

# Audio Interaction in Computer Mediated Games

**J.R. Parker and John Heerema**

Sport Technology Laboratory  
Faculty of Kinesiology  
University of Calgary  
jparker@ucalgary.ca

## ABSTRACT

The use of sound in an interactive media environment has not been advanced, as a technology, as far as graphics or artificial intelligence. This discussion will explore the use of sound as a way to influence the player of a computer game, show ways that a game can use sound as input, and will describe ways that the player can influence sound in a game. We will explore the role of sound in computer games and give some practical design ideas that can be used to improve the current state of the art.

## INTRODUCTION

The term *video game* implies a natural bias towards visual interaction in the typical computer game. It is true that most people gain 80%-90% of their perceptual input from vision. Yet a large number of people function very well with a significant degree of visual impairment. In addition, video games that lack sound have never been popular, in spite of excellent graphics and other features. Sound is something that is assumed to be there, and that can influence game sales negatively when it is badly done, but never causes high sales when well done and that is not expected to be the selling point or major feature.

Sound is a key aspect of a modern video game [4]. The development team either contains musicians or some are hired to create a sound track, usually at great expense. Foley operators, those persons responsible for making sound effects and placing them in the game (or movie) are also key to a development team. One of my favorite examples is the motion picture *Castaway* in which a lot of the sound that the audience thinks is normal background is actually inserted after the fact. In the sequence where the main character escapes from the island, almost all of the sound is generated by a foley artist, and was not recorded on location. Frequently the real sounds do not impress the audience as being realistic!

Video games also have the equivalent of Foley artists; although many effects are now recorded on CDs and sold as sets, there is still a need for *sound design* and certainly for music composition. Many games have musical themes for each character, and segues for each character between each activity; composition is a key part of the overall design and 'feel'.

A knowledge of how audio works, why it's important and how musical assets fit into the scheme of a game and how they are manipulated

## WHY IS AUDIO IMPORTANT?

At the lowest level, sound carries with it the sense of another presence and of activity. This is the very definition of background music. It is there to make the player feel less alone (I.E. connected with something else, or *continuity*) and to remind them that the game is going on, even if they are drawn away for the moment. It's also about activity in the simple sense that fast music is associated with a great deal of activity, and slow music is associated with little activity.

More importantly, sound carries with it more emotion than any other part of the game. It is said by psychologists that we learn first by seeing, next by hearing. However, the sense of hearing seems to be closely connected to the limbic system, that part of the brain responsible for primitive emotions and actions, and can cause emotions more easily and quickly than can vision.

To see why this might be, imagine that you are a primitive human living in Africa some hundreds of thousands of years ago. You are a small, slow, relatively weak primate with no sense of smell. You have a good brain, but no technology yet, and are still vulnerable to many predators. Now, predators of various types tend to make similar sounds - low-pitched growling, rumbling sounds. It should be no surprise that this kind of sound still makes humans fearful - many generations of being a prey animal could easily have done that.

In fact, it's pretty clear that sound is a more fear-making sense than vision for other reasons. If you can see a predator the you're probably OK. You know where he is and what he is doing, and unless you can see that it's interested in you, then seeing it is actually a relief. On the other hand, if you can only *hear* it, then you are in trouble.

There are other sorts of limbic affecting noises in specific categories. The *sudden, sharp* noise is a surprise, and it makes sense to react instinctively. It could be a branch breaking or a rock falling. Humans make certain kinds of noise when in pain or fear - screaming, usually *high pitched, ululating or descending*. We tend to react with emotional similarity when we hear such sound, not in sympathy so much as in fear of whatever is inflicting pain or fear on the other. Loud noises are generally disturbing because anything that could make such a noise is big enough (and close enough?) to be a real danger.

So there is a good reason why some sounds might affect us at a profound emotional level, and this can be used by designers to provide an interesting emotional experience in

multimedia. There are still more reasons why sound is interesting: we can't ignore it as easily as vision (can't 'close your ears' as it were); we use it directly for communication to a greater extent; we have the ability to isolate one sound out of many simultaneous sounds and pay attention to it; and so on. However, the point is probably made.

### AUDIO IN VIDEO GAMES

Current video games use audio in four distinct aspects: music, speech, effects, and input.

