

Computer Music Practice Examples

Introduction

instrument design, composition, performance, and presentation

by Joo Won Park

www.joowonpark.net/cmpe

What is CMPE?

CMPE is a series of interactive lectures on a creative process of computer music

Each entry shows how a complete piece is created from scratch

Each entry has an analysis of codes, scores, and step-by-step procedures

Why

Computer music practice = composition + performance + instrument design

Details on the creative process are important

“Turn your Csound into Cmusic.” - Boulanger, The Csound Book

Target Audience

Composers, performers, technologists, and researchers of electronic music

Those who seek examples on the musical application of theories and ideas of music technology

Students who learned to use a computer music tool and needs to survey creative application of the tool

Limits

It is not a SuperCollider tutorial

It is not for learning history or theory of electronic music

Examples are author's pieces specifically written for this project

Format - Tools

Google Slides: diagrams, step-by-step procedure, references

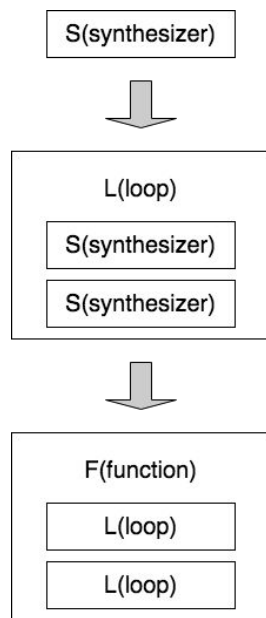
[SuperCollider](#): download and run .scd files in the presentation to follow along

