GAMEMUSIC ANDSOUND

ADAM J SPORKA

DCGI/CTU PRAGUE

PHA 2014/15

MUSIC

Outline

- Purpose of music in games
- Static music soundtracks
  - – Music design
- Dynamic music soundtracks
  - – Approaches, management
- Music production technology
  - – How the actual music is made

Purpose of Music

- Tradition
  - – Present in most of the released titles
  - – Why not in yours?

- Affirmation of the genre
  - – “Yes, this is an 8bit retro.”
  - – “Indeed, cowboys in spaceships.”
  - – “Yay, manga.”

Purpose of Music

- Presenting the emotion
  - Easygoing, happy-go-lucky, …
  - Depressing, sad, …
- Setting the expectations
  - Level difficulty
- Status monitoring
  - Changes of music over time indicate changes in game

Purpose of Music

- Essence of the gameplay
  - Dance Dance Revolution (1998)
  - Guitar Hero (2005)
  - Pugs Luv Beats (2011)
    - http://www.youtube.com/watch?v=V0i18_--8Yc
  - *Not covered by this talk*

Game Music Components

- Theme Music
  - – Often synchronized with visual events on the screen
  - – Cinematic experience
- Underscore
  - – In-game music
  - – User interface background

Game Music Components

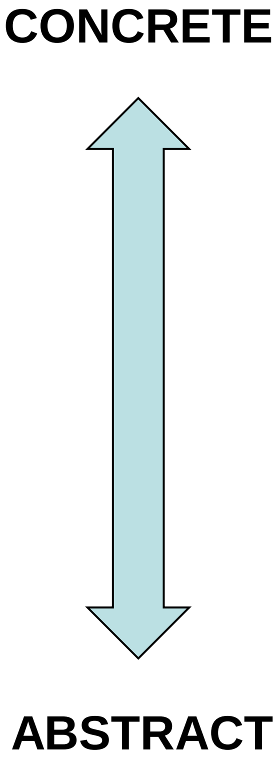
- Stingers
  - Specific music triggered by an event
  - Transition music
  - Level Notification, “Game Over” Screen, “Game Won” Screen

  - Fallout New Vegas:
    Pulling out / holstering Magnum
    - http://youtu.be/sWLpSpZR6J4?t=41s

Soundtrack

- Diegetic sounds
  - Sounds in story space
  - On-screen or off-screen
  - Voices of characters, sounds of objects, music coming from objects
- Non-diegetic sounds
  - Not implied to be present in the action
  - Narrator, sounds for dramatic effect, underscore

Sound in Games



CONCRETE

ABSTRACT

- Voices, diegetic sound effects
- Diegetic ambience
- Non-diegetic ambience
- Underscore

Game Music Characteristics

- Theme music is feature
  - Recognizable theme
  - Elaborate
  - Genre announcer
  - Everyone will hear this

- Underscore is background
  - Mood setter
  - Does not distract
  - Can be listened to throughout the gameplay

Game Music Characteristics

- Silence
  - "Dosage" of music
    - Used to emphasize music
  - Keep the ambient sounds present
  - Complete silence → "something wrong"
    - "Are my speakers on? Will I hear anything at all when I'll really have to?"

Game Music Interactivity

- Static soundtrack
  - Predefined and unchanging
  - Loops
  - Assigned to specific screens / levels

- Dynamic (adaptive) soundtrack
  - More complex control of the music playback
  - Engine “aware” of the state of the game