Music carries with it a continuity aspect (background), an emotion aspect (spooky VS comical) and a tempo aspect (fast VS slow). This is pretty well known in both music and game development quarters. The use of character themes is standard practice (E.G. a good example being *The Simpson's Hit and Run* [13]) as is composing a segue between activities (walking to running). Automating more of this might be a valuable exercise, not to eliminate musicians but to provide them with more possibilities.

Speech is used in games that involve human-like characters to communicate information to player in a natural way. One can overhear conversations that give valuable information, or one can be given a briefing or command from non-player character (or *NPC*, controlled by the artificial intelligence part of the game software). Sometimes the speech is almost purely for entertainment or mood. An example is when opponents make threats or congratulatory noises, or when the speech is by a voice known to the player: James Bond or Bart Simpson.

Sound effects serve multiple roles. They can serve as confirmation that a requested activity has taken place: as in, 'yes, your missile was launched.' They serve as a warning that something has happened - *your branch broke* - or is about to happen - *footsteps are approaching, look out!* Effects need to be representative of the sounds that things make in the world, insofar as the objects in the game are real. As such, another function of effects is to add to a sense of reality and presence in an environment. Rainfall is both seen and heard, and nearness to the ocean or a river is often heard before it is seen.

Audio input to a game is invariably speech, as in *Nintendogs* [12]. We'll discuss this in more depth very soon, but it is enough for now that audio input is quite rare in computer games at this time.

Speech is easy to record, and if specific voice talent is not required this can be very inexpensive. Sound effects can be purchased as sets of CDs, and once bought can be used in many games. Rarely a designer might need to record some new effects, but these too can be reused and in some cases can even be dumped into a collection and sold. Special purpose music is farmed out, and is composed and recorded by studios for a fee. This is not cheap, but is less expensive than maintaining a staff of musicians and composers and having your own recording studio. A lot of music is purchased and royalties are paid; well-known bands and their music are used for many games (*Grand Theft Auto* [11], racing and snowboarding games like *1080° Avalanche* [10]). Usage fees allow the original recordings to be used,

which sound correct to the player, as opposed to remixes or cover versions.

Games use MP3, WAV, or other standard format to store sounds on the CD or DVD, and simply play them back when they are needed. There is a limited degree of *mixing* going on, meaning that a few sounds need to be played at the same time and their relative volume is specific by the game developer and is played back at that level. Overall levels are relative, so if someone starts speaking while the music plays the software does something simple to maintain intelligibility levels while still playing the music.

So, in summary: how do games use sound adaptively to create an interesting interface? The theme music and segue is one, as is the use of music for emotion and tempo. Alerting the player to actions using sound effects, and speech as a narrative component is another. However, so far the use of sound is mostly *reactive* rather than interactive.

### AUDIO GAMES

An audio game is one that uses sound rather than pictures as the main game state display modality. I do not know what the first audio game was, and it may not be possible to find out. My personal vote is for *Bear Hunt*, a variation on Hunt the Wumpus, an old text-based Unix game. Bear Hunt was devised in the early 1980's by a company named *QSound* in Calgary, Alberta, Canada. The idea was simple - you would stand in the field of some stereo speakers with a gun (a toy). Once in while a bear would come at you from a particular direction, and you would be expected to shoot it. You could only hear the bear, of course, and there was no visual display. The game was in support of their 3D sound product.

There are relatively few audio games, many intended for play by the vision impaired. It is very likely that the full potential of audio as a display mode has not even near being reached. Let's look at some audio games to see what aspects of sound are being used.

#### Karaoke Revolution

Karaoke is two Japanese words: *Kara* = missing and *Oke* = orchestra. Most people know what it is about, and the appeal of the game is that it can be social, among friends, or it can be used for practice for the shy types.

In this game, you have a selection of songs [15]. Choose one and it starts to play along with the printed lyrics moving on the screen. It is therefore not completely audio (nor are most games completely video!) You are expected to sing along, and the computer scores you according to how well you follow the melody. The system extracts the main frequency in you singing and matches it to that of the original song.

This is actually a lot of fun, and the display is really not needed, since most folks know the words for the song they're going to sing. One problem - ornamentation (see section on Non-Voice Input below). If you don't sing the song exactly the way the original singer does, you will get penalized.

