

Chord: A Music Game

CIS 499 SENIOR PROJECT DESIGN DOCUMENT

Ted Aronson
Advisor: Steve Lane
University of Pennsylvania

PROJECT ABSTRACT

The term “music game” applies to a set of video games that incorporate music into the core gameplay. This differs from most games in that while many games *have* music, music games are about *creating* music. The way that each game deals with its respective musical elements varies widely. Several games, such as Harmonix’s Rock Band, require the player to actually “perform” the music they hear using special controllers. Others, like United Game Artist’s Rez, use elements from other genres of games to allow the player to dynamically control the background music. In both cases, the player’s actions in the game directly impact the music. Music games can be a very powerful sensory experience because of this. In this project, I will analyze the design of several major label and independent music games. I will specifically look at the styles of music included in each game, the visual styles of each game, the methods in which the player interacts with the music, and how well all three elements mesh together to create a powerful experience. I will then use this analysis as notes to design and create my own music game using OGRE, an open source graphics engine, and Fmod, a professional audio engine. In each level of my game, players will build a prerecorded song from its various instrument tracks. Each level will be broken down into several different puzzles. A puzzle will consist of an emitter that sends out a constant beat of pulses, and a receiver that the pulses travel to. The player must construct a path from the emitter to the receiver such that the pulses arrive at the receiver in a desired time pattern. This time pattern is determined by the particular MIDI track associated with the puzzle. If a pulse arrives at the receiver when a note is supposed to be played, the game plays that note. If a pulse arrives at the receiver at any other time, the note does not play. As the player completes and refines the path, the correct pattern of notes will emerge, giving the overall sensation that the player is constructing a machine and a song at the same time. This will likely be a two semester project, with the first semester’s work focusing on the core gameplay, including creating the basic level structure, the basic puzzle structure, a demo level with 4 puzzles, and a functional shell for saving progress and loading levels. The second semester will focus on refining the

visual style of the game, as visuals will also be critical to the feel of the game. Using OGRE and its HLSL capabilities, I will write pixel shaders that will help create the visual feel for the game.

Project blog:

[http:// chordgame.blogspot.com](http://chordgame.blogspot.com)

1. INTROUCTION

The term “music game” applies to a set of video games that incorporate music into the core gameplay. This differs from most games in that while many games *have* music, music games are about *creating* and *interacting with* music. The way that each game deals with its respective musical elements varies widely. Several games, such as Harmonix’s Rock Band, require the player to actually “perform” the music they hear using special controllers. Others, like United Game Artist’s Rez, use elements from other genres of games to allow the player to dynamically control the background music. In both cases, the player’s actions in the game directly impact the music. Music games can be a very powerful sensory experience because of this.

In this project, I will write up and post to my blog review-like analyses of music games. These games will include Rock Band, Audiosurf, Rez HD, Frequency, Auditorium, and Music Catch 2. I will specifically look at the styles of music included in each game, the visual styles of each game, the methods in which the player interacts with the music, and how well all three elements mesh together to create a powerful experience. With these analyses, I hope to gain a better understanding of what makes a music game critically successful. This knowledge will become useful when I am refining the game.

In each level of my game, players will build a prerecorded song, composed in MIDI format, from its various instrument tracks. Each level will be broken down into several different time-based puzzles, one for each track in the MIDI composition, up to a maximum of 5. Each puzzle will appear on the face of a large cube suspended in space. The sixth face will be a speaker that continually pulses as if it is playing the music. The player can rotate the cube and watch the pulses move around the already completed puzzles. To select a puzzle, the player will rotate the cube so the puzzle’s side is in full view of the camera and then press a button to zoom in on it.

Throughout the level, even if the player hasn’t completed a puzzle, a loop of background music (that is also part of the song) will play. The background music should be a part of the composition that provides a sense of rhythm, but should not be a drum track or the main bass track if possible.

A puzzle will consist of an emitter that sends out a constant beat of pulses, and a receiver that the pulses travel to. The player must construct a path from the emitter to

the receiver such that the pulses arrive at the receiver in a desired time pattern. This time pattern is determined by the particular MIDI track associated with the puzzle. If a pulse arrives at the receiver when a note is supposed to be played, the game plays that note. If a pulse arrives at the receiver at any other time, the note does not play. As the player completes and refines the path, the correct pattern of notes will emerge, giving the overall sensation that the player is constructing a machine and a song at the same time.

