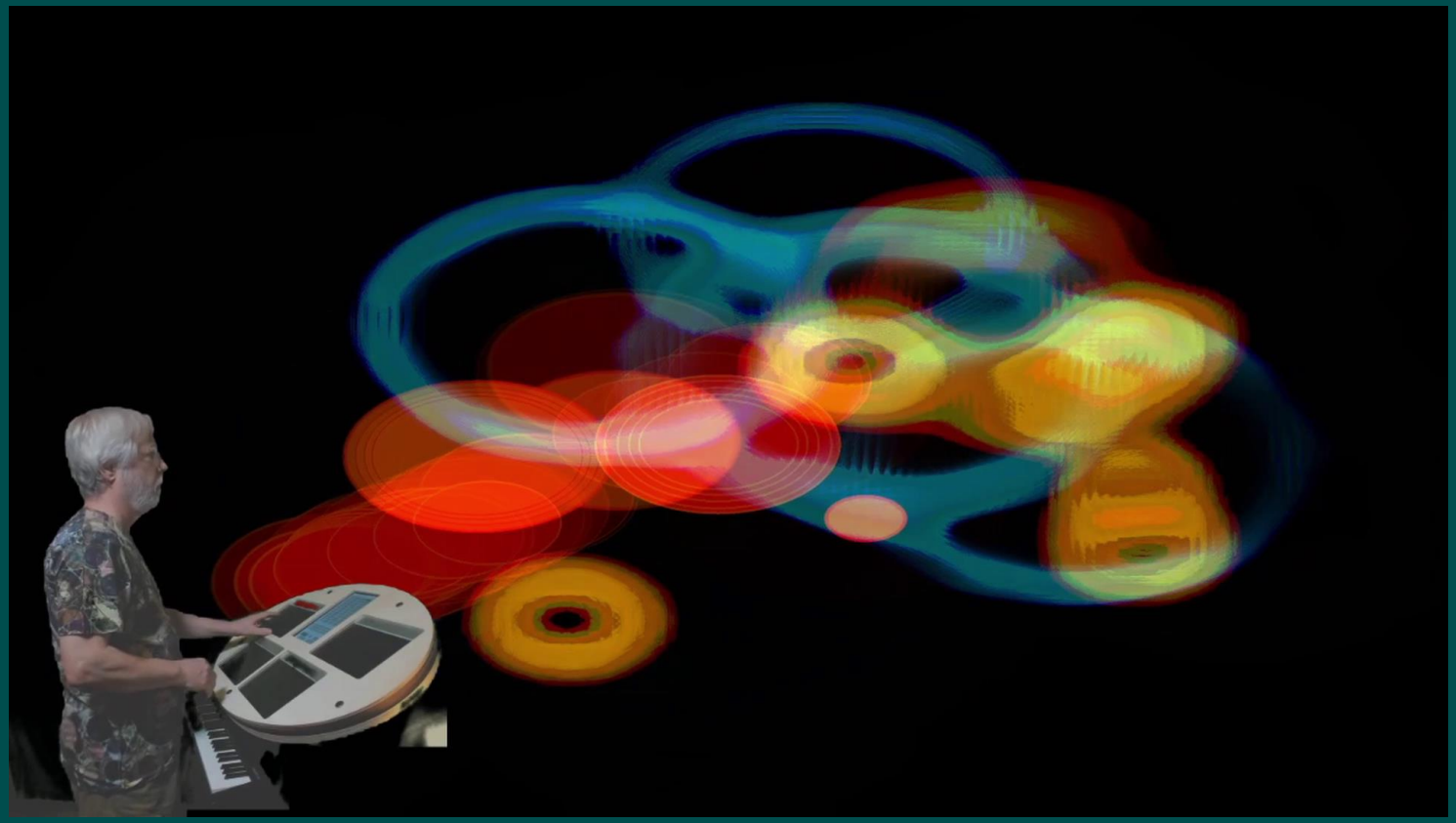
Event-Driven Techniques for Visuals



Tim Thompson https://timthompson.com me@timthompson.com

Outline

- Palette software
- Events
- Experiments

Palette Software

- Visual music instrument system
- Used in the Space Palette Pro
- Simple network-accessible APIs
  - Can be used from any language
  - GUI control and device input are separate from visual music engine, can be remote
- Experimental work-in-progress, evolving

Palette Implementation

- Realtime engine and API server is written in Go
- GUI is in Python
- Freeframe plugin is in C/C++
- Windows only
- Uses NATS - a pub/sub messaging system
- Source code: https://github.com/vizicist/palette

Event Types and Sources

- Cursor gestures with pressure (Sensel Morph)
- MIDI input
- API interfaces
  - Can simulate device input
  - OSC (to Engine and Freeframe plugin)
  - NATS
    - Locally - between GUI and Engine
    - Remotely - for remote control and collaboration

