PORTFOLIO

Adam M. Smith

Computer Scientist, Design Researcher, Software Artist

December 2013

Biography

Adam M. Smith is a hacker from Silicon Valley, coming of age just as the Dot-com bubble was bursting, who has most recently set up shop as an academic in the Seattle area with the University of Washington.

With **software** as his primary medium of expression, he is concerned with machine generativity and the potential for software to promote creativity and synthesis activity in others, whether they be software engineers, musicians, or plain folk on the internet.

He is formally trained as a computer scientist (with a BS and PhD from UC Santa Cruz), and his artistic experience derives from the pursuit of curiosity around what *else* can be done with these computer things. He has often considered being an artist as only a tertiary facet of his identity, but it's definitely on the list.

Contact Information

https://adamsmith.as
adam@adamsmith.as
(831) 295-2624

Selections

Here are few creative software projects I've done over the years. I picked ones with fun stories and pretty pictures.

Sketch-a-bit

Ethereal Dialpad

Xorex and Fusepuck

Livecoding and CFML

Troy

Tableau Machine

Design Grammars

Drive-by CTF

Sketch-a-bit (2010-present)

Collaborative art ecosystem in the form of an Android app and central web service.

http://sketchabit.adamsmith.as/

https://play.google.com/store/apps/details?id=com.superfiretruck.sketchabit2

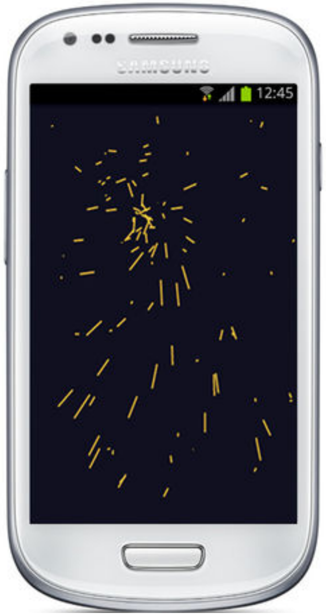
When the user launches *Sketch-a-bit*, they are greeted by a random sketch uploaded by another user of the app. After making changes to the sketch using a velocity-sensitive tool inspired by charcoal drawings, she uploads her new creation. The system keeps track of which sketch inspired other sketches, and the resulting family tree can be traversed to examine patterns of distributed, asynchronous, anonymous, and indirect collaboration.

For several years, only black-and-white drawing tools were available. We recently introduced the color red. As of December 2013, more than 150,000 sketches have been uploaded by users.

This piece was created with Kathleen Tuite.

For a review of collaboration patterns we found, see "Emergent Remix Culture in an Anonymous Collaborative Art System" (Tuite and Smith, 2012).

Ethereal Dialpad (2009-present)



Minimal synthesizer toy in the form of an Android app.

https://play.google.com/store/apps/details?id=as.adamsmith.etherealdialpad

Ethereal Dialpad (ED) is an expressive touch synthesizer in which the 2D screen location of the performer's finger controls the pitch of two independent synth lines, snapped into a tasteful key.