The playing field of the puzzles will be a grid, similar to the game Pipe Dream, where the players will place tiles with sections of the path on them. They can be positioned and rotated by the player. Certain levels may already have sections of path placed, either for the benefit or frustration of the player. There will also be several special tiles for the player to use. There will be tiles that split the pulse into multiple pulses, tiles that condense two or more into one, tiles that destroy pulses, and possibly tiles that quicken and slow pulses. Some or none of these will be available to the player in each puzzle.

The pulses will travel along the path with a speed relative to the tempo of the music. To make sure the speed is correct, the movement of the pulses will be directed by MIDI events generated from a tempo track associated with each level.

When the player exits a puzzle, the beats he's created keep playing. Even if the player hasn't successfully completed a puzzle, any properly aligned pulses still play at the appropriate times. This way, the player can keep track of how far he's progressed through the level simply by listening to the completeness of the music.

The game may also need an overarching competitive aspect. Once I've completed the basic game and my research on music games, I may try to add a competitive element to it.

Once I have created and refined the basic gameplay, I will focus on the visual style of the game. Since visuals are a very important component to music games, I will be incorporating my notes from my game analysis to determine what a proper visual style is for the game. Using OGRE's DirectX and HLSL capabilities, I will spend several months polishing the visual style to something that will create an intense and meaningful experience.

1.1. Significance of Problem or Production/Development Need

“It is a well known problem in society that my game doesn't exist. This project will create my game.” - Ted Aronson (placed here under duress from Grace Fong)

1.2. Technology

The game will use OGRE with DirectX as its graphics engine, and Fmod as its sound engine. Much of the gameplay will be driven by MIDI events. The game will be written almost entirely in C++, but will also use HLSL for shaders.

1.3. Design Goals

Our project will address what.

1.3.1 Target Audience.

This game is designed for video game players who enjoy music.

1.3.2 User goals and objectives

I hope that this game will combine gameplay elements, music, and visual style to give the player the feeling that they are constructing a song, and not merely playing it.

1.3.3 Project features and functionality

By the end of the project, I will have a complete vertical slice of the game. Players will be able to enter the game, interact with the main menu, load a game, select a level, play at least one demo level, and save their progress through the level.

2. Prior Work

There have been several successful video games in previous years that allow the player to interact with and create the music in the game. The most well known of these is the Guitar Hero series originally developed by Harmonix. These games were groundbreaking in the fact that they used custom peripherals and allowed players to actually feel like they were playing the music. The games involved colorful “notes” traveling toward the player, who was required to press a button and flick a switch (as if strumming a guitar) in time with the music. There were several modes to the game which placed the player either in a World Tour setting or in direct competition with another guitarist.

Rock Band, also developed by Harmonix, brought the Guitar Hero playing-through-peripherals concept to drums and vocals, allowing 4 players to cooperate and play full songs together. The drums were particularly noteworthy in that playing on the highest difficulty setting was said to be exactly like playing the original drum part.

Audiosurf, an independent game by Dylan Fitterer, took the music game concept in a completely different direction. Audiosurf allowed players to create levels automatically based on their own music. This, combined with an interesting take on lane-racers such as F-Zero, led players to feel as if they were “playing” their music. The game was a smash hit, also partially because it was the first independent to be released on Steam, a popular digital distribution service. It also won several awards, including the Independent Games Festival award for Best Audio in 2008 and PC Magazine's Game of the Year for 2008.

Auditorium is yet another critically acclaimed independent music game. Auditorium was released in 2008 by Philadelphia based group Cypher Prime. The goal of Auditorium is to pass a stream of particles, known as The Flow, over several boxes on screen. As more particles pass over a particular box, a loop of music grows louder. Each box has its own loop associated with it, so when The Flow is redirected over all the boxes on screen, an entire song plays. Auditorium won the 2008 Interactive Media Best in Class Award for Games.

3. PROJECT DEVELOPMENT APPROACH

3.1. Algorithm Details

Each puzzle will have a tempo track associated with it that consists only of steady beats. When played through Fmod, the tempo track should generate a steady rhythm of MIDI events. When each MIDI event fires, each pulse currently on the playing field will advance to the next cell. When a pulse reaches the receiver node, the receiver node will check if there is a MIDI event in the goal track at that moment, and if so, will tell Fmod to play the event.

