

プログラミング応用
知的エージェントの設計
後半第2回

担当・鳥海不二夫

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2～5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

- Sample04～sample09
- Sample04:ランダム配置
- Sample05:一番ぶよの数が少ない列へ配置
- Sample06:同じ色に重なるように配置
- Sample07:次のFieldを使って効率のよい配置
- Sample08:同じ色と隣接するように配置

4ぶよを接続させるために

- ある状態でぶよを落としたら, 次にどうなるか
 - `Field.getNextField()`を利用

`Field nextField = field.getNextField(puyo, i);`

あるぶよを*i*番目の列に落としたとき, どのような状態になるかが`nextField`に格納される.

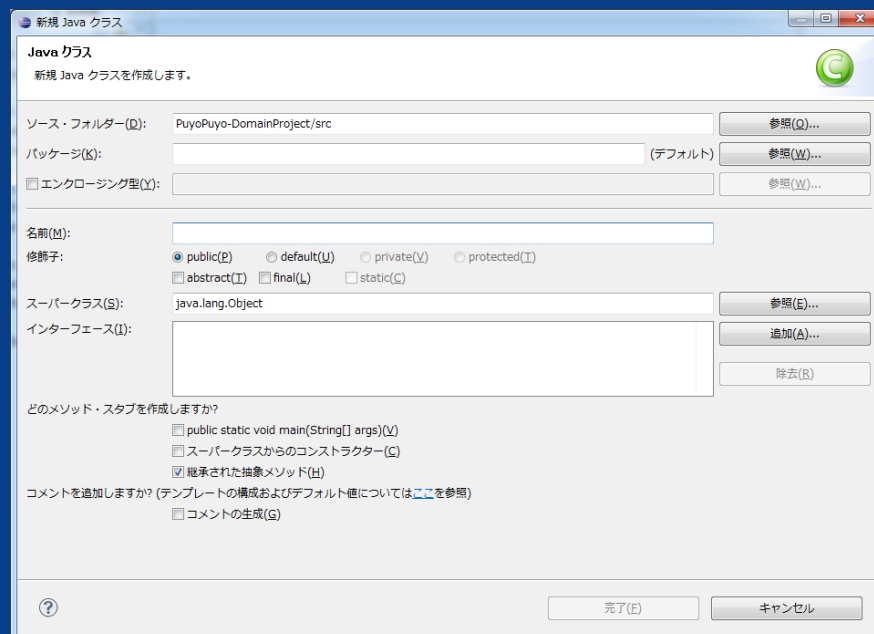
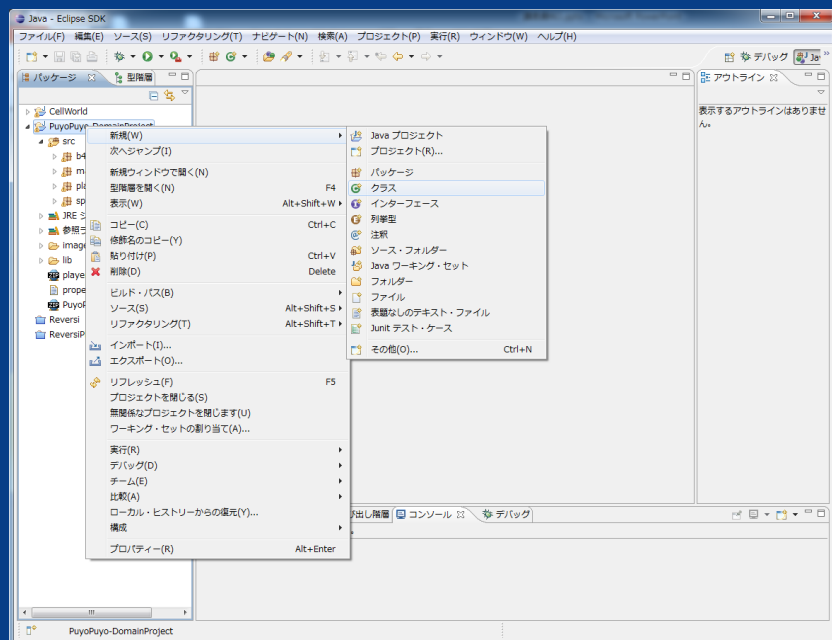
- `nextField`からぶよが消えるか数えられる

