

# 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

蔣 雨軒<sup>†</sup> 鈴木 勝貴<sup>‡</sup> 原田 智広<sup>‡</sup> ターウォンマツ ラック<sup>‡</sup>

立命館大学情報理工学研究科<sup>†</sup> 立命館大学情報理工学部<sup>‡</sup>

## 1. はじめに

本研究では, 著名なアクションパズルゲームである Angry Birds において, プレイヤーが面白いと感じる攻略可能なステージを自動生成する PCG (Procedural Contents Generation) [1] の手法を提案する. 提案手法では, 英文や算式として識別できるようにブロックを配置した Angry Birds のステージを生成することで, プレイヤーが面白いと感じる攻略可能なステージを自動生成する手法を提案する. 被験者実験を通して, 生成したステージに使用した英文の認識率を検証するとともに, 攻略可能なステージが生成されていることを示す. 本研究では, Ferreira 氏が開発した Angry Birds のクローン (Weird Aliens) を使用する. 同クローンは, GitHub<sup>1</sup> からダウンロード可能である. 本研究の提案手法を組み込んだ「FunnyQuoteGenerator」は CIG 2016 の Angry Birds Level Generation Competition<sup>2</sup> における Fun Track 部門で優勝した.

## 2. 関連研究

### 2.1 Angry Birds

Angry Birds のクローン (Weird Aliens) で使用可能なゲームオブジェクトを図 1 に示す. Weird Aliens のステージ自動生成では, 12 種類のブロック, 豚, Platform の配置と鳥の数を設定する必要がある. 12 種類のブロックには, Wood, Stone, Ice の 3 タイプ (図 2) があり, それぞれ色が異なる. Platform は他のオブジェクトの影響を受けない破壊不可能なオブジェクトである.

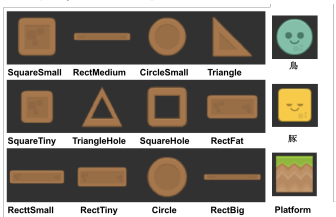


図 1 ゲームのオブジェクト



図 2 ブロックの 3 タイプ

Procedural Generation of Angry Birds Fun Levels using Sequences of Letters, Numbers and Symbols

<sup>†</sup>Yuxuan Jiang, Graduate School of Information Science and Engineering, Ritsumeikan University

<sup>‡</sup>Masataka Suzuki, Tomohiro Harada, Ruck Thawonmas, College of Information Science and Engineering, Ritsumeikan University

## 2. 2 従来研究

Angry Birds におけるステージ自動生成に関する既存研究 [2] ~ [6] について概説する.

文献 [2] [3] では, 安定的な構造単位を複数組み合わせることで複雑な構造のステージを生成する手法を提案している. 文献 [4] で提案された方法は, プレイヤーのゲームプレイの結果に基づいて遺伝的アルゴリズム (GA) のパラメータを調整することにより, プレイヤーのスキルに合ったステージを生成する. 文献 [5] では, GA を用いた Angry Birds のステージ生成において, ERA (Extended Rectangle Algebra) を使用してステージの難易度を定量化するとともに, ステージの難易度を定量化した指標に基づいて調整することができる. 文献 [6] では, 建築物の構造を記述する文法を用い, 中華風と和風の建築物を模倣したステージを生成する.

## 3. 提案手法

本研究ではプレイヤーが面白いと感じ, かつ攻略可能なステージを生成する手法を提案する. 具体的には, 図 3 に示すような Angry Birds の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, ブロックの配置を定義する Pattern-struct 及び Preset-model を提案する.

### 3.1 Pattern-struct

提案手法では, 文字や数字の形をブロックによって表現するために, 12 種類のブロックの中から RectFat, SquareTiny, SquareSmall, RectTiny, SquareHole のブロックだけを使用する. それぞれの文字に対して, その文字を表現するためのブロックの配置をあらかじめ定義したパターン構造ファイル (図 4 の左) を用意する. 3 種類のブロックタイプからランダムに 2 種類を選択し, パターン構造ファイルに定義された位置に各ブロックを配置することで文字を表現し

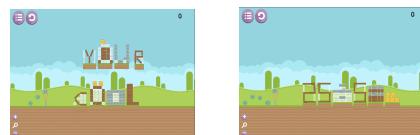


図 3 英文ステージ (左) 及び算式ステージ (右)

<sup>1</sup><https://github.com/lucasnfe/ScienceBirds/tree/NewArt>

<sup>2</sup><https://aibirds.org/other-events/level-generation-competition/results.html>

