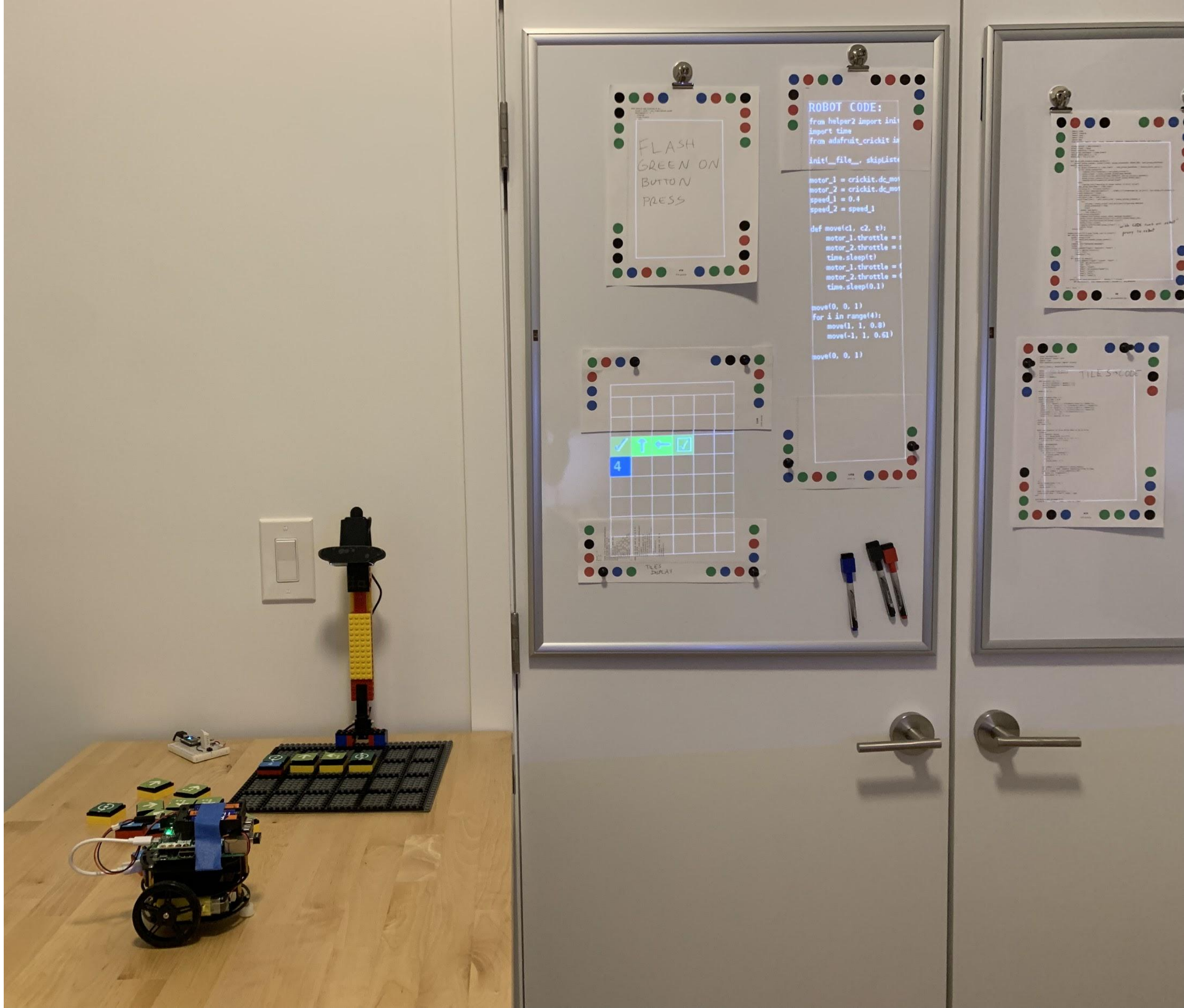
Jacob Haip Portfolio

October 2019



ROBOT CODE:

```
from helper2 import init
import time
from adafruit_crickit in

init(__file__, skipListe

motor_1 = crickit.dc_mot
motor_2 = crickit.dc_mot
speed_1 = 0.4
speed_2 = speed_1

def move(c1, c2, t):
    motor_1.throttle = 
    motor_2.throttle = 
    time.sleep(t)
    motor_1.throttle = (
    motor_2.throttle = (
    time.sleep(0.1)

move(0, 0, 1)
for i in range(4):
    move(1, 1, 0.8)
    move(-1, 1, 0.61)

move(0, 0, 1)
```

