2018-02-09

When the Soundtrack Is the Game: From Audio-Games to Gaming the Music

Kirke, Alexis

10.1007/978-3-319-72272-6_7
Springer International Publishing

Chapter 7
When the Soundtrack Is the Game: From Audio-Games to Gaming the Music

Alexis Kirke

Introduction

In most games, sound plays second fiddle to the visuals. There is always much talk about the graphical rendering power of a computer or how beautiful and/or smooth the visuals are in a new game. There are of course games that combine sight and sound but where sound plays a larger part than normal—for example Guitar Hero, in which the user attempts to play along to popular music with a fake guitar. However, these could still not be described as "audiogames", as without the visuals they are unusable. The emergence of the iPhone and other smartphones has led to a number of innovative mainstreams games being produced which are sound only, such as the Papa Sangre series. In fact audio-only games for people with sight difficulties has a long history going back to the 80s.

This chapter introduces the concept of a game—Musicraft—that is both audio-only, and in which the soundtrack is the game, and which can be played as a game or an amateur composition tool. Musicraft draws most of its inspiration from the popular children's game Minecraft (Short 2012), a game which cannot be played by audio-only. Minecraft has two modes of play: Survival and Creative. Survival is the default mode and involves harvesting resources to build structures and protect oneself from roaming monsters, and to survive and become more powerful. For most players Survival play is an enjoyable and thrilling experience. As a result they learn the ins and outs of the building and harvesting interface without any effort—they *want* to beat the game. Later many players switch to Creative mode. In this mode, the player cannot be harmed. They focus on using the skills they learned in Survival mode to contribute to the environment, often on shared servers so they can show off

A. Kirke (✉)
Interdisciplinary Centre for Computer Music Research (ICCMR), Plymouth University, Plymouth, UK
e-mail: alexis.kirke@plymouth.ac.uk

D. Williams, N. Lee (eds.), *Emotion in Video Game Soundtracking*, International Series on Computer Entertainment and Media Technology,
https://doi.org/10.1007/978-3-319-72272-6_7

their creations to their friends. The game becomes a creative and social experience, as players build structures (e.g. castles and cities) and devices (e.g. mechanical contraptions) and are able to enter and interact with them. The audiogame Musicraft is conceptualised to play as a game, during which the players learn how to construct music in an abstract audio-only world isomorphic to elements of common music notation. The music constructions are focused on surviving and avoiding musical monsters. Later they can move into a creative mode where they simply play the game to create music they like. A secondary inspiration for Musicraft is the audio memory games like Simon and Touch Me, where the user must recall the order and location of certain pitches to win the game. As will be seen, by its very nature as a truly musical game, Musicraft is a non-persistent looping world that requires a memory model to be built in the mind of the player.

Overview of Past Audiogames

Audio games as a concept have been around for many years. There have been a number of useful overviews published (Mangiron and Zhang 2016; Rovithis et al. 2014; Yuan et al. 2011; Westin et al. 2011). The first reported audiogame sold was Atari's Touch Me game (Beksa et al. 2016) in 1974. This was a memory game where a series of tones would play, and user tried to recall the tone sequence and press appropriate buttons to mimic it. Although it was also a visual game (lights flashed in concert with the tones), the game was simple enough that it could be played by people as a pure audiogame. A similar handheld game was released called Simon (Rovithis 2012) in 1978 which was even more popular. In the 1980s a number interactive fiction or adventure games were being developed. These involved users being given text descriptions of a location ("you are in a large cave" or "you are on the bridge of a spaceship", etc.) The user could then enter text to say what action they wanted to do ("go north", "pick up gun", etc.) The gameplay normally involved a number of puzzles presented in descriptions and solved by entering a series of commands. By connecting these games to a text-to-speech interface, they could be made playable by sound only (Spiel et al. 2014).

Real Sound—Kaze No Regret was a game released for the Sega Saturn and Dreamcast consoles in 1997 and 1999 respectively (Fizek et al. 2015). Like many audiogames, it was designed for people who are blind or partially sighted. It can be thought of as an interactive radio drama. The user listens to segments of story and the then game chimes and then can make selections which decide how the plot unfolds. Audio Space Invaders (McCrindle and Symons 2000) was a PC game that used 3D audio called ambisonics (Frank et al. 2015), designed for a four speaker system, but which can apparently work adequately on a two speaker set-up. The game play consists of shooting flying invading aliens. It does not require a graphical interface to be played but, like many audiogames, provides a graphical interface in addition. This is a directional audiogame—much of the game is based on the user hearing where the enemy is: to their left, right, in front, behind. Different sounds

represent different types of enemy ships, the Doppler effect indicates a ship's direction of movement, and pitch represents closeness to the player.

The Blind Eye was released in 2000 by The Blind Eye Research project (Beksa et al. 2016). Once again the focus of the game was on binaural 3D sound rather than speech sound. Binaural sound enables the simulation of surround sound on a pair of headphones/earbuds (Rumsey 2012). The game is a timed search-and-collect task set in a city environment. The user must find a series of musical instruments hidden in locations whilst avoiding cars or colliding with walls. The city sound environment is simulated for the user. In addition the user's footstep sounds change as they walk on different types of surface.

Shades of Doom, released in 2001, is an audiogame in the spirit of Doom, a new first-person shooter built from the ground up (Parker and Heerema 2008). It uses surround sound, and the user can hear things like the sound of wind in passages, footsteps echoing, and nearby equipment. There is also an optional voice-based environment analyzer. AudioBattleShip (Sánchez et al. 2003) is also designed for people who are blind. Once again 3D sound is used to implement the game, in this case the traditional battleships board game. Players are can hear the direction of their bomb drop, together with the result. A tablet is used for input, and sonic cues can also be given to remind the user where they have placed their ships.

