pima 7741

fall 2011

web page:	http://www.sci.brooklyn.cuny.edu/~sklar/pima7741/
instructors:	Prof John Jannone, TV/Radio, email: jannone@mac.com Prof Elizabeth Sklar, Computer Science, email: sklar@sci.brooklyn.cuny.edu
lectures and labs:	Thursdays 10.30am-2.30pm in 116RE
co-requisite:	PIMA 7010 (or permission of instructor)
required materials: for details see:	Max/MSP/Jitter http://cycling74.com/shop/discounts/

course description:
This course provides in-depth study of tools and techniques for designing dynamic and interactive multimedia systems for use in live performance situations. Emphasis will be on student creation of custom computer software to realize interactive projects. Video, audio, three-dimensional computer images, and computer-human interfaces will be addressed. Students will receive extensive instruction in graphical computer programming; no experience required. The course format consists of a mixture of lecture and laboratory class sessions, with strong emphasis on hands-on learning.

The following topics will be covered:

- introductory programming concepts
- sound and video playing, recording and processing
- hardware input and output
- graphics
- hardware and network integration

Program Goals Addressed By Course:
Demonstrate ability in conception and planning of professional-quality, collaborative-produced performance works (program goal 1)

Demonstrate ability in production, presentation, and documentation of professional-quality, collaborative-produced performance works (program goal 2)

Demonstrate ability in interactive multimedia programming for use in live performance, and creation of practical performance systems that include hardware and software components (program goal 8)

Objectives of Course:

1. Demonstrate an understanding of the nature, purpose, and process of computer programming.
2. Demonstrate specific knowledge of language, terminology, and techniques studied.
3. Demonstrate programming ability, including solving arbitrary programming problems.
4. Demonstrate ability to practically implement knowledge in a live performance setting.
5. Demonstrate ability in integrating hardware and software in a performance system.
6. Perform and/or present one's work capably, confidently, and with seriousness.

Outcomes Anticipated for Course:
Students will work independently and in small groups over the course of the semester to carry out practical projects in multimedia programming for use in live performance settings. A substantial software-based performance system controlling multiple media will be the final project of the semester. One or several smaller scale projects will precede the final.

Two exams will be given to all students. Students enrolled in the PIMA M.F.A. or Advanced Certificate programs will also take a “qualifying exam” in the PIMA 7010 class based on the material covered in this course, as knowledge of a common programming environment is a requirement of continuation in the PIMA M.F.A. and Certificate programs.

Method of Evaluation:

Software	20%	Objectives 1, 2, 3, 4 & 5
Performance	20%	Objectives 4 & 6
Exams	20%	Objectives 1, 2, 3, 4 & 5
Homework	20%	Objectives 1, 2, 3, 4 & 5
Class participation	20%	Objectives 1, 2, 3, 4 & 6

Method of Assessment:

Student work will be assessed according to the following rubric:

Objective	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
Demonstrate an understanding of the nature, purpose, and process of computer programming.	Students fail to show an ability to illustrate the nature, purpose, and process of computer programming through any of the following methods: writing, speaking, or creating simple computer programs.	Students are able to write, speak about, and demonstrate through the creation of simple computer programs the nature, purpose, and process of computer programming.	Students are able to write eloquently, speak convincingly, and demonstrate effectively through simple and complex computer programs the nature, purpose, and process of computer programming.
Demonstrate specific knowledge of language, terminology, and techniques studied.	Students are able to demonstrate only basic knowledge of language, terminology, and techniques in written and verbal forms, and through the creation of simple computer programs illustrating this knowledge, or are unable to so demonstrate.	Students are able to demonstrate broad knowledge of language, terminology, and techniques in written and verbal forms, and through the creation of simple computer programs illustrating this knowledge.	Students are able to demonstrate thorough knowledge of language, terminology, and techniques in written and verbal forms, and through the creation of simple and complex computer programs illustrating this knowledge.
Demonstrate programming ability, including solving arbitrary programming problems.	Students are unable to create simple functional, error-free computer programs in either of these scenarios: when realizing their own ideas or in response to an assignment.	Students are able to create simple functional, error-free computer programs realizing their own ideas or in response to an assignment.	Students are able to create simple and complex functional and error-free computer programs realizing their own ideas or in response to an assignment, and are able to do so in an interesting or novel way.
Demonstrate ability to practically implement knowledge in a live performance setting.	Students are able to create a multimedia performance system that is not robust or efficient, or are not able to create such a system.	Students are able to create a simple, robust, and efficient multimedia performance system.	Students are able to create a simple or complex, robust, gracefully degrading, fail-safe, interesting, and efficient multimedia performance system.
Demonstrate ability in integrating hardware and software in a performance system.	Students fail to successfully integrate a single hardware device successfully into a performance system.	Students are able to integrate a single hardware device successfully into a performance system.	Students are able to integrate multiple hardware devices, or an unusual or particularly difficult-to-integrate device, successfully into a performance system.