



**Ge Wang**  
[gewang@cs.princeton.edu](mailto:gewang@cs.princeton.edu)

---

**Letter of Interest**  
(Also Research and Teaching Statements)

**Curriculum Vitae**

**Research**

ChucK Programming Language  
The Audicle : Audio Programming “Environ/mentality”  
On-the-fly Programming  
TAPESTREA : Modeling Sound Scenes By Example  
Laptop Orchestra (Development and Pedagogy)  
sndpeek + rt\_lpc + rt\_pvc : Visualizing Sound  
Gigapop Ritual : Interactive Network Performance

**Teaching**

Dartmouth College Fall 2006 Graduate Seminar  
Princeton University Fall 2006 : *Composing for Laptop Orchestra*  
Princeton University 2005 : *Princeton Laptop Orchestra, Year 1*  
Teaching Assistant : Princeton + Duke Computer Science

**Music**

*On-the-fly Counterpoint* : Live coding Duet  
*Loom* : a musical tapestry for 8-channel tape  
*Non-specific Gamelan Taiko Fusion*  
*CliX* : Exploring new conducting techniques for PLOrk  
*ChucK ChucK Rocket* : Game-based ensemble performance  
*Crystallis* : Sound of wind and crystal cave via new interface  
*PLOrk Beat Science* : Duet exploring adventures in Flute and HyPLOrkussion

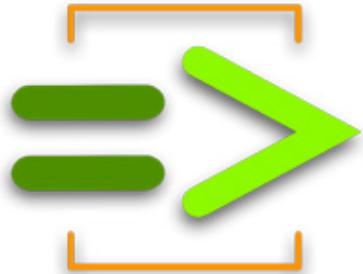
**Acknowledgements**



**Ge Wang : research**

## Research: ChucK Programming Language

A "Strongly-timed" and On-the-fly Audio Programming Language



*"A programming language that doesn't change the way you think is not worth learning."*

- Alan Perlis

**Date:** 2002 - present | **Released:** June 2004 for MacOS X, Linux, and Windows

**Role:** Chief designer and architect, co-creator, lead developer, and project lead.

**Collaborators:** Perry Cook (co-creator), Ari Lazier, Philip Davidson, Ajay Kapur, Adam Tindale, Ananya Misra, Spencer Salazar, Rebecca Fiebrink

### Publications:

(also see related articles under Audicle and On-the-fly Programming)

Wang, G., P. R. Cook. 2003. "ChucK: A Concurrent, On-the-fly Audio Programming Language". *In Proceedings of the International Computer Music Conference*. Singapore, October 2003.

Wang, G., P. R. Cook. 2004. "ChucK: A Programming Language for On-the-fly, Real-time Audio Synthesis and Multimedia." *In Proceedings of ACM Multimedia*. New York City, October 2004.

Wang, G., A. Misra, A. Kapur, and P. R. Cook. 2005. "Yeah ChucK It! => Dynamic Controllable Interface Mapping." *In Proceedings of the International Conference on New Interfaces for Musical Expression*. Vancouver, June 2005.

Wang, G., P. R. Cook, and A. Misra. 2005. "Designing and Implementing the ChucK Programming Language." *In Proceedings of the International Computer Music Conference*. Barcelona, September 2005.

### Awards:

2003 ICMA Best Presentation Award

2004 ACM Multimedia Best Open Source Software Award

### URL:

<http://chuck.cs.princeton.edu/>

## Research: **ChucK Programming Language** (*continued*)

---

A "Strongly-timed" and On-the-fly Audio Programming Language

```
SINOSC col => dac;  
  
// frequencies - chuck chuck chuck!  
[1209.0, 1336.0, 1477.0] @=> float col  
[697.0, 770.0, 852.0, 941.0] @=> float rows[]  
  
// let chuckian time advance  
25::ms => now;
```

### Description:

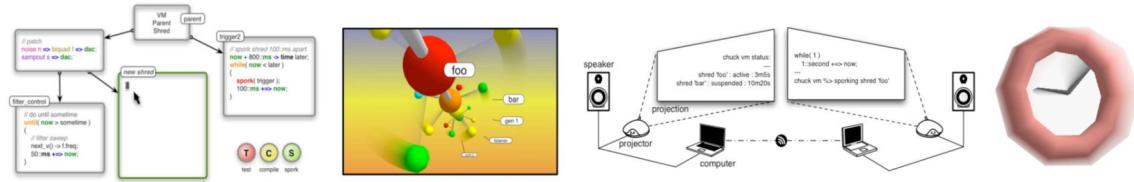
ChucK is a new (and developing) audio programming language for real-time synthesis, composition, and performance - fully supported on MacOS X, Windows, and Linux. ChucK presents a new ***time-based, concurrent programming model*** that's highly precise and expressive (we call this *strongly-timed*), as well as dynamic control rates, and the ability to add and modify code on-the-fly. ChucK supports MIDI, OSC, HID devices and multi-channel audio. Furthermore, the language is designed to ***favor readability and flexibility*** over raw performance. It's fun and easy to learn, and offers composers, researchers, and performers a powerful programming tool for building and experimenting with complex audio synthesis programs, and real-time interactive control.

The clarity and readability of ChucK code has proven to be highly effective for teaching sound synthesis, programming, generative music-making, and crafting new pieces/instruments for computer-mediated performance. We have used ChucK as a primary pedagogical tool in the Princeton Laptop Orchestra from the beginning, and as the platform for creating more than 25 (and counting) performance pieces in the ensemble. Increasingly more institutions are adopting ChucK as part of their curriculum in music technology and computer science, they include Princeton, Dartmouth, Stanford, Virginia, Tulane, University of Victoria BC, Georgia Tech, Art Institute of Chicago, and others. The ChucK users community is rapidly expanding, with 500+ subscribers on the chuck mailing lists, and more on various other forums.

The developers community is also fiercely expanding. In addition to the Princeton-based ChucK crew, we have collaborators located in Atlanta, Victoria BC, London, New York, Germany, and the Netherlands. Currently, an overall direction I would like to take ChucK is to combine both audio synthesis and analysis (music information retrieval) in the same precise, high-level real-time language platform. The possibilities for new ways to program sound and music are truly numerous and we shall continue to explore them.

