BeatTheBeat
Music-Based Procedural Content Generation
In a Mobile Game

Annika Jordan, Dimitri Scheftelowitsch, Jan Lahni, Jannic Hartwecker, Matthias Kuchem, Mirko Walter-Huber, Nils Vortmeier, Tim Delbrügger, Ümit Güler, Igor Vatolkin, Mike Preuss

Computational Intelligence Group, Technische Universität Dortmund, Germany
Contents

1. Introduction
2. Music Features
3. Game Concept
4. Self Organizing Map Use
5. Minigames
6. User Study
7. Take Home Message
Introduction

influences:
- music games
- mobile games
- PCG

usual limitations of (most) music games:
- fixed sound track
- very limited interaction with music (e.g. tap beats)
- single (sequential) play

idea: use device music as source for adapting the game
main objectives of this work:

- use music via (automated) feature extraction as source for game content creation, beyond simple reactive schemes
- establish well-structured board compositions via self-organized map, based on automatically derived features
- music-based personalization of game should be recognized by users, without being trivial (⇒ user study)
Types of Music Features

music features can be obtained from different sources:

- compute from the audio signal: e.g. filter predefined frequency ranges
- symbolic features generated from the score: key, instruments, harmonic characteristics
- metadata from manually entered information: year, genre
- user tagging (as in last.fm) or similarity-based playlist characteristics

we use only the first as it is universally applicable, even for completely new music (could be composed by user during a game)
Our Feature Base

- root mean square provides a rough estimation of signal energy
- sub-band energy ratios (4 frequency bands)
- sensory roughness: average dissonance between all pairs of peaks of the spectrum
- average angle in phase domain, this is useful to distinguish between strong and weak percussive pulses
- beat per minute (BPM) measures tempo
we add 2 more intuitive features as combinations of base features:

- bass energy (applies to your stomach)
- rhythm event (as unintentional moves, beats & loudness)
many players share a board of minigames
- each field stands for one song and one minigame (fixed)
- minigames are adjusted to 'their' song
- fields are controlled by achieving minigame high score
- overall points are computed from number of connected owned fields
SOM: Setting up the Board

- colors from 5-means clustering (for visualization)
- left: after SOM training (some nodes connected to multiple songs)
- right: after assignment phase (correction heuristics applied)
Minigames: Music Fighter

simple shooter with enemies resembling properties of chosen song
- game runs as long as song
- 5 different types of enemies
- enemy is generated when rhythm event occurs, up to 10
- bass feature level changes enemy behavior
- player cannot die but loses time when hit too often
Minigames: Tap by Tap

tapping game that uses rhythm events
- relatively similar to existing tapping games
- increasing difficulty over time (tapping plus wiping into given direction)
- game ends with song or if 3 balls are missed
Minigames: Music Tower Defense

tower defense game with music-sensitive towers

- game runs until song ends or too many enemies reach exit
- available towers are sensitive to different music features (rhythm event, bass feature, sensory roughness)
- successful tower building has to take song properties into account
- player can increase difficulty (and score) by hitting 'new wave' button
User Study on Typ by Tap

- experimental setup: each user plays 3 songs of tap by tap (only first minute) in two versions
- one version uses correct features, one randomly selected ones (of 82 songs)
- user tries to recognize which feature set is correct

<table>
<thead>
<tr>
<th></th>
<th>knows music well</th>
<th>users</th>
<th>correct</th>
<th>wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes yes</td>
<td>yes</td>
<td>7</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>yes no</td>
<td>no</td>
<td>7</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>no yes</td>
<td>yes</td>
<td>3</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>no no</td>
<td>no</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>all all</td>
<td>all</td>
<td>20</td>
<td>39</td>
<td>21</td>
</tr>
</tbody>
</table>
Lessons Learned...

- small study, so results to be considered with care
- people who neither listen to much music nor play games seem to make random decisions
- all other groups have around 2/3 chance for detecting the right parameter set
- this is actually not that bad, it should not be 100%
Take Home Message

- music is an interesting generational data source for PCG
- SOMs are good for easily create game boards out of dynamic components (e.g. songs)
- programming on different mobile devices is a HUGE difference!
- copyright issues make working with songs difficult
- open: how to play together on different individual boards?