YouTube: Demos

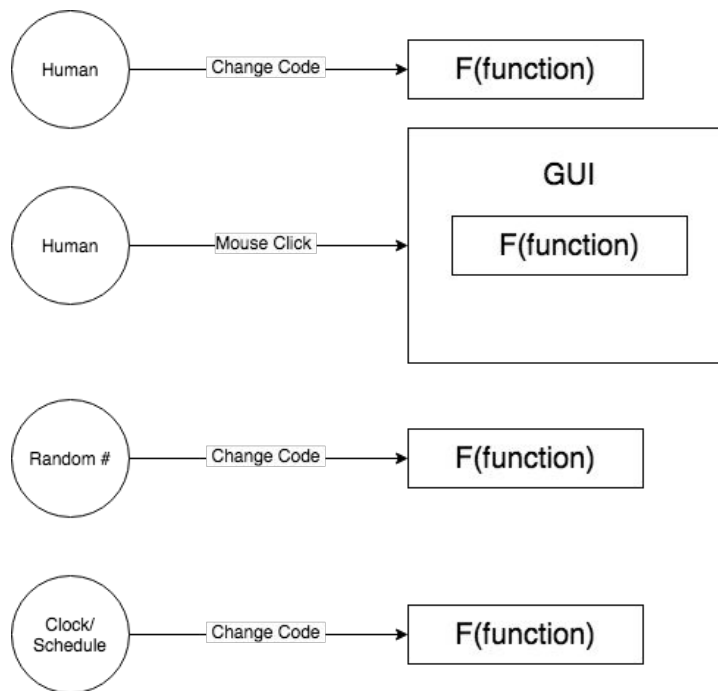
Contact: joowon (at) joowonpark.net

Format - Diagrams

Instrument Design Process

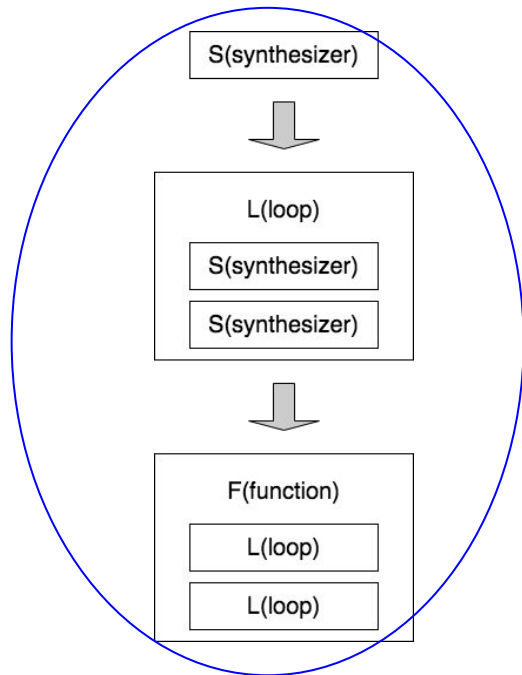


Presentation/Performance Methods

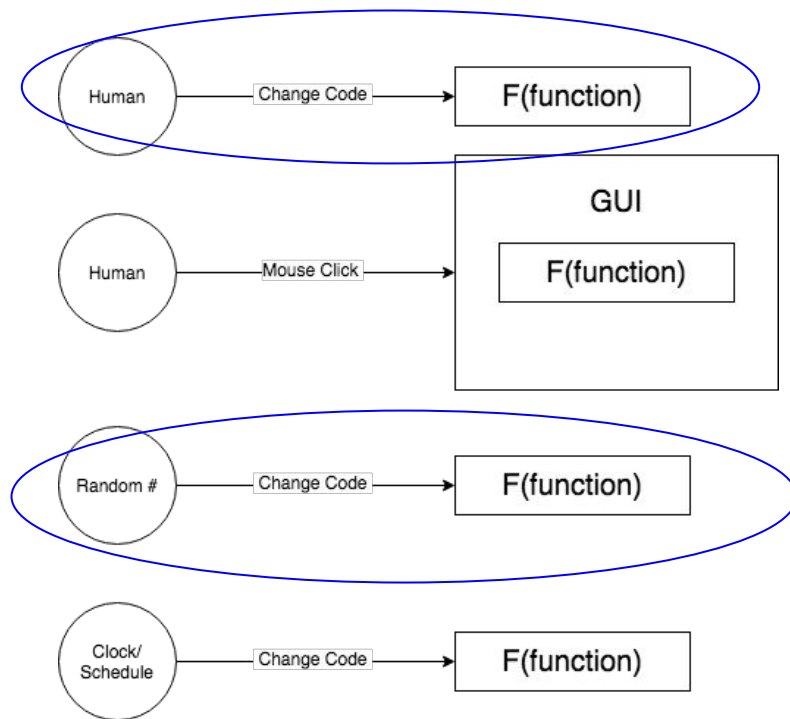


Format - Diagrams

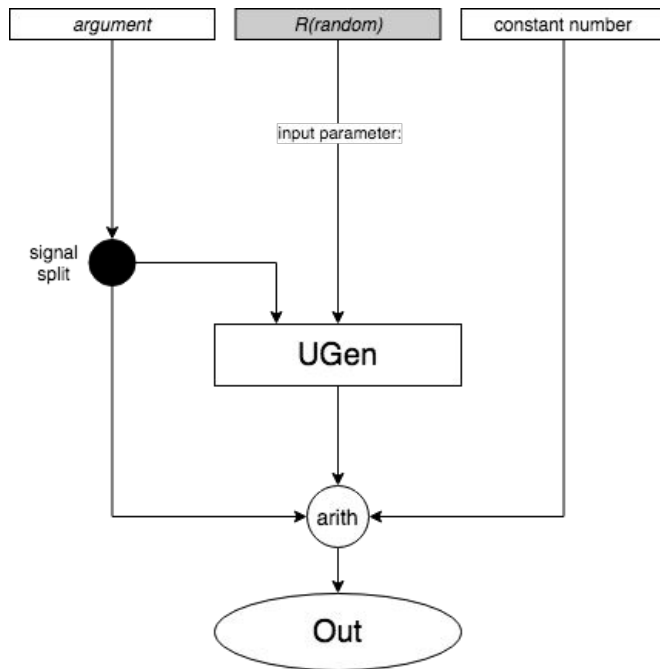
Instrument Design Process



Presentation/Performance Methods



Map



S(synthdef)

L(loop)

F(function)

