

Children of the Shred: Video Games with Analog Instruments

Clay Ewing
Parsons MFA DT Candidate
Brooklyn, NY
clay@etherealcereal.com

Bruce Drummond
Parsons MFA DT Candidate
Brooklyn, NY
brucedrummond@gmail.com

ABSTRACT

The impetus of this project is to merge two domains: playing a musical instrument and gaming in a virtual environment. Many efforts have been made to allow musicians to improvise over the internet either asynchronously like Virtual Rock Band or in real time, such as Ninjam. These applications focus on sound and music production but lack a participatory visual feedback. The rise of rhythm based games such as Rock Band and Guitar Hero have ushered in a wave of enthusiasm for playing instruments. The downside to these games is that the only connection the controllers have to an analog instrument is their shape. The goal is to create a multiplayer network game that can be controlled by playing an analog instrument. The frequency of sound coming from the instrument will be analyzed and converted into commands in the game. As notes are played, the other player will have to respond by playing the same notes or be penalized. The result is a generative virtual environment where people can play against each other as a game or together as a performance.

1. INTRODUCTION

Children of the Shred is a real-time computer game in which two players can challenge each other's music improvisational skills by playing out a narrative adventure with digital or analog instruments. The two players have the choice to be either the zombie slayer or the zombie lord. According to their character selection, sprites (either zombies or slayers) are created that correspond to the notes that are played. These note specific zombies and slayers can only kill their same note.

While there are a wide variety of musical applications for computers along with a myriad of video games, there is not a video game that a user can play with a real instrument, such as a guitar. Rhythm based games, such as Rock Band and Guitar Hero, use mock instrument controllers to follow along with the general rhythm of the song. Software such as Ninjam and eAUDiiO Jamming allow for online music collaboration yet are strictly tools for creation and not a game. There is no reason why a user playing a video game with a guitar controller shouldn't be able to transfer those skills off the screen and into the real world. The main goal of Children of the Shred is to take a guitar and map the frequencies it produces into controls for a video game. It is also important to factor in the use of the internet for remote multiplayer gaming. As the internet becomes more ubiquitous, it's becoming an inherent requirement that any new technology should be able to connect to a larger system. This game can then act as a medium for learning to play the guitar, extend pre-existing skills, as well as give musicians a new outlet for creating music whether locally or remotely.

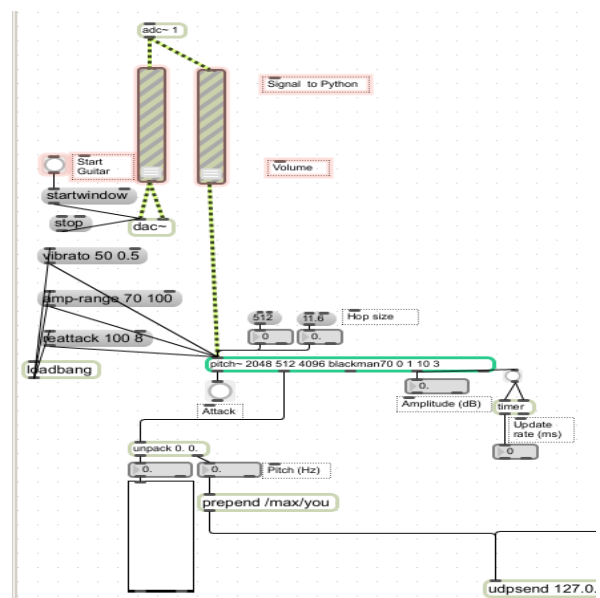
2. DESIGN AND PROTOTYPING

The design of this video game was a three part process. The first part involved turning an audio signal into something that could control gameplay, which was achieved by using Max/MSP/Jitter. The second part was receiving this newly created data in a video game environment in Python. The third part of the process was building the actual video game in Python along with the PyGame module.

