WebRTC

Connections Lab

Spring 2022

Project 3

Key Items

- Client-side + Server-side
- Ability to “connect” with other users (database or real-time)
- Client-side library (at least 1)
- Can continue on Project #1 or Project #2 or Something New
- Can collaborate with a partner

WebRTC

A **web** api that gives access to real-time communication (**RTC**) in a browser

Key Details

- 3 channels of communication - video, audio, & data
- A "low level” api
- Once a peer connection is established, clients do NOT need a server to be an intermediary (i.e. peer to peer)

MediaStream

Another web api that represents the actual media stream

Key Methods

`getUserMedia( )`

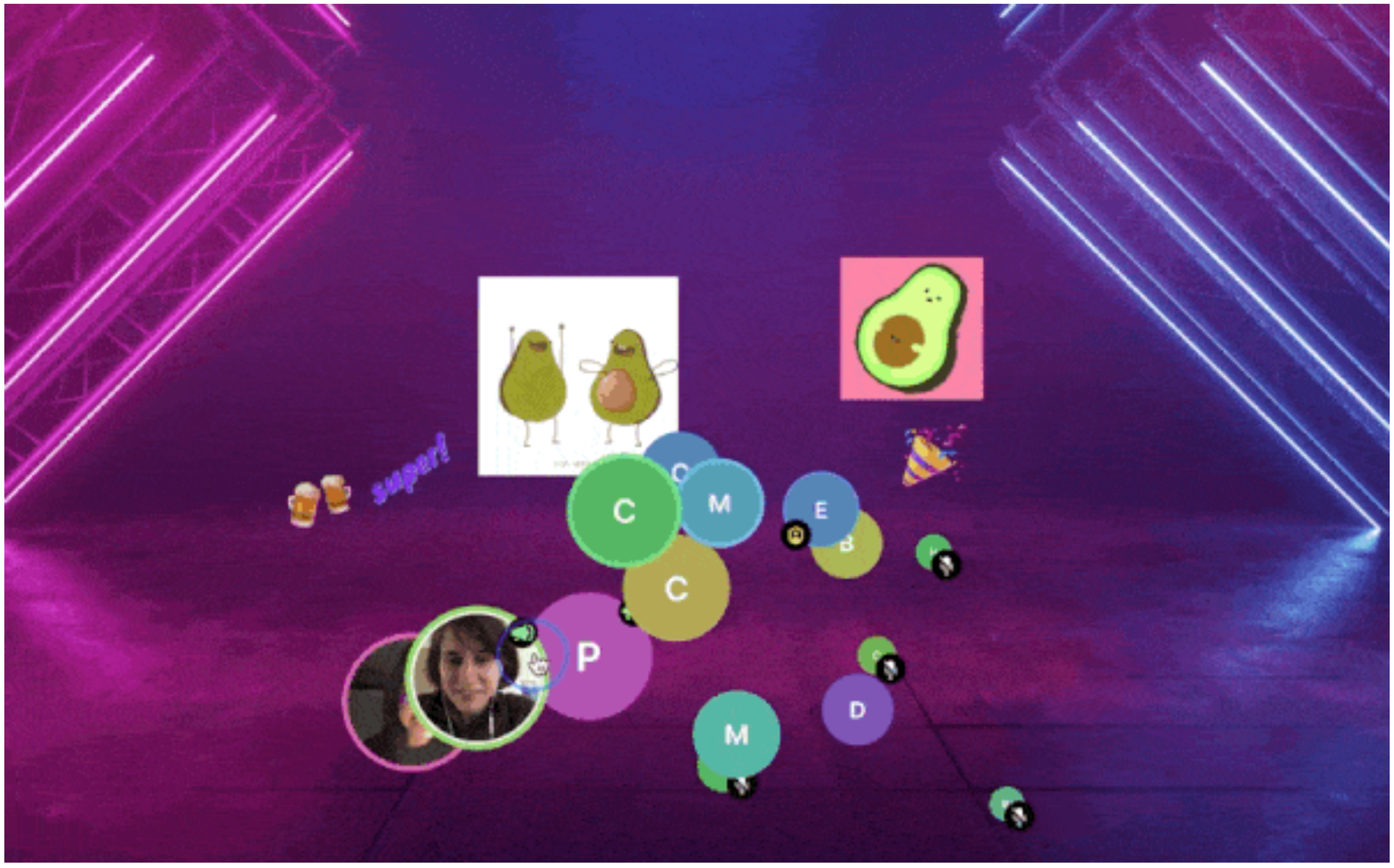
prompts for user permission to access webcam or mic

```
navigator.mediaDevices.getUserMedia( )
```

