

遺伝的アルゴリズムによる人工知能を用いた ゲームバランス調整

CEDEC2017 2017/9/1
株式会社スクウェア・エニックス
テクノロジー推進部 眞鍋 和子

謝辞

本研究を行うに当たって、元気株式会社の皆様より、技術的支援や貴重なご意見を賜りました。心より感謝致します。

また、本研究に多大なるご協力をいただきました、グリム
ノーツ開発・運営の皆様に、感謝を申し上げます。

Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

- ゲームの紹介と課題
 - プレイヤーAIの利用手法
 - プレイヤーAIの利用結果と考察
-
- まとめ

自己紹介

▶ 経歴

- ▶ 据置型ゲーム開発
- ▶ ゲームツール開発



▶ 現在

- ▶ スクウェア・エニックス
 - ▶ テクノロジー推進部 AIユニット
 - ▶ AIエンジニア
- ▶ QA, ゲームバランス調整

SQUARE ENIX



Agenda

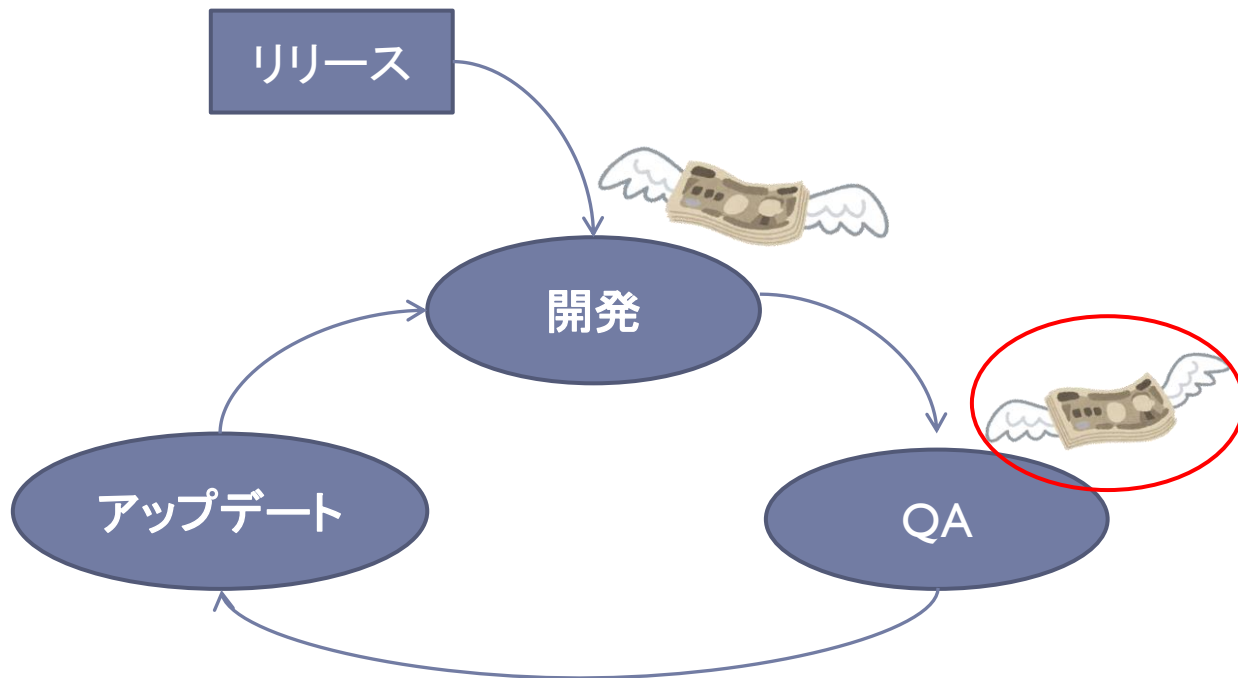
- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

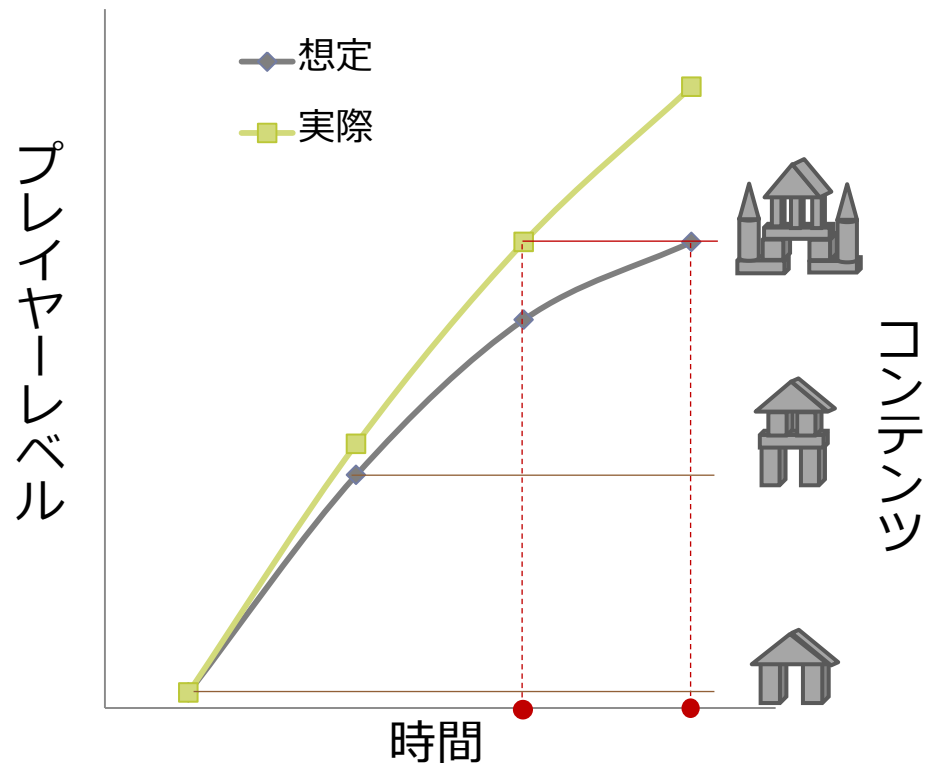
- ゲームの紹介と課題
 - プレイヤーAIの利用手法
 - プレイヤーAIの利用結果と考察
-
- まとめ

背景

▶ 開発イテレーションの短期化



背景



面白さの減少

コンテンツ
消費速度の増加

多くのQAテスターが必要



QAコストの増加



プレイヤーAI導入

Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

- ゲームの紹介と課題
 - プレイヤーAIの利用手法
 - プレイヤーAIの利用結果と考察
-
- まとめ

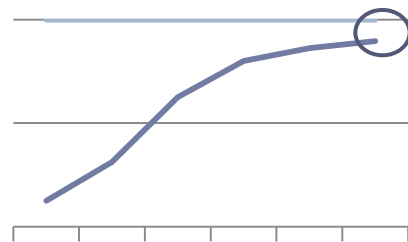
プレイヤーAI

▶ プレイヤーの代わりとなるAI

- ▶ 入力: プレイヤーが知れる情報と同等
- ▶ 出力: プレイヤーが取れる行動と同等

▶ ゲームを自動プレイ

- ▶ コンテンツを攻略させてみる
- ▶ 一定時間経過後のプレイヤーの状況を見る



▶ 自動生成が望ましい

- ▶ 新しいゲーム要素が追加された時、自動で再生成

既存の研究: プレイヤーAI

▶ City Conquest

- ▶ タワーディフェンス
- ▶ 遺伝的アルゴリズム
- ▶ 特に強い施設を探し出す
- ▶ “Making Designers Obsolete? Evolution in Game Design”, AiGameDev.com, 2012, <http://www.intelligenceenginestudios.com/cityconquest.htm>, (2017-7-5 参照)