2.1 Frequency Analysis

In order to turn an analog instrument into a game controller, we turned to Max/MSP/Jitter to analyze the incoming signal frequency. Max/MSP/Jitter is a graphical development environment for music and multimedia which is developed by Cycling '74 [4]. The frequency analysis was simplified by using a third party object, pitch~, created by Tristan Jehan at the MIT Media Lab. Each time the audio signal reached a specified peak in amplitude, the frequency was then broadcast using an OSC formatted UDP packet. OSC, or OpenSound Control, is a protocol for communication among computers, sound synthesizers, and other multimedia devices that are optimized for modern networking technology [5]. These packets could then be picked up by the Python program to be used for game control.

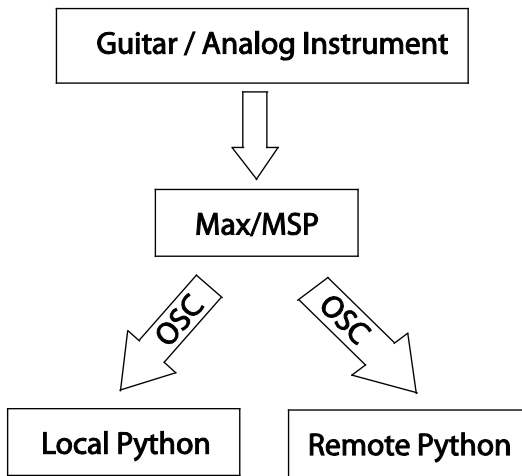
Figure 1. Guitar Translator Max Patch



2.2 Sending and Receiving Packet Information in Python from Max/MSP/Jitter

To receive a UDP broadcast in Python, a virtual port must be opened. This virtual port must be open and in a listening state the entire time the game is being played. To achieve a multiplayer environment, the Max/MSP/Jitter program sends the OSC packet with frequency information to two separate addresses. The first packet is sent to the local computer with an address of “/max/you” while a second packet is broadcast to the second player’s computer using the address “/max/opponent.” This allows Python to discern between which user is playing a note.

Figure 2. System Diagram for Processing an Audio Signal



After the frequency of the note is received, Python compares that number to a list of values that correspond to the frequency of notes that can be played by a guitar. This information is then used to generate the sprites for each player.

2.3 Game Design

With so many games involving musical instruments gearing the theme towards playing music, there was a strong desire to move away from this trend. As zombies are a well recognized and widely appreciated phenomenon in popular culture, the thought of using them as a theme felt like a good way to break away from the usual routine. The science fiction and horror theme also allow greater license to expand the game in the future as a more realistic game would require much more explanation in terms of narrative during game play [1].

Figure 3. Zombie Sprites

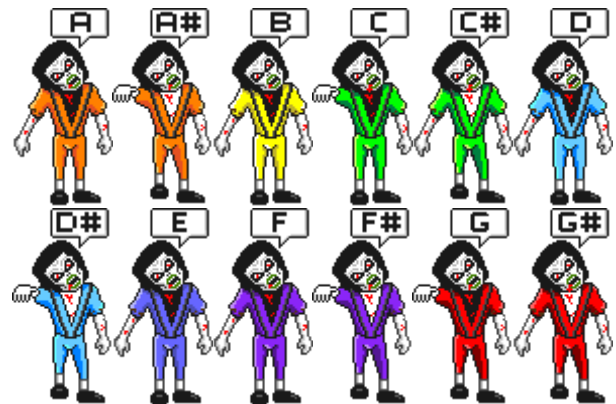
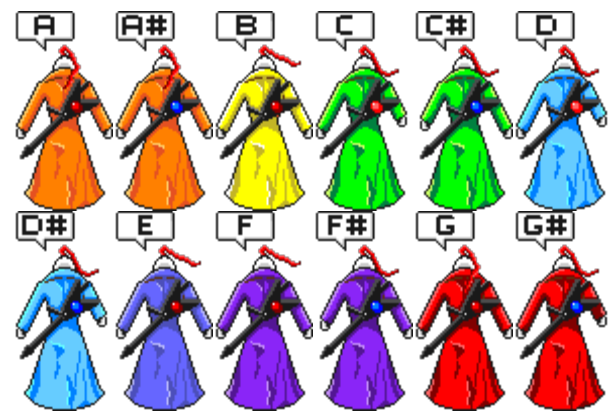