Jacob Haip
haipjacob@gmail.com

Education:

MIT

B.S. in Electrical Engineering & Computer Science
2011 - 2015

Web programming

Python, JavaScript, Node.js, React, Docker, Kubernetes, AWS, D3.js, SQL, Jenkins

Application development

Qt, C/C++, Processing

Computer vision

OpenCV, Kinect

Microcontrollers

C/C++, Arduino, Particle Photon, Raspberry Pi

Electrical design

Circuit board design, soldering, oscilloscope

Mechanical design & prototyping

CAD, 3D printing, laser cutting, milling, lathe

Design

Sketch, InVision, Video Editing

Current Research Interest:

Programmable Spaces

Programmable Spaces

2017 - Present

Personal research at the intersection of ubiquitous computing & end-user programming.

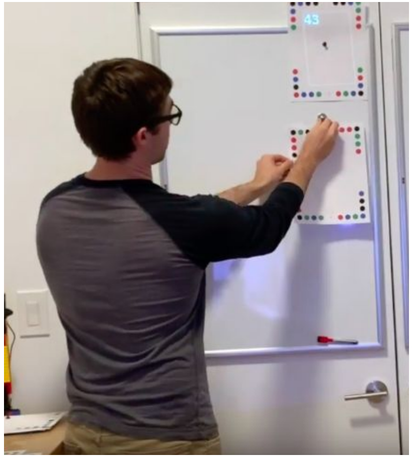
Goal:
Empower people who aren't professional programmers to make things with these properties:

- Gradual
- Room-Scale
- "Multiplayer" / Social

User test of the CityMatrix project
by 'Ryan' Yan Zhang (City Science, Media Lab)

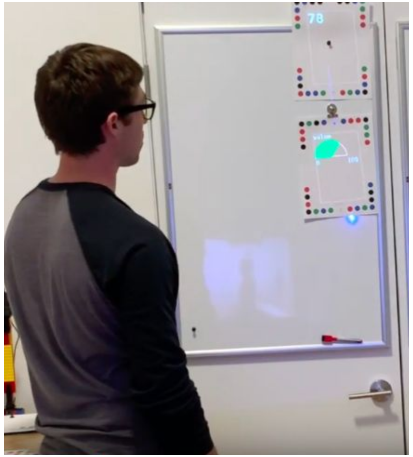
Why does a social and tangible system like CityMatrix have to be made using an antisocial and intangible tool?

Work so far has centered around the idea that pieces of code are mapped 1-to-1 with physical objects in the room.

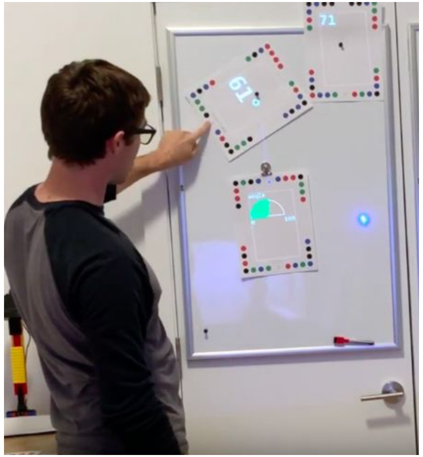


Pieces of paper have code written on them.

When the piece of paper is out in the room, the code written on that paper is run.



Cameras see the papers and projectors display the result of the code on the papers.



Papers can work together to construct bigger ideas.

The text editor used to edit code is also just another paper in the room.