Block Diagram Legend

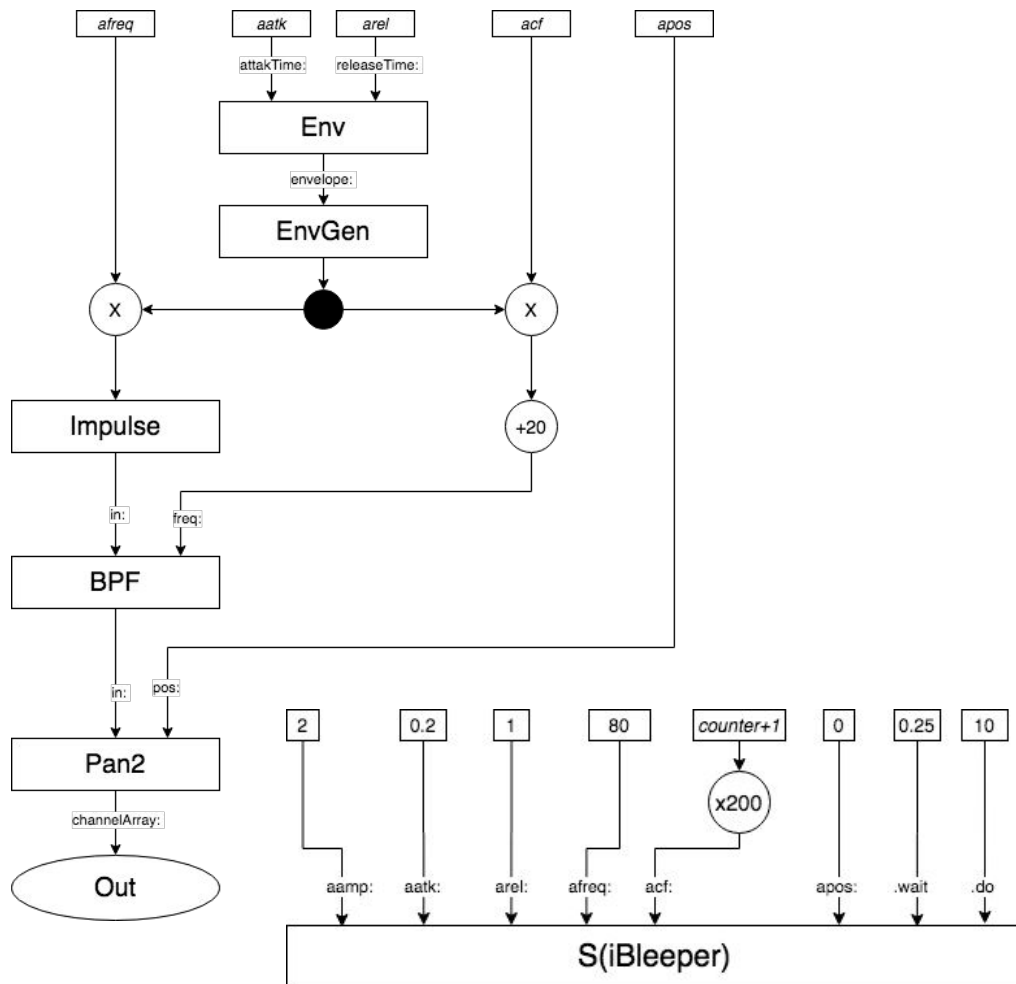
- UGen:
 - Unite Generator.
 - Processes audio or data
- arrow:
 - Shows the direction of signal
 - The text in the line (input parameter:) shows the name of the input parameter of an input in the connected UGen
- argument:
 - Controllable arguments
 - Written in italics
 - Can be a list in [arg,arg,arg] format
- constant number:
 - Discrete numeric value
- signal splitter:
 - Used when one signal is connected to multiple inputs
- Arith:
 - Arithmetics.
 - Incoming signals are added(+), subtracted(-), multiplied(x), or divided(/).
- Out:
 - Audible audio output
- S(synthdef) : SynthDef. Includes OSCFunc
- L(loop): loops including do{} and Routine
- F(function): custom function.

Example

Let's listen to this sound



Download [Intro.scd](#)

