Slow, Repeat, Voice Guidance: Automatic Generation of Practice Charts to Improve Rhythm Action Games

Shio Takidaira, Yoshiyuki Shoji[0000−0002−7405−9270], and Martin J. Dürst[0000−0001−7568−0766]

Aoyama Gakuin University
Sagamihara, Kanagawa 252-5258, Japan
takidaira@sw.it.aoyama.ac.jp, shoji,duerst@it.aoyama.ac.jp

Abstract. This paper proposes a method to automatically generate personalized charts (music scores in rhythm action games) for improving music game skills by analyzing the play logs of music games. When learning a song, dance, or musical instrument, it is common to slowly repeat only the part the learner has difficulty with, or for the instructor to say "one, two, step, turn" in rhythm. This paper aims to confirm whether this kind of practice method can be applied to learning music games. For this purpose, we have added a function to collect logs for an existing music game, and a function to automatically generate a new practice chart from the logs. The generated chart slows down according to the failure rate of the previous play, and the section with the most mistakes is repeated. In addition, voice guides are inserted, such as "Tan Ta Ta Tan" when the player needs to tap the notes, and "from the right!" or "alternating!" for specific patterns in the chart. A comparative experiment with 12 participants showed that the learning efficiency tended to increase when all the methods were combined, and to decrease when they were used individually.

Keywords: Rhythm Action Game · Chart · Play Log Analysis · Memory

1 Introduction

Music games have become deeply ingrained in our daily lives nowadays. Rhythm action games, born in the 1990s, grew in variety and exploded in video arcades. Since then, many players have played them on home game consoles and in entertainment facilities. In recent years, these games have become more and more popular and influential in society; many world championships have been held, and the games are used in the medical field to rehabilitate the elderly[1]. Music games are also attracting attention for their educational aspects; some music

[1] BANDAI NAMCO CSR Rreport 2007, p.13: "Rehabilitainment", https://www.bandainamco.co.jp/cgi-bin/releases/index.cgi/file/view/8239?entry_id=5836

games have been adopted as part of the elementary school curriculum to develop the body and the sense of rhythm[2].

However, even with such widespread use, the best way to practice a music game has not yet been found, unlike for other music-based expressive activities such as playing musical instruments, singing and dancing. Music games often have training modes, but their functionality is limited. Functions include not quitting a song midway even if a player made some mistakes, choosing the same song continuously, and changing the play speed. In such a situation, to become more proficient in a music game, a player has to start with an easy song and repeat that song from beginning to end continuously. When they play that song well enough, they can move on to the next, slightly more complex, song, repeating the same process.

One of the main reasons people start playing music games is that the player's favorite songs were added to the game. The newcomer wants to play the song, so they start playing the music game. However, if this song is complicated, the player will have to play unrelated songs over and over again to practice their skills. If they start out wanting to play a particular favorite song, but they have to practice with an unrelated song, it can cause frustration.

Therefore, we propose a method for automatically generating the **chart** (music score in rhythm action game). The goal is to help learners improve their musical game skills by playing the same song, using techniques from traditional music education. We focused on rhythm action games in which players have to react to the notes of a scrolling chart at the correct timing (see Figure 1 and Figure 2 for examples of a screen and a chart). Our method incorporates learning methods used in playing musical instruments, singing, and dancing into the practice of a such kind of game. In the field of learning music, it is common for trainers to watch the learners' behavior and give advice. The trainer may ask the learner to repeat the same part or give verbal instruction on rhythm as the learner plays. The proposed system logs the learner's behavior while playing to enable this kind of learning in music game practice. By analyzing the learner's log, the system can estimate the learner's proficiency level and areas of weakness. Then, at the end of playing a song, the system automatically generates a personalized practice chart based on the analysis results.

When practicing singing or instruments, it is common to divide the song into phrases, repeat them frequently, or practice phrases that the learner is not good at with a slower tempo. In fact, in an extensive survey of musicians, it was pointed out that repetition is often used in the practice of advanced musicians [15].

It is also common to teach music performance by putting the movements into words and rhythm. Classically, when learning a dance choreography, trainers often attach tentative lyrics to the songs, such as "One, two, step, turn!" and so on. In the same way, when learning a musical instrument, giving tentative

[2] Konami Official Website: "American Public Schools Use KONAMI's Game for Physical Education Programs - Dance Dance Revolution -"
`https://www.konami.com/sustainability/en/culture/ddr.html`

Fig. 1. Play screen of Bemuse, the open-source rhythm action game, used in the experiment

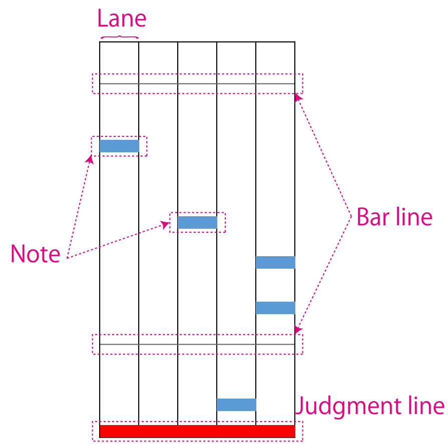

Fig. 2. An example of a chart in a typical rhythm action game

lyrics related to fingering is a common technique. These phenomena are often discussed in connection with dual coding theory [1].

When people remember non-verbal information, they can associate it with verbal information to make it easier to recall. For example, the phenomenon of skilled jazz dancers memorizing choreography by verbalizing it has been reported [19]. Some studies have shown that learning choreography is more efficient when both visual and auditory cues are used [20].

