

Angry Birds におけるドミノステージの難易度調整

Adjustment of domino stage difficulty in Angry Birds

石井 稜大[†] 蔣 雨軒[‡] ターウォンマツ ラック[†] 原田 智広[†]
 Ishii Ryota Yuxuan Jiang Thawonmas Ruck Harada Tomohiro

1. はじめに

近年、ゲーム開発の大規模化やそれに伴う開発コストの増加などが問題となっている。これらの問題を解決する手法の一つとして、ゲームのコンテンツを自動生成する **Procedural Content Generation** [1] (以下、**PCG**) が提案されている。本研究では、著名なアクションパズルゲームである **Angry Birds** の **PCG** として、ドミノ倒しを参考にステージの自動生成を行うものを提案する。また文献[2] では、生成されるステージの難易度がプレイヤーのスキルと合うステージであれば、プレイヤーはゲームを楽しく感じるということが述べられている。しかし、生成されるドミノステージは簡単にクリアできてしまうという問題点が挙げられる。よって、生成されるドミノステージを楽しく感じてもらえるように、ステージの難易度を調整する必要がある。

そこで本研究では、**PCG** に対してステージの難易度を向上させるための改良を行う。難易度を向上させるように、オブジェクトを配置する手法を用いてステージを生成した後、**AI** を用いて実験を行い、提案手法の有効性を検証する。

2. 関連研究

2.1 Angry Birds

Angry Birds とは、2009年に Rovio Entertainment 社から販売されたモバイルゲームである[3]。 **Angry Birds** は、鳥をスリングショットで飛ばし、フィールド上の緑色の豚をすべて倒すことを目的とするアクションパズルゲームである。 **Angry Birds** のステージは、図1のようにスリングショット、鳥、豚、ブロックで構成されている。本研究では、[4]にて公開されている **Unity** で **Angry Birds** をクローンした **Science-Birds** を用いる。

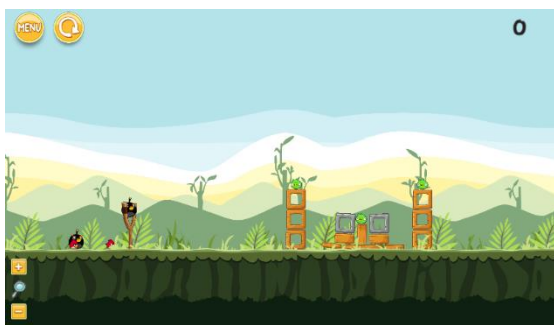


図1. Angry Birds のステージ

2.2 文字、数字、記号列による Angry Birds の面白いステージの自動生成

Angry Birds における **PCG** の関連研究として、ドミノステージを生成する **PCG** 以外にも文字や数字を用いたステージの自動生成をする **PCG** がある[5]。この研究では、プレイヤーが面白いと感じるステージを生成することを目的と

している。ステージ上に英文や数式を模した構造を生成するために、ブロックの配置を **Pattern-struct** や **Preset-model** として定義し、**Pattern-struct** では、文字や数字の構造パターンを定義している。**Preset-model** は文字の構造を多様化するために、**Pattern-struct** で作ることが出来ない形を定義している。これらで定義された文字を用いて、英文や数式を配置したステージを生成する。この研究では、ステージの面白さについて考慮されているが、難易度についての考慮はされていない。

3. 提案手法

2.1 ドミノステージを生成する PCG

ドミノステージとは、破壊可能なオブジェクトであるドミノと、それを支える土台の **Platform** で構成されたステージと定義する。この定義に基づいてオブジェクトの配置を行う。生成されたドミノステージの例を図2に示す。



図2. ドミノステージの一例

2.2 手法

本研究では、ドミノステージの難易度を向上させるためのオブジェクトの配置を提案し、その配置を用いてステージの生成を行う。

図2のようなステージにおいてドミノを倒す場合、スリングショットから最も近いドミノを狙うことで連続して倒すことができるが、ステージクリアの難易度は低くなってしまいう問題点がある。そこで、最も近いドミノの前に **Platform** を配置することで、鳥を正確にドミノに当てることができるように鳥を飛ばす角度を調整する必要があり、ドミノを多く倒せるショットが打ちにくくなるという仮説を立てる。本研究では、ドミノステージに対して図3のようにドミノの先頭オブジェクトの前に **Platform** を配置することによって難易度を向上する手法を提案する。**Platform** は重力の影響を受けず、破壊不可能なオブジェクトである。この **Platform** を、先頭のオブジェクトから図3のように赤枠で決められた範囲内に1つ生成する。赤枠は、先頭の土台の中心座標から一定の範囲で定められている。ドミノが



図3. ブロックの配置

[†] 立命館大学情報理工学部, Ritsumeikan University of Information Science and Technology

[‡] 立命館大学情報理工学研究科, Ritsumeikan University Graduate of Information Science and Technology