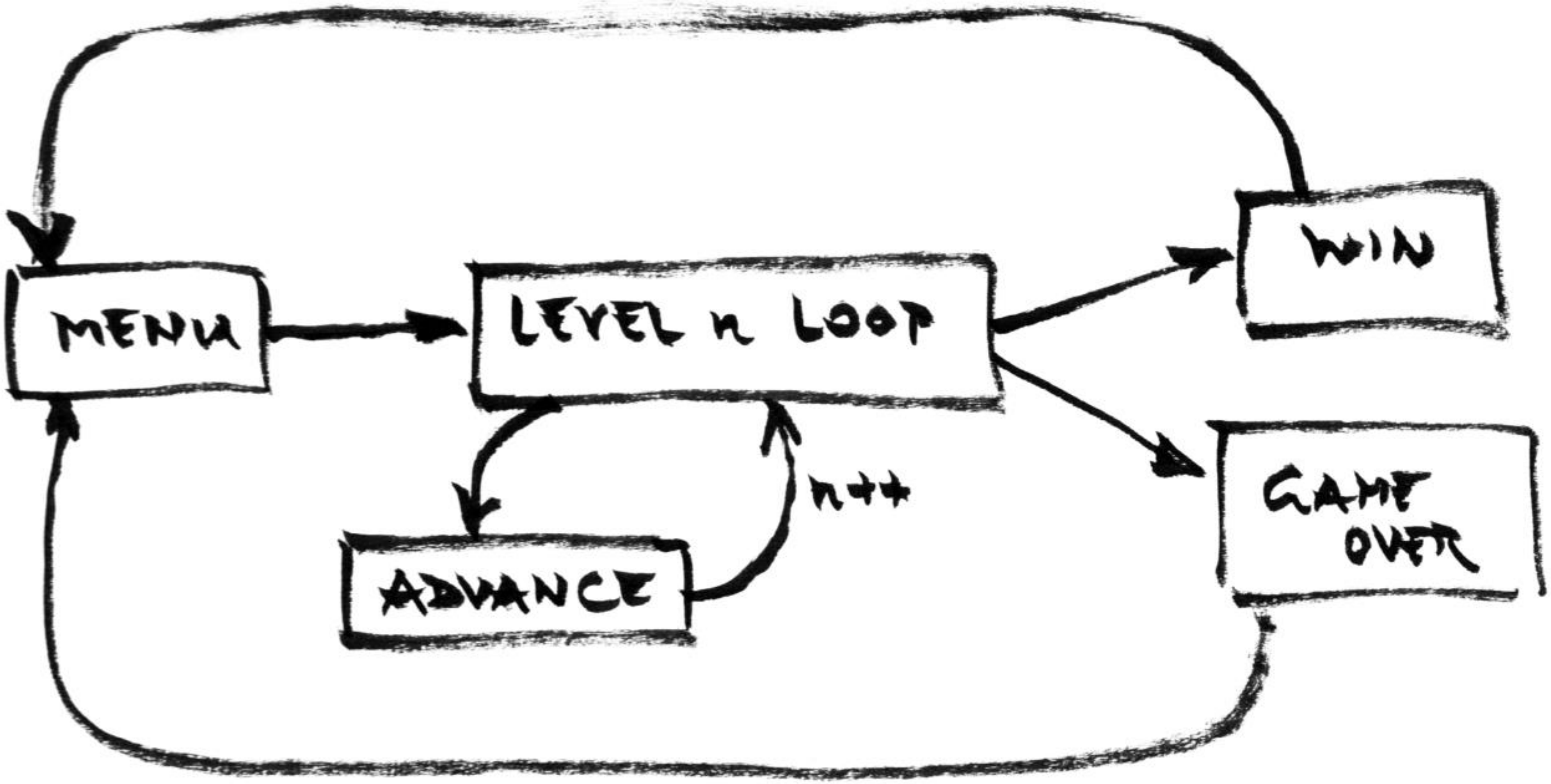
Static Soundtrack

- Early and simple games
- Pengon (1984, Atari 800XL)
  - http://www.youtube.com/watch?v=MDhLxRLvwvY
  - One music loop
  - Game over stinger

