



No Such Media

Tim Thompson

<http://timthompson.com>

me@timthompson.com

Aesthetic

- Maker
- Cross functional (software, hardware, music, visual)
- Experimenting with new things
- Fail early and often
- Deploy and document
 - Performances
 - Installations
 - Web site
- Refine, Rework, Remix

Motivation

- Fun
- Freedom
- Personal taste
- Learning
- Socialization
- Deadlines

Interests

- Algorithmic composition
- Selective Randomization
- Languages
- User Interfaces (both soft and hard)
- Networking

Initial Training

- Musician
- Software Engineer

... and then there was MIDI

KeyKit

- Textual programming language – procedural, interpreted, multi-tasking, graphics, object-oriented
- Specialized for MIDI algorithmic and realtime manipulation, first-class “phrase” data type, realtime scheduling
- Multi-window graphical user interface, pull-off menus and buttons, tools include multi-track editor, drum pattern editor
- Entire user interface and all tools written in the language and provided as source code in library, easily customized and extended

Development History

- 1.0 - BASIC-like
- 2.0 - realtime
- 3.0 - grammer, rewrite, fast enough to avoid built-ins
- 4.0 - graphics
- 5.0 - first multi-window attempts, multi-tasking, fifos, tools using tasks/fifos
- 6.0 - object-oriented, multi-window interface completely rewritten, pull-off menus/buttons
- 7.0 – support for multiple MIDI ports

MIDI Phrase is a first-class data type

- Time-ordered list of MIDI “notes” - can be system-exclusives, isolated note-on, isolated note-off, or full note with duration

- Constant value syntax

`ph = 'c e g'` # ph is a c major triad

`ph = 'dc2,e,f'` # ph is an arpeggio, channel 2

- Structure-like manipulation of attributes :

`ph.dur = 1b` # all note durations = 1 beat

`ph.pitch += 12` # transposed up an octave

Phrase Operations

- Serial concatenation

$$\text{ph} = \text{ph1} + \text{ph2}$$

- Parallel merging

$$\text{ph} = \text{ph1} \mid \text{ph2}$$

- Removing notes

$$\text{ph} = \text{ph1} - \text{ph2}$$

- Matching notes

$$\text{ph} = \text{ph1} \& \text{ph2}$$

- Nth note

$$\text{ph} = \text{ph1} \% n$$

Phrase Operations - the “select”

```
ph = ph1 { ?? .pitch > 60 }
```

```
ph = ph1 { ?? .dur > 1b }
```

```
ph = ph1 { isonbeat(??, 4b) }
```

```
ph = ph1 { ?? .number < 4 }
```

```
ph = ph1 { rand(3) == 0 }
```

```
ph = ph1 { isinscale(??, scale) }
```

```
ph = ph - ph { ?? .type == MIDIBYTES }
```

Other Language Features

- Variable arguments - ... , nargs(), argv(), varg()
- Fifos and locking
- Objects
- Graphics - primitive elements are:
 lines, rectangles, text, windows,
 phrase windows, menus
- Machine-dependent hook – mdep() – used to
 add/expose non-portable features
- TCP/IP hooks available for Windows and Linux,
 network interaction

Tasks and I/O

- All tasks are time-shared evenly, interleaved at the interpreted instruction level
- Scheduled MIDI output events are tasks as well, but performance can't be degraded by other tasks
- MIDI input is always being recorded, available in a global variable for easy and immediate processing
- MIDI, mouse, and console input events can be read from special fifos
- Reading a fifo (with no data waiting) blocks a task
- `lock()` and `unlock()` used for exclusion and synchronization


KeyKit - the GUI

- Completely implemented with Keykit code, even pull-off menus, dragging of windows, window-manager-like operations, etc.
- Each tool is independent, with consistent methods for resizing and inter-tool communication
- Consistent saving/restoring mechanism of individual tools is highly leveraged, used for:
 - Copying between like tools
 - Copy/paste of entire tools
 - Moving tools between “pages”
 - Manipulating of tools within tools
 - Broadcasting of a tool and its contents across a network


BASIC TOOLS

File View Edit Norm Aud Aud

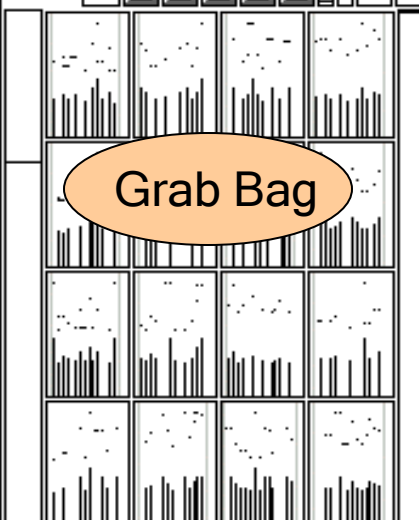
Merged Name: New #



Trk 1 S M X None



Gra M P P M 1 A W Lo



Roller Derby R Loo Mor Ch

Lo Hi Swee Sp

#0 #0 #0 #0


U D S L S L S L S L

On More

On

tim tra vol nti
1.0 0 90 1

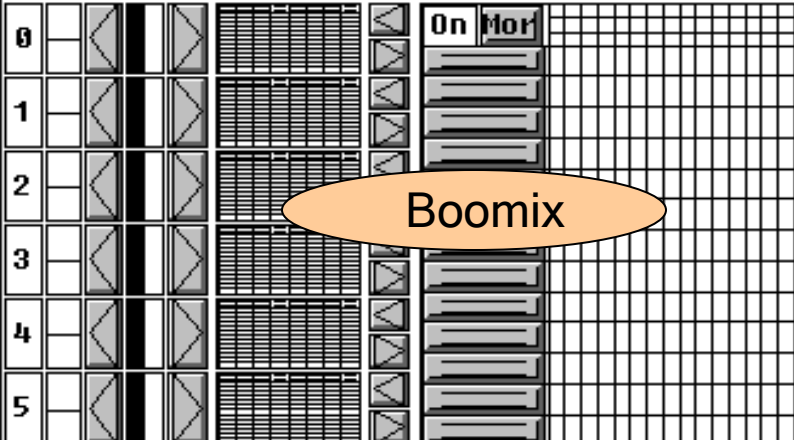
Techn 16 More Loop



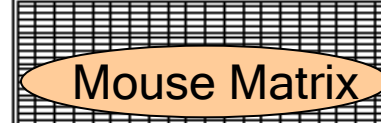
On More

Boomix! On Resync More

0 1 2 3 4 5



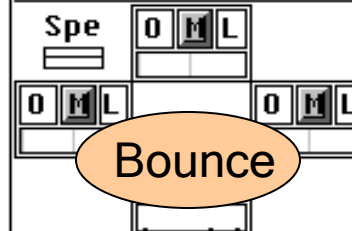
Mouse Matr 1 note



Bounce On

Spe 0 M L

0 M L 0 M L

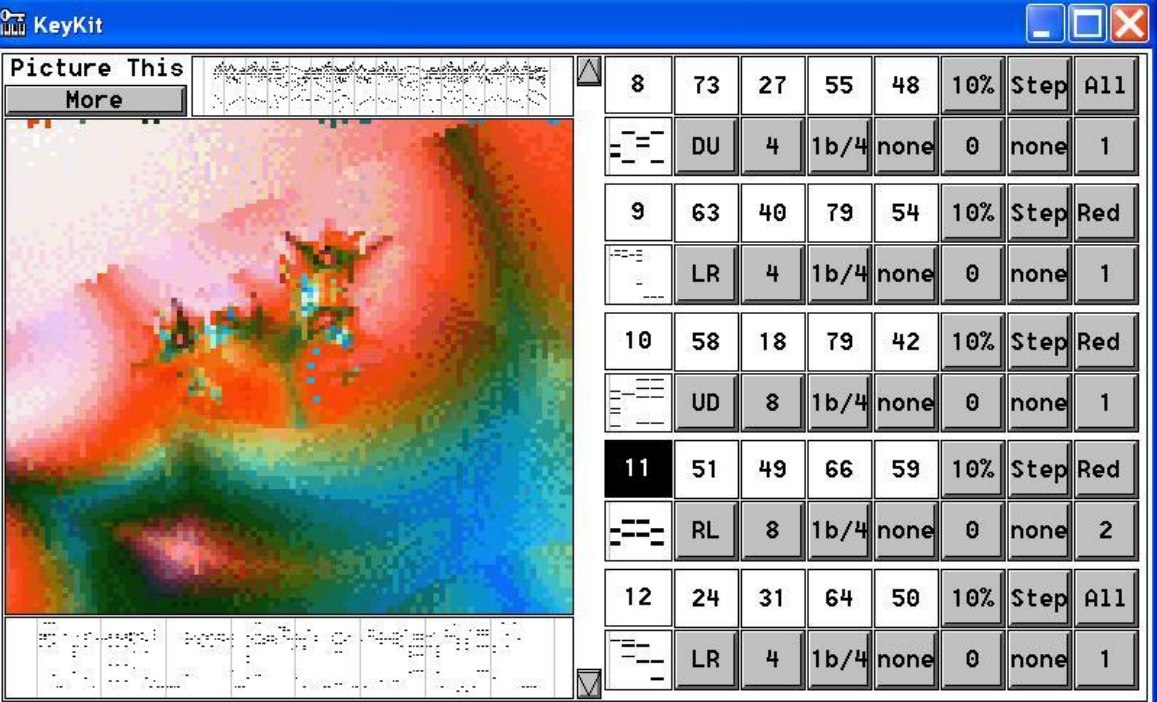


key> Turning Merge off...
Turning Merge back on...