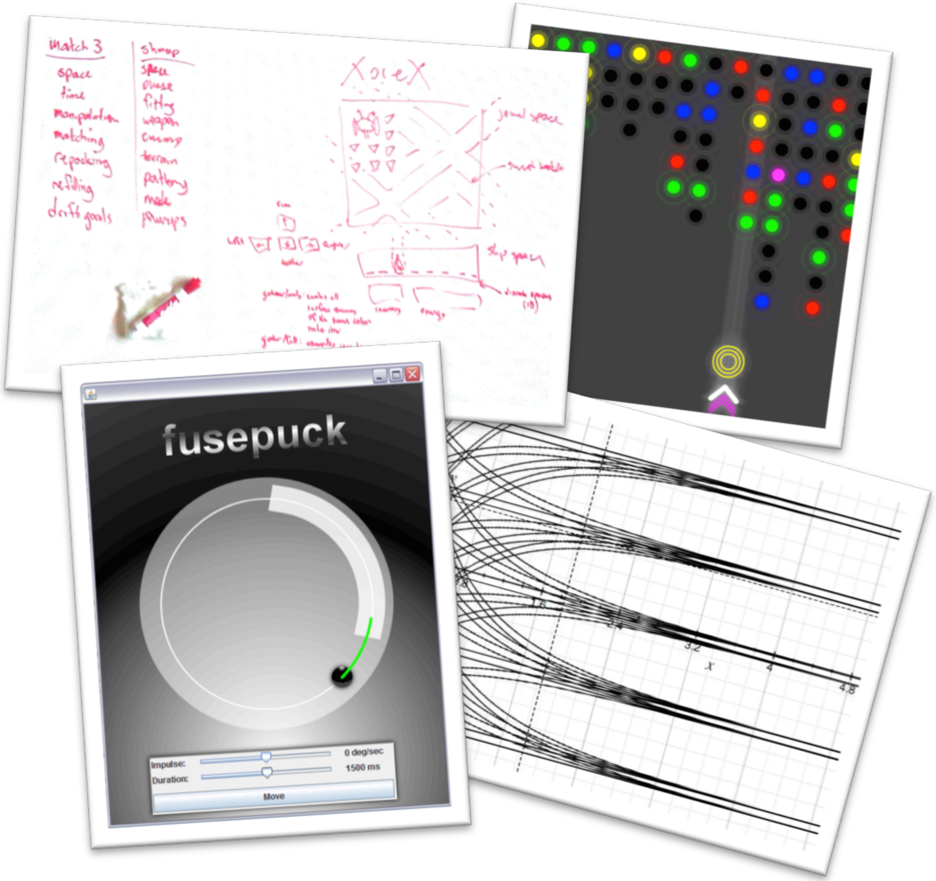
ED has been downloaded over ten million times worldwide.

The emergence of ED early in the Android music scene prompted an interview article in the influential blog *Create Digital Music.*

ED's inspires programmers. *Circle Melody Pad* (Philemon Merlet) is a free plugin for ED that provides alternative visualization. Many more plugins have been created by other developers, one even by an individual who learned to program in Java *just* so they could make the plugin. Meanwhile, *MonadPad* (Michael Helland) is a series of synthesizer toys that integrate cloud-based sharing of sonic sketches. These toys recycle both ED's synthesis engine and touchpad interaction schemes. *Saucillator* (Matt Feury) also recycles ED's synthesis engine to create a new stand-alone toy with looping and recording functionality.

ED inspires musicians. "The Rise" by COOLOUT (Christopher Davis) is a hip-hop album created with only Android devices and software, including ED. "Epoxy" by Sound in the Mist (Wes Richmond) is an ambient album that uses ED for several voices under heavy processing. Shon Parker at the McNally Smith College of music uses ED in his vocalist training classes, and countless others have reported using it for casual inspiration in their professional and amateur musical practices.

Xorex and Fusepuck (2009)



Impulse: 0 deg/sec

Duration: 1500 ms

Experimental gameplay prototypes in Flash and Java.

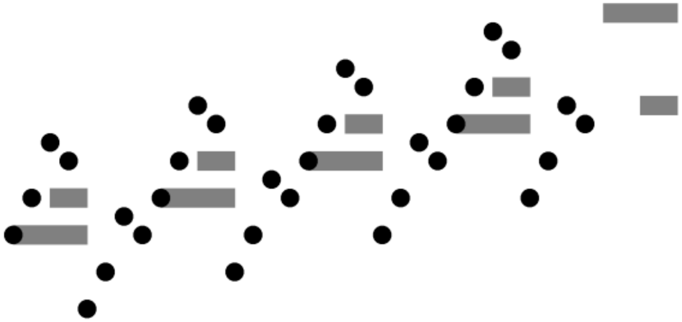
During 2009, I created a series of experimental games that mashed up mechanics and aesthetics from different genres to yield gameplay with unexpected dynamics. These experiments informed my ongoing technical research in game design automation.

Top: *Xorex* combined shooting mechanics drawn from *Space Invaders* with colored-matching mechanics drawn from *Bejeweled*. The result was a frantically paced resource management game in which colored gems must be sorted and combined to provide the ammunition needed to beat back the slow creep of advancing gems. Although I could only survive in my own game for about five minutes, my analytics show that one player on Kongregate was repeatedly able to survive through 45-minute sessions.

Bottom: *Fusepuck* took the ballistic trajectory plotting mechanics from artillery games like *Scorched Earth*, replaced parabolas with exponentials, and re-skinned the result as an abstract competitive sports game with strategic time manipulation. Various strategies over the game's non-linear dynamics were explored with a prototype created on a graphing calculator.