## Shades of Doom

This first person audio shooter follows the same basic plot as the original Doom [14] PC game - a research facility has suffered an accident, and you are expected to fix it. You guide your character through tunnels, hallways, and chambers of the laboratory and try to shut the experiment down.

But there's no graphics.

All cues are audio. You can collect items and encounter (and kill) creatures by using the arrow keys to travel the maze of rooms. You can get navigational assistance if you like. The positional sound is good, but I always get killed early.

## Mueckenjagd (Mückenjag, or "Mosquito Hunt")

There are Flash animations with this game, but again, you play using your ears. You hear a mosquito buzzing about. Your orientation is controlled by keys, and the space bar smacks your hands together, with the hope of crushing the nasty pest between them.

There is speech, but it is in German only. Still fun, though.

This game is self-voicing and only available in the German language.

## Bobby's Revenge

A 'shooter', but an odd one. Santa Claus flies past, and you can hear him (bells and such) Then you shoot him - with a paintball gun, of course. You can make him come back by zapping him with an electrical bolt (using a key). Perverse but fun - again, the positional sound is pretty good. Again, I usually lose.

## Top Speed 2

A racing game (my favorite type) that uses sound only to guide the car. You can hear the cars whizz past in 3D sound, and the sound of the side of the road allows you to steer. It has a vast number of tracks, many cars, up to 7 opponents, 4 or 5 different cars, and the ability to customize. Amazing that it is free.

## Ten Pin Alley

Guess what? Bowling. The background sounds are quite convincing, but the announcer is irritating. Simple tones guide your aim. I only have a demo version (2 frames) but it's only OK, not great.

These games and about 195 more can be found at the web site <http://www.audiogames.net> [9]. I was impressed with the sheer number, but I found that many games can't be downloaded, some are not there anymore, and a good number are just lame. Still, a great resource.

## Sound as Input - Voice

Voice recognition is being used in some games right now. The aforementioned *Nintendogs*, for example, has a speech recognition system that is trained to the player's voice, and it executes on a Nintendo DS, a portable game platform having limited processing power. The *Game Commander* software is available for converting speech into keystroke sequences for PC games, so using speech with existing games is easily possible (if sometimes frustrating).

Voice recognition has achieved an effective level of utility, although it is still weak in the general sense, and

requires training to be really useful [5]. Most applications of voice recognition take advantage of a restricted domain of discourse; that is, they involve a small domain of human endeavor that has a small vocabulary associated with it. There are two reasons for this. First, the restricted vocabulary makes it easier to match the input audio signals against the known speech examples.

The second issue is more difficult to deal with in general. It turns out that computer algorithms have had little success in parsing general human speech. That is, even if the voice recognizer succeeds in converting a sentence into text, the correct interpretation of that sentence is still impossible in the general case. Restricting the domain helps here, too.

When giving the computer an oral command, how can it be structured so as to provide a context assist, or a tight syntax that allows the domain to be restricted? We will find help from two unlikely places: the military, and the *Star Trek* television program [6]. My own experience with the Canadian Air Force taught that a command on a parade ground has four important parts: the identifier, precautionary, cautionary, and executive parts. This was reduced to a simple identifier/ acknowledge/ command protocol in *Star Trek*. We can use this syntax when giving spoken commands to a computer.

The important parts of the command, and the only two that will always occur, are the *identifier* and *executive*. The identifier states who is to execute and gives a warning of a command to come. The executive allows a detailed command and indicates that it is to be done *now*. The military command '**squad stand-at ease**' has **squad** as the identifier and **ease** as the executive (in Canada, at least).

The *Star Trek* version is directly relevant to our situation. The command is normally a dialog, as follows:

*Spock*: Computer.

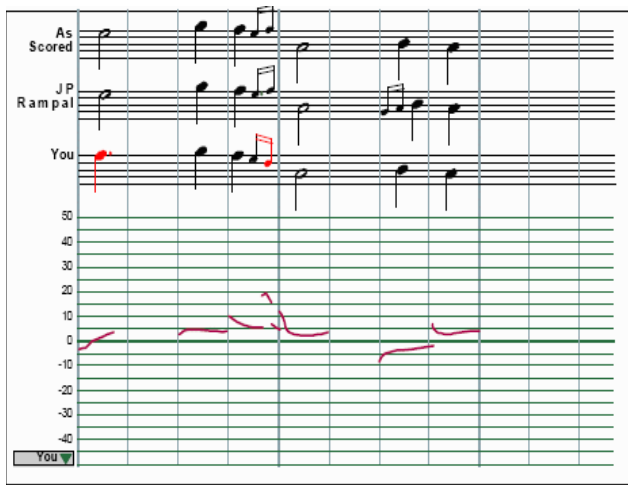
*Computer*: Working.

