

Experimental Chatroom Script

Lark and Xin

(we plan to build a website to host workshop materials)

Recommended Schedule

Mon	Tues	Wed	Thur	Friday	Sat
July 20	July 21	July 22	July 23	July 24	July 25
Opening Meeting	Reading 1 async discussion	Reading 2 async discussion	Reading 3 async discussion	Closing Meeting	H+D presentation
		Open Lab	Open Lab		
Customizing the UI tutorial	Customizing the UI tutorial	Option 2 or 1 tutorial	Option 2 or 1 tutorial		

Synchronous Meetings Agenda

- **July 20, Opening Meeting**
 - Permission to record
 - Introduction: preferred name, pronouns, “smtg playful or serious”
 - Code of Conduct
 - Look at H&D CoC
 - [Design Justice Network Principles](#)
 - Items people want to add to CoC?
 - Communication channels
 - Website
 - Async chatroom
 - Open lab hours
 - Opt-in contact list for arranging work groups

- A very brief history of experimental chatrooms & databases
- Experimental chatroom demos > distinguish between tutorial tracks: p2p and/or archival chatrooms
- Overview of workshop readings and tutorials
- What are you excited to learn in this workshop? Do you have any skills/experience/perspectives you'd like to offer?
 - Discuss via chat in workshop chatroom =)
- Closer
- **July 24th, Closing Meeting**
 - Permission to record
 - Share projects 🏠
 - Add projects to a document, exchange contact info
 - Closer

Readings

- Reading 1: Consensual Tech
 - To read: [Building Consensual Tech](#), And Also Too + a reading on UI/UX (TBA)
 - Asynchronous chat discussion:
 - How do these principles relate to your own observations of online platforms and what are some other forms of consent violations that haven't been mentioned in the zine?
 - ie. virtual :hugs: and 🍆
 - Ideas for consensual language / interface / infrastructure for chat platforms
 - Supplemental readings:
 - [Rape in Cyberspace](#), 1993 Julian Dibbel
 - Historical example of people navigating community accountability
 - [Afterword: Blaming, Shaming, and Feminization of Social Media](#), Lisa Nakamura
- Reading 2: Peer-to-Peer web
 - To watch: [Tara Vancil video](#) about Peer-to-Peer web (or maybe [this one](#): Reclaiming the Web with peer-to-peer protocols)
 - To read: a nerdy blog post on WebRTC (TBA)
 - Asynchronous chat discussion (on a p2p chatroom):
 - What types of conversations should remain temporal?
 - Ideas for if you might use p2p for your chatroom, and how?
 - Supplemental reading:
 - [Decentralize What](#), Francis Tseng
- Reading 3: Archival and Building of Collective Memory

- To read:
 - Reading on Multilinear storytelling: critical awareness of whose history we are archiving (TBA)
 - [Zine: Making History](#) from QZAP
- Asynchronous chat discussion:
 - How are zines comparable to chat rooms?
 - How can chats function as oral archives?
 - How does your chatroom operate as an archive?
- Supplemental reading:
 - [Run your own social](#), Darius Kazemi

References

Experimental social medias (WIP)

- Scuttlebutt - <https://scuttlebutt.nz/>
- NYC Mesh Networks - <https://www.nycmesh.net/>
- Mastodon - <https://joinmastodon.org/>
- Diaspora - <https://diasporafoundation.org/>
- Fritter - <https://github.com/beakerbrowser/fritter/blob/master/README.md>
- guifi - <https://guifi.net/>

Glossary of terms (WIP)

- Surveillance capitalism (link to [Shoshana Zuboff on surveillance capitalism](#), VPRO Documentary)
- Residual data
- Predictive data
- Training models
- Peer to peer (p2p)


Tutorial outline

General resources

- javascript resources
 - [javascript.info](#) website
- [Getting started with p5.js](#)
- HTML/CSS
 - [W3 schools](#)






- MDN [CSS Tutorials](#)
- [HTML + CSS: Design and Build Websites](#) book
- [CSS tricks tutorials](#)
- DOM navigation with vanillajs
- Responsive v mobile-first design.
 - [Flexbox Froggie](#), great game to learn flexbox
 - [Twitter Bootstrap](#) - Lark's YouTube tutorials on Bootstrap

CUSTOMIZING SIMPLE CHATROOM UI

: denotes beginner-friendly track (mostly HTML/CSS, light on JavaScript)

