

Thesis: research brief

I did much of the research for my thesis before ever imagining I would produce a thesis. I did this by playing video games. Take that, academia!

I am studying a) video games as systems to promote literacy and b) alternative forms of musical expression, in order to develop a new form of music-based gameplay that transforms the player's comprehension of musical notation.

The first domain might otherwise be stated as follows: the potential for games as representational, interactive systems to educate the player. This is the thinking advocated by James Gee in his groundbreaking book What Video Games Have to Teach Us About Learning and Literacy, some of which I read for my "Game Studies" class. The book has little or nothing to do with the conventional thinking and methodologies attached to the (unfortunately stigmatized) phrase "educational games." Rather than the stale, often poorly-designed games created to teach the content of the classroom, Gee discusses the most popular and commercial games, which have no educational agenda. His insight is to separate out the process of learning from the content, and to suggest that players do learn from the games they play, in ways (and including content) that are valuable.

He gives the example of a game called "Deus Ex," a deeply-involved role-playing game, which he finds almost impenetrable, but which his very young relative is able to learn with ease. Somehow this very complex game teaches its young audience its intricacies in such an engaging way that the player not only learns it all but actually claims to be having fun! This is similar to another example he gives about how children playing the Pokemon Trading Card game regularly do high-level math as a part of playing the game, though they struggle with lower-level material presented in the usual classroom fashion. Somehow games construct a system and

give the players agency such that they are able to figure out the rules of the system and then put them to use, often much more successfully than in non-game contexts.

Building on a similar theme is Ian Bogost's Persuasive Games (also a "Game Studies" reading). Bogost discusses the concept of "procedural rhetoric," a way to use interactive, procedurally-based software to simulate systems and promote understanding through the player's interaction with those systems--a perfect match for games. As the term "rhetoric" implies (and the book's title makes explicit), the focus is on making an argument of some kind, yet the process of education through interaction with a system is consistent with Gee's theory.

It's worth noting that I didn't have the theory for this domain until I read those books, well after I had started developing my thesis ideas. I was already in the right country, though, I just didn't speak the language. This is thanks to the direction provided by my other domain, alternative forms of musical expression. This domain encompasses many different ideas and approaches, from alternative musical notation that promotes greater readability or the creation of new types of sounds, to the work we did in our recent collaboration with Miami's New World Symphony, in which we developed software to provide accompanying visuals for a classical music concert. My interest being games, I focused specifically on different ways to experience and interpret music through gameplay, rather than simply listening to it passively or making it react to the play in an emotive way. I didn't want a dynamic soundtrack that would change tone as the action heats up; I wanted the music itself to *be* in the gameplay, in a way that had never been done before.

Here's where my history with game-playing turns out, with blissful serendipity, to have been research all along. I'm a big fan of "music games" (which are popular, numerous and varied enough to constitute a genre) and I've played many of them. Enough of them not only to have developed favorites but also to see their limitations. While they all do attempt to synthesize

music and gameplay, none of them go far enough to satisfy me, though they fall short in different ways. None of them really makes the music essential to the play. It is valuable to consider how this is true, and so I thought upon all the music games that I had played or learned about and discovered I could divide them into four rough categories, grouped by the way the gameplay relates to the music. These four approaches represented the history of the genre and the modes that I would need to transcend for my project to succeed, though I would likely end up making use of all of them in that venture.

These categories (in ascending order of musicality) are--Mimicry: rhythm/dexterity; Mimicry: pitch; Synesthesia/sync; Music manipulation/creation. They often overlap, but each has its own specific focus. I proceed to give examples of each, with analysis of the categories' strengths and weaknesses and the individual games' successes and failures.

Category 1--Mimicry: rhythm/dexterity

This first category focuses on mimicry with an emphasis on rhythm and dexterity. For the quintessential (though certainly not first) example, consider the video game *Guitar Hero*, which originally came out for the PlayStation 2. It looks like this:



Fig 1. Gameplay from Guitar Hero for PlayStation 2.

Essentially, *Guitar Hero* is about pressing the correct button at the correct time. You could say this to some degree about all video games, but games like *Guitar Hero* are unique in that there is never any flexibility about when is the right time and which is the right button, and the timing is dependent upon an accompanying music track. Hence the common label “music/rhythm game.”

