

Computation, Technology, and Culture (CTC)

RISD undergraduate concentration

Overview

- Very, very new RISD concentration
- Officially launching in spring 2017
- Similar to RISD's other concentrations (like a “minor” not a major)
- Mixes studio, seminar, and lecture courses

Overview

- Through the CTC concentration students gain an understanding of the ideas and techniques of writing in programming languages, while engaging with critical analysis, history, and theory concerning software systems, computational platforms, and associated technologies shaping society.
- Students hone their ability to write source code, author software, and program machines for making works of art and design.

Inter-Divisional structure

- CTC bridges Divisions and Departments. It is administered by EFS.
- CTC provides students with disciplinary investigations into new media, computation, and programming languages to augment the existing and ever-evolving digital methods inherent to their own major.

Overview

- The act of *coding* increasingly intersects with material-based practices. Code itself can also be understood as material – a substance with which we compose, that engages physical channels.
- Critical engagement with computation is key for work that involves getting machines to perform and fabricate what we envision.
- Gaining knowledge in this area, students are able to move beyond being “software users” constrained by proprietary software and become cultural producers who are artist-programmers.

Programming languages are mediums

- *Computation* is ubiquitous in everyday life and contemporary culture. It is pervasive in present-day technology and underpins future advances.
- Artists and designers actively shape the use and potential for computation within our society by expressing, communicating, and implementing their ideas by writing in programming languages.

Learning goals

- *Procedural literacy.* “The ability to read and write processes, to engage procedural representation and aesthetics, to understand the interplay between the culturally-embedded practices of human meaning-making and technically-mediated processes.” (Mateas)
- *Algorithmic thinking.* The ability to abstractly construct and understand algorithms independent from use within a programming language.

Learning goals

- Understanding of the fundamental ideas and techniques of writing in computer programming languages toward their utilization in a range of art and design fields...
- Preparedness for industries and practices where computer language competence is an increasingly important factor...
- Capacity for meaningful collaborations and communication in technology-oriented work that spans disciplines...
- Analytical skills in the survey of historical and contemporary works, artifacts, and schools of thought...

In completing the CTC concentration, students will be able to...

- Write source code, author software, and program hardware for making their own works of art and design...
- Discuss significant historical developments in computational technology...
- Compare and contrast theories surrounding the cultural reception and production of new and emerging media technologies...
- Demonstrate ability in critical discourse surrounding computation, technology, and culture through speaking, dialogue, writing, and artistic production...

Credits

- Credits earned by taking a mix of required and elective courses
- 15 credits (minimum), e.g. take five 3-credit courses
 - ◆ Required: introductory course (3 credits)
 - ◆ Required: advanced course w/ large-scale project (3 credits)
 - ◆ Electives: 9 credits (usually as 3 additional courses)

Declaring and completing

- Students will typically decide to become concentrators toward the end of their first-year or in their sophomore years as they register for courses, but it is also possible to begin earning credits in the junior year.
- Students are eligible to officially become concentrators when enrolling and then successfully passing a required introductory course.
- Students will generally take the required advanced course as the last course in the concentration sequence, after completing the minimum number of electives.

Pathways

- Multiple pathways to completion
- Potential to begin at first-year, sophomore, or junior levels

Example: starting first-year, elective possible in Wintersession

	Fall	Winter	Spring
1 st		Elective #1	
2 nd	Required #1 <i>Sophomore and above, Concentration declaration</i>		
3 rd	Elective #2		Elective #3
4 th			Required #2 <i>After electives complete</i>

Example: starting sophomore year, without using Wintersession

	Fall	Winter	Spring
2 nd	Required #1 <i>Sophomore and above, Concentration declaration</i>		
3 rd	Elective #1		Elective #2
4 th	Elective #3		Required #2 <i>After electives complete</i>

Example: starting sophomore year, using Wintersession alot

	Fall	Winter	Spring
2 nd		Elective #1	
3 rd	Required #1 <i>Sophomore and above, Concentration declaration</i>	Elective #2	
4 th		Elective #3	Required #2 <i>After electives complete</i>

Example: starting junior year, Wintersession necessary

	Fall	Winter	Spring
3 rd	Required #1 <i>Sophomore and above, Concentration declaration</i>	Elective #1	Elective #2
4 th	Elective #3		Required #2 <i>After electives complete</i>

The example pathways are not the only possibilities...

- One could take more electives before taking the introductory requirement.
- One could earn more than 3 credits in one semester by taking more than one eligible course or a 6-credit course (when offered).
- Other situations may arise. The Concentration Coordinator will advise you on your course plan.

Courses

- Some CTC-eligible courses are already offered
- More courses to be announced, *watch for the catalog*
- Some courses are sponsored directly by CTC
- Some cross-registered courses are sponsored by Departments
- Some Brown University courses may be eligible

Some current CTC-sponsored courses

- Introduction to Computation e.g. introductory requirement
- Ambient Interfaces: Activated Objects e.g. elective
- Programming Sound: Performance Systems e.g. elective
- Spatial Audio: Envelopment & Immersion e.g. elective
- Argentina: High/Low Tech e.g. elective
- CTC: Research Studio e.g. advanced requirement

setup paper variable
 setup pen variable
 draw oval = (length 3 inch, height 1inch)
~~O1 = oval~~
 align O1 ~~to 1 inch from left margin~~
 draw oval (length 3 inch, height 1 inch)
~~O2 = oval~~
 align oval ~~(1 inch from top)~~
 draw circle (diameter 1 inch)
 align circle in the center of O1
~~draw circle(diameter 1 inch)~~
 align circle in the center of O2
 draw oval [length 3 inch, height 1 inch]
~~O3 = oval~~
 align O3 in the center between
~~O1 + O2~~
 draw circle (diameter 1 inch)
 align circle in the center of O3
 draw vertical line ($\frac{1}{2}$ inch)
 align line ~~to under O1~~
 1 inch from left margin
 line = L1
 copy ~~L1~~ $L1 \times 100$
 align incrementally $\frac{1}{2}$ inch between each line

produce 25 lines going across
 X axis to 1 inch from right
 margin. ~~(1 inch)~~
 produce 25 lines under above lines
 repeat x 2

IN A SINGLE LINE DRAW A PERFECT CIRCLE.

