

クラウドソーシングを用いたゲームレベルデザインの効率化

Improvement of Game Level Design with Crowdsourcing

三木 康暉 佐藤 晴彦 小山 聡 栗原 正仁
Kohki MIKI Haruhiko SATO Satoshi OYAMA Masahito KURIHARA

北海道大学
Hokkaido University

Recently, development of computer games requires more and more human resources than before due to the increasing complexity in tuning game balances. In this paper, we propose a method for improving the efficiency of game development, especially for game level design, by using human resources available from crowdsourcing.

1. はじめに

近年、ゲーム開発は複雑さを増している。特にゲームのバランス調整や不具合検出と言った調整作業は多大な労力を要する。ゲーム中に利用されているパラメータ数が増大し、最適な組み合わせを発見する作業が容易ではないためである。ゲームパラメータの調整を行い、プレイヤーにとって適切な難易度を提供するための作業はレベルデザインと呼ばれている。

また、一般的に遊ばれるゲームの質の変容も著しい。近年は、MMO^{*1}と呼ばれる大規模オンラインゲームや、ソーシャルゲームへの移行が進んでいる。これらのゲームでは、リリース後もユーザーのプレイ状況を受け、継続的にサービスを改善し続けていくことが求められている。そのため、ゲームパラメータの調整を、実際のプレイヤーログを集計して行う手法の重要度が増している。

本研究では、ゲーム開発者にとって有用な次の2点を満たす手法を考案することを目的とする。

(1) 作成した「レベル」の難易度推定や改善点の検出、修正をある程度自動化して行いたい。(2) テストプレイヤーのプレイログを簡単に収集して視覚化したい。

また、最終的には、ゲームレベル設計の改善、レベル修正を効率的に行えるシステムの開発を目指す。

今回は、前段階として、従来、多数のテストプレイヤーを動員する必要があったプレイログの収集をクラウドソーシングによって行い、解析、可視化を Web 経由で簡便に行えるシステムの構築について論じる。

2. 関連研究

ゲームのレベルデザイン改善を扱った研究はいくつか存在する。特にプラットフォームゲームと呼ばれる種類のゲームのレベル設計改善をプレイヤーの経験やログを用いて行った研究 [1][2][3] や遺伝アルゴリズムによるアプローチで行った研究 [4] が存在する。また、ゲーム AI のためのコンペティション [5] においても、ゲームのレベルデザインが扱われている。ただし、これらの研究ではクラウドソーシングを用いたゲームレベルデザインは行われていない。

3. 用語の定義

3.1 レベルとレベルデザイン

本稿で用いる用語にはゲーム開発者以外には馴染みの浅い言葉がある。そのため、用語について以下に定義を与える。

レベル ゲームを構成する小単位を示す。本稿での「レベル」はプレイヤー熟練度の意味は持たない。

レベルデザイン 「レベル」を具体的に設計、パラメータの調整を行うこと。

3.2 レベルデザインの改善

本研究では、以下のような性質を満たすレベルデザインを良質であると仮定した。

序列の一貫性 レベルの難易度が、ゲームのプレイ進行度に応じて徐々に上昇していくこと。

また、新たなゲーム中の要素が順序立てて出現し、プレイヤーへの手解きを兼ねていること。

クリア可能性 全てのレベルがクリア可能であること。いわゆる「ハマリ」が起こらないこと。

難易度の適切性 極端にクリア率の低い難易度を持つレベルが含まれないこと。

4. 提案手法

本研究では、ゲーム開発者によるレベルデザイン改善の効率化を図るため、クラウドソーシングを用いて、プレイヤーのプレイログ収集を大規模に行うことができるシステムを構築した。

今回、設計したシステムは以下の機能を有する。

(1) 開発者によるレベルの登録 (2) プレイヤーへのレベルデータの配信 (3) プレイヤーアンケート収集 (4) レベル毎、プレイヤー毎の各種統計処理 (5) プレイログの再生 (6) プレイログの収集