AiGameDev City Conquest

検索



▶ FINAL FANTASY Record Keeper

- ▶ RPG
- ▶ 遺伝的アルゴリズム, ニューラルネットワーク, 強化学習
- ▶ バトルの難易度評価
- ▶ 友部 博教, 半田 豊和, “AIによるゲームアプリ運用の課題解決へのアプローチ” (CEDEC2016)
- ▶ http://cedil.cesa.or.jp/cedil_sessions/view/1511

CEDiL AI 解決

検索



既存の研究: 自動プレイ

- ▶ 森田 和則, “『ベヨネッタ2』におけるゲーム品質を上げる為の自動化 ～オートプレイと継続的なパフォーマンス計測～” (CEDEC2016)

▶ https://cedil.cesa.or.jp/cedil_sessions/view/1454

CEDiL ベヨネッタ

検索

- ▶ 折田 武己, “ゲーム開発におけるデバッグ作業の自動化 ～OpenCVの「眼」で捉え、Pythonの「脳」が思考し、Appiumの「指」で動かす” (CEDEC2016)

▶ https://cedil.cesa.or.jp/cedil_sessions/view/1575

CEDiL デバッグ作業

検索

- ▶ その他、テスト自動化ツール

Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

- ゲームの紹介と課題
- プレイヤーAIの利用手法
- プレイヤーAIの利用結果と考察
- まとめ

グリムノーツ

- ▶ ジャンル: RPG
- ▶ 料金体系: アイテム課金制 (基本プレイ料金無料)
- ▶ 対応OS: Android™ 4.0以上, iOS8以上(iPhone5以降, iPad®第4世代以降)



グリムノーツ: バトルシステム

▶ リアルタイムアクションバトル（自動モードあり）



勝利条件

敵全滅

敗北条件

リーダーが戦闘不能

※勝利条件は、バトルによって
違うこともあります。

グリムノーツ: 主人公とヒーロー

- ▶ 1主人公につき、2人の「ヒーロー」をセット
 - ▶ セットできるヒーローの職種は、ほぼ固定



グリムノーツ: パーティ編成

▶ 主人公



▶ 主人公数: 8
(スロット数: 4,
内リーダー: 1)

(2017年7月26日現在)

グリムノーツ: パーティ編成

▶ ヒーロー



職種数

- ▶ アタッカー: 102
- ▶ ディフェンダー: 60
- ▶ ヒーラー: 60
- ▶ シューター: 89

(計: 311)

(スロット数: 8)

(2017年7月26日現在)

グリムノーツ: パーティ編成

▶ 武器・アクセサリ



武器種数

- ▶ 片手剣: 68
- ▶ 大剣: 62
- ▶ 箠手: 42
- ▶ 槍盾: 60
- ▶ 槌盾: 57
- ▶ 魔導書: 62
- ▶ 片手杖: 57
- ▶ 弓: 56
- ▶ 両手杖: 73
- ▶ 大砲: 24

アクセサリ種数

- ▶ 合計: 4

(スロット数:

各ヒーローごとに武器1,
アクセサリ1)

(2017年7月26日現在)

グリムノーツ: パーティ編成

▶ スキルコア (武器の強化)



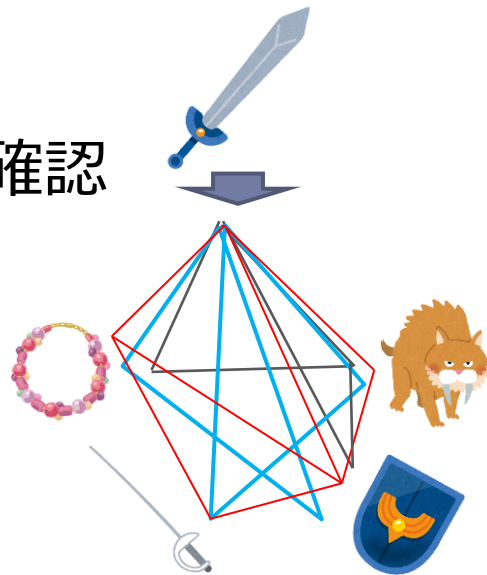
- ▶ スキルコア種類数: 65
(スロット: 各武器ごとに最大5)
(2017年7月26日現在)

グリムノーツの課題

- ▶ たくさんのヒーローや装備
- ▶ 毎月、新しいヒーローや装備が追加
 - ▶ 既存のものとの組み合わせ数が膨大
- ▶ バランスブレイカーがないか、要事前確認
 - ▶ でも調べきれない…



プレイヤーAIにバトルをさせる



Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

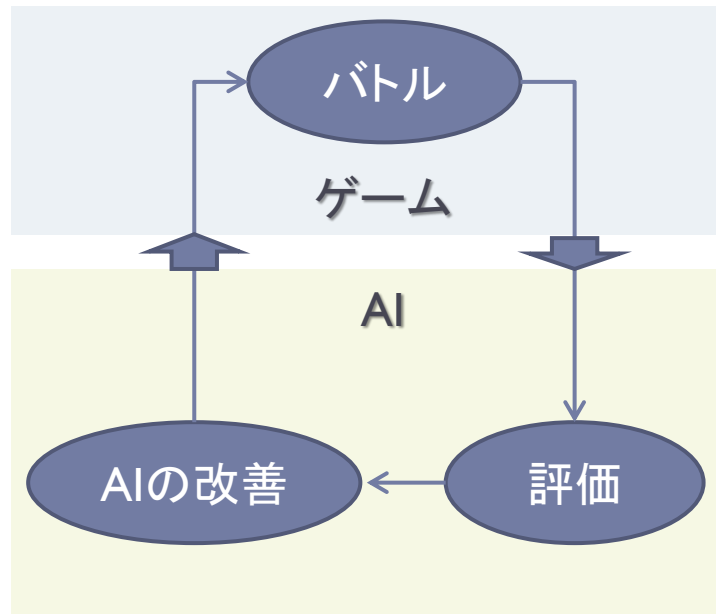
- ゲームの紹介と課題
- プレイヤーAIの利用手法
- プレイヤーAIの利用結果と考察
- まとめ

グリムノーツのプレイヤーAI

▶ AIが目指すゴール:

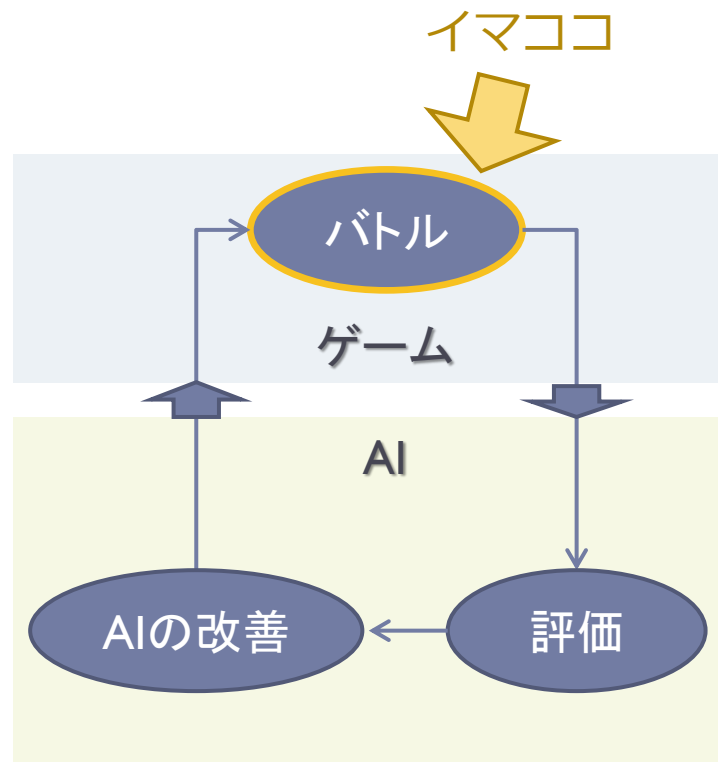
与えられた要素の中から、
指定されたバトルに対し、
最適な組み合わせを探す

- ▶ 主人公
- ▶ リーダー
- ▶ ヒーロー
- ▶ アクセサリー
- ▶ 衣装
- ▶ 武器
- ▶ スキルコア



グリムノーツのプレイヤーAI

- ▶ モデル化されたバトルを使用
 - ▶ ダメージ計算式は、実ゲームと同じものを使用
 - ▶ いくつかの要素を省略
 - 距離・移動の概念なし
 - 被弾率、被ターゲット抽選率、攻撃頻度などのパラメータで置き換え
 - 必殺技なし (回復技のみあり)
 - ヒーロー切り替え機能なし
 - ▶ **短時間で実行可能**
 - ▶ 1バトル約1秒 (シングルスレッド)
 - ▶ 実際のゲームでは1バトル数分