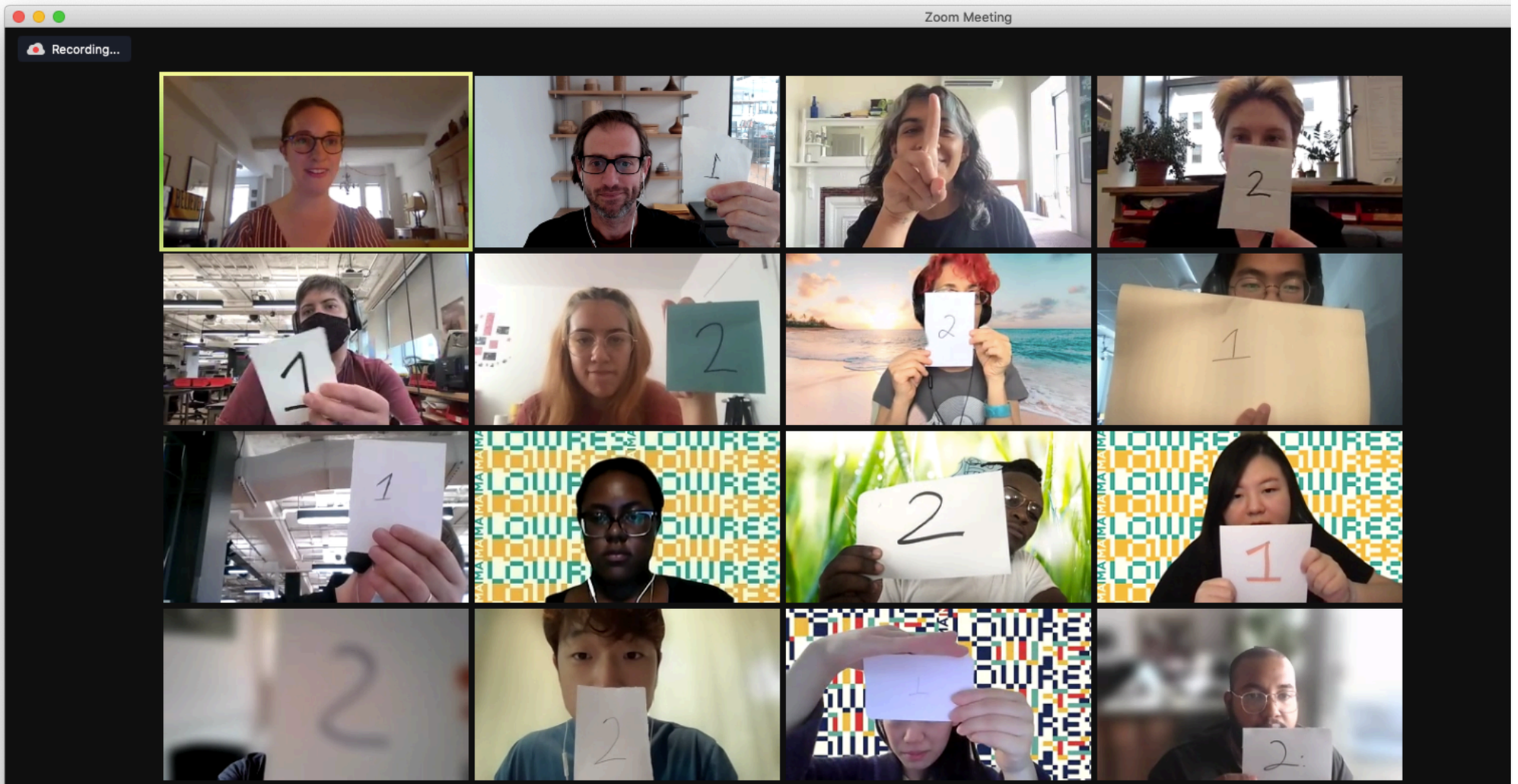
`captureStream( )`

real-time capture of the canvas (p5)

