Teaching Pd-extended/GEM within an artistic engineering, cooperative learning model

John Harrison
Wichita State University
Hack.Art.Lab
1701 E. Marion Rd.
Wichita, KS 67216 USA
john.harrison@alum.mit.edu

ABSTRACT
This paper is a discussion of the methods and materials I have developed to introduce Pd-extended/GEM to classes of college students of mixed disciplines. Over four years of offering and revising the class, my focus in developing this method has shifted from teaching the material directly toward facilitating an internal community of instructors and students all acting as both teachers and learners to explore the material collaboratively. Besides being more effective in introducing the material to the students in a meaningful way, this Constructionist-inspired approach also provides students a guided introduction to the open-source community.

Keywords
Pd, GEM, Pd-extended, Pedagogy, Constructionism, Tutorial

1.INTRODUCTION
Technology: Art and Sound by Design (TASD) is a semester-long course offered at Wichita State University (WSU) focusing on the application of digital technology in visual and performing arts [9]. Now in its fourth year, TASD is open to both fine arts students and engineering students; they work together across disciplines to create art pieces which incorporate digital technology. The course culminates in a public exhibition of team-built artwork.

In TASD, students are introduced to Pd-extended as well as the Arduino[1]. All final projects for the course are built using at least one and frequently both of these tools.

The constraints under which students are guided to become fluent using Pd-extended are rather severe; after completing our 3-week Pd-extended module, students must have acquired enough information and intuition with the software to develop meaningful projects. This is followed by a similar 3-week Arduino unit. While advancing through the technical material, students are also engaging in artistic observation assignments as well as discussion of the role of art in society and their individual lives. The remaining 9 weeks of the semester are spent exploring the artistic element of their projects, developing, implementing, and evolving ideas.

Over time I have found it most effective to guide the students to develop a sense of how to learn and think about Pd-extended/GEM, rather than teaching them the material directly through a more traditional instructional model. Applied methods of teaching include the use of introductory tutorials but also make extensive use of a wiki, blog, and email discussion list. These tools support the collaborative element of Constructionist philosophy. My goal is to guide students to develop sustained interest in the material and tools so that they continue to expand their Pd-extended skills in order to express themselves artistically or creatively after the course's completion.

2.TARGET AUDIENCE
Wichita State University (WSU) students enroll in Technology, Art and Sound by Design (TASD). Most of these students are electrical and computer engineering majors, some are art majors, and all have had at least two years of university study.

In their studies, the engineering students have developed theoretical knowledge with no strong application component. They have almost no experience building real-world hardware or software, and have little comprehension of rapid prototyping or development. Most of their education has been individual-based, and their instruction, assignments, and tests follow a rigid framework with clearly-defined objectives and a quantitative measurement of success. Their program does not emphasize unique work, instead focusing on a understanding of basic principles as enforced by problem sets and tests[4]. Consequently, their concept of documentation consists mostly of proof that they understand the answers that they submit in their assignments and tests.

These engineering students typically come from local Wichita families and attend school in pursuit of jobs and promotions within the aircraft industry. They do not look toward engineering or their academic work as creative or an outlet for exploration but instead see it as a way to make a decent living and be a productive working member in their community.

In contrast, artists in the class understand their work as creative and expressive. Surprisingly, they are often more experienced at building and at practical real-world design than the engineers are. However, they are used to their work as individual, personal, and independent. While their lack of technical experience creates a greater challenge when learning Pd-extended than for their engineering counterparts, they are often hesitant to initially embrace outside contribution to the projects in which they are invested.

3.OBJECTIVES
The primary objective is to create sustained interest in the material. Our biggest indicator for success is how many students continue to use Pd-extended after the class is over. This objective is in contrast to the often assumed objective of covering as much material as possible within the time-frame of the class. Instead, I prefer to consider the course content a medium through which to communicate how to “think” in Pd-extended. Once the students have developed an intuition for the language and their own personal method for exploration, they can self-direct their learning of the material independent of the class.

4.TEACHING PHILOSOPHY

I have seen more success with my primary objective by pursuing a Constructionist model of teaching. *Constructionism* is a learner and community-based model. As the wikipedia puts it, "Constructionist learning involves students drawing their own conclusions through creative experimentation and the making of social objects. The constructionist teacher takes on a mediational role rather than adopting an instructionist position. Front of class teaching 'at' students is replaced by assisting them to understand —and help one another to understand—problems in a hands-on way.[3]"

5.PEDOGOGICAL IMPLEMENTATION

Guided by pedagogical philosophy of Constructionism, and over four iterations of the course, I have developed 10 basic principles for teaching Pd-extended.