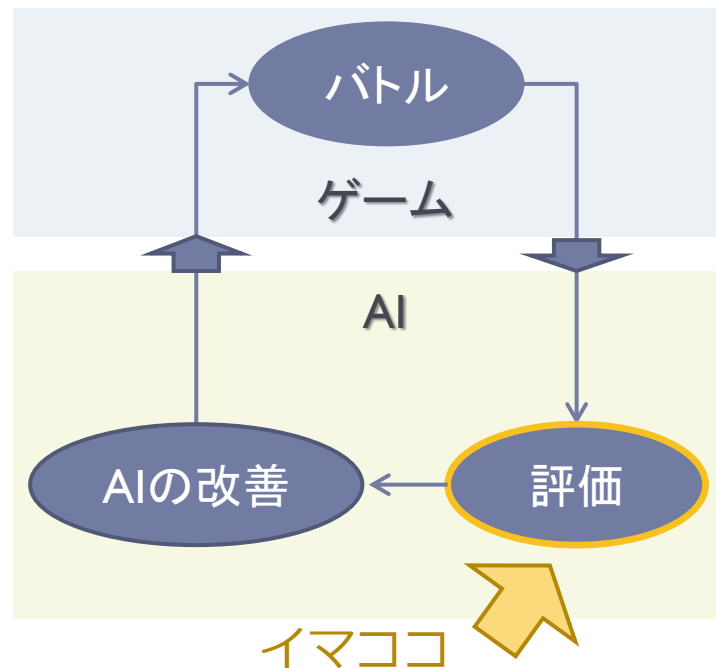


グリムノーツのプレイヤーAI

▶ 評価値

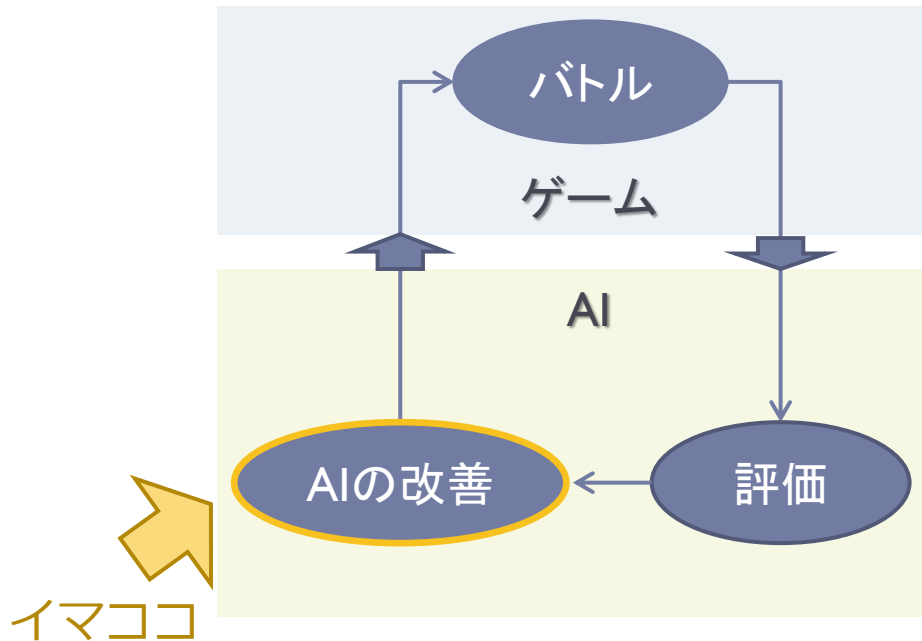
- ▶ バトルの内容が**どれだけ良かったか**を示す指標
- ▶ 勝利時: バトル時間の短さ
- ▶ 敗北時: 与ダメージの多さ

と定義



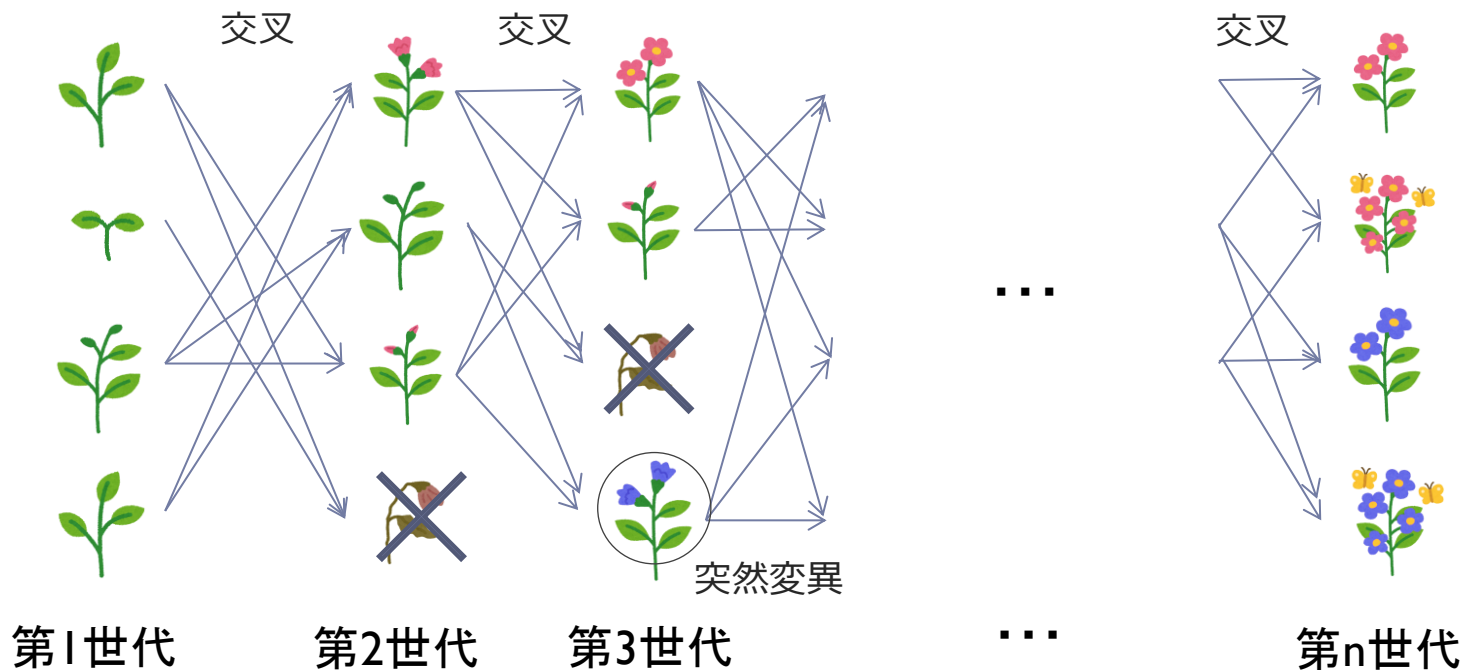
グリムノーツのプレイヤーAI

- ▶ 遺伝的アルゴリズム
 - ▶ 装備の組み合わせの最適化



遺伝的アルゴリズムとは

- ▶ 良い個体同士の遺伝子を掛け合わせて、新しい個体を生み出す手法



なぜ遺伝的アルゴリズムか

- ✓ メリット1: 成長過程の観察が可能
 - ▶ AIの思考の可視化
- ✓ メリット2: 幅広い強さを持った、プレイヤーAIの生成が可能
 - ▶ 実際のゲームをプレイするのは、強いプレイヤーだけではない
- △ デメリット: アルゴリズムのパラメータ調整の難しさ

