Collaborative web-based audiovisual systems

Mani Nilchiani

https://github.com/maniart/tadaex14

Fall 2014 - Tehran - Mohsen Gallery

Session One

Mani Nilchiani

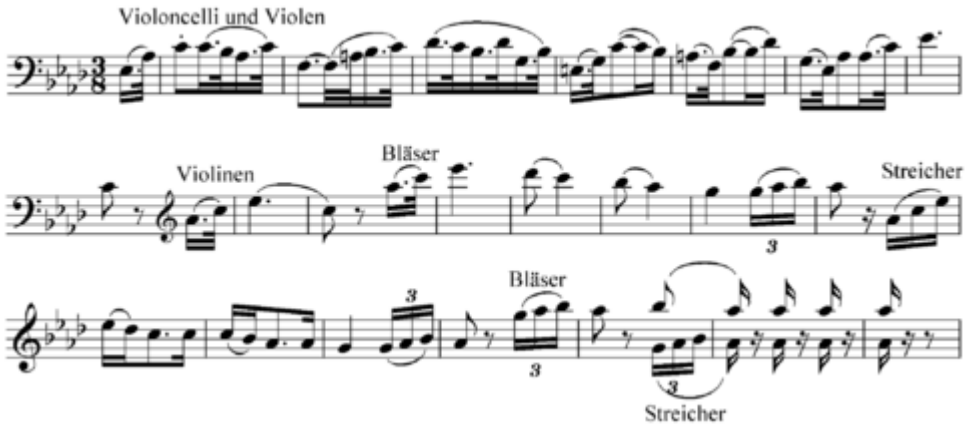
https://github.com/maniart/tadaex14

Fall 2014 - Tehran - Mohsen Gallery

Session One: Agenda

- Sensorial translation
- Precedence [analog]
- Precedence [digital]
- Digital tools
- Why Web [patatap]
- Some programming concepts
- WebAudio API
- Two.JS
- WebAudio API Nodes
- Input / Output
- Camera as input

Sensorial Translation



Precedence [analog]

- Fantasia [1945]

Precedence [analog]

