AmbiDice: An Ambient Music Interface for Tabletop Role-Playing Games

Axel Berndt, Simon Waloschek, Aristotelis Hadjakos, Alexander Leemhuis
Center of Music and Film Informatics, University of Music Detmold, Germany
{a.berndt, s.waloschek, a.hadjakos}@cemfi.de, alexander.leemhuis@gmx.de

ABSTRACT

Tabletop role-playing games are a collaborative narrative experience. Throughout gaming sessions, Ambient music and noises are frequently used to enrich and facilitate the narration. With AmbiDice we introduce a tangible interface and music generator specially devised for this application scenario. We detail the technical implementation of the device, the software architecture of the music system (AmbientMusicBox) and the scripting language to compose Ambient music and soundscapes. We presented AmbiDice to experienced players and gained positive feedback and constructive suggestions for further development.

Author Keywords

Generative Ambient Music, Tangible Interaction

ACM Classification

H.5.5 [Methodologies and techniques] Sound and Music Computing, H.5.2 [Human computer interaction (HCI)] Interaction devices.

1. INTRODUCTION

Tabletop role-playing games are a form of collaborative games. Each participant represents a character within a fictional story world. Their actions take place within the boundaries of a formal rule system. This leaves room for improvisation, constantly influencing the direction the game takes. The story is narrated by the moderator (game master) through speech. To enrich this experience, many game masters compile a selection of musical pieces to be used as background ambience during the game. Music serves a similar purpose as it does in films and video games. It supports storytelling, immersion, dramaturgy, and the emotional experience.

This music is mostly taken from film scores or free music archives. In recent years, some role-playing systems and adventures are accompanied by music that is specifically composed for them, e.g., for the Dungeons & Dragons system [8] and some Dark Eye adventures.[1] In terms of music interaction, the game masters simply navigate through playlists. Some dedicated projects offer not just the music for particular types of scenes but also ambient noises and mixing tools.[2] All these tools require the presence of a CD player, laptop computer, smartphone, or media player on the table—tools that often do not fit to the scenario of the role-playing game. They are, hence, detrimental to immersion and are additional effort for game masters. This lead us to considering the dice, which is the primary tool in most (if not all) role-playing games, as musical control mechanism. Rolling the dice is often the trigger for narrative events, the success or failure of player actions, and, hence, as closely related to the storytelling as the music.

A considerable amount of research on computer games can be found in the NIME context, such as [10, 17, 24]. However, traditional board or pen-and-paper games are rarely addressed. Reunion2012 [21] and "Music for 32 chess pieces" [18] generate music based on the game of chess in the context of concerts, improvisations and interactive installations. In this paper, we explore how a musical interface can support tabletop role-playing games. For this purpose, we introduce AmbiDice, a 12-sided dice with integrated electronics. It controls a specially devised ambient music generation engine called AmbientMusicBox. This engine runs on a regular computer that outputs the music and remains hidden in the background.

2. RELATED WORK

Extended Board Gaming

Tangibles have been used in augmented reality and interactive tabletop games. The survey by Thomas [20] provides an overview of augmented gaming, covering both academic and commercial developments. One of the first tangible augmented reality games was PingPongPlus [13], that uses multiple microphones beneath a ping pong table to localize the impact of the ball. Games such as Jumanji Singapore [25] or the game by Ulbricht and Schmalstieg [22] use augmented reality technology but the interaction paradigm is clearly influenced by traditional board and pen-and-paper games: The players sit or stand around a common playing area and use game pieces or dices. However, tangibles in augmented reality gaming can also be used for less traditionally inspired gaming concepts. Ninja on a Plane [7], for instance, lets the player guide a ninja downwards from an initial high position by using boxes or other real-world objects to build a sequence of planes that the ninja can climb down. In interactive tabletop games, tangibles have been used as (passive) game pieces that are tracked with fiducial markers, e.g. Weathergods game [2]. More complex tangibles are featured in the flipper game pOwerball [6] or in Dragon's Cave [16], a game based on the Dungeons & Dragons role-playing game series.

[1] `https://dsa-soundtracks.bandcamp.com`, last access: Jan. 2017.

[2] `http://tabletopaudio.com`, `https://battlebards.com/` `http://asoftmurmur.com`, last access: Jan. 2017.

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

Tangible Music Selection

Tangible music selection interfaces have been describe in literature, e.g. the Tangible jukebox [9] or the Music Wall [12]. The most similar system to that proposed in this paper is the MusicCube [1] system, which lets the user play, change playlists, pause, scroll, change volume and shuffle with a cube-shaped tangible that features a scroll wheel similar to that which was used by the iPod. In contrast to these related approaches, we concentrate on the application area of tabletop role-playing games.