5.1Teach Pd-extended

When I first began teaching this class, I started with Pd-vanilla, then discussed external libraries as we moved toward video manipulations within Pd. However, I have found that for beginners, the concept of external libraries adds a barrier for both learning and sharing of code. First, students need to understand what the libraries are, how they are installed, and how to use them. Then, when sharing code, they need to make sure that the people they share with all have the same external libraries with matching versions.

For beginners, Pd-extended eliminates the need for any of this conversation. Within Pd-extended, most libraries exist across platform. Students can immediately get started with large sets of external libraries without being distracted by installation issues and configuration differences between machines. Instead, all the students basically start with the same version of the same tools, whether on Windows, Mac OS, or Linux. This greatly reduces their barrier for learning Pd itself and facilitates students sharing their patches while working on different computers on a variety of platforms.

5.2Create a Framework Which Facilitates Collaboration

Our Pd-extended introduction, in keeping with the general course structure, makes extensive use of a class wiki[9], blog[7], and email discussion list[8]. All tutorials, instruction, instructors notes and similar content go on the wiki. Class assignments are posted to the blog. All student questions, related or inspirational resources, and any relevant chatter are sent to the discussion list.

The wiki, blog, and email discussion list together support the social nature of the Constructionist teaching method. Students can correct instructor mistakes, clarify confusing content or offer their own instruction on the wiki. Blog posts and comments facilitate collaboration. Finally, the class discussion list gives students the opportunity to answer each other's questions and otherwise converse about material they find interesting that could be applicable to the course.

The class discussion list especially creates the opportunity for students to feed on each other's enthusiasm. The entire online collaborative structure helps support a commitment to student learning within a social community context. It has the added byproduct of introducing them to some of the tools and collaborative methods of the open source community.

5.3Start With an Introduction to GEM

Pd-extended is first and foremost an audio synthesis language. Why introduce Pd-extended by starting with an external library for video?

After trying various approaches over the iterations of this course, I have found that mainstream students are generally most interested in video manipulations. By starting with video, the students become self-motivated more easily and quickly.

5.4Teach to the GEM Help Patches

If students are to be self-directed in their learning, they need to become comfortable with Pd-extended's primary source of documentation for beginners: help patches. Specifically, given the primary objective of the course, I have the responsibility of getting the students comfortable with GEM's help patches as quickly as possible.

In preparing to teach the course, I went through all of the GEM help patches in Pd-extended's 5.reference/GEM and examples/GEM directories to determine basic objects and concepts that occur throughout most of these patches. By focusing on the concepts and objects used consistently, I helped the students become comfortable enough with them that they could learn to modify the patches themselves.

Besides being an effective way to learn Pd-extended, modifying pre-existing patches helps students more generally comprehend the value of using and learning from existing code. Our students are not used to this. They have instead written code for their previous university classes from scratch so as not to be accused of cheating. Our model gives them a more efficient method for creating and designing code. More importantly, it offers an exploratory and social component to the learning process.

5.5Give the Opportunity for Public Recognition to all the Work

Students feel rewarded and more often hold themselves accountable when their work is recognized by their community and by the public. Therefore, all assignments are submitted on a publicly-viewable class blog. This allows each student to share their work with their peers and the larger community. Moreover, I encourage and sometimes assign students to copy, credit, then modify each other's work.

Each class begins with guided discussion of the homework, citing specific examples from student submissions. Students can share their difficulties, successes, and outstanding questions. Together, we brainstorm possibilities for expanding on their submitted work to include greater artistry, different functionality, and adherence to coding convention. Besides providing motivation, this activity facilitates students learning from each other as part of a community.

At the conclusion of the Pd-extended module, the students submit a “Pd Mini Project.” Selected projects are displayed in prominent areas around campus.

5.6Teach through Inquiry

To satisfy the primary objective, I teach students to learn how to learn Pd-extended, instead of teaching them Pd-extended itself. Therefore, I avoid giving the students direct information about Pd-extended whenever possible.

For example, when creating tutorials I do not give answers to what Pd-extended will do in this or that case. Instead, I ask the students to create patches to discover these answers themselves. Moreover, I do not give students the code to the patches we use in

our tutorials, but instead provide screen shots. This way, students become more familiar with the Pd-extended environment as they recreate the patches. Moreover, during this process of rebuilding the patches, students are more likely to individually modify them, driven by their own curiosity.