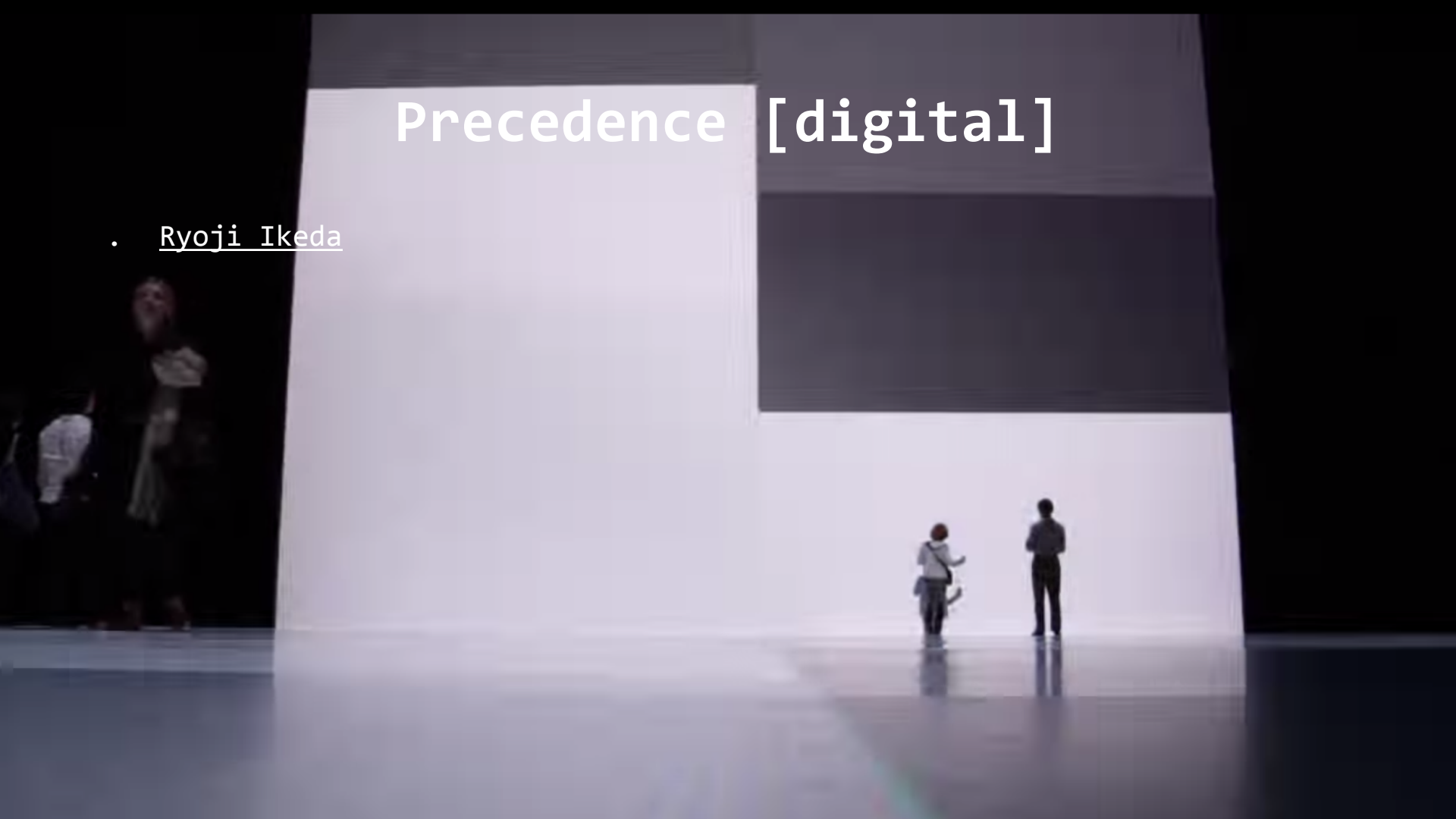
- John Whitney

Precedence [digital]

- Sam Aaron

Precedence [digital]

- Ryoji Ikeda



```
(λ (beat pitches dur)
  (play s (random pitches) 80 (* 1.5 dur))
  (callback (*metro* (+ beat (* .5 dur))) 'pads (+ beat dur) pitches dur)))

ads (*metro* 'get-beat 4) 8)
```

Precedence [digital]

- Andrew Sorensen

```
:-  xtm          All (6,28)    (Extempore:7098(TCP) +2 ElDoc yas Fill) 2:41PM 0.73
aded 14 files into bank#: 3
aded 71 files into bank#: 4
und _recdata_ >>> double*
mpiled record >>> [double,double]*
mpiled save-recording >>> [i64,i8*,i64]*
mpiled fade >>> [double,double,double]*
mpiled dsp >>> [double,double,double,double,double*]*
w Client Connection
tempting to return a string from non-symbol obj ()
ace: callback-adapter

:-  sh1          Bot L849    (Shell:run +1) 2:41PM 0.73
```

Digital Tools

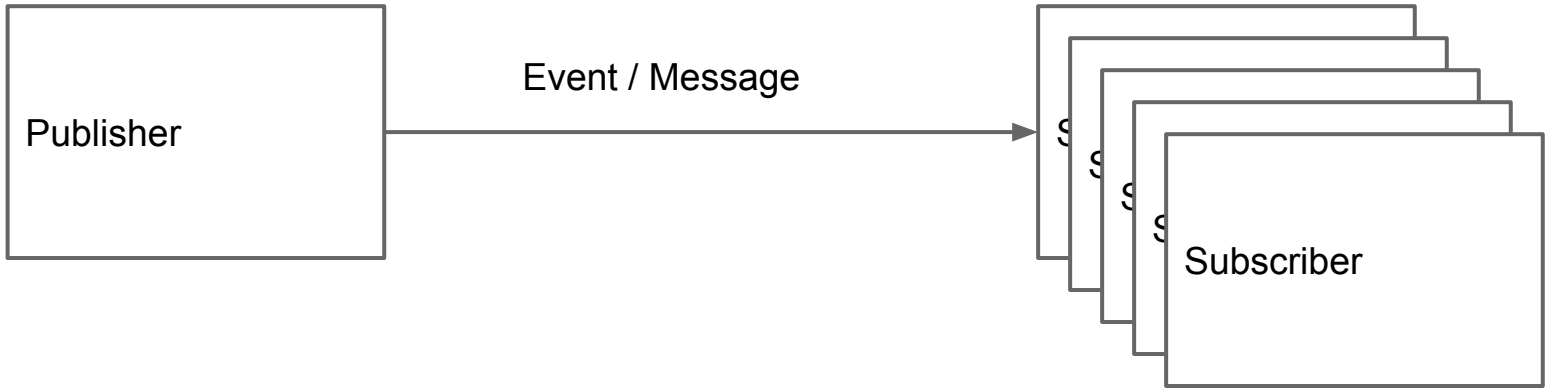
- CSound
- Max/MSP
- Chuck!
- Processing
- OpenFrameworks
- etc.