Ambient Music

One of the most frequently referred narrative functions of game music is the mediation of a scene's mood to the players. This links directly to the genre of Ambient music. Music of this genre is capable of establishing and maintaining "a single pervasive atmosphere" by "non-developmental forms, regularly or irregularly repeating events or cycles of events, modal pitch-sets, choice of a few limited parameters for each piece, and a pulse that is sometimes uneven, sometimes 'breathing', and sometimes nonexistent" as Eric Tamm summarizes his analysis of Brian Eno's Ambient music [19]. Holmes characterizes the whole genre as follows "If there is a unifying element in all ambient music it appears to be a continuity of energy that enables a suspension of tension. Like minimalism, contemporary ambient music often relies on a persistent rhythm and slowly evolving wash of sound textures." [11] Many Ambient compositions do not even rely on repetition. They constantly vary, move around, but never leave their character of expression. The impression of timelessness in many Ambient compositions is reinforced by sonic properties.

1. Sounds with bell-like amplitudes (piano, vibraphone, harp, bells etc.), i.e. with a percussive attack and very long release phase, seem to endlessly fade away. These mostly arpeggiate more or less randomly over the underlying pitch-set.

2. Pad sounds (choir, strings, synth pads etc.) with smooth attack and release phases serve mostly as chordal instruments.

3. Ambient noise (nature, urban, synthetic) occur less frequent. In the gaming context this is the virtual scene's soundscape.

Nonlinear Game Music Technologies

A role-playing session is nonlinear. The progress of an interactive scene isunpredictable. How long does a situation last? Music has to wait that same period. Which direction does the story take? Music has to follow. These opposing claims are commonly accomplished by two types of music arrangement, sequential and parallel.

The concept of *sequential arrangement* constantly reorganizes a sequence of musical snippets according to the interactive context. Its root lies in the classic musical dice games [14]. A famous implementation is the *iMuse* engine [15]. *Parallel arrangement* works with the dynamic mixing of a multitrack recording. The audio tracks present different musical material for different interactive situations and are combined by fading. *Parallel composing* is a subspecies of this concept. Here, playback jumps dynamically between different parallel audio tracks [23]. Approaches that combine both arrangement concepts are presented in [5].

Further approaches to dynamic music are variation of the expressive performance (tempo, dynamics, articulation), variation of music on a compositional level (melody variation, reharmonization) and algorithmic composition [3, 4].

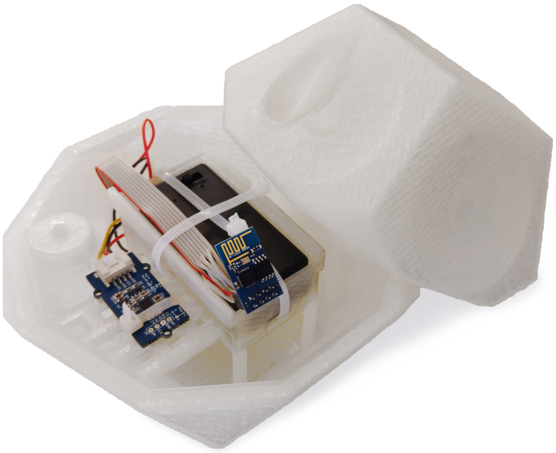

Figure 1: AmbiDice is 3D printed and equipped with electronics (diameter ∼ 17 cm).

Regarding Ambient music, the latter approach is at hand. Simply put, changing the parameters of a realtime music generation will change the music created by it.

3. THE DICE

AmbiDice consists of two separate parts: the dice with its built-in electronics (see figure 1) and a Java-based server. Data acquisition is based on an ESP8266 microcontroller that is Arduino/C++ compatible and has WiFi functionality on board. In order to detect the upwards facing side of the dice, an accelerometer connected via I^2C is used.

The user can connect to a WiFi access point that the dice provides. The raw sensor values are then transmitted to all connected clients as broadcast UDP packets. These messages are received by the server software on the connected computers which then compares the received 3d vector with 12 reference vectors (one for each face) and calculates the correct orientation by maximizing the cosine similarity over all faces. False calculations that might appear during the actual throw of the dice are effectively minimized by averaging the incoming 3D vector over multiple sensor values. Once the upward pointing face is determined, the corresponding method in the AmbientMusicBox is executed.

4. AMBIENT MUSIC GENERATION

Audio output is generated by AmbientMusicBox, a self-contained software written in Java. It is responsible for both, reatime music generation and ambient noise mixing.