Drive is an audio-only "racing" game released in 2003 (Velleman et al. 2004). However rather than focusing on steering (like most racing games) it focuses on speed, which the player must maximise. The user drives their car over power-ups spread along the track which they pick up and activate with keypresses. As they speed up the power-ups get harder to pick up. The speed also causes a musical soundtrack to change tempo and a co-pilot's voice to change behaviour. There are various other sounds in the environment designed to distract the user from their task, such as doppler-effected sirens and helicopters. Another driving-related game was made by the company that made Shades of Doom: GMA Tank Commander (Denham and McComas 2004). It is based on the classic arcade game. This is another surround sound game. It includes audio targeting systems, the sounds of approaching tanks, environmental ambience and audible recommendations from a tank crew.

Terra formers (Westin 2004) is a sci-fi action and adventure game which uses 3D sound to provide a sonic compass, and sonar to provide a sense of distance from objects in the direction the player is turned. There is also voice guidance. Footstep sounds indicate what the user is walking on. The player has to walk through different rooms and levels solving problems, and avoiding robots and other danger, to progress through the story. As mentioned, early audiogames were often based on text adventures. The Last Crusade is a text-based role playing game (RPG) released in 2004 (Dwyer and VanLund 2008). The game speaks out text but also gives useable items their own name sound. They also have a unique sound when they used by the player. Similarly with mobs and non-player characters.

Some research has been done on extending non-audio games into the audio realm. An example—in the spirit of Shades of Doom—is AudioQuake (Atkinson et al. 2006), which does not build a game from the ground up, but aims to make the

groundbreaking first-person shooter Quake playable without visuals. The player has certain extra objects, a radar scanner, and an item-detection scanner. These give audio outputs and alert the player to an object or event. A more obvious candidate for audiogame conversion are music games designed for sighted-people, such as dance games and games where the user plays a fake guitar to attempting to synchronize with the music. In "Finger Dance", a new game rather than an audio overlay, users listen to music, and try to match the rhythm with key press patterns using four keys, responding to different types of drum rolls. There is also an audio-based menu system. Another audiogame from the music/dance genre is AudiOdyssey (Glinert and Wyse 2007). The player is a DJ and uses the keyboard or Nintendo Wiimote to play dynamically generated sounds. The aim is to trigger them in time with the current song playing to excite the virtual audience on the dancefloor. Stereo spatial cues tell the user how to move the Wiimote.

Speed Sonic Across the Span (Oren 2007) is a platformer audiogame that uses voice, auditory icons and earcons. Objects are represented to the left and right of the player by using stereo panning. The player can jump, walk and stand still. Jumping causes the pitch to go up and then down and the player lands with a noise. A voice announces which platform level platform the player is on (1, 2, 3 or 4). Running or jumping into a solid surface causes vocal sounds to trigger. A form of sonar helps the player find platforms in front of them. The sounds of enemies (bees and dogs) are panned so the player can tell what direction an enemy is approaching from.

Blind Hero is a music rhythm game (Yuan and Folmer 2008)—but unlike the earlier Finger Dance, it is an overlay on the Frets on Fire, a game similar to the well-known guitar hero. Blind Hero involves a specially-designed haptic glove that the user wears. The glove gives vibrations on different fingers to indicate that there are notes arriving that need to be played. As well as engineering the glove, much work had to be put in to design the timing of the vibrations, which indicate both how long to press the fake guitar buttons and how long to press them.

The release of the Apple iPhone had a significant impact on audiogames as it created a powerful piece of technology that was heavily headphone based, and had a very small screen. The 2010 game Papa Sangre (Östblad et al. 2014) was an audio-only released for iPhone which won or was nominated for a number of gaming awards, and fascinated both people who were blind and people who were not. It is based in a supernatural world. The player moves by tapping the screen as a footstep metaphor, and turns by turning the phone. The game uses binaural audio to create a surround effect in the headphones. The player must avoid hazards, solve problems and collect musical notes in the environment. A sequel was also released. Sound Swallower (Beksa et al. 2015), another iPhone game, was published in 2011 and is an audio-only game that uses the device's GPS. The player must avoid the Sound Swallower and run to collect sounds from around them before they are erased.

The Nightjar (Mangiron and Zhang 2016) is an iPhone and uses the same engine as Papa Sangre. Rather than supernatural world it is based in a science fiction environment, on a space ship. It utilizes the voice of the ship's computer and a separate voice guide, sometimes they deliberately contradict each other. The game received much mainstream publicity partly because the guide voice was provided by the

popular actor Benedict Cumberbatch. Blindside (Brieger 2013) was a post-apocalyptic themed adventure story game released in 2012. It is compatible with Windows and Mac as well as iOS. In the game the player wakes up to find that they, and most of the people in their city, have gone blind. The player must navigate the city streets avoiding monsters, and discover more about what has happened. On the iPhone screen touch allows the player to walk around and phone tilt controls other movement and direction.

Released in 2014, Audio Defence: Zombie Arena (Brammer 2016) was a more fast-paced game than some of the iPhone puzzle solvers. It is a 3D sound first person zombie shooter. The user can hear the zombies approaching around them. They must turn to the face the zombie and fire a gun to destroy them. The user can select different guns with different shooting sounds and abilities, including a sonic cannon.

Moving back to the adventure/puzzler type iPhone games, A Blind Legend (Csapó et al. 2015), is an audio only action adventure game. The touch screen is used for general movement, and for combat actions. There is also a companion who provides a voice guide, saying things like "Look out—behind you!" A related game, but developed for PC, is Legend of Iris (Allain et al. 2015). It is a game for children who are blind, but has a specific non-entertainment function. Its aim is to help train the children's navigation skills. Although it is designed for audio only A Blind Legend uses Oculus VR to take advantage of the headset's directional audio system. Thus players can move their head to change the sound direction. The adventure game involves a number of puzzles is inspired by Legend of Zelda and it's goal is to get a giant out of a spirit village. Solving puzzles involves practicing skills such as locating the origin of a sound and then remembering that location, and following moving objects by sound only. Interestingly the game was designed in collaboration with a composer who lost his sight at the age of 14.

Looking over the last few decades of audiogame releases and research, some patterns can be discerned. There is often a direction-based component, and the worlds created are usually attempts to simulate (even abstractly) a real world. Another common method is the use of speech to build environments. Where music is used, it tends to be a background element, except in the music/dance games. It is interesting to note that games such as AudioQuake and AudioBattleShips were partly built to provide extra social activities for people who are blind, and that Legend of Iris aims on providing an educational tool. This is of relevance to the game we will describe in this chapter.