Examples



SpatialChat



zoom


press ESC to exit
MUSIC:
CONTROLS:
ESCAPE.....exit
WASD.....move
SPACEBAR.....jump
MOUSE.....look
C.....webcam
M.....mic
1.....paint
learn more about YORB here: yorb.online
CREDITS
Aidan Nelson (code)
Nicole Aptekar (model)
The YORB Team

YORB

digital stage

More WebRTC examples

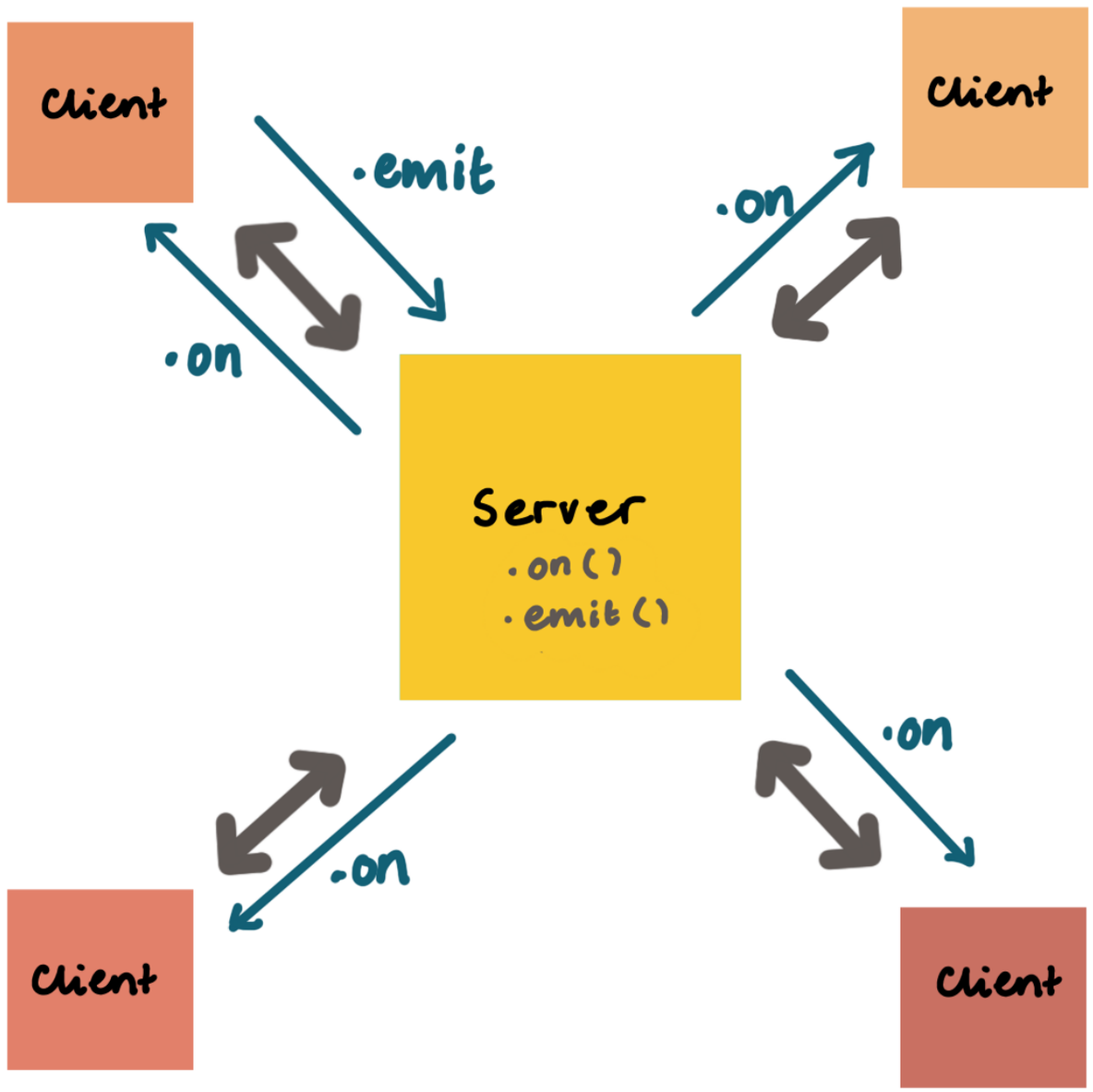
websockets vs webrtc

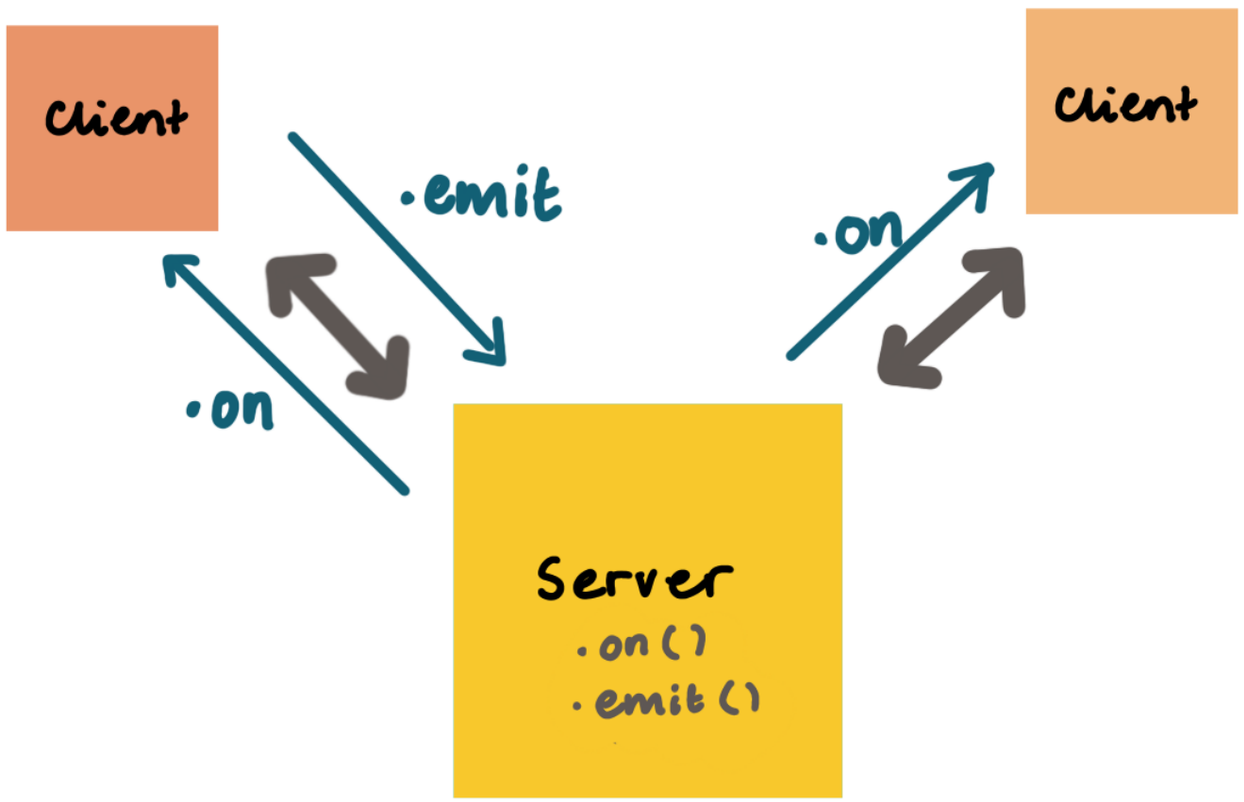
websockets vs webrtc

ideal for two-way “text” exchanges
data passes through centralized server
example - social feed

ideal for streaming audio and video
data sent from peer to peer
example - multi-media video chat

Websockets	**WebRTC**
Communication protocol	Communication protocol
Client-server-client	Peer-to-peer
Triggers	Real-time
Data	Video, audio, data


Client
.emit
.on
.on
Client
Server
.on()
.emit()
.on
.on
Client
Client



server uses socket.io
to connect clients

Server
.on()
.emit()

server is no longer part of the communication

Libraries

PeerJS

SimplePeer

p5LiveMedia

Questions?