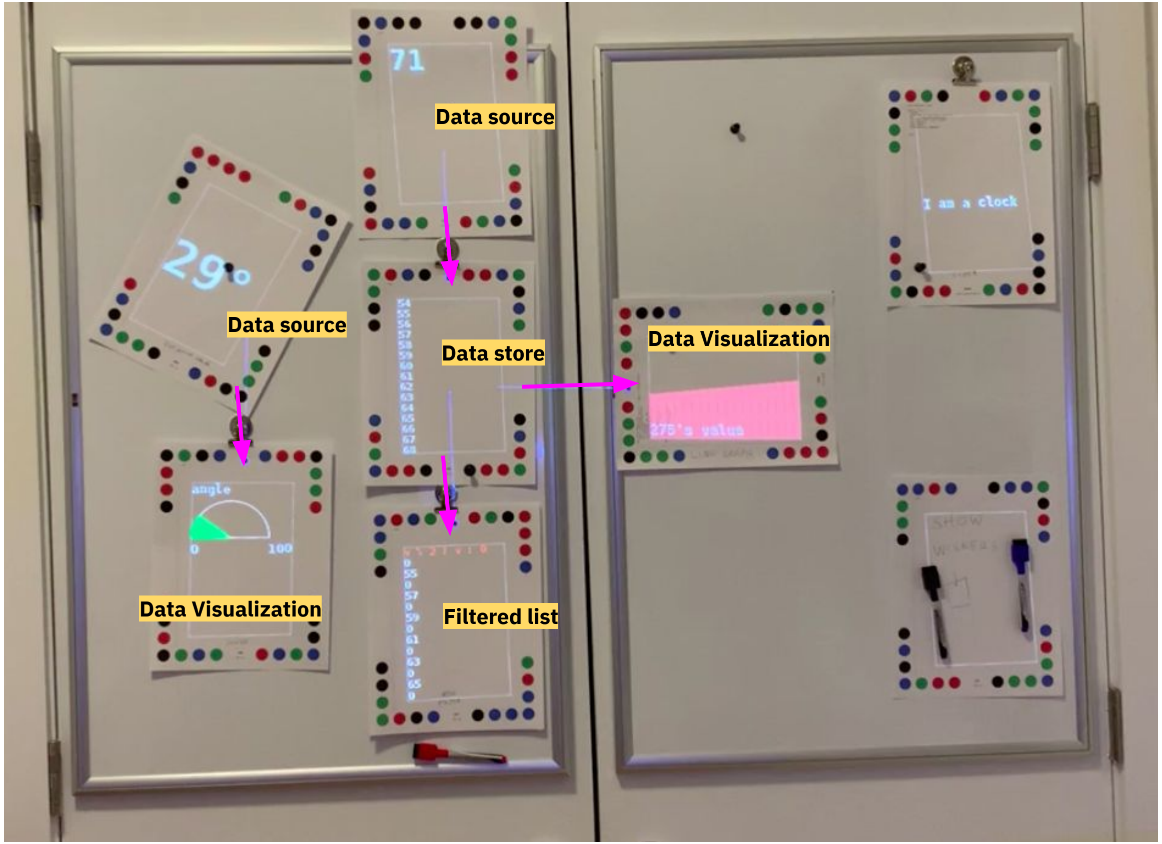
Dynamicland in Oakland, California has been an inspiration for both the goals and implementation of my research.

Demo: Spatial Programming

An exploration about how people can construct logic without textual code: by reordering and swapping programs that depend on the spatial location of other programs.

Data flows from a source and is collected, transformed, or visualized by reordering adjacent programs.

The code for each step is printed on the paper for people that want to understand the underlying logic.

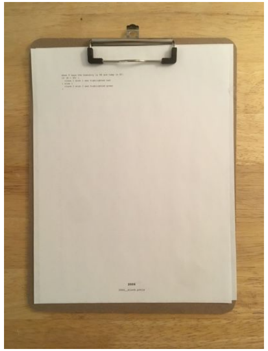


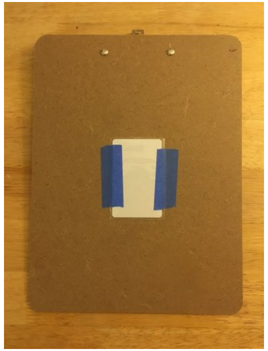
Demo: Microcontroller programming

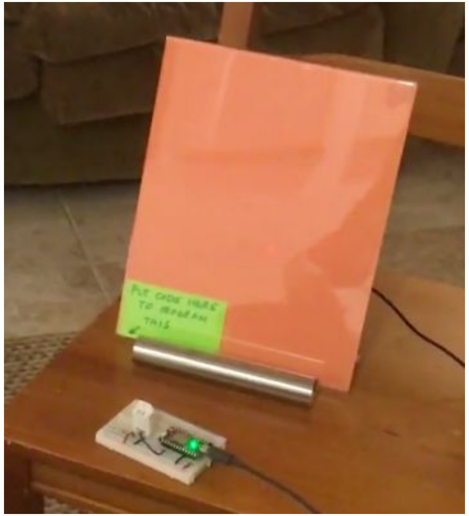
How might you reprogram objects in the room without switching back to a GUI computer?

Programs are represented by clipboards with a paper printout of the code they represent. RFID cards and sensors are used to detect individual clipboards.