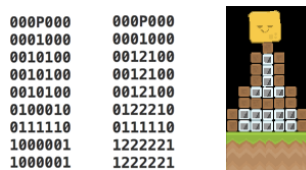


図4 パターン構造ファイルと生成した文字ブロックの例



図5 Preset-model の例

た構造を生成する。

パターン構造ファイルの“1”は文字の輪郭を意味し，“2”は文字の輪郭をサポートするブロックを意味する。パターン構造ファイルを生成する際にはまず輪郭部分を“1”としたのち、文字ブロックを安定させるために“1”の下部を“2”で満たす。“P”は豚を意味し、豚の位置はパターン構造ファイルごとに、各文字ブロック上で安定的に配置可能な箇所に記述される。

### 3. 2 Preset-model

Preset-model は、文字パターンの多様性を高めるために Pattern-struct で定義が困難なブロックの組み合わせを事前に定義する。Preset-model の例を図5に示す。

Preset-model はE, F, H, I, Tを除く大文字および1~9の数字のみを模倣し、モデルを作成した。豚の位置は Pattern-struct と同様に安定的に配置可能な位置に定義される。英文ステージを生成する場合は95%の確率で、算式ステージを生成する場合は30%の確率でそれぞれ Preset-model を使用し、それ以外の場合は Pattern-struct を用いて生成する。

## 4. 実験と結果

### 4. 1 実験

生成した英文ステージの文字認識率及び攻略できることを検証するために、以下の被験者実験を実施した。

10名の被験者に、男性は8名、女性は2名、年齢は21歳~25歳、平均年齢は23.3歳である。参加者の国籍は日本(7名)、中国(1名)、タイ(1名)、ベトナム(1名)である。

各被験者に、15個の英文ステージを重複しないようにランダムに生成し、豚の数は2匹から6匹の間で設置される。豚を配置可能なすべての位置に豚を配置したのち、設置数を上回る豚をステージの右から順に削除する。鳥の数は豚の数と同数とする。各ステージは2回までリトライ可能なものとし(最大で3回プレイ可能)、3プレイ以内にクリアできれば成功、できなければ失

敗とする。

被験者がステージをプレイする前に、ステージが表している文字を入力してもらい、3回正しい文字を入力できない場合に文字を認識できないと判断する。

### 4. 2 結果

被験者実験の結果、文字の認識率は91.33%であり、文献[7]で認識成功とされている90%以上の認識率であったことからブロックの組み合わせによって認識可能な文字を表現できたといえる。クリア率は94%であり、提案手法によってクリア可能なステージを生成できることが示された。

## 5. まとめ

本研究では、英文や数式と識別可能な Angry Birds のステージを自動生成する手法を提案した。具体的には、文字や数式を表すブロックの構造を定義する Pattern-struct と Preset-model の2つの手法を提案した。被験者実験の結果、提案手法によって、英文として識別可能、かつクリア可能な Angry Birds のステージを自動生成できることを示した。

今後の研究では、ステージの難易度を自動的に調整するとともに、Pattern-struct と Preset-model を改善して文字の認識率を向上する予定である。

### 参考文献

- [1]N. Shaker, J. Togelius, and Mark J. Nelson, "Procedural Content Generation in Games: A textbook and an overview of current research," Springer, 20 Nov. 2016.
- [2]M. Stephenson, J. Renz, "Procedural Generation of Complex Stable Structures for Angry Birds Levels," IEEE Computational Intelligence in Games 2016, pp. 178-185.
- [3]M. Stephenson, J. Renz, "Procedural Generation of Levels for Angry Birds Style Physics Games," Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference. 2016.
- [4]M. Kaidan, CY. Chu, T. Harada, and R. Thawonmas, "Procedural Generation of Angry Birds Levels That Adapt to the Player's Skills Using Genetic Algorithm," Proc. of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE 2015), Osaka, Japan, pp. 535-536, Oct. 27-30, 2015.
- [5]M. Kaidan, T. Harada, CY. Chu and R. Thawonmas, "Procedural Generation of Angry Birds Levels with Adjustable Difficulty," Proc. of the 2016 IEEE Congress on Evolutionary Computation (CEC 2016), Vancouver, Canada, pp. 1311-1316, Jul. 24-29, 2016.
- [6]YX. Jiang, M. Kaidan, CY. Chu, T. Harada and R. Thawonmas, "Procedural Generation of Angry Birds Levels using Building Constructive Grammar with Chinese-Style and/or Japanese-Style Models," Proc. of ASIAGRAPH 2016, To-yama, Japan, pp. 53-54, Mar. 5-6, 2016.
- [7]K. Chellapilla, K. Larson, P. Simard and M. Czerwinski, "Designing human friendly human interaction proofs (HIPs)," Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005), Portland, Oregon, USA, pp. 711-720, Apr. 2-7, 2005.