3.2. Target Platforms

3.2.1 Hardware

PCs

3.2.2 Software

The game will be playable on Windows machines only. Eventually, I'd like to incorporate the Steamworks API and release it on Steam.

4. WORK PLAN

4.1.1. Project Milestone Report (Alpha Version)

By the Alpha Milestone, I should be able to demonstrate:

- The game loads data from a level file
- The player can interact with a level by rotating the cube and zooming in on the puzzles.
- The player can complete a puzzle, which plays the full instrument track upon completion.
- The player can save his progress through the game, which gets reloaded when the game relaunches.

By the Alpha Milestone, I do not foresee being able to demonstrate:

- The audio from multiple puzzles combined when zoomed out of a puzzle.
- More than 2 puzzles completed.
- A menu screen that the player can select levels and saved games from.
- Multiple levels.

4.1.2. Project Final Deliverables

By the end of the first semester, I should be able to demonstrate:

- A menu from which the player can load several different save files
- Multiple playable levels that can be navigated and selected from a user interface
- At least one complete level that contains 5 puzzles which contribute their audio output to the larger level's music.

With an additional semester of work, I should be able to add a polished visual style to the game, including shaders and personally created artwork.

4.1.3 Project timeline.

I'll be building this game from the ground up, focusing on the music playback first, followed by the puzzles, followed by the levels, followed by a shell to wrap the whole thing up into a game.

4.1.4 Gant Chart

Chord										
Ted Aronson										
Project Lead: Ted Aronson										
Today's Date: ### (Sun) (vertical red line)										
Start Date: 9/19/2009 (Sat) First Day of Week (Sun=1): 2										
WBS	Tasks	Task Lead	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining	
1	Music Playback	Ted Aronson	9/19/09	10/01/09	12	0%	9	0	12	
1.1	Fmod ramp-up		9/19/09	9/21/09	2	0%	1	0	2	
	OGRE/Fmod									
1.2	Integration		9/19/09	9/23/09	4	0%	3	0	4	
1.3	Play basic MIDI file		9/23/09	9/28/09	5	0%	4	0	5	
	Play multiple MIDI files simultaneously									
1.4			9/28/09	10/01/09	3	0%	4	0	3	
2	Puzzles	Ted Aronson	10/01/09	11/03/09	33	0%	24	0	33	
2.1	Mouse placing pieces		10/01/09	10/09/09	8	0%	7	0	8	
	Timing track moves									
2.2	single pulse		10/09/09	10/19/09	10	0%	7	0	10	
	Pulse generator creates stream of pulses									
2.3	Receiver node		10/19/09	10/22/09	3	0%	4	0	3	
2.4	collecting pulses		10/22/09	10/24/09	2	0%	2	0	2	
2.5	Event playback for correct pulses		10/24/09	10/28/09	4	0%	3	0	4	
2.6	Special sections		10/28/09	11/03/09	6	0%	5	0	6	
3	Levels	Ted Aronson	11/03/09	11/17/09	14	0%	11	0	14	
	Display multiple puzzles around a cube									
3.1			11/03/09	11/06/09	3	0%	4	0	3	
3.2	Puzzle selection		11/06/09	11/10/09	4	0%	3	0	4	
	Play audio from all completed puzzles									
3.3			11/10/09	11/17/09	7	0%	6	0	7	
4	Shell	Ted Aronson	11/17/09	12/06/09	19	0%	14	0	19	
4.1	Switching levels		11/17/09	11/21/09	4	0%	4	0	4	
4.2	Main Menu		11/21/09	11/25/09	4	0%	3	0	4	
4.3	Loading Screen		11/25/09	11/29/09	4	0%	3	0	4	
4.4	Score Screen?		11/29/09	12/06/09	7	0%	5	0	7	

5. REFERENCES

I'll be using my design notes from the following games:

Rock Band
 Rez HD
 Auditorium
 Audiosurf
 Donkey Konga
 Music Catch

I will also be reading the tutorials on the Fmod and OGRE wikis to learn about how to properly use them.