The system now discussed in this chapter Musicraft, can be compared to Legend of Iris in an educational sense, and to the music/dance games, in that the environment is music not a simulated real world. Unlike most music/dance audio games, in Musicraft the music is the environment—not part of the environment. The main addition of this type of game to the field is that it has a creative mode. The fact it provides an intuitive ability to learn common music notation and can be viewed as a form of memory game, are auxiliary additions, which would require significant extra testing.

Fig. 7.1 A standard Minecraft survival environment

Musicraft

The audiogame designs discussed in the previous section do not have a seriously creative musical component, even Guitar Hero uses a highly simplified guitar and musical notation. There are iPhone games like Isle of Tune on iOS (Consalvo 2011) that are quite sophisticated and creative toys for building tunes. However Isle of Tune is more of a toy and is graphically-based, not accessible as such.

Most audiogames that are based on a "geography map" the physical geometry into a sound-space experience across the headphones. Musicraft takes a different approach, mapping a geography across time rather than spatialization. The map is metaphorical rather than representative. It acts as though the elements in a bar of common music notation have physical properties, rather than mapping a physical space onto the bar. Also, like Minecraft, it is both a game and creative and can be played ignoring all of the game elements to make music of different genres.

The simplest way to explain Musicraft, is to introduce Minecraft. Minecraft is initially a survival game. To win the game one must collect resources in an environment to defend against mobs (the Minecraft name for monsters) and starvation. Fig. 7.1 shows a typical Minecraft survival environment. The entire environment is potentially collectable, rebuildable and restructurable. As a result Minecraft also became very popular in a mode known as Creative Mode. Fig. 7.2 shows an example result of creative mode. In creative mode, one cannot die, and the mobs cannot harm you. In fact you can generate mobs. A player can fly through the air and instantly get any building/resources they desire from a menu rather than having to seek them while avoiding mobs. Fig. 7.3 shows the most basic building block of Minecraft. It can be seen that much of Fig. 7.1 is made using this block. In survival mode the player has to build increasingly powerful tools and weapons to mine better

Fig. 7.2 A Minecraft environment made by a user in creative mode

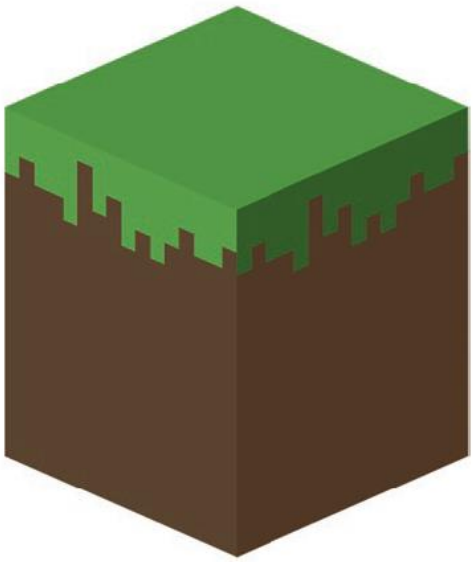

Fig. 7.3 An example building block from Minecraft

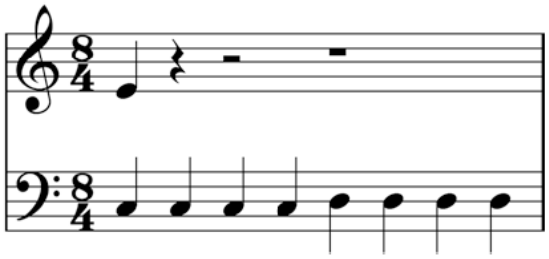

Fig. 7.4 A basic Musicraft World

resources and kill more powerful monsters. But in creative mode these are all available instantly.

Most people start playing Minecraft in the survival mode, and then many go on to play it in creative mode. The players often build huge and complex structures and machines which could be shared with friends by logging onto the game on shared servers. Minecraft evolved into a giant dynamic shared Lego as well as a game.

Musicraft is built in a similar spirit. It is designed to first be played as a musical game where the player avoids mobs. However, the skills that are learned playing this game can eventually be used in a creative mode, in which the player can be immune to mobs and build their own dynamic musical worlds. In fact the mobs become a part of that musical world. The game also has the potential to implicitly aid the learning of common music notation, and to exercise audio-based cognitive skills due to its method of looping rather than the standard soundscape. The last of these elements needs a great deal more investigation, as such a looping approach may make the game less attractive, and there is only limited evidence for how games can improve cognitive function (Rebok et al. 2014).

The transition from survival to creative mode is the element designed to make Musicraft unique. Players should not feel they are being forced to become skilful at making music with common music notation, but at surviving a game. It is only afterwards they realise they have now become algorithmic music makers. That they have learned certain principles of music while playing the game. (Note that although Minecraft has block types which are musical, the use of such blocks and Minecraft in general is not accessible to those who are blind.)

So Musicraft is a highly simplified 2D Minecraft, whose "block types" are musical notes in common music notation twinned with a timbre. Movement paths are only available on lines or spaces in the staff. The player is a quarter note starting at, without loss of generality, middle E at the beginning of the looped bar. To go above middle E the player can "jump" (by, for example, pressing the space bar) but will return to middle E unless they jump in the direction that has a note that is low enough for them to climb onto.

Figure 7.4 shows a basic Musicraft world. Worlds can be of different lengths and tempos. They can also be of different tonalities and timbre sets. This is an 8 beat world and, we will say, 100 beats per minute world. A world is audio-displayed by playing it as a loop. The bass clef in the following example is not the "ground" but the backing. It is designed to provide the beat and also to make it clear where abouts in the display loop the output is. That is the reason for the first four note pitches in

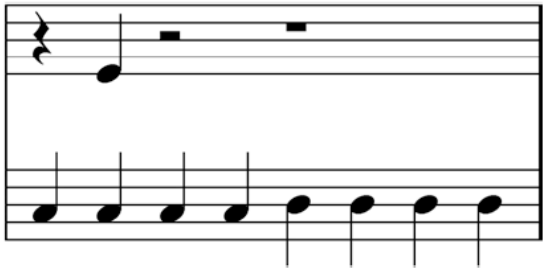

Fig. 7.5 Player moves to the right in the world

the bass clef being different to the second four notes. The single note in the treble clef represents the player and their location. They are at the "leftmost" point in the 8 beat world. The player can move left, and right and jump.