During class time, if a student asks how Pd-extended will behave in a particular situation, I ask them how they might test for that behavior. If they are unsure, we work together as a class. I may also ask directed questions to facilitate the process.

5.7Keep the Structure Flexible

Our primary objective is addressed first and foremost when students are continually engaged in the material, both individually and as a community. Therefore, students need to be allowed to pursue their individual interests within Pd-extended and be rewarded for sharing them with the class.

To facilitate this, we offer a category on our class blog where students can post whatever Pd-extended patches they wish, whether directly connected with the course material or not. Students are also encouraged to ask questions and openly discuss anything on the email discussion list. As interesting questions and interesting patches arise from these conversations, we take class time to recognize, discuss and develop them.

5.8Use Analogies and Metaphors

Students best grasp the material when it is related to concepts with which they are already familiar. Moreover, abstractions are best understood when they are compared to familiar things existing in the real-world.

For example, when describing Pd-extended initially, I talk about how component stereo systems are connected and draw a metaphor between the audio signal of a stereo system, and Pd-extended's cables. Messages in Pd-extended are like knobs and sliders on the stereo. [dac~] is just a set of speakers.

As we review material, I help the students recreate tutorial patches by reviewing our analogies and metaphors. Documentation in our patches includes our metaphorical descriptions, and I ask the students to do the same with their own patches.

5.9Always give examples

Since students by definition are not already familiar with the end-goal of their exploration with Pd-extended, they lack the context to fully understand the motivation for my tutorials, assignments, and instruction. Without this context, it can be difficult for them to understand how to learn the material or how to complete the assignments.

If I am to keep the students' focus on the material, I must not distract them with these less-related questions. Instead, I can answer these questions by continually modeling everything I ask them to do. If I give an assignment, I accompany it with at least one example assignment. When confronted with a question to which I do not immediately know the answer, I puzzle it out with them so they see my thought process exposed. This can be more valuable for them than the answer itself.

By taking this approach, I help the students stay focused on exploring the material itself. They are less distracted trying to "guess" what I am looking for in their submissions.

5.10Keep It Fun, Uninhibited, and Inspired!

The primary objective is most satisfied if students fall in love with the material. This is more likely to happen if the students witness my own excitement for both the material and sharing it with them. My goal is that my students know me as a Pd-extended enthusiast. Whenever they want to talk Pd-extended, I make time for them. As I ask myself questions and discover new possibilities with Pd-extended, I share them on the email discussion list. I am freely available to them for Pd-extended conversation.

Fortunately, this behavior is authentic for me. It perhaps makes me a bit silly and goofy, but I hope fun as well. And while this approach might be a bit unconventional within the academic framework, there's no compelling reason to hide it. On the contrary, perhaps the joy I experience from learning and teaching the material might be contagious. Ideally, it might help inspire some of the students to reexamine their own motivations for learning.

6.Environment for learning

To set an environment most conducive to learning, I exercise several practices throughout my Pd-extended instruction:

6.1Demo Everything Live

Instead of telling students what Pd-extended does, I always demonstrate. This requires that for all classes I am teaching continually from a projector and powered speakers attached to my laptop running Pd-extended. As much as possible, I come in with no pre-made patches but instead generate the patches in front of them. This way, they pick up many aspects of the Pd-extended environment through demonstration. This approach also helps me remember to tell them things about the environment I might otherwise forget to mention.

6.2Use a Workshop Model

I strongly encourage students to bring their laptops to every class. During class, they follow along, making Pd-extended patches that mirror the ones I am creating at the front of the class. As they run into issues and questions, they can freely interrupt the demonstration.

6.3Set The Tone Before The Class Starts

The time students are filing in before class is often overlooked. As students shuffle in, their context for learning becomes set by the atmosphere of the classroom.

To help set an inquisitive environment in the classroom, I get to class early and have Pd-extended running on my laptop, projecting as they enter the classroom.

Depending on my own mood, I may at that point run a pre-existing Pd-extended patch from the Pd-extended community, run one of my own created patches, or just explore Pd-extended with some of my own inquiry. In demonstrating patches I always show work I find personally inspiring. For example, when Cyrille Henry released Vivarium[10], I could not wait to show a part of it before class started.

