Procedural Content Generation of Rhythm Games for Self-Training

1610211 LIANG YUBIN

Rhythm game is a genre of music-themed (action video) game in which players play by taking actions in accordance with rhythm and music. Since music or songs are familiar to ordinary people, it is easy for people to understand how to play such games. In addition, both of easy stage and hard stage can be created from one music, therefore, rhythm game becomes a popular game genre in the whole world.

In many cases, the contents (required action and its timing) of rhythm game are handcrafted by human designers from music material. Also, there are countless pieces of music, but only a part of them have already been used as game contents. Therefore, in our opinions, automatic contents generation is required.

Apart from this, it is frequently pointed out that rhythm games are hard to practice. For example, assume that a player is not good at a part (5 seconds) of a rhythm game stage. Even when he wants to repeat only the part, it is impossible and he needs to play the whole stage (3 minutes). This is one of reasons that rhythm games are hard to practice.

Based on those existing requirements, we propose a self-training support system with automatic content generation capability. The system possesses capabilities corresponding to each task such as "generate content from audio file", "evaluate the level of a player", "modify the action combinations in content to produce appropriate difficulty".

In this research, we proposed an approach that generates contents automatically from music materials by deep learning. We used supervised learning method which inputs audio data, outputs timestamps (timing of action) and action type. As a machine learning task, this task has some difficulties such as, 1) even when the same music is used as an input, the outputs may be completely different, due to author variations and level variations, 2) proportion of positive/negative samples is ill. To deal with those tasks, in this research we adopt that, 1) handle the difficulty settings as one input feature, 2) using fuzzy labels to increase positive samples. By applying the proposed training methods, it has been proved that the training performance and prediction accuracy were increased. The combined training method has improved the F-score of the timestamp prediction from 0.8159 (by the existing method) up to 0.8430.

Furthermore, in this research we proposed various methods to support self-training for players. First, after a player played some games, his/her mistakes will be analyzed and shown to the player, from several viewpoints

such as, “frequent mistake types (too late, too early, pushed a wrong button and so on)”, “the situations which the player often failed (many actions, frequently changing buttons, many long press actions and so on)”, or “what kind of action combinations are difficult for the player”. Those analysis results allow players to notice their own weaknesses.

Besides, to make the player's practice more efficient, we implemented a function to increase proportion of difficult action combinations in the next content generation. Finally, regarding the self-training support capabilities, we conducted a questionnaire on subjects such as “was the analysis result accurate?”, “were weak action combinations increased?”, to verify the usefulness of the proposed methods, and received positive answers.