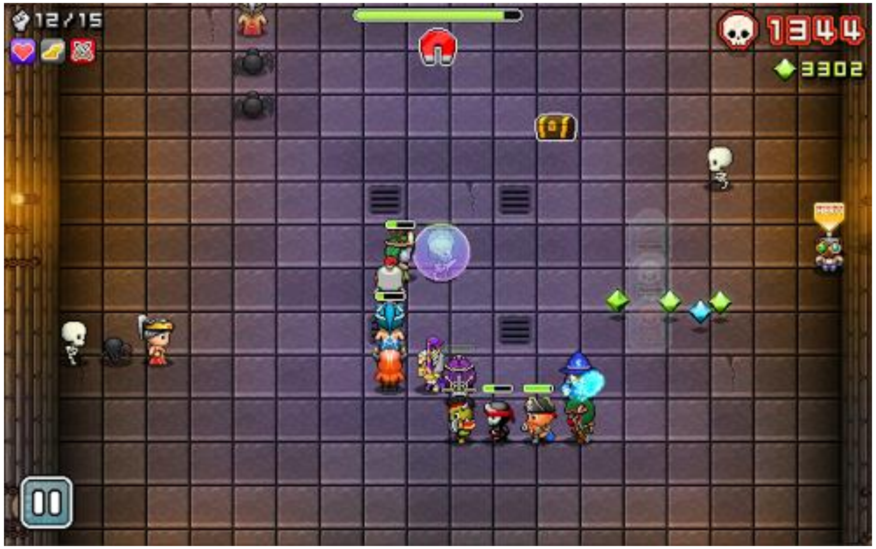
Static Soundtrack

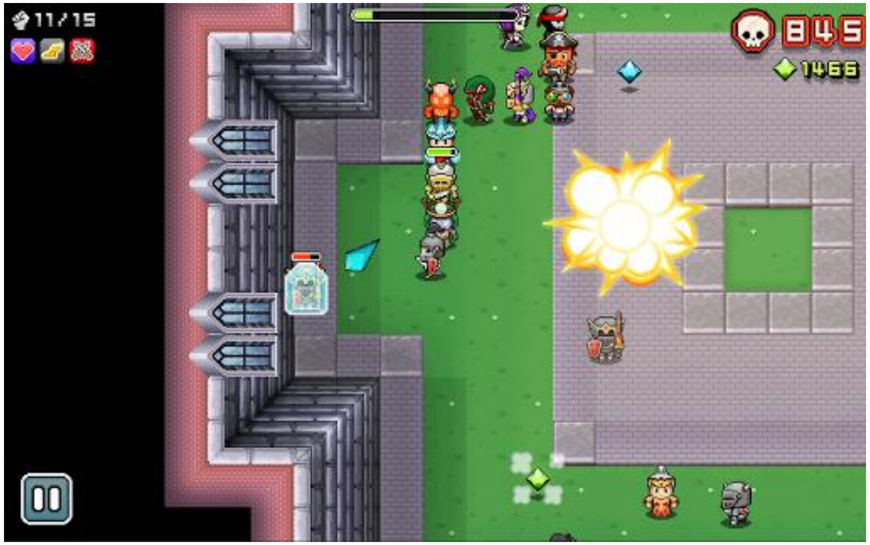
- Supaplex (1991, MS-DOS)
  - Played throughout the game
  - http://www.youtube.com/watch?v=yknubWX2KYI
- Goonies (1985, Atari 800XL)
  - Two loops, one used for title, then alternating with each level
- Jazz Jackrabbit (1994)
  - Separate tracks for levels

Static Soundtrack



Case Study: Nimble Quest





- 2013; NimbleBit; iOS, Android
- Arcade / RPG
- Early 2000s pixel art graphics

Case Study: Nimble Quest

- Music by Whitaker Blackall
- Contents:
  - Title / menu
  - Loop per level
- Genre:
  - Contemporary chiptune
  - http://wtrebella.bandcamp.com/album/nimble-quest

Case Study: Nimble Quest

- Typical length of a loop: 2’
- Unobtrusive
- Reflects the increasing difficulty of the levels
  - – L1 http://www.youtube.com/watch?v=6mh7qJgnpJI
  - – L2 http://www.youtube.com/watch?v=2hAtghKVK7I
  - – L4 http://www.youtube.com/watch?v=YKIDg8VkF0A

Case Study: Nimble Quest

- Sounds good on small speakers
  - – Chiptune sounds
  - – Lot of square waves / triangles
    - Better estimation of pitch by the human low- as well as high-pitched tones

- General remark:
  - PC gamers often don’t spend money on a high-quality speakers

Case Study: Quido (2013)