We focus on three common techniques used in such kind of musical practice:

1. **Slow:** The Slow function changes the playback speed so that the player can practice slowly. Specifically, the charts of the music game consist of background music and sound effects that are played when the buttons are tapped correctly, so that these can be played slowly without changing their pitch.
2. **Repeat:** The repeat function allows the learners to practice only their weak parts over and over again. The proposed method estimates the learner's weak parts from the play log. The method rewrites the chart so that the weak parts are repeated an arbitrary number of times. Blank bars are inserted before and after each repeat.
3. **Voice Guidance:** According to the pattern of notes in the score, the voice guide function gives tentative lyrics that make it easier to remember the action. In other words, to make it easier for the learners to grasp the rhythm, the method gives them a "Tan, Ta Ta Tan" voice to go with the notes. Also, for specific patterns, such as when the player needs to tap the left and right notes one after the other with two fingers, the system will say "Alternating!",

and when the player has to tap from one note to the next in succession, the system will say “From the right!”

For these three methods, we implemented a system that dynamically generates practice charts while analyzing the logs. A BMS (Be-Music Source, a file format) player named Bemuse [3], an open-source rhythm action game engine, was modified for this purpose. Using this system, we confirmed whether these methods are also suitable for practicing music games through subject experiments. The experimental results suggest that a combination of these three methods can be effective.

The structure of this paper is as follows. This section describes the social background of this study and the outline of our method. Section 2 describes related studies. Section 3 describes the details of our actual proposed method. Section 4 describes the evaluation experiment, and Section 5 discusses the findings from the experiment. Section 6 summarizes our research and discusses future work.

2 Related Work

This research aims to improve music game skills by automatically generating personalized practice charts for each player. This is related to existing research on improving music games and effective practice methods for improving music based on music learning theory. Therefore, we will explain the existing studies from these perspectives and show the position of this research.

2.1 Music Games

In recent years, research on music games has been very active. A typical example is developing serious games for learning music. Other research includes the automatic generation of music game charts, analysis of music games themselves, and various other research.

Many studies have been done on music games using games as a method of gamification and learning. As an example, Denis *et al.* [5, 4] propose a music game that can be used for music education by applying the theory of gamification.

As an example of the broader use of music games, Dannenberg *et al.* [13] proposed a music game for the elderly to improve dual tasks in motor cognition. Through an experiment with older adults over 65 years, the authors point out that music games can improve cognitive abilities. The music game, combined with exercises, shows the possibility of practical use for cognitive training of the elderly.

Charlotte *et al.* [18] investigate how a serious music game can encourage independent learning behavior in players. They propose three models: The feedback model, the incentive and achievement model, and the progression model.

[3] Bemuse: “Bemuse - Beat Music Sequence”
`https://dt.in.th/Bemuse.html#etymology`

In these studies, serious games still need to be improved for practical learning management, even though they have already been used in education in some cases.

Many studies automatically generate music game charts for various applications. Liang *et al.* [11] use fuzzy labels and C-BLSTM models to generate music scores for music games automatically. Lin *et al.* [14] also propose generating charts for a real rhythm action game using deep neural networks. Their system estimates the timing of the note's placement from arbitrary music files, and generates appropriate charts for each player based on their skills. Halina *et al.* [8] have proposed TaikoNation, a method for automatically generating more natural charts by focusing on humans' patterns. Similarly, Donahue *et al.* [6] proposed a method to generate a dance game score from a song using a convolutional neural network. In these studies, the goal is to automatically generate charts for new songs. On the other hand, we do not generate charts from songs in this study, but edit and adjust existing charts for learning.

Regarding the social demand for music games, Johanna *et al.* [9] have conducted a large-scale user survey from the perspective of the social and physical impact of the game series Dance Dance Revolution on players' lives. The environment surrounding the music games themselves, which are becoming popular, is also a research topic. For example, research has been conducted to detect bots in online music games [12].

2.2 Music Training Support by Using Information Technology

There has been a lot of research on the use of information technology to create practice content that meets the needs of learners [2, 3]. Roger *et al.* [21] proposed an electronic tutoring system for piano learning in the early days of computer use. This tutoring system is a classical expert system with piano input and multimedia output. The system reads the learner's input from the piano keyboard and presents followup material according to the level through multimedia. Similarly, some research aimed to support piano practice with computers comprehensively [7].

Another example of music learning support, Nakamura *et al.* [16] have conducted research on motion capture and rich information presentation interfaces to help learners train dance. For this kind of computer-based music learning, technologies have been proposed that use sensors to detect performance errors in more detail, and technologies that present information such as visualizing finger movements during practice [17, 10, 5].

Our research also refers to these methodologies, but as an objective, our proposed method aims to improve the skills of music games themselves through playing music games.

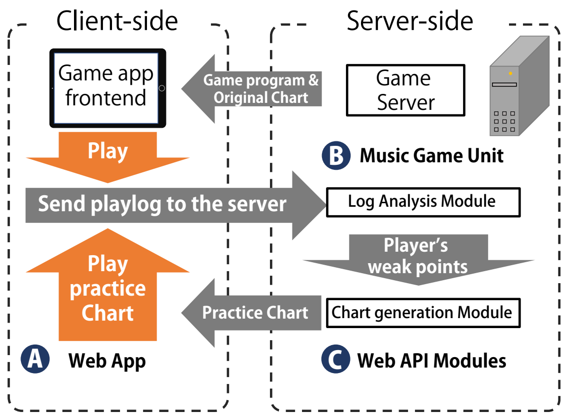

Fig. 3. Overview of the system. It consist of three parts; Client (A), Game Unit (B), and Web API Modules (C).