グリムノーツのプレイヤーAI

▶ 遺伝的アルゴリズムで成長

▶ 装備の組み合わせの最適化

▶ **パーティ編成を遺伝子とする**

▶ **ゲームの詳細知識は不要**

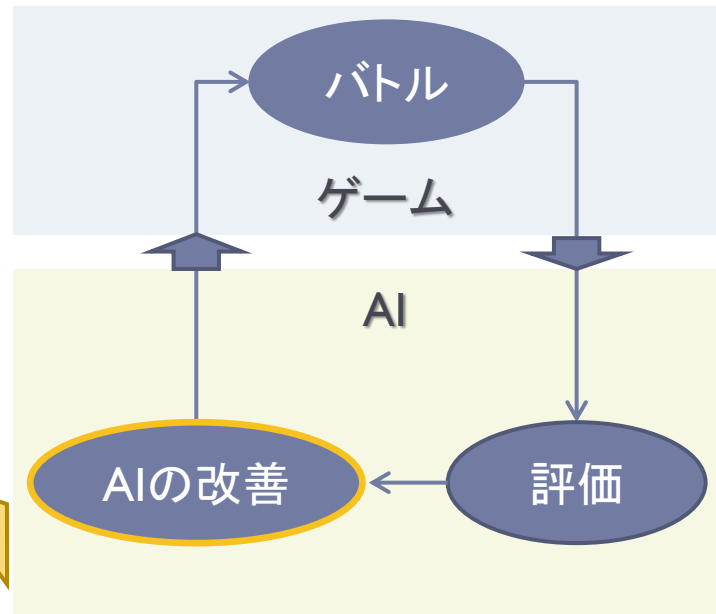
⇒ プレイヤーAIの修正なしで、

▶ 新しい属性の追加

▶ 新しいスキルの追加

▶ 大幅な装備パラメータの変更

などに対応可能



イマココ

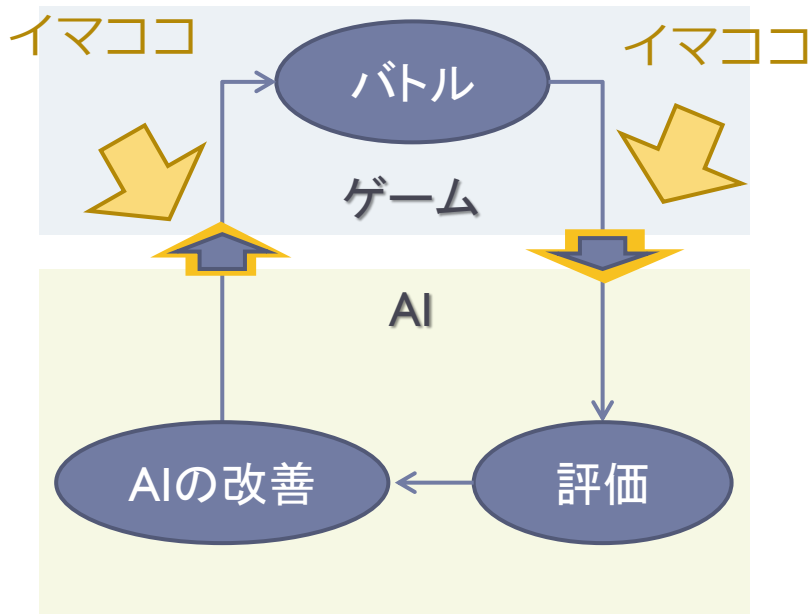
グリムノーツのプレイヤーAI

▶ AIとゲームの間の繋ぎ

▶ APIを設置

- ▶ ゲームに対し、指定した設定でバトル要請
- ▶ バトル結果を数値でもらう

- ▶ 本来は、人間と同等が理想的
= コントローラと画面
 - ▶ 入出力そのものは可能
 - ▶ 特に画面の意味の理解が困難

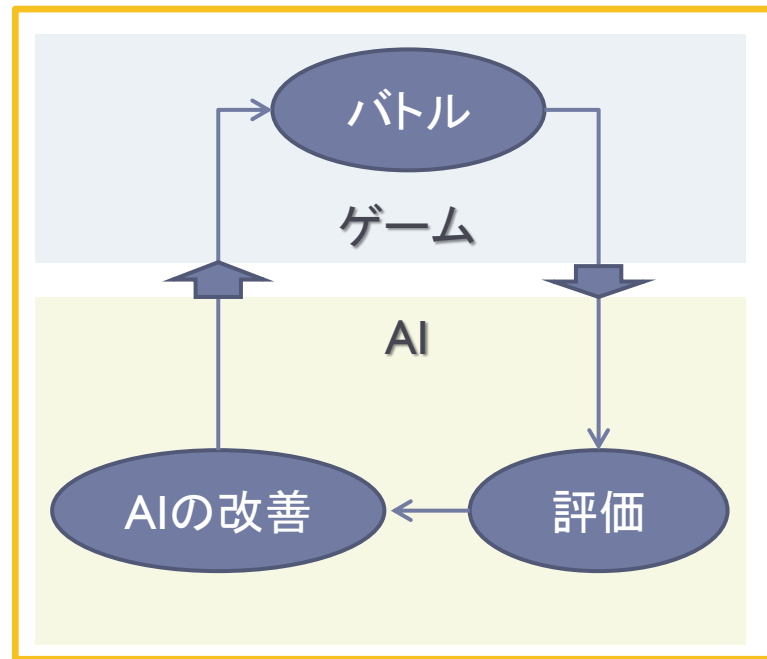


グリムノーツのプレイヤーAI

▶ AIが目指すゴール: (再掲)

与えられた要素の中から、
指定されたバトルに対し、
最適な組み合わせを探す

- ▶ 主人公
- ▶ リーダー
- ▶ ヒーロー
- ▶ アクセサリー
- ▶ 衣装
- ▶ 武器
- ▶ スキルコア



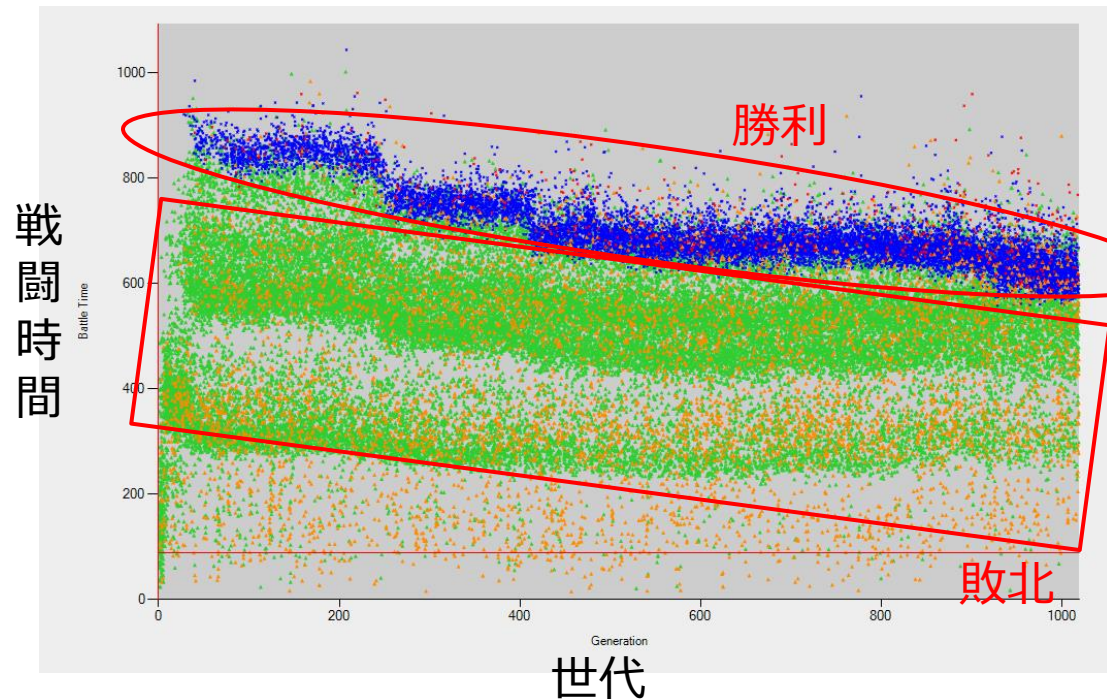
Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

- ゲームの紹介と課題
- プレイヤーAIの利用手法
- プレイヤーAIの利用結果と考察
- まとめ

プレイヤーAIの成長結果例



戦闘に
勝利
敗北

: 青
: 緑

(勝利 かつ 突然変異: 赤)
(敗北 かつ 突然変異: 橙)

プレイヤーAIの成長結果例

▶ 世代を重ねるごとに成長

②長い時間をかけて、
ぎりぎり勝利

③戦闘時間の短縮

①即敗北

戦闘
時間

START!