プレイヤーは Web ブラウザで特定のページにアクセスするだけでゲームをプレイでき、システムの動作を意識することなくログの集計を行うことができる。

図 1 は本システムで開発者向けに提供されているレベル管理画面の様子を示している。

本システムは Web アプリケーションとしてサーバー上に構築されており、以下のモデルが実装されている。

連絡先: 〒 060-0814 札幌市北区北 14 条西 9 丁目北海道
大学大学院情報科学研究科複合情報学専攻 三木康暉
miki@complex.ist.hokudai.ac.jp

*1 Massively Multiplayer Online-Game

変更するレベルを選択

Level Title	Play count	Gameover rate	Clear rate	Defection rate
CAVE	20	100.0%	0.0%	0.0%
CAVE	20	89.232941176%	11.7647058824%	0.0%
CAVE	3	50.0%	50.0%	0.0%
LAKE	9	85.7142857143%	0.0%	14.2857142857%
LAKE	8	100.0%	0.0%	0.0%
LAKE	3	100.0%	0.0%	0.0%
LAKE	27	95.652173913%	4.34782608696%	0.0%
LAKE	7	75.0%	75.0%	0.0%

図 1: レベル管理画面

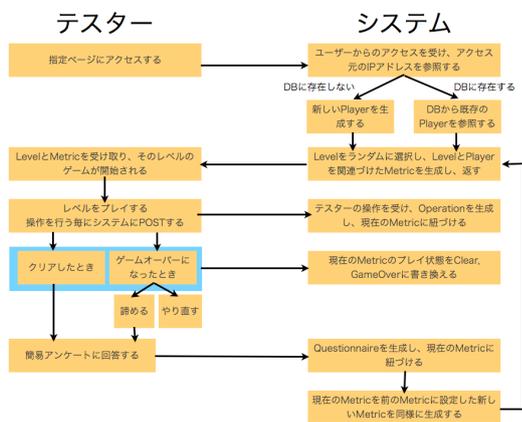


図 2: システムのフロー

Level ゲームにおけるレベル 1 つ 1 つを表すモデルである。

通常はシステム管理者がシステムに登録する。

Level は通常、マップデータなどがテキスト形式で保存されているが、データ構造はゲームによって異なる。

Player プレイヤーを一意に特定するためのモデルである。

今回は、Player は IP アドレスのみを保持し、それが同一である場合に、同じプレイヤーと判定する。

同一の Player によるプレイログを辿ることにより、プレイヤーのプレイ回数による習熟度の変化を観測することができる。

Metric プレイヤーのプレイログを表すモデルで、1 回のゲームプレイにつき一つが生成される。

Metric はそれぞれ次の 6 つの情報を保持している (1) プレイされた Level (2) プレイした Player (3) Operations のリスト (4) 一つ前の Metric への参照 (5) Metric の最終状態 (6) 回答された Questionnaire

これらの情報を格納することで、プレイヤーの操作を再現できる。

Operation プレイヤーの行動 1 つ 1 つを示したモデルである。例えばクリック位置や、ゲーム中で選択したコマンドなど、プレイヤーの行動が記録されている。

Questionnaire プレイヤーのアンケート結果を示したモデルである。

システム管理者は任意のアンケートを発行し、各プレイの終了後、プレイヤーに回答させることができる。回答は対応する Metric に関連づけられる。

本システムは、これらのモデルを利用し、図 2 に示したフローで動作する。

5. 対象とするゲーム

システムの有用性を検証するために、今回は以下のようなサンプルゲームを考案し実装した。



図 3: ゲーム画面

5.1 基本ルール

今回、対象としたゲームは、ターン型完全情報のパズルゲームである。このゲームは、10 マス x10 マスの盤面を操作し、キャラクターをゴール状態に辿り着かせることを目標とする。プレイヤーは 1 人でプレイする。