- Experimental arcade game
  - Developed at DCGI/CTU
- Showcasing non-standard haptic modalities
  - Exhibition item on Designblok 2013
- http://ulab.cz/naviterier/quido/
- Genre:
  - Contemporary chiptune
  - https://quido.bandcamp.com/

Quido: Technology used

- Reaper
- VSTs
  - SuperWave (leads / chords / bass)
  - EXD-80 (beats)
  - Some reverb, some volume compressor

Quido: Music

- Theme music
  - Exposition of the main theme
  - Happy, easygoing
  - Not used in the end
    - Explicit request from the exhibition organizer: Silence between the games

Quido: Music

- Level 1: Theme Park
  - A player will be likely to be receiving instructions from a fellow player
  - Expected playtime: 3—4 minutes
  - Not “dense”
  - Diverse in terms of the structure
  - Performance of the main theme

Quido: Music

- Level 2: City
  - Variation of the theme.
  - Structure: [A B A B$_{br}$ A$_{void}$]
  - Expected playtime: 2 minutes
  - Continuous music

Quido: Music

- Level 3: Sky (Heaven)
  - Variation of the theme. Barroque.
  - Continuous music
  - Expected playtime: 2 minutes

Quido: Music

- Level completed
  - Joyous jingle
- Advancing to the next level
  - Expectation

Quido: Music

- Game Over
  - Definitive
  - Not too tragic

Quido: Music

- Game won
  - – Player enters their initials
  - – Reiteration of the main theme

Adaptive Music Styles

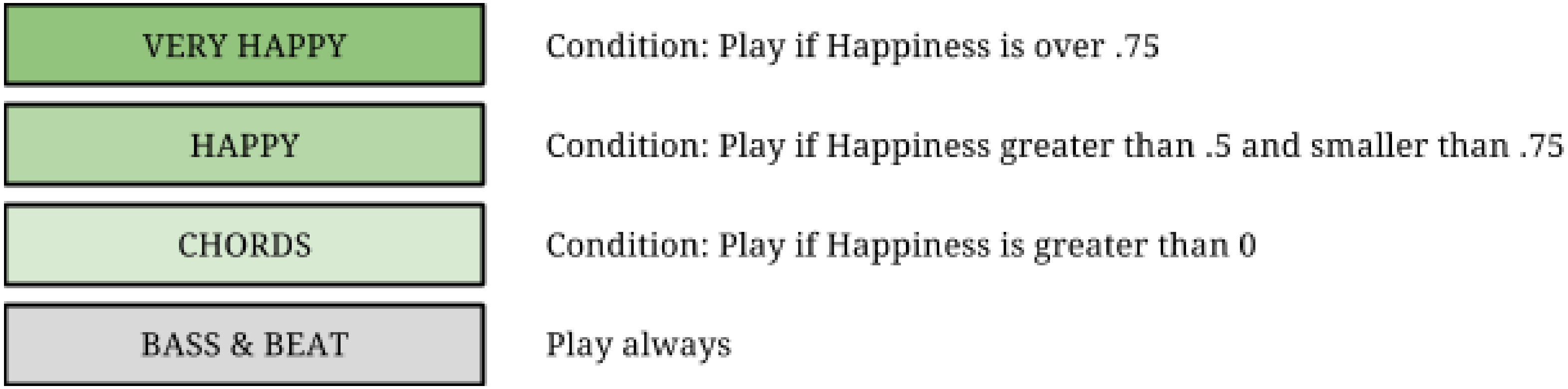
- Resequencing
  - Selecting what to play next
- Reorchestration
  - Selecting which instruments to play
- Modulation
  - Modifying the timbres of notes
  - Related to Data Sonification

The Space Game

- Game loops:
  - Menu / Idle
  - The enemy is approaching
  - The enemy opens fire

