

dragica kahlina ([www.kahlina.com](http://www.kahlina.com)) sound artist + coder

game design  
and  
procedural sound

# Disclaimer

this is not going to be an in-depth introduction of  
procedural sound

but

use of procedural sound for game design

(I will give you some pointers, where to start at the  
end of the talk)

this will hopefully be

an open discussion

and

an exchange of ideas

feel free to interrupt me with  
questions, ideas, your own experiences

# definition ?

## procedural

- dynamic
- synthetic
- generated on the fly

## Negative:

- coded
- CPU consuming
- somewhat

Terra Incognita

## vs

- sampled / produced
- controllable
- human touch

## Negative:

- “fixed”
- needs resources
- memory consuming
- production standards

my music ...

<http://youtu.be/X2hGVfC9XqU>

... to give you some context

but

I am no purist

mix and match,  
use what works

# do

- use procedural sounds if you have the coder
- use composed music pieces for emotional impact
- invite other musicians to contribute pieces for variety
- don't reinvent the wheel (more on that later)

the story



wotokah



# the problems

- No money  
→ no time
- “the team”
  - Janina : level designer
  - Sebastian : script coder
  - Dragica : sound artist (← coder)

=> the motto : make do

# the pluses

- the team ;-)
  - all more or less generalists
  - open minded
  - driven
  - we know what we are doing

# the given

- Unity3d
- Oculus VR headset

# open world (minus)

- can't be static, needs lots of assets to simulate life and we technically have no 3d artist
  - how do we fill this with quest, points of interest (where is our story writer, quest designer)
  - how do we guide the player, animate her to explore, but not get her lost and bored
- but : make do (and yes we are crazy)

# horror game (plus)

- the less you see, the scarier it gets
- we can get away with night only
- you can't run away, look away using a headset (yes, we are evil)
- plus for me
  - horror means weird, slightly crazy, disturbing sounds
  - there are lots of sounds

build on strengths

fantastic level to explore

but

use sounds instead of graphics

for

interest, life, quest

# synergies

- environment sounds to give world an ambience
- creature sounds, with a “life cycle” to give level life
- horror sounds to control the fear level
- disturbing sounds to unbalance the player
- use music like a horror sound
- dynamic player sounds
- dynamic monster sounds

and  
everything  
can be influenced  
by everything else



# examples

- ambience gets darker as quest progresses
- ambience ramps up or down to push player fear level or cool down
- monster sounds change continuously
  - to show its emotions like anger, frustration
  - depending on player parameter like nearness, health, quest

# the ex AI coders POV

everything (ok almost everything)

can be solved by an AI

(or more accurate a FSM)

# HAI (Horror AI)

- really simple FSM that controls how dark the sounds get and how much scary stuff happens
- influenced by:
  - random
  - time
  - how much already happened
  - part of the level
  - point in the quest
  - monster

# ambient, disturbing, shock sounds

- they work essentially the same and use the same basic FSM
- are controlled by
  - HAI
  - proximity of the player
- only difference are :
  - continuous vs discrete events
  - local versus global

# creature + monster sound

- basic Enemy / NPC AI
- not really special

# music

- basically a FSM
- use short music pieces and chain them together (Xenakis)
- use permutations, reverse, transpose (sorry I still think in math not music)

# BUT

to make it really dynamic  
we need to make our sound ourselves  
not just intelligently  
control sampled sounds

# so music needs an instrument ...

- always envisioned a lonely flute, at some times almost wind like
- and sometimes some ghost folk dance music, strings, drums, voices
- lucky me, this is all doable
  - strings and flute can be synthesized quite easy by physical modeling (Karplus-Strong)
  - as can be drums



# and the rest

- can be synthesized too
- there are tons of webpages that tell you how to do it (see [Procedural Sound in Unity](#) )
- there is a book with usable sounds

# where do I start

- get up to date on different synths
  - additive (not so interesting)
  - subtractive (& filters)
  - fm & am
  - granular (& granulation)
  - physical modelling
    - search sound synthesis or dsp
    - Martin Neukom: Signale, Systeme und Klangsynthese (only German)

# and

- look at composers like
  - Iannis Xenakis : Markov chains
  - Arnold Schönberg : twelve-tone technique
  - Karlheinz Stockhausen
  - Pierre Boulez
- composition techniques
  - use of “game of life” or automata (New Kind of Science)
  - fractals, attractors, L-Systems
  - generative music
  - evolving music

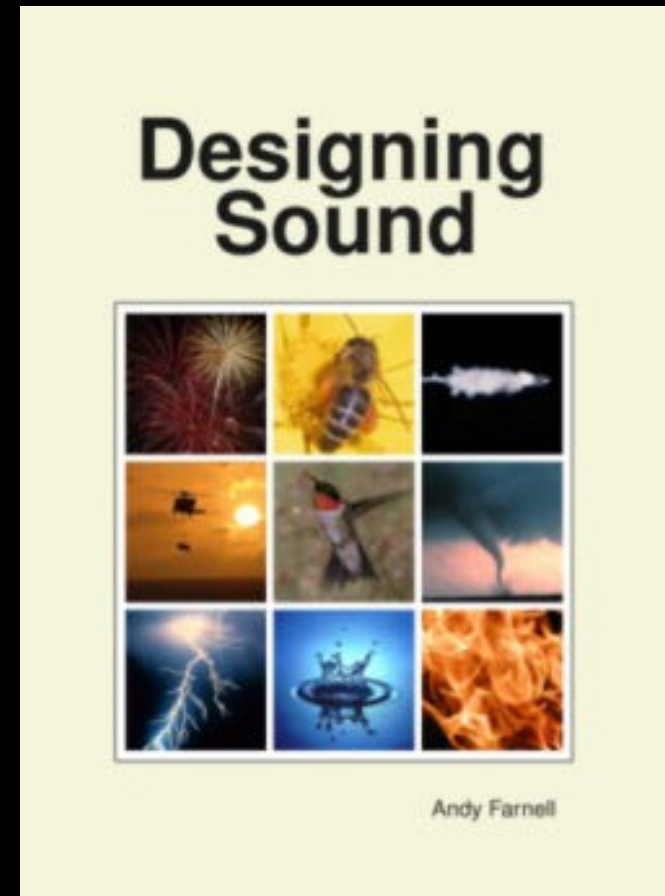
# software to learn & prototype

- CSound (free, C-like, older style)
- Pure Data (free, visual programming)
- Max/MSP (paid, visual programming, can be embedded in Ableton Live 9)
- Supercollider (free, more coding like)
  - overtone
- pwgl, open music, chuck ...
- good old C++ (look at the STK synthesis toolkit)

# usable procedural sound

<http://obiwannabe.co.uk/>

- has tons of examples
- uses PD (use libpd or try to convert it)
- don't reinvent the wheel



# Example footsteps

- Book
- PD patch
- in Unity

# and now ...

- prototypes → implement into game
- refine
- make it a work all together

wotokah.makegames.ch





# Questions ?