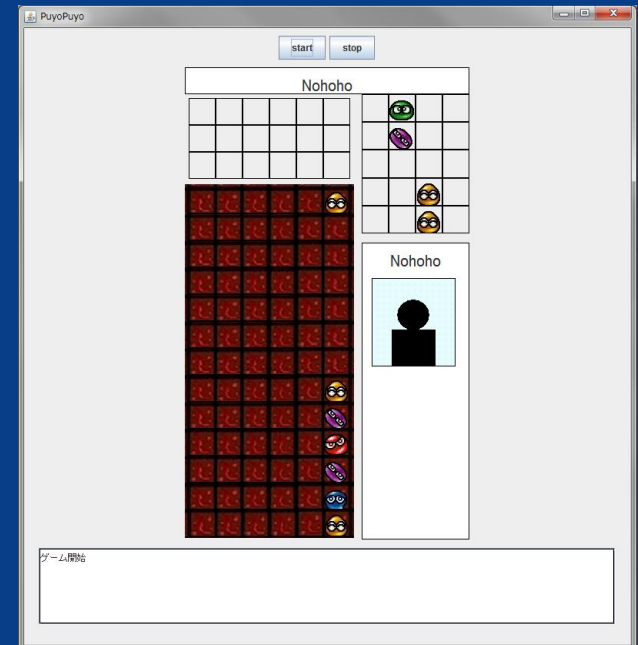
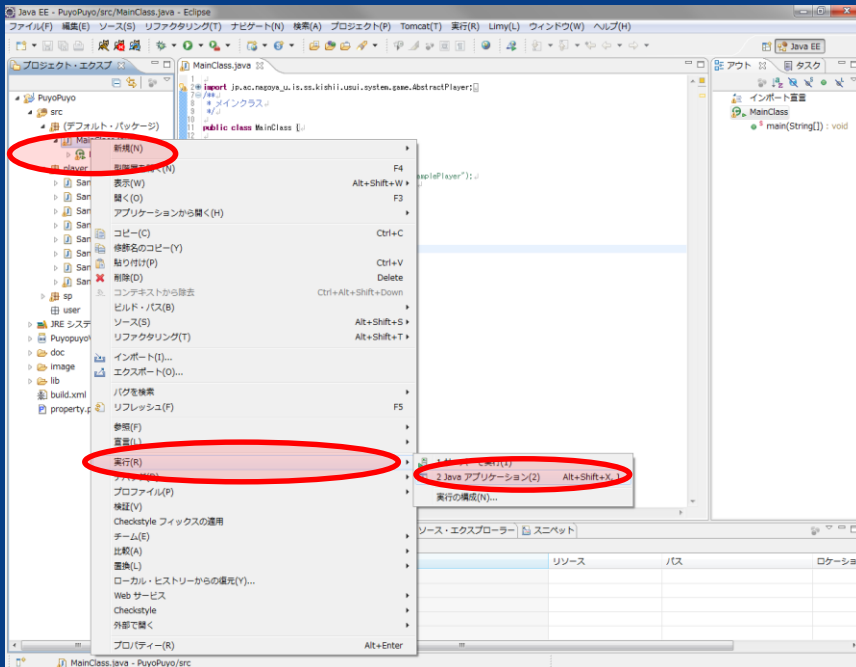
- プロジェクトをインポート

- Eclipseで「ファイル⇒インポート」をクリック
- 「一般⇒既存のプロジェクトをワークスペースへ」を選んで「次へ」
- 先ほどのPuyoPuyoを選んで「完了」