This makes clear a key feature of Musicraft—it is non-persistent. Most previous audiogames create a sense of a persistent world. The whole world can be heard at the same time. In Musicraft the world is non-persistent and requires the user to listen to the whole loop to know what is in the world. Note that this does not mean that Musicraft is a sonification (Kramer 1993) of Minecraft in any sense. The Musicraft world is the common music notation.

Jumping will move the player up one note for one cycle. A cycle if one playing of the World. Then the next cycle the player will return to the lowest line. Moving left and right will move the player back in time or forward in time. If the player is at the start of the world (bar) they will wrap around to the end of the bar (world). Figures 7.5 and 7.6 show these two examples.

After pressing the jump control, a player can also build using the build control. This places the note below where the player was before they pressed jump. Fig. 7.9 shows this happening. In the second bar, the F in the treble clef is the player, the E is a permanent note the player has built. The player will remain at F until they jump or move. A player can also "double" jump which means they press the jump control twice in quick succession. This causes them to jump two pitches up. The result is shown in Fig. 7.10. When this is combined with pressing the build control after the double jump, the result is shown in Fig. 7.11.

A player can jump. Pressing the jump control will cause the player's note to move up one key pitch for one cycle of the world and then return back to the previous key pitch. This is shown in Fig. 7.7. A player can also build or place "blocks"—i.e. notes. Pressing the appropriate control will cause a permanent note to be placed at the left or right of the player, as seen in Fig. 7.8. These blocks can be a different timbre to the player's timbre.

If the player jumps and then presses the place block control straight afterwards a block will be created on the line they were just on. This will cause the player to land on that block and not fall down to their original position, as seen in Fig. 7.9. Musically jump can create intervals of a second or minor section. Double jump allows the creation of thirds. Double jump involves the user pressing the jump control twice in rapid succession, and the result is shown in Fig. 7.10. If they press the build control straight afterwards a block will be placed, but a note higher than usual. Fig. 7.11 shows a player using jump and double jump, and placing blocks both times.

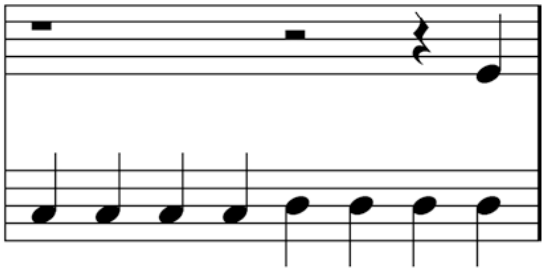

Fig. 7.6 Player moves to the left in the world

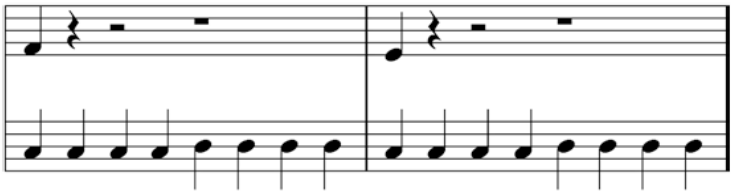

Fig. 7.7 Player jumps up one note, for one cycle of the world

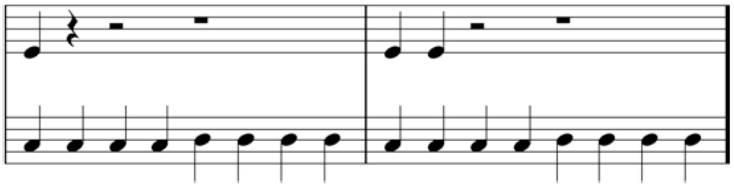

Fig. 7.8 Player builds wall to right of themselves

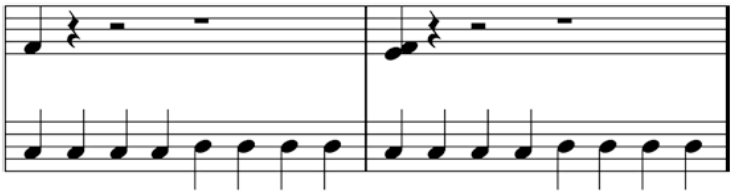

Fig. 7.9 Player jumps up one note, and builds a note under themselves

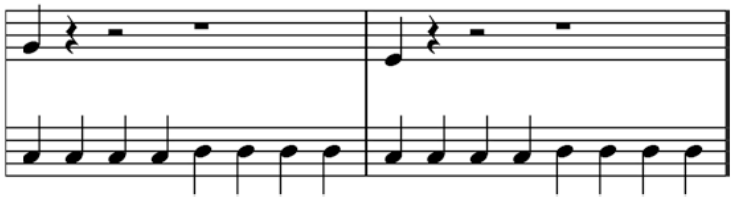

Fig. 7.10 Player double jumps up two notes

alexis.kirke@plymouth.ac.uk

The next element to be described in Musicraft is "mobs". Mobs are the monsters of Minecraft. In Musicraft a mob is a note or notes, in a mob timbre. Fig. 7.12 shows a one note mob appearing in the world, and then moving towards the player over two cycles. The player does not move as the mob approaches. In the fourth world cycle the mob destroys the player. Given that timbres cannot be "seen", letters have been used above the bars to clarify. Note how the player in Fig. 7.12 is faster moving than the player (not the case with all mobs). The player can only move one beat a cycle, but the mob can move two. So how can the player escape the mob? Fig. 7.13 shows how. The player jumps and builds a wall, then moves behind the wall by wrapping around the world. Then they build another wall. They are safe from the mob.