## Research: The Audicle

An On-the-fly Audio Programming "Environ/mentality":  
for writing ChucK code and visualizing the audio programming process.



**Date:** 2004 - present | **Released:** Winter 2005 for MacOS X, Linux, and Windows

**Role:** Chief designer and architect, co-creator, project lead, lead programming

**Collaborators:** Perry Cook (co-creator), Philip Davidson (co-creator), Ananya Misra (co-creator), Spencer Salazar

### Publications:

(also see ChucK and PLOrk for related information)

Wang, G., P. R. Cook. 2004. "Audicle: A Context-sensitive, On-the-fly Audio Programming Environ/mentality." *In Proceedings of the International Computer Music Conference*. pp. 256-263. Miami, November 2004.

Wang, G., A. Misra, P. Davidson, and P. R. Cook. 2005. "Co-Audicle: A Collaborative Audio Programming Space." *In Proceedings of the International Computer Music Conference*. Barcelona, September 2005.

Wang, G., A. Misra, and P. R. Cook. 2006. "Building Collaborative interFaces in the Audicle." *In Proceedings of the International Conference on New Interfaces for Musical Expression*. Paris, June 2006.

### Awards:

2004 ICMA Best Presentation Award

### URL:

<http://audicle.cs.princeton.edu/>

### Applications:

- Visualizing programming process
- Live coding performance
- Teaching audio programming, debugging synthesis algorithms
- Platform for implementing new instruments/graphical interfaces for PLOrk

## Research: The Audicle (*continued*)

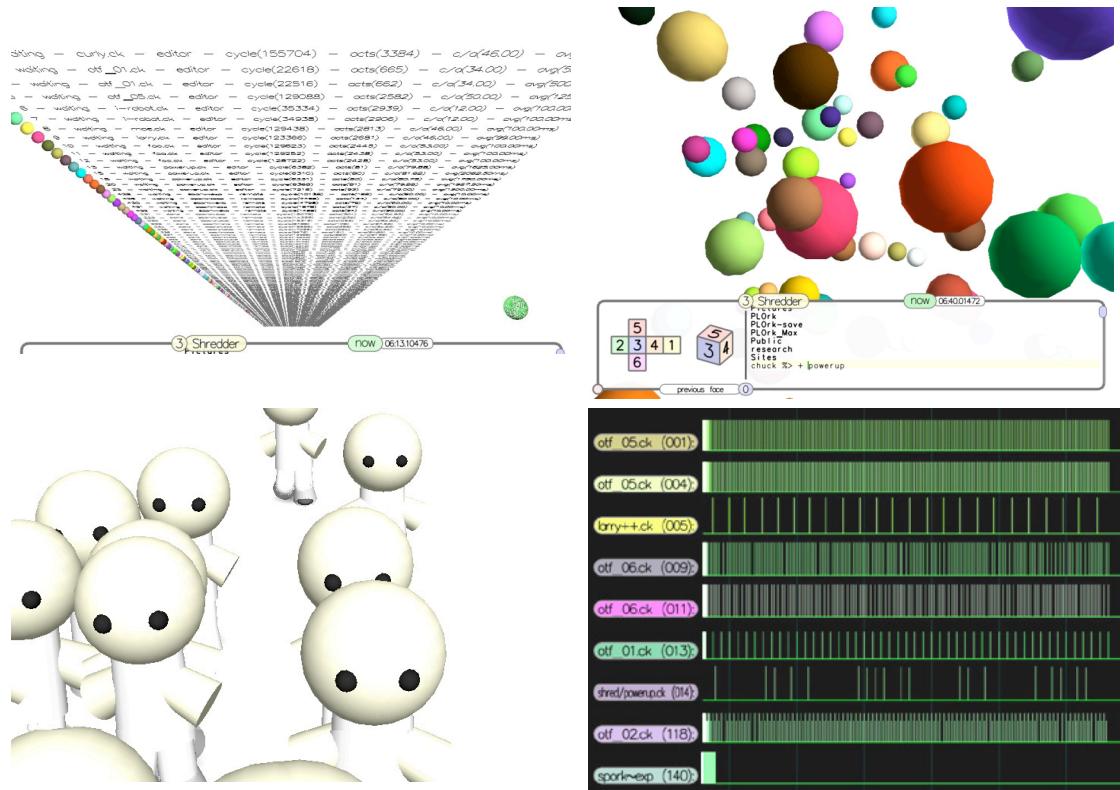
An On-the-fly Audio Programming "Environ/mentality":

For writing Chuck code and visualizing the audio programming process.

### Description:

Many software environments have been developed for computer music. Programming environments typically provide constructs to implement synthesis or musical algorithms, whereas runtime environments allow performers to exert parametric control over their programs onstage, in real-time. We present a potentially new type of audio programming environment that integrates the programmability of the development environment with elements of the runtime environment. The result, called the Audicle, is a “semi-duct-taped” intersection of a concurrent smart editor, compiler, virtual machine, and debugger, all running in the same address space, sharing data, and working together at runtime. We believe this augmentation has the potential to fundamentally enhance the way we write and visualize audio programs both offline and on-the-fly.

Our model of the Audicle is integrated with the ChucK programming language and inherits many of its fundamental properties, including: decoupling of data-flow and time, concurrency, and modularity for on-the-fly programming. The Audicle not only provides an amusing class of programming tools for real-time composition and performances, but also motivates a new type of on-the-fly programming aesthetic: one of visualizing the audio programming process. Below are screenshots of various “faces” of the Audicle:



## Research: **On-the-fly Programming** (also Live Coding)

*Using (Chuck) Code as an Expressive Musical Instrument*

**Date:** 2004 - present

**Role:** Investigator and developer, performer

**Collaborator:** Perry Cook

### **Publications:**

Wang, G., P. R. Cook. 2004. "On-the-fly Programming: Using Code as an Expressive Musical Instrument". In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Hamamatsu, Japan, June 2004.