- <demo>
  - http://www.candystand.com/play/the-space-game

Reorchestration



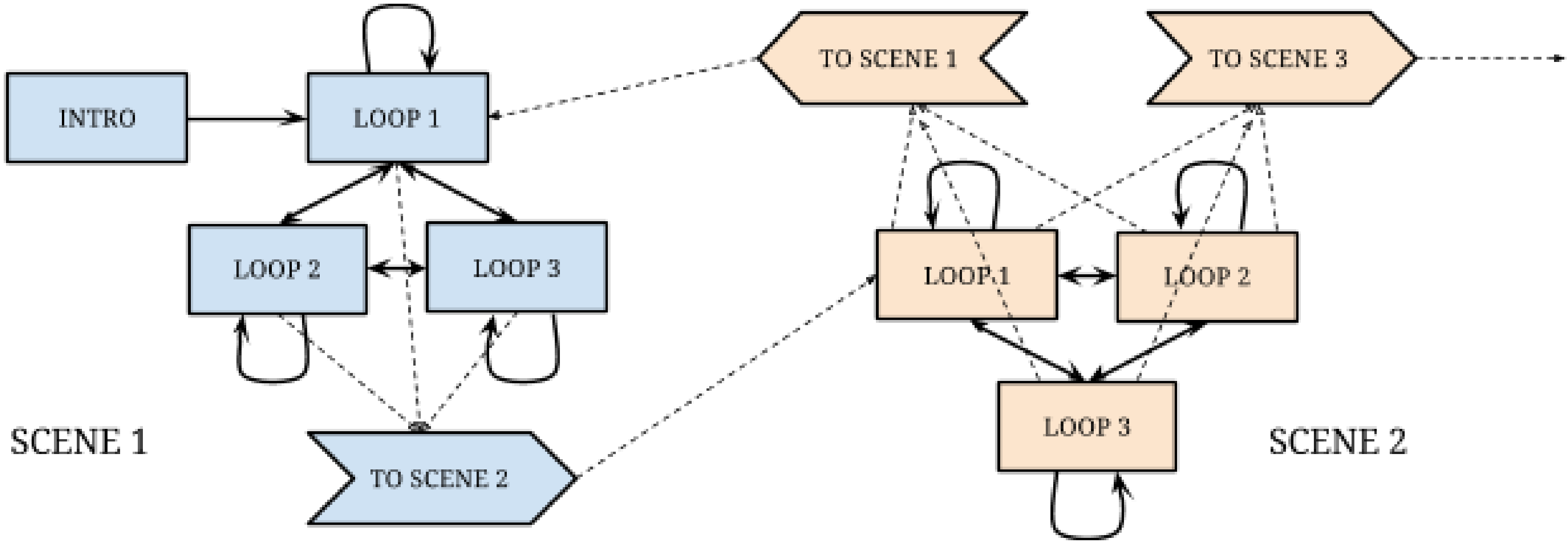
VERY HAPPY	Condition: Play if Happiness is over .75
HAPPY	Condition: Play if Happiness greater than .5 and smaller than .75
CHORDS	Condition: Play if Happiness is greater than 0
BASS & BEAT	Play always

<Demo 1: Quido reorchestration>

Reorchestration

- Fallout: New Vegas
  - 2010, Obsidian Entertainment, Bethesda Softworks
  - Three renderings of the same track
    - “low”, “mid”, “high”
  - <demo; NewCaliforniaRepublic>

Resequencing



Resequencing

- Slices chosen according to game state
- Precise timing of transitions

- <demo>

Resequencing and Reorchestration

- Example: Monkey Island 2
  - 1991, LucasArts
  - http://www.youtube.com/watch?v=Nsc5nTrCzmw#t=322
  - iMUSE system
    - Parallel tracks for different locations
    - Transition patterns to return back

Modulation

- Analog variables in the game
  - Health
  - Completion
  - …
- Parameters of sound synthesis
  - Volume
  - Low-pass filter cut-off
  - …

Modulation

- Mapping
  game state ⇔ sound synthesis

Music Production

- Music is typically created aside from the entire project
- Game is defined
- Timing is known

Music Production

- Input
  - – Artistic intent
  - – Gameplay description
    - List of supported locations
    - List of supported situations
  - – Available technology
    - Platform, available storage, …
  - – Legal considerations
    - Licensing of the game

Music Production

- Orchestration
  - What instruments are used
- Structure
  - Duration, planning
- Production
  - Composing
- Mixdown / Mastering
  - “Locking in” the music
  - Optimizing the volumes, polishing