4.1 AmbientMusicBox Architecture

The sound synthesis of AmbientMusicBox utilizes the JSyn audio synthesis library. [3] The AmbientMusicBox API is designed for simple integration into other software projects. The system architecture and API is shown in figure 2.

Applications load music scripts (XML files), trigger playback and recording, and adapt multi-channel mixing. Three types of channels are implemented: realitme synthesis channels, audio file channels (e.g. for playing back prerecorded ambient noises), and a specialized wind synthesis channel (a frequently used ambient noise; different types of wind/storm can be achieved by its parameters `howlStrength` and `howlSpeed`). Under the hood works a sequencer that reads and performs the music scripts in realtime, controls the mix of

[3] `https://github.com/philburk/jsyn`, last access: Jan. 2017.

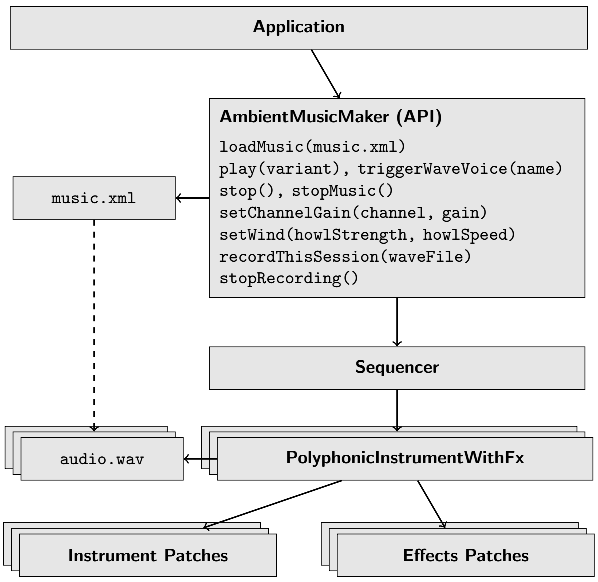
Application
AmbientMusicMaker (API)
loadMusic(music.xml)
play(variant), triggerWaveVoice(name)
stop(), stopMusic()
setChannelGain(channel, gain)
setWind(howlStrength, howlSpeed)
recordThisSession(waveFile)
stopRecording()
music.xml
Sequencer
audio.wav
PolyphonicInstrumentWithFx
Instrument Patches
Effects Patches

Figure 2: The AmbientMusicBox architecture.

all channels and accesses the sound synthesis. Jsyn's real-time scheduling mechanism is utilized to ensure that each musical event is performed on time. AmbientMusicBox offers a number of ready-to-use synthesis and effects patches (Java/JSyn classes). Developers may add further patches.

4.2 The Scripting Language

The AmbientMusicBox scripting language is an XML-based format that lets the composers specify instruments, effects, audio files, wind synthesis, and assign them to mixer channels. It further offers the routines to define the procedural music, particularly dedicated to the peculiarities of Ambient music. The root node `<music/>` holds one or more `<voice/>` elements that represent individual instruments. The structure of a `voice` element is as follows.

```
<voice channelLeft="" channelRight=""
       name="" instrument="" polyphony=""
       fadeSpeed="" relativeMix=""/>
```

Each instrument and effects patch can be set mono or stereo. In case of a mono output the attributes `channelLeft` and `channelRight` are replaced by attribute `channel`. Attribute `name` specifies an id that can be used to address the voices. Attribute `instrument` indicates the synthesis patch to be used for this voice, `polyphony` specifies its polyphony, i.e. the number of copies to be instantiated. With `fadeSpeed` volume changes (usually triggered by the application) can be ramped, `relativeMix` sets an initial volume gain (typically between `0.0` and `1.0`). Each voice can have an optional child element `<Fx/>` whose child element refers to an effects patch to be loaded and its initial parameter settings.

If `instrument="wav"`, the only child element is `<wav uri="" loop=""/>`. Attribute `uri` denotes a wave file to be loaded. If `instrument="wind"`, the only child element is `<wind howlStrength="" howlSpeed=""/>`.