FROM THE POINT AT WHICH THE LINES TWO ENDS MEET
 TO ANOTHER POINT DIRECTLY TWO INCHES BEHIND, DRAW
 A VERTICAL LINE.
 PERFECTLY STRAIGHT

NOW DRAW A PERFECT HALF CIRCLE (IN ONE LINE) IN WHICH
 THE TWO END POINTS DO NOT CONNECT AND THE EXACT
 MIDPOINT OF THIS LINE MEETS THE THE BOTTOM POINT
 IN THE EARLIER DRAWN VERTICAL LINE.*
 *

THE NEXT LINE WILL BE A HALF-ELLIPSE, CUPPING
 THE FIRST, WITHOUT TOUCHING. (LIKE A BIG SPOON) THE END
 POINTS OF THIS LINE MEET THE BOTTOM EDGE OF THIS PAPER
 SHEET AND THE MIDPOINT WILL BE DIRECTLY ABOVE THE
 MIDPOINT OF THE PRIOR DRAWN "HALF CIRCLE" AND WILL ALSO
 INTERSECT WITH THE VERTICAL LINE.*
 *

THE FINAL LINE WILL BEGIN EXACTLY WHERE YOU LIFTED
 YOUR PEN FROM THE PAGE WHEN DRAWING THE HALF ELLIPSE
 (BIG SPOON) AND WILL MOVE DIRECTLY UPWARDS REACHING THE
 EXACT HEIGHT OF THE ~~center~~ CENTER OF THE "PERFECT CIRCLE"
 AND THEN
 END OF T

INTRODUCTION TO COMPUTATION #0

- 1 - Make an edge to edge line that is 1/4" below the shorter side of the paper.
- 2 - Make an edge to edge line that is 1/4" below the previous line.
- 3 - Make an edge to edge line that is 1/4" below the previous line .
- 4 - Make an edge to edge line that is 1/4" below the previous line .
- 5 - Make an edge to edge line that is 1/4" below the previous line .
- 6 - Make an edge to edge line that is 1/4" below the previous line .
- 7 - Make an edge to edge line that is 1/4" below the previous line .
- 8 - Make an edge to edge line that is 1/4" below the previous line .
- 9 - Make an edge to edge line that is 1/4" below the previous line .
- 10 - Make an edge to edge line that is 1/4" below the previous line .
- 11 - Make an edge to edge line that is not below the previous line .
- 12 - Make an edge to edge line that is 1/4" below the previous line .
- 13 - Make an edge to edge line that is 1/4" below the previous line .
- 14 - Make an edge to edge line that is 1/4" below the previous line .
- 15 - Make an edge to edge line that is 1/4" below the previous line .
- 16 - Make an edge to edge line that is 1/4" below the previous line .
- 17 - Make an edge to edge line that is 1/4" below the previous line .
- 18 - Make an edge to edge line that is 1/4" below the previous line .
- 19 - Make an edge to edge line that is 1/4" below the previous line .
- 20 - Make an edge to edge line that is 1/4" below the previous line .

```

var canvas, pen;
var circle, ground;
var light, shadow;
  
```

function setup()

```

  canvas= (width, height);
  pen = new WritingUtensil();
  shapeAlign (CENTER);
  circle= new circle(width/2, height/2, 1);
  ground = new plane([0, height*(2/3)], [width, height*(2/3)], [0, height], [width, height]);
  light = new lightSource ([width*(2/3), 0]);
  shadow = render(lightShadow);
  speculate();
}
  
```

function speculate()

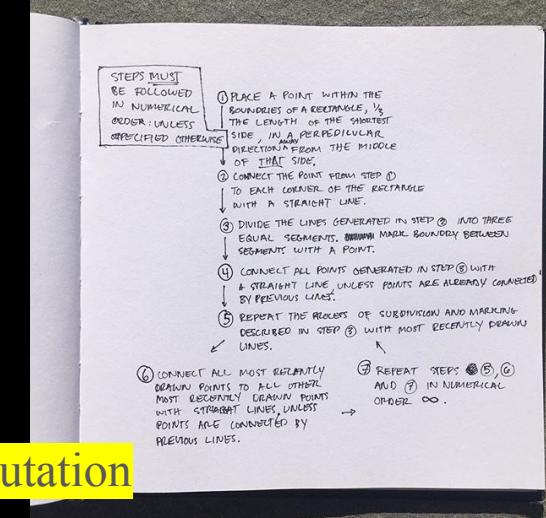
```

  if (drawing feels done){
    speculate (drawing= done)
    if (drawing= done){
      var kill = return( full understanding of light and space);
    }
  } else{
    notice(light);
    speculate( light = depth);
    speculate( shadow.presence = depth);
    speculate( circle.dimension = 3D);
    if( circle.dimension = 2D & circle.dimension = 3D){
      speculate( circle.dimension = 2D & circle.dimension = 3D);
    }
  }
}
  
```

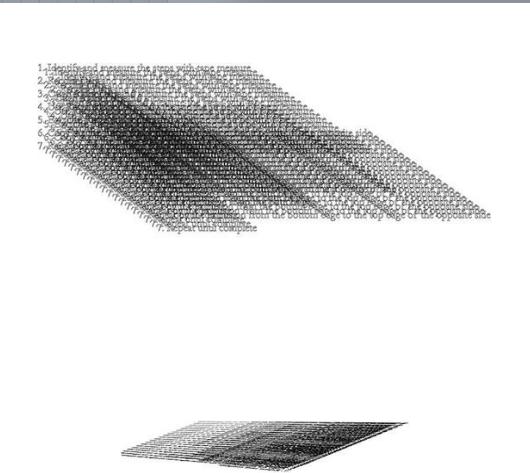
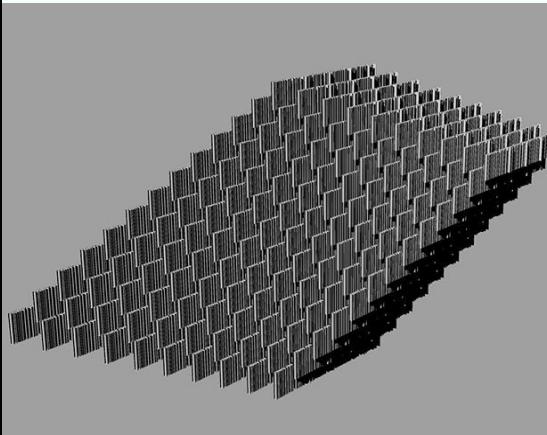
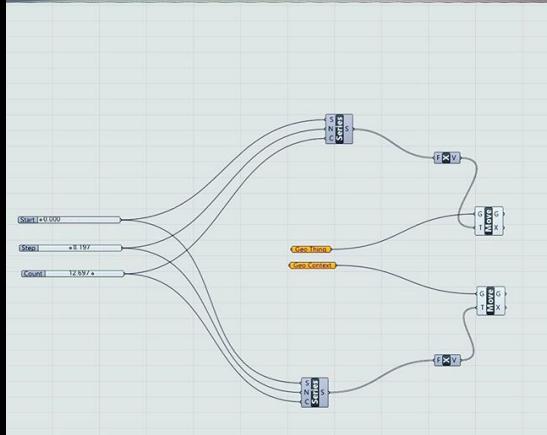
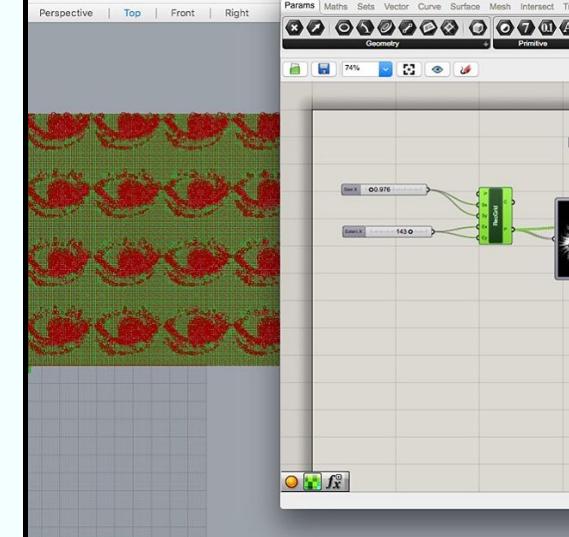
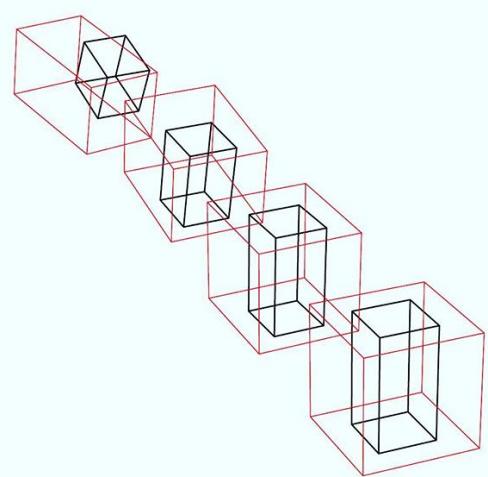
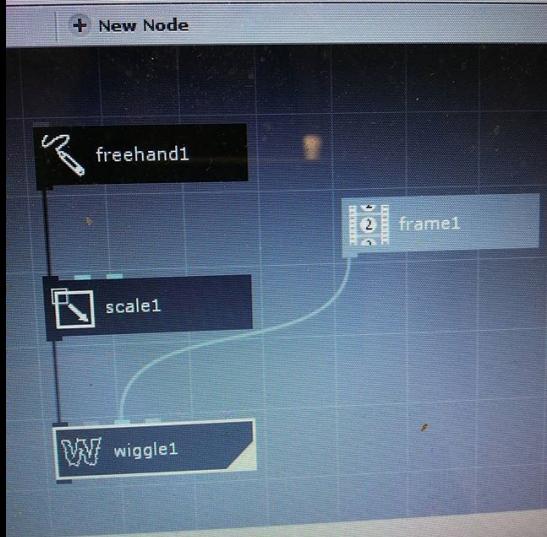
<hue:1, opacity:50%, +tint:75%, width: ∞ ,
height:0+ ∞ >

