Corpus for Angry Birds Level Generation

1st Adeel Zafar
Faculty of Computing
Riphah International University
Islamabad, Pakistan
adeel.zafar@riphah.edu.pk

2nd Shahbaz Hassan
Faculty of Computing
Riphah International University
Islamabad, Pakistan
shahbaz.hassan@riphah.edu.pk

3rd Qazi Shuja uddin
Faculty of Computing
Riphah International University
Islamabad, Pakistan
qazi.shujauddin@riphah.edu.pk

Abstract—Levels testify as an important game element for Procedural Content Generation. Recently, machine learning techniques have been used for the automatic content generation of games. However, there is a need for a corpus for training purposes. Considering this, we provide a corpus for the famous angry birds game. Our corpus contains 200 levels in total, with 100 levels from the original Angry Birds game and 100 from the Baseline level generator. These levels are encoded using a tile-based approach and are freely available for applying machine learning techniques. Further to this, our next step would be to mimic the style of these levels in the forthcoming angry birds level generation competition.

Index Terms—Level Generation, Corpus, Angry Birds

I. INTRODUCTION

Procedural Content Generation (PCG) in games refers to the algorithmic creation of game content with limited human intervention. PCG has been used for a broad category of content generation in games however level generation is the most worked upon and oldest domain in that category. A subsequent amount of level generation work have used constructive approaches [1] and search based approaches [2]. Lately, machine learning techniques [3] i.e. Bayes Nets [4], Markov Chains [5], clustering [6], non-negative matrix factorization [7], and others have gathered the attention of researchers. All the machine learning techniques mentioned require training data in order to perform their tasks. Hence, our work focuses on providing a corpus for angry birds(a puzzle-action game developed by Rovio Entertainment). The gameplay of angry birds involves birds, which uses a slingshot to damage enemies. Different published studies also discuss the usage of procedurally generated game levels for angry birds. However, there is no corpus that could be used to generate procedural levels. Keeping in mind this motivation, we provide a corpus for angry birds levels [18]. The corpus includes 200 different levels and is freely available to apply machine learning techniques. Out of 200 levels, 100 levels are generated using the baseline level generator [8] and other 100 are from the original game of angry birds [9].

In the rest of the sections, we would first explain the background work on video game level corpus, angry birds level generation competition and different generators for angry birds level. In addition, we would explain in section 3, the details of the dataset, in section 4 we would explain the potential usage of our corpus and finally, we would conclude our paper.

II. BACKGROUND

A. Video Game Level Corpus

Datasets are widely used for machine learning based tasks. Mostly, these data sets are used for applying diverse machine learning techniques and to get statistical and graphical insights. Without datasets, it is practically impossible to mature the advances in learning algorithms. In different machine learning tasks, a data set is needed to perform training. Different corpus is available for the machine learning community that includes: Credit Card Fraud Detection [10], European Soccer Database [11], hand-writing recognition [12] and many others. For a procedural content generation of game content, there are not many publically available datasets. To the best of our knowledge, there is only a single corpus [13] that exists for levels. The corpus has overall 428 levels from 12 games and 3 different formats.

B. CIG Angry Birds Level Generation Competition

The CIG Angry Birds Level Generation Competition used a game implementation of "Weird Aliens" by Lucas Ferreria using Unity 3D [8]. While the code for angry birds is not open source yet, the competitors had to use the replica mentioned. The game had fewer objects as compared to original angry birds game. The generators are evaluated based on certain metrics and each generator has to follow certain constraints for level generation. Yuxuan Jiang's generator [14] (discussed in next section) was the winner of level generation track for the year 2016 and Matthew Stephenson's [15] (discussed in next section) generator was the winner of the level generation track for the year 2017.

C. Angry Birds Level Generators

As explained earlier, angry birds is a famous action puzzle game. A number of generators have been developed for the generation of angry birds levels. Ferreira et al. [16] developed a generator using the genetic algorithm and optimized the stability of levels. GA estimates the simulation-based surfaces during which time movement of the elements is estimated. The algorithm aims to minimize this metric to produce stable structures. A level is represented by an array of columns composed of blocks and pigs. The evolutionary process is
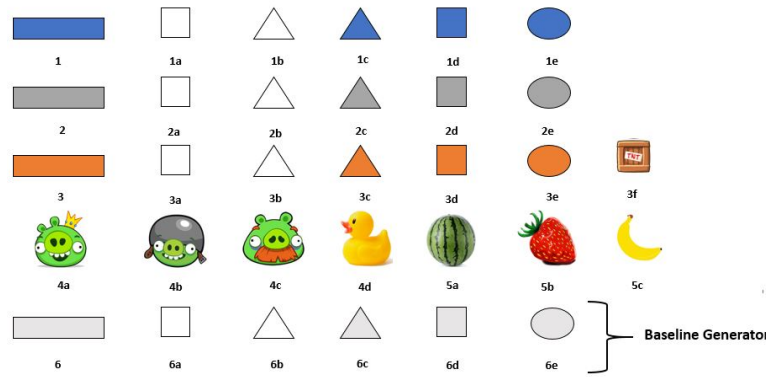
Fig. 1. This figure explains the encoding scheme used for making the dataset.

used with crossover and mutation to generate new candidate levels. The crossover operation acts in the recombination of columns and the mutation directly affects the elements, randomly changing block type and distance between columns with a certain probability. All the individuals were weighed by a fitness function which contemplates the velocity of the blocks. It also considers the number of blocks and pigs inside the level.