7.IMPLEMENTATION

The Pd-extended introduction was divided into 3 sections, one for each week. Below constitutes my outline for each of those 3 weeks:

7.1Week 0: Preparation

The week before beginning the Pd-extended module, students are given instructions to install Pd-extended and check to make sure the sound is working. I also set up a lab time for them to seek help from me and each other if they have any problems. For the first

class of the module, I ask them to bring their laptops, ready to follow along as I create and project Pd-extended patches.

There are generally few problems with installation; almost all of those on Windows Vista and solved by increasing the latency of the audio buffers in Pd. For students interested in better audio performance in Windows, I help them install and configure ASIO4ALL, a free universal ASIO driver for WDM audio[2]. ASIO4ALL gives vastly improved latency times over WDM audio drivers, but did not run successfully on all machines.

7.2Week I: Introduction

7.2.1Pd-extended demo

It is hard to see the value of a language without also being exposed to its literature. To help students immediately grasp the value of Pd-extended, I start the module with a demo of various patches written by me and by previous students. Besides demonstrating what the patch does, I also show what the patch itself looks like and offer a high-level explanation of how the patch works.

Students are not expected to understand with any detail how these demo patches work. As they get used to this, they may become more comfortable exploring and using code they do not completely understand. At the same time, the demos help them start to intuitively grasp the graphical nature of Pd-extended as well as the concept of a dataflow language.

7.2.2Introduction to Pd-extended and GEM

At the point when a critical mass of students appear to be developing a curiosity for Pd-extended itself, I shift from demos to discussion.

So as to give them a context from which to understand conversations about Pd in the general Internet community, I compare and contrast Pd-vanilla and Pd-extended for them. Then I explain GEM as a cross-platform external library which is included in Pd-extended, allowing the Pd framework to also describe video manipulations.

I start the conversation with a quick overview of Pd-extended as a dataflow language. With pictures, we compare Pd-extended to old telephone consoles and module synthesizers.

I expand the dataflow concept by drawing an analogy to components of a stereo system. Objects in Pd-extended can be compared to components such as CD players, turntables, 8-track tape players, FM radios and speakers. Cables in Pd-extended are like cords that connect components together.

I introduce messages as analogous to the knobs, plugs, sliders of components. For example, a CD player can't make any noise until you turn on the power button, then hit “play.” Therefore, if a CD player were a Pd-extended object, messages to it might include [on(and [play(.

Last, we talk about *edit* and *performance* mode. For this I draw on the analogy of being both the creator and user of your own stereo.

7.2.3Baby's First Patch: a live video camera

Once the students understand how Pd-extended objects, messages and cables map to components, knobs and buttons, and cords, I ask them how they imagine we might show a live video feed in Pd-extended. Together we map out the components and connections necessary.

As the students describe the components needed, I create the analogous objects in Pd-extended, documenting their mapping to real-world components. Eventually we end up with working patch similar to the one shown in Figure 1:

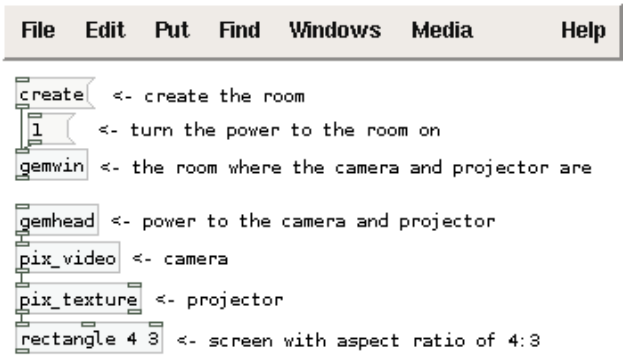

Figure 1: Baby's First Patch

7.2.4A recorder and a player

Using the same process, we work together as a class to create one patch which records from the webcam using [pix_record], and another patch which plays back the recording using [pix_film].

For reasons unimportant for the class or this paper, the movie is upside down when played back by [pix_film] in Linux[6]. This gives me the opportunity to introduce the first video processing pix_ object: [pix_flip], but I do not at this point explain the technical reason for the issue; I only discuss the object that will provide the intended effect.

I tell the students about [pix_movie] but discourage them from using it. I feel [pix_movie] by not requiring [pix_texture] breaks the analogies we are beginning to establish in our mapping from Pd-extended objects to real-world objects.

7.2.5Pd/GEM tutorial

Having completed webcam and video file recorders and players, the students can continue their learning outside of class time. To guide them through this, I created a 3-part Pd/GEM tutorial.