3 Automatic Generation of Practice Chart

This section describes our method for generating personalized practice charts containing slow, repeat, and voice guidance, by analyzing players' operation logs.

Since such a method for automatic chart generation is inseparable from the system, it was necessary to implement a working system. Therefore, this section explains the method with a description of the working system. The proposed system first collects logs, then identifies the player's weak part, and finally generates a personalized chart.

Because we used Bemuse, an open-source BMS (Be-Music Source) player, in the experiment, the rules of the game, the chart's specifications, and the system's outline will be described using this player as an example.

3.1 System Overview

In order to actually create such a practice score automatically, it is necessary to add a function to the front-end of the game that collects and sends logs. It also requires a program to analyze the logs and generate the score from the analysis results. Figure 3 shows the overview of the entire system.

When a player plays the chart of a song, a log is generated of the results of that play. The server-side application automatically generates a score for the practice based on the log. By playing the generated chart, the player can practice in a personalized way.

The proposed system is a Web application that consists of a client and a server. The client consists of a music game front-end and a log collection/sending module. The server consists of a music game server, a log receiving/analyzing module, and a personalized chart generation module. The log collection/sending module collects the logs of the chart played on the music game front-end, and sends them to the server side. The server receives the logs from the log receiving/analyzing module, and analyzes each player's weak points in the chart. The analysis results are passed to the chart generation module, which automatically generates a chart for practice. This chart is for one song that a player can play through, including slow, repeat, and voice guidance.

In the experimental implementation in this paper, we needed to prepare variant methods for comparison, such as voice guidance only, repeat only, and so on. Thus, each module is implemented as a Web API, and can be played under different conditions by simply switching the API.

3.2 Collecting and Formatting Play Logs

First, gameplay logs are collected to identify the weak points of each player. Since this music game is a web application, it is divided into a front-end that runs on the client and a server that delivers the program and charts. The front-end program runs in the browser. When users access the Web application, the program (written in HTML and Javascript) is downloaded and executed. Then the player can choose the chart to be played and download it from the server. Next the player can actually play the chart. Logging is performed by modifying the front-end program to record what is tapped at what timing sequentially (A in Figure 3). Formatting and cleansing of the logs is performed by the API that receives the logs (B in Figure 3).

The play screen in this game is shown in Figure 1 (this screenshot is the same as the original Bemuse screen, because we did not modify it except for the log collection function and the guiding voice). A schematic diagram of the chart is shown in Figure 2. A song chart consists of bars, and a bar contains several notes. Each song has a BPM (Beats Per Minute) and a time signature (*e.g.*, four-quarter time). In this game, the notes fall from the top of the screen to the bottom in time with the music. If the player taps the corresponding lane on the screen just when the note touches the judgment line, a sound effect will be played, and the player will get a score. The player will be given a grade of "Excellent", "Good", or "Bad", depending on how well they tap the screen with the right timing.

Results of the gameplay are logged in units of play. In this log, when and where the player tapped is recorded. By checking this against the chart, this is formatted into note units. Specifically, the format is generalized to the form of *(note ID, bar number, grade)*. For example, if a player got a Good evaluation by tapping the 20th note in bar 12, the result would be *(20, 12, GOOD)*. In this way, the system can handle the player's play log in an uniform way.

3.3 Estimating Player's Weak Part

Next, from the collected logs, the method estimates the player's proficiency level and the parts they are not good at (C in the Figure 3). As shown in Figure 2, a song consists of many bars, and each bar contains several notes. It is then judged for each note whether the player was able to tap the screen correctly or not. Based on the percentage of correct taps, the player's weak bars can be determined. The weakness score $weak(i)$, the degree to which a player is not good at playing the i-th bar of a song, can be defined as

$$\mathrm{weak}(i) = \begin{cases} 0 & (O_i = \phi), \\ \dfrac{\sum_{note \in O_i} \mathrm{loss}(v_{\mathrm{note}})}{|O_i|} & (otherwise), \end{cases} \tag{1}$$

where notes in the i-th bar are defined as O_i. The function $loss(v)$ normalizes the gap v between the time the note is actually tapped and the ideal timing. Here, the note $note_j$ can be expressed from the bar i it belongs to and the evaluation value v as

$$note_j = (i, v). \tag{2}$$

Using the above definition, the total degree of weakness $pro(s, e)$ from the s-th bar to the e-th bar can be expressed as

$$\mathrm{pro}(s, e) = \sum_{i=s}^{e} \mathrm{weak}(i). \tag{3}$$

Here, the sequential parts where $pro(s, e)$ is the maximum among all pairs of s and e, where $e = s+n$ for the arbitrary number of n, were defined as the player's weak parts. The method automatically generates practice chart data for these extracted weak parts.

3.4 Automatic Generation of Charts with Slow and Repeat

Next, the method generates practice chart data for the extracted weak parts by cutting out parts of the score and modifying them. The method applies slow and repeats to the chart as a simple change. The Figure 4 shows an overview of the application of slow and repeat to the chart in the system.

Slow is a modification that reduces the playback speed of a song, just as a learner slows down the playing tempo when practicing an instrument. In order to actually reduce the playback speed of music games, some implementation techniques are necessary. The output sounds during music gameplay can be categorized into two main types:

- **BGM (background music)**: sounds that are generated regardless of the player's input, and
- **Operation sound**: Sounds made in response to player actions.