: covered by xin

: covered by lark

- 1. Overview of Bare Bones Template 
 - 1.1 Remixing the Glitch Starter 
 - Intro to Glitch interface
 - 1.2 Tour of HTML/CSS 
 - Brief, beginners don't worry! Section 2 customizing the UI will break all this down in detail
 - 1.3 Moving into javascript: Intro to p5 environment
 - For people familiar with javascript: beginning-intermediate
 - Show p5 reference > mention p5 editor
 - setup and draw functions, you can write custom functions but they need to be called within setup or draw
 - 1.4 Variables and selecting HTML Elements
 - Declare variables
 - Define DOM
 - Selecting HTML elements (document.querySelector + HTML ids)
 - 1.5 Event listeners
 - addEventListener
 - 1.6 Callback function
 - a type of custom function
 - Also looks like function(){} or Arrow (modern)
 - 1.7 Dot operator (in looking at messageInput.value - property v. method of object)
 - >> give them link to tutorial reference
 - Template literal / string (backticks)
 - Auto scroll
- 2. Changing the UI  
 - HTML for beginners

- Duckett Book, W3 schools
 - CSS styling
 - What is a tag / selector
 - Custom classes
 - Change things in classes we built
 - Some cool pointers:
 - Custom fonts
 - Custom borders
 - Box shadows
 - Background images
 - CSS animations
 - Point them to general resource on CSS
 - Position of elements
 - Splitting up message div into sent and received message divs
 - Adding other divs
 - Repositioning using absolute position
- ~~3.1 Baby's First p5.js Drawing App - Experimental Chatroom Tutorial Series~~
 - Drawing 🍄
 - p5 references
 - createCanvas()
 - background()
 - 2D primitives
 - fill(), [HTML color picker](#)
 - stroke(), noStroke()
 - mouseX, mouseY
- ~~3.2 Nesting p5 Canvas Inside a Div - p5.js Drawing App - Experimental Chatroom Tutorial Series~~
- 3.3 Creating Multiple Drawing Layers using createGraphics() - p5.js Drawing App - Experimental Chatroom Tutorial Series
 - createGraphics()
- 3.4 Storing Drawing Coordinates - p5.js Drawing App - Experimental Chatroom Tutorial Series
 - 2D array (light learning)
 - To be continued!
 -
- 4.1 Trigger function in p5 when send button pressed 🤖
 - Background changes color
 - Images > appear when send button pressed
 - Sounds play when button pressed
 - Have multiple send buttons for different images / sounds
 - Provide resources to learn additional visualization tips such as animation

Option 1: STARTING FROM SCRATCH

Learning to build the infrastructure for sending / receiving chat messages between users

5A Build the p2p template from scratch (with peer.js, p5.js) 🍄

- An overview of WebRTC
- Getting started with [peer.js' reference](#)
- Set up a basic peer.js connection in the p5 editor
- Integrating peer.js with the *Customizing UI template*

5B Build archival template (with nedb, feathersjs, p5.js) 🤖

- Building from scratch
 - Follow along with [feathersjs tutorial](#)
 - Restructure for p5 environment
 - Change syntax for accessibility
 - Change all const to let
 - No arrow functions
 - Integrate with *Customizing UI template*

Option 2: BUILDING FROM TEMPLATE

Learning chatroom design tricks through examples, starting from templates with integrated p2p or archive code

---> for p2p

- 6A Collaborative drawing 🍄
 - Connecting p5 drawing with the peer.js port to send to the other client
 - Cross-reference a video showing how to do the same with archival chat
 - createGraphics() to distinguish and erase different layers of drawings
 - Consentful drawing - giving and taking control of the other client's drawing (e.g. brush stroke width, color)
- Invisible Ink Writing 🍄
 - Redirecting texts to show up on p5 canvas
 - p5 text attributes
 - Conditionals - create fade in / fade out effect; create edge detection

---> for archival 🧠

- 6B Brief video outlining the base template!!!! 🧠💧
 - Include toggling of login and chat screens
- Archive as architecture 🧠💧
 - Use CSS to build the space for the archive
- Full screen p5 canvas animates over/behind chat
 - reacts to sending or clicking on messages
 - demo falling animations (?)
- Add sounds
- Delete messages from archive - add an 'x' at top right to delete a message
 - P5 canvas reacts to deleted message
- Building threads
 - Display threads as dropdowns
 - Display threads on new screen
 - as branching trees
- Also need demo to incorporate collab drawing into nedb