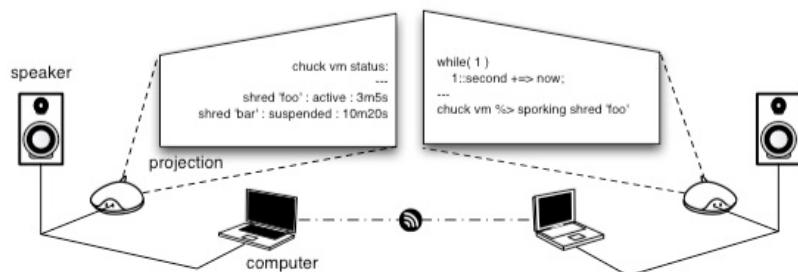
### **URL:**

<http://on-the-fly.cs.princeton.edu/>

### **Performances:** (*see music section for details*)

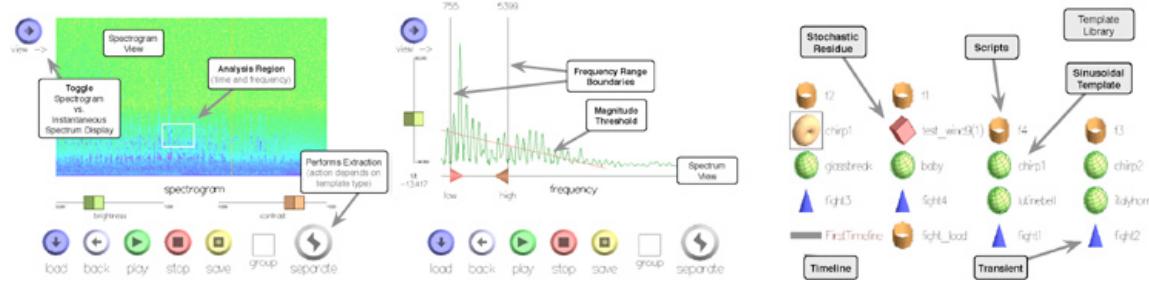
### **Description:**

On-the-fly programming (or live coding) is a technique and practice in which the programmer/performer/composer augments and modifies the program while it is running, without stopping or restarting, in order to assert expressive, programmable control for performance, composition, and experimentation at run-time. Because of the fundamental powers of programming languages, we believe the technical and aesthetic aspects of on-the-fly programming are worth exploring.



## Research: TAPESTREA (a.k.a. TAPS)

New Sound Design Paradigm and Software Environment, combines sinusoidal modeling, wavelet-based stochastic decomposition/reconstruction, and a new class of user interface for template-based sound design. (Also see Loom under Music section)



**Date:** 2004 – present | **Released:** August 2006 for MacOS X, Linux, and Windows

**Role:** Co-designer/implmenter of core system, lead architect of user interface, ChucK integration, and department of TAPS public relations

**Collaborators:** Ananya Misra (project lead), Perry Cook

### Publications:

Misra, A., P. R. Cook, and G. Wang. 2006. “TAPESTREA: Sound Scene Modeling by Example” (*Sketch*) *ACM SIGGRAPH*. Boston, August 2006.

Misra, A., P. R. Cook, and G. Wang. 2006. “A New Paradigm for Sound Design.” In *Proceedings of the International Conference on Digital Audio Effects*. Montreal, October 2006.

Misra, A., P. R. Cook, and G. Wang. 2006. “Musical Tapestry: Re-composing Natural Sounds.” In *Proceedings of the International Computer Music Conference*. New Orleans, November 2006.

### Award:

2006 ICMA/Swets and Zietlinger Distinguished Paper Award

### Composition:

*Loom (Etude II pour un enfant seul)*

by Ge Wang, Perry Cook, and Ananya Misra

for 8-channel tape, ICMC 2006 (see music section)

### URL:

<http://taps.cs.princeton.edu/>

## Research: PLOrk: Princeton Laptop Orchestra

---

*New (and completely insane) computer-mediated performance ensemble:*

*15 x humans, 15 x laptops, 15 x 6-channel hemispherical speakers.*

(Please also see teaching and music sections of portfolio for related involvement)

**Date:** 2005 - present

**Roles:** Teaching assistant (Fall 2005 - Spring 2006), co-director and instructor (Fall 2006), co-developer of core curriculum, designer and developer of new interfaces for performance and new techniques for conducting and synchronization, composer, performer.

**Collaborators:** Dan Trueman (founder and director), Perry Cook (founder and director), Scott Smallwood (fellow co-director / founding teaching assistant)

### Publications:

Trueman, D., P. R. Cook, S. Smallwood, and G. Wang. 2006. "PLOrk: Princeton Laptop Orchestra, Year 1." *In Proceedings of the International Computer Music Conference*. New Orleans, November 2006.

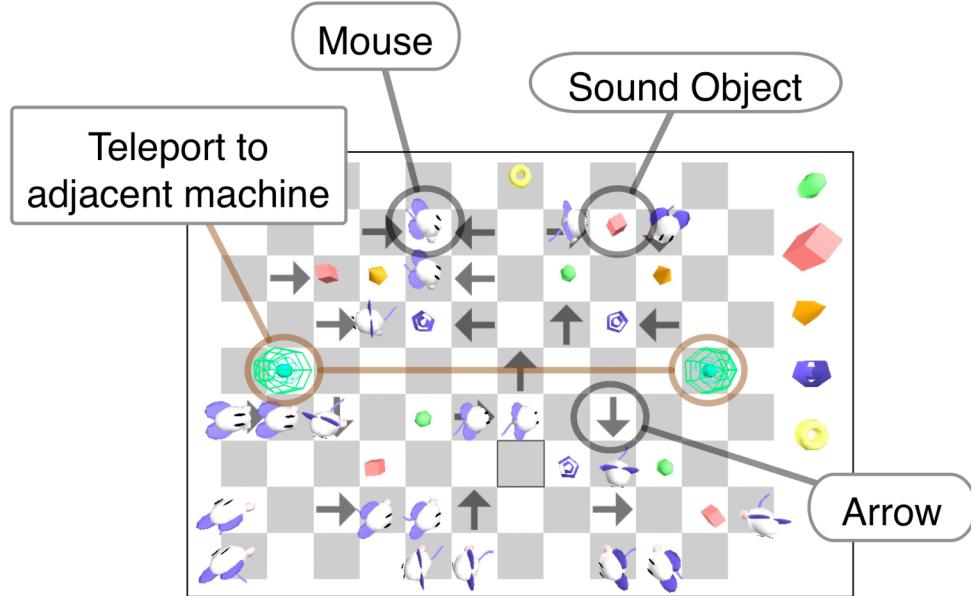
### URL:

<http://plork.cs.princeton.edu/>

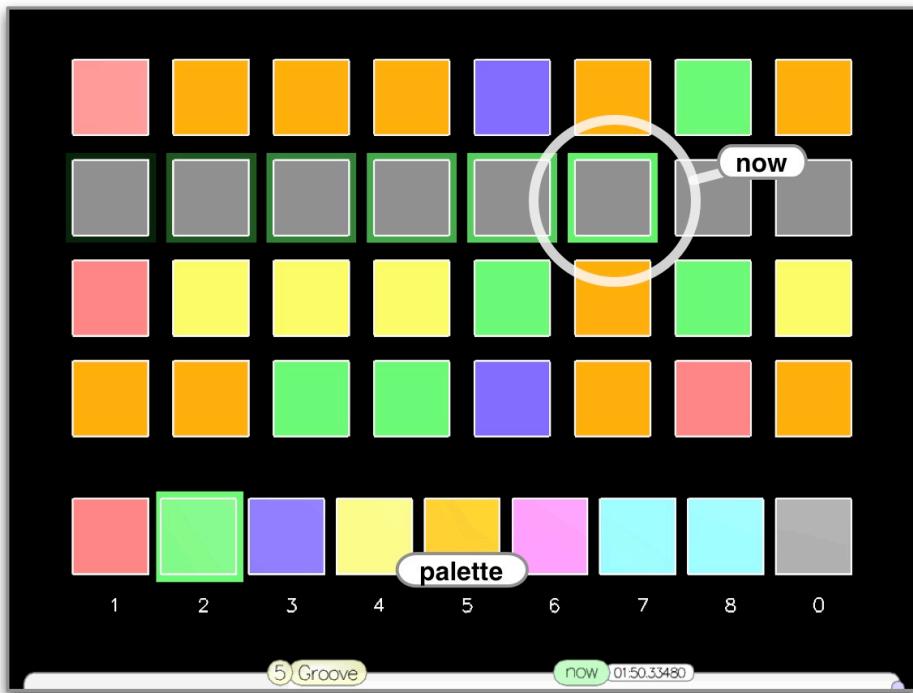


## Research: PLOrk: Princeton Laptop Orchestra (*continued*)

New (and completely insane) computer-mediated performance ensemble:  
15 x humans, 15 x laptops, 15 x 6-channel hemispherical speakers



networked game-based performance interface (above); networked step sequencer (below)



## **Research: sdnpeek + rt\_lpc + rt\_pvc**

*Real-time visualization of sound, basic audio features, and synthesis algorithm  
(for fun and pedagogy)*

**Date:** 2004 - present

**Roles:** Project lead, creator of sdnpeek, and co-creator of rt\_lpc and rt\_pvc

**Collaborators:** Perry Cook, Ananya Misra, Paul Botelho, and Ahmed Abdullah

### **Publications:**

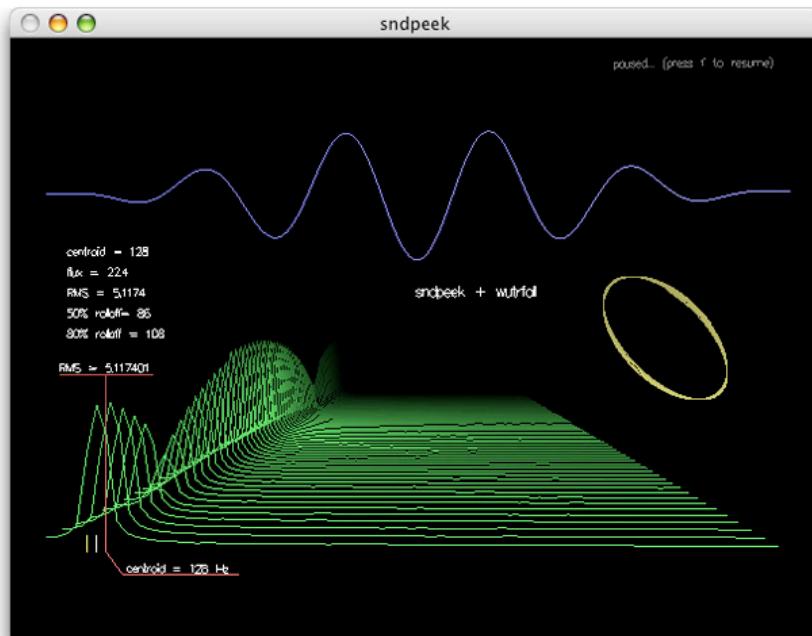
Misra, A., Wang, G., and P. R. Cook. 2005. "SndTools: Real-time Audio DSP and 3D Visualization." *In Proceedings of the International Computer Music Conference.* Barcelona, September 2005.

### **URL:**

<http://sndtools.cs.princeton.edu/>

### **Applications and Milestone:**

- sdnpeek used in John Chowning's recent performance of *Stria*
- I first created sdnpeek to visualize audio features, but discovered it's memorizing and fun appeal to children: they immediately learn to play with the sound of their own voice and see how it "looks".
- have demo'ed sdnpeek to more than 1000 kids of all ages for learning about sounds
- used in various classrooms to visualize waveforms, spectra, basic audio features
- sdnpeek currently being used in museum exhibit



## **Project: Gigapop Ritual : Audio Over Networks for Live Performance**

*Interactive networked musical performance between Princeton, NJ and Montreal Canada, over Internet2 and CA2Net, at International Conference on New Instruments for Musical Expression, 2003.*

**Date:** May 2003

**Roles:** Chief audio + MIDI + video networking developer, digital spoon player

**Collaborators:** Ajay Kapur (project lead), Perry Cook, Philip Davidson, Dan Trueman, Manjul Barghava, and Tae Hong Park

### **Publication:**

Kapur, A., G. Wang, P. Davidson, P. Cook. 2005. "Interactive Network Media: A Dream Worth Dreaming?" *Organized Sound*. 10(3): 209-219.

### **URL:**

<http://gigapop.cs.princeton.edu/>





**Ge Wang : teaching**

Teaching: ***In the Service of Electro-Acoustic Music:  
Digital Signal Processing + Software Design/Implementation Techniques***  
Dartmouth College Electro-Acoustic Graduate Seminar

**Date:** Fall 2006

**Instructor:** Ge Wang

**Duties:** Designed curriculum integrating DSP topics and software engineering techniques for computer music, delivered weekly lectures and held discussions on various topics, held extensive lab sessions.

