

Music 320C: Audio Plugin Development in FAUST and C++

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

Spring Quarter, 2021-2022

1 Music 320C Overview

Music 320C focuses on audio plugin development in the FAUST and C++ languages, organized and supported by the JUCE multiplatform development framework. JUCE GUIs are constructed using Plugin GUI Magic. The principal activity is a software project due at the end of the quarter, after some initial assignments providing starter code and examples. Class meetings are devoted to discussing project progress, and going over portions of the JUCE Framework for audio plugin development, and the FAUST language for audio signal processing and the JUCE framework. Familiarity with C++ and elementary UNIX-style software development is assumed. Signal processing experience on the level of Music 320A and/or 320B is assumed.

2 When, Where, Who

Term: Spring Quarter
Location: CCRMA Class Room (Knoll 217)
Time: Mondays and Wednesdays, 4 PM
Instructor: Julius Smith (jos@ccrma.stanford.edu)
TA: Champ Darabundit (champ@ccrma.stanford.edu)
Website: <https://ccrma.stanford.edu/courses/320c/>

Prerequisite Software

While C++ development experience is assumed, no prior experience with the FAUST language or JUCE development framework is required. It is helpful but not required to have experience with an interactive programming environment supporting FAUST modules, such as ChucK, Pure Data, Max/MSP, and/or SuperCollider. Introductory materials appear below in the class schedule.

3 Administrative Information

Announcements

Class announcements are often made via *email*. For this we are presently using Piazza:

<https://piazza.com/stanford/spring2022/mus320c>

If you have signed up for the class in axess, by the first day of the quarter, you should receive an invitation from Piazza to join the class (using the email address known to axess). Otherwise, please join by visiting the above URL and entering your preferred email address.

Units

You may sign up for 1 to 10 units. The first unit corresponds to weekly class attendance (about three hours per week). Each additional unit should represent approximately 3 hours of focused activity for the class per week.

Prerequisite Knowledge

You are expected to know

- C++ (able to read and understand the basics, and able to look up anything new to you)
- Elementary signal processing on the level of Music 320A and/or 320B

Skills to Develop

This course should move you forward along the following dimensions:

- UNIX proficiency (working at the shell command line)
- git (and git submodules)
- C++ (at the level of JUCE, which is pretty modern)
- Faust (which compiles to C++ by default)
- JUCE (a large set of very useful C++ classes for audio plugins)
- Application and plugin development on the platform(s) of your choice
- 320-level audio signal processing

Your progress will naturally depend on how experienced you were coming in.

When you are up to speed on the above skills, extend your knowledge and expand your project!

Project Requirements

Project requirements apply to those enrolled for more than one unit. Your project must include

1. a JUCE audio plugin (which can be a standalone app as well),
2. at least one FAUST module (either by you or from the FAUST Libraries et al.), and
3. something you can show to the class during the last week of classes.

Project Schedule

1. Project plan due by the end of 2nd week of classes

2. Project presentation during the last week of classes
3. Final project turned in by email by the end of the quarter

Grading

Grades are based on class engagement, project updates/demos, and final project work. Those enrolled for one unit (class attendance only) should sign up for CR/NC.

4 Schedule

Below is the schedule of weekly in-class presentations, with pointers to all associated reading, lecture overheads, and so on. The default weekly topic is learning the FAUST language for signal processing and the JUCE development platform for C++ audio plugins. Additional discussion will be driven by the interests, projects, and any research activities of the participants. Results and associated pointers will be logged here.

- Week 1: Course Overview, Project Planning, and Introduction to the JUCE Framework
 - Course Overview:
 - * Plugin Gui Magic Equalizer Example (JUCE Audio Plugin)¹ [10:23]
 - * FAUST Standalone App: Graphic Equalizer Example² [15:33]
 - Read MUS320C main handout³ (this document)
 - Plan your Audio Plugin Project
 - * What novel signal-processing plugin would you like to have?
 - * Default Project: Modify a JUCE plugin example (such as one covered in class) to use a new Faust module in its signal-processing function(s)

¹<https://www.youtube.com/watch?v=8bZVHQk4L7k>

²<https://www.youtube.com/watch?v=o0WQQ2ABEHc>

³<https://ccrma.stanford.edu/~jos/intro320c/>