<hue:2, opacity:50%, +shade:75%, width: ∞ ,
height:0>

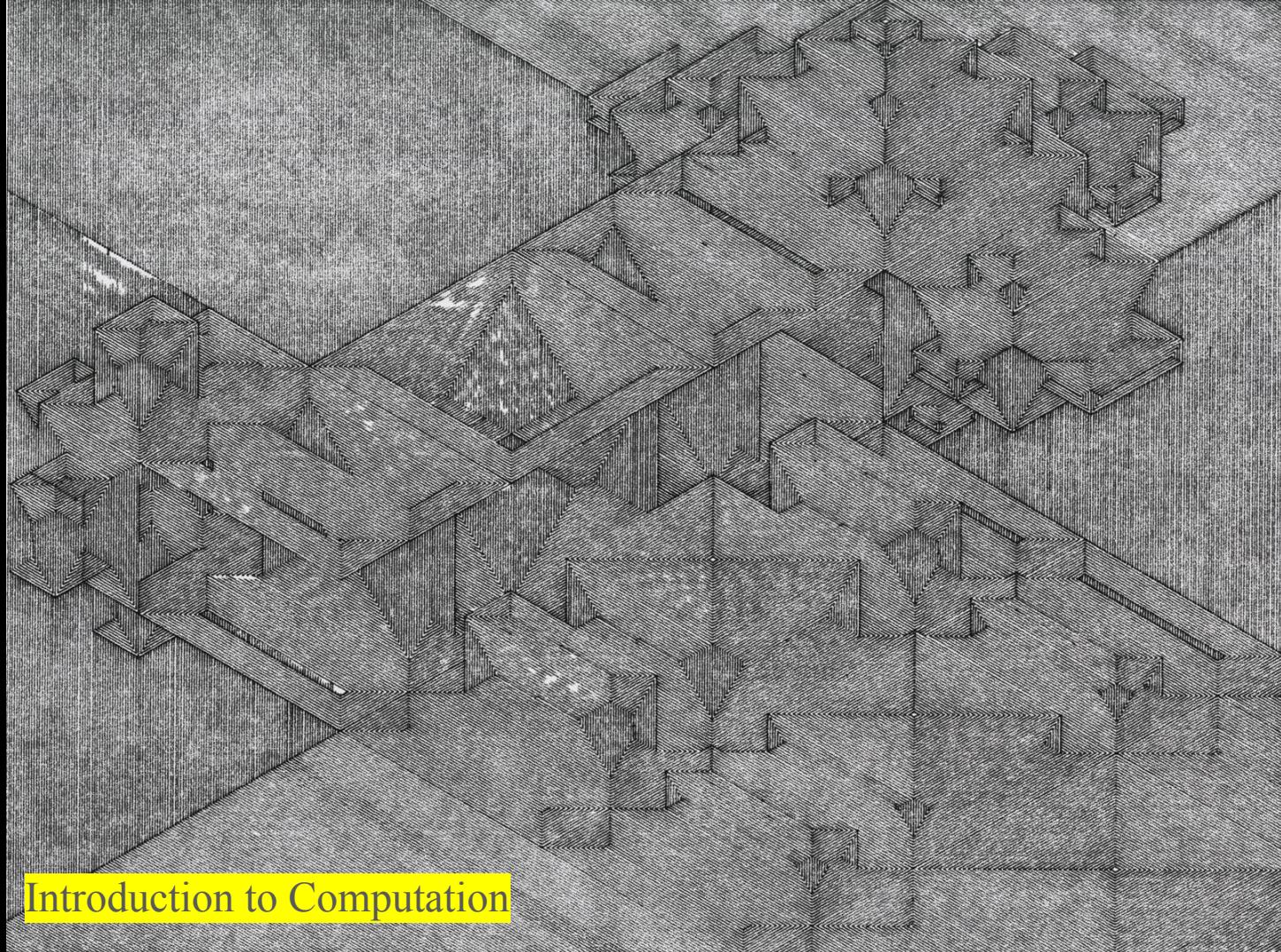
<hue:2, opacity:50%, +shade:25%, width: ∞ ,
height:0- ∞ >