Consistent with the outline of the module itself, the primary objective of the tutorial is to give the students the motivation and knowledge to understand GEM help patches. A full detailed description of the tutorial is beyond the scope of this document, however the tutorial itself is available online at: http://cratel.wichita.edu/cratel/PdVideoTutorial

The tutorial breaks down into 3 parts:

7.2.5.1Part I: Review of Week I course material

The tutorial starts with a brief summary of all aspects covered in class: Pd as a dataflow language, objects, messages, etc.

7.2.5.2Part II: Numbers and Sliders

Part II walks the students through most of the basic objects and concepts they will need to understand GEM help patches. Specifically, the tutorial introduces them to:

- hot and cold inlets
- Atoms
- Horizontal and Vertical Sliders
- Bang
- Depth-first-search
- Choosing order of execution with trigger
- Dollar signs for parameter passing
- Loadbang
- Interactive help

Consistent with the class instruction, the tutorial provides direction by giving the students small patches to make and asking them questions about them. For example, Figure 2 shows a small excerpt from the tutorial, showing the students the use of dollar signs:

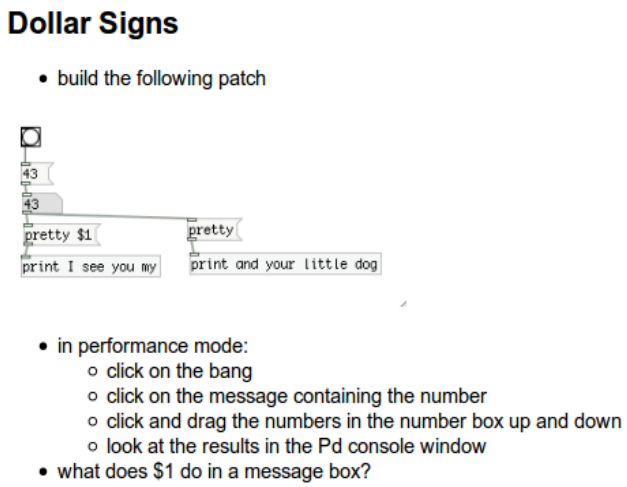

Figure 2: Introduction to Dollar Signs

7.2.5.3Part III: Manipulating GEM help patches

The purpose of Part III is get the students familiar with the concept of modifying existing patches to learn about Pd/GEM.

The beginning of the tutorial focuses on giving the students a method to become familiar with a patch:

1. Choose a patch
2. Save the patch in your own directory with your own filename
3. Close and re-open the patch to confirm that it still works
4. Read and assess the comments
5. Click on messages, toggles, bangs etc. to figure out what it does

The last part of the tutorial gives two examples, stepping the students through a modification of a GEM help patch.

7.2.6Assignment: hello Pd A

Following the example in Part III of the tutorial, students modify a GEM help patch and post it to the wiki. Their post includes a narrative of what they modified, any instructions somebody would need to interact with the patch, a screenshot of the patch, and the patch itself.

7.2.7Concepts and Keywords covered

I keep a list of keywords and concepts in front of me when presenting in class. I continually try to weave these into class instruction as students ask questions, suggest ideas and we explore the material together. My list for the first week is:

- objects
- messages
- comments
- Edit and Performance mode
- GEMhead and GEMwin
- right-clicking (or control-clicking) objects for interactive help
- List of objects is available at http://en.flossmanuals.net/PureData/ListofObjects
- rendering

7.3Week II: Audio

7.3.1Review of student assignments

I start the class demonstrating and discussing a few assignments submitted by students in the class that class as a whole might find interesting and that naturally will lead us to subsequent lesson material. We brainstorm together possible further work for each patch and may try a few modifications.

Besides providing the students with positive feedback in the classroom environment, this review facilitates giving students standard methods for tweaking patches. These methods include smoothing transitions using [line].

7.3.2Sampling: Digital audio and digital video

My introduction to audio begins with a comparison of frame rate and control rate. With frame rate specifically, the students already have some intuition that they are seeing multiple pictures each second when watching a digital video.

We extend the conversation to consider audio rate. The students explain to each other what audio sampling rate is. We have completed this section when the students understand that audio sampling is fast but consists of only one number typically between -1 and 1, while video sampling is slow but consists of a matrix of numbers which together describe an entire picture.

7.3.3Audio objects

Together, we initially explore audio objects as a means to automate GEM patches. For example, [phasor~] and [osc~] can be used with [snapshot~] to move through frames of a video. I draw an analogy between "Compute Audio" in the Pd-extended console window and [GEMhead], both serving as a means to turn media objects "on."

