Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220662

Maimai Chart Generation Model Based on Machine Learning Algorithms

Tianrun Lin[1]

[1]Beijing world youth academic, Beijing, China,100000

lin.tianrun@outlook.com

Abstract. Chart generation for rhythm games is a demanding task that requires specialized design knowledge. In the past, there were several models proposed to generate charts by AI for different rhythm games. However, no full model has yet been designed for Maimai DX. In this paper, the author presents a model with high modularity. The main research approach is based on prior models and fits existing algorithms into the model presented by the author. The main purpose of this paper is to propose a possible chart generation model for Maimai DX and to propose improvements for future iterations. The model is designed by blocks, and each block is composed of specific goals and algorithms. The design is characterized by a high degree of modularity in the model, which provides space for future iterations because parts can be easily replaced.

Keywords: Maimai, Machine learning, Chart design, Rhythm games, Procedural content generation.

1. Introduction

In recent years, there has been an increase in the number of connections between game design and artificial interpellants. The paper focuses primarily on a relatively unexplored genre that could potentially benefit from AI: rhythm games, specifically, the possibility of using an AI agent in the process of rhythm game chart creation. Rhythm games require players to input signals according to the rhythm of the song. Within the process, designers of the games will create an input desired by the player, called a "chart". When charts are created manually, there are a series of challenges that designers need to overcome. Finding rhythm patterns and creating difficult patterns are two major methods to determine the level of chart designers, and achieving them will require specialized knowledge of abstract concepts, for example, those characterized by disposing of structure[1].

There has been prior research into automated chart creation and procedural content generation via machine learning approaches(PCGML) applications in Taiko no Tatsujin[2, 3]. KiaiTime, as an example, is a system that includes mixed-initiative co-creative functions with a selection of difficulties. However, the limitation of Taiko is that, from the gamer's perspective, it is a 1K rhythm game, meaning that the player will only have to focus on the note from a single lane.

Humans display a natural tendency to react, spontaneously or deliberately, to the beat of rhythmic stimuli, such as music. In rhythm games, the problem of onset detection is not perfectly solved[4]. Onset detection is an analysis of a piece to locate and match notes to a certain beat. In Dance Dance Convolution, it has been the focus of chart generation. The real challenge for AI chart generation is to create engaging charts at a high difficulty level[1, 2, 5]. Like prior research in TaikoNation, which uses

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2/20220662

neural networks to generate charts, "patterning" is an essential element for the high-end level[2]. Currently, the problem of placing or generating the pattern in the right place to create a consistent chart remains unsolved.

Based on related literature, the task of this paper is to discuss possible future chart-generating models using the existing machine learning models for the game Maimai DX, an arcade rhythmic game. "Maimai" has 8 keys arranged equally around the arc of a circle. Notes will appear in the center of the circle and flow towards the keys. In addition, Maimai also has notes or movements that players must perform in the center of the screen: slide and tap. Maimai is selected for the research due to its variation in the types of keys. Unlike Taiko no Tatsujin in prior research, which only has two types of basic keys[2]. Maimai has three most fundamental elements and up to eight keys. In addition, Chart generation in Maimai needs to take into consideration the playability of the chart[2, 4]. Due to the special mechanism of sliding, two identical or partially identical slides are not allowed to appear in successive beats, which requires an extra model to test potential errors in the chart.

The paper begins by studying previous work on other rhythm games involving chart generation using machine learning. Next, the problem of onset detection will be discussed, with a focus on the chart characteristics of Maimai. In this part, the additional issues created by slide and tap will be extensively discussed. Then, the possibility for the model to learn from the previous chart will be presented with examples of famous patterns previously used. Design and evaluation of the chart will be under the same subtopic and discussed together with both positive and negative examples. Additionally, possible models for analyzing errors in the chart will be included. In general, the main focus of this paper is on Maimai DX. The model is designed with high modularity to simplify improvement for future iterations. The model described can be the foundation for future research on similar games.