TOOL VARIETY

Interactive Tool - Picture This

- Use RGB values of an image in various ways



The KeyKit software interface displays a color image of a bird in the center. To the right of the image is a table with 12 rows and 9 columns. The table contains numerical data, musical notation, and other information. Below the table is a command prompt window showing the following text:

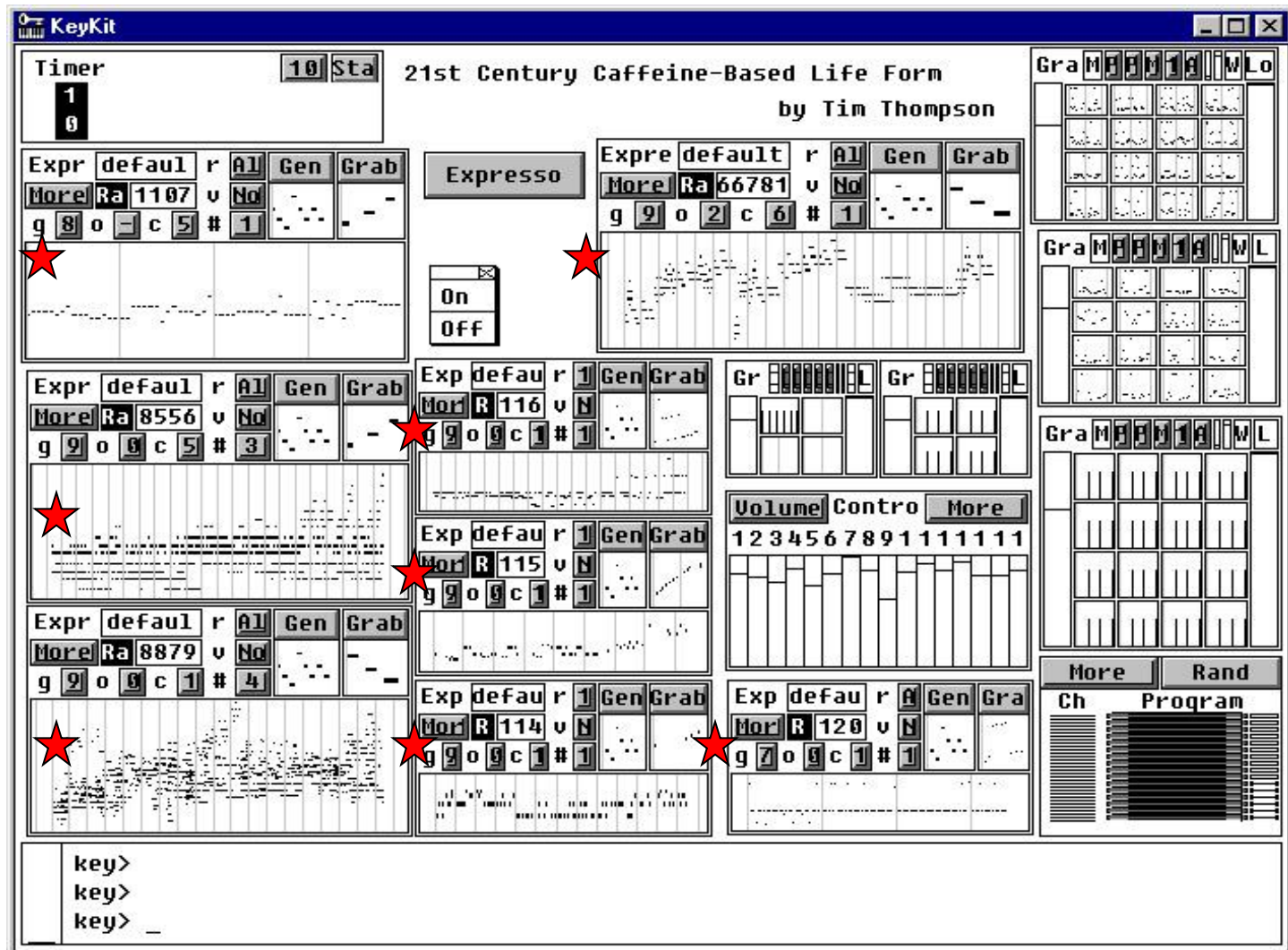
```
key>
key> Dumping page to C:\Documents and Settings\tjt\My Documents\Home\key
\key72f\music\snapshot1.kp...
Done.
```

8	73	27	55	48	10%	Step	All	
	DU	4	1b/4	none	0	none	1	
9	63	40	79	54	10%	Step	Red	
	LR	4	1b/4	none	0	none	1	
10	58	18	79	42	10%	Step	Red	
	UD	8	1b/4	none	0	none	1	
11	51	49	66	59	10%	Step	Red	
	RL	8	1b/4	none	0	none	2	
12	24	31	64	50	10%	Step	All	
	LR	4	1b/4	none	0	none	1	

Then I moved to Silicon Valley...

- Woodstockhausen
- Skronkathon
- Y2Kx LoopFest
- Dorkbot
- Illuminated Corridor
- Maker Faire
- BArCMuT (Bay Area Computer Music Technology) Meetup
- Visual Music Meetup
- LoveTechSF
- SHARE San Jose
- Zero1.org, ZER01, 01SJ Biennial, SubZero Street Faire

An Algorithmic music “instrument” used for “21st Century Caffeine-based Life Form” at Woodstockhausen 2000



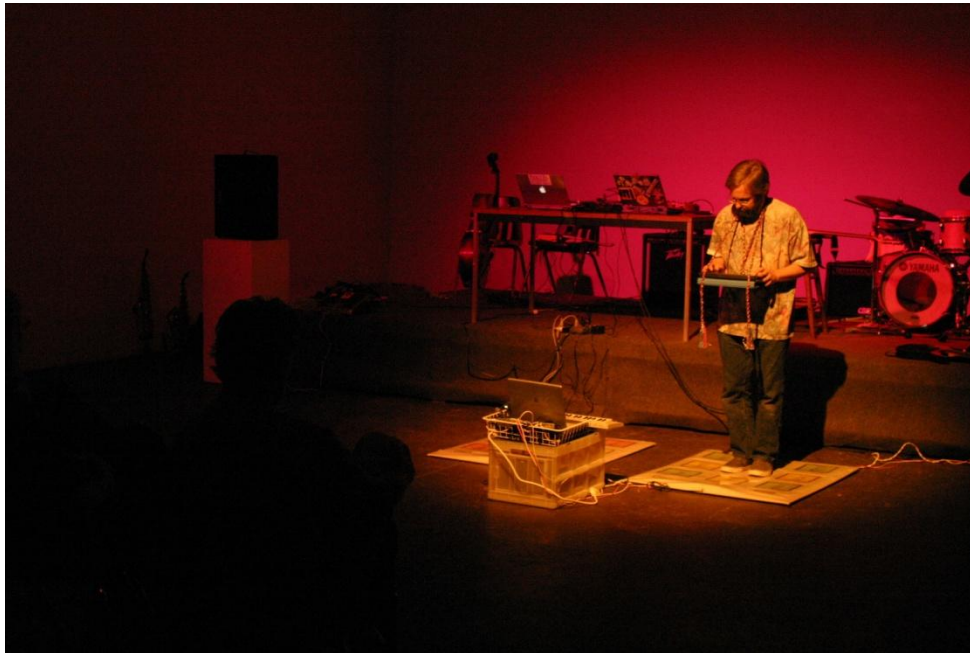
Oops, I made a typo

- Woodstockhausen 2001
- All-QWERTY Performance



Performances with dance pads

- Woodstockhausen performance – Happy Feet
 - Done entirely on 4 dance pads
 - No hands, No light, and EL-wire-outlined pants
- With Wireless QWERTY keyboard



The Mother of All Opportunities



Dance Pads at Burning Man 2002



[Movie](#)

Burning Man 2003



Dancing under the Stars of Lyra



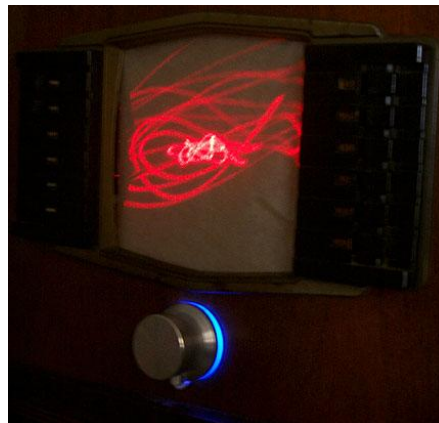
[Movie1](#)
[Movie2](#)

Dance Pads at Burning Man 2003

- Electronics and dance pads were very reliable
- Generator was not



Burning Man 2004 - Radio Free Quasar



Audio

Radio Free Quasar at Burning Man



Radio Free Quasar at Burning Man



Radio Free Quasar – the pieces

- 10 robust VST plugins chained serially
- Collection of WAV files
- Python program:
 - selects wave files
 - enables/disables/randomizes VST plugins
 - allows interactive control from keyboard
- Big knob on radio sends keypresses
- Automatic randomization if no user input

VST Manipulation in Python

- `nosuch.vstutil` module
 - `AudioSnippet`
 - `AudioLoop`
 - `PaAudioDevice`
 - `VstPlugin`
 - methods: `can_receive_midi`, `is_synth`, `is_vst2`, `name`, `num_inputs`, `num_outputs`, `num_params`, `num_programs`, `param_label`, `param_display`, `param_name`, `program_name`, `send_midi`, `get_enable`, `set_enable`, `set_param`, `set_program`, `set_input`

Basic use of `nosuch.vstutil.VstPlugin`

- Instantiate it:
`v = VstPlugin(dll="ringmod.dll")`
- Connect its input to the output of other plugins
`v.setinput(v2)`
- Randomize its parameters
`n = v.num_params()`
`for i in range(n):`
`v.set_param(i,random())`

Connecting a VST synth and effect

```
effect = VstPlugin(dll="BJ Ringmodulator.dll")
synth = VstPlugin(dll="StrataVar.dll")

# Connect output of synth to effect
effect.setInput(synth)

# Open audio and connect effect's output to it
a.open()
a.start()
a.attach(effect)

# Send random MIDI notes to synth
# while randomizing parameters of both VSTs
for i in range(100):
    time.sleep(2.0)
    pitch = int(random() * 128) % 128
    vstrandparam(synth)
    vstrandparam(effect)    # see next slide
    synth.send_midi(1, pitch, 8000, 80)
```