Furthermore, operation sounds in music game charts are classified into

- **Keypad tone**: sound that is a component of a song, played when the player correctly taps a note, and
- **Sound effect**: sound played by the player's input that is unrelated to the completion of the music.

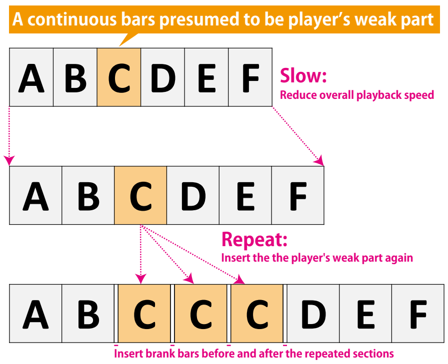

Fig. 4. The structure of the chart with Slow and Repeat. Reduce the overall playback speed and repeat the part estimated to be player's weakness many times.

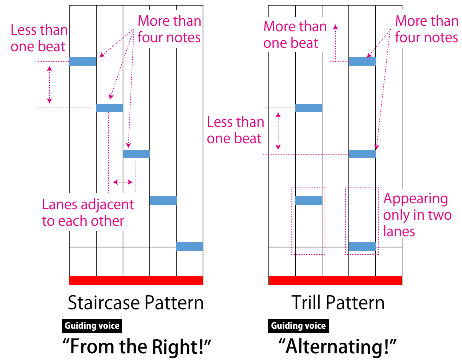

Fig. 5. Examples of a voice guide for multiple note sequences. In the case of a staircase or alternating (called drumrolls), the corresponding voice is played.

In games that adopt a keypad tone, a missed tap will cause the song's melody to be choppy. Whether a keypad tone or a sound effect is played when a player taps notes depends on the game and the song. For example, in Beatmania and "pop'n music", pressing a button will play the composition sound of the song. On the other hand, in games such as Dance Dance Revolution and Taiko no Tatsujin (Drum Master), when the player operates in time with the background music, sounds unrelated to the music are played.

When slowing down a song in a game that adopts the keypad tones, it is necessary to reduce the playback speed of both the background music and the operation sounds. In this case, only the playback speed must be reduced without changing the pitch of the sound. On the other hand, in the case of a game where sound effects are played when the player taps a note, reducing the playback speed of the sound effects would be unnatural. Since the sound effects are sounds that do not originally exist and are independent of the music, the playback speed should be reduced only for the background music.

The system used in this experiment can have both patterns depending on the song. Since it is difficult to estimate this automatically, we limited the songs used for the evaluation to songs that adopted keypad tones. Therefore, the tempo of both the background music and the operation sounds was slowed down in our system.

In addition, the tempo was not slowed down only for the weak parts that the player had difficulty with, but for the entire song. In a preliminary experiment, it was observed that beginners tended to be unable to respond quickly to sudden changes of speed during play. Since this research aims to support first-time learners, the slow method's personalization was limited for evaluation with novice participants.

Repeat is a modification that allows the player to play the weak parts over and over again. The method modifies the chart so that the player's weak parts are repeated an arbitrary number of times. The number of repetitions can be determined based on the degree of weakness of the part. In our experiment, the number of repetitions was fixed at four. Bars of silence were inserted before and after each repeat section (see Figure 4). This is to avoid players being confused because they are unaware of the loop section.

3.5 Automatic Generation of Charts with Voice Guidance

In order to help beginners practice, the system reproduces the instructional voice guide commonly used when studying music. We prepared three types of guiding voice suitable for beginners to help them remember the finger movements, which are physical actions:

- **form of note sequence**: voice when multiple notes appear in a particular sequence, such as a staircase,
- **number of notes**: voice when a series of notes appears together in a series or in parallel, and
- **rhythm**: voice simply to learn the rhythm of the notes.

The **form of note sequence** verbalizes the appearance of the group of notes and plays the guiding voice according to the form of the sequence. Figure 5 shows an example of notes flowing from the upper direction. The system discovers staircase and trill patterns based on the temporal difference and the number of lanes between one note and the next.

If all the time differences are less than one beat and subsequent notes are shifted in a specific direction, a group of notes is considered a staircase. This only applies to cases where the gap is one or two lanes. A guiding voice was created to help the user remember that the notes appear as a staircase, intuitively matching the finger movements. Since the staircase notes have a direction, we give them a specific voice: "from the right" or "from the left".

The left part of Figure 5 shows an example of a staircase. The time difference between each subsequent note is less than one beat. In addition, each note is placed one lane to the left to the previous note. So, when the first note (*i.e.*, the one displayed at the bottom) should be tapped, the voice guidance "From the right!" will be played.

If a group of four or more notes, all with a time gap of one beat or less, are alternately located in two specific lanes, then the note group is considered a trill-like pattern. This pattern of alternately tapping two lanes is typical in music games in general, and is often called drumrolls. The right part in Figure 5 shows an example of a trill pattern. The five notes shown have a time difference of less than a beat each. The five notes are placed alternately in the second and fourth lanes. Therefore, the system considers this group of notes as a trill pattern. The guiding voice will say "Alternating!" when the lowest note touches the line.

The **number of notes** guiding voice tells the player the number of notes that need to be tapped, when notes appear in succession at equal intervals, or when multiple notes need to be tapped simultaneously. Figure 6 shows an example of two types of the number of note guiding voices. In the left part, there are three or more notes in a row, evenly spaced on the same lane. The system recognizes this group of notes as a Serial Pattern. It announces the number of notes that need to be tapped in succession. In this example, the voice "5 notes" is played when the first note touches the judgment line.

