

ドミニオン A I

現状と展望と課題

現在のプログラムの構成

- DomCore
 - Card, CardData, Supply
 - Player
 - Result
- GaSys
 - GaSysPlayer
- LogLerning
 - LogLerningPlayer
- LoadPlayer
- Params, GaUtils
- DomGUI
 - DomController
- SmithyParamTuner

第 1 章 DomCore

- **DomCore**
 - **Card, CardData, Supply**
 - **Player**
 - **Result**
- GaSys
 - GaSysPlayer
- LogLerning
 - LogLerningPlayer
- LoadPlayer
- Params, GaUtils
- DomGUI
 - DomController
- SmithyParamTuner

DomCoreの役割

- ドミノオンのゲームの統括
- サプライとプレイヤーを所持していてプレイヤーに順番に手番を回す
- ゲームの終了を判定して終了したらResultクラスに情報をまとめる
- 例 : LoadPlayer.java 43行目
- `Result re = core.executeGame(pl, 100, Params.HAND_RANDOM, null);`
- `re.printResult();`
- Plというプレイヤー配列のプレイヤーを使って上限100ターンで初手ランダム,GUIを使わない試合を実行し、得られた結果を表示するプログラム

実際のドミノオンの流れ 1 (コード追いながら)

```
core.executeGame(pl, 100, Params.HAND_RANDOM, null);
```