Other Algorithmic music (using KeyKit)

- Algorithms create lots of possibilities, but I choose which ones to keep
- Example using L-Systems (fractal expressions)
 - [23 Shots of Espresso](#)
- Example using the digits of PI (3.14...)
 - [Irrational](#)
 - [Irrational Too](#)

Graphics gets interesting

- Affordable equipment
 - Webcams
 - Video mixers
 - Security cameras
 - Video processors
- GPUs and CPUs fast enough for realtime

Visual Music experiments

- Python used for OpenGL support and convenience
- KeyKit is interface to input devices (MIDI, iGesture)
- Messages sent over TCP/IP to Python process
- MIDI sliders and buttons control graphics parameters
- MIDI from drummer triggers graphics
- Text typed interactively is used as graphics
- Words typed interactively can immediately search clip-art database whose images are then used as graphics
- Used in dud (improvised art ensemble) - www.dudland.com

iGesture pad

- www.fingerworks.com
- Inexpensive multi-touch pad
- Excellent responsiveness
- KeyKit interface to event stream
- Event data: x, y, proximity, eccentricity, orientation, contacts, device, finger, hand, xvelocity, yvelocity
- Multiple pads can be used simultaneously



A Custom Controller for Performing Graphics

- Fingerworks iGesture pads
 - Multitouch with area-detection
 - Extremely responsive



Dud - examples

- Chico : multitouch drawing, Python OpenGL graphics, webcam
 - [movie](#)
- Illuminated Corridor : Photoshop realtime drawing, one camera, dancer
 - [movie](#)
- Novato : two cameras
 - [movie](#)
- Musicians' Union : camera, Python OpenGL graphics
 - [movie1](#)
 - [movie2](#)

Dud - examples

- John Patrick's : drum-triggered graphics, camera
 - [movie](#)
- Musician's Union : dancer, FreeFrame, tracking
 - [movie](#)
- 21 Grand : dancers, indoor/outdoor cameras, four projectors, FreeFrame video looping
 - [movie](#)

Different Skies 2007



- 20 Electronic musicians and one visual performer gathered for a week-long workshop at Arcosanti in Arizona
- Music was composed for a concert at the end of the week
- Interactive graphics (no clips) was composed to match each piece of music
- The graphics performance rig:
 - [Interview](#)
- The two-hour concert:
 - [Movie](#)
- A time-compressed video of the two-hour concert:
 - [Two hours in two minutes](#)



Different Skies 2007



Other uses of Fingerworks iGesture pads

- Finger Fresco at Maker Faire 2007



[Movie](#)

Finger Painting with Planets

- Maker Faire 2008



[Movie](#)

Finger Painting with Planets

- Installation for people to play with
- Generates music and graphics simultaneously
- Controller with buttons, knobs, LCD, multitouch pad
- Fingers on pad trigger music or graphics
- Graphics motion is simulated gravitational attraction
- Collisions of planets trigger music
- Musical keyboard controls (only) selection of notes

Big Pieces

- KeyKit – input and realtime processing
- Plogue Bidule – VST host for sounds
- Salvation – Freeframe host for visuals
- Planets – Freeframe plugin
- Cairo – drawing on bitmap (from Python)
- Chipmunk – Physics simulation (from Python)
- OpenCV – raster manipulation (from C)
- OSC – communication between KeyKit and Planets

Multiple Languages

- Once avoided, now embraced
- Library availability
- Device I/O availability
- Robustness
- Ease of Development
- Familiarity

In Hindsight, the Good Things

- Python integration in low-level code works well
- Bitmap manipulation with multiple toolkits can work
- OSC is a simple and lightweight transport format
- Local sockets for inter-app API invocation good for:
 - Flexibility in choice of languages and applications
 - Portability
 - Firewalling
 - Robustness
 - Separating device I/O from graphics/audio output

Finger Painting with Planets

- Night Lights show at Climate Theater



Movie

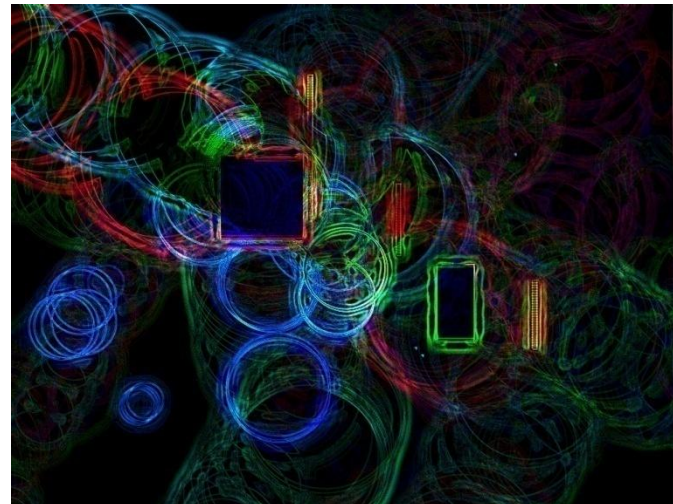
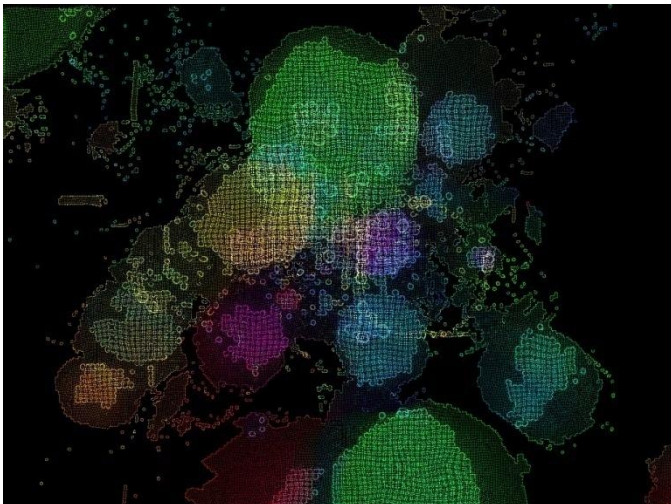
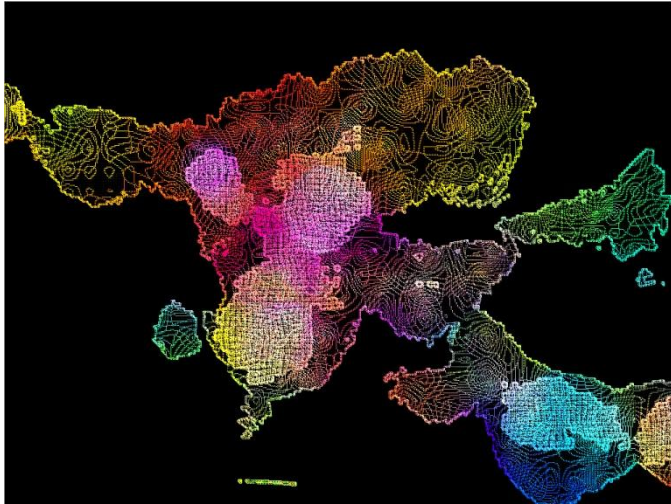
Finger Painting with Planets

- Yuri's Night

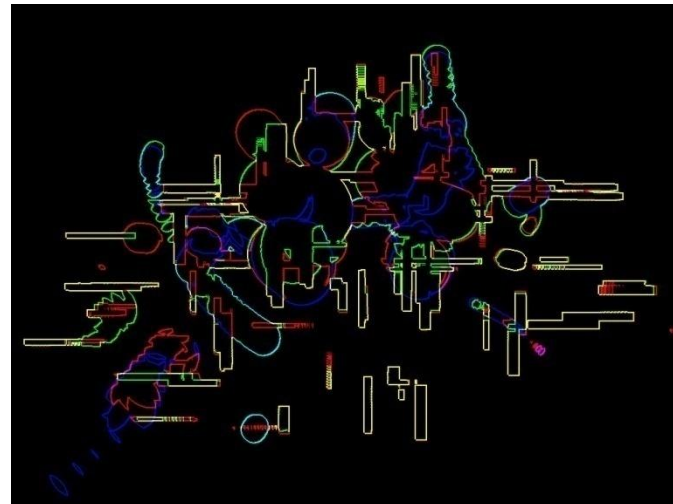
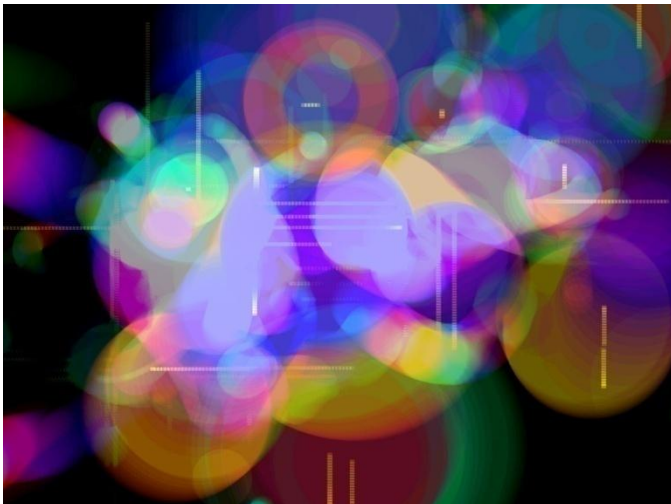
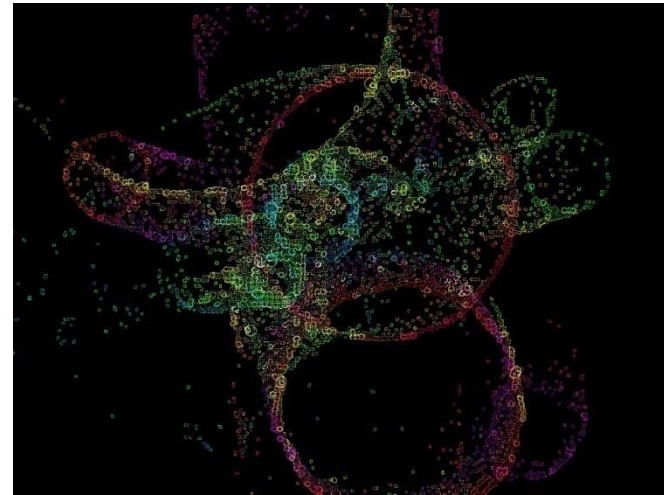
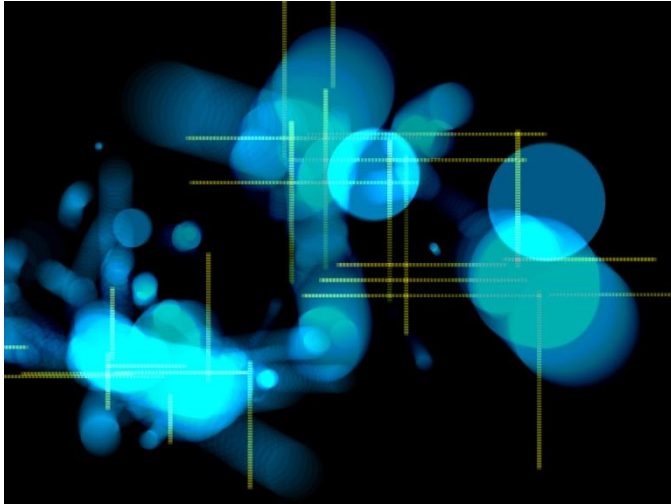