All other voices are synthesis instruments. Their child elements (apart from `Fx`) are of type `<variant name="" relativeMix=""/>`. Each variant defines a musical instance. If the application calls `play("Adagio")`, all voices that have a variant named `"Adagio"` will start playing it. Attribute `relativeMix` sets the volume level of the voice for this variant. Besides the possibilities to set instrument and effects parameters, variants specify the music to be played, typically by defining sequences, such as the following example.

```
<sequence loop="inf" maxStartDelay.seconds="10">
    <note pitch.midi="70" velocity="0.4"/>
    <rest dur.seconds="4"/>
    <note pitch.midi="75" velocity="0.3"/>
    <note pitch.midi="71" velocity="0.5"
          dur.seconds="25"
          dur.variation="0.025"/>
    <rest dur.seconds="4"
          dur.variation="0.025"/>
</sequence>
```

If a variant contains more than one sequence, they are all performed simultaneously. Attribute `loop` is set to a non-negative integer value or `inf` to specify how often the sequence repeats. This accounts for the typical approach of composing Ambient music from event cycles. With attribute `maxStartDelay.seconds` the beginning of this sequence can be delayed by a random float value not greater than the given value (10 seconds in the above example).

The basic building blocks of sequences are `note` and `rest` elements. Notes are specified by pitch and velocity. Duration is optional. For rests, duration is mandatory. Durations are specified in seconds. This seems unintuitive in a musical context, but a peculiarity of many Ambient compositions is the absence of musical meter in a traditional sense. Here, it is easier to work with absolute timing. Durations may also be subject to random variation which is specified by attribute `dur.variation` that defines a maximum variation radius (positive and negative). Further important to have in mind is the fact that rests specify inter-onset intervals. This means, the second and third note element in the above example are played at the same time (as chord).

Sequences may contain further sequences. These can be played in succession (by using rests) or simultaneously. The fourth possible type of child elements in a sequence is `<procedure/>` in one of the following three variants.

```
<procedure mode="random choice"/>
<procedure mode="permutation"
           numberOfPermutations=""/>
<procedure mode="permutation sequence"
           numberOfPermutations=""/>
```

They may contain notes, rests, sequences, and procedures.

`random choice`: preforms one of the child elements.

`permutation`: When the procedure is processed the first time, the first child element is performed, next time the second, and so on. When all are performed, the given `numberOfPermutations` is applied to the series of children and playback starts anew with the first.

`permutation sequence`: The procedure should have only one child of type `sequence`. The first time the sequence is played as is. From then on, the given `numberOfPermutations` is applied to all non-rest children of the sequence before it is performed again.

4.3 Discussion

The scripting language comprises only relatively few very basic building blocks with an accordingly low learning hump. Through combination and nesting (e.g., procedures within procedures) more complex mechanisms can be created and a huge bandwidth of possibilities unfolds. Nonetheless, the procedure formalism is a potential candidate for future extensions.

So far, we have prioritized absolute timing (in seconds) over symbolic timing, such as musical meter or MIDI ticks. This accounts for a peculiarity in many Ambient styles. However, in other situations a clear meter is more useful. The rhythmic coupling of multiple voices would be easier. Advanced coupling mechanisms are a further candidate for future extensions, e.g. voices waiting for each other at certain synchronization points.

Finally, synthesis parameter automation (other than mixing, pitch and velocity) is not implemented so far. A more complex work with timbre requires specialized formalisms.

5. EXPERIENCES & FUTURE WORK

To get feedback on the AmbiDice system, we visited a local tabletop role-playing game group. The group meets in two-weekly intervals. During the first hour, before they start playing, we presented the AmbiDice system and animated a group discussion about the usefulness of the system and possible new directions. Six players were present. Two of them regularly act as game master.

Assessment: The players welcomed AmbiDice with enthusiasm. Both game masters use music regularly in their games. They usually prepare playlists on their laptops and switch tracks in the media player. They would, however, prefer using AmbiDice for music selection. All players agreed that the new interaction is less distracting and more adequate for the gaming context. The group was interested to obtain an AmbiDice for their regular gatherings and asked for eventual plans to commercialize the system or to open source the design. A 12-sided dice was deemed optimal, 10 sides would also be acceptable while six sides were deemed to be too little and too conventional.

Ideas and future directions: The size of a fist was deemed optimal for dice rolling (e.g. to announce the result of a random encounter by dramatic or happy music) while it is still big enough to indicate its special role. We also discussed the technical setup. Instead of relying on a laptop computer, the AmbiDice should ideally be either a self-contained device with internal speaker or provide an easy-to-use interface, to connect to smartphones (to prepare the playlist) and Bluetooth loudspeakers.

6. REFERENCES

[1] M. B. Alonso and D. V. Keyson. Musiccube: making digital music tangible. In *CHI'05 extended abstracts*, pages 1176–1179. ACM, 2005.

[2] S. Bakker, D. Vorstenbosch, E. van den Hoven, G. Hollemans, and T. Bergman. Tangible interaction in tabletop games: studying iconic and symbolic play pieces. In *ACE'07*, pages 163–170. ACM, 2007.