7.3.4Common Pd control objects

The students and I brainstorm modifications to patches with which we are familiar. I use their ideas as we build patches that introduce the following objects and concepts:

- [metro]
- [select]
- [delay] and [pipe]
- [moses] and [spigot]
- subpatches and abstractions

7.3.5Assignment: Hello Pd B

Now that the students are starting to get familiar with Pd-extended, they can begin to create Pd-extended patches that express their own ideas and concepts. In this assignment, the students choose their own idea then build a patch to explore that idea. Example ideas have been explored in concurrent modules for the class and include the passage of time, experimentation with light, and relatedness amongst unrelated objects.

To encourage further exploration and collaboration, I urge students to "steal" each other's ideas and modify each other's patches from the *Hello Pd A* assignment. To complete the assignment, students submit to the blog their idea, how their patch demonstrates their idea, a screenshot of their patch, the patch itself, and any other supporting files such as images and videos.

7.3.6Assignment: Pd-extended mini project

While not due until the end of the Pd-extended module, I assign the Pd-extended mini project so the students can be thinking about throughout the rest of the module. The assignment is simple: create a standalone video and/or audio piece which is written in Pd-extended and can be displayed publicly.

Specifically the students are to:

- use Pd-extended
- optionally make the patch interactive
- make the patch clear and usable by anyone completely unfamiliar with pd-extended
- illustrate a conceptual idea they have developed
- post all patches, instructions, video, audio files and stills to the blog with full instructions so anybody can download the patch piece and try it fully

The students know I will choose a selection of projects for display in prominent areas around the campus.

7.4 Week III: Advanced Concepts

7.4.1Review of student assignments

As with Week II, I use student assignment submissions as a springboard for conversation and the introduction of new material.

7.4.2Audio manipulation

By the end of the last week, the students are familiar with the following concepts and ideas:

- recording and playing back a sound
- playing a sound backwards
- the difference between phasor~ and line~
- scratching through a sound
- pitch shifting, time stretching and contracting
- filters: low pass, high pass, band pass
- subtractive synthesis
- [soundfiler], [readsf~], [writesf~]

7.4.3Pduino

As the Arduino module directly follows the Pd-extended module, I end the Pd-extended unit with a few examples of controlling Pd-extended using an Arduino, sensors, and Pduino[5].

8.Conclusion

Currently there are limited resources for those teaching Pd-extended and GEM to beginners. Having taught a Pd-extended module to university students for four years, I have come up with a framework for teaching Pd-extended. While I continue to evolve my method, I feel it has been effective in its primary objective to create sustained interest in Pd-extended beyond the time-frame of the class. The framework I use borrows heavily from the Constructionist model, leading students to be active creators and collaborators within their own learning community.

9.ACKNOWLEDGEMENTS

I wish to thank Wichita State University's College of Fine Arts and College of Engineering for allowing me to continue to pilot the experimental course, *Technology: Art and Sound by Design*. The course is team-taught and I get continual, valuable feedback from the other instructors: Kristin Beal-DeGrandmont, Lauren Hirsh, Tom McGuire, Keith Neufeld, and Ann Resnick. Thank you to the developers of Pd, Pd-extended, and GEM. The Pd email list (pd-list@iem.at) and the GEM developers email list (GEM-dev@iem.at) have been an invaluable resource of information to me. Finally, thank you to all the students who have taken *Technology: Art and Sound by Design.*

10.REFERENCES

[1] Arduino. http://www.arduino.cc/

[2] ASIO4ALL. http://www.asio4all.com/

[3] Constructionism (learning theory). http://en.wikipedia.org/wiki/Constructionist_learning

[4] Introduction to WSU Electrical and Computer Engineering Undergraduate Education. http://webs.wichita.edu/?u=ECE&p=/undergrad_intro/

[5] Pduino. http://at.or.at/hans/pd/objects.html

[6] [pix_video] upside down. http://www.mail-archive.com/pd-list@iem.at/msg06356.html

[7] Technology: Art and Sound by Design (blog). http://cratel.wichita.edu/blogs/TASD2009/

[8] Technology: Art and Sound by Design (email list). http://groups.google.com/group/WSU-Tech-Art-and-Snd

[9] Technology: Art and Sound by Design (wiki). http://cratel.wichita.edu/cratel/TASD2009

[10] Vivarium. http://www.chdh.net/vivarium