In the case of *Guitar Hero*, notes scroll down the screen toward the player in five columns, with each column corresponding to a different button on the controller. The player has to press the right button when a note reaches the bottom of the screen. If the player makes enough mistakes, the game ends preemptively. Getting to the sequence’s end wins the level.

Put this way, the play is very abstract and extremely banal: a series of prompts to press specific buttons at specific times. The game transcends this by synchronizing the prompts to a music track in such a way that they, in a very loose sense, “become” the notes of the music. Pressing the right buttons at the right time, as cued visually, represents “playing” the “note.” Furthermore, the sequence of the note-prompts is spatialized in a way that conforms to the progression of the music, so that a note on the left is lower in pitch than a note on the right, and a simple progression up four notes is represented as the sequence shown in Fig 1. There are more than five notes in the musical spectrum, so there is not a one-to-one locational relationship (in the way of actual sheet music), but the spatial positioning of the notes relative to one another serves as a useful metaphor for the general trend of what is happening in the music. The effect--a not insignificant one--is that, instead of feeling engaged in an arbitrary session of Simon Says, the player feels as though he or she is “playing along” to the music--indeed, generating it, since the music cuts out when the player makes a mistake. Each note sequence is a specific song, which the player wins by playing without too many errors.

This impression--moving from simply keeping time along with music to feeling

responsible for producing it—is present more or less strongly in every music/rhythm game. As much as *Guitar Hero* is about good rhythmic sense and dexterity and enjoying the music, it is also about living a fantasy of rockstardom. Look again at the shot of gameplay in Fig 1. The stripe on which the actual play occurs occupies perhaps a quarter of the entire screen. The rest of it is devoted to content which has nothing to do with the gameplay, but everything to do with the aesthetic. The stylish musicians (including the guitarist, representing the player), the stage, the fancy equipment, and the dramatic lighting all contribute, as do the screaming crowds who are not pictured in that shot but to whom the camera constantly cuts. Indeed, the failure mechanic is tied to the crowd’s disappointment, such that the worse the player does, the less he or she “rocks” in the crowd’s eyes (note the “rock” meter on the right of the stripe) and the more displeasure they show. The music in the game is drawn from the popular canon, songs the player is sure to recognize and want to rock out to. Finally—and by far most importantly—the player plays the game using this:



Fig 2. Guitar Hero’s guitar controller.

This is a “guitar controller,” i.e. a regular controller built in the shape (and with some of the affordances) of a real guitar, at smaller scale. Compare that with this, the traditional controller:



Fig 3. The PlayStation 2 controller.

Which of these would you rather play the game with? Because you can play it with either. I deliberately avoided mentioning the guitar controller until now, because from a gameplay perspective it is completely inessential. Buttons are buttons; the shape of the controller doesn't matter. But from an *aesthetic* and emotionally-engaging point of view, the guitar controller is crucial. The colored buttons, which match up with the note-prompts in the game, are on the neck of the guitar, just like the frets to play real notes on a real guitar. To play a note in the game, the player must press the appropriate neck button and flick the "strum bar" in the center of the guitar's base. This is analogous, conceptually and spatially, to fretting a note with one hand on a real guitar and strumming the string with the other hand to play the note.

So what happens when I play *Guitar Hero*? I plug my guitar controller into the game console, load a song, watch for the button prompts (sort of like sheet music or guitar tabs), and then I fret the notes with one hand, strum with the other hand, and play the popular music of my era on my guitar as my avatar swaggers around a stage in front of screaming fans who boo or cheer me depending on how well I play, and if I do really well I swing the guitar neck upward to

trigger “star power,” resulting in spectacular visual effects on-screen and a huge point boost as long as I continue to play well (not to mention how cool I feel playing my guitar in such an impractical pose). I feel like a rock star.

This is something the regular controller cannot match. Yes, one can play *Guitar Hero*, and play it well, without using the guitar controller, and a lot of the illusion is still present, with the challenging gameplay fully present and as rewarding as ever. But the high of feeling like I really am the one playing that guitar and producing that incredible music and chewing up the stage in front of people who came just to see me, is only possible by using the special controller to complete the metaphor, because it adds to the on-screen representational rhetoric a very convincing real-world physicality. Does *Guitar Hero* teach you how to play guitar? That’s debatable. Does playing it with the guitar controller feel sort of like playing guitar? Yes it does, and that’s extremely powerful, which is why the franchise was *incredibly* successful.