Figure 4. Slayer Sprites



The game board is setup with the zombie lord occupying the top of the screen, with the zombie slayer at the bottom of the screen. As notes are played on the guitar, zombies and slayers appear on the screen and move towards the opponent’s side. When the sprites collide, if the notes are the same, the location of the sprite is analyzed. If the sprite is destroyed on the opponent’s side, the player gets more points. If the sprite is destroyed on the player’s side, the opponent gets more points. The idea is that the faster a person responds with the correct note, the more they will be rewarded. This back and forth lasts for a total of 3 minutes and then the game is over.

The decision to use Python as a programming environment came down to three points: cross platform compatibility, a wide range of modules, and simplicity of code. As Python compiles at runtime, there is no need to create a version of the software for each operating system. The modules available for use with Python grow every day, but specifically PyGame (a module for game development) and SimpleOSC (for receiving and sending OSC packets) made the project a lot less daunting.

Since the target audience for the game was casual gamers and casual musicians, we decided to focus on socializers. Socializers are people for whom the greatest reward is interacting with other people, through the medium of the virtual world [1]. Through playing music and killing zombies, the socializers will have an outlet to interact with their friends in a virtual world. Therefore the game play should be simple and not overly complex. As this

game's main purpose is interacting with another player, the socializer should enjoy playing this game.

2.3.1 Story Boards

In order to get a clear idea of how the game would be played out, story boards were created. Initially, the options screen would present the player with the choice of two modes of play (battle and performance) as well as a location. The location would function as purely a backdrop during the battle.

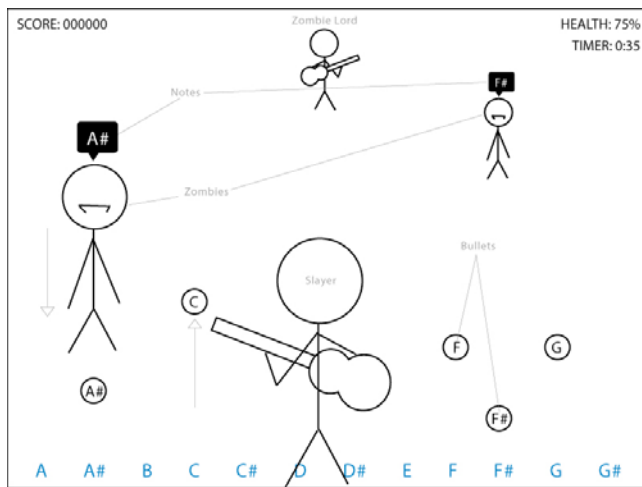
Figure 5. Options Screen for Storyboard

OPTIONS

- BATTLE**
- PERFORM
- CHOOSE A LOCATION
- QUIT

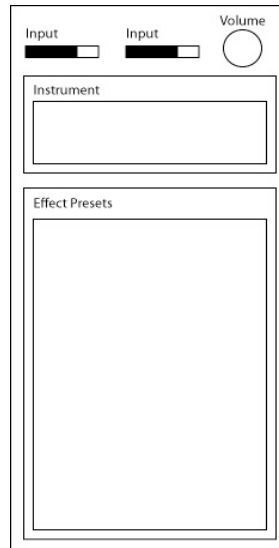
The battle screen shows both the *Zombie Lord* and the *Zombie Slayer* on opposite sides of the screen. The sprites are released from the top and bottom of the screen and get closer as they get closer to the opposite player. Bullets would be shot out to kill the approaching zombies, at a specific location based on the note played. If the zombies or bullets get past the bottom or top of the screen, the opponent's health would be affected negatively. If the health wasn't diminished completely by a certain point, the timer would run out and cause the game to end.

Figure 7. Battle Screen for Storyboard



The instrument control panel would be contained as a Max/MSP/Jitter runtime. This program would allow for adjusting the input level of the incoming instrument, overall output volume, instrument type selection and different effects presets.