世代

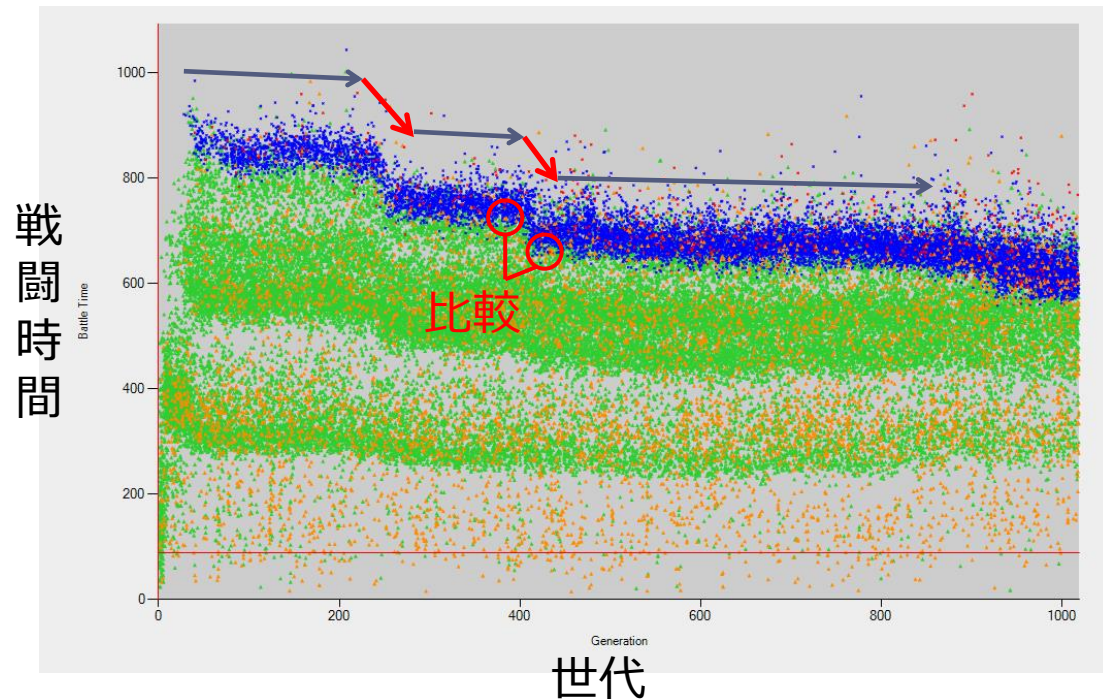
戦闘に
勝利
敗北

: 青
: 緑

(勝利 かつ 突然変異: 赤)
(敗北 かつ 突然変異: 橙)

プレイヤーAIの成長結果例

▶ 突然変異による局所解脱出



戦闘に
勝利
敗北

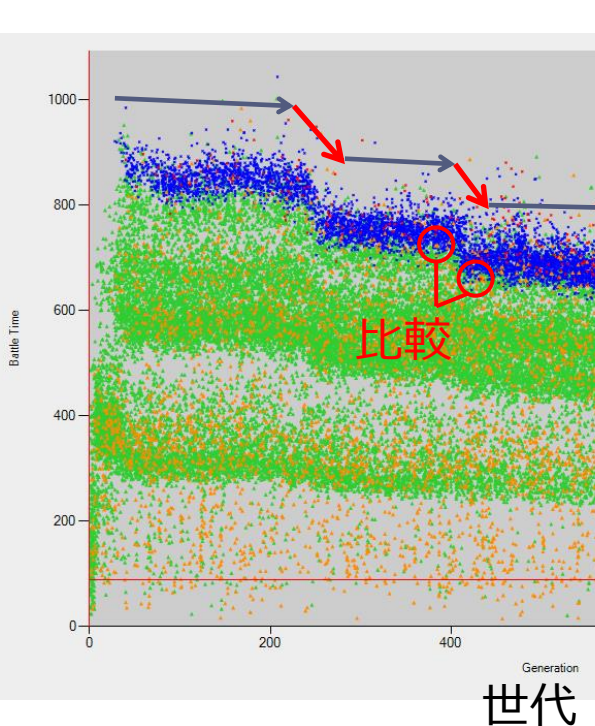
: 青
: 緑

(勝利 かつ 突然変異: 赤)
(敗北 かつ 突然変異: 橙)

プレイヤーAIの成長結果例

▶ 突然変異による局所解脱出

戦闘時間



Material1: 毒のコアⅢ ☆3
Material2: 氷のコアⅢ ☆3
Material3: 光のコアⅠ ☆1
Material4: 耐凍えのコア ☆4
Material5: 闇のコアⅡ ☆2

Hero2: カオス・ジャンヌ (ID.202) ☆5 Lv.50 (アタッカー)
限界突破: 0 試練突破: 3
Dress: レッド
Accessory: 鉄壁のネックレス ☆1 Lv.20
Weapon: 鶴雲寿乃太刀・松 Lv.10 (片手剣)
Material1: 強欲のコアⅠ ☆1
Material2: 剛力のコアⅢ ☆3
Material3: 火のコアⅡ ☆2
Material4: スタンのコアⅡ ☆2
Material5: 雷のコアⅠ ☆1

(Support) エクス
Hero1: カーミラ (ID.3) ☆3 Lv.30 (アタッカー)
限界突破: 0 試練突破: 3
Dress: ---
Accessory: 鉄壁のネックレス ☆1 Lv.20
Weapon: 緋聖夜の太剣・改 Lv.10 (両手剣)
Material1: スタンのコアⅠ ☆1
Material2: 耐闇のコアⅠ ☆1
Material3: 光のコアⅡ ☆2
-----: ---
-----: ---

Hero2: 沙悟浄(オリジン) (ID.209) ☆5 Lv.50 (ディフェンダー)
限界突破: 0 試練突破: 3
Dress: ---
Accessory: 剛力の腕輪 ☆1 Lv.20
Weapon: 紅桜乃突槍・竹 Lv.10 (槍盾)
Material1: 剛力のコアⅢ ☆3
Material2: 氷のコアⅢ ☆3
Material3: 活力のコアⅡ ☆2
Material4: 氷のコアⅠ ☆1
-----: ---

Material1: 毒のコアⅢ ☆3
Material2: 氷のコアⅢ ☆3
Material3: 光のコアⅠ ☆1
Material4: 活力のコアⅡ ☆2
Material5: 闇のコアⅡ ☆2

Hero2: カオス・ジャンヌ (ID.202) ☆5 Lv.50 (アタッカー)
限界突破: 0 試練突破: 3
Dress: レッド
Accessory: 鉄壁のネックレス ☆1 Lv.20
Weapon: 鶴雲寿乃太刀・松 Lv.10 (片手剣)
Material1: 強欲のコアⅠ ☆1
Material2: 剛力のコアⅢ ☆3
Material3: 活力のコアⅡ ☆3
Material4: 剛力のコアⅡ ☆1
Material5: 雷のコアⅠ ☆1