**Course description:**

This course explores computer music synthesis and analysis, from the angles of DSP theory and software design/implementation issues and strategies. We will discuss digital signal processing concepts and their practical applications in sound synthesis. We'll also cover (and practice) important software design principles, and write lots of cool code. Additional emphases include software reuse, real-time systems, and designing interactive software. Programming will primarily be in C++ and Java, with additional uses of MATLAB and other environments and languages.

Topics draw from signal processing (theory and applications, convolution, filter analysis, Fourier analysis, windowing, z-transform, classic synthesis techniques, physical modeling, speech modeling, etc.) to computer science (interactive system design, object-oriented principals, design patterns, real-time audio, data structures for sound synthesis, optimization, C++/Java).

**Notes:**

- Seven (stellar) Electro-Acoustic Music graduate students in seminar
- Curriculum especially prepared to teach DSP concepts with applicable software engineering techniques
- Weekly teaching load: 2 x 2 hour lectures + additional all day lab/help session
- Commuted weekly between Princeton and Hanover, NH in order to teach seminars in both locations
- It was incredibly fun to teach, and I think/hope the student found it useful and fun.

## **Teaching: Composing for Laptop Orchestra**

*Princeton University (Music/CS) Graduate Seminar*

---

**Date:** Fall 2006

**Instructors:** Perry Cook and Ge Wang

**Duties:** Designed curriculum for PLOrk classroom, prepared and delivered weekly lectures on various topics (see below), held weekly help sessions, rehearsals, composed pieces for and co-directed public performances.

### **Course description:**

This seminar explores issues in creating performance pieces for computer-mediated ensembles, such as PLOrk. Topics include:

- programming and software design using ChucK and Max/MSP
- composing for laptop orchestra and plork performance
- networking, synchronization, conducting; sensors & mapping
- performance interface design
- sound synthesis / analysis, sound design (and whatever else might be useful)

The format will consist of presentations, individual and group assignments (programming, creating compositions for various size plork ensembles, performance), and a beer laboratory component, which takes place immediately after each class. There will be weekly in-class demos and performances, as well as several public PLOrk performances (we'll write pieces together for these). The class will be hands-on, example-driven, and very open.

### **Points of Interest:**

- Unique classroom setting “naturally” combining pedagogy in computer science, computer music, synthesis, music performance and composition
- 15 students (Computer Science and Music graduate students and upper-level undergraduates)
- 20 (and counting) new PLOrk pieces this semester
- Co-directed (with Perry) 4 public concerts
  - *PLOrk NYC Debut* at the Ear to the Earth Music Festival
  - Genomics Institute Performance in honor of Princeton President Shirley Tilghman and guests
  - *PLOrktastic Chamber Music: Duets, Trios, Quartets, and Quintets*
  - (upcoming) *Winter Concert* to take place in January 2007

### **Course webpage:**

<http://plork.cs.princeton.edu/courses/fall2006/>

## Teaching: *Princeton Laptop Orchestra, Year 1*

Princeton University (Music/CS) Freshman Seminar and Graduate Ensemble

**Date:** Fall 2005 + Spring 2006

**Instructors:** Dan Trueman, Perry Cook, Scott Smallwood, and Ge Wang

**Duties:** Co-developed PLOrk as an ensemble and classroom, co-designed curriculum for PLOrk classroom, prepared and delivered lectures on various topics held extensive studio sessions, rehearsals, composed pieces, working closely with students.



## Teaching Assistantship: Computer Science

*Princeton University + Duke University*

---

**Date:** 1997-2000 (Duke); 2001-2003 (Princeton)

**Duties:** Held office hours and precepts, tutoring, graded homework, developed software libraries for project use.

### **Award:**

*Outstanding Undergraduate Teaching Assistant Award.* Duke University Computer Science Department, 2000. For *Computer Graphics* (Fall 2000) and *Advance Object-oriented Design* (Fall 2000).

### **Courses assisted:**

Princeton:

*Advanced Programming Techniques* (Spring 2003)

Professor: Brian Kernighan

*Human Computer Interaction Technology* (Fall 2002)

Professor: Perry Cook

Duke:

*Computer Graphics* (Fall 2000)

Professor: Pankaj Agarwal

*Advanced Object-Oriented Design* (Fall 2000, Spring 1999)

Professors: Owen Astrachan and Robert Duvall

*Data Structures* (1997-1999)

