

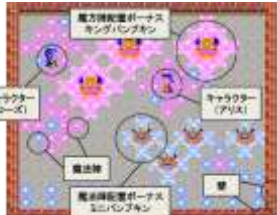


AIプログラミングを通して参加する教育向けゲームシステムに適したソフトウェアアーキテクチャ

早稲田大学 情報理工学専攻 鷲崎研究室 坂本 一憲 kazuu@ruri.waseda.jp
http://exkazuu.jimdo.com/

ゲーム概要

- **コンセプト**: AI対戦型陣取りゲーム
ユーザーが作ったAIプログラム同士で競い合う
- **ルール**: ターン制, **AI動作**: 上下左右移動と陣取り行為
- ICPC 2009 アジア地区 / 2010 国内予選の余興で開催



- 足元を陣地化
- 特定図形の陣地作成
- 2010版

- ボンバーマン風
- 爆弾配置
- 爆風で陣取り
- 2009版

要求の洗い出し

- AIプログラムによるチート防止
情報取得の制限(非公開な情報)
情報設定の制限(書き換え禁止)

- AIプログラムのバグ対処
AIが無限ループしても落ちない

- AI実装が簡単, API理解の支援

- AI持ち時間の制限(一定周期描画)

- ゲームシステムメンテが容易
シンプル化, 録画が容易

品質特性, 副特性

- **機能性**: セキュリティ
- **信頼性**: 障害許容性
- **使用性**: 理解, 習得性
- **効率性**: 時間効率性
- **保守性**: 解析, 変更性
- **移植性**: 該当なし

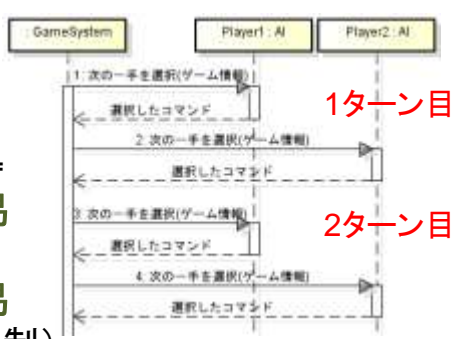
要求の抜けを防止

ゲームシステム開発におけるノウハウ集(パターン)

システム概観(アーキテクチャパターン)

- AIは1メソッドで表現
- Javaのinterfaceを提供
- implement sしてAIを実装
⇒ IDEのIntelliSence利用可
- ✓ **API理解の支援**
- システムからAIを呼び出す
⇒ AIは1引数のみから情報取得
- ✓ **情報取得の制限, 実装が容易**
- ⇒ AIは戻り値のみで動作選択
- ✓ **情報設定の制限, 実装が容易**
- 交互に次の一手を選択(ターン制)
- ✓ **シンプル化, 録画が容易**

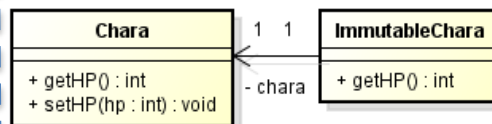
```
public interface Player {
    Action execute(GameInfo info);
}
```



ターン制の様子

クラス設計(デザインパターン)

- イミュータブルオブジェクト(書き込めない不変オブジェクト)
- ✓ **AIによるチート防止**
- 不変オブジェクトでラッピング
⇒ 既存オブジェクトを不変化



- CharaオブジェクトをAIに渡す
× HPを勝手に書き換えられる
- ラップしたImmutableCharaを渡す
○ HPの読み取りのみOK

実装(イディオム)

- AIの手持ち時間の制限
- JavaのThreadを利用
- 一定時間超えると強制終了
- ✓ **AIの持ち時間の制限**
- 別AIのThreadへの干渉防止
- ✓ **AIプログラムのバグ対処**

```
playerThread.start();
playerThread.join(MAX_TIME);
if (playerThread.isAlive()) {
    playerThreads[i].interrupt();
}
```

Javaのイディオム

AIサンプル

```
public class SampleAI implements Player {
    Action execute(GameInfo info) { // infoから全情報取得
        // 自分の得点 > 相手の得点ならば左に移動
        if (info.getMyPoint() > info.getEnemyPoint())
            return Action.Left; // AIの動作は戻り値で通知
        else // 自分の得点 <= 相手の得点ならば右に移動
            return Action.Right; // AIの動作は戻り値で通知
    }
}
```

パターンってなに?

- 繰り返し現れる問題に対する知見と適用可能な解決手法
- ✓ **簡単にいえば再利用可能なノウハウ集だよ**
- 今回の利用パターン: フレームワークパターン(アーキテクチャ), イミュータブルパターン, デコレータパターン(ラッピング)
- 書籍紹介(パターンをより深く勉強したい方)
- ✓ 『Java言語で学ぶデザインパターン入門』
- ✓ 『オブジェクト指向における再利用のためのデザインパターン』

品質特性ってなに?

- ISO9126で規定されたソフトウェアの品質に関する定義
- ✓ **目に見えないソフトウェアの質を明文化**
- 6種類の品質特性, 27種類の品質副特性が存在
- ✓ **品質特性, 品質副特性をチェックシート代わりに使える!**
- 品質の評価や要求の整理に役立つよ(^-^)
- ゲーム開発においても注力すべき点が見えてくるかも?!

宣伝

プログラミングコンテスト
~最強のAIを作ろう!~
in 楽天テクノロジーカンファレンス2010
powered by 早稲田大学 鷲崎研究室 × 楽天
AI開発期間: 2010年09月24日 ~ 10月03日