2段で構成されている場合も同様に、2段目の先頭オブジェクトの前に Platform を生成する。この手法をドミノステージに適用しステージを生成する。この手法を提案手法 A とする。

しかし提案手法 A の場合、Platform の上部に破壊可能なオブジェクトが配置されないことからドミノステージの定義にあてはまらないステージが生成されてしまう。この問題を解決するために、提案手法でステージに配置される Platform の上に破壊可能なオブジェクトを置き、ドミノステージの定義を満たすように提案手法の改変を行う。改変を適用して生成されたステージの例を図 4 に示す。この手法を提案手法 B とする。

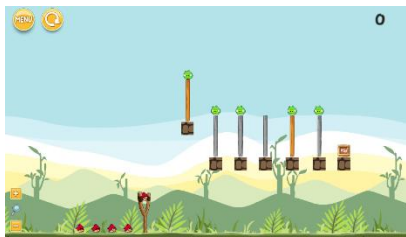


図 4. 改変後のドミノステージ

4. 評価実験

提案手法によって生成されたステージの難易度が向上したかどうかを検証するために AI を用いて実験を行う。

4.1 実験設定

本実験では、手法を適用する前のステージと手法を適用後のステージをそれぞれ 10 種類生成し、AI にそれぞれ 10 回ずつプレイさせる。AI は、[4]にて公開されているステージ検証用の AI である Naive Agent を用いる。鳥の数はすべてのステージで 5 に設定されている。また、提案手法 A 及び B で生成されるステージはすべて既存手法で生成されたステージに基づいて生成される。

4.2 評価基準

本実験では、提案手法によって難易度が向上したかを検証するため、ステージごとのクリア率を元に比較を行う。クリア率 P を求める式を(1)に示す。

$$P = \frac{C}{T} \times 100 \quad (1)$$

ここで、 C はステージのクリア回数、 T はステージの試行回数である。

4.3 実験結果

既存手法で生成されたステージのクリア率、提案手法 A を適用し生成されたステージのクリア率、提案手法 B を適用し生成されたステージのクリア率を表 1 に示す。表 1 より、提案手法 A は Stage5 以外のステージで既存手法よりもクリア率が下がっていることが分かる。既存手法と提案手法で生成されたステージのクリア率の平均は、Wilcoxon の符号付順位和検定より有意水準 1% で差があることが確認された。このことから、提案手法によってステージの難易度が向上したことが分かる。また、提案手法 B と提案手法 A で生成されたステージのクリア率を比較すると、検定による有意差は見られなかった。このことから提案手法 A で

表 1: 各ステージのクリア率 [%]

Stage	既存	A	B
1	90	70	10
2	70	0	0
3	90	10	10
4	20	10	10
5	100	100	90
6	30	0	0
7	60	0	10
8	100	60	50
9	40	30	20
10	50	10	0

配置される Platform の上に破壊可能なオブジェクトを置く改変を行っても難易度を変化させることなく、ドミノステージの定義を満たすステージを生成できていることが分かる。

今回の実験では、提案手法を適用させたステージにおいてクリア率が 0% となるものがいくつか見られた。このような結果となった原因は、AI がスリングショットから遠く離れた豚を狙う際に、鳥を飛ばす角度が大きすぎて豚まで届いていなかったことや、ステージに残っている土台を避けたり、跳ね返りで距離を伸ばしたりして豚を倒す飛ばし方を見つけることができず見つけられなかったことが挙げられる。

5. おわりに

本論文では、著名なゲームの 1 つである Angry Birds において、PCG によって生成されたドミノ倒しのステージの難易度を向上させる手法を提案した。AI を用いた実験の結果、提案手法によってステージの難易度が向上することが確認できた。しかし、AI による実験ではクリアできないステージが生成されることが確認された。

今後の研究では、被験者実験を行い AI での実験との比較をする予定である。また、それらの結果に加えて多目的最適化を用いた新しいドミノ倒しの自動生成手法を提案する予定である。

参考文献

- [1] Noor Shaker, Julian Togelius, and Mark J. Nelson (2016), "Procedural Content Generation in Games: A textbook and an overview of current research", Springer International Publishing.
- [2] Chen, Jenova. "Flow in games (and everything else)." Communications of the ACM 50.4 (2007): 31-34.
- [3] Rovio Entertainment. "AngryBirds.com - The Official Home of Angry Birds". (2009) <https://www.angrybirds.com>. Accessed July 22, 2017.
- [4] Jochen Renz, XiaoYu (Gary) Ge, Peng Zhang and Matthew Stephenson. Australian National University. "AI Birds.org Angry Birds AI Competition". <https://aibirds.org>. Accessed July 24, 2017.
- [5] 蔣雨軒, 鈴木勝貴, 原田智広, ラック ターウォンマット, "文字, 数字, 記号列による Angry Birds の面白いステージの自動生成," 情報処理学会第 79 回全国大会予稿集, 6P-07. 2017 年 3 月 18 日.