Figure 6. Instrument Control Panel for Story Board



2.3.2 First Prototype

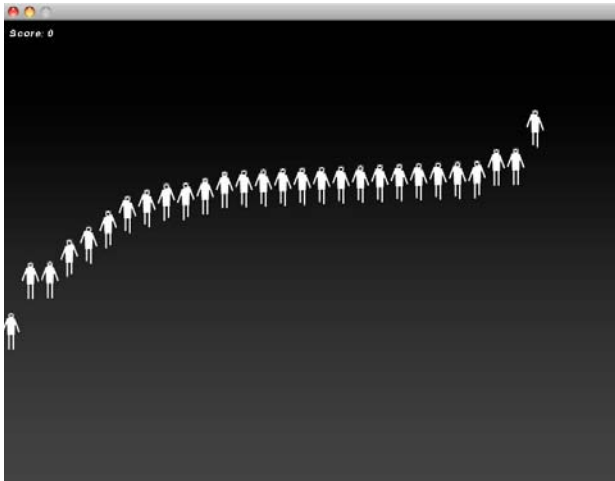
Initially, the game was a free for all, with players being allowed to play whatever they wanted whenever they wanted. The size of the sprites in the first version also allowed for 32 of them per row on the screen. Each note was differentiated by a different color sprite. The amount and size of sprites proved to be particularly troublesome as the screen could very quickly fill up with vast quantities of zombies and slayers.

The concept of having an actual player on either side of the screen was dropped as it took up too much valuable screen space. The idea of introducing a 3D perspective was also dismissed as one sprite could eventually crowd the entire screen and ruin the flow of the game.

By allowing both players to create sprites at the same time, another problem was brought to light: there was no way to specify the location the zombie would be generated at without perfect timing. Even if they managed to get their timing down properly, there were too many choices of notes coming at the player. Any meaningful game play was ruined by the chaos that ensued on the screen.

The decision was also made to limit the amount of notes recognized by the game. While a standard guitar is a 6x24 matrix, resulting in numerous possibilities for notes in many different octaves, the game treats each octave the same. This way, if an E is played on the low string or high string, it sees it as the same thing. Therefore, the game was limited to 12 possible notes (C, C#, D, D#, E, F, F#, G, G#, A, A#, and B).

Figure 7. First Prototype Screenshot



2.3.3 Second Prototype

To tone down the amount of chaos, a few new rules were implemented into the logic of the game. The sprites were enlarged 400% so that it took only 8 of them to fill up a row of the screen. A turn based system was also added so that each player was allowed 8 notes and then cut off until the opposite player responded with their 8 note turn.

With fewer sprites on the screen, the game became more visually digestible. The enlargement of sprites also allowed for the name of the note to be placed to the side of the sprite which created an additional visual cue to tell the player what note to play. By setting the number of notes to be played to the same size as a row of sprites, it became a lot easier to decide what note should be played as compared to the first version.

Blood spatter was also added when the opposing sprites collided.

Figure 8. Blood Spatter Graphic



2.3.4 Third Prototype

With the basic game play taking shape, the focus of the third prototype turned to semantics. An overall game timer was added so that each game would last 3 minutes.

Figure 9. Third Prototype Screenshot



A feedback system was also added to encourage or taunt the player or opponent when a certain number of correct notes had been played.

Figure 10. Positive Feedback Screenshot



3. METHODOLOGY

3.1 Concept

The idea of music making video games is not new. The push for real time online music collaboration has been chased somewhat unsuccessfully for a long time. However, after surveying all that is available in these realms, there are very few options for learning to play an instrument and none that allow to people to improvise on an instrument and play against each other in a game at the same time.

The impetus of this project is to create a virtual environment that allows music creation, improvisation, and skill development. A game that only allows a certain set of commands in a specific order will eventually become boring. A game in which a player can walk away with a new found skill is not only useful, but exciting. If a video game player, like Daniel Piacampo, who is considered one of the best Guitar Hero players in the world [7] was learning the guitar while mastering a video game, he could become a great musician. The possibilities for learning through

game play present a wide variety of opportunities in many fields, but this project the sole focus is on musical instruments. Furthermore, the computer is an ideal medium to challenge the musical improvisational skills of two people. As a computers can automate complicated procedures [2] the task of identifying each note played and comparing it to another requires no effort on the player's part. If an analog version of an improvisational music game was created, the referee would have to have perfect pitch, as well as the ability to count every individual note played by a person.