[Movie](#)

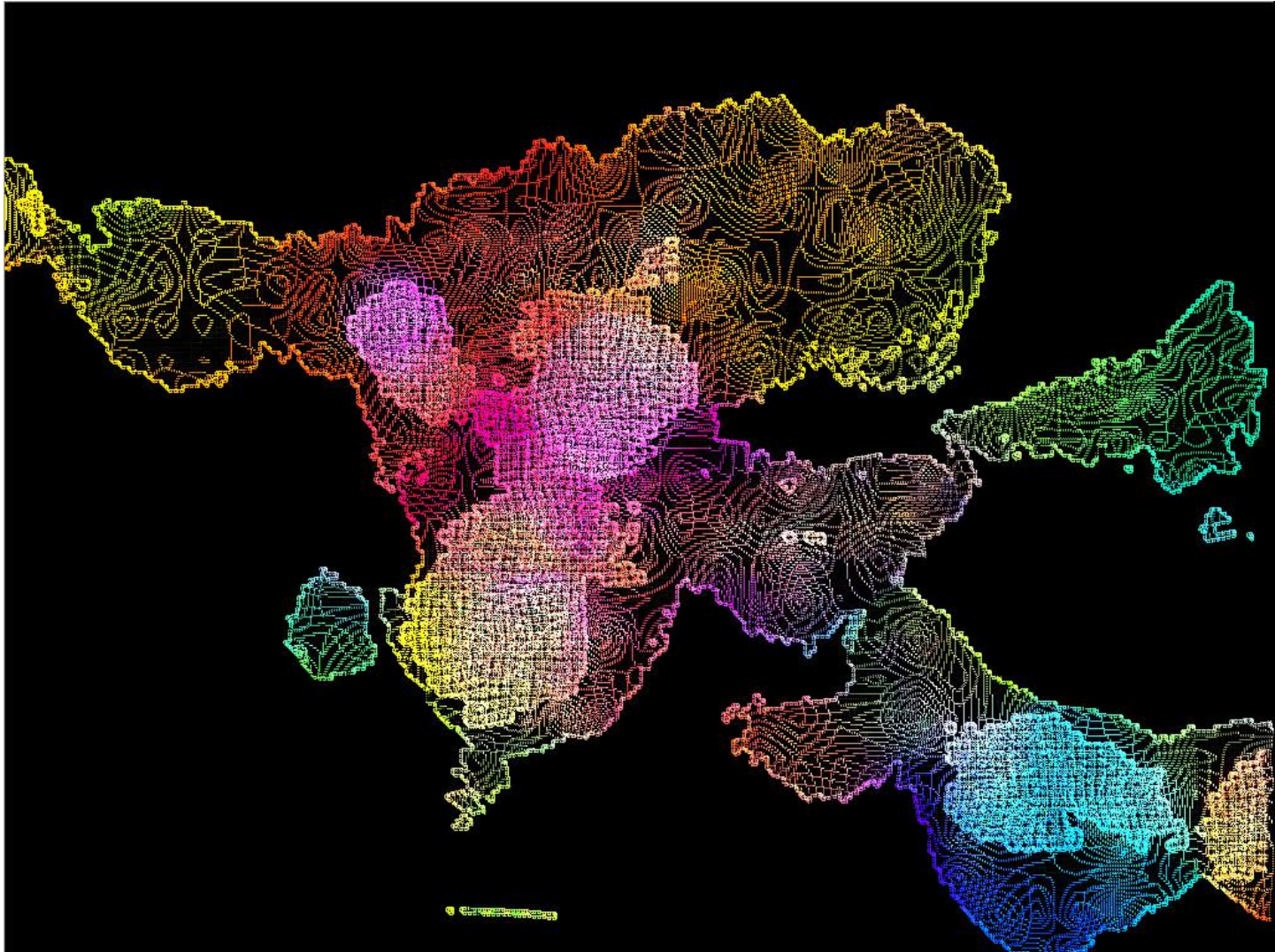
Examples of Visual Output



Examples of Visual Output



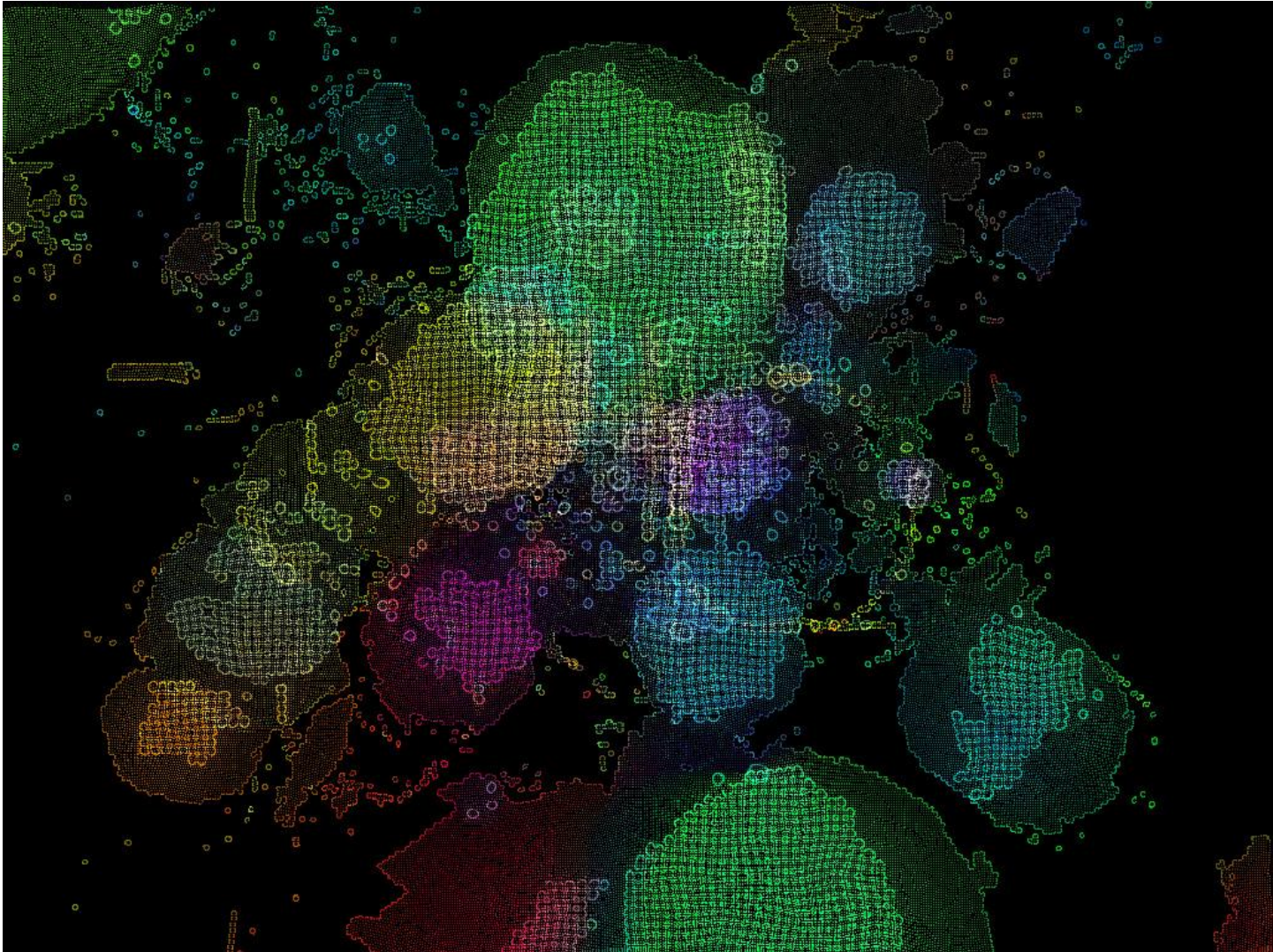
Examples of Visual Output



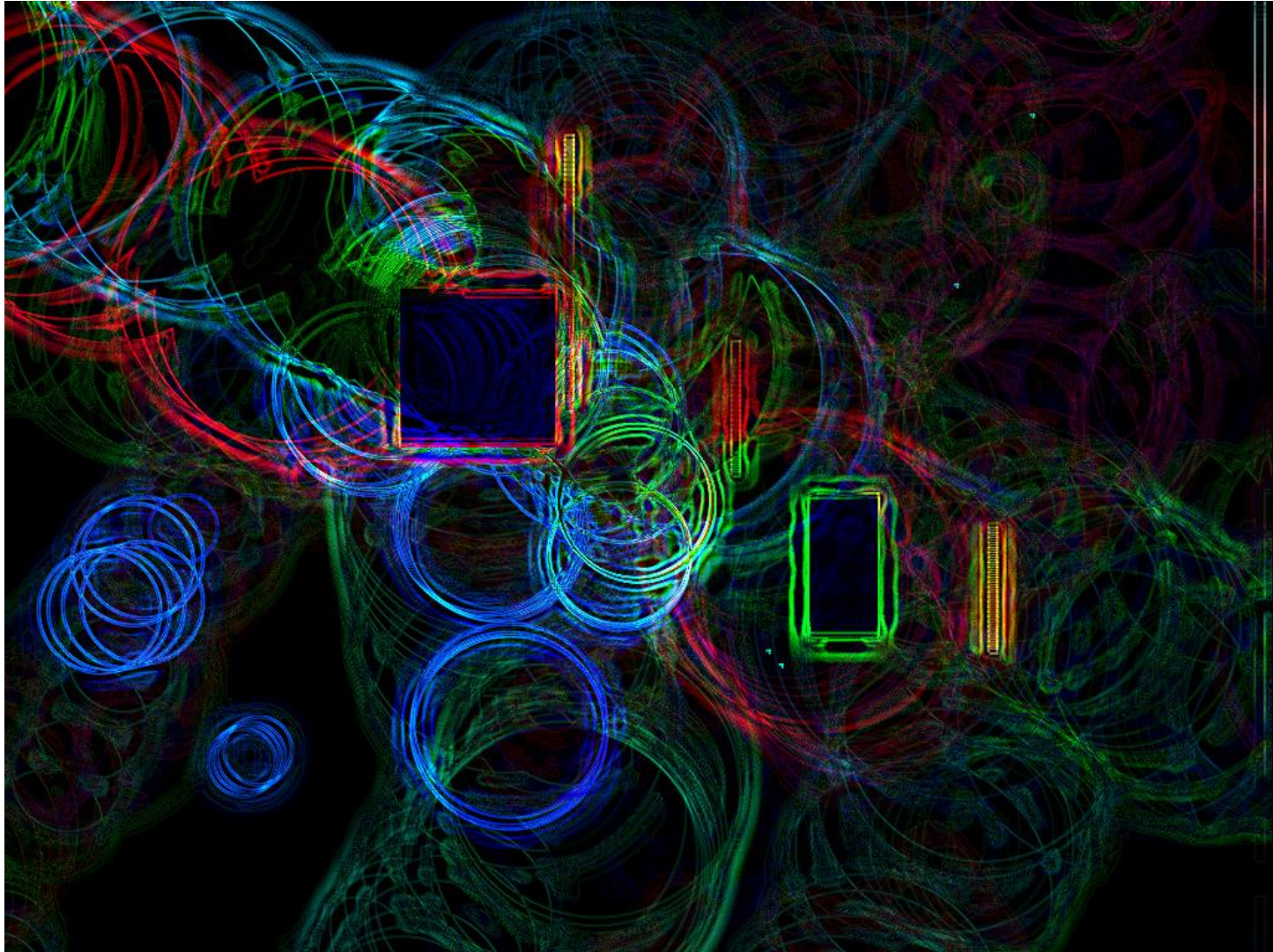
Examples of Visual Output



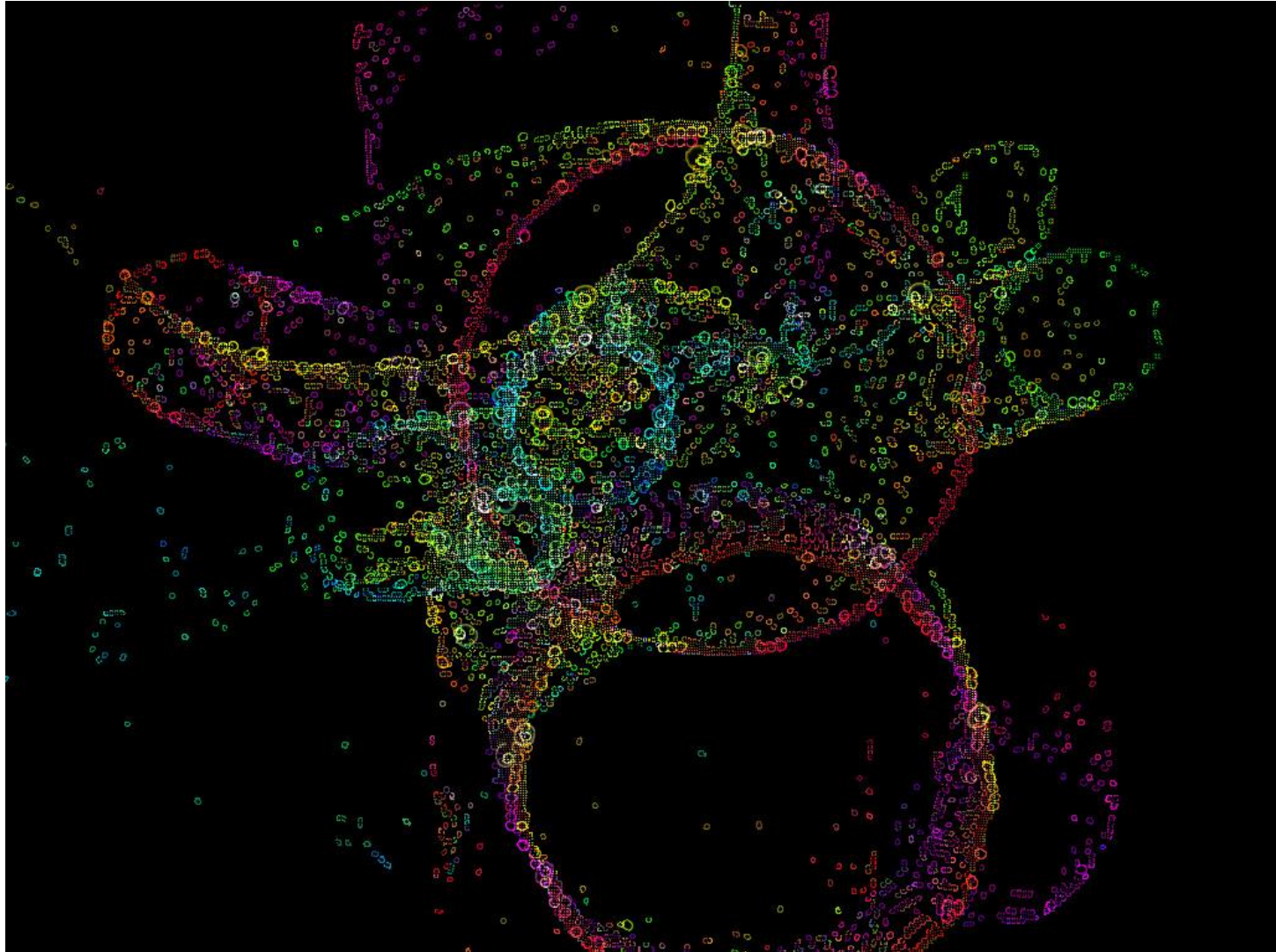
Examples of Visual Output



Examples of Visual Output

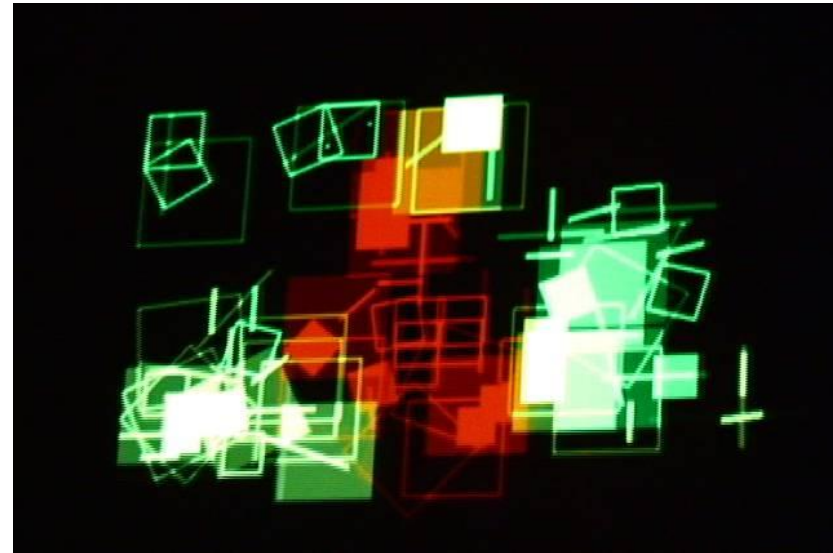


Examples of Visual Output



Double Vision

- Collective of dancers, musicians, and media artists
- Free innovation within a particular theme
- First show's theme was DNA
- Installation consisted of Conway's game of Life, audience could add DNA letters as patterns to it, and the generations of cells would trigger music and graphics



Double Vision – SpectraBall

- Dance pads used to control balls bouncing around in a maze
- Balls hitting the walls would trigger sounds in 4 speakers surrounding you - the sound location would match the direction in which the wall was hit



Double Vision – Cellspace show

- Steering wheel controllers used in two installations:
 - Art of Driving: drive around and “fire” graphics
 - Bouncing off the Walls: drive around a maze, firing balls, resulting in 4-channel music around you

[Movie](#)



DoubleVision – performing graphics

- Red Ink Studio – performing graphics with dancers and musicians



Finger Fresco 2.0

- First attempt at playing music and generating graphics simultaneously in an actual performance
- Used Fingerworks multitouch pads for playing music (same controller built for performing graphics, previously)
- Music keyboard controlled scales/chords
- Notes of the music triggered graphics
 - [Movie](#)

LoopyCam

- Camera-based visual performance instrument
- Performer controls camera position and visual processing with a single integrated device – a security camera screwed onto a \$10 game controller
- First version used a USB webcam, but lighting was always an issue.
- Latest version uses a security camera which automatically turns on LEDs in low-light situations.