Why Web?

- It’s portable
- It’s real-time [well, potentially]
- It’s collaborative in nature
- Example: patatap
- Example: plink

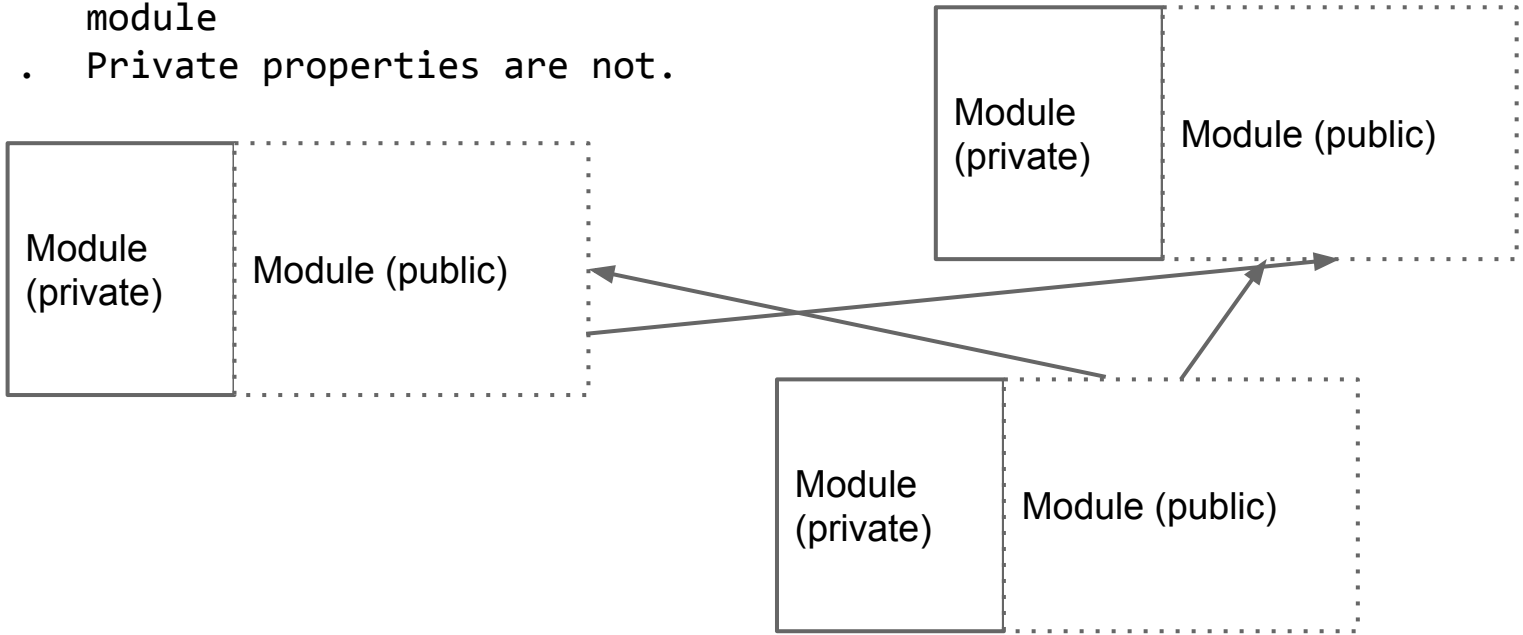
Some programming concepts

. Communication between different components of program



Some programming concepts

- Modular design - programming
- Public properties are available from outside each module
- Private properties are not.