3.2 Previous Explorations

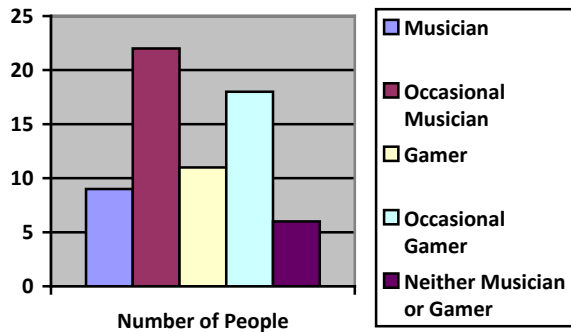
As casual musicians ourselves, we fall into the category of not having a regular group of people to make music with. That being said, anything musically oriented is appealing. The rhythm based games are very fun but offer no way to improve our actual musical abilities.

For the past few months, our explorations with Max/MSP/Jitter have led us down two paths: the endless possibilities for controllers, and the flexibility of OpenSound Control. Without Max/MSP/Jitter, the task of analyzing frequencies would seem very daunting but with the tools available today, it's not farfetched at all.

3.3 User Surveys

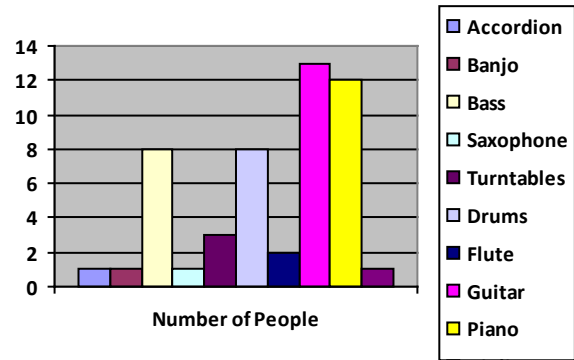
A simple online survey was used to gather responses regarding online music collaboration and musical abilities. 54 surveys were completed, with most of the responses coming from our target audience of casual musicians and casual gamers.

Figure 1. Respondent Classification



The guitar was the most played instrument, followed closely by the piano. The data seems to show that the majority of people with musical skills gravitate towards an instrument that could be included in a typical 4 piece ensemble (guitar, piano, bass, and drums).

Figure 11. Types of Instruments Played



The most encouraging piece of data from the survey was that about 40% of non-musicians had the desire to jam.

Table 1. Desire to Jam (non-Musicians)

Response	Number of People
Yes	22
No	23

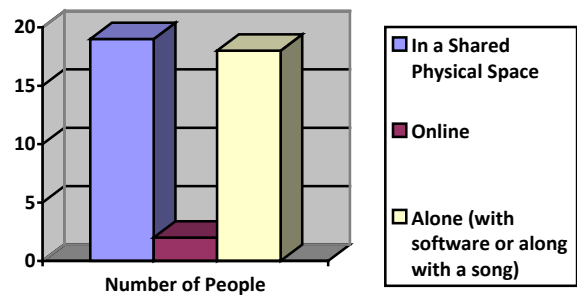
Less than half of our respondents said that they jammed. This was expected as our target audience was casual musicians and gamers, so the likelihood that every respondent would have a group of people to make music with was low.

Table 2. Number of Respondents That Jammed

Response	Number of People
Yes	20
No	34

The respondents seemed to prefer jamming by themselves or with a group of people in a shared physical space. Hardly anyone jammed using a network.