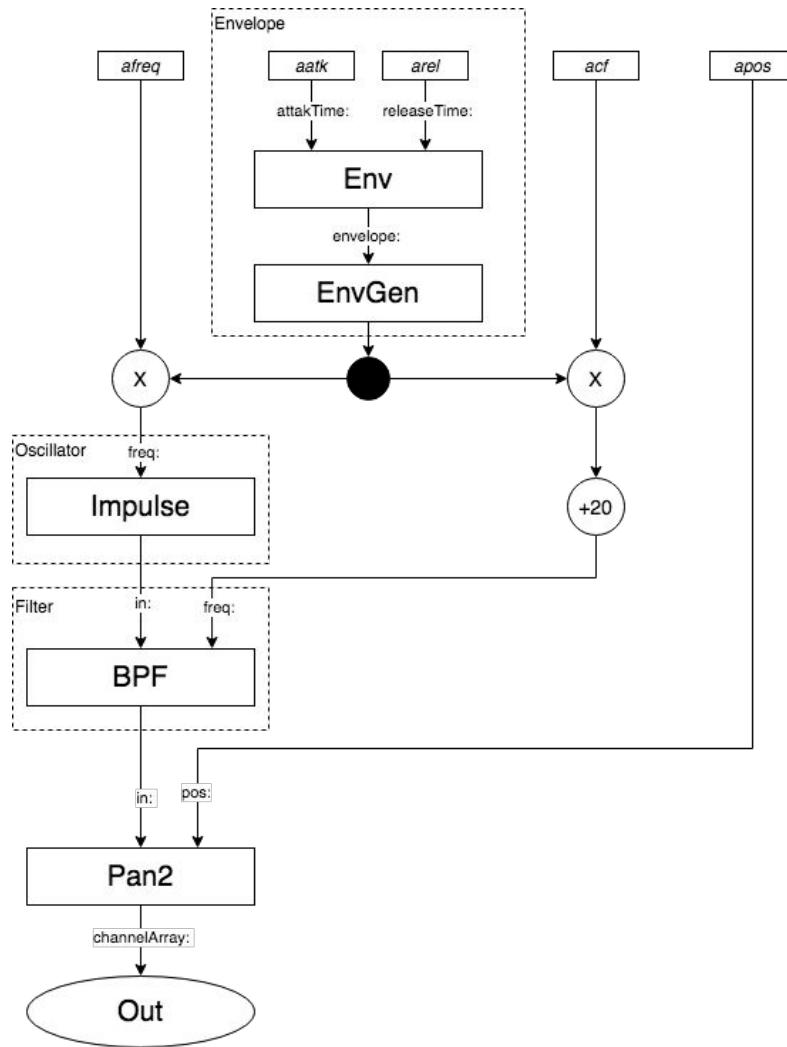


Example - Step 1

Make **S(iBleeper)**



Step 1 Audio Example



Example - Step 1

Make S(iBleeper)



Step 1 Audio Example

```
//Step 1 - Make an Instrument
SynthDef("iBleeper",{
  arg aamp,aatk,arel,afreq,acf,apos;
  var env,sound,filter,panner,mix;

  env=Env.perc(aatk,arel);
  env=EnvGen.kr(env,doneAction:2);

  sound=Impulse.ar(afreq*env);
  filter= BPF.ar(sound,acf*env+20);
  panner= Pan2.ar(filter,apos);

  mix=panner*env*aamp;

  Out.ar(0,mix);

}).add;

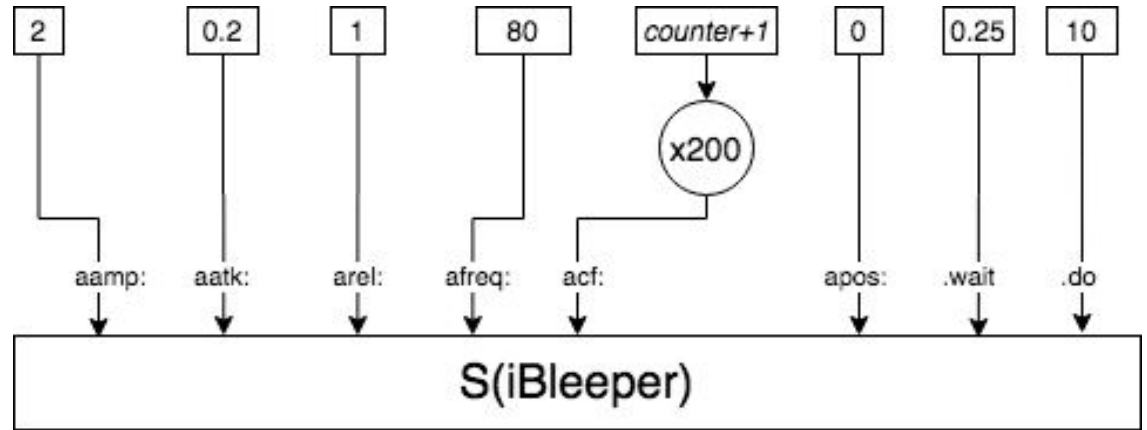
//test it
Synth("iBleeper",[aamp,1.5,aatk,3,arel,3,afreq,80,acf,3000,apos,0]);
```

Example - Step 2

Make a Loop



Step 2 Audio Example



In a loop, the argument *counter* increase by 1 at each iteration. The counting starts at 0, so `counter+1` outputs an increasing number series starting at 1.