プレイヤーは任意の 4 マスを選択し、左右いずれかに回転をする操作を繰り返す。以下のいずれかの状態になったとき、1 ゲームが終了する。

- (1) プレイヤーキャラクターがゴールマスに到達したとき
- (2) プレイヤーキャラクターがマップ外にはみ出したとき
- (3) 死亡判定のあるギミック (穴や針など) に触れたとき

5.2 対象ゲームの特徴

サンプルゲームは以下のような特徴を持つため、提案手法の評価に有用であると考えている。

- (1) 操作が容易でプレイヤー間の実力差が出にくい
- (2) 確定完全情報ゲームであり、解法の数がある程度推測しやすい
- (3) リプレイ性が高く、同じプレイヤーのプレイ回数による操作の変化を観測しやすい
- (4) 問題空間が小さく、解の探索が容易である
- (5) レベル作成が困難である

6. 今後の研究方針

今回実装したシステムを用いて実際にプレイログを収集し、レベル設計に利用する実験を行う予定である。現段階で考えられる研究課題を以下に述べる。

SPAM の判定とフィルタリング クラウドソーシングを用いてプレイログを収集する場合、多くのゲームを適当にプレイして多くの金銭を得ようという SPAM プレイヤーが存在する可能性がある。真面目に行われたプレイと SPAM プレイとを判別する機構が必要となる。

プレイヤー習熟度を考慮したレベルの難易度判定 クリアにかかった時間を計測することや、アンケートでレベルの難しさを尋ねることで、レベルの難易度を推定することが可能となる。ただし、プレイヤーによって習熟度が異なるため、単にクリア時間や主観的な難易度の平均を取るだけでは、正確な難易度の推定ができない可能性がある。

例えば、習熟度の高いプレイヤーや実際には難しいレベルでも短時間でクリアし、易しいと回答する可能性がある。そこで、レベルの難易度の推定においてワーカーの習熟度を考慮した手法の研究が必要となる。

汎用的なシステムの設計 今回提案したシステムは、対象としたゲームに特有な部分があり、多くのゲーム作品に汎用的に利用可能なシステムにすることも必要である。

具体的には、データの保存、取得を行う API を提供し、ゲーム開発者がプレイログを収集したいカ所に特定のコード片を埋め込むことで、汎用的にプレイログの収集を可能にする方式を検討する。

レベルの改善点検出とその修正 本研究の最終的な目的はレベルの改善点の検出とその修正である。収集したログからの異常値検出などの手法を用いて、レベルの問題点を検出し、改善に繋げていく手法の研究を行う。

7. おわりに

本稿では、ゲームのプレイログをクラウドソーシングを用いて大規模に収集するシステムについて述べた。提案システムにより、レベルの難易度を推定するためのデータの収集やプレイログの可視化が可能となり、レベルの改善点の検出が容易になると考えている。今後実際にプレイログを収集し、その有効性を検証していきたい。

参考文献

- [1] Anders Drachen, Alessandro Canossa: Towards Gameplay Analysis via Gameplay Metrics, *Proceedings of the 13th International MindTrek Conference (MindTrek '09)*, Pages 202-209, 2009.
- [2] Chris Pedersen, Julian Togelius, Georgios Yannakakis: Optimization of Platform Game Level for Player Experience, *Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2009.
- [3] Heather Desurvire, Charlotte Wiberg: Game Usability Heuristics (PLAY) for Evaluating and Designing Better Games: The Next Iteration, *Proceedings of the 3d International Conference on Online Communities and Social Computing (OCSC): Held as Part of HCI International 2009*, Pages 557-566, 2009.
- [4] 五木宏, 松原仁: 遺伝アルゴリズムの視覚化を用いたゲームのレベルデザイン効率化技法の開発, *IPSJ SIG technical reports 2007(18)*, Pages 1-7, 2007.
- [5] Mario AI Championship <http://www.marioai.org/>