*Spock*: Calculate, to the last digit, the value of PI.

The idea is to alert the computer that it is to pay attention to the next spoken sequence and execute it as a command. The alternative is that the computer listens to everything, and may execute commands that are spoken but not intended. For example, if your dog starts to chew on a chair leg while your system is recording a favorite TV program onto the hard drive and you shout "Hey, stop!", the computer may interpret this as a command to stop recording. (The "my dog ate my homework" excuse may take on a whole new meaning) Precautionary and cautionary commands can be used to further restrict the domain; After 'computer' the user may say 'music', meaning to search the music library/database function of the system for the next command and then execute it.

## Sound as Input - Non-Voice (*Visual Music*)

Although there has been a little research and development work done on the use of voice for non-voice audio control of a computer application (E.G. [2]) almost none has been done of the input of non-speech sounds in games. Partly because of this, we are developing a game for teaching music, and will use this as an example of the use of non-



**Figure 1:** Intonation visualization section of the Visual Music game/toy.

voice input. Some detail will be given so as to show how much useful information can be extracted from what seems to be straightforward audio input.

The game, *Visual Music* is designed to make practice time more effective by making the player more aware of how well they are playing or singing. This is done by engaging both the auditory and visual centers while music is being played. The system listens to the performance of the player/student and assesses it on the basis of rhythmic quality, melodic accuracy, and in advanced instances the intonation and expression.

In coaching mode the score is displayed on the screen along with the notes that are being played by the student:



Red notes don't always mean mistakes, of course. They could be quite reasonable interpretations of the score, what is called *ornamentation*. To accommodate this, there are two lesson types: *learn the music* mode and *ornamentation* mode, the latter for more advanced players.

An example of a visual display of intonation is provided in Figure 1. The lower part of the screen shows the intonation, and the exact timing of each note played. In Figure 1, *Visual Music* has been set to display a heavy vertical line every bar, and a light line every quarter note. If every note was perfectly, and had no vibrato at all, everything would be drawn along the middle "0 cent" line – the line that represents the exact pitch of the note. If the instrument is out of tune, then every note would be same amount sharp or flat (the light horizontal lines are 5 cents apart<sup>3</sup>). One of the primary things that keep pitch lines from being a straight horizontal line on a sustained note is vibrato.

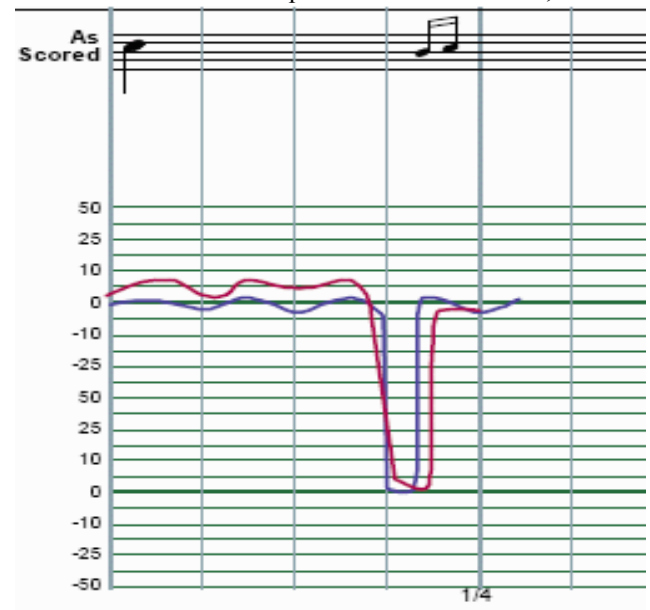
If your intonation (pitch) is perfect, a wavy burgundy line marching across the screen will always be right over the

center "on pitch" line. Sustained tones should be centered over the big horizontal "on pitch" line.