In this light it’s easier to see that *Guitar Hero*’s gameplay is not really about music after all. The entire musical aspect of the game is an illusion. Pressing buttons rhythmically is not inherently musical, except insofar as music is a ready source of rhythm. The music in *Guitar Hero* serves as a useful analogy for making that rhythmic button-pressing easier and more intuitive. More importantly, it serves the illusion of being a rock star, which is the real point. It’s called “Guitar Hero,” after all. The agenda is clear in the name.

You can play *Guitar Hero* with a regular controller, and you can play it with the sound off. Is it as much fun? No, it is not. In fact it’s pretty dull, and quite difficult since the music is not there to cue the rhythm. But you can play it. And the fact that you can play it without the music demonstrates the irrelevance of the music to the core gameplay.

Let’s look at a few other examples of this category. Here is another favorite of mine, *Elite Beat Agents* (released for the Nintendo DS):



Fig 4. Gameplay from Elite Beat Agents for Nintendo DS.

Words cannot describe how much I love this incomparably bizarre game. You play as the “Elite Beat Agents,” a group of male cheerleaders (modeled after the Japanese Ōendan) who fix the world’s problems through dance. When there is a problem, the EBA show up and they dance until everything is better.

The role of the player is to help them do this by using the DS stylus to tap, drag and scratch on the DS’s touchscreen in rhythm with pop songs. Very similar to *Guitar Hero*, except that in this case there’s no guise of actually playing the music, no mapping of the interactions

onto the musical progression. It's pure rhythm. The timing is synchronized to the music, so taps often match with percussive beats or other identifiable "moments" in the song, and so on with the other interactions. Just as with *Guitar Hero*, the music is there to be enjoyed and to make it easier to play the game, but you can play EBA with the sound off, too.

Here's one more, called *Amplitude*, also for the PlayStation 2:



Fig 5. Gameplay from *Amplitude* for PlayStation 2.

This has some visual similarity to *Guitar Hero*, which is not surprising because the same company made both games. In *Amplitude*, the player controls a ship that races down one of several tracks. Each track functions like the stripe (i.e. guitar neck / fretboard) in *Guitar Hero*, in that notes come down the track toward the player in several columns. When a note reaches the player, the player "shoots" it with the appropriate one of the ship's three lasers, each mapped onto a different button on the controller.

What really differentiates *Amplitude* from *Guitar Hero* is that each of the several

“racetracks” maps to a specific “track” (i.e. instrument) within the piece of music playing. For instance, one track might be for drums, another for vocals, another for bass, etc. The goal of the game is to keep as much of the song playing simultaneously as possible, which means the player must constantly be moving the spaceship between tracks, hitting enough correct “notes” to keep the track in play for a while then moving to the next track. The notes are mapped out according to the progression (with the same degree of metaphor as in *Guitar Hero*) and it can get extremely difficult, especially with the drums. In this case not having a unified focus (a single track) or a metaphor-supporting special controller really ramps up the difficulty and makes the game more about the gameplay than about the conceit. Still, the gameplay is about pure rhythm.

I call this category “mimicry” because, though the details of the implementation vary, the gameplay revolves around mimicking a musical performance, through some kind of abstracted interaction that maps in a physical way onto the music. Everything about the performance is fixed—there is no way to change the notes of the music itself, and performing anything other than the prescribed rhythm results in failure. Furthermore, the player only has real control over the rhythm of her actions. Consequently there’s no real musicality. I would need to move beyond an emphasis on Simon-Says-esque rhythm-based gameplay toward something more creative.

Category 2--Mimicry: pitch

This category is very similar to the previous one and requires little additional discussion other than an example. Here is another PS2 game, *Karaoke Revolution*:



Fig 6. Gameplay from Karaoke Revolution for PlayStation 2.

In this case we see play for two people simultaneously, but if you focus on a single track the similarities to the previous games are probably apparent. Once again there is a series of prompts moving down a track, in this case from right to left. There is also a score and several meters (including one that looks like the “rock meter” from *Guitar Hero* and functions identically). The player uses a microphone rather than a guitar or other controller and must attempt to sing the correct pitch when it reaches the base, with the vertical position of the prompt indicating how high the pitch is.