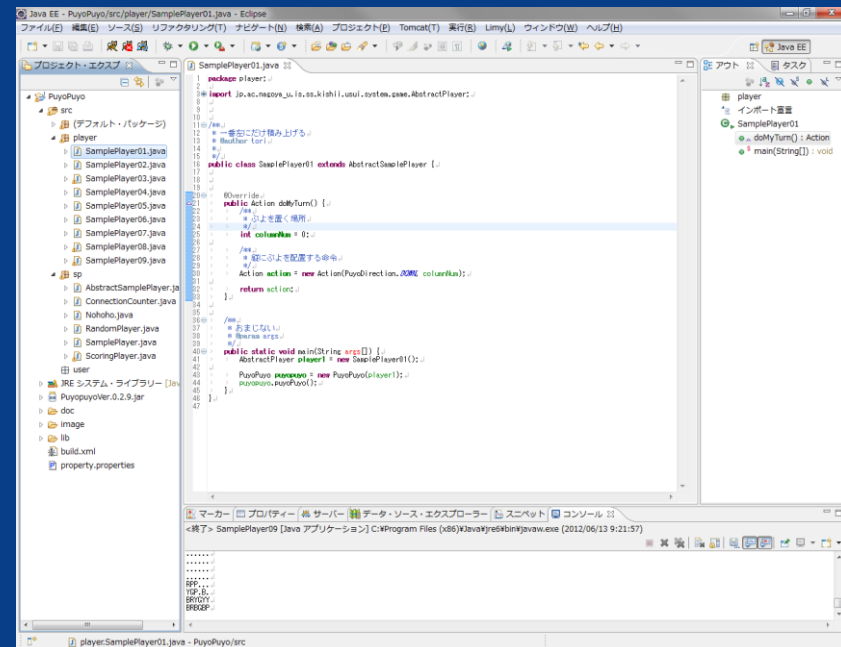
サンプルの実行

- プロジェクトエクスプローラ
 - PuyoPuyo/src/デフォルトパッケージ/MainClass.java
 - 右クリック⇒実行⇒Javaアプリケーション



SamplePlayer01の実行と改良

- パッケージエクスプローラの
 - player/SamplePlayer01をダブルクリック
 - コードを変更してみる
 - MainClassを実行



ぷよぷよを落下させる場所の指定

```
@Override
public Action doMyTurn() {
    /**
     * ぷよを置く場所
     */
    int columnNum = 0;

    /**
     * 縦にぷよを配置する命令
     */
    Action action = new Action(PuyoDirection.DOWN, columnNum);

    return action;
}
```

落下するぷよぷよの向き

落下するぷよぷよの場所

作成したActionクラスのインスタンスをreturnする

盤面の取得

```
@Override
public Action doMyTurn() {
    Board board = getMyBoard();
    Field field = board.getField();

    PuyoType puyoType = field.getPuyoType(0, 0);
    System.out.println(puyoType);
}
```

- 盤面のぷよの色を取得
- PuyoTypeはPuyoの色
 - PuyoType.Blue
 - PuyoType.Redなど
- 何もなければnull



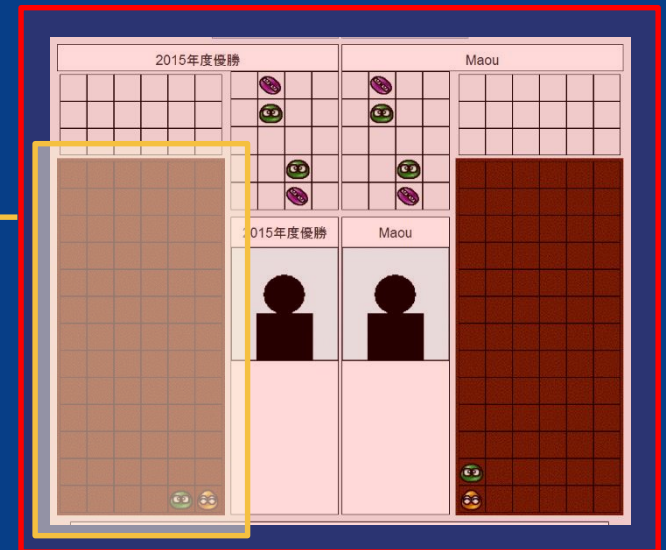
JavaDocを読んでみる

- <http://syrinx.q.t.u-tokyo.ac.jp/tori/lecture/agent/document/doc/index.html>
- JavaDocを読むと何ができるかがわかる