Sample07が基本となる

- SamplePlayer07をチェック！
 - Sample07を改良すると楽
 - 「次の状態」がベストになるようにすべし
- あんまり色々やると時間がかかる
 - 一回につき1秒以上かかると大会で動かない

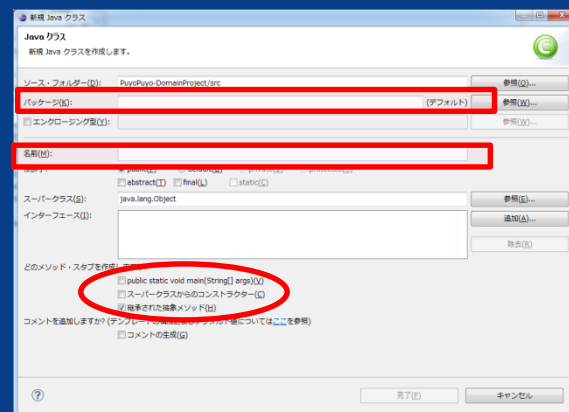
自分だけのプレイヤーを作る

- パッケージエクスプローラでメニューを表示
– 新規作成⇒クラス



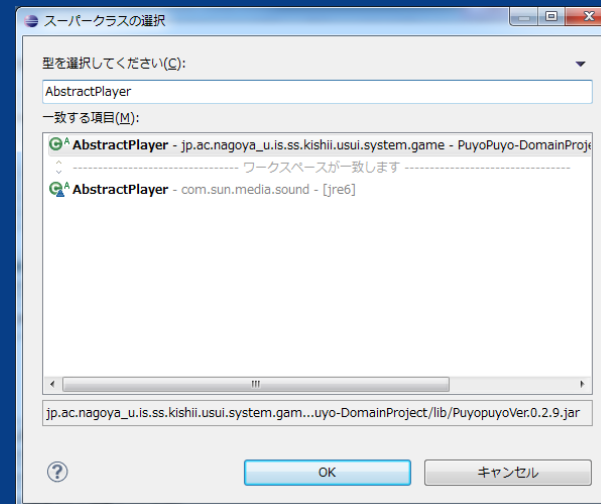
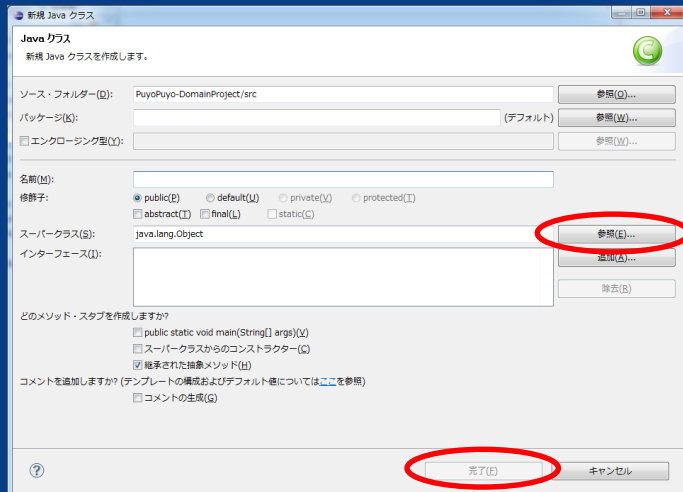
自分だけのプレイヤーを作る(1)

- パッケージ: 自分だけのパッケージ名
 - 自分のメールアドレスを逆から @を.にしたもの
 - tori@sys.t.u-tokyo.ac.jp
 - jp.ac.utokyo.t.sys.tori
- 名前: クラス名
 - ToriPlayerなど.
 - 各単語の最初の文字を大文字にしたもの
- static public void mainにチェック
- スーパークラスからのコンストラクタにチェック