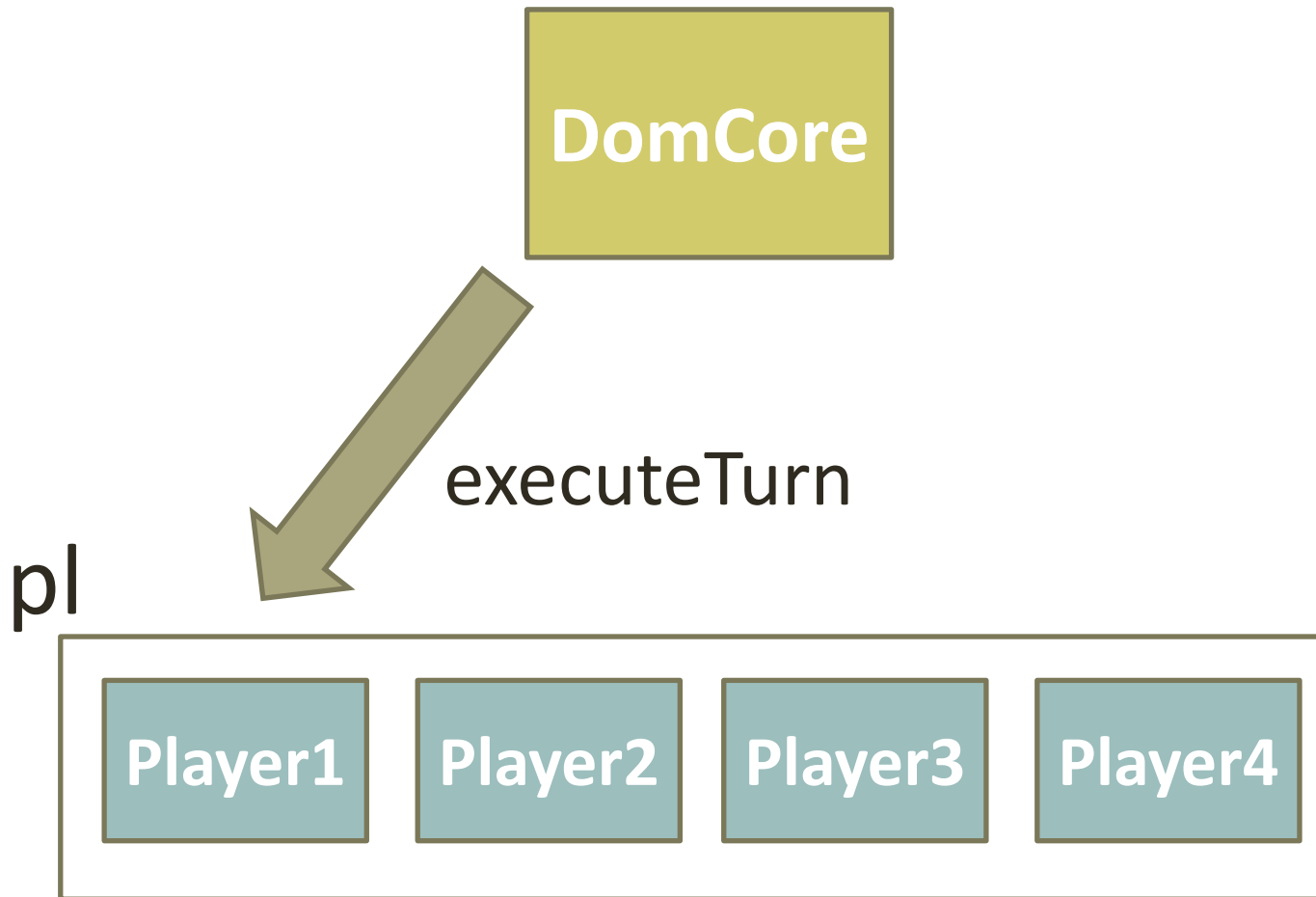


初期化とかなんとか

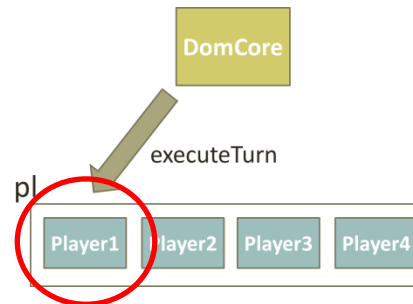
pl



実際のドミノオンの流れ 2



実際のドミノオンの流れ 3



aiTurnの中身は？

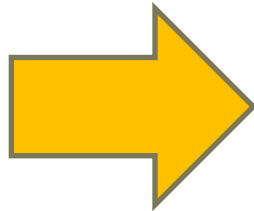
- Playerクラスはあくまで抽象クラス
- Playerクラスを継承するクラスを作ることによって初めてドミニオンをプレイさせることが可能

aiTurnの例 : GaSysPlayerの場合

aiTurn1:アクションフェイズ

GaSysPlayer

useAction()
312行目で呼び出し
本体は88行目から



Player

useCard(card)
528行目
残りアクションがなかったり、
アクションカード以外を指定する
とエラーを返すようになっている

カードの効果の定義はここで
行われている

useCardの例 1 : 屑屋を使おう !

- Player.java 609行目
- `_hand. remove (judgeCard (card.getId ())) ;`
- judgeCardとは(383行目)、カードを使用した時に付随する判断を行う関数 (この場合どのカードを廃棄するか)
- これはTODOにも書いてある通り抽象関数にすべき
- そもそもこの関数の形だと泥棒のようなマルチ入力に対応できないし手先のようなマルチ選択もめんどくさいので改良が必要

useCardの例 2 : 魔女を使おう !

- Player.java 598行目
- `_core.executeAttack(Params. CARD_WITCH, this);`
■
- DomCore 273行目
- 使用者以外全員のPlayer.executeAttackを呼んでいる
- (これ名前一緒なのすごい分かりづらいね ゴミ)
- Player.java 773行目
- まず最初に堀と灯台の処理を行う
- 793行目に魔女の処理が
- 議事堂もここでやってる

aiTurnの例 : GaSysPlayerの場合

aiTurn2:財宝フェイズ

- アクションフェイズでやることが終わったら
- `endActionPhase();`を呼ぶ GaSysPlayer 316行目
- これによりUseCardでアクションが出せなくなり、財宝カードが出せるようになる
- UseCardで財宝カードを渡すだけ
- 例:金貨 570行目

aiTurnの例 : GaSysPlayerの場合

aiTurn3:購入フェイズ

- `endTreasurePhase();`を呼ぶと(326行目)カードがプレイできなくなり購入が可能になる
- 闇なので評価値の計算は後で
- GaSysPlayer 353行目 `buyCard(Card);`
- Player 208行目
- 色々チェックした後買ってる

結果の出力

- 結果はDomCore64行目のgetResultで作成されます
- 要するにResultクラスのインスタンスを作ってるだけ
- Resultクラスには結果をきれいなフォーマットで出力する関数が用意されています

ターン終了！お疲れ様でした やっとドミノオンが出来ます！

- ここまでのスライドに1時間以上かかっている
- ここまでの説明に1時間くらいかかりそう
- コードが汚らわしいレベルで分かりにくい
- 死にたい

ところで忘れてませんか？

今のは

「第一章」

にすぎないということを

第 2 章 GaSys

- DomCore
 - Card, CardData, Supply
 - Player
 - Result
- **GaSys**
 - **GaSysPlayer**
- LogLerning
 - LogLerningPlayer
- LoadPlayer
- Params, GaUtils
- DomGUI
 - DomController
- SmithyParamTuner

GaSysとは

- 遺伝アルゴリズムによってプレイヤーのパラメータを探索するクラス
- GaSysPlayerで使うパラメータを探索する
- 遺伝アルゴリズムは交配して子孫を残して優秀になるというとても素晴らしいアルゴリズムです（説明がめんどくさい）

GaSysの流れ 1

- マルチスレッド化したので多少可読性が落ちてます
- 74行目から（スレッドの中身）
- 80-94はプレイヤーをランダムで選択している
- 同じプレイヤーを選ばないように、また、他のスレッドで使ってるやつは使わないようにしたり極端に弱い奴を途中ではじいたりしているので長くなっている
- 97行目でゲーム実行

GaSysの流れ2

- 実は試合が終わると各AIに順位に応じたポイントが入る
- 入るポイントは大会に準拠している
- 1世代が終わるとmakePlayersで次の世代のプレイヤーが生成される
- 156行目

WARNING!