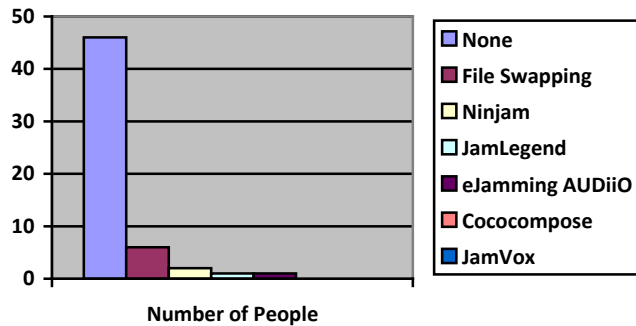
Figure 12. How People Jam



The most surprising result of the survey was that even though there are many venues for online music collaboration, they are hardly ever used. As many people jam to songs alone, this gives

weight to the idea that people would use online methods for jamming if they were widely known and reliable.

Figure 13. Use of Online Music Collaboration Tools



4. ANALYSIS

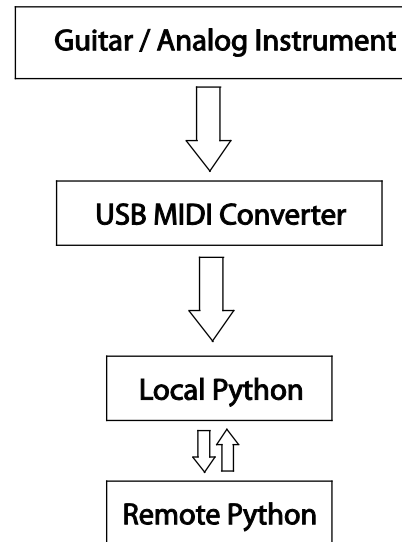
4.1 Using an Instrument as a Controller

The biggest problem of using an instrument like a guitar for frequency analysis is that each note played produces more than one frequency. Even when the guitar is tuned properly, a controller that works 95% of the time ends up being rather unreliable for a game where time matters. The other drawback is that chords are especially hard to decipher as six notes are being played at the exact same time.

Max/MSP/Jitter is a great environment to build rapid prototypes, but the reliability was not at a level where the average person could use our patch as a bridge. Using software to decipher everything, in the end, is not the best method and would be avoided in future iterations.

When a keyboard was used to play the game, the note detection was flawless. The reliability of the controller made the game much more enjoyable. In the future, a guitar with a hexaphonic MIDI pickup would be the ideal solution as note detection would be extremely accurate. This would also reduce a layer of complexity to the code, as much of the processing would be done on the hardware level.

Figure 14. Revised System Diagram for Processing an Audio Signal



4.2 Networked Game Play

Perhaps the most ambitious portion of this game was allowing for networked play. There are two issues with this setup and they both deal with latency: audio signals and note detection.

Even through a local area network, the latency of playing a note to showing up on the screen at times was around 80 milliseconds. While this is a very short period of time, for a musician to play to their full ability, the lag must be non-existent. If a note doesn't show up on the screen as soon as it's played it can very easily throw the player into confusion if the note was picked up by the computer or not.

While UDP broadcasts of frequencies being played are very small packets, sending an entire audio signal over the internet requires much more bandwidth. The latency was so bad when trying to stream audio that it didn't survive the first prototype. One way that this could be avoided might be to send the other player's commands across the network as MIDI notes. The lag wouldn't be as noticeable as it is the other player's instrument that needs to be heard. This would definitely be acceptable in a tutor mode of play, not as much in a performance setting.

4.3 Game Design

Throughout the user testing, the feedback on the game was overwhelmingly positive. People seemed to respond very well to the idea of playing an instrument and battling zombies at the same time. Many people were excited to play a game that used a real instrument instead of a control that just looks like one.

The biggest improvement to the game would involve creating a one player version. This version could include a tutor mode that would allow the player to slowly learn to play the guitar. Beginners would also find tablature notation helpful at the bottom of the screen.

Another problem encountered once turns were implemented was that a player couldn't tell if what they were playing was being

interpreted correctly by the computer. This could be fixed by adding an in-game tuner that would show the player what note the computer is picking up each time the guitar is played.

Another interface that could be developed would be anchoring notes to a specific portion of the screen and eliminating turn based play. While this would allow the screen to fill up relatively fast, the players wouldn't be locked into a call and response type of play.