A player starts with only a certain number of such "wall" notes available. A little like a Minecraft blocks inventory in survival mode. The player can mine notes beneath themselves to add to the inventory. So the player would need to be part of a chord, harmony or genreto do that.

Looking at this basic functionality of Musicraft it is clear to see how the creative mode allows a user to create their own musical loops. If the mobs cease to be dangerous to the player, then the player can simply build and dig notes wherever they wish. The mobs become non-deterministic features of the tune. Though the user can bound their behaviour by building walls. In fact in Musicraft creative mode, as in Minecraft, the user can spawn mobs. This allows the user to embed dynamic sonic objects of different timbres into their loop.

One advantage that games which simulate a physical environment, or which simulate dance or musical playing, and which Musicraft doesn't have, is that there

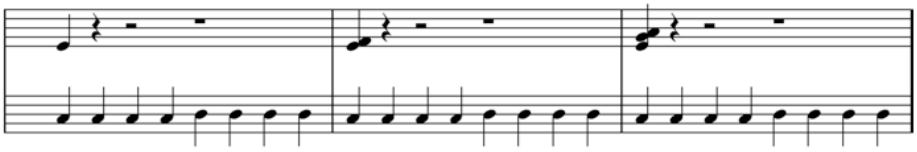

Fig. 7.11 Player double jumps up two notes and builds wall

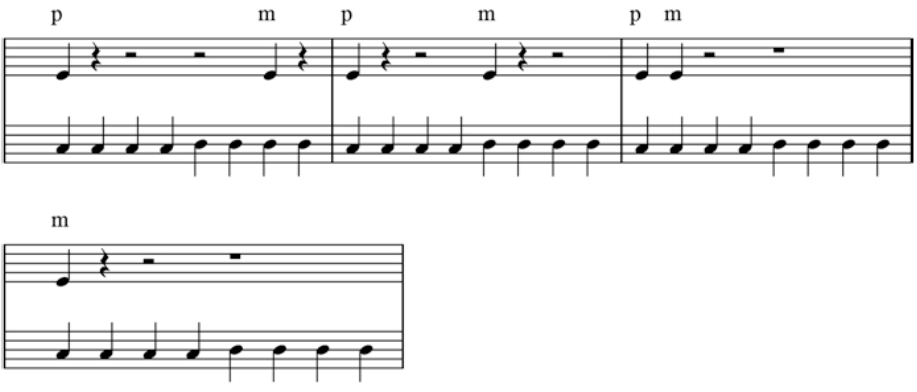

Fig. 7.12 Player pursued and killed by mob, p = player, m = mob

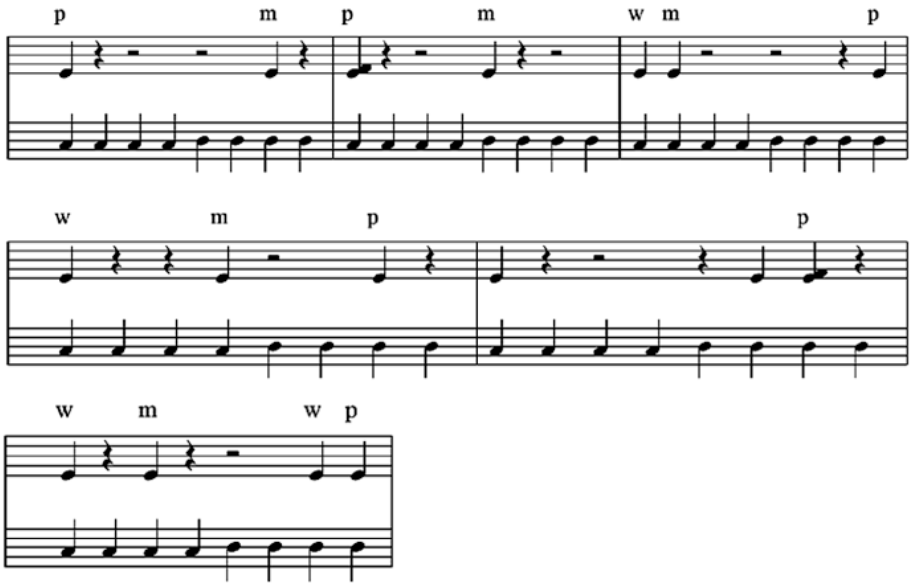

Fig. 7.13 Player protected from mob by building wall, p = player, m = mob, w = wall

is a sense of persistence. This comes from the modelled world seeming natural. The idea of a world existing in a musical loop is less natural. Musicraft is so abstracted compared to these, that it can be expressed as a loop-based music/rhythm game instead of a mob and wall building game. In this view the player controls a note cursor that can be used to place notes. The player can move their note to earlier and later in the loop. The basic game is: sometimes the player hears notes that are coming earlier and earlier each time the loop happens. In this case the player needs to place a note at the same pitch as this note or be destroyed.

There are however potential advantages to thinking in terms of "mobs" and "walls". One is that it provides an opportunity to teach the basics of common music notation. But more importantly it provides a creative metaphor for generating new game features, as will now be demonstrated.

Extending Musicraft

A first obvious element of Minecraft that can be mapped to Musicraft is different block types. Some Minecraft block types are shown in the right hand part of Fig. 7.14. In Musicraft, rather than differing by colour and texture, block types differ by note length and note timbre. Examples of note lengths are crotchets, quavers, semi-quavers, and minims. Timbres include piano, bass drum, violin, snare drum, electric organ, vibraphone, any synthesized sound, and even singing. In fact, one only needs to think of the vast number of sounds available in classical and contemporary music across the world, to the see that the opportunities for sound blocks are vast.

Fig. 7.14 Some Minecraft block types on the right with the sort of items that can be used to build Musicraft blocks on the left

Fig. 7.15 Four of the Minecraft mobs: skeleton, enderman, creeper, zombie

Minecraft also has a variety of mobs which look and behave differently and have varying abilities. Fig. 7.15 shows four of these mobs: the skeleton, the enderman, the creeper, and the zombie. The zombie is the closest to the Musicraft mob described earlier. It simply moves towards the player and kills the player by "touching" them. In Minecraft, the skeleton fires a bow and arrow—so has a ranged attack. The enderman can pick up blocks and can teleport between locations almost instantly. If a creeper gets too close to a player it explodes a few second later, destroying an number of blocks around it.