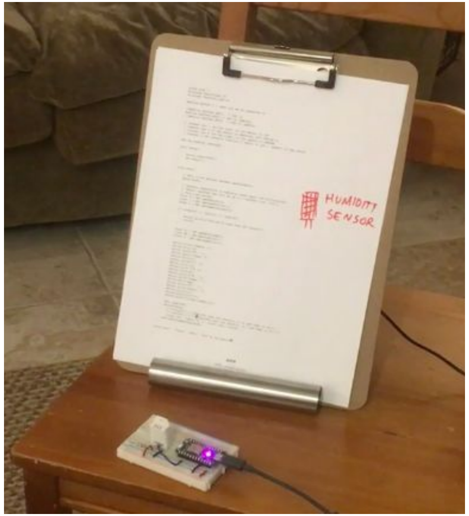
A microcontroller wired to a humidity sensor is an example reprogrammable object.







To program a microcontroller: programs are placed on a “code stand” that flashes code on the stand to the microcontroller.



Programming becomes the *physical action* of swapping clipboards.

The code stand and clipboard also serve as a visual reminder of what code is running on the microcontroller.

https://www.youtube.com/watch?v=xSSVTTso0LM

Demo: Augmented Restaurant Map

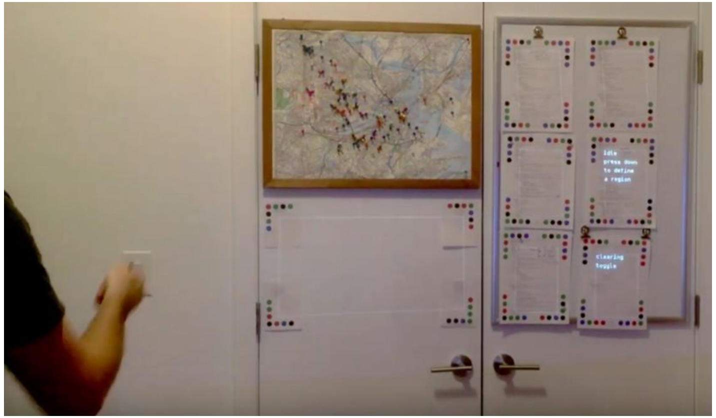
A map with pushpins of restaurants I've been to that shows the restaurant names when you point a laser pointer at a neighborhood.

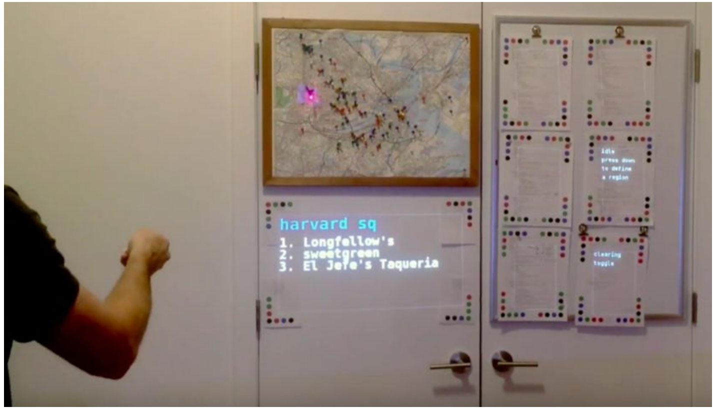
A few programs in the room allow anyone to outline regions in the room using a laser pointer, give the regions a name, and make the regions selectable.

Augmenting the map:

1. Outline neighborhoods and give them a name.
2. Make a new program with a little code like

```
when laser in region X and
     region X has name “harvard sq”:
draw_text(“1. Longfellow’s”)
```