Similarly, if two or more notes are horizontally lined up simultaneously, a similar guiding voice will be played. This is considered a Parallel Pattern. The right part of Figure 6 shows an example of a Parallel Pattern.

Finally, as **rhythm guide**, we assigned a simple guiding voice to notes that did not fit into the two patterns described above. The purpose of this voice is simply to make the user remember the timing of tapping. Specifically, as shown in Figure 7, the guiding voice is like "Tan Ta Ta Tan". If the number of beats between a note and the next note is more than one, the phrase "Tan" is played, and if it is less than one beat, the phrase "Ta" is played. The form pattern and the number pattern don't apply to many parts of a song's chart. Therefore, in many parts of the song, a voice like "Tan Ta Ta Tan" is constantly played.

These guiding voices can be generated by combining only a few dozen phrases. In this implementation, each phrase was created using female voices from a commercially available synthetic voice library, and the guiding voice was realized by combining them on time during play.

These guiding voices are automatically assigned in decreasing frequency of appearance (*i.e.*, the order of form, number, and rhythm). However, in actual music game charts, it is common for multiple notes to appear at the same time at all times, especially in songs of high difficulty. The scope of this rule-based implementation is limited to simple songs with not more than a single melody line.

Note that the system was implemented in Japanese, and the experiments were also conducted in Japanese. Therefore, the guiding voice is also implemented in Japanese, and translated into English in this paper.

Through this series of processes, the system becomes able to estimate the weak part of the player for the song from the play log. Then, it is possible to support the learning of music games with slow, repeat, and voice guidance.

4 Evaluation

In order to confirm whether the system we created is effective in improving players' music game skills, we conducted a subject experiment. Twelve experimental participants were each assigned two of four learning methods, including the proposed method, and played the game seven times with each method. The increase in gaming score between the seven games and the correct tap rate for each game were then measured.

The contributions in our method can be broadly divided into changing the structure of the chart by **slow and repeat** and changing the sound of the song

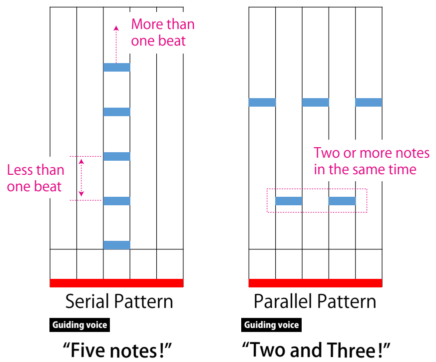

Fig. 6. Examples of a guiding voice when multiple notes appear in series or parallel. When multiple notes appear in the same lane at equal intervals (called jackhammers), the number of notes is announced. When multiple keys need to be pressed simultaneously (called chords), the number of notes is announced in the same way.

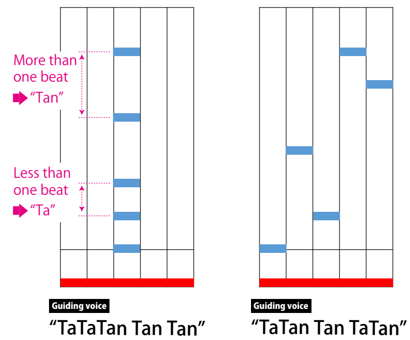

Fig. 7. Example of voice guidance in the case the notes do not correspond to either the form pattern or the number pattern. Simply, the sounds of "Ta" and "Tan" are played at the correct timing.

by **voice guidance**. Therefore, to discuss how effective each of these is, we created comparative methods in combination with them.

The subject experiments were conducted in the years 2020 and 2021. A browser on an iPad was used to access our system and play the game. The experiment took about an hour per participant, and the participants received a gratuity of about ten US dollars.

4.1 Implementation

The system was implemented as shown in Figure 3. We modified Bemuse, an open-source online BMS player. The front-end of Bemuse is developed in Javascript, and the server-side is developed in node.js. We modified the front-end to send a detailed log in the JSON format to the server when the player finishes playing a song and transitions to the result screen. The server side has been modified to deliver a new chart at any time.

As a set of APIs in the server, we created a module that receives logs, cleanses and reshapes them, and analyzes them. We also created a module that actually generates a BMS chart file based on the log analysis result. This module outputs a new chart for practice when receiving an arbitrary BMS file and the log analysis results. To implement these APIs, we used flask, a python web framework. One limitation of the implementation is that the current rule-based guiding voice assignment function is incomplete. Therefore, it only works correctly for songs with simple melody lines and not so many notes appearing at the same time.

Table 1. Four comparison methods used in the subject experiment

Method	Description
Both	Proposed method. The chart was modified with both voice guidance, and slow and repeat.
Voice Only	A comparative method for testing the effectiveness of voice guidance. The chart structure remains the same as the original one; only voice guidance was added.
Slow+Repeat	A comparative method to test the effectiveness of changing the chart structure. Slow down the entire song and repeat the weak part four times.
Baseline	Play the original score without any modifications.

Table 2. Assignment of methods and songs for each task. All participants played the proposed method, and the songs were assigned without bias.

Task ID	1st play		2nd play	
	song	method	song	method
1	A	Baseline	B	Both
2	A	Voice Only	B	Both
3	A	Slow+Repeat	B	Both
4	A	Both	B	Baseline
5	A	Both	B	Voice Only
6	A	Both	B	Slow+Repeat
7	B	Baseline	A	Both
8	B	Voice Only	A	Both
9	B	Slow+Repeat	A	Both
10	B	Both	A	Baseline
11	B	Both	A	Voice Only
12	B	Both	A	Slow+Repeat

4.2 Comparison Methods

We prepared four comparative methods to evaluate the effectiveness of changing the chart structure and the voice guidance. Table 1 shows the list of methods, including our proposed method. The method **Both** includes both modifications to the chart structure and voice guidance. The methods **Slow+Repeat** and **Voice Only** only change one of them, respectively. Note that **Voice Only** is not personalized, as the guiding voice is given throughout the entire song.