Professors: Owen Astrachan, Robert Duvall, Susan Rodger



**Ge Wang : selected musical works**

## Music: *On-the-fly Counterpoint*

*Double Projection Live Coding Performance Duet*

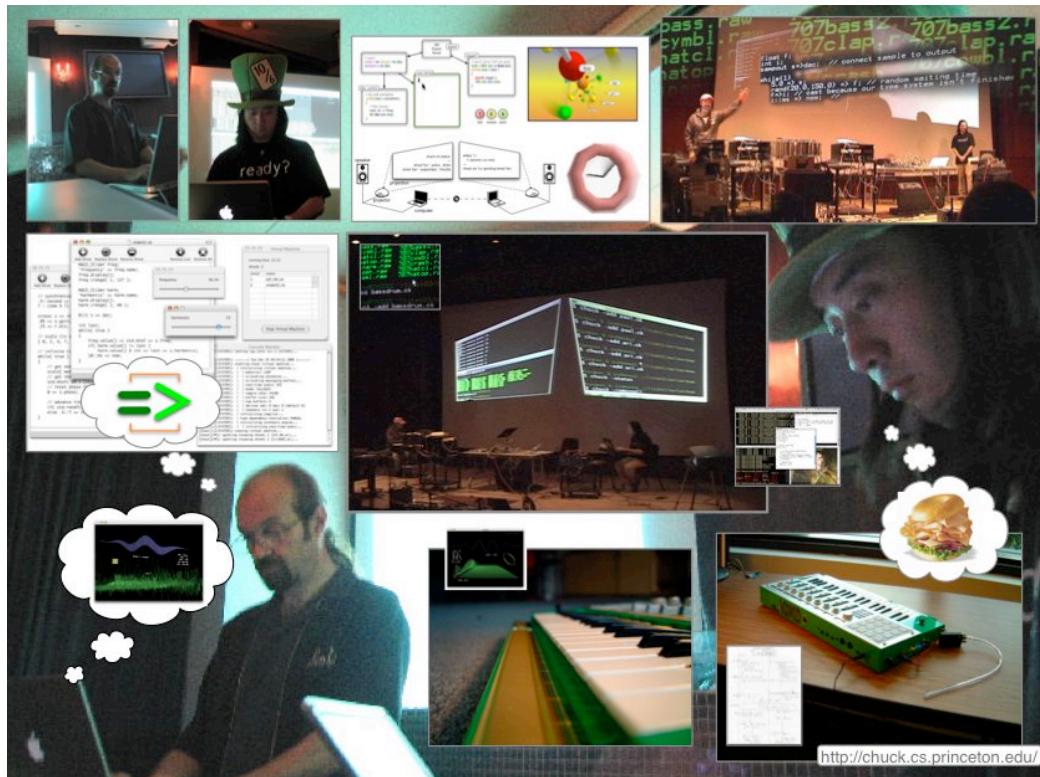
**Date:** 2003 - present

**Collaborator:** Perry Cook

### Program notes:

This piece is a study of the technical and aesthetic aspects of "on-the-fly" audio programming for synthesis and performance. We use the new ChucK synthesis language, which supports real-time, sample-synchronous, concurrent audio programming, and a highly "on-the-fly" style of programming, in which the composer | performer | programmer augments and modifies multiple programs while they are running, without stopping or restarting.

"On-the-fly Counterpoint" begins with a blank ChucK program. As part of the performance, we project the entire process on the screen for the audience to see and follow. We construct the counterpoint piece-by-piece in real-time, using the facets of concurrent audio programming and on-the-fly programming in ChucK. Contrapuntal simultaneities can be separated and compartmentalized into autonomous, concurrent entities. We can program and reason about each entity independently, as well as interact with other entities and with the program as a whole. This is part of our ongoing investigation into using code as an interactive and expressive musical instrument. It is also an instantiation of the ideas in our NIME2004 paper: "On-the-fly Programming: Using Code as an Expressive Musical Instrument".



## Music: *Loom (Etude II pour un enfant seul)*

A Musical Tapestry for 8-channel tape

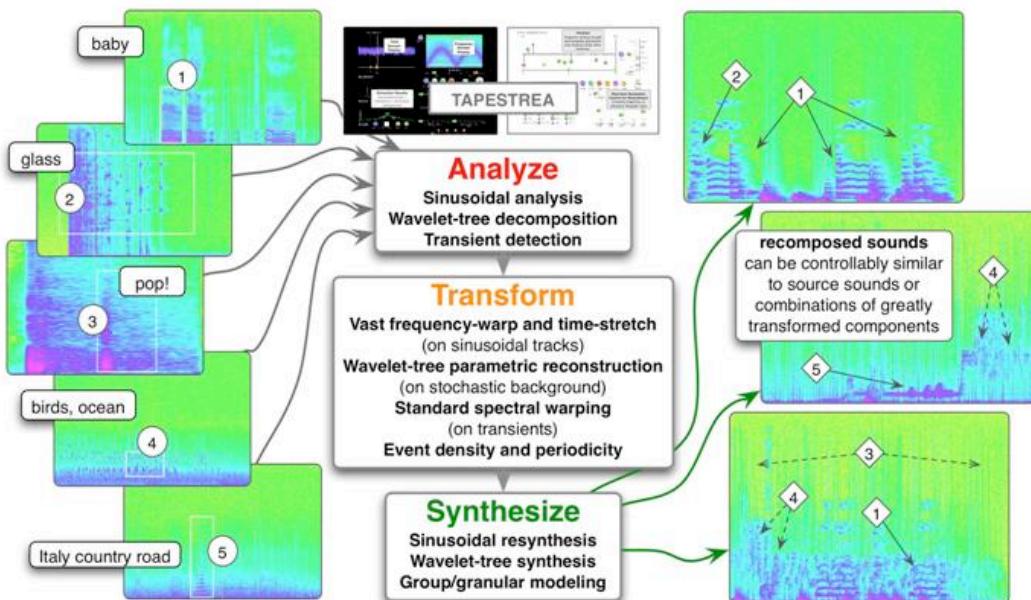
**Date:** Spring 2006

**Composers:** Ge Wang, Perry Cook, and Ananya Misra

**Performance:** International Computer Music Conference 2006, New Orleans

### Program notes:

"Loom" was created using a new technique of composition called Musical Tapestry, with which the sound sculptor is able to selectively separate real world sound scenes into reusable component templates and re-compose these templates (with potentially massive time, frequency, and other transformations) into a sonic tapestry. The template extraction, transformation, and re-synthesis are enabled by the new software framework TAPESTREA, which integrates sinusoidal modeling, wavelet-tree decomposition, and transient extraction / manipulation with a novel interactive interface, providing a complete workbench for separating, manipulating, and re-composing natural and other sounds. "Loom" was re-composed using only a handful of different templates extracted from recordings of natural sounds. They include 1) a bird squawk, 2) bird chirp, 3) duck quack, 4) lutine bell, 5) children yelling (9 instances). We re-compose these in the tradition of *Musique Concrète*, but with tools previously unavailable. For example, a bird cadenza shows the massive time-scale differences one can achieve by morphing a flock of birds (created from a single chirp template) over a wide range. Also, the children's drone in the second movement is achieved using 100x time-stretching and 50x frequency-warping. Granular synthesis, stochastic modeling, and other techniques are also exploited in our system and in the composition "Loom."



## Music: *Non-specific Gamelan Taiko Fusion*

*Human controlled, machine synchronized gamelan matrix*

**Date:** Fall 2005

**Composers:** Perry Cook and Ge Wang

**Performances:**

- January 2006 PLOrk Debut Concert
- April 2006 PLOrk Premier
- May 2006 Dartmouth College
- October 2006 Performance for University President Tilghman

Recording:

<http://plork.cs.princeton.edu/listen/debut/non-specific.mp3>

**Program notes:**

This piece is an experiment in human controlled, but machine synchronized percussion ensemble performance. Various percussive sounds are temporally positioned by PLOrk members, and the piece gradually transitions from tuned bell timbres to drums as the texture and density grows. Uses networked step sequencer interface (see PLOrk under *research* section for screenshot).