LoopyCam – how it works

- VVVV hosts FreeFrame plugins for visual effects
- One custom FreeFrame plugin records up to 8 video loops and controls their playback and positions
- KeyKit reads joystick buttons and sends OSC and MIDI messages over to VVVV to control the effects and looping
- Extensive “chording” of the buttons allows a large number of operations to be performed with the game controller’s buttons



LoopyCam Evolution

- First version restricted to 4 loops and quadrant positioning
- Was used at a DoubleVision event – showed great promise, especially for use with dancers
- Was installed in “automatic” mode in the window of the ATA Theater for the entire month of December, 2009, capturing and looping passers-by
- Lots of effort spent on making the installation foolproof and able to run reliably, because I was 50 miles away.

LoopyCam Evolution

- FreeFrame plugin enhanced to allow more flexible control and positioning of the loops
- Performances
 - Loop Salad solo performance at Luggage Store
 - This Here shows at Temescal Arts Center
 - SHARE San Jose jams at Villa Montalvo
 - With a Butoh dancer at Zeum in SF
 - I, Norton opera at the SF Electronic Music Festival

Galaxy – a Visual Music performance

- New Nothing Theater, part of Visual Music meetup
- Looping music played on a normal keyboard
- Graphics triggered by the notes of the music
- Graphics and music controlled by the “Finger Painting with Planets” controller
- Graphics makes use of Python within a FreeFrame plugin, allowing post-processing of the graphics with other FreeFrame plugins
- [Movie 1](#)
- [Movie 2](#)

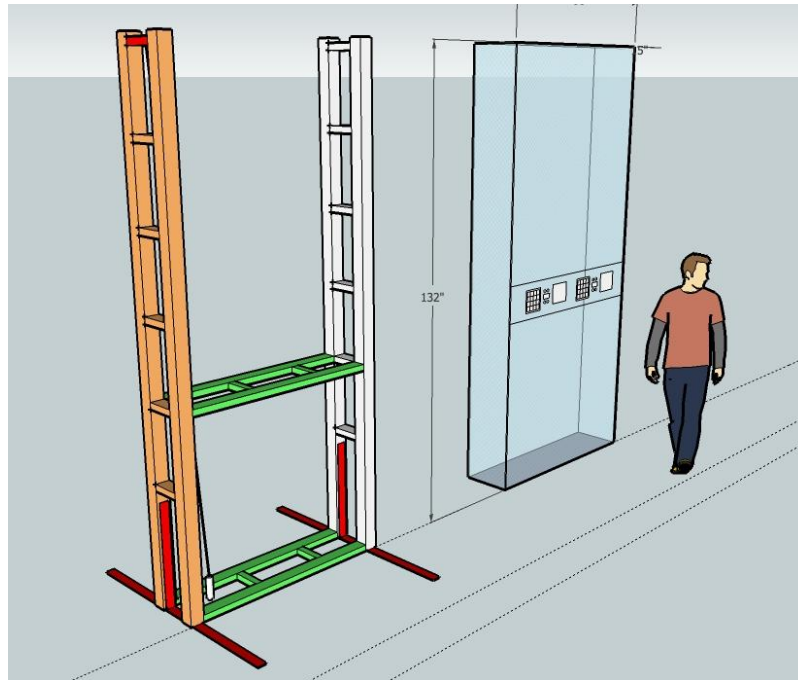
Loop Salad

- Combination of Galaxy and LoopyCam
- Visuals generated by Galaxy were projected, and also fed to LoopyCam (in “automatic” mode) to be processed and projected with a second projector
- Movie

Monolith 2.0

- Burning Man 2009 theme: Evolution
- 2001 Space Odyssey monolith
- One side is a highly evolved musical instrument
 - Two-person looper with over 100 controls
- The other side is a simple visual instrument
 - Chalkboard and chalk
- Built in my back yard over the summer
- Controller panels are usable independently

Monolith 2.0 – the construction



- Built to withstand 80 mph winds
- All battery-powered (swapped daily, recharged with solar panels at camp)
- Top had solar-powered fans for ventilation (though not really necessary)

Monolith 2.0 – in my backyard



Monolith 2.0 on the playa



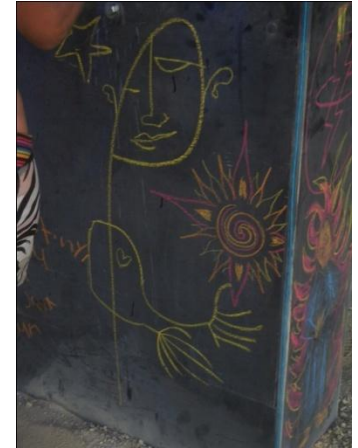
Monolith 2.0 – on the playa



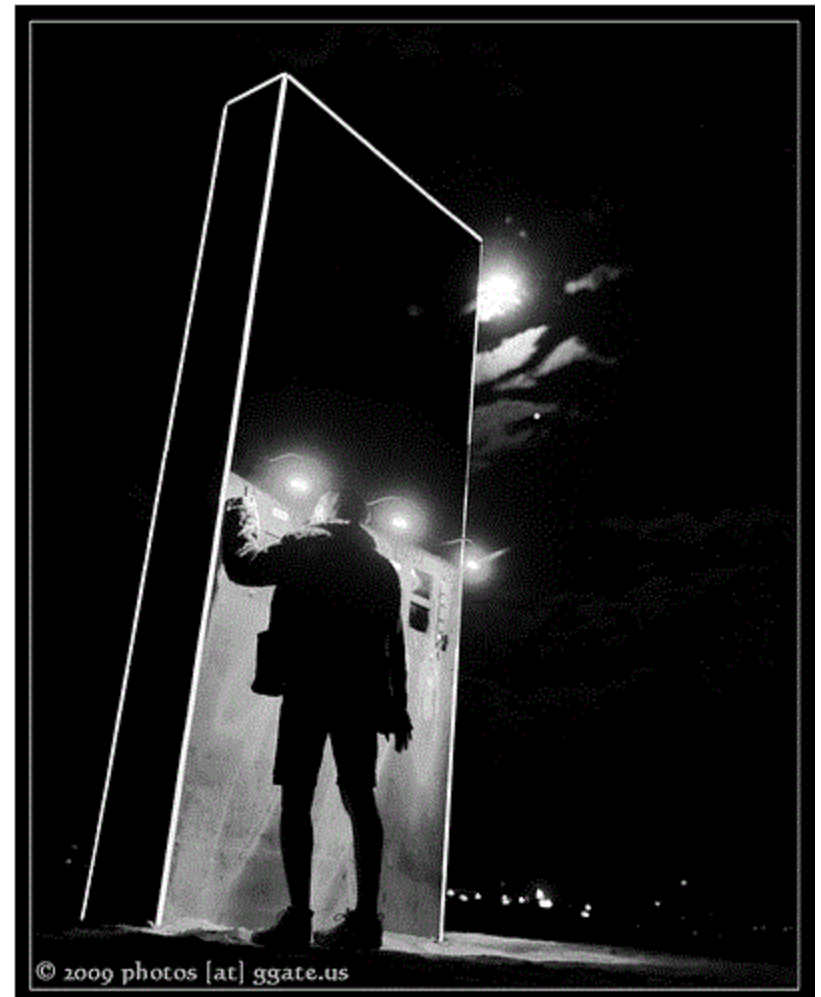
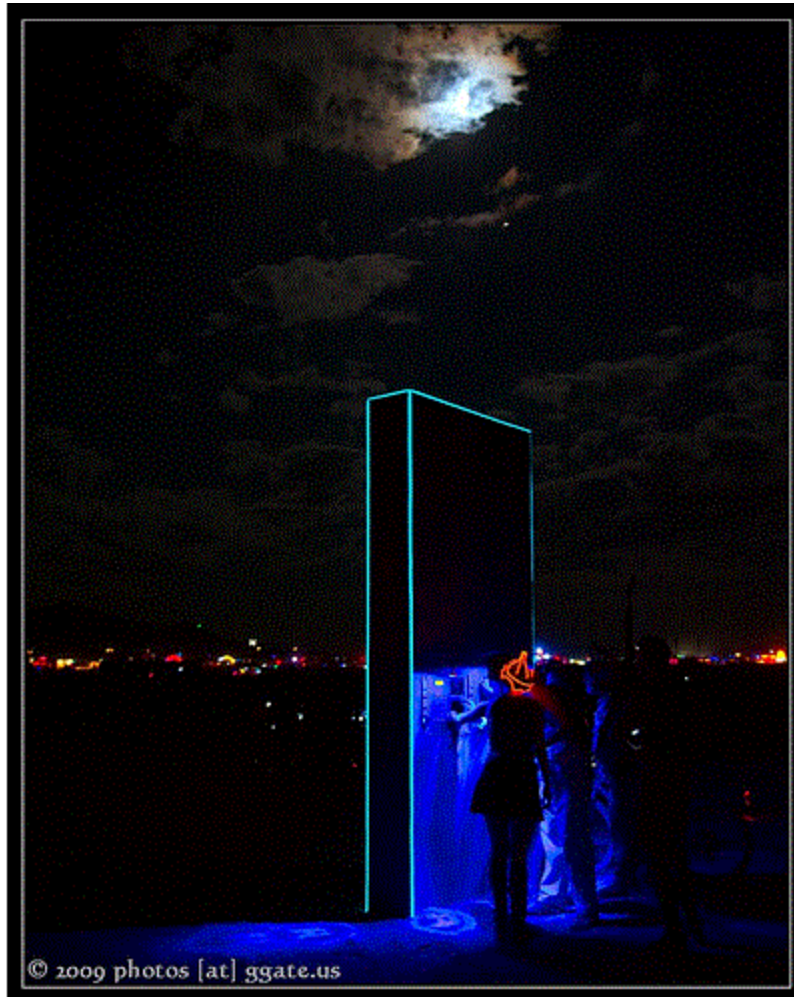
Movie