自分だけのプレイヤーを作る(2)

- スーパークラス (SuperClass) を決める
 - 参照ボタンをクリック
 - 型に「AbstractPlayer」と記入
 - 「jp.ac.nagoya_u.is.ss.kishii.usui.system.game.AbstractPlayer」を選択
- 完了



クラスができる

```
ToriPlayer.java
package jp.ac.utokyo.t.sys.tori;

import jp.ac.nagoya_u.is.ss.kishii.usui.system.game.AbstractPlayer;

public class ToriPlayer extends AbstractPlayer {

    public ToriPlayer(String playerName) {
        super(playerName);
        // TODO 自動生成されたコンストラクター・スタブ
    }

    @Override
    public Action doMyTurn() {
        // TODO 自動生成されたメソッド・スタブ
        return null;
    }

    @Override
    public void initialize() {
        // TODO 自動生成されたメソッド・スタブ
    }

    @Override
    public void inputResult() {
        // TODO 自動生成されたメソッド・スタブ
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
    }
}
```

実行のためのおまじない

```
/**
 * @param args
 */
public static void main(String[] args) {
    AbstractPlayer player = new ToriPlayer("Tori's Agent");

    PuyoPuyo puyopuyo = new PuyoPuyo(player);
    puyopuyo.puyoPuyo();
}
```

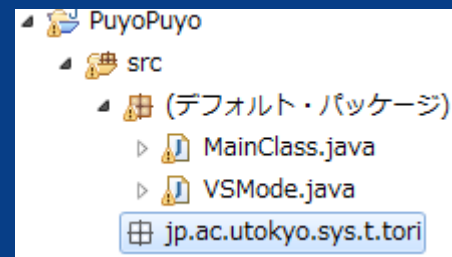
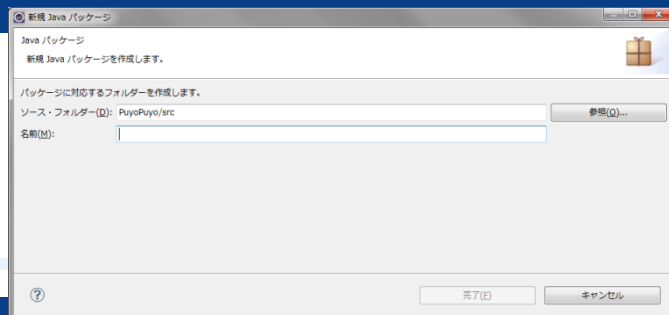
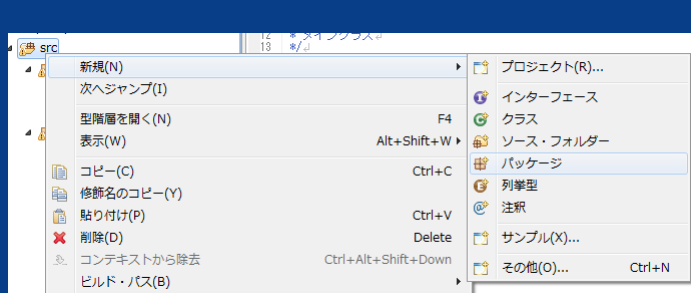
または, MainClass.javaを書き換える