Livecoding and CFML (2008)

```
(define (bump)
   (literal 1/2 '((0 3 0 1 1/2))))

(define (lump)
   (after bump
          (tra 2 bump)
          (tra 4 bump)
          (tra 5 bump)))

(define (string-step)
   (literal 2 '((0 4 0 1 2)(1 4 7 1 1)(0 4 2 1 2))))

(define (string-end)
   (literal 2 '((0 4 4 1 1)(1 4 -1 1 1))))

(define (song)
   (choose (vol 2/3 string-end)
           (tra +2 (after (during string-step
                                 (after lump (tra -4 lump)))
                         song))))

(perform song 120 (pc:scale 0 'dorian))
```



Various musical improvisations and a new programming language for livecoding musicians.

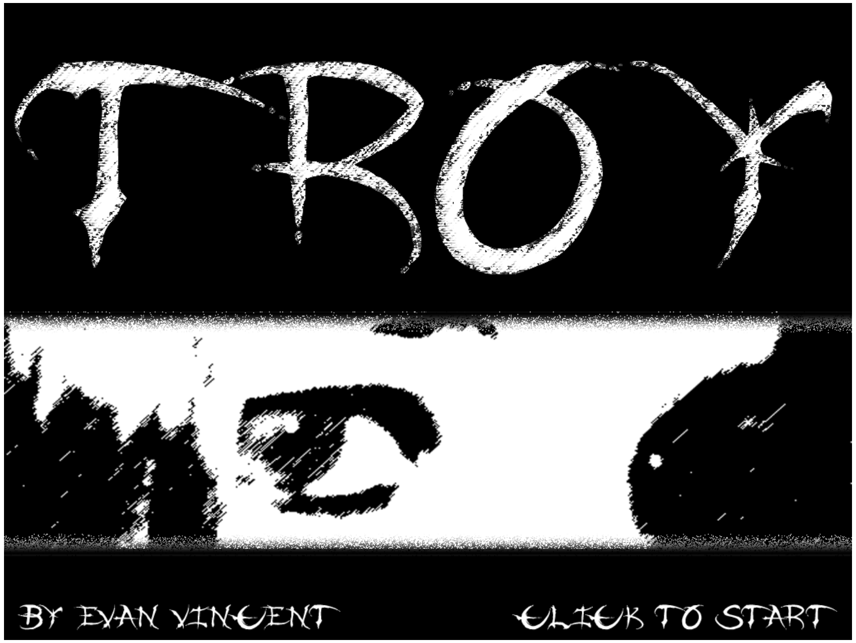
During 2008, I became interested in livecoding. Livecoding is improvised interactive computer programming, often used to create digital media including sound and images. In my practice, I learned to use ChucK, Impromptu, Fluxus, Supercollider, and PureData. I combined these specialized livecoding systems with my previous experience with Processing and Python to jam with friends, creating new musical instruments from game controllers and webcams as the inspiration arose.

I loosely associate with TOPLAP (The "(Temporary|Transnational|Terrestrial|Transdimensional) Organisation for the (Promotion|Proliferation|Permanence|Purity) of Live (Algorithm|Audio|Art|Artistic) Programming") collective of livecoding musicians.

Taking the step from practice with others' livecoding systems to the creation of my own system, I developed CFML, the context-free music language (https://github.com/rndmcnlly/cfml/wiki). CFML is a sub-language of Impromtu's TinyScheme designed as a musical analogy of ContextFreeArt and StructureSynth for 2D and 3D geometric visual art. I regularly use examples from CFML and these other languages in introductory programming courses to explain the abstract concepts of composition (versus inheritance), recursion, and nondeterminism. The strong structural analogies between the three languages allows experience in one to inspire creative action in the others.

See "Strange Loops in CFML: A Livecoder's Riddle" (Smith 2012) for an exploration of computer music theory guided by concrete insights from CFML.