(Support) エクス
Hero1: カーミラ (ID.3) ☆3 Lv.30 (アタッカー)
限界突破: 0 試練突破: 3
Dress: ---
Accessory: 鉄壁のネックレス ☆1 Lv.20
Weapon: 緋聖夜の太剣・改 Lv.10 (両手剣)
Material1: スタンのコアⅠ ☆1
Material2: 耐闇のコアⅠ ☆1
Material3: 耐光のコアⅡ ☆2
-----: ---
-----: ---

Hero2: 桃太郎(オリジン) (ID.150) ☆5 Lv.50 (アタッカー)
限界突破: 0 試練突破: 3
Dress: ヒロイック
Accessory: 剛力の腕輪 ☆1 Lv.20
Weapon: 緋聖夜の太剣・絶 Lv.10 (両手剣)
Material1: 剛力のコアⅢ ☆3
Material2: 氷のコアⅢ ☆3
Material3: 活力のコアⅡ ☆2
Material4: 剛力のコアⅢ ☆3
Material5: 強欲のコアⅠ ☆1

装備の使用頻度の統計

- ▶ 価値の高い主人公・ヒーロー・装備の発見
- ▶ コストとの比較



GrimmsAnalyzer / Frequency

Untitled(6).gadat Untitled(7).gadat Untitled(8).gadat Total

Character Hero Accessory Weapon Materia

	Name	Frequency	Count
	カオス・ドン・キホーテ	0.38	65518
	ルイス・キャロル(オリジン)	0.29	50154
	ジル・ドレ(寄託)	0.20	34880
	沙悟浄(オリジン)	0.17	29808
	武蔵坊弁慶	0.15	26517
	三月ウサギ(鎧の国)(オリジン)	0.14	24362
	カーミラ(天空)	0.13	21698
	ルートヴィッヒ・グリム(オリジン)	0.12	21111
	カオス・赤ずきん	0.12	20264
	ジョン・シルバー	0.11	19847
	猪八戒	0.11	19815
	山賊の娘チアナ	0.10	18129
	ルイス・キャロル	0.10	16565
	ニッケ・ボトム(オリジン)	0.09	15835
	ニッケ・ボトム	0.09	15801
	アマデウス・モーツァルト(オリジン)	0.09	15487
	三蔵法師	0.09	15470
	アラン・ゲルヒニ	0.09	14997
	アリス(鎧の国)(オリジン)	0.09	14928
	シンデレラ(サマー)	0.08	14677
	カリーネ	0.08	14588
	オデッサ・クレエール	0.08	14404

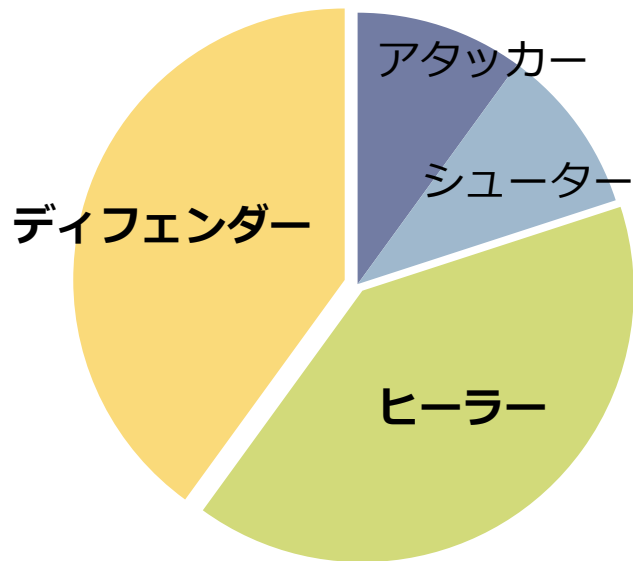
※サンプルデータです。

プレイヤーAIの成長の傾向

▶ 高難易度のバトル

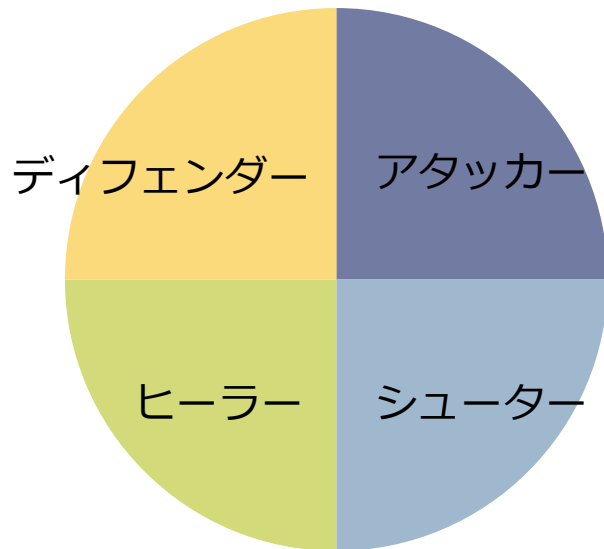
成長途中（敗北が多い世代）

ディフェンダーとヒーラーをメインに配置



成熟後（高確率で勝利できる世代）

全ての職種をバランス良く配置

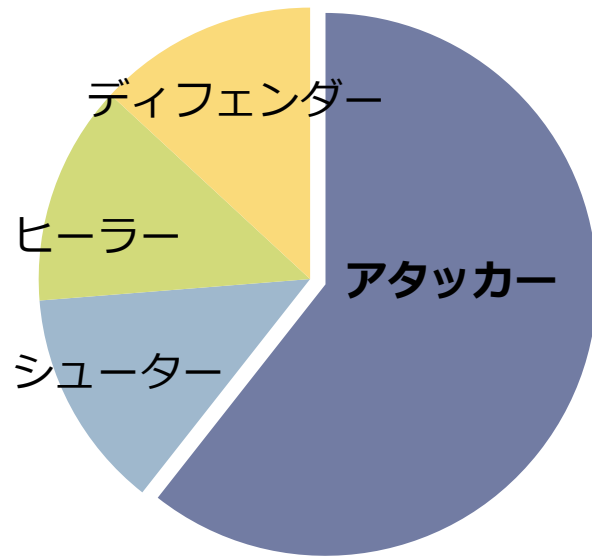


※実際の厳密な分析データの数値ではありません

プレイヤーAIの成長の傾向

▶ 低難易度のバトル

- ▶ アタッカーを多く配置
 - ▶ 戦闘を短時間化
- ▶ ディフェンダー・ヒーラーの中でも、攻撃寄りのヒーローを配置
- ▶ シューターも攻撃職だが、ボス戦向き



※実際の厳密な分析データの数値ではありません

プレイヤーAIの成長

「アタッカーを多く配置」？

- ▶ モデル化されたバトルでは、ヒーローの切り替え機能がない
- ▶ 最初にバトルに参加するのは、必ず上段のヒーロー



プレイヤーAIの成長

- ▶ 強制ヒーロー切り替えの方法を習得
 - ▶ 1人目に**打たれ弱い**ヒーローを設定
 - ▶ → 素早く戦闘不能に
 - ▶ → 2人目の攻撃職に切り替え



課題

- ▶ モデル化されたシミュレーションでは、情報が一部欠損
 - ▶ XXXのこの結果は、シミュレーションだからか？実際のバトルでもそうなのか？
- ▶ 実プレイの高速化（早回し）
 - ▶ クラウド利用
 - ▶ 高速シミュレーションモード
 - ▶ 描画や物理のみ切り離す等の機構
- ▶ データの分析
 - ▶ 客観的な視点によるゲームバランスの数値化
 - ▶ 個体同士の“距離”などを使ったクラスタリング、価値の算出など

Agenda

- ゲーム開発における課題
- プレイヤーAIとは

グリムノーツ:

- ゲームの紹介と課題
 - プレイヤーAIの利用手法
 - プレイヤーAIの利用結果と考察
-
- まとめ

まとめ

- ▶ プレイヤーAIによるゲームバランス調整
- ▶ グリムノーツにて**プレイヤーAIを獲得**
- ▶ **ゲームの詳細知識なし**で、ゲームシステムに対応
- ▶ 価値の高い要素を、**自動**で発見
- ▶ QAコスト削減への期待
- ▶ 実プレイでのAIの成長が次のゴール
- ▶ データ分析の改善

ご清聴ありがとうございました。

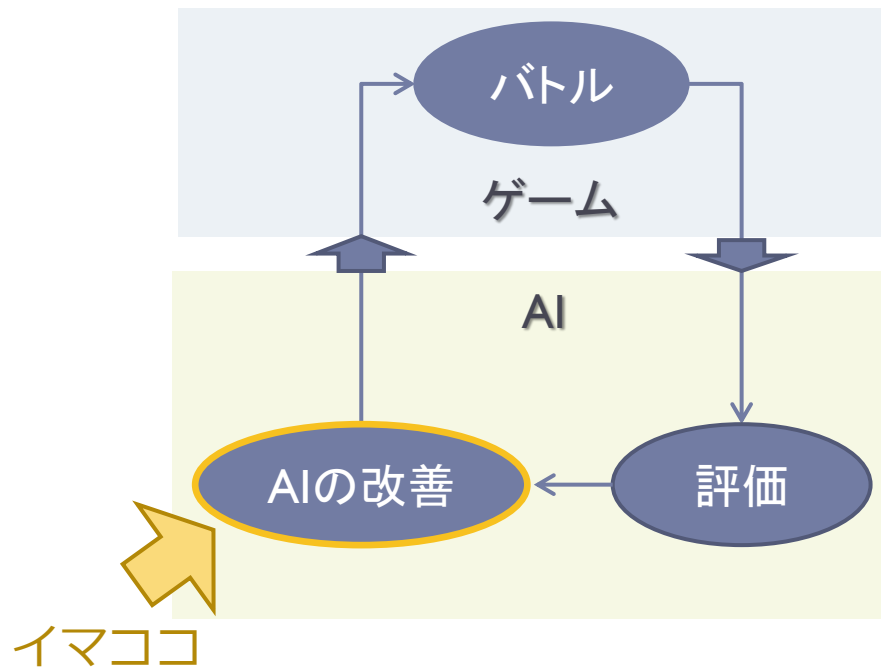


付録1

グリムノーツプレイヤーAIの評価値

事例：グリムノーツのプレイヤーAI

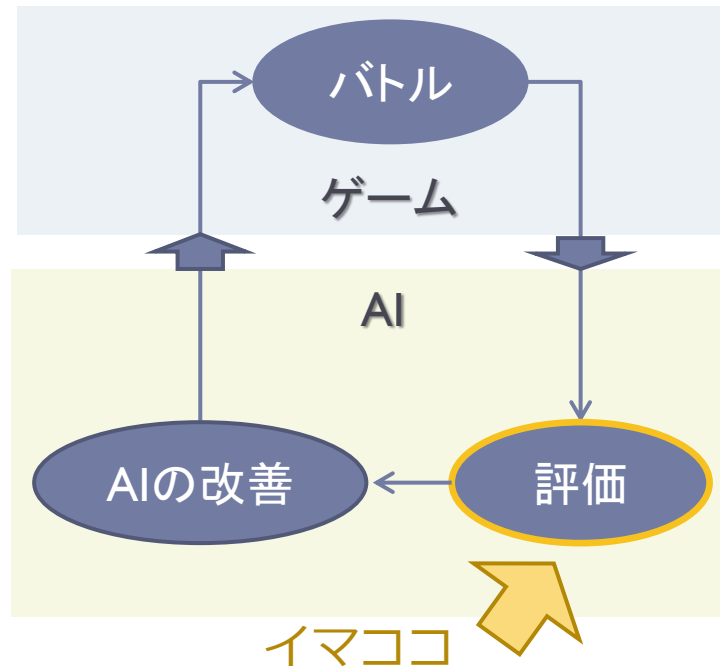
- ▶ 遺伝的アルゴリズム
- ▶ 次世代に残す個体選択は、**評価値の大小**に依存



事例：グリムノーツのプレイヤーAI

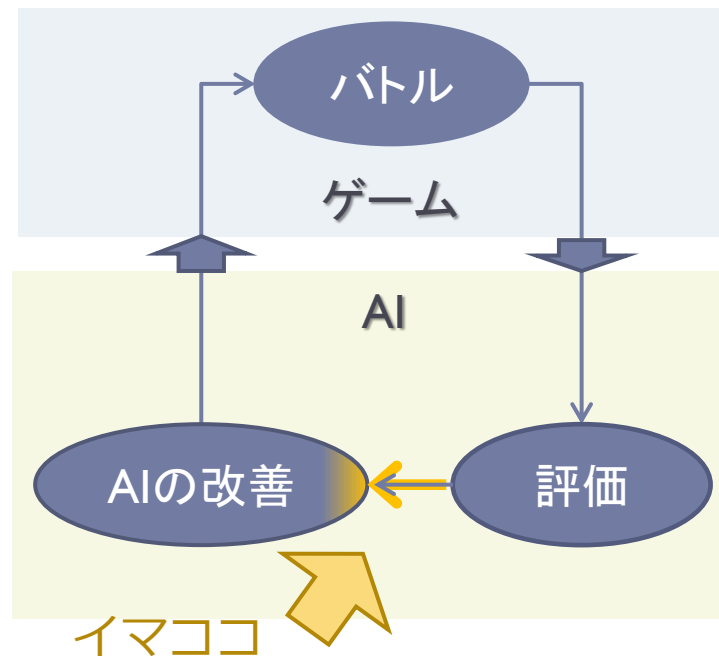
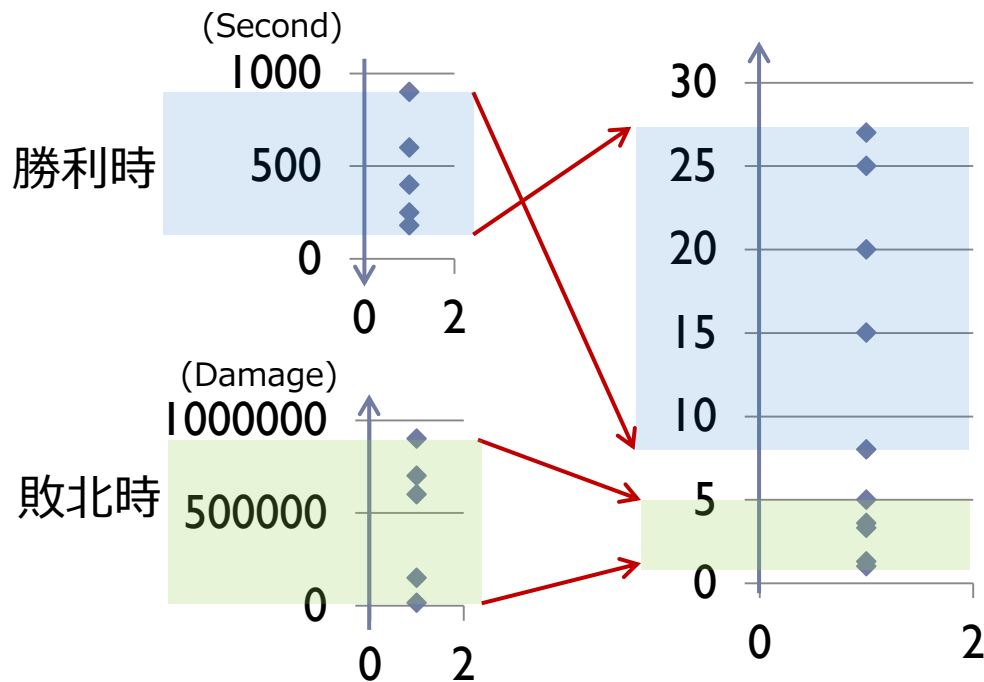
- ▶ 遺伝的アルゴリズム
- ▶ 次世代に残す個体選択は、**評価値の大小**に依存
- ▶ ところが、評価値の単位が違う
 - ▶ 勝利時：時間(秒)
 - ▶ 敗北時：与ダメージ