Monolith 2.0 on the playa



Monolith 2.0 on Flickr



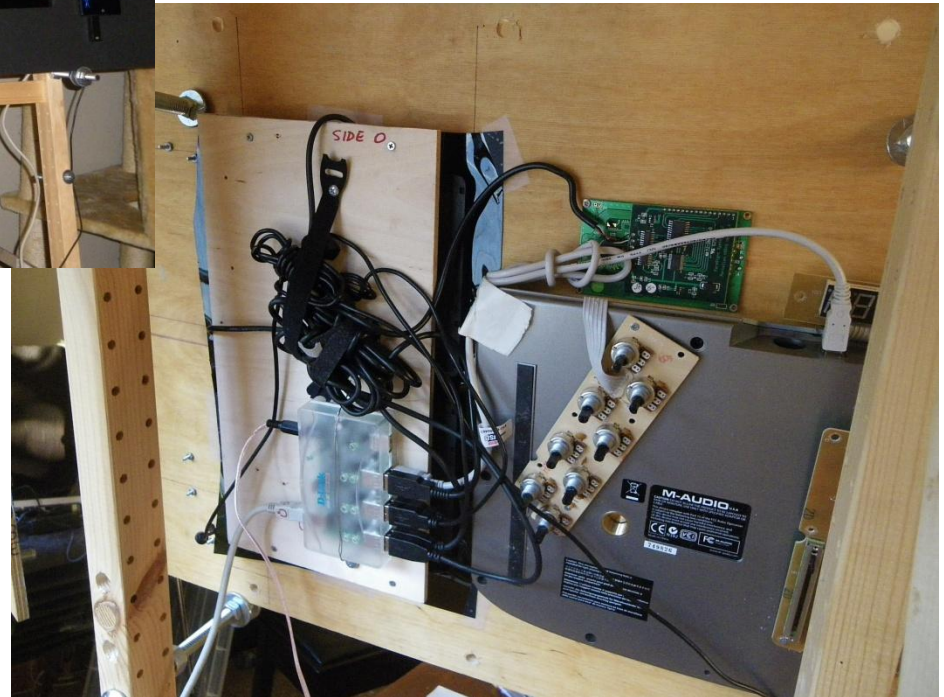
What's the controller part all about?

- Two independent controllers
- Each one is a 5-track looper - 1 track each for lead/bass/pads/drums/other
- The two loopers share tempo, scale, and transpositions, so they don't conflict musically
- Lots of labeled buttons to control:
 - Loop length, fading out
 - Chords, sounds, scales
 - Saving/loading loops
 - Audio effects (using one of the iGesture pads)

Monolith 2.0 – the controllers



- Korg Nanokeys used as buttons
- M-Audio Trigger Fingers used for drum pads



Buttons

- Originally was going to use arcade buttons
- Worldwide shortage of buttons due to release of Street Fighter IV
- Korg Nanokeys are cheap and come in black
- Convenient for labels
- Removed configuration buttons and covered in flexible acrylic for dust



Other Hardware

- Power consumption was a primary concern
- Asus Eeebox - Atom N270 running Windows XP, hosting a total of 15 USB devices
- Edirol UA-20 audio interface
- Sonic Impact (T-class) amplifier and two 6x9 speakers
- FM transmitter
- Lights - EL-wire outlining the monolith, and gooseneck LEDs
- 12 Volt deep-cycle batteries, swapped daily, charged with solar
- Battery 1: computer, amplifier, and both controllers (33 watts)
- Battery 2: EL-wire, lights, and FM transmitter

Software

- **Keykit**
 - Handles all input: 4 Nanokeys, 4 iGestures, 2 Trigger Fingers
 - MIDI looping
 - Controls the 2 LCD displays
- **Plogue Bidule**
 - VST hosting
 - Excellent routing features
- **Native Instruments**
 - Primarily FM8 (low CPU usage)
 - Battery 3 for drums

Feedback and Lessons Learned

- Both sides were well-used
- Burners always draw outside the box
- Message boards on the playa are useful
- People had lots of fun with the looper
 - At all hours of the day and night
 - Many came back repeatedly
 - Some people actually read labels
- Best feedback: non-musicians made music **and** realized that **they** were the ones making it

LoopyCam 2



LoopyCam 2

- LCD display displays menus and status
- Now usable by people other than myself
- Number pad (with chording) allows more operations
- Cinder-based application
- Uses both Freeframe 1.0 (raster-based) and FreeFrame 1.5 (OpenGL) plugins

Recent Focus

- 3D continuous input with hands
- Third dimension can be:
 - Pressure (Continuum, Eigenharp, Linnstrument, etc)
 - Area (Fingerworks, Magic Trackpad, iPad)
 - Vibration, Orientation (smart phones)
 - Depth (Kinect)

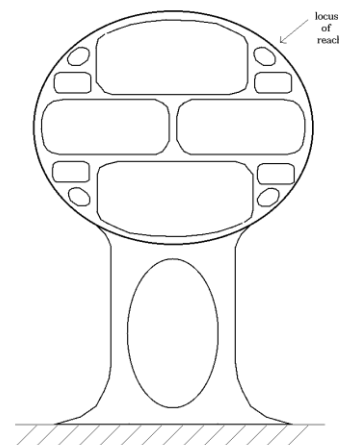
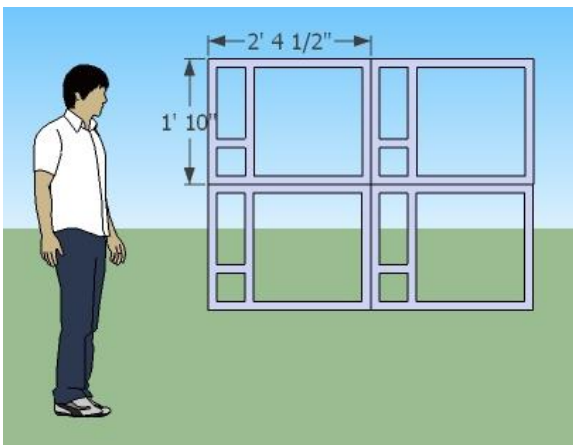
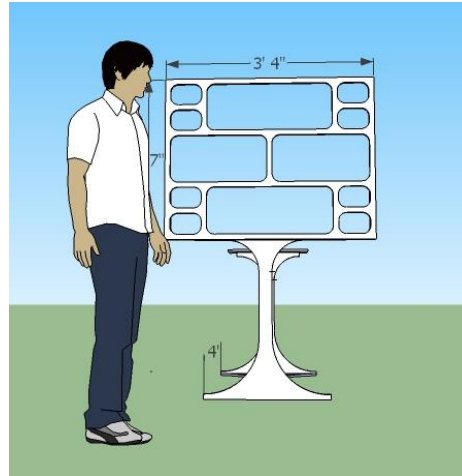
Kinect !

- Inexpensive and ubiquitous
- Good resolution and robustness
- Easy to use from C
- Tolerant of dust (yay!)
- Intolerant of sunlight (boo!)

Space Palette - a 3D Instrument Interface

- Holes in a frame become 3D multitouch surfaces
- Any number of hands or objects, simultaneously
- Flexible layout allows many control possibilities
- Immediate access to different sounds/graphics
- Provides frame of reference for player and audience
- Larger visual footprint is more interesting to audience
- Immediately playable, no initial dexterity required
- Larger and less-restricted motion by player is relaxing and expressive

Space Palette - Evolution and Variations



Movies

[Lightning in a Bottle 2011](#)

[Burning Man 2011](#)

[SF Decompression 2011](#)

[LoopFest 2011](#)

[Symbiosis 2012](#)

[Burning Man 2012](#)

Controlling the Music

- Each large hole plays a different sound
- Horizontal position is pitch
 - All notes forced onto a particular scale and key
 - Typically two octaves across
- Vertical position controls timing quantization - “time frets”
 - Three bands: one beat, half-beat, quarter-beat
- Depth position:
 - Converted to MIDI aftertouch, used for vibrato, filtering, and mixing
- Small holes are buttons
 - Performance UI - change key, scale, sounds, looping
 - Casual UI – select presets

Controlling the Graphics

- Each large hole is an independent 3D drawing surface
- Each hole's drawing has independent shape/color/motion
- Depth controls the size of graphics
- Small holes are buttons
 - Performance UI - change color, shape, motion
 - Casual UI – select presets

Types of Instruments

- Casual Instruments

- Prioritize fun and enjoyment
- Few or no instructions
- Immediate gratification, no learning curve
- Path to proficiency is nice to have, but not required

- Performance Instruments

- Prioritize proficiency and control
- Instructions or training usually required
- Proficiency requires practice, learning curve
- Visualizations (real or virtual) for audience are beneficial

Where does the Space Palette fit?