2. Related Work

2.1. PCGML Approaches for Sequence Generation in Games

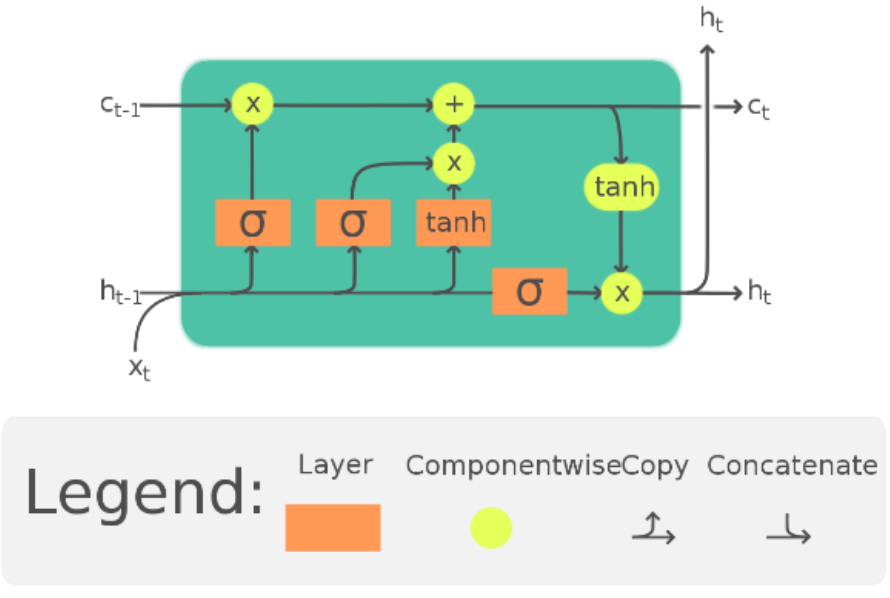

Figure 1. LSTM model diagram.

PCGML utilized machine learning models combined with the existing game content to generate new content. In prior work, this approach is frequently used. In this paper, the discussion regarding PCGML will focus more on chart generation based on existing levels. LSTMs(as shown in Figure 1), a sequence-to-sequence model was commonly used for sequence generation problems[6, 7, 8]. This model can also be applied outside the field of rhythm games, for example, card games like Magic the Gathering [9].

DOI: 10.54254/2755-2721/2/20220662

2.2. TaikoNation Attempts in Human-like Patterning

Human-like patterning is the appropriate placement of notes in a rhythm game in the correct position that responds to the event or music. In TaikoNation, a short-time Fourier transform(STFT) is performed, which uses segments of human-created patterning to compute data. TaikoNation approaches may be more complicated than the model in this paper, but due to fundamental differences in in-game mechanisms, the details for using these models may vary greatly.

2.3. Chart generation with machine learning

There have been multiple examples that used PCGML for chart generation for rhythm games. For instance, DDC utilized two separate machine learning models to solve the tasks of placing notes and note selection[2]. The chart generation model proposed for Maimai, however, resembles those utilized in GenerationMania and Beatmania IIDX[10]. In a more traditional rhythm game, the key players' input will directly influence the song. Furthermore, GenerationMania can potentially be a critical model for the model discussed in this paper because of their similar in-game mechanisms, which include multiple lanes and special notes.

2.4. Dance Dance Convolution's Onset Detection

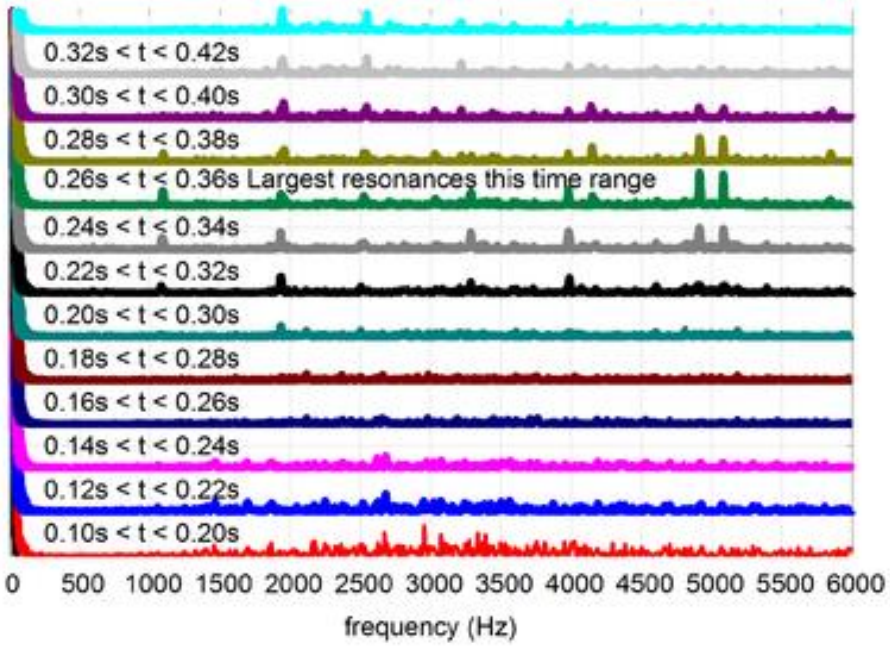

Figure 2. STFT model diagram.