[3] A. Berndt. Musical Nonlinearity in Interactive Narrative Environments. In G. Scavone, V. Verfaille, and A. da Silva, editors, *ICMC 2009*, pages 355–358, Montréal, Canada, Aug. 2009. McGill University.

[4] A. Berndt, R. Dachselt, and R. Groh. A Survey of Variation Techniques for Repetitive Games Music. In *Audio Mostly 2012*, pages 61–67, Corfu, Greece, Sept. 2012. Ionian Academy, ACM SIGCHI.

[5] A. Berndt, K. Hartmann, N. Röber, and M. Masuch. Composition and Arrangement Techniques for Music in Interactive Immersive Environments. In *Audio Mostly 2006*, pages 53–59, Piteå, Sweden, Oct. 2006. Interactive Institute/Sonic Studio.

[6] B. Brederode, P. Markopoulos, M. Gielen, A. Vermeeren, and H. De Ridder. pOwerball: The design of a novel mixed-reality game for children with mixed abilities. In *Interaction Design and Children 2005*, pages 32–39. ACM, 2005.

[7] D. Chekhlov, A. Gee, A. Calway, and W. Mayol-Cuevas. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *6th IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pages 1–4. IEEE Computer Society, 2007.

[8] E. Douglas and G. Goszka. Dungeons & Dragons: Official Roleplaying Soundtrack. Entity Productions Darkcell, 2003.

[9] D. Gallardo and S. Jordà. Tangible jukebox: back to palpable music. In *TEI*, pages 199–202. ACM, 2010.

[10] F. Grani, R. Paisa, J. Banas, I. Vogiatzoglou, and S. Serafin. Design and evaluation of a gesture driven wave field synthesis auditory game. In *NIME 2016*, 2220-4806, pages 188–193, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[11] T. Holmes. *Electronic and Experimental Music: Technology, Music, and Culture*. Routledge/Taylor & Francis, New York, NY, USA, 4th edition, Feb. 2012.

[12] C. Hu, K. Tung, and L. Lau. Music wall: a tangible user interface using tapping as an interactive technique. In *APCHI*, pages 284–291. Springer, 2008.

[13] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso. Pingpongplus: design of an athletic-tangible interface for computer-supported cooperative play. In *CHI*, pages 394–401. ACM, 1999.

[14] J. P. Kirnberger. *Der allezeit fertige Polonaisen und Menuetten Komponist*. G. L. Winter, Berlin, Germany, 1767.

[15] M. Z. Land and P. N. McConnell. Method and apparatus for dynamically composing music and sound effects using a computer entertainment system. United States Patent Nr. 5,315,057, May 1994.

[16] J. Marco, S. Baldassarri, and E. Cerezo. Toyvision: a toolkit to support the creation of innovative board-games with tangible interaction. In *TEI*, pages 291–298. ACM, 2013.

[17] B. Olson. Transforming 8-bit video games into musical interfaces via reverse engineering and augmentation. In *NIME 2016*, 2220-4806, pages 73–77, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[18] D. Parson and P. Reed. The planetarium as a musical instrument. In *NIME 2012*, Ann Arbor, Michigan, 2012. University of Michigan.

[19] E. Tamm. *Brian Eno: His Music And The Vertical Color Of Sound*. Da Capo Press, Boston, MA, 1995.

[20] B. H. Thomas. A survey of visual, mixed, and augmented reality gaming. *CIE*, 10(1):3, 2012.

[21] A. Tveit, H. Wilmers, N. Thelle, M. Bugge, T. Johansen, and E. M. Sæther. Reunion2012: A Novel Interface for Sound Producing Actions Through the Game of Chess. In *NIME 2014*, pages 561–564, London, UK, 2014. Goldsmiths, University of London.

[22] C. Ulbricht and D. Schmalstieg. Tangible augmented reality for computer games. In *VIIP*, pages 950–954, 2003.

[23] T. van Geelen. Realizing groundbreaking adaptive music. In K. Collins, editor, *From Pac-Man to Pop Music: Interactive Audio in Games and New Media*, chapter 6, pages 94–102. Ashgate, Hapshire, England, 2008.

[24] G. Wang. Game design for expressive mobile music. In *NIME*, 2220-4806, pages 182–187, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[25] Z. Zhou, A. D. Cheok, T. Chan, and Y. Li. Jumanji singapore: an interactive 3d board game turning hollywood fantasy into reality. In *ACE*, pages 362–363. ACM, 2004.