Note transitions are another aspect of musical performance that can be evaluated by this system. If you are a singer, the transition between notes is apt to be a continuous change in pitch from one note to the next. With other instruments, the transition may be more abrupt. Regardless of the instrument, with the exception of keyboard instruments, you may find that your intonation is not perfect during the transition from one note to another. This fact too can be easily seen when the music is visualized.

Finally, the system permits various multimedia assists to be used, such as videos of professional musicians playing the piece in various ways. You can also compare your performance against those of virtuoso players. In the example in Figure 2, the intonation display is set to show Jean-Pierre Rampal's performance (in blue) of the overture to the *Spring* movement of Vivaldi's *The Four Seasons* over the player's performance (in red) of an ornament: the intonation region shows exactly a semitone. In this example, it is seen that the player's performance of the ornament matches the notated interval more closely than Mr. Rampal's. When he played the ornament, the first note took only about a third of the 16th note, while the player's was closer to half. However, Mr. Rampal's transitions are faster and cleaner. Looking more closely, we see that Mr. Rampal managed to play the ornament so that it matched the timing of his vibrato.

The game play aspects of this software are in development, but will involve scores relating to constancy in melody and rhythm, vibrato, and the ability to read and play melodies in real time. Players will be able to play along with famous musicians and have their performances evaluated, and will



**Figure 2:** Comparing ornamentation and intonation between the player and that of Jean-Pierre Rampal.

ultimately be able to play along with other players at distant locations on the Internet.

The details of the game play aspects have not been described, and will only be sketched here. There are really a set of mini-games, each contribution to a whole. Rather like racing games with distinct tracks, one aspect is to compete with well-known musicians at certain performance aspects. Another is the use of different musical selections. Errors accumulate, intonation and expression is evaluated, and the player has the chance to improve their score.

At beginning levels simply completing the piece is worth points, but as skill levels increase it gets more difficult to score. The game is, quite naturally, a piece of educational multimedia as well.

The key novel aspects of the *Visual Music* system lie in the ability to simply play music and have the computer and software accept that as input. This is an example of a *natural* interface. An interface to a game (or any software, really) is *natural* if the activity used in the real world situation is recognized and used by the interface, and means the same thing to the software system or game. A non-natural interface causes an interruption in the *flow* [1] of the activity being performed so as to perform the interaction, and this often results in a splitting of attention that is not productive or amenable to the effective completion of the task being performed. Clear, audio input can be used in inappropriate and non-natural ways, too, which must be avoided.

#### GAMES INVOLVING THE CONTROLLING OF SOUND

So far we've examined video game audio (briefly), games that use audio as the main output, and games that use audio as input, both speech and non-speech. What remains is to look at games that manipulate sound as a major goal or game play feature.

As usual, there aren't many of these. A simple example would be the *Amplitude* [16] video game for the PS2. In these games you get to remix songs given a set of audio tracks. An attractive feature is that the songs are from well known bands like Blink 182 instead of a set of anonymous loops. A problem lies in the assessment of goals - music has an aesthetic component that is difficult to analyze by computer.

#### PC Conductor

Our group is developing a game that uses special devices [6] along with gestures and motions to allow a player to conduct an orchestra and to 'compose' music. This game will never be complete; the idea is to use it as a platform for evaluating interface technologies in the context of games in general, and audio games in particular.

Conducting is done by selecting a score and having the computer play it. Hand motions can be used to control aspects of the performance - cues to instruments, volume, and tempo. Foot motions can control tempo as well, through pressure sensors in the player's shoes.

#### Pressure Sensors

A computer keyboard is a primitive sort of pressure sensor, wherein an impact on a specific part of the sensor is recognized and coded for transmission to a computer. The

switch is either *on* or *off*, with no degree of pressure associated with it. They are really simple switches.

The degree of pressure can be measured using piezoelectric sensors that respond to pressure with an electrical voltage. This voltage is sampled and converted into pressure (*pounds* or *pascals*) by the computer [3].