The last issue with the game play is the scoring method. Having a score at the top of the screen didn't seem to grab a player's attention the way it was intended to. Most likely, a health meter would work better.

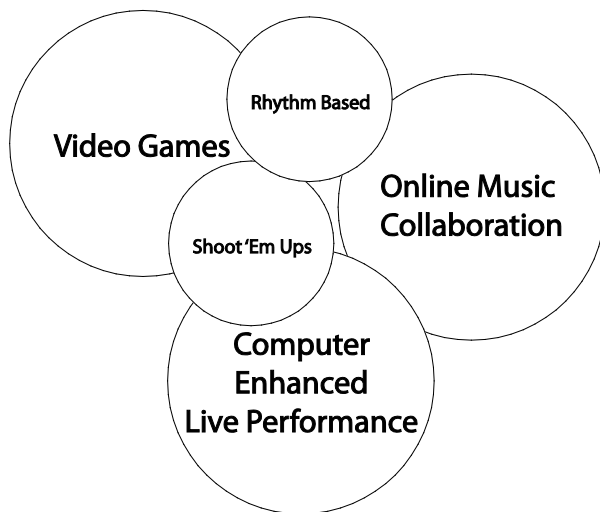
Figure 15. Future Battle Screen



5. RESEARCH

The domains of this project include video games (specifically rhythm based and shoot 'em ups), online music collaboration, and computer enhanced live performance.

Figure 16. Domain Map



5.1 Video Games

There are many different video games that have inspired Children of the Shred. Each game has its pros and cons, so an overview of titles is appropriate.

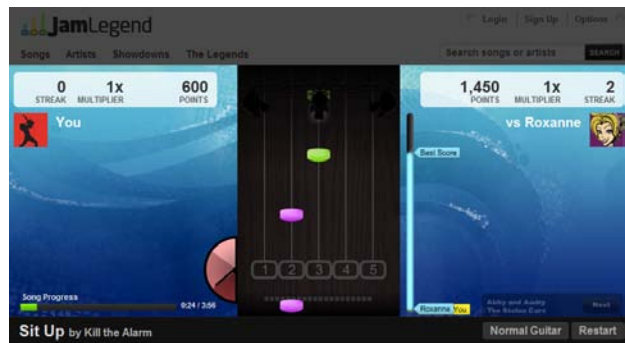
Rock Band is a console based video game where players use controllers that look like instruments to perform songs. Players have the option of using a guitar, bass, drums, or vocals. Each controller (aside from the microphone for vocals) has 5 buttons that can be pushed. These buttons are mapped to the song in a way that allows the player to feel as if they're playing a long with the song because of the rhythmic arrangement. While this game is very fun and entertaining, it does not use real instruments. Without a real instrument as a control, the only skill that can be improved is pushing buttons rhythmically.

Figure 17. Rock Band Screenshot



JamLegend is another rhythm based video game for the computer that uses a web interface. Unlike Rock Band, users of JamLegend can upload their own songs to play along with or to challenge others with. The biggest drawback to JamLegend is that instead of an instrument like controller, users are required to use their keyboard to play. [jamlegend website] The ability to play online and upload your own songs is a great addition but the fact that the user is limited to their keyboard as an input device is a severe drawback.

Figure 17. JamLegend Screenshot



Typing of the Dead is a 3D shoot 'em up where players have to type the word on the screen in order to shoot the approaching zombies. The faster the player types the word, the higher the score they receive. This game achieves everything it sets out to do: teach people to type and make it entertaining.

Figure 18. Typing of the Dead Screenshot



5.1 Software for Music Collaboration

Numerous attempts have been made at creating systems for networked music collaboration, whether real time or asynchronous. Some of these services cost money while others are free.

Ninjam is an online public collaboration room that subscribers can access for an hour a day. Due to bandwidth issues, the others hear and play with everything precisely one bar after the musician plays it [6]. The whole system is cost prohibitive as a musician either has to pay hourly to use public servers, or pay a lot more to setup their own server. Furthermore, the streaming audio is not at a high sample rate and there is no visual feedback for what is going on.