*A performance of Fusion; human conductor to the right.*

## Music: *Clix*

*Human conducted, machine synchronized maelstrom of clicks and raindrops*

**Date:** Spring 2006

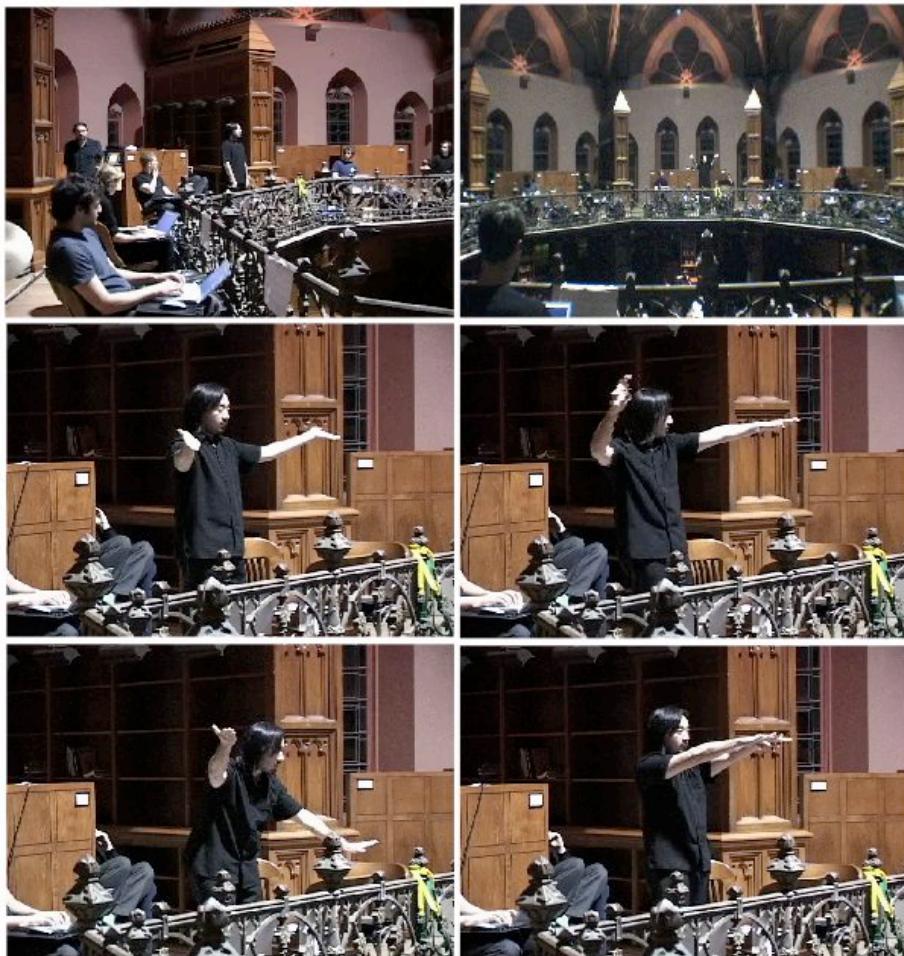
**Composer:** Ge Wang

**Performance:**

- May 2006 PLOrk in the Round
- May 2006 Dartmouth College
- October 2006 Performance for University President Tilghman

### **Program notes:**

In this piece, human operators type to make sounds, while their machines synthesize, synchronize, and spatialize the audio. Every key on the computer keyboard (upper/lower-case letters, numbers, symbols) is mapped to a distinct pitch (using the key's ASCII representation) and when pressed, emits a clicking sound that is synchronized in time to a common pulse. A human conductor coordinates frequency range, texture, movement, and timing (see below).



## Music: ***ChucK ChucK Rocket***

---

*Game-based, networked musical performance*

**Date:** Spring 2006

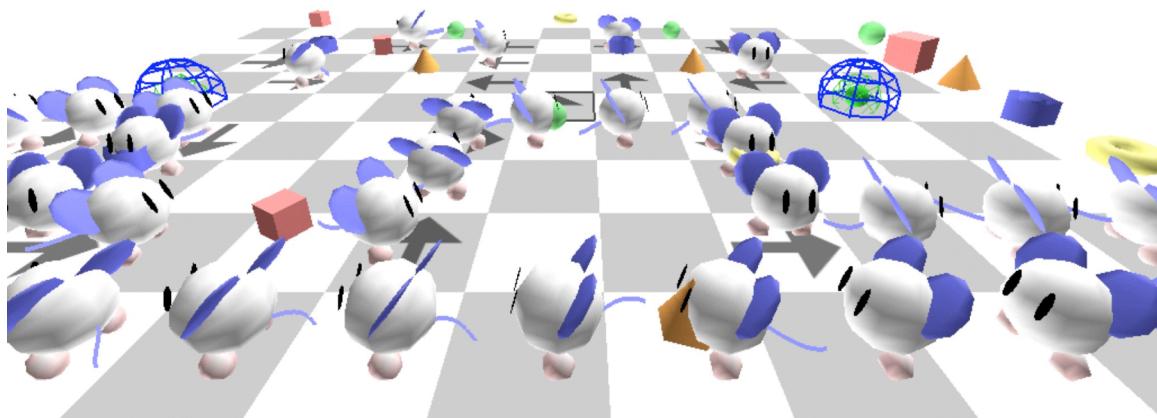
**Composers:** Scott Smallwood and Ge Wang

**Performance:**

- April 2006 PLOrk in the Round

**Program notes:**

This game piece is a study that reflects our interest in creating games scenarios in which the sounds produced are part of an interactive sound composition. In this game, based on Chu Chu Rocket, mice are released onto a large grid. Each player has a piece of this grid, and is able to cause the running mice to change direction by placing arrows in their path, and they are also able to place objects in their path, which make sound when the mice run over them, synchronized with those of other players. Thus, a player can create a kind of instrument with their piece of the grid, trapping groups of mice into loops that contain sound objects of their choosing. They can also send mice to and receive mice from their neighbors through network portals, thus the mice are shared throughout the entire group.