This is a little more “authentic” than the rhythm games, given that some actual musicality is required. Pitch is pitch, not a button-press. However, it is still mimicry and does not demand any technical knowledge or inform the player’s perspective. The game still demands rhythmic and pitch accuracy and does not allow for improvisation or variation. Note that there are lyrics as well for the player to sing to the pitch, but the game cannot tell what the player is saying so the lyrics become irrelevant in terms of gameplay performance. Essentially this is still pressing the

right button at the right time.

Category 3--Synaesthesia/sync

On the other side of the music/rhythm spectrum is a category of games that does not even pretend to base its gameplay around performing music, but attempts to provide a sort of synaesthetic experience, in which the gameplay yields a musical response or the music is otherwise synchronized to the play. Here are a couple quick examples:



Fig 7. Gameplay from Super Hexagon for iOS.

Super Hexagon (available for various platforms; I played on iOS) is a game that is difficult to explain and difficult even to comprehend in screenshots and visuals. The player taps the sides of the screen to rotate a tiny triangle about a hollow hexagon, while a series of walls close in relentlessly. If the triangle hits a wall, it's game over, but there's always a gap and the player must rotate quickly to get the triangle through each gap, with the goal of surviving for one minute. It is a very long minute.

This description of the gameplay does not even mention music, and yet the driving soundtrack of the game is a significant part of the experience. The music is very intense and

overwhelming (as is the visual presentation), and its propulsive beat seems in-sync with the play. It's not spatialized, in the sense that the music could somehow tell the player which way to rotate, and it's dangerous to rely on the beat as a cue for when to rotate at all, yet the music and play do seem deeply connected, such that it just feels wrong to play with the sound off (though again, it is possible and maybe even easier).

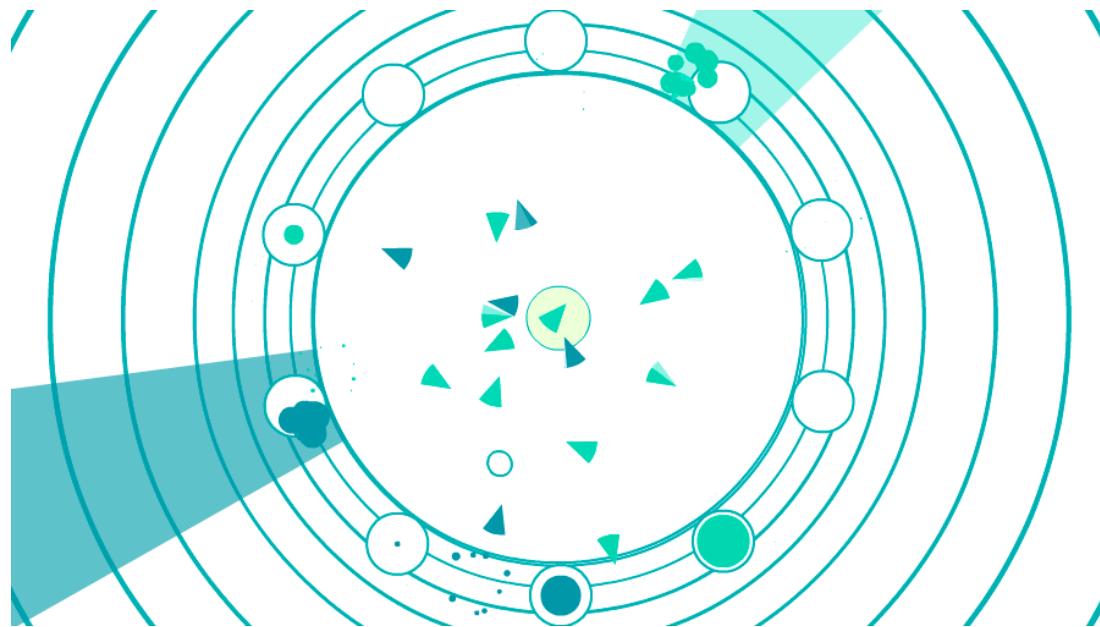


Fig 8. Gameplay from Soundodger for web browser.

Soundodger (a free browser game) is a trip of an experience that is also hard to explain. The player moves a tiny circle about the center of a circular arena, while a series of circles outside shoot triangle-shaped attacks inward. At any time the player can slow down time to make it easier to dodge attacks. The music and sound effects are so integrated with the experience that it's hard to articulate, but one example is that the music syncs with the attacks, such that when there's a big percussive beat certain enemies (acting as the "drums") will shoot triangles. Thus, though the player does not generate the music or interact with it directly, the music is expressed through the game's world, as though it were an interpreter. Getting into the groove may actually improve play performance.