The skeleton in Musicraft is shown in Fig. 7.16. The skeleton has its own timbre and sends a deadly musical note along a random line near itself. It can be seen in Fig. 7.16 that the skeleton to the far end of the loop is firing an "arrow" made of a quarter note. Luckily for the player, the note misses them. Had the skeleton fired it along middle E, the player would have died unless they avoided it or built a wall. An Underman can be seen in action in Fig. 7.17. It is teleporting around the loop, and in the last bar it removes a block from the player's wall.

Figure 7.18 demonstrates the behavior of a Crawler. In this case the player allows it to get too close, and this triggers the explosive countdown of the Crawler. Then a

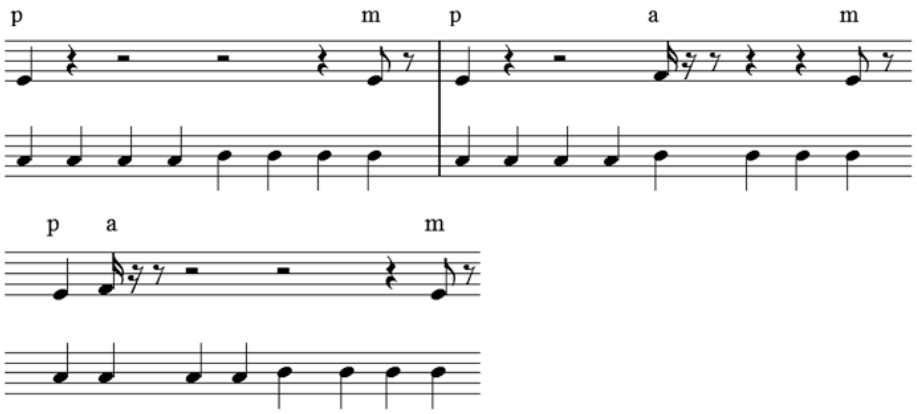

Fig. 7.16 Skeleton mob in Musicraft, p = player, m = mob, a = arrow

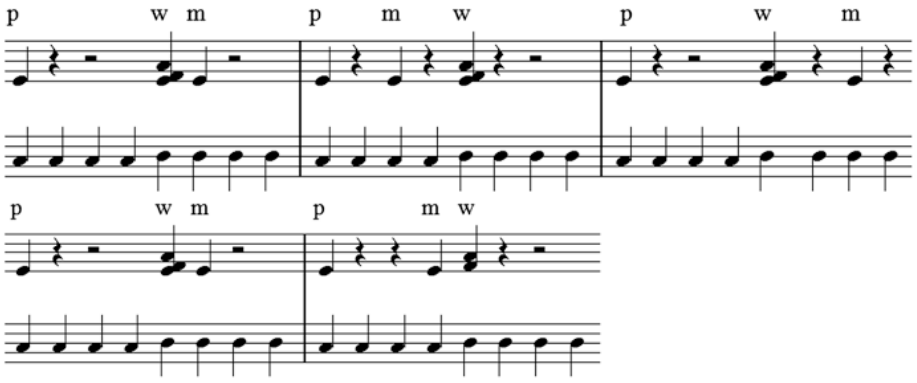

Fig. 7.17 Underman in Musicraft, teleporting and removing blocks

little later the crawler explodes, destroying the nearby wall. The player will hear the Crawler has been activated by a change in timbre.

Minecraft also has a concept of Biomes. These are environments such as plains, desert, and savannah. Fig. 7.19 shows a few Minecraft Biomes. A related concept in Musicraft is genre. Figures 7.20 and 7.21 give two examples of a player building a wall, one in a dance music genre and one in a classical music genre. In both cases the user is in the lowest line, and the other lines are accompaniment.

There are many other extensions which can be envisioned for Musicraft that have stronger or weaker connections with Minecraft. For example multi-player worlds, in which two people control a note each; objects and mobs the move in uniquely musical ways—such as arpeggios, or chords; armor (moveable walls) and weapons (note-structures that damage mobs), ceilings (notes that repeat across time), the ability for players to move between parts of a genre and play there—i.e. into the drum section or the bassline, and finally active note structures which are comparable to “redstone” blocks in Minecraft. Redstone in Minecraft allows the building of