## Music: *Crystallis*

*"Bowing" the laptop to generate the sonic environment of wind and crystals*

**Date:** Fall 2006

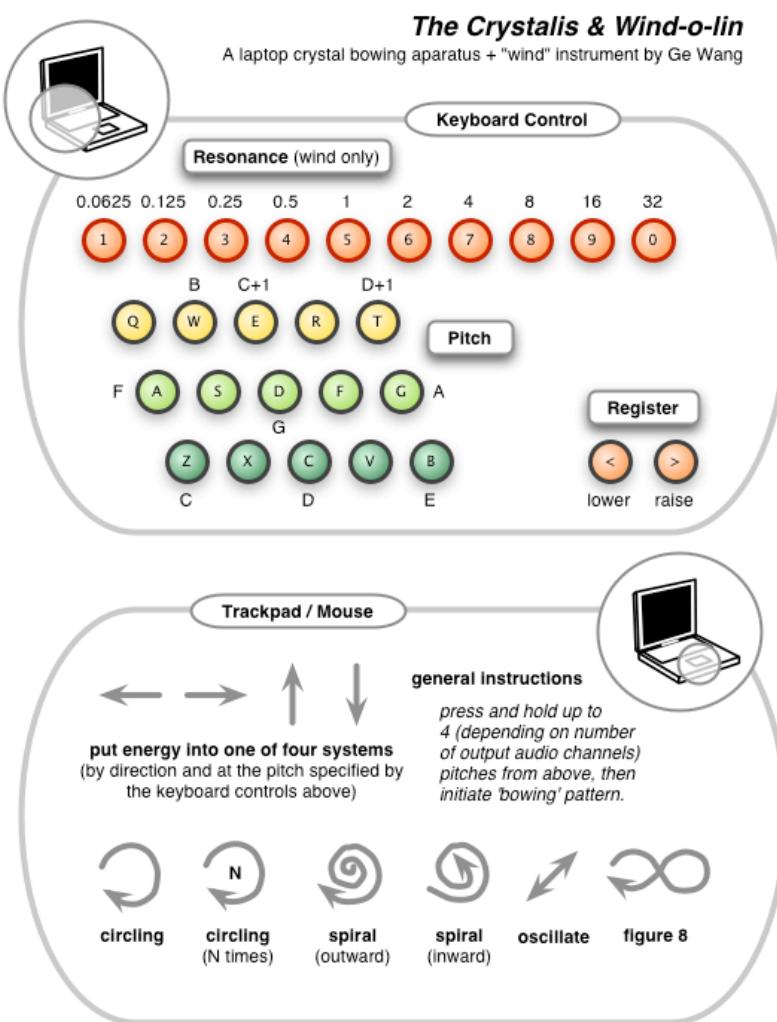
**Composer:** Ge Wang

**Performance:**

- October 2006 PLOrk NYC Debut at Ear to the Earth Festival
- October 2006 PLOrk Performance for President Tilghman

### Program notes:

This is a sonic rumination of crystal caves in the clouds, where the only sounds are those of the wind and the resonances of the crystals. It uses two simple instruments called the *crystallis* and *wind-o-lin*. These instruments make use of the laptop keyboard (which controls pitch and resonance) and the trackpad (which the players "bow" in various patterns to generate sound). The instrument instructions are reproduced below:



**Music: *PLOrk Beat Science***

---

*Duet: Adventures in Flute and HyPLOrkussion*

**Date:** Fall 2006

**Composers:** Rebecca Fiebrink and Ge Wang

**Performance:**

- November 2006 PLOrktastic Chamber Music, Princeton NJ

**Program notes:**

Adventures in Flute and HyPLOrkussion. 2 humans + 1 flute + 5 laptops + 5 MIDI drum pads + 30 audio channels.

[Too new for pictures...]

## Acknowledgments

In making the works described in this portfolio, I am indebted to a great many people for teaching, helping, guiding, and encouraging me throughout. I am deeply grateful to Perry Cook for his teaching, mentorship, friendship, and for granting me the freedom to explore and forge new (sometimes crazy-seeming) directions while providing the guidance and encouragement to help me to see things through. Immense gratitude to Dan Trueman for his always considerate and generous help and guidance, and (to both Dan and Perry) for trust and confidence in me to help develop the laptop orchestra. I've also been amazingly fortunate to work with Paul Lansky, Ken Steiglitz, Brian Kernighan, and Andrew Appel. I thank them for guidance from and to many directions.

Every project described in this document have benefited immensely from people working hard together. It continues to be my great honor to work with peers and fellow students and soundlab folks Ajay Kapur, Ari Lazier, Philip Davidson, Spencer Salazar, Adam Tindale, Paul Botelho, Matt Hoffman, Jeff Bernstein on everything from ChucK, the Audicle, PLOrk, visualization controllers, to networked concerts. I am truly grateful to Ananya Misra – together we've helped to create and pulled through many projects, demos, papers, and presentations. I am incredibly honored to work with Scott Smallwood, whether it's surviving PLOrk or composing/performing together. Very special thanks to Rebecca Fiebrink, for being amazing in everything we do.

I've been incredibly fortunate to also collaborate with many great mentors and colleagues outside of Princeton. My profound thanks to: Jon Appleton for his unfaltering support and friendship. Also to Larry Polansky and Kui Dong for their confidence and encouragement, including inviting me to teach the graduate seminar at Dartmouth. To the graduate students in the Dartmouth Electro-acoustic Music program, Charlie, Sean, Katia, Danny, Carmen, Courtney, and John, for a great experience and seminar. To Yuri Spitsyn for his friendship and help, and for just being great. Special thanks to Gary Scavone for being a great colleague (even though we rarely see each other in person), John Chowning for a wonderful project to visualize *Stria*, Chris Chafe for his support in exploring audio networks (and for a refreshing summit), George Tzanetakis for endlessly fascinating discussions on software design for audio systems, Georg Essl for his advice and encouragement, Nic Collins and Shawn Decker for great encouragement and interest in my work, Brad Garton for crazy and wonderful ideas. I profusely thank the ChucK users and developers communities, as well as Nick Collins, Alex McLean and other fellow live coders and colleagues at TOPLAP. I leave out many other wonderful folks to whom I am indebted and whom I keep at heart.

Finally, thanks to my grandparents who are responsible for the good in me, my parents for standing behind me and encouraging my interests, to Manman for her sacrifices and undying support.

Thank you!