エージェントを作る

- doMyTurnを書く
 - これまでに書いてきたものをコピペすればOk
- initialize(), inputResult()は無視して良い

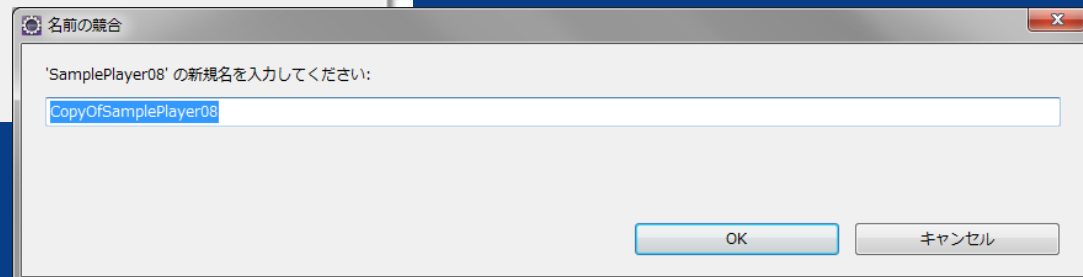
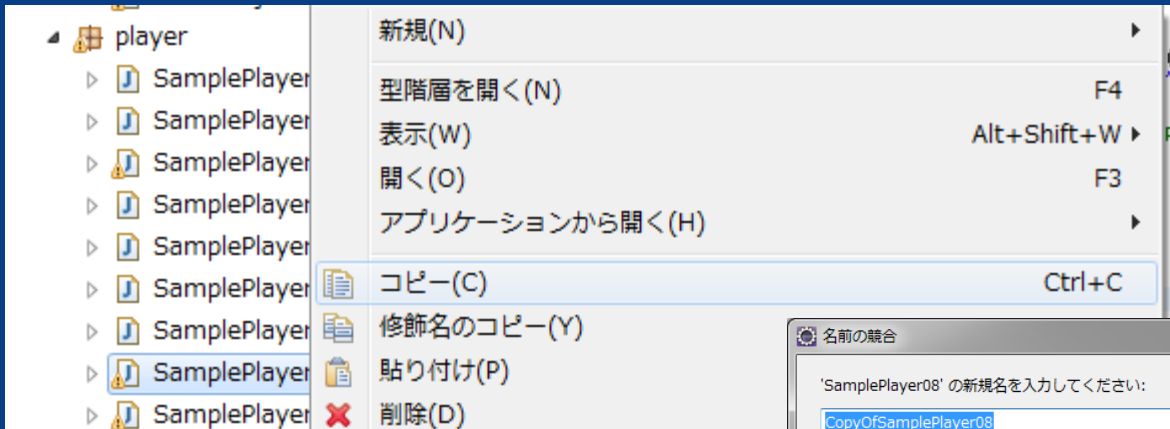
今までのエージェントをコピーする(1)

- srcフォルダで右クリック
 - 新規作成⇒パッケージ
- パッケージ名を入れる
 - 先ほどと同じメールアドレスを逆から
- 新しいパッケージができる



今までのエージェントをコピーする(2)