This category is similar to mimicry in that the music makes the gameplay easier (or, in the case of *Super Hexagon*, at least feels very connected to it), but different in that the gameplay is framed as existing separately from the music, and the music is not intended as a cue. Instead the connection is more profound, less about causation (pressing a button causes a sound to play) and more about alternative expression (when a sound plays, also something glows). Thus turning off the music doesn't really defeat the point of the gameplay in the way it does in *Guitar Hero*, but the experience feels less full. This is fascinating territory and well worth exploring in my project as a presentational aspect, but again it's not really about gameplay centered in musicality, creative or otherwise.

Category 4--Music manipulation/creation

This final category is much rarer, for reasons I will delve into. Games in this category give the player some degree of direct control over the soundscape, whether it is producing audio/music directly or by generating it through other actions. I want to distinguish this kind of generation as responsive (meaning the result depends on the specific action) to separate it from the artificial "generation" of games like *Guitar Hero*, where the music plays if the player hits the right note and simply stops playing if he or she hits the wrong note. Here are two examples:



Fig 9. Synthpond for iOS.

Synthpond (iOS) is really a tool rather than a game, so its inclusion is questionable; however this category is so barren that better examples are hard to find, and *Synthpond* certainly does support playful interaction. In *Synthpond*, the player can place nodes of various kinds that either generate or receive sound waves. The waves are represented as concentric circles that expand outward. They play no sound until a wave hits a receiver, which then plays its associated pitch. Because it is a “pond,” i.e. a 2D space, and the player can decide what nature and pitch to give a node as well as set various other parameters before finally placing it anywhere in the pond, the pleasure is in experimenting with different placements of different types of nodes, moving them around and changing their behaviors to see what kind of sound one can produce. This app--made by Parsons MFADT grad Zach Gage--is very interesting because it spatializes a kind of musical experimentation and composition in a unique way.



Fig 10. Gameplay from Beatbuddy for Steam.

Beatbuddy (available on Steam) is an interesting counter-approach to almost anything else discussed so far. It is a game that lets the player remix the soundtrack, live, through gameplay. The gameplay mechanics themselves are firmly rooted in traditional principles (move through a 2D world and overcome obstacles) but all of the audio is attached to creatures in the game, which the player can interact with, turning on and off parts of the audio almost at will. For instance, a clam enemy might have control over the percussion, and by attacking it so the clam retreats into its shell, the player causes the percussion to stop until the clam once again emerges. Thus the timing and relation of the percussion to the other elements of the soundtrack may be adjusted by playing the game.

This category is closest to what I have in mind, because it promotes actual musical creativity, experimentation and composition. However, this is difficult to do. *Beatbuddy*'s

gameplay itself is pretty traditional, it just has an additional sonic layer. *Synthpond* isn't even a game. The advantage of the other forms discussed so far is that they are accessible to most people, because a sense of rhythm is fairly widespread (much more so than an aptitude for musicality or an interest in theory) and the gameplay doesn't ask much technically of the player. Something that embraces other aspects of music and demands real musicality is hard pressed indeed not to exclude many people. This is why there are so few games in this category. And even *Beatbuddy* limits the player to the music tracks included in the levels, whatever the influence on their relative timing. The music depends on the gameplay, but not vice-versa.

Conclusion

There is so much richness in music beyond rhythm that it seems a shame not to explore alternative ways of expressing it in gameplay, despite the inherent challenges and potential for alienating part of the audience. My project explores a way to map an understanding of basic principles of musical composition onto an accessible, trope-based gameplay framework through clever spatialization. Specifically it utilizes the familiar navigation of a 2D space present in a game like *Super Mario Bros.* but turns sheet music into that 2D space, rendering the musical notation as physical objects in the world with which the player interacts. The music for the game is generated by the notation present and tangible on-screen, and the player exerts some degree of control over the notes, using familiar gameplay mechanics. In other words it's a mixture of *Synthpond* and *Beatbuddy*, using the latter's accessible gameplay framework within the context of the former's transformation of notation into a physical and playful space. It pursues the additional goal of transforming the player's understand of musical notation--interacting with the notes as physical objects illustrated in a dynamic version of how they appear on the printed page promotes a new form of thinking about the symbols on the page itself.