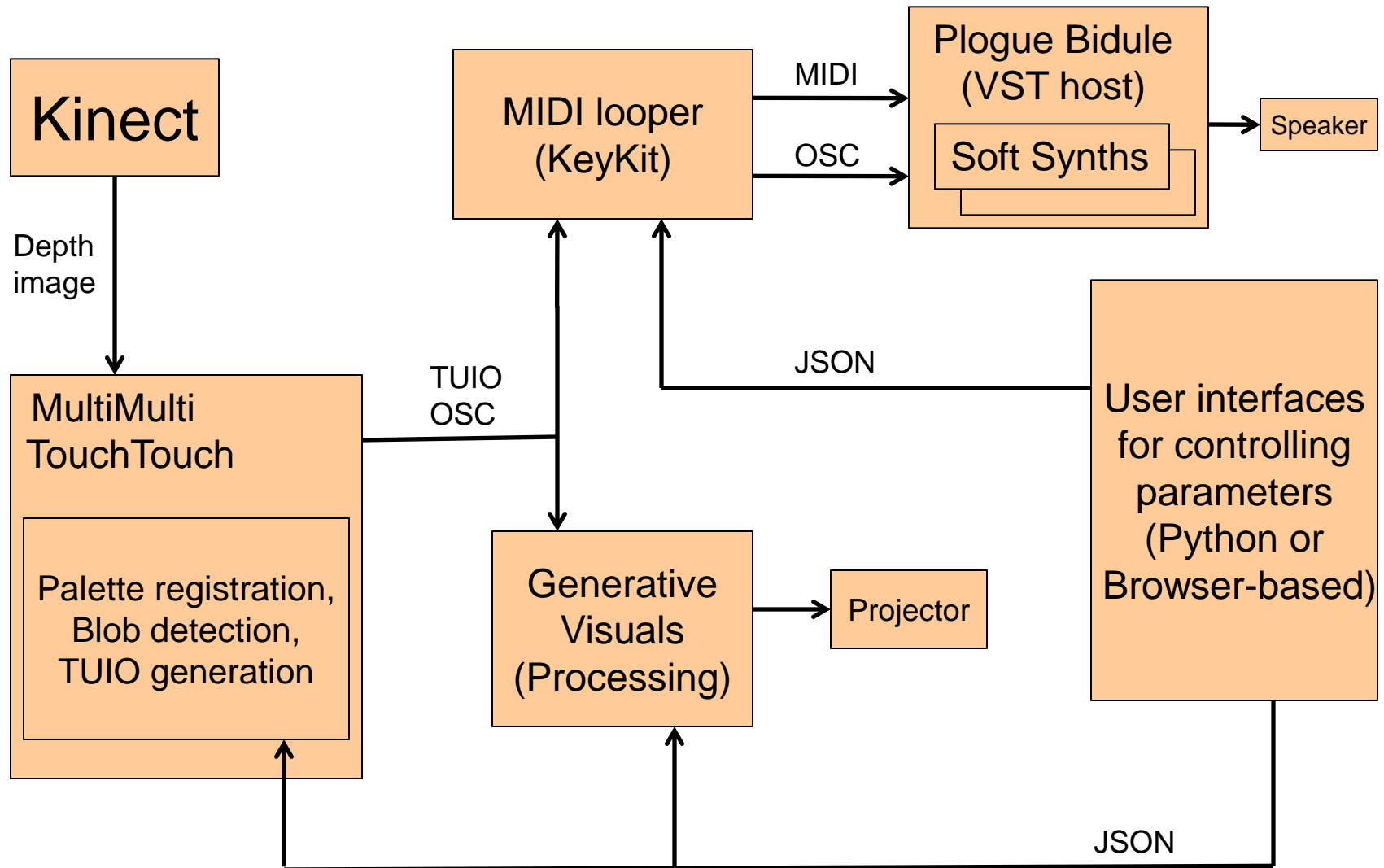
- As a Casual Instrument
 - No learning curve: walk up, play, sound good
 - Natural interaction, effortless, engaging
 - Control over individual notes
 - Players recognize that they're the ones controlling it
- As a Performance Instrument
 - Physical presence is more visible and entertaining
 - Performer's larger movement engages the audience
 - More obvious correlation of physical actions to output
 - Frame of reference allows more and better control

Where does the Space Palette fit?

- Both casual and performance?
 - User interface is greatly affected by the choice
 - A single physical interface can serve both
- Casual use is the current sweet spot
 - Confirmed by several years at Burning Man and other events
 - UI continues to be simplified, visual feedback added
- Performance use has been explored
 - Several solo and collaborative performances
 - Several UI style experiments (e.g. shift-select style)
 - Open source MultiMultiTouchTouch allows others to explore

Initial Evolution - 2011

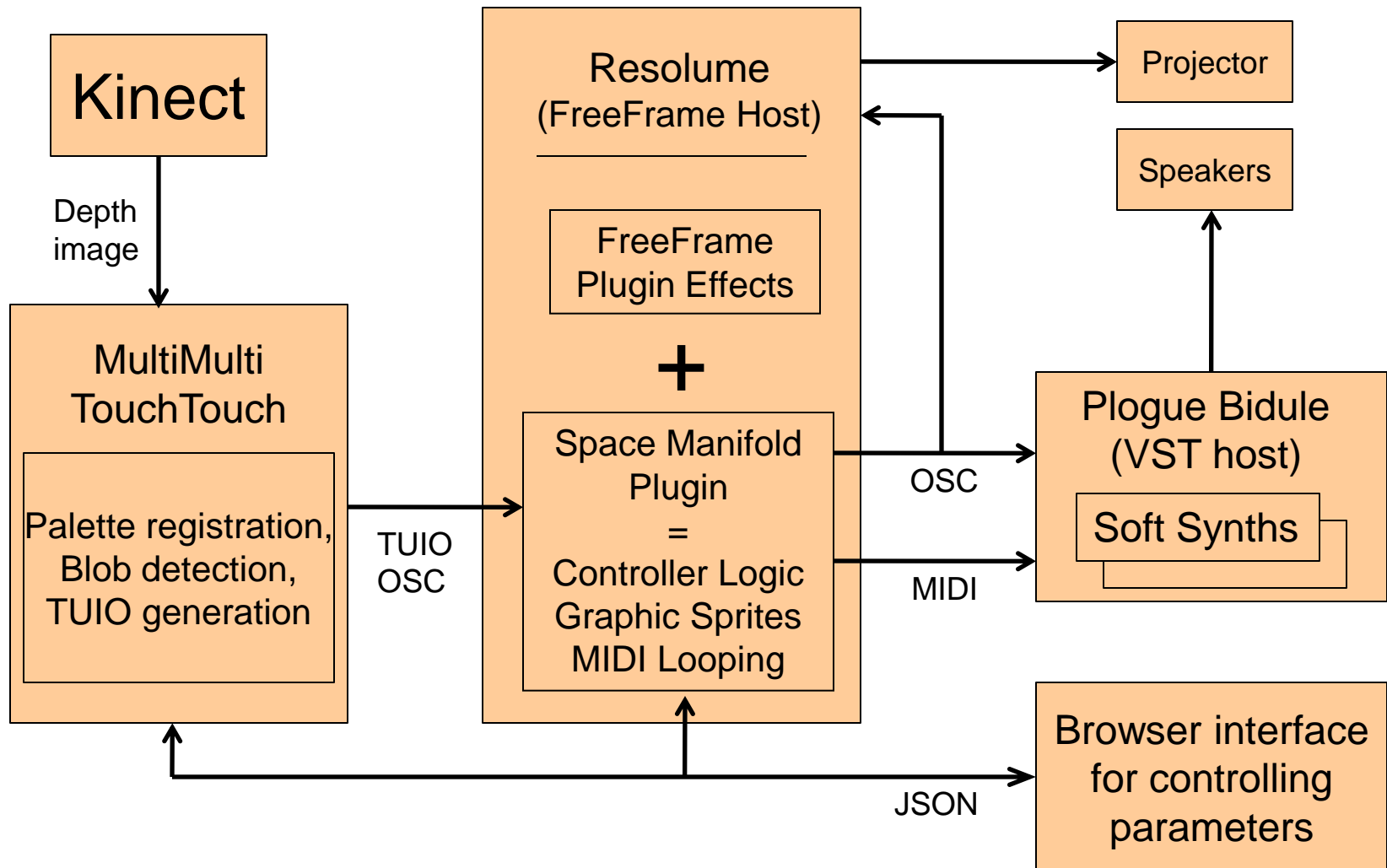
- Initial prototype: 4 regions
- A little more control: 7 regions, 8 buttons
- Lightning in a Bottle
- West Coast Controller Battle
 - Tennis Ball !
- Simultaneous graphics using Processing (Java)
- Burning Man 2011
 - Multi Multi Touch Touch theme camp
- MusicTech Summit, Venice Art Crawl, Decompression, etc



Recent Evolution - 2012

- Oval version – 4 regions, 12 buttons
- FreeFrame plugin inside Resolume
 - Eliminates KeyKit and Processing
 - More complex visual effects using other FreeFrame plugins
 - Resolume can be controlled with OSC
 - HTTP listener, JSON API = browser-based UI for parameter control
 - Single OSC listener and looping mechanism, better synchronization
 - Potential for interaction between graphics and music
- Python integration within FreeFrame plugin
 - Interactive development, more robust error handling
 - Live coding
 - Easier code sharing and distribution

Design - 2012



Things Observed and Learned

- Casual use vs. performance use influences the design a lot
- Small holes are magnetic
- Labels are rarely read
- Musicians know how to rest, listen, and be selective
 - They aren't the only ones who can do those things
- Looping can be confusing
- Multiple users is fun, but can be confusing

More Things Observed and Learned

- Correlation of graphics to hands is often not noticed due to extreme visual effects
- People love seeing their hands (debugging display)
- Effect of depth often needs to be explained, but is immediately appreciated
- Time-frets aren't intuitive, but provide useful variety even if you aren't aware of how it works
- Hand motion tendencies limit the pitch range used:
 - Depth-only with no up-down or left-right motion
 - Up-down with no left-right motion

Comments about the Space Palette

- Most common:
 - I want one in my living room.
 - I could stay here all night.
- Most interesting:
 - Why, it really opens up what an instrument is, right?
 - You gotta try it, you gotta try it, you gotta try it!
 - For those who can't cross that barrier [of playing music], they're literally crossing that barrier [hands reaching through].

Software

- MMTT (MultiMultiTouchTouch)
- Resolume (FreeFrame host)
 - Projection mapping and visual effects
- Space Manifold (FreeFrame Plugin)
 - Receives TUIO/OSC, generates graphics AND music
 - Looping mechanism
- Plogue Bidule (VST host)
- VST Soft Synthesizers
 - Battery 3, Alchemy
- Browsers (local and remote)
 - User interface (and JSON API) to control MMTT, Space Manifold

MultiMultiTouchTouch (MMTT)

- C++ program uses libfreenect to talk to Kinect
- Uses depth image only
- Blob detection using OpenCV
- Trainable interactively on new frames, holes of any shape
- Trainable without a frame, using a specially-colored image
- Browser interface to control it, using JSON over HTTP
- Output is TUIO (a standard multitouch format) over OSC (a standard UDP protocol)
- Windows-only, source code freely available:

<http://multimultitouchtouch.com/dist>



No Such Media

Tim Thompson

<http://timthompson.com>

me@timthompson.com