- コピーしたいPlayerクラス上で
右クリック⇒コピー(copy)
- 先ほどのパッケージで「貼り付け(paste)」
- 適当なクラス名をつける



エージェントを持ち運ぶ

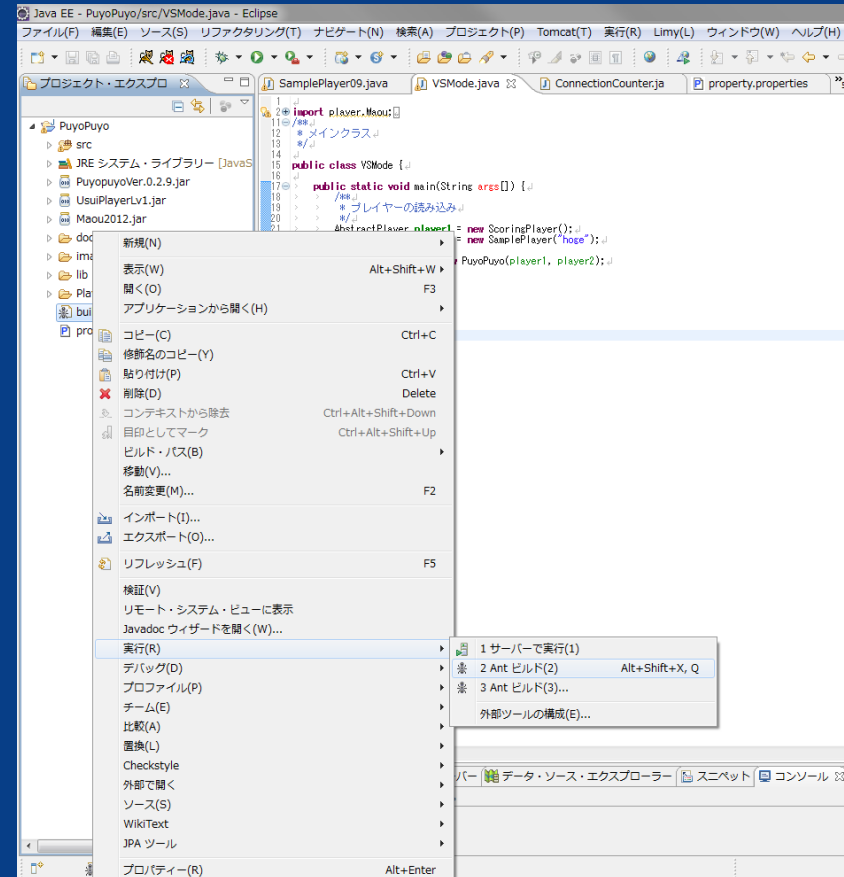
- Javaのプロジェクトはjarファイルというライブラリで移動可能
 - 自分のエージェントのjarファイルを作れば, 他人と対戦が可能

Jarファイルの設定をする

- build.xmlを開く
- `<property name="jar.name" value="./PuyoPuyo-YourName.jar"/>`
- PuyoPuyo-YourName.jarの名前を変更する
 - PuyoPuyo-Tori.jar

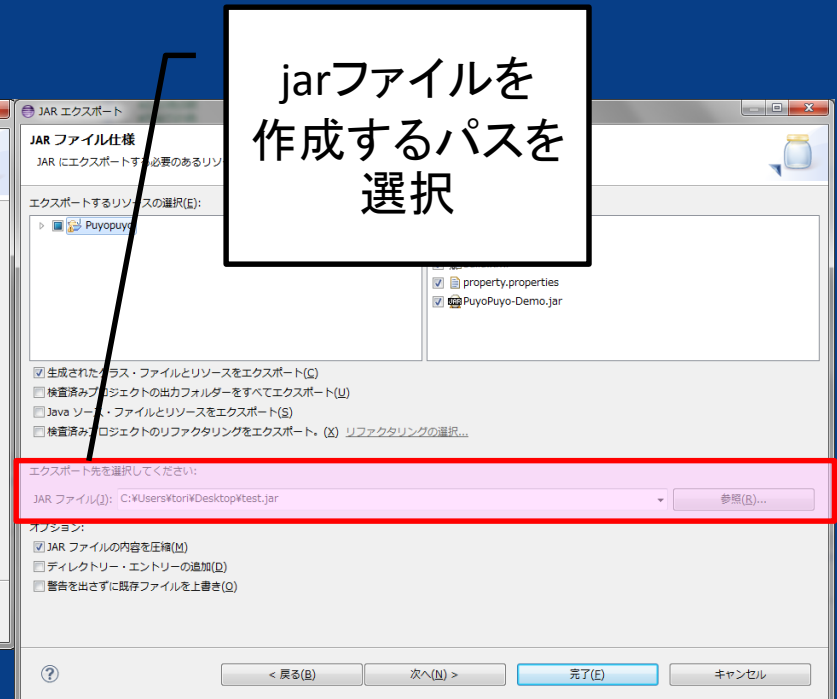
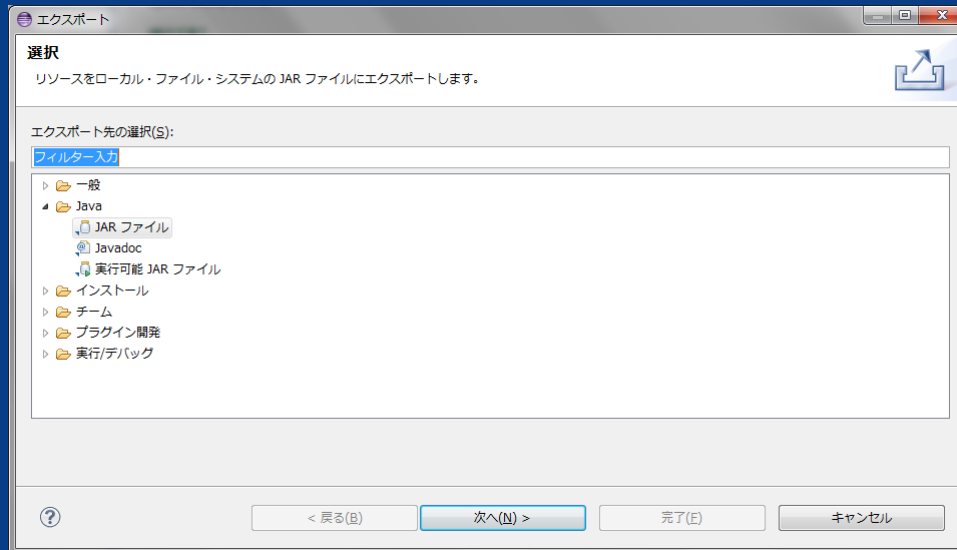
Jarファイルを作る