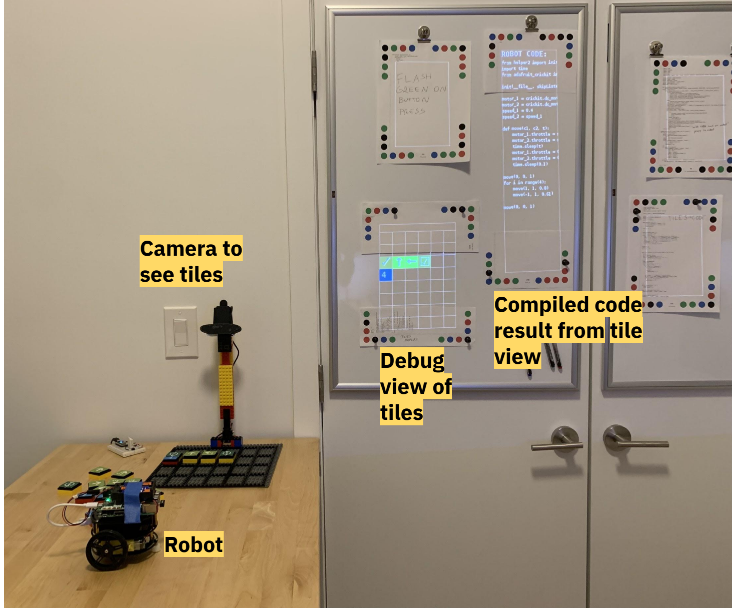


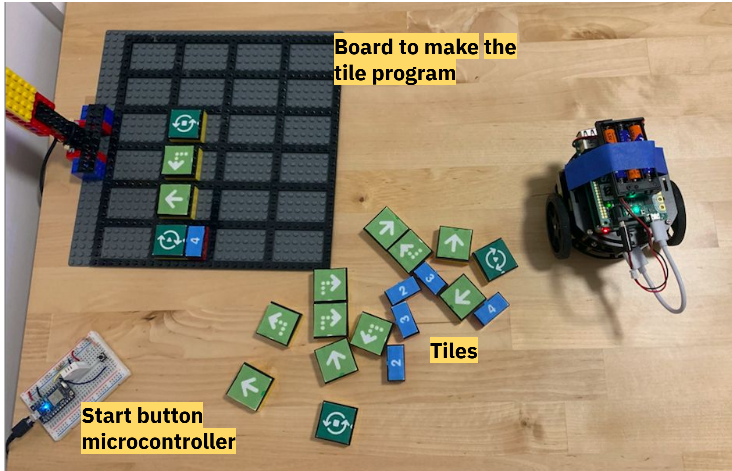
Demo: Programmable Robot

A remake of the Matatalab coding robot where robot movement is programmed using tiles like instructing the robot to move forward or turn.

The robot, tile sensing, start button, and debug visualizations are all objects in the room.

Demonstrates how the system supports non-textual programming and how many objects in the same space can work together.





https://www.youtube.com/watch?v=277migTZoKs

Professional Experience

Web Software Engineer @ Formlabs

2014 - Present

Remote monitoring and control for Formlabs 3D printers

Growing the online dashboard product from prototype to use by the majority of customers

- Full stack web development:
  Python, JavaScript, C++, Go, AWS, Docker, React, Jenkins, Kubernetes
- Integration with hardware and manufacturing
- User testing

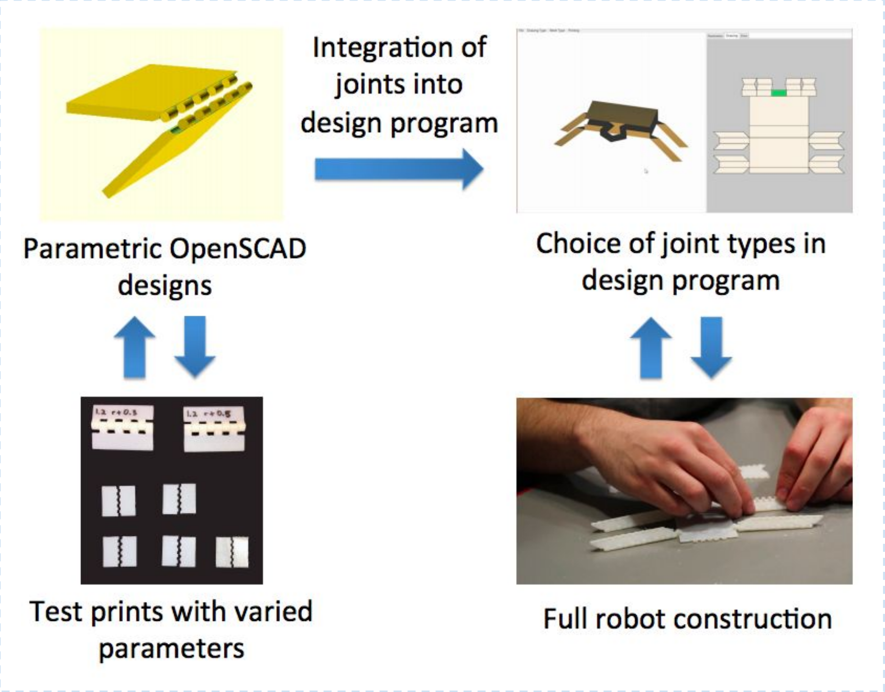
Undergraduate Research @ MIT

Computational Fabrication @ MIT CSAIL

Fall 2015 - Spring 2016

Working with Dr. Adriana Schulz on “Interactive Robogami” [IJRR ‘17]: a tool for composition-based design of ground robots that can be fabricated as flat sheets and then folded into 3D structures.