We imposed several restrictions on each method to fit the experimental setup of this study. For methods containing personalization, the number of repetitions was fixed at four. In addition, we fixed the number of week parts to one place of four bars. This is to avoid significant differences in playing time between personalized and non-personalized methods. Such practice time differences would have a more substantial effect than the method differences we are investigating. For the same reason, we set the slow methods' playback speed to be uniformly 0.8 times the speed of the original song.

4.3 Experimental Settings

We chose two songs for the task and adjusted the game for the experiment. In choosing the songs, we considered the following conditions:

- The song's tempo is not changed in the middle of the song,
- The length of the song is around 100 seconds (or the first part of the song is around 100 seconds and can be cut),
- The number of notes to be tapped at the same time does not exceed four, and
- The density of the notes is less than four notes per second throughout the entire song.

We eventually used the following two songs for our experiment:

- **Song A**: Dream Map and Our Journey[4]
  (Original title: Yume no Chizu to Bokura no Tabi.), and
- **Song B**: My Own Affairs[5]
  (Original title: Jibun Goto).

These two songs were used in a famous BMS event. In our experiments, we removed the prelude of the songs and used only the first half of the songs. The player needs to tap about 3.5 notes per second in these songs.

We also assigned each participant songs and methods, taking into account order to avoid bias. Table 2 shows the actual method and song assignment for each task. For each method, the song is used the same number of times. In this experimental setting, the order effect in the experiment is expected to be strong because participants are beginners. Therefore, we ensured that the number of times each method was used was equal in the first play and second play. Since we wanted to focus mainly on the difference between the proposed method and the others in the analysis, the proposed method was set to be played by all subjects. In this experiment, each task was assigned to a single participant.

As in other detailed experimental settings, the time to determine if a note was tapped correctly was set to 120 milliseconds. This is the standard value for combo (a rule that gives a player bonus points when they tap notes in succession) in BMS. The Lead Time, which determines the speed throughout the game, was set to 1,800 milliseconds. For the experiment, we used an iPad with a 10.2-inch display. During the experiment, the 3D mode was used, as shown in Figure 1, and participants tapped directly on the lanes on the screen. Participants used earphones during the experiment.

4.4 Experimental Task

The participants in the experiment were asked to play a rhythm action game to measure their playing skill improvement. First, the participants were given iPads and were explained the rules of the rhythm action game and the basic operation of the system. Next, each subject played each song seven times. Figure 8 shows an overview of the sequence of steps performed by the participants during the experiment. Except for the baseline, participants played three different charts. They played the original chart the first time. Then, the next two times, they played a practice chart generated based on the first play. The fourth time, they played the original chart again. Then they played the practice chart generated from the fourth play log twice. Finally, they played the original chart again.

[4] G2R 2018 Climax Song Information: "Dream Map and Our Journey" (in Japanese) `http://manbow.nothing.sh/event/event.cgi?action=More_def&num=287&event=123`

[5] The BMS Fighters XVI Song Information: "My Own Affairs" (in Japanese) `http://manbow.nothing.sh/event/event.cgi?action=More_def&num=181&event=133`

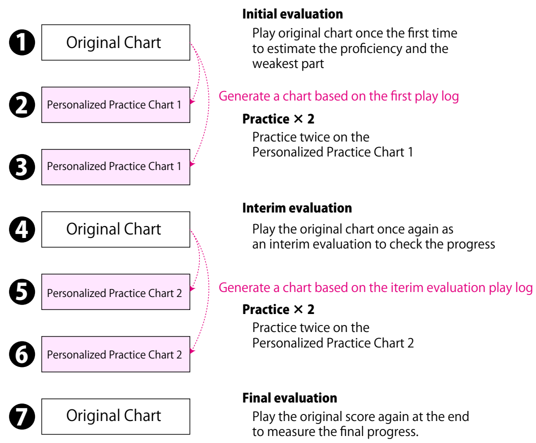

Fig. 8. Participants' tasks during the experiment. Participants played the game a total of seven times. To fix the experimental setup, all participants played the original chart three times and two generated training charts two times each.

There are two reasons why we choose such task settings. One is that if the system created practice charts from the play logs of the practice charts, there would be a difference in playing time depending on the participants' initial skill. For example, if an awkward player makes a mistake in a repeated part again and again, the chart gets longer with each turn. The other reason is that we wanted to compare methods on the same chart.

The participants performed two sets of the task, with each set consisting of seven such plays. They played different songs and methods in the first and second sets. There was a break in between the two sets.

4.5 Experimental Results

To see how much the participants actually improved, we calculated the correct tap rate during each play, and the amount of progress throughout the seven plays. The correct tap rate was calculated by dividing the number of correct taps (*i.e.*, the number of times the participant tapped the lane within 60 milliseconds before or after the note touched the judgment line) by the total number of notes that appeared in the chart. If all notes are tapped within this interval, the rate is 100 percent; if none are tapped, the rate is 0 percent.