- build.xmlで右クリック
- 実行⇒Antビルド
- PuyoPuyo-YourName.jar
ができる



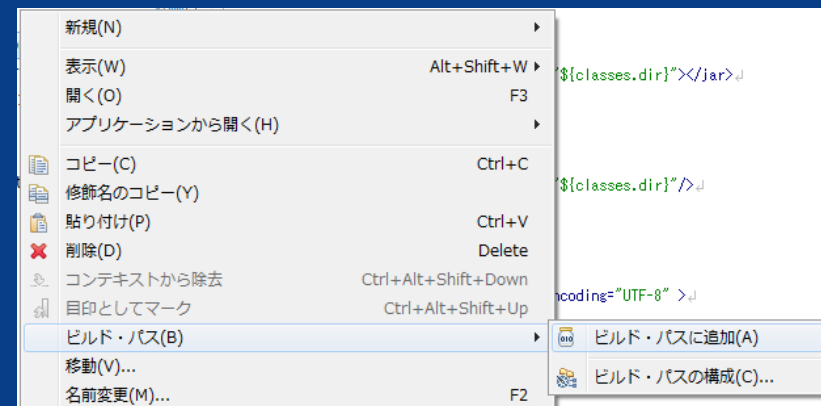
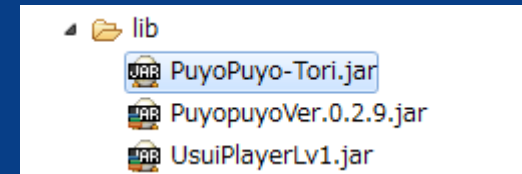
Jarファイルを作る2 出来なかった人用

- プロジェクトで右クリック
 - エクスポート
 - jarファイル



他人のエージェントと対戦する(1)

- 他の人が作ったjarファイルをUSBメモリなどでもらう
- libフォルダにコピーする
- 右クリック⇒ビルドパス⇒ビルドパスに追加



他人のエージェントと対戦する(2)

- VSModeを開く
 - Player1, player2を対戦させたいエージェントに変更する
 - player1を自分の作ったエージェント
 - player2を相手の作ったエージェント
- VSModeを実行する

```
public class VSMode {  
    < <  
    < public static void main(String args[]) {  
    < < < /**  
    < < < * プレイヤーの読み込み  
    < < < */  
    < < < AbstractPlayer player1 = new ScoringPlayer();  
    < < < AbstractPlayer player2 = new SamplePlayer("hoge");  
    < < < PuyoPuyo puyopuyo = new PuyoPuyo(player1, player2);  
    < < < puyopuyo.puyoPuyo();  
    < < < }  
    < }  
}
```