My work on the fabrication side of the project:



Parametric OpenSCAD designs

Integration of joints into design program

Choice of joint types in design program

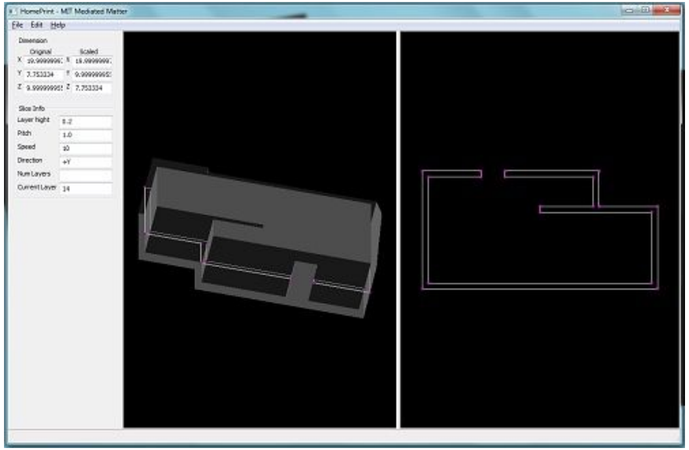
Test prints with varied parameters

Full robot construction

Mediated Matter @ MIT Media Lab

Fall 2012

Working with the late Steven Keating to explore the possibilities of building scale 3D printing using walls of insulation foam molds filled with concrete.



Dimension

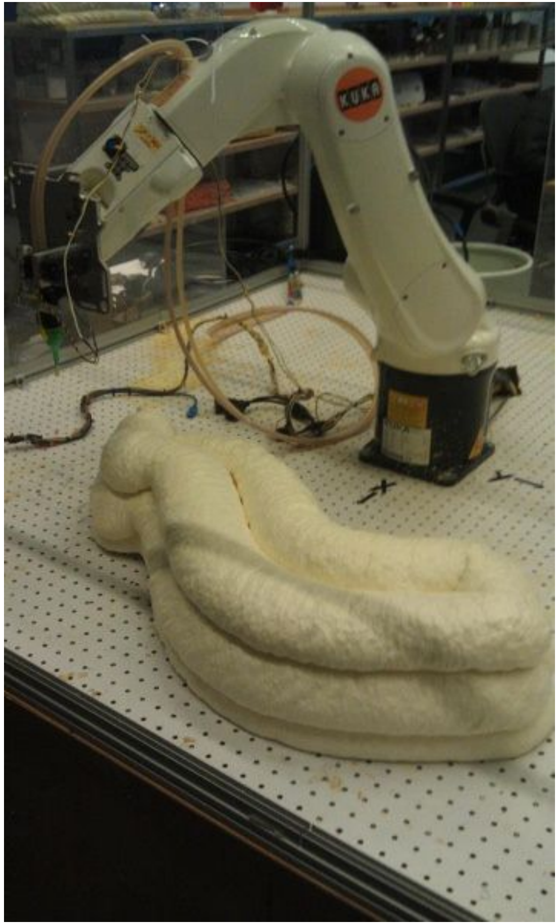
	Original		Scaled
X	19.9999996	X	19.9999999
Y	7.753334	Y	9.9999999
Z	9.9999999	Z	7.753334

Slice Info

Layer hight	0.2
Pitch	1.0
Speed	10
Direction	+Y
Num Layers	
Current Layer	14

I wrote a Python application to:

1. Import 3D model of a home
2. Configure printing parameters
3. Export GCODE to control a KUKA robot arm



Projects

Robot “Seeing Tool”