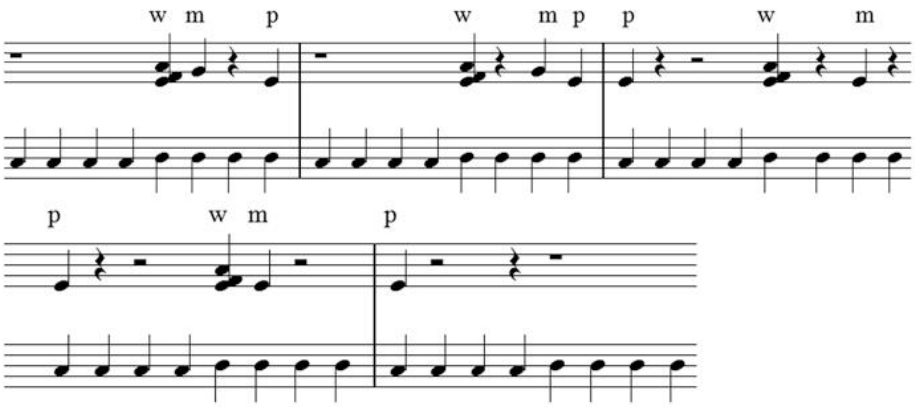

Fig. 7.18 Crawler in Musicraft, exploding and removing a wall

Fig. 7.19 Three Example Minecraft Biomes

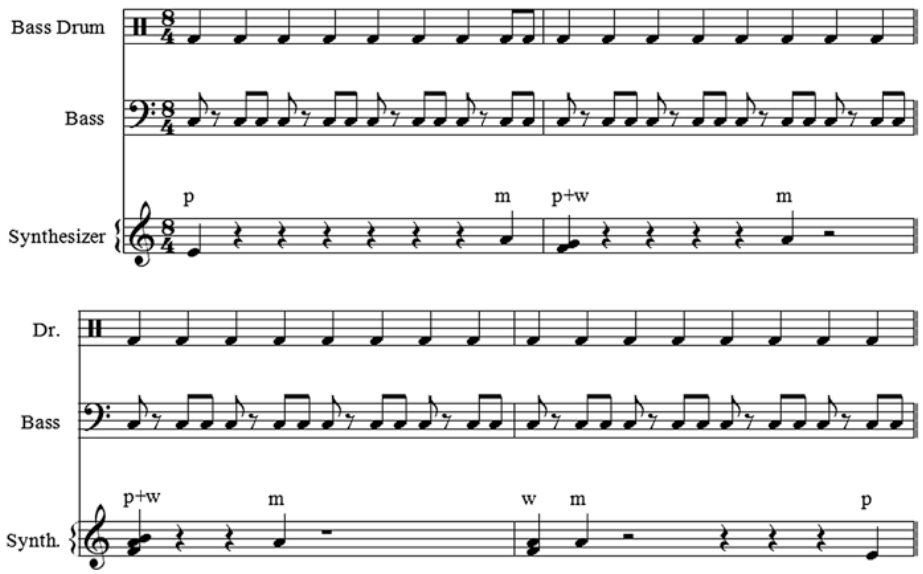

Fig. 7.20 Musicraft genre of dance music, showing a player building a wall using double jumps, to avoid an approaching mob

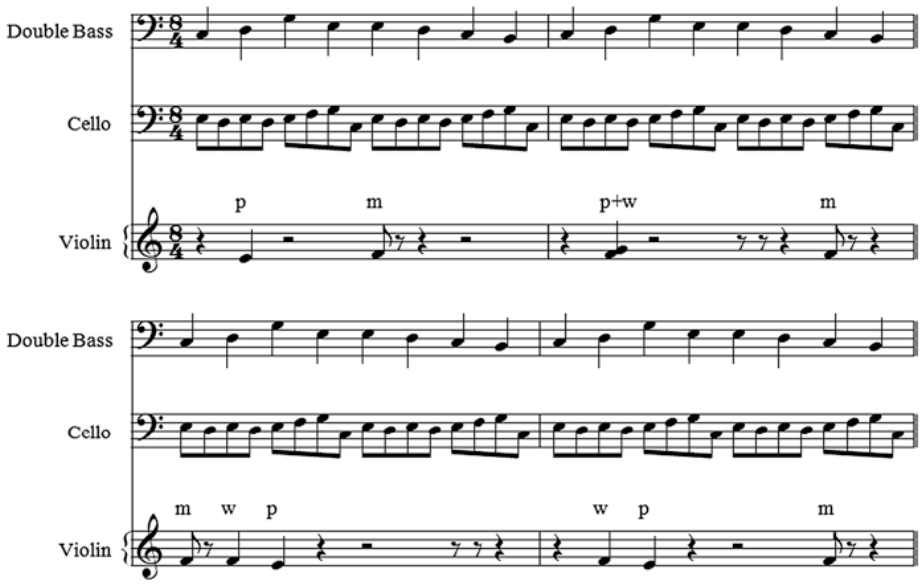

Fig. 7.21 Musicraft genre of classical music, showing a player building a one note wall, to avoid an approaching mob

transmitting and conditionally active structures. They can be used to build computers or factories. However, before implementing such extensions, some core issues need to discussed with moving audiogames and Minecraft-type ideas into a world of common music notation.

Discussion

The key issue with Musicraft is the same as the issue with many audio games: degrees of freedom and persistence. There is a limit to how many different sounds the ear can work with in an auditory scene, and—by its very nature—sound is usually temporary. It fades away unless the sound is continuously played. Visual games can take full advantage of the high resolution visual modality, together with its natural persistence over time. Musicraft is particularly vulnerable to this limitation as it represents the space as a loop. Thus the when the second part of the loop is playing the first half is no longer playing. So to increase the size of a Musicraft world horizontally requires the player to increase their memory model of the space. If rather than 8 beats, it is increased to 16 beats, then the player will need to increase the size of their model. Also the larger the size of the space, the longer it will be before the player can either do something, or hear the results of their actions. So there needs to be a balance between the space size being too large, unresponsive and unwieldly, with the space being too short and low resolution. Of course this can be somewhat offset through the use of tempo. Increasing the tempo of a Biome means it refreshes more quickly. But it also means the player may have to listen more carefully and react more quickly to the space.

Other ways to deal with the resolution issue is allowing play to move between instruments in a Biome, thus increasing freedom in the vertical dimension. Yet more ways include the use of structures whose length is the width of the world. For example an 8 beat note in an eight beat world. This structure is in essence persistent. In fact one can envisage rotating the whole of a Musicraft world 90 degrees and making object locations pitch-based rather than time based. Then a user can hear the whole content of the space constantly. This has the issue that pitch and timbre are likely to clash more in a single time instant, than across multiple beats.

These issues do not remove from the main and the secondary contributions envisaged by the concept of Musicraft to audiogame design. The secondary contribution is the way that whole game forces the user to think in terms of common music notation. For a sighted user this can be a way of learning the basic structures of common music notation without even realizing it. The primary contribution is the idea of bringing a creative mode to audiogames. Through the use of wall building, mob spawning, and many easy-to-imagine tools in Musicraft, it can be seen how once the user has learned the skills in competitive survival gameplay, they can easily transfer these to real-time composition.

Summary

This chapter has discussed the concept of audiogames—where the entire focus of the game is its sound—and introduced a new game concept for audiogames using an example. The concept is of the game being entirely made up of its musical soundtrack—the game is the soundtrack, and of having a creative mode. In particular, this game called Musicraft, is inspired by Minecraft to have both a survival and a creative mode. This is a natural extension of a game that is essentially a piece of music, to allow the player to take the skills they learned in playing the game, to actually creating new pieces of music with those same skills.