クラス説明

- Board
 - ゲーム全体を管理するクラス
 - 自分や相手のフィールド
 - 次に来るぷよ
 - お邪魔ぷよ
- Field
 - ゲームフィールド
 - 幅や高さ
 - ぷよの状態

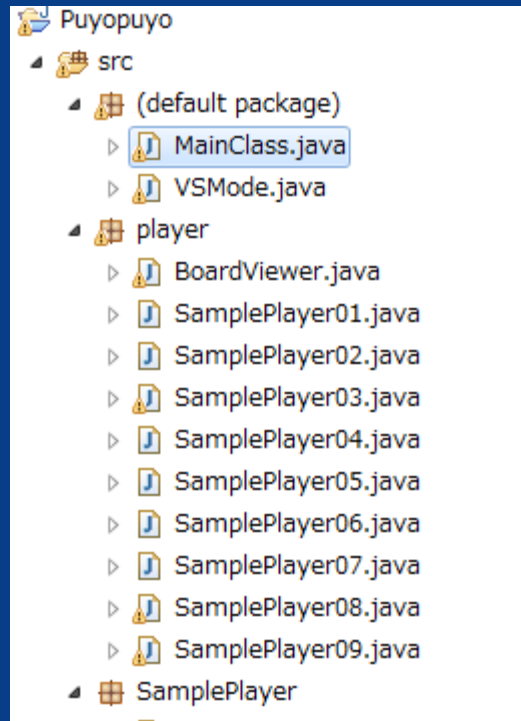


たとえば・・・

- getFieldで得られるFieldクラス
 - getPuyoType(x, y)でx,y座標にあるぷよの種類 (PuyoType)を取得
 - PuyoTypeクラス
 - BLUE_PUYO, YELLOW_PUYO, OJAMA_PUYOなど
 - getNextField(puyo, n)で, n列目にpuyoを落としたら, 次のフィールドがどうなるか取得
 - isOnField(x, y)でx,y座標がフィールド内か取得
- JavaDocを読んでみよう
 - <http://bit.ly/1sPvf3V>

当面の目標

- SamplePlayer01-09までをいじってみよう.



Sample01の場合

- 左端にぷよをおいていくプログラム
 - 右端にしてみよう
 - 真ん中にしてみよう
 - その他色々な場所に変更してみよう

```
public class SamplePlayer01 extends AbstractSamplePlayer {  
  
    @Override  
    public Action doMyTurn() {  
        /**  
         * ぷよを置く場所  
         */  
        int columnNum = 0;
```

今日の目標

- Sample09までできるだけ動かしてみる
 - それぞれのSampleに指示がある
 - 指示に従って改良したり, ソースコードを読み込んだりしよう
- メソッドについては
 - <http://bit.ly/25gxMBE> (全メソッド)
 - <http://bit.ly/1sPvf3V> (Fieldが持つメソッド)を参考に

今後の展開

- 自分なりのプレイヤーを作成する
 - Javaの勉強は各自で行うこと
- 強いものができたと思ったら、対戦させてみる
 - デフォルトのパッケージ/VSMoDeに実行のさせ方のサンプルがある
 - Nohoho: ただの雑魚
 - TA1(UsuiPlayer): やっぱり雑魚
 - TA2(UsuiPlayerLV2): これに勝てなければ不可
 - TA3(UsuiPlayerLV3): この辺になら勝てるはず
 - KajiGodGod: これに勝てればOk
 - maou: かつて最強だったエージェント
 - hiraUltimate: 昨年ぶっちぎった最強エージェント
- 最終回に大会を行う
 - 大会の結果は成績に加味される