Event Responses and Manipulation

- Create sprites (OSC to Freeframe plugin)
- MIDI output
- Publish to cloud
- Looping, Recording
- Quantization
- Filtering events within an existing loop
- Realtime MIDI input can control scales

Experiment in the Cloud

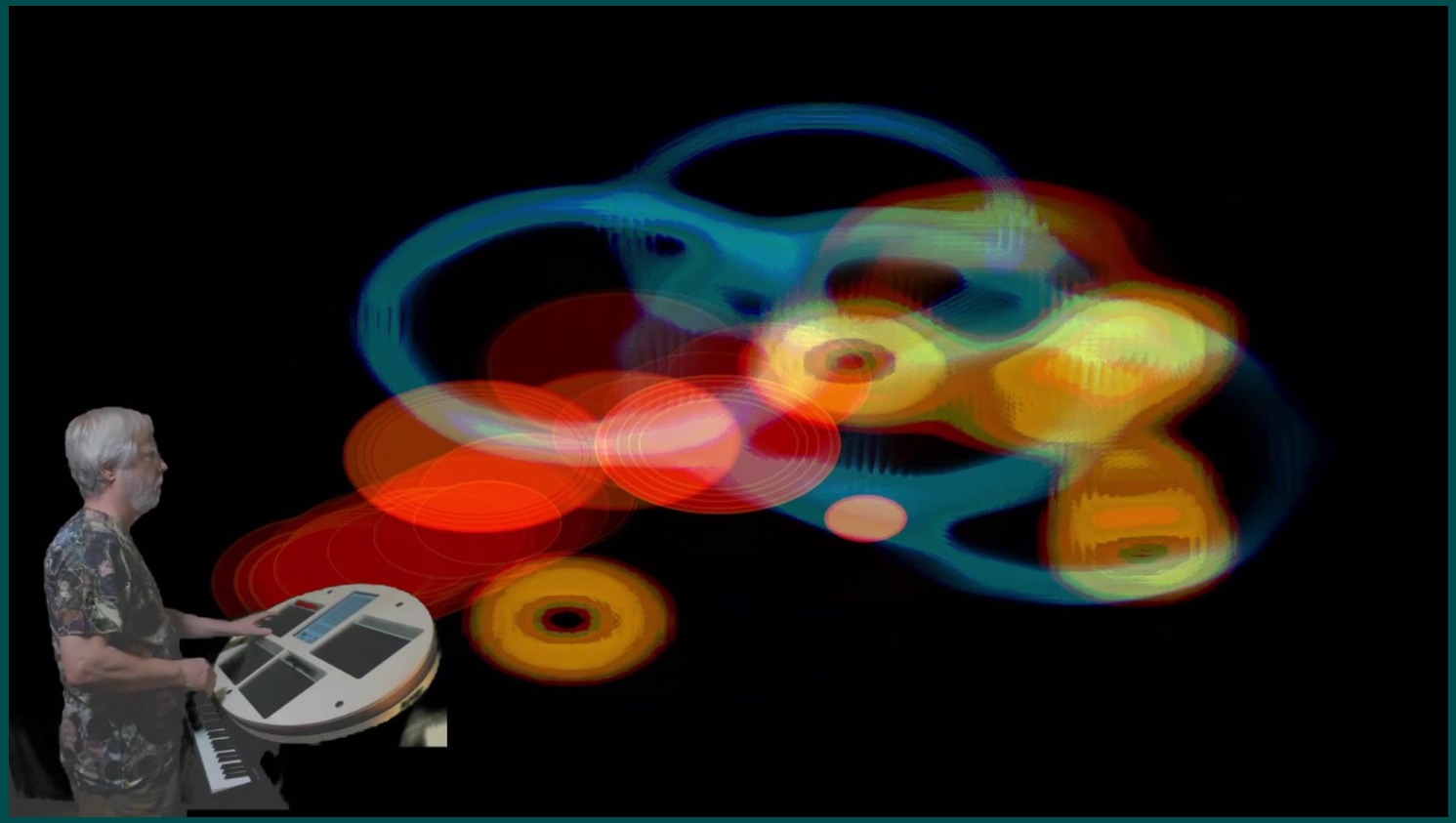
- Multi-user remote control of a visual music instrument
- Space Palette Pro (SPP) at my home, streamed live
- Four remote players, each one:
  - Has their own Sensel Morph
  - Controls one of the four pads in the Palette Pro
  - Has simultaneous and independent control of cursor movement, MIDI, and GUI parameter control

Experiments leveraging APIs

- Script-based
  - Python library to send events
- File-based
  - Read midifile (SMF) and generate sprites as realtime visualization.
- Simulation-based
  - Simulate balls bouncing around a maze and trigger things when balls hit the walls

Demo

Event-Driven Techniques for Visuals



Tim Thompson https://timthompson.com me@timthompson.com