Table 3 and Figure 9 show the correct tap rate for each play time, and the degree of growth from the first time to the seventh time for each method. The degree of growth is the difference of rates between the seventh time and the first time. The method using both voice and slow and repeat had the highest final correct tap rate (81.83 %) and the highest growth (21.81 points). Participants

Table 3. Rate of correct taps for each method (%), growth rate (points) and standard deviation (SD) of correct tap rate for each condition.

Play times	Both	Voice Only	Slow+Repeat	Baseline
1	60.02	51.20	74.19	68.20
2	72.56	60.04	79.92	74.01
3	73.34	59.85	80.34	69.87
4	76.53	62.90	78.09	78.03
5	70.95	62.51	75.27	71.03
6	74.66	67.08	79.33	73.27
7	**81.83**	66.16	78.24	75.84
Growth	**21.81**	14.95	4.05	7.64
SD @ 1	21.60	17.95	12.25	9.99
SD @ total	18.86	15.83	12.25	13.17

who practiced with the proposed method became able to tap more than 80% of the notes correctly after seven practice sessions. On the other hand, the method with the smallest difference was the slow-repeat method, with a difference of only 4.05 points, which resulted in less improvement than simply playing the regular chart repeatedly.

5 Discussion

This section discusses the characteristics of the proposed method and its usefulness based on the experimental results obtained. First, the usefulness of method **Both** was significantly higher in terms of growth rate. From Table 3, we can see that the degree of increase in the correct tap rate of method **Both** is 21.81 points, which is higher than the other methods. This result suggests that the proposed method may actually be effective in improving rhythm action game skills.

Focusing on the possibility that some participants improved faster than others in the music game, from the first row in Table 3 we can see that even though all the participants played the same chart on the first play, there was a difference in the scores for each method. This trend can also be seen in the standard deviation of the first play and the average. It indicates that people have different suitability for rhythm action games. However, all participants had played the proposed method once, which was effective for them. Therefore, we can assume that this result does not depend on the potential of the participants.

Next, we consider the cause of the decrease in progress efficiency in the **Slow+Repeat** method. Compared with the other methods and the baseline method, the increase in the correct tap rate of the **Slow+Repeat** method was relatively low, at 4.05 points. One reason could be that the change in speed was too burdensome for the novices. As described in Section 3, beginners may not be able to respond quickly to speed changes or sudden changes in their gameplay.

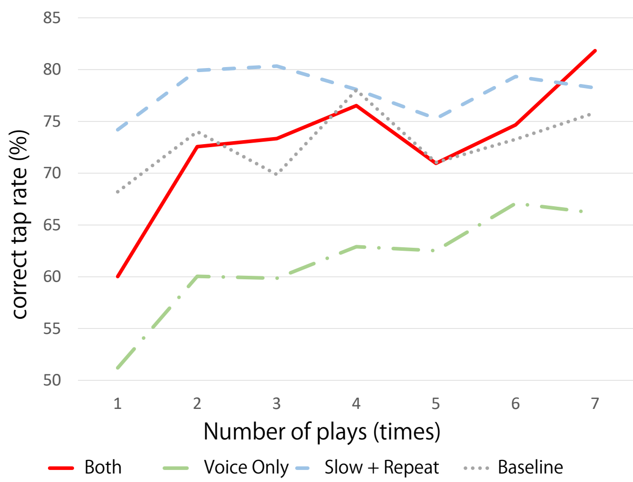

Fig. 9. Number of plays vs. correct tap rate of notes for each method.

In the experiment, the speed changes during the fourth and seventh task in the set, which may have hindered the beginner's progress.

These results suggest a synergistic effect between the voice guidance and **Slow+Repeat** methods. The proposed method, which includes slow-repeat, may also have inhibited the improvement due to playback speed changes. However, the participants who practiced with the **Both** method improved faster.

Considering this result, it is possible that the voice guidance accelerated the memorization of phrases. It is important to remember and internalize which notes to tap when learning to play. It is difficult to feel the beat if the tempo is too slow when trying to internalize the rhythm without audio guidance. It can be assumed that the phrase was learned quickly by practicing the weak part slowly and repeatedly, with the tap timing and fingering indicated by the voice guide.

Intuitively, it is similar to the behavior when learning a song; once a song is learned perfectly, it can be sung at different tempos. Once the player has internalized the rhythm, performance may become more resistant to tempo variations. One of the reasons why the **Voice Only** method did not promote growth could be that the original tempo was too fast and only be practiced once, thus failing to promote the internalization of phrases.

In addition, order effects should be discussed. We focused on which of the methods and songs was tried first during the evaluation experiment. The participants played different songs in the first and second sets in the experiment. Table 4 shows the correct tap rate and its standard deviation for each song and position. Overall, it can be seen that the participants improved more in the second set than in the first set. This may be because their skill to play music games improved while playing the first set of songs, and they were able to demonstrate

Table 4. Correct tap rate (%) and standard deviation of the first set for each song

Song played earlier	Average rate		SD	
	Earlier	Later	Earlier	Later
Song A	50.72	78.67	16.44	9.60
Song B	60.94	58.77	21.93	16.26
Total	55.83	68.72		

the skill even when they saw the score for the first time in the second set of songs.

We discuss the difficulty and quality of the songs themselves. Comparing the standard deviation, we can see that when Song A was played as the first song, the standard deviation was lower in the second song than when Song B was played as the first song. There also exists a difference in the average correct tap rate between songs. It can be seen that different songs have different difficulty levels and different variations in how well they can be played by players who have never played the song before. In this experiment, due to the limitation of the scale of the experiment, we were only able to compare two songs. In order to actually compare training methods on a large scale, of course, it would be necessary to conduct experiments using a more significant number of songs.