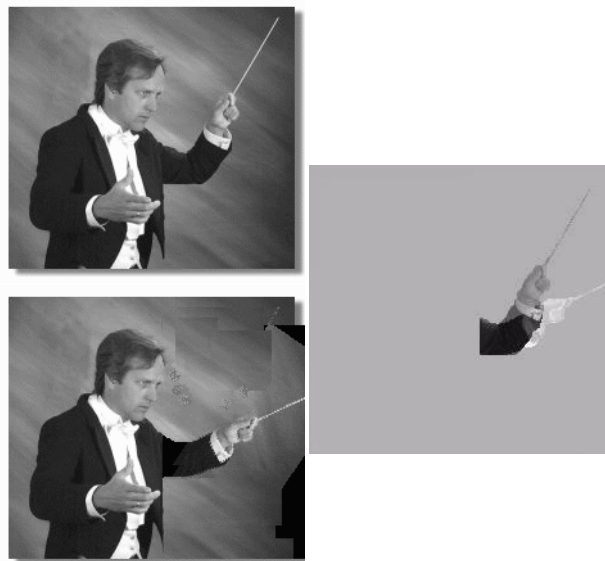
If we now create a small grid of these pressure sensors and place them inside of the shoes of a player then the amount and location of pressures on the foot can be identified. The entire grid of pressures is sent to a PC many times each second and is converted into a pressure *image*. A machine learning algorithm can be used to match the pattern against those for that player that were used for training.

The player can keep rhythm by tapping their foot, a quite natural activity. However, the PC Conductor game allows players to choose what each sensor does in terms of audio control, and will remember.

#### Computer Vision

Kinesiology researchers have been using machine vision technology for quantitative work on human performance for many years. Machine vision is weak in a practical sense as a general input mechanism. Many of the algorithms that work at all well either require too much real time to perform, or only function correctly in very specific circumstances. Fortunately, a set of specific circumstances can be generated for the *PC Conductor* game.

Most of the hard problems in vision concern three dimensional problems; inferring depth from 2D views is hard, even with multiple cameras. Recognizing objects in any orientation is likewise difficult. Avoiding the hard vision problems is a key to workable vision-based game interfaces, and is what you see in current vision oriented



**Figure 3:** Subtracting two frames from a video of a conductor gives essential clues to the nature of the gesture, and thus to the action that the game is to perform. (Dr. Steven Errante, Conductor, Wilmington Symphony Orchestra. From: <http://www.wilmington-symphony.org/AboutUs/Conductor.asp>)

games. In *PC Conductor* the player's 2D image drawn on the monitor, largely for feedback purposes. The player can select the meaning of body parts and gestures, but for the sake of discussion assume that the left hand controls volume and the right hand controls tempo. The player's image is sampled many times each second and a volume level is determined from the height (or size) of the left hand. The down stroke of the right hand will be made to correspond with the beat of the music (Figure 3).

A trick is to recognize certain simple objects that permit easy classification. A ball, for instance, is a circle however you look at it; being the only circular moving object, it is easy to recognize and to follow. Another trick is to use specific colors. We are experimenting with the use of a conducting baton that has a small brightly colored sphere on the end. This would be used instead of the hand, and since the color makes it easy to track, it should be possible to use it for everything for which a real conductor would use their baton.

#### *Evaluation*

Every time someone plays the game the system records data for later analysis, data on preferences on the interfaces, training times, and time to achieve skill levels using specific interfaces. These data will be evaluated after enough has been collected.

Conducting is an art, and computer science is, well, a *science*. The analysis of data involves using a ground truth, or data for which the correct answer is known. For this we require musical pieces that have been analyzed by a conductor and that can be used in a moment by moment comparison. Getting this done is currently the major difficulty in the work.

#### **DESIGN AND IMPLEMENTATION ISSUES**

In a typical video game there are between 50 and 100 hours of game play connected with a typical game CD. There cannot, of course, be this much novel audio stored on a CD, even if it were compressed. A CD can hold no more than a few hours of sound. Without reusing sound sequences, the games could not present a consistent audio presence, so it is clear that the sounds have to be reused somehow. Games usually solve the problem by repeating the same sounds over and over again. Music is played in a loop, and the same one or two sound effect files are used for all instances of the event that cues them. Overall sound quality suffers for this, but it has worked so far. Imagine how well it would have worked for graphics, though: if the same visuals were used for damage to vehicles all of the time, and the same explosion animation was played for all explosions. This would not be acceptable in a high quality game.

