

A complete Computer Music System

Description:

Implement a complete computer music system with interaction design principles.

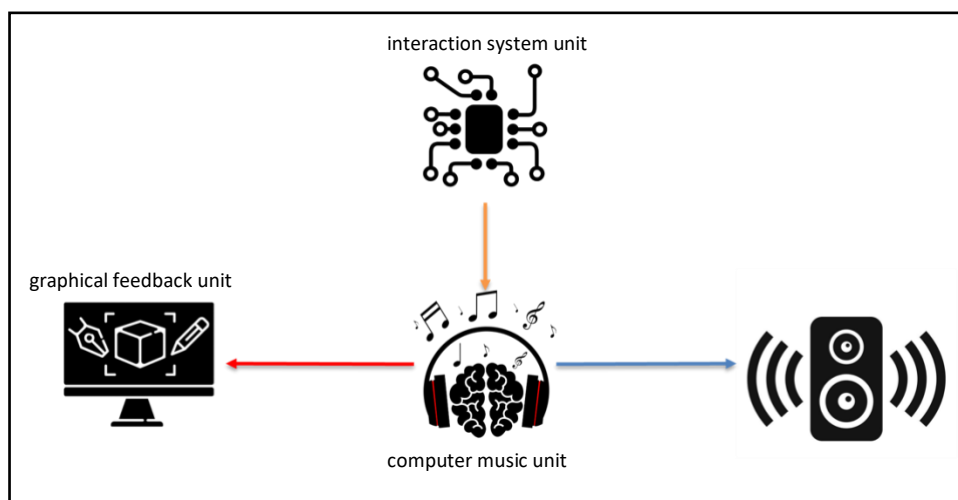
Learning objectives:

- Design the whole CMS: from the purpose of your final system (i.e., music performance system, human-interactive effects, augmented reality synthesis, and different cool things!);
- Decide and state the tools used and the communication protocols for sharing information;
- Implement your computer music system.

Tools:


Each project needs to comply with the interaction scheme presented below. You can expand your solution with other blocks and connections, if desired. Anyhow, the following three main blocks are required:

- interaction system unit
You can use any control device you wish to capture the user inputs (i.e., Arduino, Bela, Leap motion, Mobile phone, Kinect, etc...).
- computer music unit
You can use “SuperCollider” or “JUCE”.
For the JUCE case, you can decide to implement an effect that modifies a pre-defined virtual instrument (e.g., from Logic or other companies), or to implement a synth with an optional chain of external effects.
- graphical feedback unit
Implement a GUI with “Processing” to provide some graphical feedback to the user.



NB:

Do not focus on complicated things, but on original ones!

 POLITECNICO MILANO 1863	Academic Year 2021-2022 052483 Computer Music: Languages and Systems	Interaction Design Homework
--	---	--------------------------------

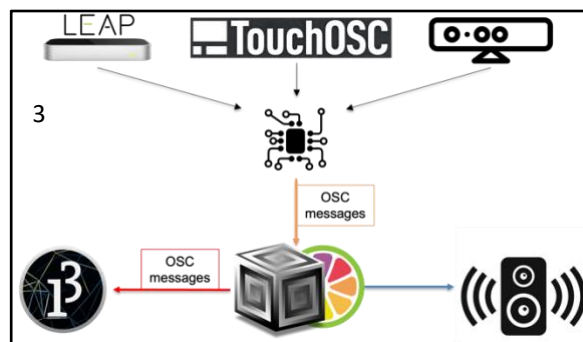
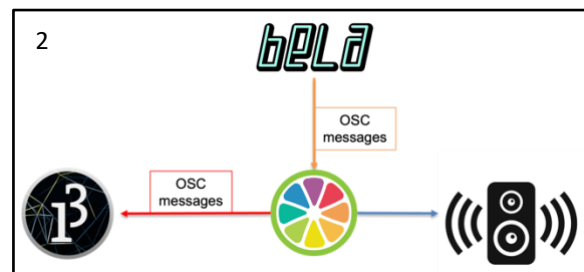
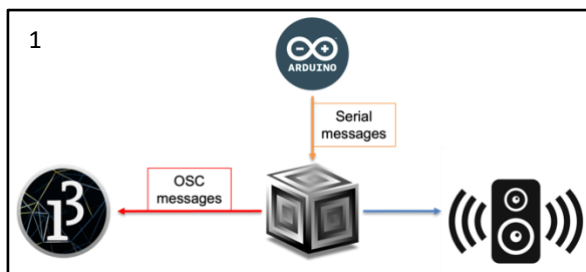
Be creative, feel free to exploit interesting solutions!

Tips:

Here below, we propose three possible schemes that satisfy the required tools.

You can implement one of the following interaction schemes or just take inspiration on them.

- scheme 1: use Arduino as control device, transmit gestural information to SuperCollider with Serial messages, provide the feedback events to Processing with OSC messages.
- scheme 2: use Bela as control device, transmit gestural information to JUCE through OSC messages, provide the feedback events to Processing with OSC messages.
- scheme 3: use any other external devices for control, share information to SuperCollider or JUCE with OSC messages, provide the feedback events to Processing with OSC messages.



Output:

- A more detailed report in which you illustrate your system and its implementation (max 8 pages). To be uploaded on GitHub (see the following item) by May 31.
- A public repository in the general GitHub repository <https://github.com/polimi-cmls-22>. Add in your repo a general README file with a brief description of the project. Identify the repo as specified in the guidelines. To be created and finalized by May 31.
- A brief presentation and demonstration of your work that will be given during the class (max 5 minutes). To be prepared by June 1st, 2022.