In DDC, Short-time Fourier transform(STFT)(as shown in Figure 2) is used to process audio[4]. It divides audio into lengths under regulations, which simplifies the analysis of audio clips. A similar operation can be applied in any model for the chart generation of rhythm games. For the model in this paper, however, further details regarding the characteristics of the sound segments need to be analyzed for a better human-like pattern. Similar to other rhythm games, Maimai also has a special note for players to gain extra credits. Placement of these bonus notes will add challenges to onset detection.

3. Audio processing

3.1. Methods Defined

In this paper, the model integrated a prior solution. The pipeline of audio processing in suggested models is defined into four elements: feature extraction, algorithm procession, peak-picking process, and

DOI: 10.54254/2755-2721/2/20220662

suggestion of the appropriate type of notes placed. In this section, the author only mentions certain algorithms because they are the optimum choice for the model proposed in this paper.

Music files collected will be encoded constantly at 44.1khz, and stereo PCM audio will be averaged into two separated monophonic channels for a more general representation. Then models utilize an algorithm to divide the soundtrack into specific lengths for different purposes[11]. For example, a longer segment can be used to identify BMP, melody, or rhythm, while a shorter segment can be used to preserve low-level features like pitch or timber. Then, using Fabfilter, the model reduces the dimensionality of segments, or multiple-timescale short-time Fourier transform (STFT), and separates the sound with the EQ tool. Using the built-in functions can also help to automatically adjust loudness to a standard level. From prior work, however, other models also utilize Mel-scale filters to manage the audio files, which is also a possible approach[12].

With methods defined, the model only needs specific data to analyze the audio. A possible division of the audio file could be 20*50*3 sectors. These values represent 200ms from the audio context with 50 frequency bands and 3 different window lengths[13]. Window lengths are defined by the BPM of the audio files, which can be obtained by functionality in librosa from prior work. This part of the model closely resembles prior work from DDC, which also developed a representation to perform onset detection with a Convolutional neural network (CNN)[9, 14].

3.2. Peak detection

Peak detection is an alternative approach to evaluating the possibility of note placement. With a constant threshold, peaks can be detected[15]. If possible, a compressor can be applied before peak detection to prevent errors. Again, the model can use functionality in Fabfilter for this task. A standard compressor also has the function of adjusting the attack and release of the signal, which helps to prevent double-peak. Subsequently, a patterning algorithm can place notes based on the pre-set difficulty.

4. Pattern

4.1. Combine Mechanism

First, in this model, the focus is not on trying to generate patterns from scratch. The combined mechanism in this model is based on PCG techniques. In prior work, there were examples in which PCG was utilized to alter existing patterns to blend in with songs[2, 8]. Similar approaches can be used to generate appropriate patterns based on the characteristics of a soundtrack[14]. In fact, in recent human-designed patterns, we can identify some forms of combination of different patterns used in the past. Besides PCG techniques, it is also possible for machine learning algorithms to study human approaches to combine different patterns and integrate them into the model.

4.2. Pattern generation

In this paper, patterning is interpreted in two major ways: game objects concerning the track and game objects about other similar or different game objects. For example, a continuous 32nd note that forms a circular shape is a common pattern in Maimai, and it is placed mostly at the build-up or main period of a track, where the musical element is abundant, which is a specific character of the track. This pattern usually appears only in higher-level songs considering that it is difficult to perform for players. This pattern can occur on multiple scales that respond to both level and rhythm. Human-authored charts also include variations regarding rhythmic placement and types of notes used. For instance, human-authored charts can combine and integrate multiple patterns into a circular pattern and form a new pattern. The model focuses mainly on identifying rhythmic characteristics while matching the appropriate type of existing patterning to exact locations to increase both the difficulties and experience of the chart created.

For each pattern, a separate algorithm will evaluate the types of notes it contains and the placement in the charts. For the evaluation of the type of notes, an n-gram model has been proven to be effective in prior work(DDC)[4]. In DDC, a 5-gram MLP was taken to predict the steps in the chart using two connected layers with 256 and 128 nodes. In general, it demonstrates its ability to understand

DOI: 10.54254/2755-2721/2/20220662

information like time, beat, and some features in the soundtrack. For placement, the model can perform average placement of similar patterns and calculate separately on each inputted chart. A per-token accuracy will be set, and two values will be compared to determine the accuracy of the model. By configuring different weights of features via conditioned LSTM, the accuracy of the model is expected to increase [16, 17].