JamVox is a jam and practice tool for guitar players. Users can load their favorite song and the guitar track will be muted so that they can play the part themselves. The software also has a built in effects unit, which is important considering most people plugging an instrument into a computer don't process their signal afterwards. However, there is no network functionality to connect to other musicians.

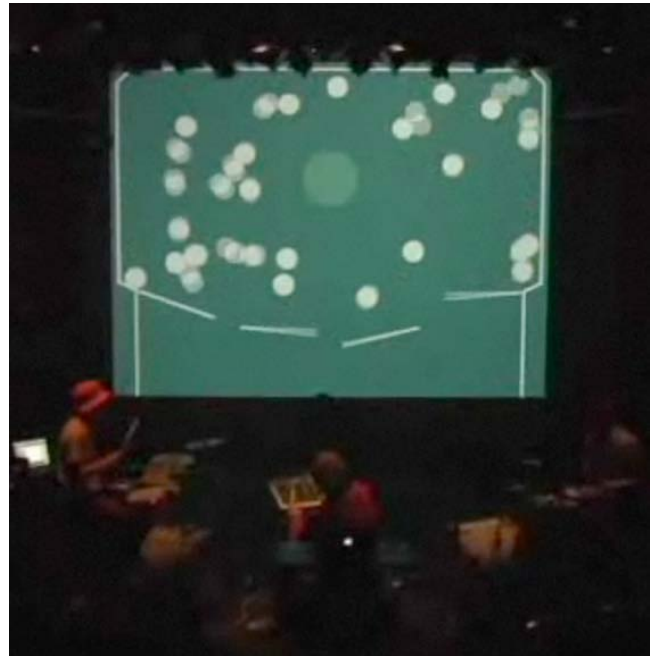
Virtual Rock Band is a community driven website where musicians can meet up and swap their tracks. While this process allows for online collaboration, the users never get to experience playing along with someone else in real time. There is also the possibility that no one will listen to your tracks and your jamming desires will never be remotely filled.

eJamming AUDiiO is software that enables musicians to collaborate and record music online. eJamming also produces software that can broadcast these sessions in real time. While the service is still in beta, it is highly promising. [3]

5.2 Computer Enhanced Live Performance

d.v.d: Watch the Beat of the Drums is an interactive performance where two drummers play and create visualizations generated by the sound of the drums. The end result is a game of pinball being played in front of an audience. People who watch this performance are listening to original music and seeing a video game being played at the same time. A limitation of this is that the game is really just a visualization and there is a lack of variety on the screen.

Figure x. Watch the Beat of the Drums Performance



6. CONCLUSION

Games using real instruments as controllers are the next logical step for musical video games. They will help educate a mass audience about music and music theory. Beyond basic learning, these games can also help musicians practice and improve their improvisational skills. The performative nature of people playing instruments to control a game also offers a new venue for musicians to showcase their work. All of these pieces are part of a larger puzzle in gaming and music that represent a growing and fascinating field of study.

7. REFERENCE

- [1] Bartle, Richard A. Designing Virtual Worlds, New Riders Publishing, 2003.
- [2] Salen, Katie and Eric Zimmerman, Rules of Play: Game Design Fundamentals, MIT Press, 2004.
- [3] DEMO Innovators Achieve Outstanding Success; A Round Up of Early 2007 DEMO Demonstrator News. (16 May). *PR Newswire*,***[1]***. Retrieved December 17, 2008, from ProQuest Newsstand database. (Document ID: 1271649911).
- [4] "Max (software)" in Wikipedia. 2008. http://en.wikipedia.org/wiki/Max_MSP

- [5] "OpenSound Control" in Wikipedia. 2008.
http://en.wikipedia.org/wiki/OpenSound_Control
- [6] Gregory Walker (2008, January). Going
Global. *Strings*, 22(6), 22,24-25,8. Retrieved December 17,

- 2008, from Research Library database. (Document
ID: 1396235961).
- [7] Colin Campbell (2008, November). UNLEASHING
GUITAR HERO. *Maclean's*, 121(46), 26-27. Retrieved
December 17, 2008, from ABI/INFORM Global database.
(Document ID: 1603285121).