Music Production

- Practical problems
  - Multiple stakeholders
  - Tons of files (as with any content development)

  - Hiring composers
  - Finding instrumentalists

Wwise

- Audio development platform
- Support of multiple audio tasks
  - Authoring, editing, setting up the adaptive behavior
  - Support of a number of platforms
- Example:
  - http://www.youtube.com/watch?v=Zvnt3tbL3OU&feature=youtu.be&t=10m28s
  - http://www.youtube.com/watch?v=STAdQwgDYHQ&feature=youtu.be&t=4m57s

8-bit Music

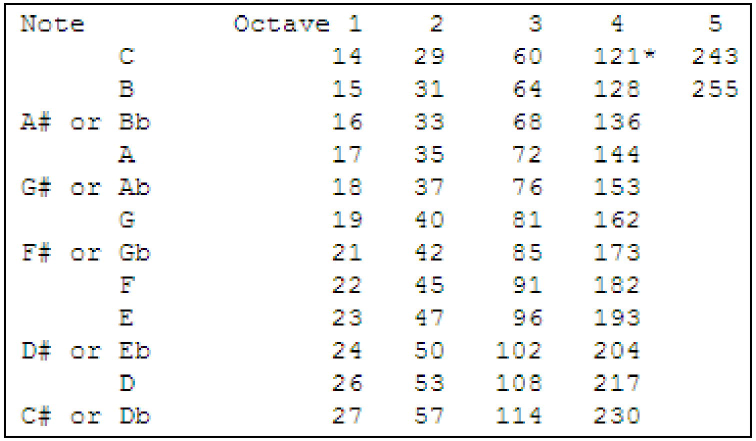
- Synthesizers with digital control
- Typically 8-bit values (or less)
  - Discrete volume levels (16 on ATARI)
  - Discrete tuning

- Sound Sources:
  - ATARI 800 XL (POKEY)
    - http://asma.atari.org/
  - Commodore 64 (SID)
  - Nintento Entertainment System (Famicom)

8-bit Music: ATARI 800

- 4 oscillators
  - 3 wave forms
- 16 discrete volume levels
- Discrete tuning

- Common tweaks:
  - Arpeggio
  - Chorus



Note	Octave 1	2	3	4	5
C	14	29	60	121*	243
B	15	31	64	128	255
A# or Bb	16	33	68	136	
A	17	35	72	144	
G# or Ab	18	37	76	153	
G	19	40	81	162	
F# or Gb	21	42	85	173	
F	22	45	91	182	
E	23	47	96	193	
D# or Eb	24	50	102	204	
D	26	53	108	217	
C# or Db	27	57	114	230	

8-bit music: RMT

Music tracker

<demo>

Module Music

- (a.k.a. tracker music)
- Late 1980s; Wave audio
- First format specification: MOD
  - Amiga computer platform
  - 4 channels, up to 15 instruments
- Main components:
  - Instruments (available sounds in the song)
  - Tracks (how these sounds should be sequenced)

Module Music: Instruments

- Instrument in a MOD file
  - Uncompressed audio wave
  - Defined reference playback frequency
  - Pitch control: Changes of the playback frequency
- Limited resources (memory, computation)
  - Lower sampling frequency of the samples

Module Music: Tracks

- Matrix-like music representation
  - Columns: Channels
  - Rows: Equidistant points in time
  - Cell:
    - Note On (pitch, instrument, command)
    - Note Off
    - Nothing

Volume
Pan
Tremollo
Portamento
Retrig
Tempo
…

Module Music: Notable Work

- Future Crew: Second Reality
  - 1993
  - Demoscene
  - http://www.youtube.com/watch?v=8G_aUxbbqWU

Module Music: Notable Work

- Jazz Jackrabbit
  - http://www.youtube.com/watch?v=b16upFloYak

Module Music: Demo

<demo>

SFX

Sound Effects

- Diegetic Sounds
  - – Related to events in the game

- One-shot Samples
  - – Have a collection of different instances of the same sound!
- Looping Sounds
  - – They should not sound loop-y

Looping Sounds

- <demo>

Organization of Sounds in Unity

- Sound sources
  - Triggers
- Environment
- Doppler effect