Curve-Based Level Generation for a Full-Body VR Rhythm Game

Erin J.K. Truesdell
Georgia Institute of Technology
Atlanta, GA, USA
erinjktruesdell@gatech.edu

Abstract

Due to the large amount of content authoring required to produce levels for rhythm games, procedural content generation provides an attractive alternative for generating levels in large numbers. Frequently, however, level generators for rhythm games rely only on a "difficulty" measure to describe desired level properties. This work presents level generation tool for BEAMS, a full-body virtual reality rhythm game. The BEAMS level generator allows designers to shape curves describing four characteristics of the obstacles needed for a game in which the primary interaction involves the human body in three-dimensional space.

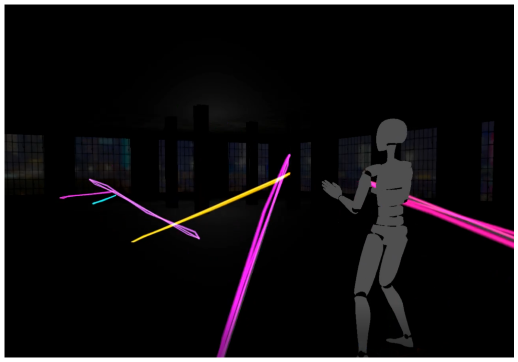
Figure 1: BEAMS gameplay.

Introduction

In recent years, virtual reality games (including *Beat Saber* and *HoloDance*) have emerged that combine body movement and rhythmic coordination, drawing on previous works such as *Just Dance* and *Dance Dance Revolution* and taking advantage of advancements in consumer game control technology. BEAMS is a virtual reality rhythm game that uses a wireless motion capture suit to engage the player's full body as its controller. Drawing inspiration from scenes in heist movies such as *Ocean's Twelve* and arcade laser mazes, BEAMS times laser-shaped obstacles to the beat of music, asking its players to dodge oncoming laser beams as though they are moving through a maze. Unlike contemporary commercial motion platforms, the controller for BEAMS is capable of reading the motion of the human body in three dimensions, thus offering enhanced levels of avatar control. BEAMS seeks to combine this high level of fidelity with virtual reality space to create an immersive rhythm experience. A level generator application has been developed specifically for BEAMS with the goal of generating a large number of levels for the game without requiring prohibitive amounts of hand-authoring.

Related Work

The challenges of authoring levels for rhythm games is well-documented. A number of tools have been developed to aid in level creation for designers seeking the ability to quickly generate large numbers of levels as well as casual creators who may not have the design expertise needed to craft a level from scratch. Level generation strategies for two-dimensional platformers often generate levels based upon specific elements of gameplay, such as the rhythm of a player's movement through a level (Smith et al. 2009) or play styles (Summerville et al. 2016). Approaches to rhythm game level generation have leveraged techniques including the use of neural networks (Donahue, Lipton, and McAuley 2017; Tsujino and Yamanishi 2018) and selection from a set of prerecorded moves (Martin et al. 2019). These approaches typically limit designer input to a "difficulty" parameter.

Such approaches work well for games in which there are a limited number of discrete inputs (such as controller buttons or a *Dance Dance Revolution* pad). However, projects such as BEAMS that focus on continuous interactions in three-dimensional space require additional measures of control. "Difficulty" in such a project may take many forms, and is difficult to describe with a single value. For this reason, the BEAMS level generator includes four curves describing properties of level components over time, affording greater designer control over the level attributes.

Level Generator

The BEAMS level generator creates a level by reading in and analyzing a .wav file and using information from four user-defined curves describing desired properties of the obstacles.

Curve Definition

The BEAMS Level Generator affords the definition of four curves, each describing the desired characteristics of lasers spawned over the course of the level. The curve's x-axis describes time, where 0.0 is the beginning of the song and 1.0 is the end. The Y-values of curve points describe the desired value for each characteristic at the corresponding time point in the song.

"Height" and "left/right position" curves describe the placement of the centers of the laser obstacles relative to the player. Two transverseness curves control the angle at which the laser obstacle is spawned. An up/down transverseness value of 0.0 will lead to selected lasers that are highly horizontal; a value of 1.0 will generate portions of a level where lasers approach vertical. Figure 2 illustrates laser obstacles exhibiting high and low values for each characteristic. The

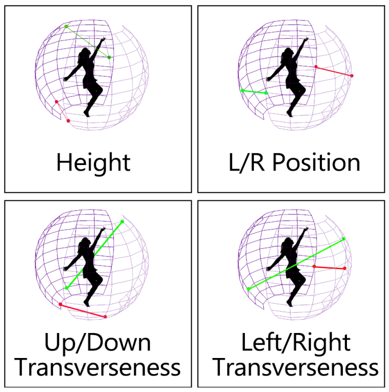

Figure 2: The four values described by curves in the BEAMS level generator.