5. General level designing

5.1. Aim

When generating chats, playability and readability are two essential factors to consider. To generate a readable and entertaining chart, the model needs to achieve the following aims. Firstly, relativity between the soundtrack and the chart. This is usually what affects the gaming experience for a chart the most. Pitch relevancy and intuitive patterns are both effective in increasing difficulties and enjoyment[18]. Secondly, appropriate note placement. From the example above and feedback from users, it can be easily understood that certain configurations or approaches to placing notes are bad for the game experience. The third is to avoid impossible patterns. Because Maimai includes elements that require the player to interact with the inner screen, extra attestation is needed when we are trying to use a machine learning model to solve the problem.

Several aspects affect chart readability. In general, the majority of "problematic" charts existing in Maimai are how the player can effectively relate individual keys to upcoming keys. For example, "Outlaw's Lullaby" includes multiple "arcs"(three or four 32nd notes in one direction simultaneously) in an intersecting manner. In general, players will tend to use a single hand to perform an arc, and this is how players relate "arc" shaped notes together. However, the configuration in "Outlaw's Lullaby" prevents the player from sliding over keys because there are a total of three "arcs" in a single pattern. As a result, this chart was poorly rated and vastly criticized[19].

5.2. Databases

Following the discussion in section 5.1, a database for analysis purposes will be required. The database in the model is defined as follows. Firstly, the audio part and chart will be stored separately in the database. Audio files are analyzed via the approach discussed in section 3 and a similar slicing approach used in audio files will be implemented in the chart. Combining information from both sources, the model will generate a probability chart to describe the relationship between different audio features and note placements. Secondly, charts included in the database could be studied using machine learning to avoid invalid note placements. Thirdly, a database could also help with the generation and placement of patterns as described in section 4.

5.3. Generation

In general, the chart generation pipeline is designed based on modular pieces. The modular design has several advantages for solving the problem discussed in this paper. Firstly, each modular can be easily updated or changed based on changing user experience and feedback, and this is important for rhythm games because the chart will directly affect the gaming experience. Secondly, modular design allows artificial interaction with the model. Human work can easily replace any module in this model to serve special purposes. Thirdly, modular design provides space for improvement in further development.

In this model, the chart generation pipeline begins with audio processing as discussed in part 3. Each segment obtained from the audio processing module is then converted into monaural.wav files. For each feature, the model will find similar pieces from the database defined in 5.2 and concludes appropriate types, positions, and time of occurrences of a single or series of notes. The specific models applied are the PCGML approaches, which are defined as the generation of content based on existing charts[7, 20, 21]. Before the note is placed, the model will feed the concluded information into a randomizer to generate a new pattern or placement. The randomizer will be implemented with the random forest and tools discussed in section 6.

DOI: 10.54254/2755-2721/2/20220662

According to the module defined above, the model will predict the highlights of the audio files. These marked highlights will first go through the database to search previous patterns. Then these patterns are fed into the combined mechanism described above to form new patterns. Then the pattern is placed according to the features[22]. In the end, the model will feed a scratch version of the chart in a validation check. Lastly, the model adds around one to two seconds in the end and the start of the chart to increase playability. Eventually, the model outputs the resulting chart.

6. Discussion

Like other rhythm games, certain configurations in the chart are impossible to perform by the player. In Maimai, the limitations are listed below. For starters, because Maimai has multiple components which require the player to perform by touching the screen, the overlap of those notes is problematic. For example, in terms of the slide elements, "slide" is the special note that requires players' hands to move with the pre-defined track in the chart from point A to point B. As with other notes, the time a player finishes the slide will be counted for scoring. However, if the end of a slide has an additional note that players need to hit at the same time, these two notes will likely be played at poor timing. Second, some configurations are not physically possible for a player to perform. Maimai has eight keys in total, and the easiest example of this is that three keys appear separately in different locations. Usually, a chart made by a human, unless for special patterns, can only have two notes at the same time at most. As discussed in section 5, certain patterns are extremely hard to perform.

This component includes a database with pre-defined patterns that is impossible or hard to perform. A machine learning algorithm in this component then compares the inputted chart to elements in the database. The component will return, if any, problematic patterns within the chart and generate a report indicating the location and types of errors in the chart for human intervention. The main algorithm in this component is like the reverse of PCGML, which uses machine learning models trained on existing patterns to identify similar elements in the chart delivered. Similar to prior work, LSTM is used to train the model used in this component. LSTM has a lot of advantages in this model compared to other feedforward models, making it suitable for the models in this paper.

7. Conclusion

In this paper, the author suggests a possible model for chart generation for the arcade game "Maimai DX". The model was designed in terms of blocks, and each block consists of a specific aim and algorithm. This design features a model with a high degree of modularity, which provides space for future iterations since parts can be easily replaced. The current model has issues regarding specific algorithms for different modules. The paper only covers a limited number of algorithms, which is a possible direction for future research.