Introduction to Computation



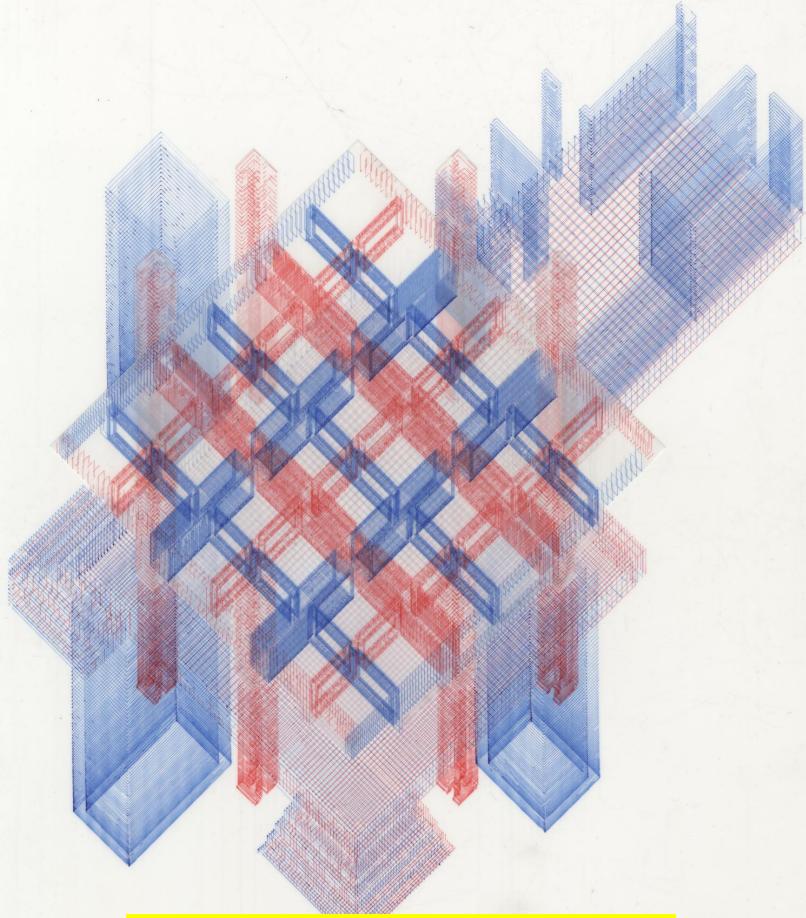
Introduction to Computation



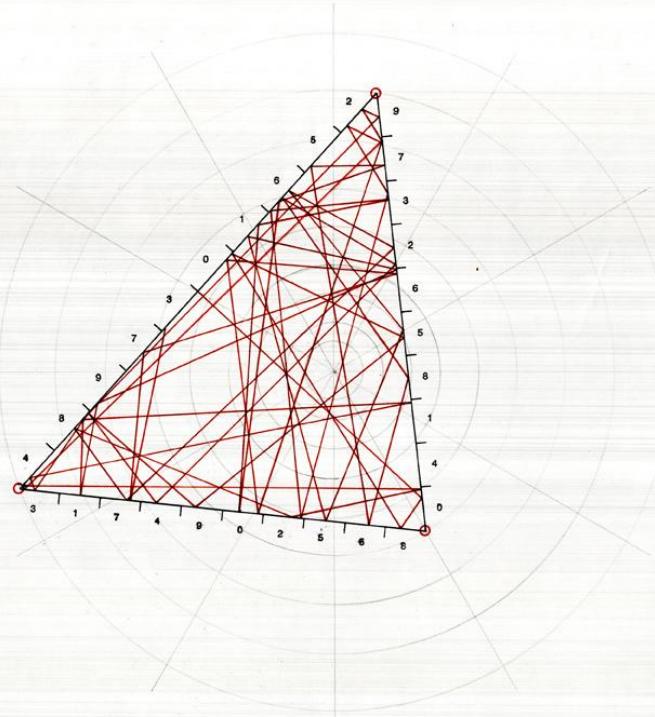
Introduction to Computation



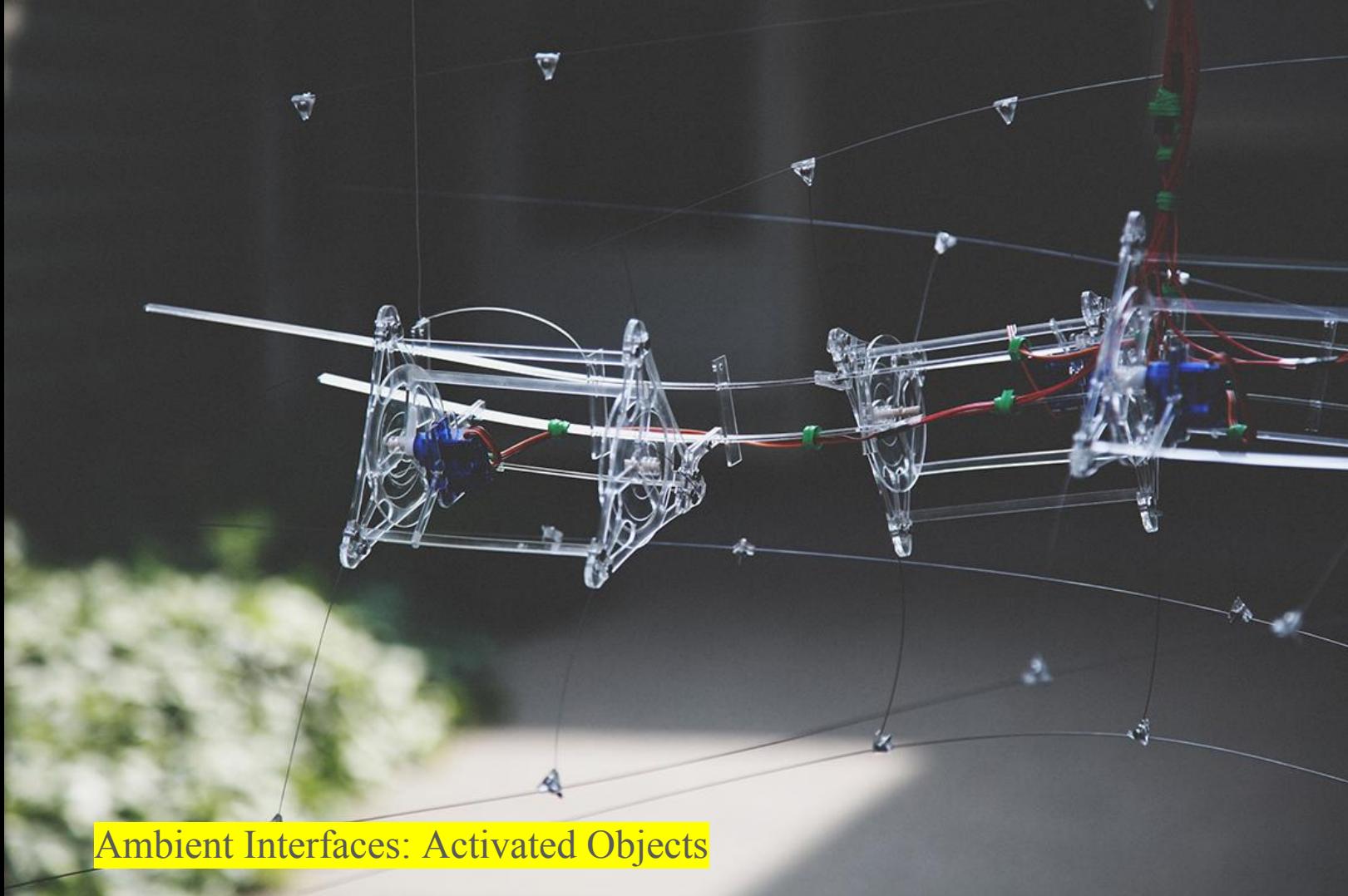