It is possible to generate realistic sounds using software, and so create as much of any given sound as is needed while creating needed changes in audio appearance. Sound synthesis from first principles is not sufficiently realistic yet for most purposes; computer created wind or surf sounds often sound artificial. One way to solve the problem is to use small samples of a desired sound and to reconstitute them to for a new, longer, and non-repeating sample. We did this a few years ago to great effect [7][8].

Of course, this method will not work for all sounds. Human speech would be an example where a person would quickly determine whether the sound made 'sense' or not. Sounds that are too complex may also present problems; Still, the synthesis of new sounds from existing samples has the potential to reduce repetition and improve the audio presentation of games.

The audio generation algorithm has an interesting secondary application: it permits the audio to be slowed down and speeded up without concomitant changes in the sound frequency. This is the technology that permits *PC Conductor* to modify the tempo of the music according to the player's motions.

#### **CONCLUSIONS**

When looking carefully at the design of the existing audio games, it's clear that they use sound in only a few, banal ways. Not to offend the designers, who almost certainly had limited funding and restricted goals, but the variety of ways in which sound is used in games at this time is limited, partly by the effectiveness of the available technology, partly by cost concerns, and partly by a too-visual way of thinking. Most of the existing games are really retreaded video games, and specifically use positional audio to replace vision.

That's ok for providing ways to share computer games with visually impaired players. It leaves novel aspects unexplored, though. Perhaps a measure of success would be if sighted players *choose* to play.

Two specific examples of mixed video/audio games have been described: *Video Music* and *PC Conductor*. Both could be made more accessible to visually impaired players. The point was to mix modes, including physical gestures and motions, and use more natural interfaces to the game software. Experiments with the PC Conductor platform should expose further possibilities.

#### **ACKNOWLEDGEMENTS**

Thanks to Sonny Chan and Brad Behm for programming help, and to the Social Sciences and Humanities Research Council of Canada for funding parts of this work.

#### **REFERENCES**

1. Mihaly Csikszentmihalyi, **Finding Flow: The Psychology of Engagement With Everyday Life**, Basic Books (HarperCollins), New York. 1997.
2. Takeo Igarashi John F. Hughes, **Voice as Sound: Using Non-verbal Voice Input for Interactive Control**, Proceedings of the 14th Annual Symposium on User Interface Software and Technology (UIST-01), pages 155-156, New York, November 11-14 2001.  
[www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/voice.pdf](http://www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/voice.pdf)
3. Jacob Munk-Stander, **Evaluation of Piezoelectric Film Sensors for In-Shoe Pressure Measurement** (Joint technical report with University of Cambridge) Technical Report no. 06/04 ISSN: 0107-8283
4. Jim Parker, **Start Your Engines: Developing Driving and Racing Games**, Paraglyph Press, Scottsdale, AZ. 2005.

5. J.R. Parker, **Using Games to Advance Language Training and Education**, Serious Games Summit, Washington DC, Oct. 31 - Nov. 1 2005.
  6. J.R. Parker, **Human Motion as Input and Control in Kinetic Games**, FuturePlay 2005, submitted.
  7. J.R. Parker, S. Chan, B. Behm, **Generating Audio Textures by Example**, Journal of Game Development, 2006
  8. N. N. Saint-Arnaud, and K. Popat, **Analysis and Synthesis of Sound Textures**, *Proc. IJCAI-95 Workshop on Computational Auditory Scene Analysis*, Montreal, August 1995, Pp. 125-131.
  9. van Tol, R. and Huiberts, S., AudioGames.net, (2006) <http://www.audiogames.net/>, (June 30, 2006)<sup>1</sup>
  10. Nintendo Software Technology (2003). 1080° Avalanche, Nintendo.
  11. Rockstar North (1998). Grand theft auto, Rockstar Games.
  12. Nintendo (2005). Nintendogs, Nintendo.
  13. Radical Entertainment (2003). Simpson's hit and run, Vivendi Universal.
  14. John Romero (1993). Doom, id Software, GT Interactive, Activision
  15. Harmonix (2003). Karaoke revolution, Konami.
  16. Harmonix Music (2003). Amplitude(PS2), Sony.
- 
1. Game references and electronic sources are given according to the *Columbia Guide to Online Style*, Janice R. Walker and Todd Taylor (Columbia UP, 1998). <http://www.columbia.edu/cu/cup/cgos/basic.html>