In the end, the proposed method showed a tendency to be useful, but the details are not precise. It will be necessary to evaluate in detail what kind of support contributes to the improvement of rhythm action games through a large-scale evaluation experiment with a larger number of songs.

6 Conclusion

We proposed an automatic generation of rhythm action game charts for practice by analyzing the game's operation logs to match each player's weak parts. We modified Bemuse, an open-source game engine, to implement a system that can receive logs and play practice scores. The music practice methods of slow, repeat, and voice guidance were applied to a music game and evaluated in a subject experiment. The experimental results showed that the combination of all these methods can help novice players improve their game skills faster. The experiment suggested that voice guidance and slow+repeat may synergize and help players internalize phrases.

On the other hand, the results of this experiment are limited. Some issues for future research were found. One of the issues is to improve the accuracy of extracting the parts that players are not good at. We simply used consecutive bars with a low correct tap rate as the weakest sections; however, we did not consider the difficulty of the bar itself or the number of notes it contains. The scale of the experiment is also an important issue. This paper conducted a small-scale experiment using two songs for 12 participants. Subjects practiced the prescribed song seven times, interspersed with the original charts. However, this task is unnatural compared to a real player trying to improve their skills while playing a music game. We hope to conduct a larger-scale evaluation in the future.

Acknowledgements

This work was supported by JSPS KAKENHI Grants Number 18K18161 and 21H03775.

References

1. Clark, J.M., Paivio, A.: Dual coding theory and education. Educational psychology review **3**(3), 149–210 (1991)
2. Dannenberg, R.B., Sanchez, M., Joseph, A., Capell, P., Joseph, R., Saul, R.: A computer-based multi-media tutor for beginning piano students. Journal of New Music Research **19**(2-3), 155–173 (1990)
3. Dannenberg, R.B., Sanchez, M., Joseph, A., Joseph, R., Saul, R., Capell, P.: Results from the piano tutor project. In: Proceedings of the Fourth Biennial Arts and Technology Symposium. pp. 143–150 (1993)
4. Denis, G., Jouvelot, P.: Building the case for video games in music education. In: Second International Computer Game and Technology Workshop. pp. 156–161 (2004)
5. Denis, G., Jouvelot, P.: Motivation-driven educational game design: Applying best practices to music education. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology. p. 462–465. ACE '05, ACM, New York, NY, USA (2005)
6. Donahue, C., Lipton, Z.C., McAuley, J.: Dance Dance Convolution. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 1039–1048. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017)
7. Gower, L., McDowall, J.: Interactive music video games and children's musical development. British Journal of Music Education **29**(1), 91–105 (2012)
8. 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. FDG'21, ACM, New York, NY, USA (2021)
9. Höysniemi, J.: International Survey on the Dance Dance Revolution Game. Comput. Entertain. **4**(2), 8–es (Apr 2006)
10. Inoue, N., Fujimoto, Y., Plopski, A., Okahashi, S., Kanbara, M., Hsu, H.Y., Kuo, L.C., Su, F.C., Kato, H.: Effect of display location on finger motor skill training with music-based gamification. In: International Conference on Human-Computer Interaction. pp. 78–90. Springer (2020)
11. Liang, Y., Li, W., Ikeda, K.: Procedural Content Generation of Rhythm Games Using Deep Learning Methods. In: ICEC-JCSG (2019)
12. Lin, R.M., Ho, H.C., Chen, K.T.: Bot detection in rhythm games: a physiological approach. In: Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology. pp. 1–8 (2011)
13. Lin, Y.H., Mao, H.F., Lin, K.N., Tang, Y.L., Yang, C.L., Chou, J.J.: Development and Evaluation of a Computer Game Combining Physical and Cognitive Activities for the Elderly. IEEE Access **8**, 216822–216834 (2020)

14. Lin, Z., Xiao, K., Riedl, M.: Generationmania: Learning to semantically choreograph. In: Proceedings of the Fifteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. AIIDE'19, AAAI Press (2019)
15. Maynard, L.M.: The role of repetition in the practice sessions of artist teachers and their students. Bulletin of the Council for Research in Music Education pp. 61–72 (2006)
16. Nakamura, A., Tabata, S., Ueda, T., Kiyofuji, S., Kuno, Y.: Multimodal presentation method for a dance training system. In: CHI '05 Extended Abstracts on Human Factors in Computing Systems. p. 1685–1688. CHI EA '05, Association for Computing Machinery, New York, NY, USA (2005)
17. Nakamura, T., Nakamura, E., Sagayama, S.: Real-Time Audio-to-Score Alignment of Music Performances Containing Errors and Arbitrary Repeats and Skips. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(2), 329–339 (2016)
18. Pierce, C., J. Woodward, C., Bartel, A.: Learning Management Models in Serious Mobile Music Games. In: Proceedings of the Australasian Computer Science Week Multiconference. ACSW '20, ACM, New York, NY, USA (2020)
19. Poon, P.P.L., Rodgers, W.M.: Learning and remembering strategies of novice and advanced jazz dancers for skill level appropriate dance routines. Research Quarterly for Exercise and Sport **71**(2), 135–144 (2000)
20. Shehan, P.K.: Effects of Rote Versus Note Presentations on Rhythm Learning and Retention. Journal of Research in Music Education **35**(2), 117–126 (1987)
21. Smoliar, S.W., Waterworth, J.A., Kellock, P.R.: pianoFORTE: a system for piano education beyond notation literacy. In: Proceedings of the third ACM international conference on Multimedia. pp. 457–465 (1995)