⇒ 比較ができない



事例：グリムノーツのプレイヤーAI

- ▶ 勝利時・敗北時の評価値を、指定した範囲にマッピング
- ▶ 値の比較が可能に



事例：グリムノーツのプレイヤーAI

▶ 採用しなかった評価式

▶ 残りHP/最大HP

→ City Conquestでの評価式

→ グリムノーツでは、敵の最大HPは、プレイヤーは知ることができない（プレイヤーAIのポリシー違反）

▶ 単位時間あたりのダメージ量（DPS）

→ 武器の強さを測る、一般的な指標

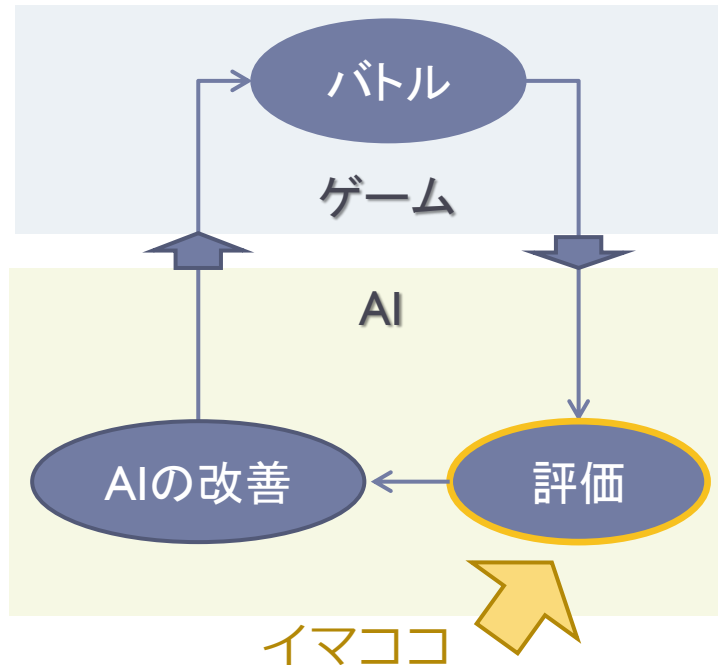
→ 華々しく攻撃し、すみやかに全滅するのが最適になる

▶ 与ダメージ/被ダメージ

→ DPSと同様。被ダメージを減らそうと、リーダーに弱いヒーローをセットし、早々に全滅する*選択が入る

▶ 敗北時：バトル時間の長さ

→ ガチガチに防御を固めて攻撃せず、ジワジワと全滅



*戦闘の敗北条件：リーダーのヒーローが2人とも戦闘不能

付録2

グリムノーツプレイヤーAIの世代生成

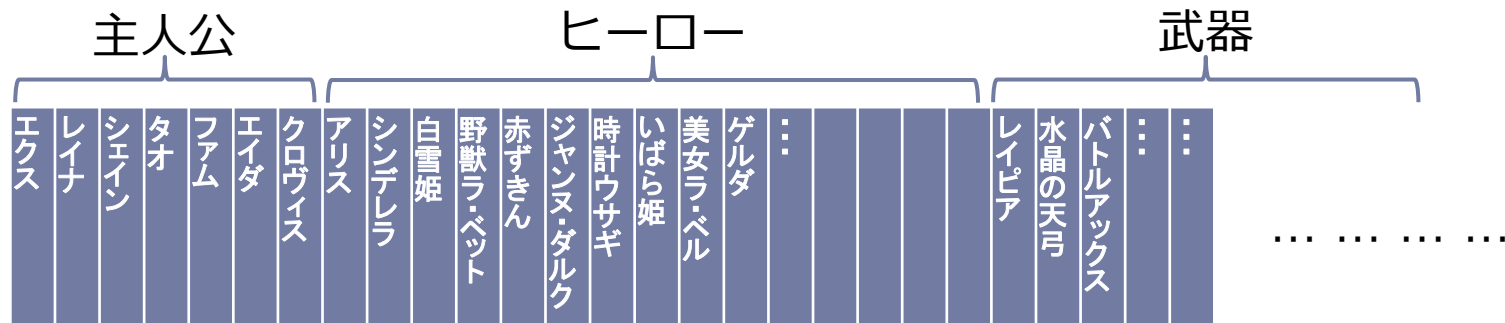
グリムノーツプレイヤーAIの世代生成

- ▶ エリート保存
 - ▶ もっとも評価値の高かった1個体を、次世代にそのままコピー
- ▶ 次世代に残す個体の選択
 - ▶ 評価値を基準にしたルーレット選択
(評価値マッピングについて、付録1を参照)
- ▶ 交叉
 - ▶ 多点交叉
- ▶ 突然変異
 - ▶ 遺伝子の一部をランダムに置き換え
- ▶ 大変異
 - ▶ しばらくの世代数、改善が見られなかった場合の対策
 - ▶ 1世代だけ、突然変異発生確率、突然変異でランダムに置き換えする染色体の個数を、大幅に上げる

付録3

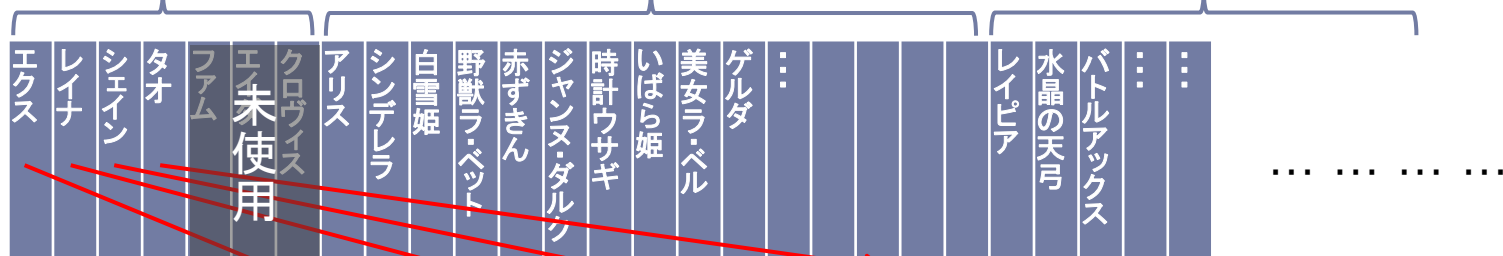
グリムノーツプレイヤーAIの遺伝子

▶ 主人公・ヒーロー・装備の、候補の並び



▶ 利用できる主人公・ヒーロー・装備を探して編成

武器



遺伝子からパーティ編成への変換

- ▶ 利用できる主人公・ヒーロー・装備を探して編成



グリムノーツプレイヤーAIの遺伝子

- ▶ 主人公・ヒーロー・装備の、候補の並びが遺伝子
 - ▶ 候補を並べた遺伝子から、装備できるものを探して使う
 - ▶ 複数所持できるものは、全種類数の数倍程度、候補をランダムに作成する
 - ▶ 生成されない武器が存在する可能性あり
 - ▶ 突然変異での、遺伝子ランダム置き換え個数を、多めにとることで対応
 - ▶ 1つしか所持できないもの（主人公・ヒーローなど）は、全種類を1つずつ並べる
 - ▶ 使われない染色体もある（生物的には自然）
 - ✓ 汎用性の高さがメリット
 - ▶ 他タイトルにも適用しやすい
 - △ 交叉や突然変異との相性により、進化の公平性が失われるリスク
- ▶ 依然として、改善・他の遺伝子構造との比較検証が必要

- ▶ ベヨネッタは株式会社セガゲームスの商標または登録商標です。
- ▶ Androidはグーグル インコーポレイテッドの商標または登録商標です。
- ▶ iPadはアップル インコーポレイテッドの商標または登録商標です。
- ▶ その他掲載されている会社名、商品名は、各社の商標または登録商標です。