Troy (2007)



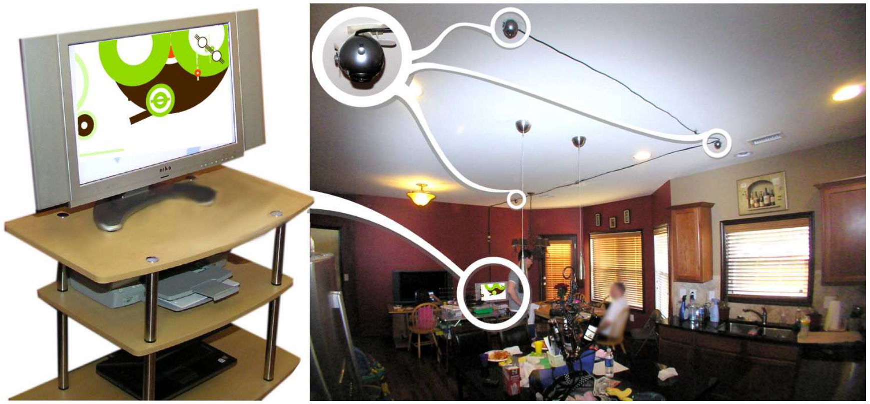
An experimental non-fiction text adventure.

http://adamsmith.as/games/troy/

Troy is a work of interactive non-fiction created with Inform 7. Using textual commands such as "click [link]" and "examine [image]", the audience recreates my experience of playing a web-based alternative reality game (ARG), also called *Troy*, created by TJ Jackson in 2005. As of 2007, the 2005 ARG had partially decayed, leaving pages explaining that the game was indeed a game as dead links. In Jackson's *Troy*, the player searches for a prototype game called *Troy* created by the fictional developer Evan Vincent, which turns out to be a fiction of its own (oh, uh, *spoilers*). Multiple layers of fiction and non-fiction are tangled in an experience that is both a game and a primary and secondary source of historical documentation for several other prototype games (including one that, retroactively, never admits to *not* not being a game).

```
> examine self
You don't have a webcam.

> go north
Your cursor is constrained to the display in front
of you.

> drop fileshare login
In trying to forget it, the fileshare login
becomes more strongly rooted in your memory.
```

Tableau Machine (2007)



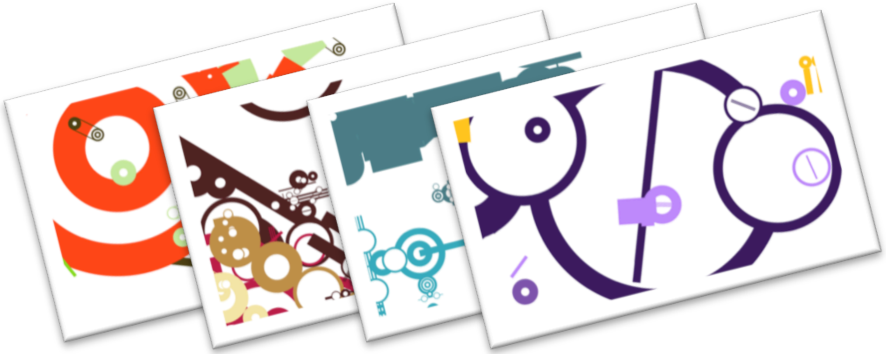


Tableau Machine (TM) was a long-term generative art installation with networked cameras, display, and printer. TM was deployed in three homes for two months as part of a human-computer interaction (HCI) research project related to the study of affective presences.