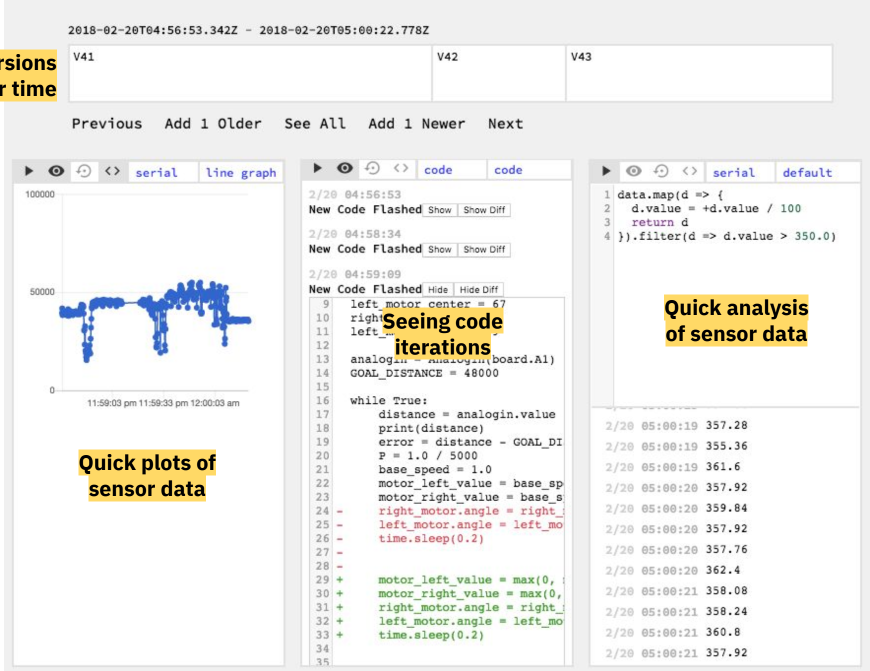
Spring 2018

A tool to help tell a more complete story of the making process. Code versions, serial log data, and notes are automatically saved and shown across time.

An exploration about how a software interface can support informal documentation and quick analysis while making electronic/robotic projects.

https://github.com/jhaip/streaming-queue-viz

https://www.youtube.com/watch?v=LqK2nMTE90g



Code versions over time

Quick plots of sensor data

Seeing code iterations

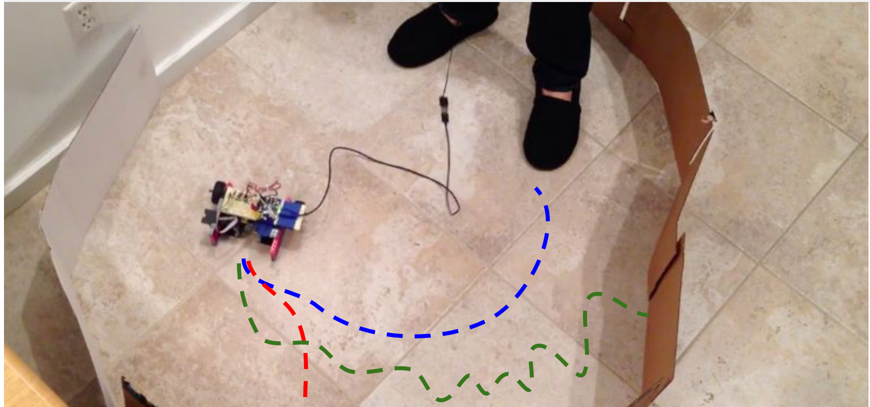
```
2/20 04:56:53
New Code Flashed
2/20 04:58:34
New Code Flashed
2/20 04:59:09
New Code Flashed
```

```
  left_motor_center = 67
  right
  left_

  analogin = AnalogIn(board.A1)
  GOAL_DISTANCE = 48000

  while True:
      distance = analogin.value
      print(distance)
      error = distance - GOAL_DI
      P = 1.0 / 5000
      base_speed = 1.0
      motor_left_value = base_sp
      motor_right_value = base_s
-     right_motor.angle = right_
-     left_motor.angle = left_mo
-     time.sleep(0.2)
-
-
+     motor_left_value = max(0,
+     motor_right_value = max(0,
+     right_motor.angle = right_
+     left_motor.angle = left_mo
+     time.sleep(0.2)
```

Quick analysis of sensor data

```
data.map(d => {
  d.value = +d.value / 100
  return d
}).filter(d => d.value > 350.0)
```

```
2/20 05:00:19 357.28
2/20 05:00:19 355.36
2/20 05:00:19 361.6
2/20 05:00:20 357.92
2/20 05:00:20 359.84
2/20 05:00:20 357.92
2/20 05:00:20 357.76
2/20 05:00:20 362.4
2/20 05:00:21 358.08
2/20 05:00:21 358.24
2/20 05:00:21 360.8
2/20 05:00:21 357.92
```

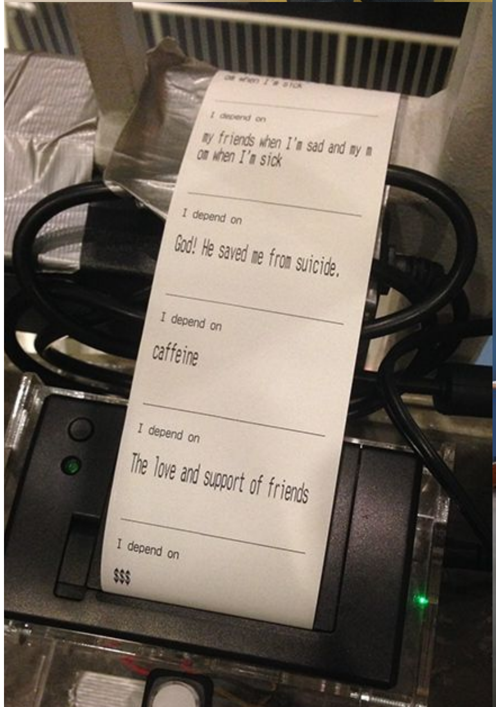


Printer Discourse

Spring 2014

Installation in the MIT Stata Center were anonymous responses to fill-in-the-blank prompts were collected and printed in real-time on an array of receipt printers. The strands of receipt paper grew longer as more people participated.

