Livecoding TouchDesigner in Javascript

Functional programming and caffeinated COMPs

Links / Downloads

Workshop files: https://github.com/ulyssesp/livecode-td-workshop

VSCode: https://code.visualstudio.com/

vscode-ldjs:
https://marketplace.visualstudio.com/items?itemName=ulyssesp.vscode-ldjs

Who am I?

- Freelance programmer based in Brooklyn
- Visuals since 2014, TouchDesigner since 2016
- Started because visualists then were only using computers as output devices but they can handle inputs too
- Most DIY-type spots in BK, and bigger e.g. HoY and Knockdown Center
- Created visual systems in threejs, webgl, Cinder, and TouchDesigner
- Now LambdaDesignerJS today, which is a TouchDesigner wrapper implemented in Typescript

Definition of livecoding

““In general, live coding is where people write code using a programming language, in order to change a live process. Rather than writing code, compiling it and then running it, live coders write code while a computer reads and processes it.”

(Live Coding Research Network, 2014)

Livecoding in practice

- Show your work!
- Failure is OK
- Use of algorithms as a form of expression
- Tools - TidalCycles, Hydra, Supercollider, Sonic Pi, Arcadia
- Most tools are type-checked to prevent crashes

Livecode community

- A global network of local groups
- We work hard to encourage diversity and create spaces free from harassment
- Performances aren’t just about the coding - people dance!

Livecoding vs. Node-based programming

- Node-based programming is a subset of livecoding
- Livecoding enables structural changes as well as parameters
- No three-button mouse required!

LambdaDesigner

- The lambda (λ) is a functional programming icon. It also looks cool.
- LambdaDesigner models TouchDesigner nodes as objects with parameters and an array of node inputs
- The vscode integration automatically fills node types and has a function to look up parameters
- Two node types, .run(), and .out()
- Javascript functions can create entire node networks using parameters - caffeinated COMPs!
- A custom node can also be created using a function of inputs

LambdaDesigner Parameters

- Params are parsed to python, so LD has tools for parameter operations (c.multp, c.addp, c.seconds, etc)
- Channels can be stored in variables and used as float parameters

LambdaDesigner Technicalities

- Tree-like object representing network is built in js
- Object is flattened into array of nodes with unique names
- JSON.stringify + send to TouchDesigner
- Python script diffs new json from last received json and makes changes
- npm package

Coding time!

1. Make sure everything is installed
2. Basic nodes
3. Connections
4. Functions
5. Functional networks
6. JavaScript (the good parts)
7. Nero

Next steps

- Practice livecoding! Either using LambdaDesigner or a different framework of your choice
- Join the livecode community! There are livecode communities in a ton of cities worldwide. Connect with me afterwards and I'll point you to the right people.
- Go to an algorave! Most livecode communities host algoraves, which is like a rave but with algorithms.
- Read about livecodiing! https://toplap.org/ is a good place to start

Thanks for coming!

Github repo for workshop: https://github.com/ulyssesp/livecode-td-workshop

livecodenyc website: livecode.nyc

Instagram: @ulysses_le_sees

Github: @ulyssesp

Twitter: @ulyssespopple

Facebook: not gonna be difficult to find me here

Email: ulysses.popple@gmail.com