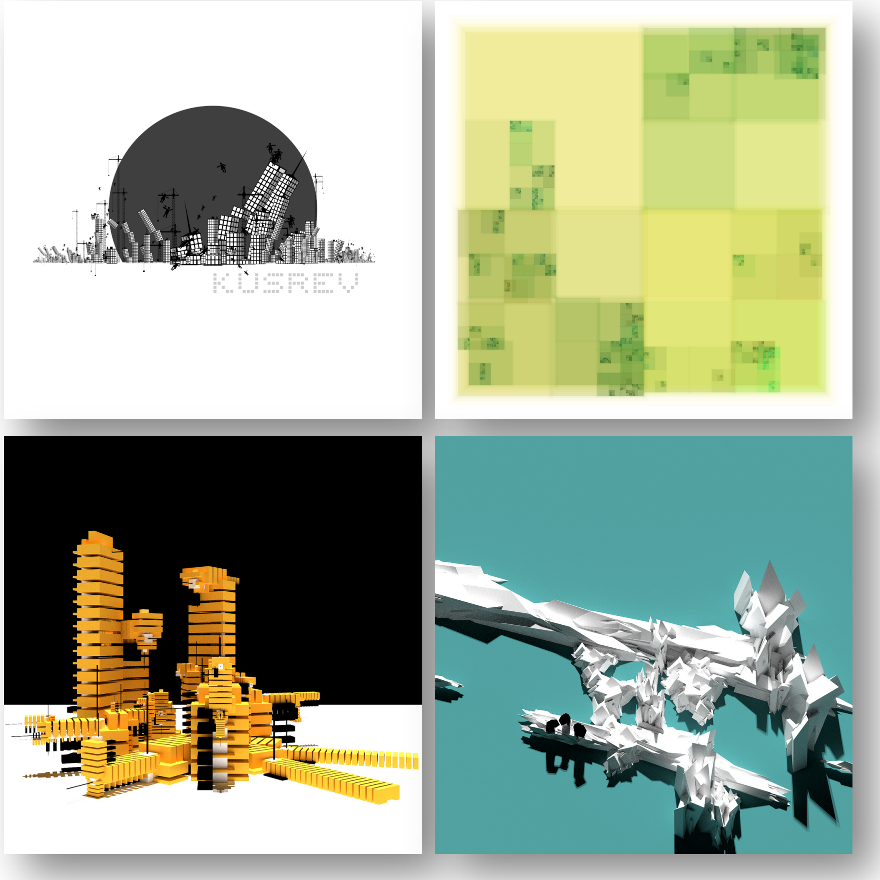
This piece was created in collaboration with Michael Mateas, Zach Pousman, and Mario Romero at the Georgia Institute of Technology. I created the pattern detection component (which dynamically clustered activity in the house into abstract patterns) and the visual synthesis component (which conveyed these abstract activity patterns with the system's alien language based on context free design grammars).

The abstract art generator was displayed in the gallery at the Beall Center for Art and Technology at UC Irvine.

For a description of my artistic and techincal contribution to the piece, see "Tableau Machine: A Creative Alien Presence" (Smith, Romero, Pousman, and Mateas 2008). For the HCI findings of this experiment, see "Living with Tableau Machine: A Longitudinal Investigation of a Curious Domestic Presence" (Pousman, Romero, Smith, and Mateas 2008).

An interactive, generative art installation.

Design Grammars (2005-2009)



Several pieces of generative visual art in the form of context-free design grammars.

After being introduced to Chris Coyne's CFDG (context free design grammars) at an early SuperHappyDevHouse event, I began a long period of on-and-off exploration with CFDG and its later descendants. Of hundreds of compositions, I have selected two 2D examples and two 3D examples.

Top-left: An exploding city with a randomly generated pronounceable name. This piece literally inserts chaos generators into the cracks of a fairly traditional application of shape grammars to building architecture.

Top-right: An abstract pastoral scene evoking views of farmland from the air. This piece uses offset layering and structural recursion to explore color mixing.

Bottom-left: Greeble built from a single repeating white-black-yellow unit. The use of an architectural projection (two-point perspective) and rendering with path tracing evoke concept architecture while the hard white and black background reinforce the abstract geometry.

Bottom-right: Three penguins on an iceberg. The icy climate is evoked more with geometric sterility than direct representation. Only in very few random variations do all three penguins manage to be placed together on an appropriate sized iceberg and not to float mid-air.

Drive-by CTF (2004)

An outdoor game for cars and laptops.

~~http://drivebyctf.com~~

(I let the domain name expire years ago.)

Drive-by CTF is a game of capture-the-flag played on a multi-neighborhood scale. Two teams, usually composed of one driver and three operators, compete to connect to as many open wireless networks as they can during a match.

When a match begins, team members enter their vehicles and boot their laptops. As they drive through the quiet streets in early-AM hours, they look for open wireless networks. Upon connecting to a network, they try to load the game's webpage for the match. The game checks to see if the player is connecting from a never-before-seen public IP address. If the team did not have the flag, they steal the flag. If the team does hold the flag, the team scores a point.

Over the course of several sporadic months of play, players developed a number of interesting strategies and tactics. Some players taped external wi-fi antennas to the outside of their cars for better reception. Others wrote custom software to sonify the signal strength of nearby access points, providing better feedback to the driver than verbal communication from the operators did in the past. Some even devised time-coordinated attacks in which a driver would leave operators behind on a long street and then use a flash of the vehicle's headlights to signal for them to all try to connect at once.

I designed and implemented the game with Matthew Rossetta. Before the widespread adoption of internet-enabled cellular networks, much remote debugging could *only* be done after connecting to each a new house.