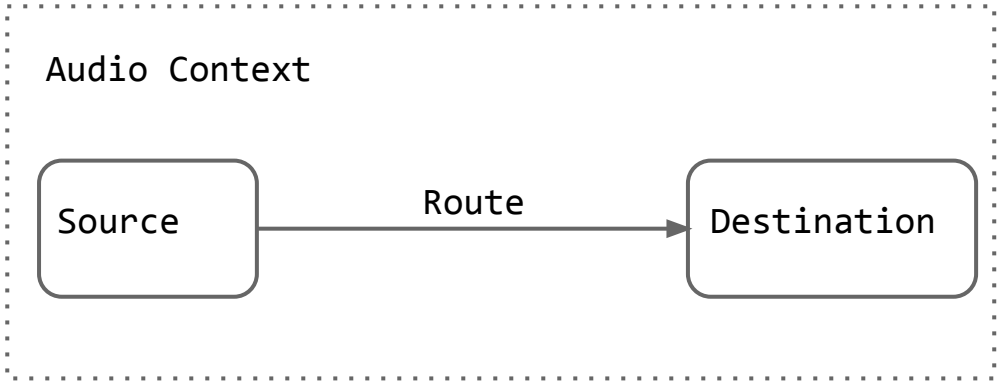


Audio : Web Audio API

- Before, it used to be Quick Time or Flash
- Web Audio API is used for Audio loading, playing and synthesis in the web
- Made for gaming
- W3C standard
- Still not compatible with all browsers, but work in progress.
- Learn more
- Learn even more
- even more

Audio : WebAudio API: Playing basic sound

- Create a audio context : new webkitAudioContext()
- Load the audio buffer : request.responseType = 'arraybuffer';
- Connect a buffer source to the context destination : src.connect(ctx.destination);
- Trigger a play : source.noteOn(0);
- **CODE EXAMPLE ONE**



Visual : Two.js

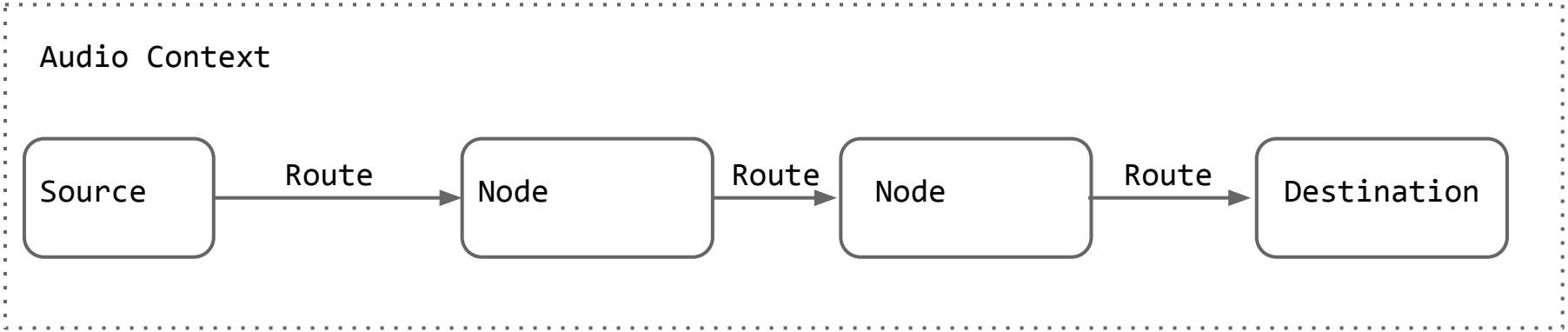
- Easy to use
- Rendered-agnostic (WebGL, SVG, Canvas)
- Abstraction for drawing and animation
- Learn more
- Lean even more
- **CODE EXAMPLE TWO**

Shape of a note

- How do we decide to translate sound into shape?
- What do you think a C, or an A looks like?
- **CODE EXAMPLE THREE**

WebAudio API: nodes

- Node Types:
- Positional: AudioPannerNode and AudioListener JS
- signal processing: JavascriptAudioNode
- Volume: AudioGainNode
- Environment: ConvolverNode
- Common Filters: BiquadFilterNode
- Learn More
- **CODE EXAMPLE FOUR**



Input / Output

- That’s basically what all computers are about, no?
- Question of interaction design.
- Keyboards are … OK. but boring.
- Mesa Di Voce: Realtime Interactive Sound > Visual / performance
- Camera as input
- **CODE EXAMPLE FIVE**

More soon!

- Next session we will cover:
- Waves and their shapes
- Oscillation
- OpenCV
- Realtime networks
- You should checkout this guy's talk at JSConf 2014

- write to me: mani.art @ gmail . com
- http://maninilchiani.com