Example - Step 2

Make a Loop



Step 2 Audio Example

//Step 2 Make a Loop

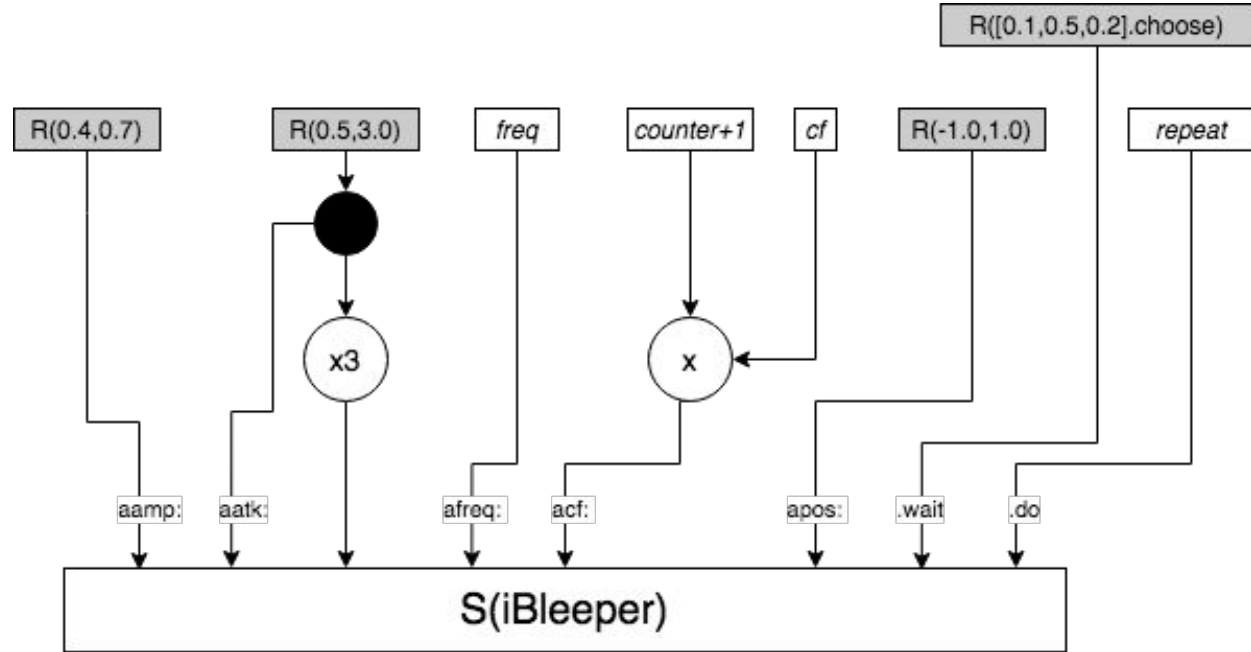
```
Routine({  
  10.do{  
    arg counter;  
  
    Synth("iBleeper",[\aamp,2,\aatk,0.2,\arel,1,  
      \afreq,80,\acf,(counter+1)*200,\apos,0]);  
    0.25.wait;  
  }  
}).play;
```

Example - Step 3

Make a Function with
rancomization elements



Step 3 Audio Example of
three different parameter
values



Example - Step 3

Make a Function with
randomization elements



Step 3 Audio Example of
three different parameter
values

```
//Step 3 Make a function with randomization
~beepout={
  arg repeat,freq,cf;
  Routine({
    var atk;
    repeat.do{
      arg counter;
      atk=rrand(0.5,3);
      Synth("iBleeper",
        [\aamp,rrand(0.4,0.7),\aatk,atk,\arel,atk*3,
         \afreq,freq,\acf,(counter+1)*cf,\apos,rrand(-1.0,1.0)]);
      [0.1,0.15,0.2].choose.wait;
    }
  }).play;
};

//~beepout.value(repeat,freq,cf increments);
~beepout.value(30,15,100);
~beepout.value(30,100,100);
~beepout.value(25,500,80);
```


Example - Step 4

Make a Function with
randomization elements



Step 4 Audio Example

Performance Instruction

//run the lines below. Improvise timing.
//Multiple runs are OK

```
//~beepout.value(repeat,freq,cf increments);  
~beepout.value(30,15,100);  
~beepout.value(30,100,100);  
~beepout.value(25,500,80);
```

What's Next

[RMHS : Ambient Sound Generator](#)

[SuperCollider Composition Examples](#)

Contact joowon@joowonpark.net if
you have questions or see errors.

RMHS

Start (with fade-in)

☐ record at start

loudness (range: 0.0-1.0)

amp

number of harmonics (range: 1-20)

repeat

fundamental frequency (Hz)

fundamental

interval (half steps, range: 0-24)

ratio

duration (seconds, excluding fade-in time)

dur |

☐ randomize

by Joo Won Park (www.joowonpark.net)