8. Acknowledgment

Firstly, I would like to show my greatest gratitude to my thesis instructor, who provided valuable help in all stages of this paper. Further, I appreciate professors from Carnegie Mellon University, who provided the basic knowledge regarding machine learning and made this paper possible. Lastly, I would like to show my appreciation to the language guidance, who made the language of this thesis more professional. This help and guidance are essential to the completion of this thesis.

References

[1] 5argon. Game design and notecharting. (2018). https://exceed7.com/native-audio/rhythm-game-crash-course/game-design-and-notecharti

[2] Halina, E., & Guzdial, M. Taikonation: Patterning-focused chart generation for rhythm action games. In The 16th International Conference on the Foundations of Digital Games (FDG) 2021 (pp. 1-10).Chris Donahue, Zachary C Lipton, and Julian McAuley. 2017. Dance dance convolution. In International conference on machine learning. PMLR, 1039–1048. (2021).

DOI: 10.54254/2755-2721/2/20220662

[3] Halina, E., & Guzdial, M. A Demonstration of KiaiTime: A Mixed-Initiative PCGML Rhythm Game Editor. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 17(1), 240-242. (2021).

[4] Donahue, C., Lipton, Z.C. & McAuley, J. Dance Dance Convolution. Proceedings of the 34th International Conference on Machine Learning, in Proceedings of Machine Learning Research 70:1039-1048. (2017).

[5] Brandon Johannsen. 2016. osu!mapping tutorials. https://docs.google.com/spreadsheets/d/10Z5kWe084MQVHUcuZ5VATwz-ciIpt89 XLmrCjibwyLw/edit#gid=0

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation 9, 8, 1735–1780. (1997).

[7] Adam Summerville, Sam Snodgrass, (2005).Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and J. Togelius. Procedural Content Generation via Machine Learning (PCGML). IEEE Transactions on Games 10, 257–270. (2018).

[8] Sutskever, I., Vinyals, O., & Le, Q. V. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27. (2014).

[9] Summerville, A. J., & Mateas, M. Mystical tutor: A magic: The gathering design assistant via denoising sequence-to-sequence learning. In Twelfth artificial intelligence and interactive digital entertainment conference. (2016).

[10] Zhiyu Lin, Kyle Xiao, and Mark Riedl. Generationmania: Learning to semantically choreograph. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Vol. 15. 52–58. (2019).

[11] Hamel, P., Bengio, Y., & Eck, D. Building musically-relevant audio features through multiple timescale representations. (2012).

[12] Bogdanov, D., Wack, N., Gómez Gutiérrez, E., Gulati, S., Boyer, H., Mayor, O., & Serra, X. Essentia: An audio analysis library for music information retrieval. In Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8. International Society for Music Information Retrieval (ISMIR). (2013).

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215. (2014)

[14] Schlüter, J., & Böck, S. Improved musical onset detection with convolutional neural networks. In 2014 ieee international conference on acoustics, speech and signal processing (icassp) (pp. 6979-6983). IEEE. (2014).

[15] Ullrich, K., Schlüter, J., & Grill, T. Boundary Detection in Music Structure Analysis using Convolutional Neural Networks. In ISMIR (pp. 417-422). (2014).

[16] Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., & Sandler, M. B. A tutorial on onset detection in music signals. IEEE Transactions on speech and audio processing, 13(5), 1035-1047.

[17] Adam Finn Nogaj. A genetic algorithm for determining optimal step patterns in Dance Dance Revolution. Technical Report. Technical report, State University of New York at Fredonia. (2005).

[18] Queenie, V. The Experience of Rhythm Games. The Library. https://queerqueenlywriting.wordpress.com/2021/07/30/the-experience-of-rhythm-g ames/ (2022).

[19] Outlaw's Lullaby. MaimaiChinese wiki. https://maimai.fandom.com/zh/wiki/Outlaw%27s_Lullaby. (2022).

[20] Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., & Togelius, J. Procedural content generation via machine learning (PCGML). IEEE Transactions on Games, 10(3), 257-270. (2018).

DOI: 10.54254/2755-2721/2/20220662

[21] Guzdial, M., Reno, J., Chen, J., Smith, G., & Riedl, M. Explainable PCGML via game desi gn patterns. arXiv preprint arXiv:1809.09419. (2018).
[22] Gonzalez, J., & Yu, W. Non-linear system modeling using LSTM neural networks. IFAC-P apersOnLine, 51(13), 485-489. (2018).