Introduction to Computation



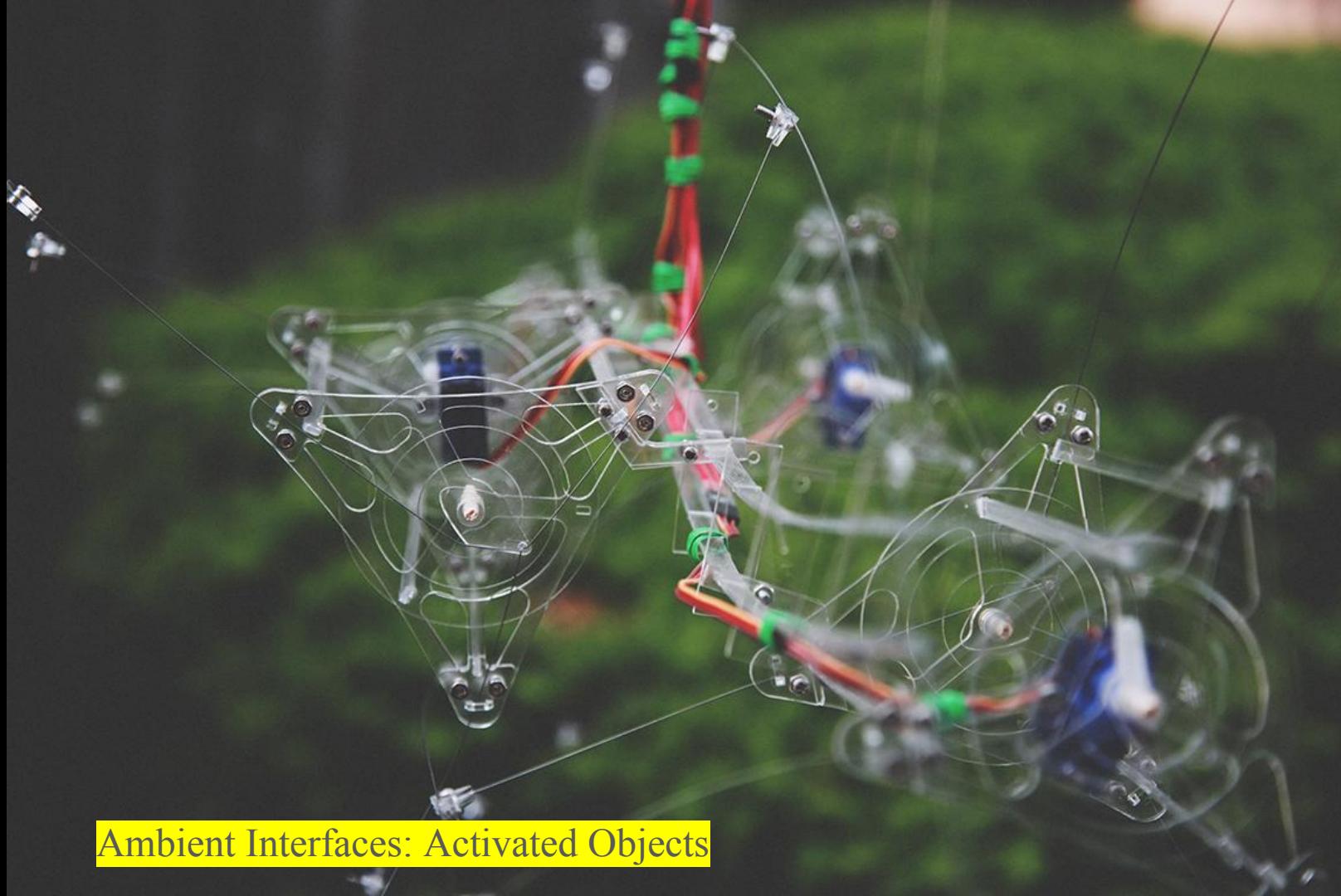
Introduction to Computation



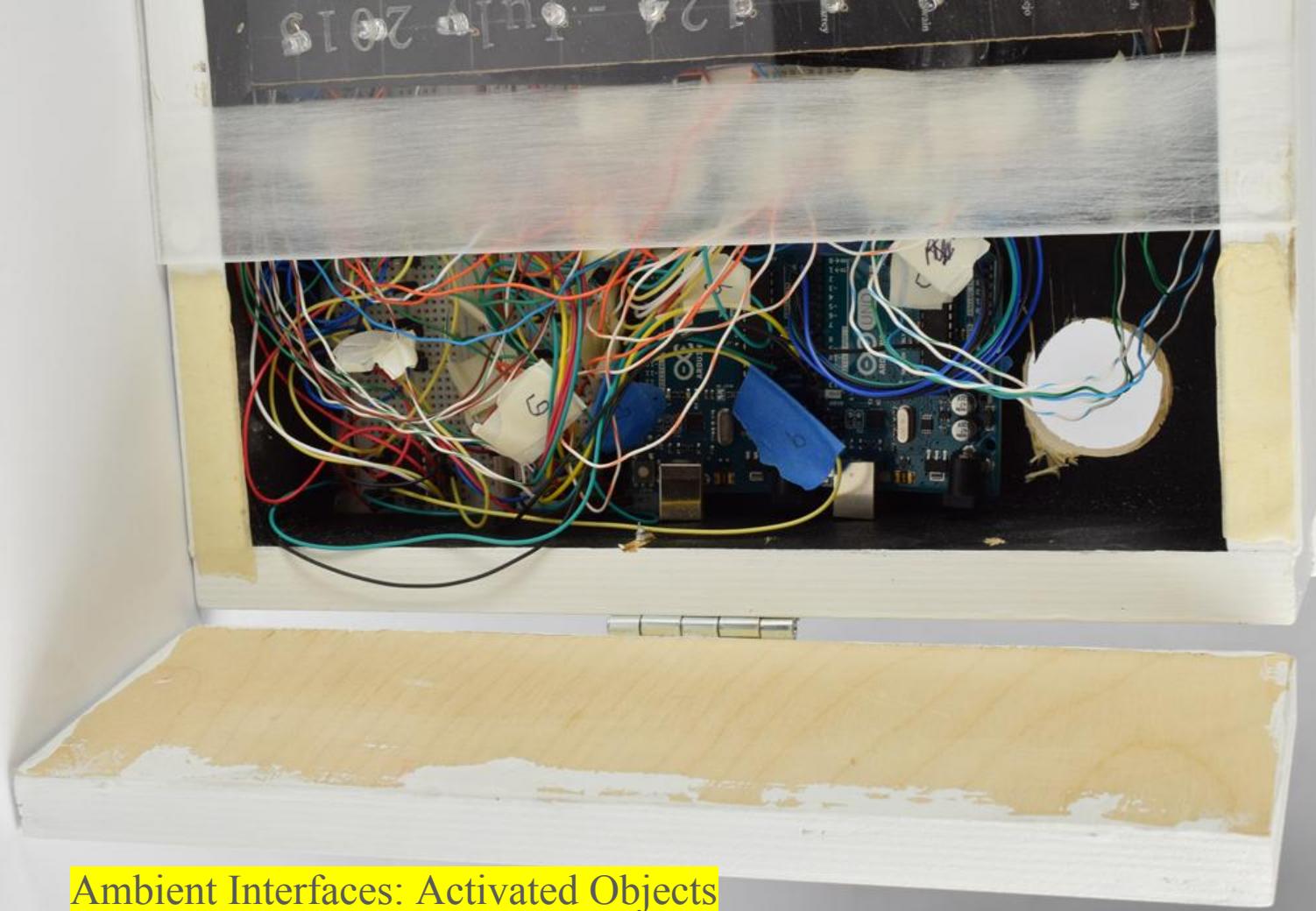
0805984268
6576013623
2734028726
6516918143
8927006773