Level Generator interface allows users to edit the curves before the level is generated. Users may insert, remove, and manipulate points on the curve to describe the desired properties of spawned obstacles from the level's beginning to its end. This interface, within which a left/right position curve is being edited, is depicted in Figure 3.

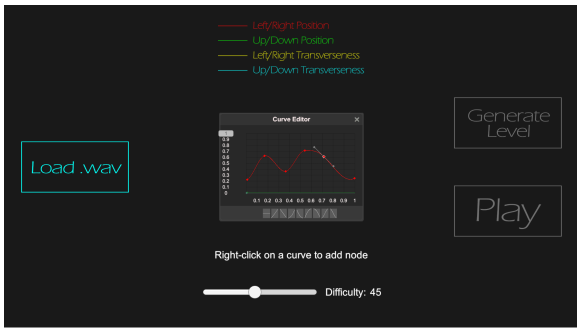

Figure 3: The Level Generator interface, featuring a left-right position curve that varies over the course of the level.

Beat Detection

While many rhythm game level generators use more advanced technology to analyze music for levels, BEAMS applies a simpler approach that is augmented by the designer's control over level properties. Before a BEAMS level is generated, analysis is performed on the loaded sound file to determine where in the music laser obstacles should be spawned. This relies on the ability of the system to identify both the tempo of the music uploaded, as well as its ability to identify local peaks where laser obstacles may be spawned "on the beat." Pre-analysis of the loaded sound file reduces the information in the original file to a single channel, and further reduces it down to 50 samples per second by averaging the sample values for each 1/50 second step. The level generator uses this reduced sample set to detect local peaks in the music, and determine which peaks will be matched to the introduction of an obstacle to the player's space. A modifiable "difficulty" parameter allows users to specify the maximum number of lasers occurring for each 1000 samples in the song, and thus provides a means of adjusting the density of obstacles in the level.

Laser Candidate Generation and Selection

Laser obstacles are generated for each moment in the level where they must occur; for each moment where a laser is required, the generator produces 35 candidate lasers with random heights, left-right positions, and rotation values. Each candidate is assigned a cumulative score reflecting its closeness to or distance from the characteristics specified by the curve at that point in the level. Closeness to a curve-specified value corresponds to a lower component score for that characteristic. All four component scores are weighted equally. The candidate with the lowest cumulative score is selected for inclusion in the level, which is stored as a single text string comprising 0s and parenthetical information containing positional and rotational data about the laser obstacles in the level. The level information is then able to be exported for use in the main BEAMS game, where a level reader parses the exported value and uses the information to generate laser obstacles during play.

Discussion

Early use of the BEAMS Level generator have produced playable BEAMS levels whose properties reflect the designer's input curves. Future work on this project will investigate human players' experience playing BEAMS, and will be used to evaluate the effects of various curve properties on player experience and perception of difficulty. Additionally, a significant limitation of BEAMS is its reliance on hardware that is not intended for consumer use; future iterations of this project, and by extension, this level generator, may take into account a version of BEAMS that can be played on consumer hardware, and will include level generation strategies to reflect this alternate control scheme.

References

Donahue, C.; Lipton, Z. C.; and McAuley, J. 2017. Dance Dance Convolution. In Precup, D.; and Teh, Y. W., eds., *Pro-*

ceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, 1039–1048. PMLR. URL http://proceedings.mlr.press/v70/donahue17a.html.

Martin, A.; Farines, J.; Paliyawan, P.; and Thawonmas, R. 2019. Dancing ICE: A Rhythm Game to Control the Amount of Movement Through Pre-Recorded Healthy Moves. In *Motion, Interaction and Games*, MIG '19. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369947. doi:10.1145/3359566.3364691. URL https://doi.org/10.1145/3359566.3364691.

Smith, G.; Treanor, M.; Whitehead, J.; and Mateas, M. 2009. Rhythm-Based Level Generation for 2D Platformers. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, 175–182. New York, NY, USA: Association for Computing Machinery. ISBN 9781605584379. doi:10.1145/1536513.1536548. URL https://doi.org/10.1145/1536513.1536548.

Summerville, A.; Guzdial, M.; Mateas, M.; and Riedl, M. 2016. Learning Player Tailored Content From Observation: Platformer Level Generation from Video Traces using LSTMs. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 12(1). URL https://ojs.aaai.org/index.php/AIIDE/article/view/12895.

Tsujino, Y.; and Yamanishi, R. 2018. Dance Dance Gradation: A Generation of Fine-Tuned Dance Charts. In Clua, E.; Roque, L.; Lugmayr, A.; and Tuomi, P., eds., *Entertainment Computing – ICEC 2018*, 175–187. Cham: Springer International Publishing. ISBN 978-3-319-99426-0.