SONUS
ROOM

The Concept

★ Interactive online music collaboration room

★ Communicate with attributed custom synthesizers

★ No UI - Minimal visualization of sound source with color codes, locate other players

★ Goal is to promote alternative forms of collaboration

★ Discovering user behavior
- ○ Communicating with sounds, back & forth
- ○ Playing music together, composing, jam
- ○ Making as much noise as possible

★ Web visuals and synthesizers
  ○ P5.JS

★ Managing connections
  ○ PHP
  ○ MySQL
  ○ AJAX

★ Visual inspiration
  ○ Bomomo.com
  ○ Abstract online visuals editor
  ○ Will be generated by audio

★ Controls inspiration
   ○ Websid (http://www.igorski.nl/experiment/websid)
   ○ Online web audio synthesizer
   ○ Would be simplified

★ Sounds inspiration
  ○ Inudge (http://tonematrix.audiotool.com/)
  ○ Simplified web audio synthesizer and sequencer
  ○ Similar digital, simple sounds

Technical **Research**

★ **p5.sound**
- ○ Sound oscillators (Sine, Tri, Saw, Sqr)
- ○ Basic sound manipulation (Delay, Reverb, Distortion, Amplitude)
- ○ Web audio functionality

https://p5js.org/reference/#/libraries/p5.sound

★ **p5.gibber**
- ○ Advanced sound effects and manipulation
- ○ Existing sound instruments (Drums, Pluck instrument)

http://charlie-roberts.com/gibber/gibber-lib-reference/

★ Websocket
- ○ Direct connection between multiple users, through server
- ○ Not very compatible, requires custom server configuration

https://www.html5rocks.com/en/tutorials/websockets/basics/

★ Long polling
- ○ Sending requests using Ajax, while keeping connection open
- ○ Allows for instant data transfer

https://github.com/panique/php-long-polling

★ **p5.js Particle System**
  ○ Generate particles with simulated physics in p5
  ○ Will be generated with data from generated synths

https://p5js.org/examples/simulate-particle-system.html

Project Scope

BEGIN WITH BASICS

1. Create database, setup server-side data (PHP, MYSQL)
2. Setup active connection to server (AJAX)
3. Create one synth in P5 and link to server
4. Create basic visualization

BUILD UPON

5. Add more sound synths
6. Add server-side detection of user data
7. Work on looks and visualization

SENDER
type int (0… 1)
posX int (0… 100)
posY int (0… 100)
time timestamp
valR int (0… 255)
valG int (0… 255)
valB int (0… 255)
key int (65… 90)

initialize()
onKeyPress()
sendData()

0..*   1

SERVER

1   1

RECEIVER
type
posX
posY
time
valR
valG
valB
key

onReceivedData()
generate()

1   0..*

COLOR BURST
posX
posY
valR
valG
valB

showBurst()

1   0..*

INSTRUMENT
frequency int (50… 3200)
amplitude float (0… 1)
oscillator int (0… 4)
delay float (0… 1)
reverb float (0… 1)

playNote()

PIANO