Ambient Interfaces: Activated Objects



Ambient Interfaces: Activated Objects



Ambient Interfaces: Activated Objects



Ambient Interfaces: Activated Objects



Ambient Interfaces: Activated Objects



Spatial Audio: Envelopment & Immersion



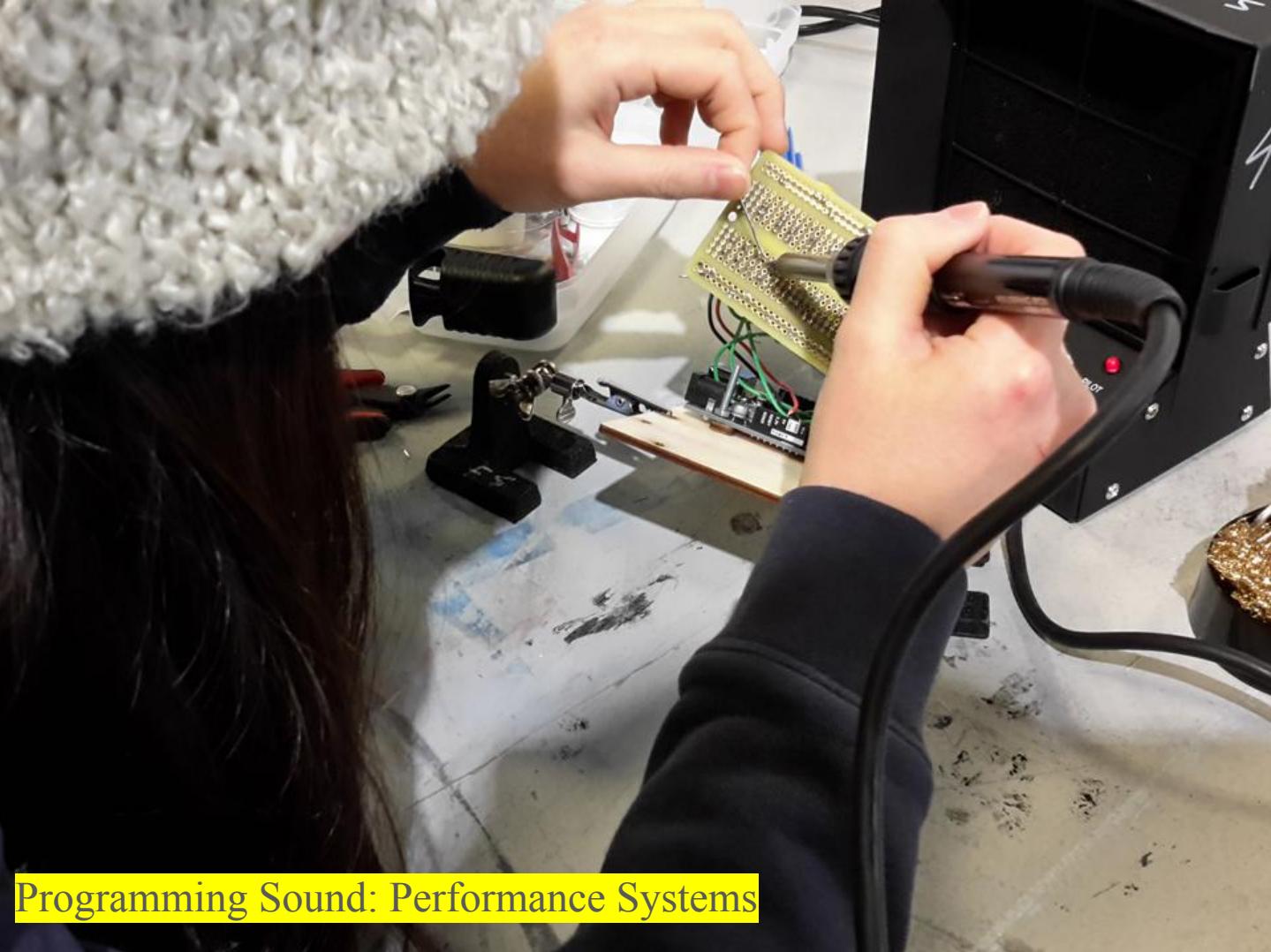
Spatial Audio: Envelopment & Immersion



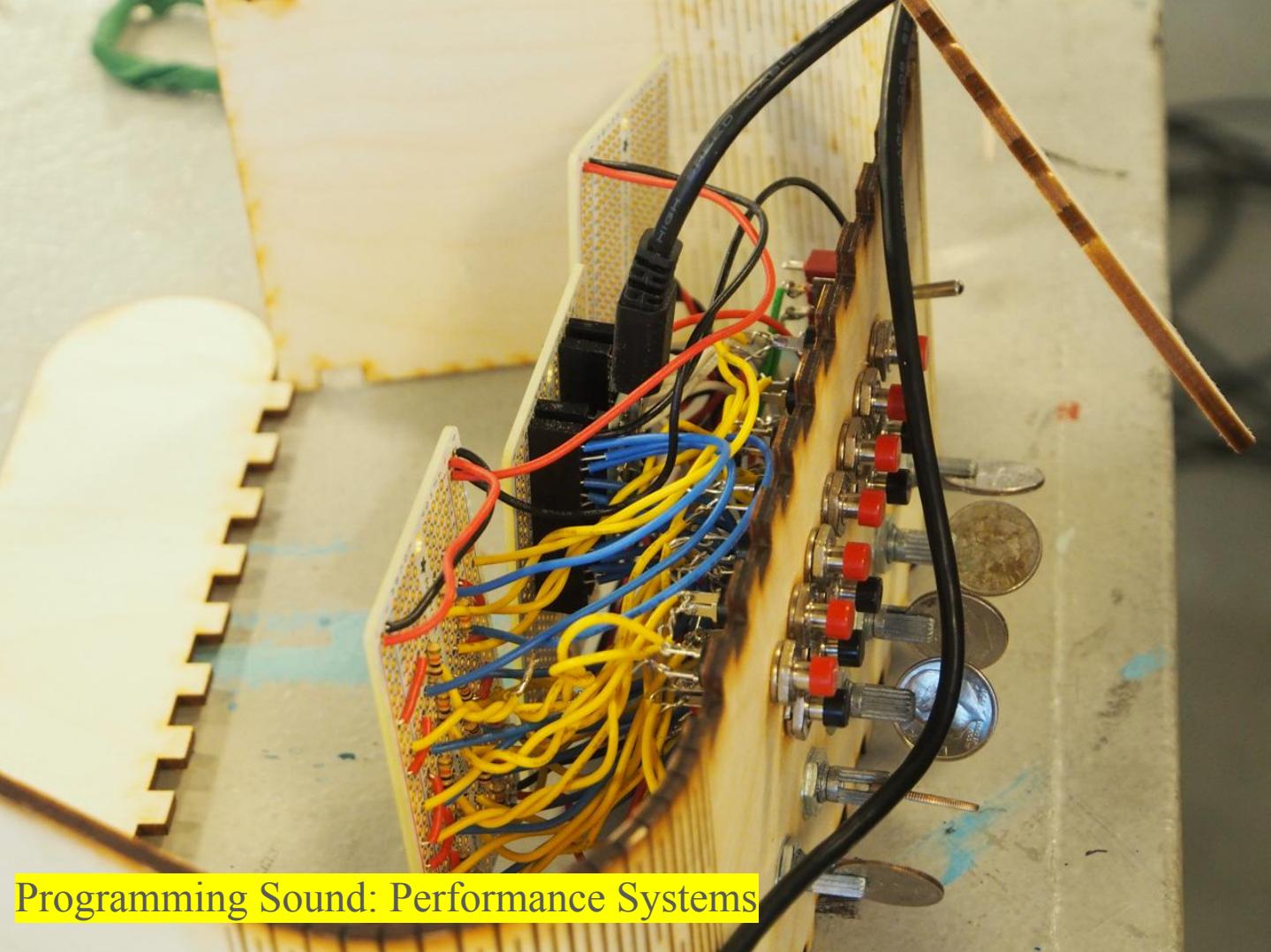
Spatial Audio: Envelopment & Immersion