I made the text message collecting system using Twilio and the printers using 8 Raspberry Pi's and thermal printers.

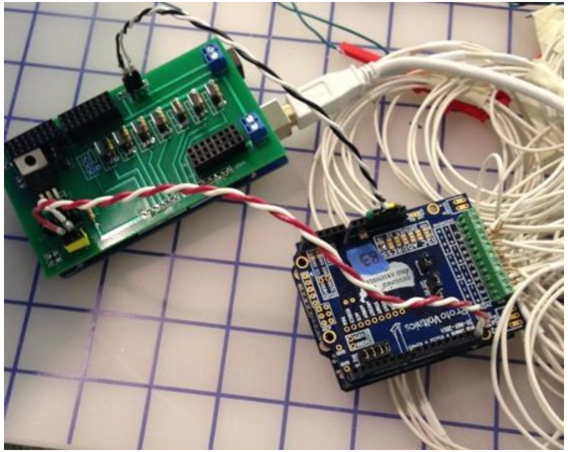


I depend on
my friends when I'm sad and my mom when I'm sick
I depend on
God! He saved me from suicide.
I depend on
caffeine
I depend on
The love and support of friends
I depend on
$$$

MIT 2.009 Product Design

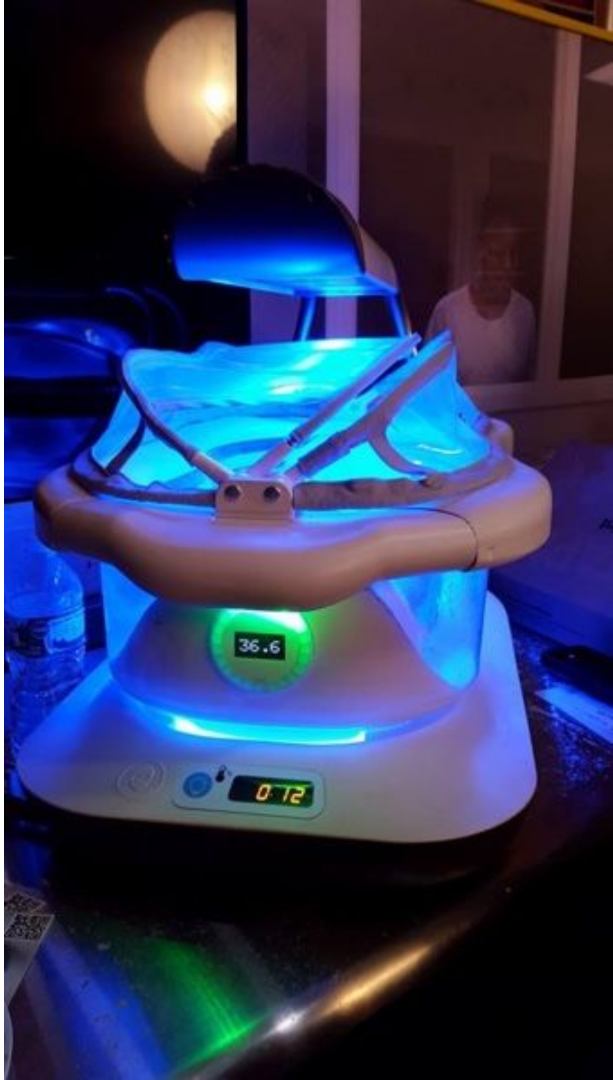
Fall 2014

Sunflower: a bassinet using phase changing material that keeps premature babies warm when electricity isn’t reliable.



Electrical engineer in a team of 24 Mechanical Engineering students.

- Designed PCB
- Purchased and tested components
- Wrote heater controls



Multi-touch Screens

2009 - 2011 (high school)

Built and wrote software for home-made multi-touch screen desks using IR light sources and P3Eye webcams modified to see infrared light.

Inspired by Jeff Han's work at NYU and Perspective Pixel.

My introduction to the world of Human–computer interaction research.