ここから先はコードが汚く理論も
めんどくさいため理解が難しい
と思われるヤバイ領域です！

いったん休憩するなどして
やる気を十分補充してから
進んでください！

GaSysPlayerのパラメータについて

- さっき説明を避けたがとうとう説明しなければいけない時が来てしまった...
- 購入時にカードの評価値を決める際に使うパラメータを持っていて、それを**GA**で探索している
- 大きく分けて7つのパラメータが存在して、行列の形で管理されている

GaSysのパラメータ1

myDeckMat

- 自分のデッキがカードに与える影響をパラメータ化したもの
- 評価値は
 - (自分のデッキに入ってるカードの枚数) * (そのカードの評価値) のすべてのカードの種類の合計
 - となっている
- 評価値はあるカードAに対して他全カードの評価値が存在するためパラメータは $N*N$ 個である

GaSysのパラメータ2

othersDeckMat

- 他人のデッキがカードに与える影響をパラメータ化したもの
- 評価値は
 - (他人のデッキに入ってるカードの枚数の合計) * (そのカードの評価値) のすべてのカードの種類の合計
 - となっている
- 評価値はあるカードAに対して他全カードの評価値が存在するためパラメータは $N*N$ 個である
- 他人の持ってるカードが9-0-0でも3-3-3でも全く同じ扱いとなる

GaSysのパラメータ3

supplyMat

- サプライにあるカードが与える影響をパラメータ化したもの
- 評価値は
 - サプライに存在するカードの評価値の合計
 - となっている
- 評価値はあるカードAに対して他全カードの評価値が存在するためパラメータは $N \times N$ 個である
- 現時点では基本サプライに対しての評価値も存在する

GaSysのパラメータ4

deckCntMat

- いまのデッキ枚数を考慮に入れるパラメータ
- 評価値は
 - 現在デッキにあるカードの枚数＊パラメータ
 - となっている
- 評価値はN個である

GaSysのパラメータ5

endTurnMat

- 予測終了ターン数を考慮に入れるパラメータ
- 評価値は
 - 予測終了ターン数 * パラメータ
 - となっている
- 評価値はN個である
- `estimateLeftTurn()` GaSysPlayerの468行目
- 予測アルゴリズムは極めて雑

GaSysのパラメータ 6

victryDiffMat

- トップとの点数差を考慮に入れるためのパラメータ
- 評価値は
 - トップとの点数差 * パラメータ
 - となっている
- 評価値はN個である

GaSysのパラメータ7

positionMat

- 手番を考慮に入れるためのパラメータ
- 評価値は
- 手番 * パラメータ
- となっている
- 評価値はN個である

実際のコードを見ると

- `getBestScore` 382行目
- 再帰による探索を行っている
- 順番に評価値を計算してるので、
- 鍛冶、村と買うのと 村、鍛冶と買うのでは
- 評価値が違う
- ここを修正する案はあるがめんどくさいのでやってない
- あと A I が26金8購入とかすると全探索なので時間がかかりすぎる（実話）

遺伝アルゴリズムは？

- 現在やってることの概要（一般的に紹介されているアルゴリズムとは若干異なる）
 - ①まずエリートの数だけ上から順に親リストに加える
 - ②次に親リストがいっぱいになるまで多少ランダムを加えながら上から順に親リストに加えていく
 - ③親リストから次世代の子供を作成
 - ④エリートの数だけ親リストから子供リストに直接追加

どのように子を作っているか？

- ソース読め (GaSys 156-594)
- このプロジェクトの中で最高峰に汚いプログラム
- 一応作用させるオペレータは366-のswitch文に書いてあるから頑張って
- 各オペレータの確率とか種類とかはノリでやってるのでよく分からない

第 3 章 LogLerning

- DomCore
 - Card, CardData, Supply
 - Player
 - Result
- GaSys
 - GaSysPlayer
- **LogLerning**
 - **LogLerningPlayer**
- LoadPlayer
- Params, GaUtils
- DomGUI
 - DomController
- SmithyParamTuner

もう疲れた

- LogLerningはログから評価値を計算してGAをする
- じゃあGaSysと規格統一してスーパークラス作れば？
- →その通りです、だれかやってくれー
- LogLerningの評価値の計算方法は
- LogStudyPlayer92行目calculateDiffFromLog
- ざっくり説明すると、AIが最もいいと思ったやつとログの手の評価値の差が小さいほど良いとする
- 正解の手以外が最善だとしたらペナルティを与える

第 4 章 DomGUI

- DomCore
 - Card, CardData, Supply
 - Player
 - Result
- GaSys
 - GaSysPlayer
- LogLerning
 - LogLerningPlayer
- LoadPlayer
- Params, GaUtils
- **DomGUI**
 - **DomController**
- SmithyParamTuner

そろそろ帰りたい

- GUIを記述しているところ
- これがないと人間がプレイできない
- JAVAFXを使っていてJAVAFXを知らないと理解が難しいので説明は省略（ちなみにこれのためにJAVAFXを1から勉強した）
- DomCoreを作る時にGUIを渡せばいいタイミングで更新してくれるようになってる

残りは読者の演習課題とする

(正直どうでもいい)

- DomCore
 - Card, CardData, Supply
 - Player
 - Result
- GaSys
 - GaSysPlayer
- LogLerning
 - LogLerningPlayer
- **LoadPlayer**
- **Params, GaUtils**
- DomGUI
 - DomController
- **SmithyParamTuner**

おわり