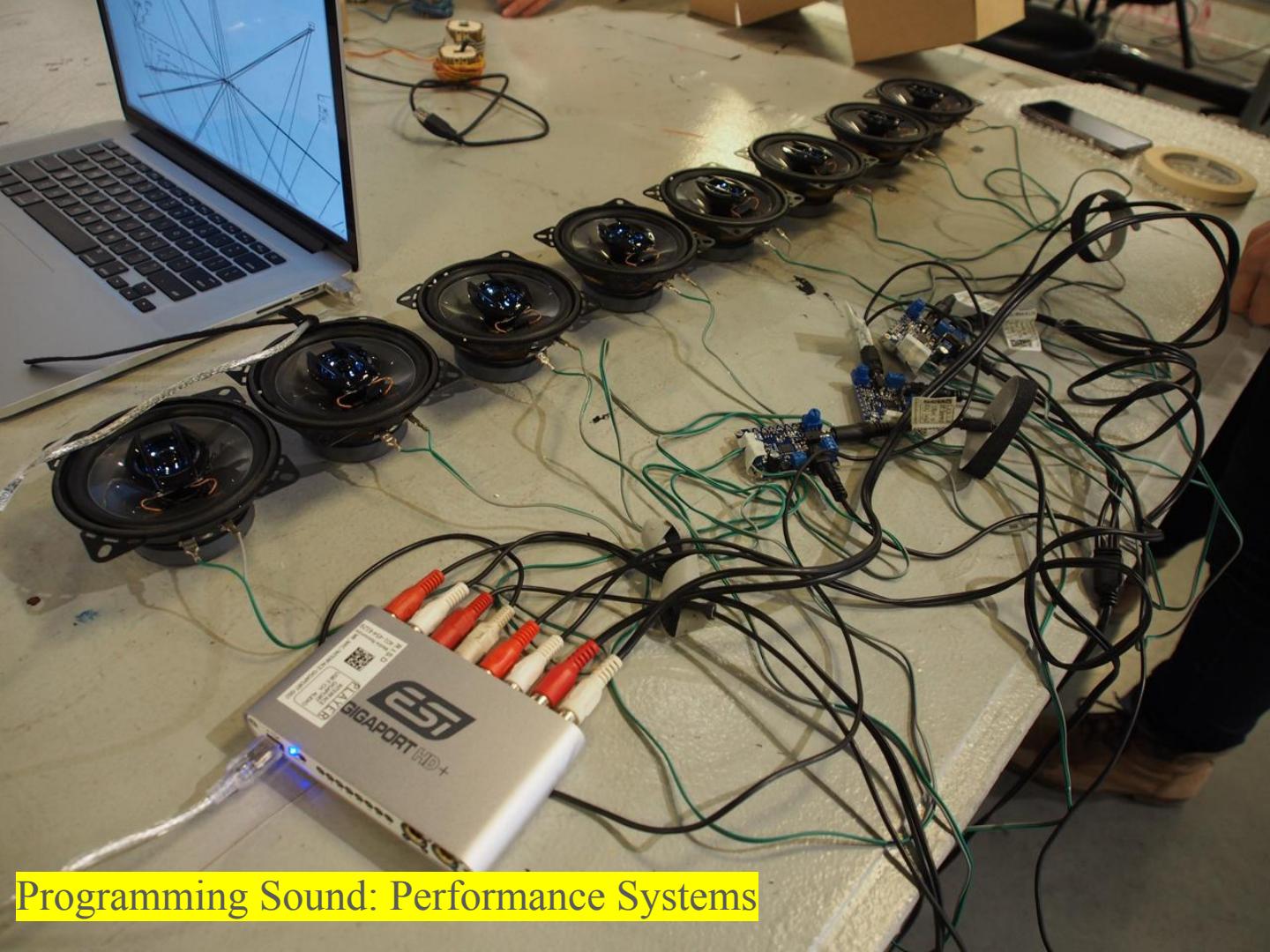
Spatial Audio: Envelopment & Immersion



Programming Sound: Performance Systems



Programming Sound: Performance Systems



Programming Sound: Performance Systems



Programming Sound: Performance Systems



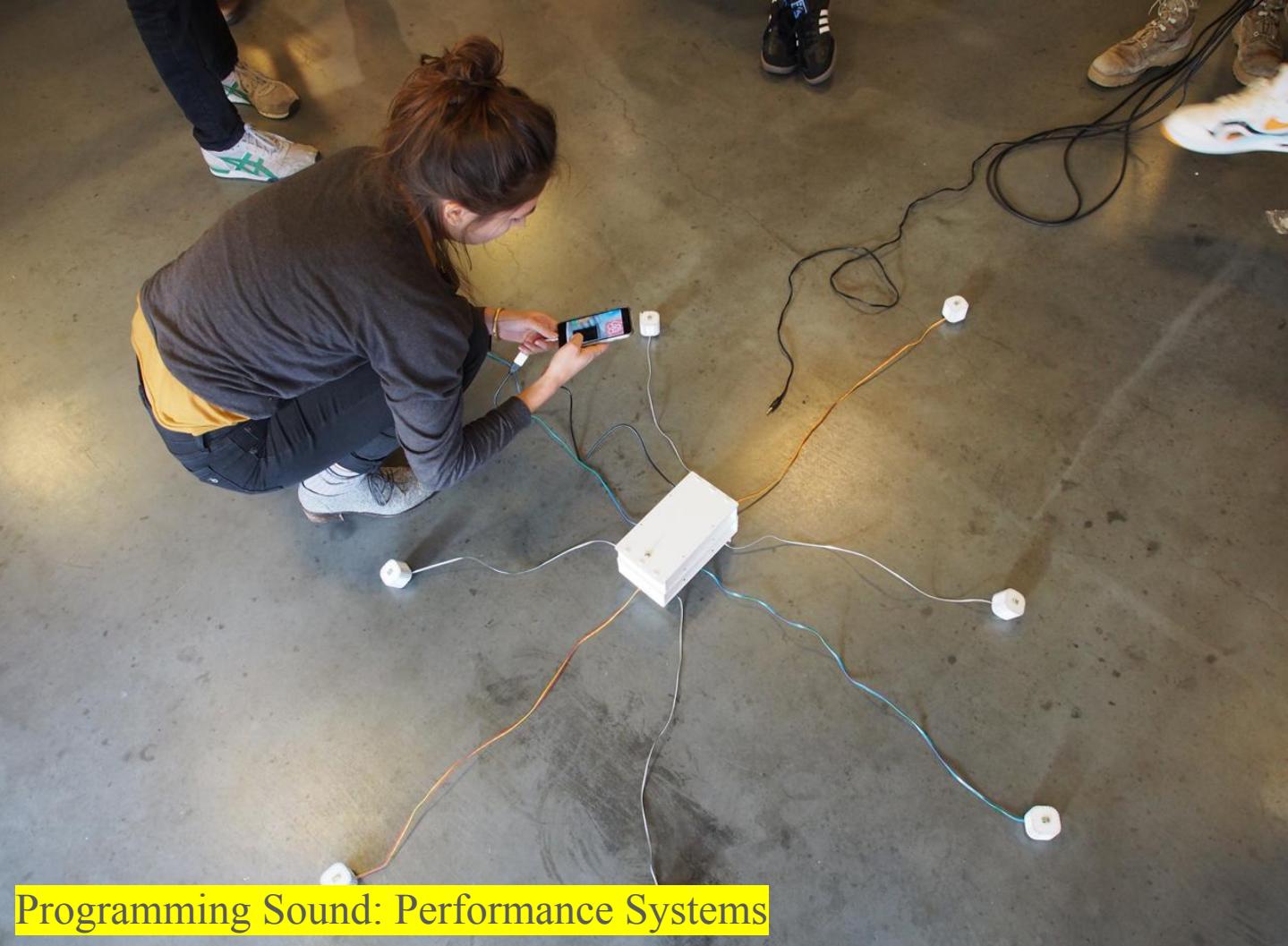
Programming Sound: Performance Systems



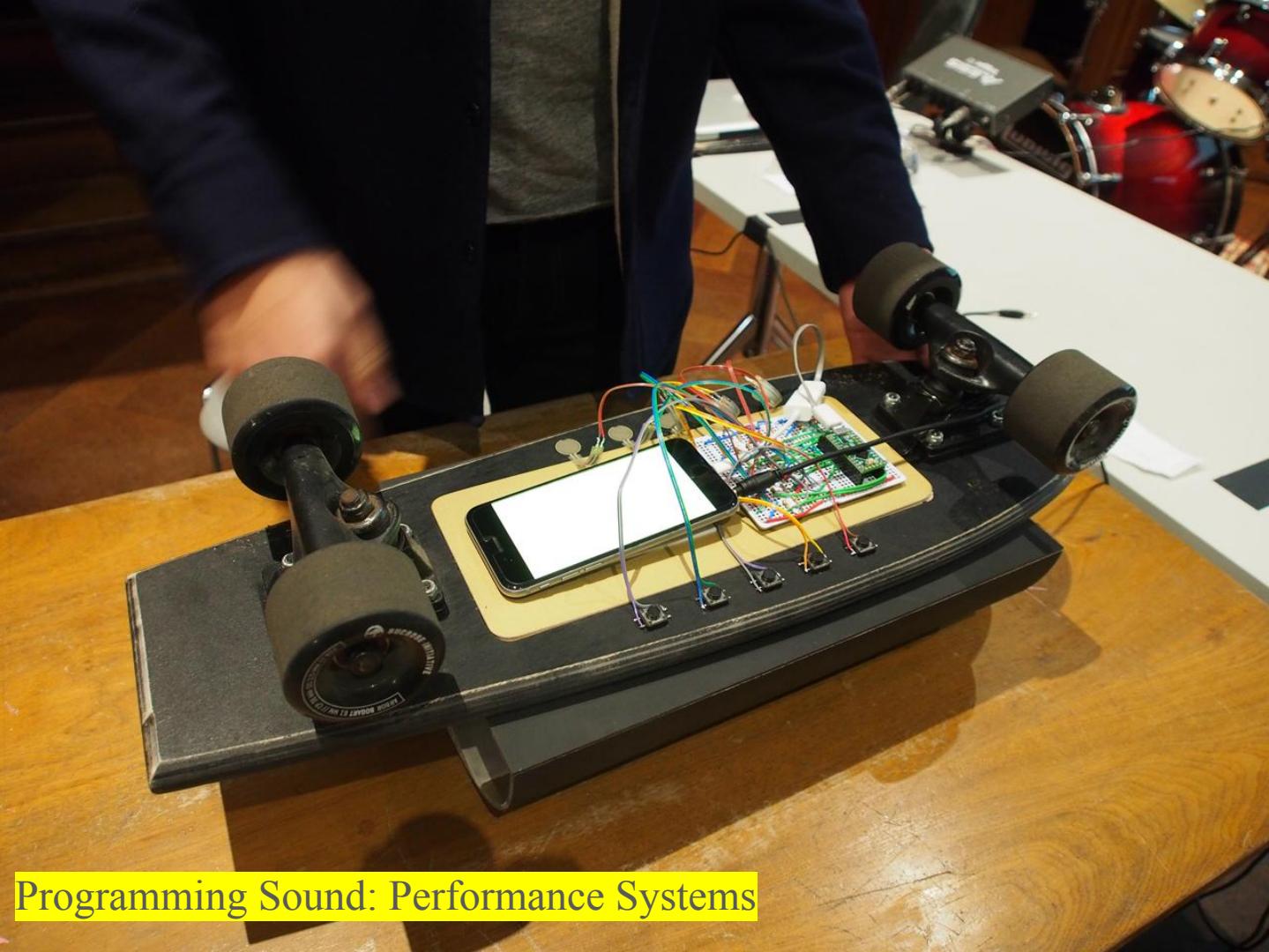
Programming Sound: Performance Systems



Programming Sound: Performance Systems



Programming Sound: Performance Systems



Programming Sound: Performance Systems

If interested in becoming a CTC Concentrator...

Contact CTC Coordinator Shawn Greenlee by e-mail: sgreenle@risd.edu

You will be put on an e-mail list that will be used to announce news about:

- Future information sessions
- CTC course offerings and events
- Further concentration details as they are published