Kaidan et al. [17] build upon the work of ferreira et al. [16] and proposed a generator that adjusts the dynamic difficulty for players. Jiang et al. [14] presented a level generator (winner of the 2016 level generation track for angry birds) that used pattern-structure and preset-model for the generation of levels. Stephenson et al. [15] created a generation that created complex and stable structures using different 2D objects. The structures were moreover evaluated using a fitness function. Majority of techniques discussed above-used search-based approaches for level generation hence our immediate priority is to introduce machine learning techniques in this study.

III. DATA-SET

In total, we have included 200 levels in the corpus. 100 levels are from the original angry birds game and the other 100 are generated using the baseline level generator (free source code is available for the generator). The levels are formatted using a tile-based format. The tile-based approach is mostly used in games, where the game design involves a grid which is further divided into tiles. The levels are represented as a 2D grid with height and width. Each entry in the grid represents a game object i.e. blocks, enemies or ice. Table I explains the dataset sources. Since we have used two different types of game sources. One is the first version of angry birds game and the second one is the baseline level generator (open sourced). The matrix size explains the level size.

Figure 1 explains the encoding scheme used for making the corpus for the angry birds game. Each element is encoded using a character value and it could be easily used for further manipulation. Figure 2 and 3 explains the level generated using the angry birds game 1 and baseline level generator respectively along with their datasets.

TABLE I
THE GAME LEVELS INCLUDED IN THE CORPUS.

Data Collection Source	Matrix Size	Levels
Angry Birds Game 1	12 * 15	1-60
Angry Birds Game 1	12 * 30	61-74
Angry Birds Game 1	15 * 15	75-92
Angry Birds Game 1	12 * 30	93-100
Baseline Generator	16 * 32	1-10
Baseline Generator	15 * 20	11-80
Baseline Generator	16 * 32	81-100

In this dataset each element of the angry bird was represented by an integer value from 1 to 6 and each integer number was further divided into character encoding.

- 0 value is used for Space/Empty
- 1 value is used for all items related to Blue-Ice
    - 1: Ice Rectangular Big
    - 1a: Unfilled Ice-Square
    - 1b: Unfilled Ice-Triangular
    - 1c: Filled Ice-Triangular
    - 1d: Filled Ice-Square
    - 1e: Filled Ice-Circle
- 2 value is used for all items related to Stone
    - 2: Stone Rectangular-Big
    - 2a: Unfilled Stone-Square
    - 2b: Unfilled Stone-Triangular
    - 2c: Filled Stone-Triangular
    - 2d: Filled Stone-Square
    - 2e: Filled Stone-Circle
- 3 value is used for all items related to Wood
    - 3: Wood Rectangular-Big
    - 3a: Unfilled Wood-Square
    - 3b: Unfilled Wood-Triangular
    - 3c: Filled Wood-Square

Fig. 2. This figure explains the level along with the dataset of angry birds-1 game.

```
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   4a  0   0   0   4b  0   0   0   4a  0   0
3   3   3   0   3   3   3   0   3   3   3   0
0   2   0   0   0   2   0   0   0   2   0   0
0   2   0   0   0   2   0   0   0   2   0   0
0   2   0   0   0   2   0   0   0   2   0   0
```


Fig. 3. This figure explains the level along with the dataset of baseline generator for angry birds game.

```
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   6e  3f  3f  0   0   0
0   0   0   0   0   0   3   3   3   0   0   0
0   0   0   0   0   2d  2d  0   2d  2d  0   0
0   0   0   0   0   3d  3d  3d  3d  3d  0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   3c  0   0   0   3f  3e  0   0
0   0   0   0   2a  0   0   0   0   3a  0   0
0   0   0   0   2d  2d  0   0   2d  2d  0   0
0   0   0   0   2d  2d  0   0   6d  6d  0   0
```

3d: Filled Wood-Square
3e: Filled Wood-Circle
3f: TNT Block
- 4 value is used for Pig
  4a: Green Pig
  4b: Green Enemy with Black Cap
  4c: Green Enemy with Red Face

4d: Duck
- 5 integer is used for Fruits
  5a: Watermelon
  5b: Strawberry
  5c: Banana
- 6 value is used for all items related to White-Ice(only used for the Baseline Level Generator)

6: Ice Rectangular-Big
6a: Unfilled Ice-Square
6b: Unfilled Ice-Triangular
6c: Filled Ice-Triangular
6d: Filled Ice-Square
6e: Filled Ice-Circle

IV. Potential Usage

A. Style Transfer

A rather straightforward use of this corpus could be the style mimicking or transfer of style from the original game. While the original game is famous we can use a text mining technique i.e. n-gram or Markov chain for style transfer.