References

Allain, K., Dado, B., Van Gelderen, M., Hokke, O., Oliveira, M., Bidarra, R., Gaubitch, N.D., Hendriks, R.C., Kybartas, B.: An audio game for training navigation skills of blind children. In: Serafin, S., Avanzini, F., Geronazzo, M., Erkut, C., de Goetzen, A., Nordahl, R. (eds.) 2015 IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE), pp. 1–4. IEEE, Piscataway, NJ (2015)

Atkinson, M.T., Gucukoglu, S., Machin, C.H., Lawrence, A.E.: Making the mainstream accessible: redefining the game. In: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames, Boston, MA, 30–31 July, pp. 21–28. ACM, New York (2006)

Beksa, J., Fizek, S., Carter, P.: Audio games: investigation of the potential through prototype development. In: A Multimodal End-2-End Approach to Accessible Computing. Springer, London (2015)

Beksa J., Garkavenko A., Fizek S., Vodanovich S., Carter P.: Adapting videogame interfaces for the visually impaired: a case study of audio game hub. In: 25th International Conference on Information Systems Development, Poland (2016)

Brammer, J.: Audio screen: Unsighted game mechanics for mobile devices. Doctoral dissertation, University of Washington (2016)

Brieger, S.: Sound hunter: Developing a navigational hrtf-based audio game for people with visual impairments. In: Proceedings of 2013 the Sound and Music Computer Conference (2013)

Consalvo, M.: Using your friends: social mechanics in social games. In: Proceedings of the 6th International Conference on Foundations of Digital Games. ACM (2011)

Csapó, Á., Wersényi, G., Nagy, H., Stockman, T.: A survey of assistive technologies and applications for blind users on mobile platforms: a review and foundation for research. J. Multimodal User Interfaces. **9**(4), 275–286 (2015)

Denham, J., McComas, H.: Not just playing around: a review of accessible windows-based games. Access World Mag **5**(5), American Foundation for the Blind (2004)

Dwyer, P., VanLund, P.: The last crusade. https://www.audiogames.net/page/news_55 (2008). Accessed Dec 2017

Fizek, S., Woletz, J.D., Beksa, J.: Playing with sound and gesture in digital audio games. In: Proceedings of 2015 Man and Computer Conference (2015)

Frank, M., Zotter, F., Sontacchi, A.: Producing 3D audio in ambisonics. In: Proceedings of 57th Audio Engineering Society Conference, Audio Engineering Society (2015)

Glinert, E., Wyse, L.: AudiOdyssey: an accessible video game for both sighted and non-sighted gamers. In: Proceedings of the 2007 conference on Future Play. ACM (2007)

Kramer, G.: Auditory Display: Sonification, Audification, and Auditory Interfaces. Perseus Publishing, New York (1993)

Mangiron, C., Zhang, X.: Game accessibility for the blind: current overview and the potential application of audio description as the way forward. In: Researching Audio Description. Springer, London (2016)

McCrindle, R.J., Symons, D.: Audio space invaders. In: 3rd International Conference on Disability, Virtual Reality and Associated Technologies, Alghero, Italy, 23–25 September (2000)

Oren, M.A.: Speed sonic across the span: building a platform audio game. In: CHI'07 Extended Abstracts on Human Factors in Computing Systems. ACM, New York, NY (2007)

Östblad, P.A., Engström, H., Brusk, J., Backlund, P., Wilhelmsson, U.: Inclusive game design: audio interface in a graphical adventure game. In: Proceedings of the 9th Audio Mostly: A Conference on Interaction with Sound. ACM (2014)

Parker, J.R., Heerema, J.: Audio interaction in computer mediated games. Int. J. Comp. Games Technol. **2008**, 178923 (2008)

Rebok, G.W., Ball, K., Guey, L.T., Jones, R.N., Kim, H.Y., King, J.W., Marsiske, M., Morris, J.N., Tennstedt, S.L., Unverzagt, F.W., Willis, S.L.: Ten-year effects of the advanced cognitive training for independent and vital elderly cognitive training trial on cognition and everyday functioning in older adults. J. Am. Geriatr. Soc. **62**(1), 16–24 (2014)

Rovithis, E.: A classification of audio-based games in terms of sonic gameplay and the introduction of the audio-role-playing-game: Kronos. In: Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound. ACM (2012)

Rovithis, E., Floros, A., Mniestris, A., Grigoriou, N.: Audio games as educational tools: design principles and examples. In: Proceedings of 2014 Games Media Entertainment. IEEE (2014)

Rumsey, F.: Spatial Audio. CRC Press, Boca Raton (2012)

Sánchez, J., Baloian, N., Hassler, T., Hoppe, U.: Audiobattleship: blind learners collaboration through sound. In: CHI'03 Extended Abstracts on Human Factors in Computing Systems, 05–10 April, pp. 798–799. ACM, New York (2003)

Short, D.: Teaching scientific concepts using a virtual world—Minecraft. Teach. Sci. **58**(3), 55 (2012)

Spiel, K., Bertel, S., Heron, M.: Navigation and immersion of blind players in text-based games. Comp. Games J. **3**(2), 130–152 (2014)

Velleman, E., van Tol, R., Huiberts, S., Verwey, H.: 3D shooting games, multimodal games, sound games and more working examples of the future of games for the blind. In: Proceedings of 2004 International Conference on Computers for Handicapped Persons. Springer, Berlin Heidelberg (2004)

Westin, T.: Game accessibility case study: terraformers–a real-time 3D graphic game. In: Proceedings of the 5th International Conference on Disability, Virtual Reality and Associated Technologies, ICDVRAT 2004 (2004)

Westin, T., Bierre, K., Gramenos, D., Hinn, M.: Advances in game accessibility from 2005 to 2010. In: Universal Access in Human-Computer Interaction. Springer, Heidelberg (2011)

Yuan, B., Folmer, E.: Blind hero: enabling guitar hero for the visually impaired. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility. ACM, New York, NY (2008)

Yuan, B., Folmer, E., Harris, F.C.: Game accessibility: a survey. Univ. Access Inf. Soc. **10**(1), 81–100 (2011)