B. Mixed-initiative Design

Another use of this corpus could be to use it as an input for any machine learning algorithm and the designer can thus reduce the training time, errors and other barriers. The corpus could also be used to auto-complete some part of the level which the designer has left for the machine learning algorithms.

C. Data Compression

The initial application of PCG methods was for data compression. Since there was limited disk space, the compression of data was valuable. By using the likewise techniques, we can efficiently store data and it could also store distinctive features cheaply.

V. Conclusion

Machine learning is becoming a trending research area in the field of procedural content generation. In this paper, we discussed the ongoing work on level generation for angry birds game using machine learning. Our initial effort was to provide a corpus for applying machine learning and generating automatic content. For the aforementioned purpose, we have developed a dataset with 200 levels. Out of 200, 100 are from the original angry birds game and 100 are from the baseline level generator. The dataset would help the game community as researchers can use this corpus for different machine learning based tasks. Furthermore, we would use this corpus for style mimicking and would create a generator for the upcoming level generation competition.

References

[1] Shaker, Noor, Julian Togelius, and Mark J. Nelson. Procedural Content Generation in Games. Springer International Publishing, 2016.

[2] Togelius, Julian, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. "Search-based procedural content generation: A taxonomy and survey." IEEE Transactions on Computational Intelligence and AI in Games 3, no. 3 (2011): 172-186.

[3] Summerville, Adam, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgrd, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. "Procedural Content Generation via Machine Learning (PCGML)." arXiv preprint arXiv:1702.00539 (2017).

[4] Summerville, Adam J., Morteza Behrooz, Michael Mateas, and Arnav Jhala. "The learning of zelda: Data-driven learning of level topology." In Proceedings of the FDG workshop on Procedural Content Generation in Games. 2015.

[5] Snodgrass, Sam, and Santiago Ontann. "Generating maps using Markov chains." In Proceedings of the 2013 AIIDE Workshop on Artificial Intelligence and Game Aesthetics, pp. 25-28. 2013.

[6] Guzdial, Matthew, and Mark Riedl. "Toward game level generation from gameplay videos." arXiv preprint arXiv:1602.07721 (2016).

[7] Shaker, Noor, and Mohamed Abou-Zleikha. "Alone We Can Do So Little, Together We Can Do So Much: A Combinatorial Approach for Generating Game Content." AIIDE 14 (2014): 1-1.

[8] Angry Birds Level Generation Competition. "https://aibirds.org/other-events/level-generation-competition.html" (accessed January 25, 2018).

[9] Angry Birds Game. "http://www.rovio.com/games/angry-birds" (accessed January 25, 2018).

[10] Credit Card Fraud Dataset. "https://www.kaggle.com/dalpozz/creditcardfraud/data" (accessed January 25, 2018).

[11] Soccer Dataset. "https://www.kaggle.com/hugomathien/soccer/data" (accessed January 25, 2018).

[12] Hand-Writing Recognition Dataset. "https://www.kaggle.com/mloey1/ahcd1/data" (accessed January 25, 2018).

[13] Summerville, Adam James, Sam Snodgrass, Michael Mateas, and Santiago Ontann. "The vglc: The video game level corpus." arXiv preprint arXiv:1606.07487 (2016).

[14] Jiang, Yuxuan, Tomohiro Harada, and Ruck Thawonmas. "Procedural generation of angry birds fun levels using pattern-struct and preset-model." In Computational Intelligence and Games (CIG), 2017 IEEE Conference on, pp. 154-161. IEEE, 2017.

[15] Stephenson, Matthew, and Jochen Renz. "Generating varied, stable and solvable levels for angry birds style physics games." In Computational Intelligence and Games (CIG), 2017 IEEE Conference on, pp. 288-295. IEEE, 2017.

[16] Ferreira, Lucas, and Claudio Toledo. "A search-based approach for generating angry birds levels." In Computational intelligence and games (cig), 2014 ieee conference on, pp. 1-8. IEEE, 2014.

[17] Kaidan, Misaki, Tomohiro Harada, Chun Yin Chu, and Ruck Thawonmas. "Procedural generation of angry birds levels with adjustable difficulty." In Evolutionary Computation (CEC), 2016 IEEE Congress on, pp. 1311-1316. IEEE, 2016.

[18] AngryBirds Dataset. "https://github.com/AdeelZafar123/AngryBirdsDataSet" (last accessed January 8, 2019).