POLITECNICO DI MILANO

Music and Acoustic Engineering

Creative Programming and Computing

E.L.V.I.S.

Enhanced Looper with Visual Interaction and Sonification

STUDENTS

Attolini Silvio

Gorni Alessandro

Martinelli Riccardo

DATE

02/10/2023

PROFESSOR

Zanoni Massimiliano

E.L.V.I.S.

Enhanced Looper with Visual Interaction and Sonification

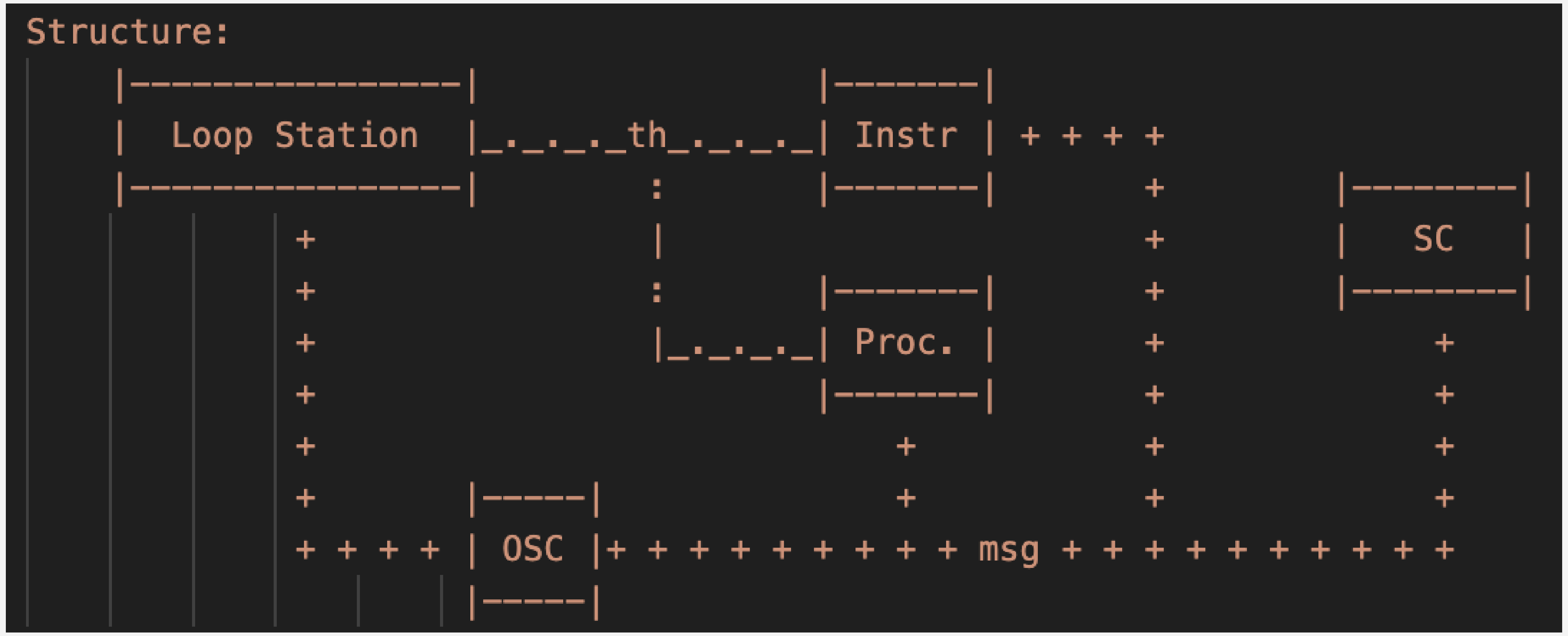
DESCRIPTION

TECHNOLOGIES

GOALS

STRUCTURE

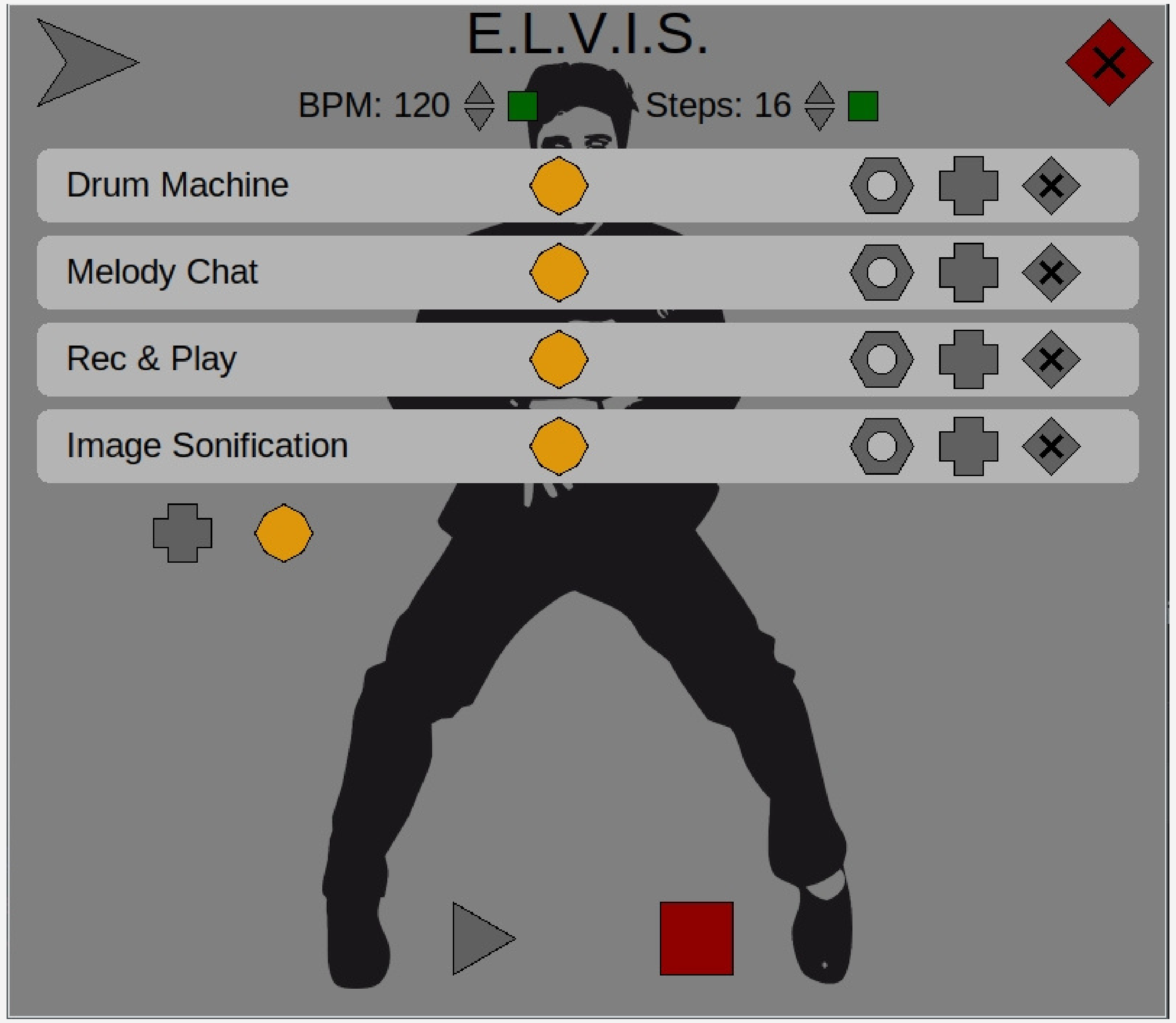
Simple and user-friendly Loop Station as artistic installation



```
Structure:
    |----------------|                |-------|
    |  Loop Station  |_._._._th_._._._| Instr | + + + +
    |----------------|        :       |-------|       +        |--------|
             +                |                       +        |   SC   |
             +                :       |-------|       +        |--------|
             +                |_._._._| Proc. |       +             +
             +                        |-------|       +             +
             +                            +           +             +
             +       |-----|              +           +             +
             + + + + | OSC |+ + + + + + + + + msg + + + + + + + + + +
                     |-----|
```

E.L.V.I.S.
Creative Programming and Computing

LOOP STATION MANAGER



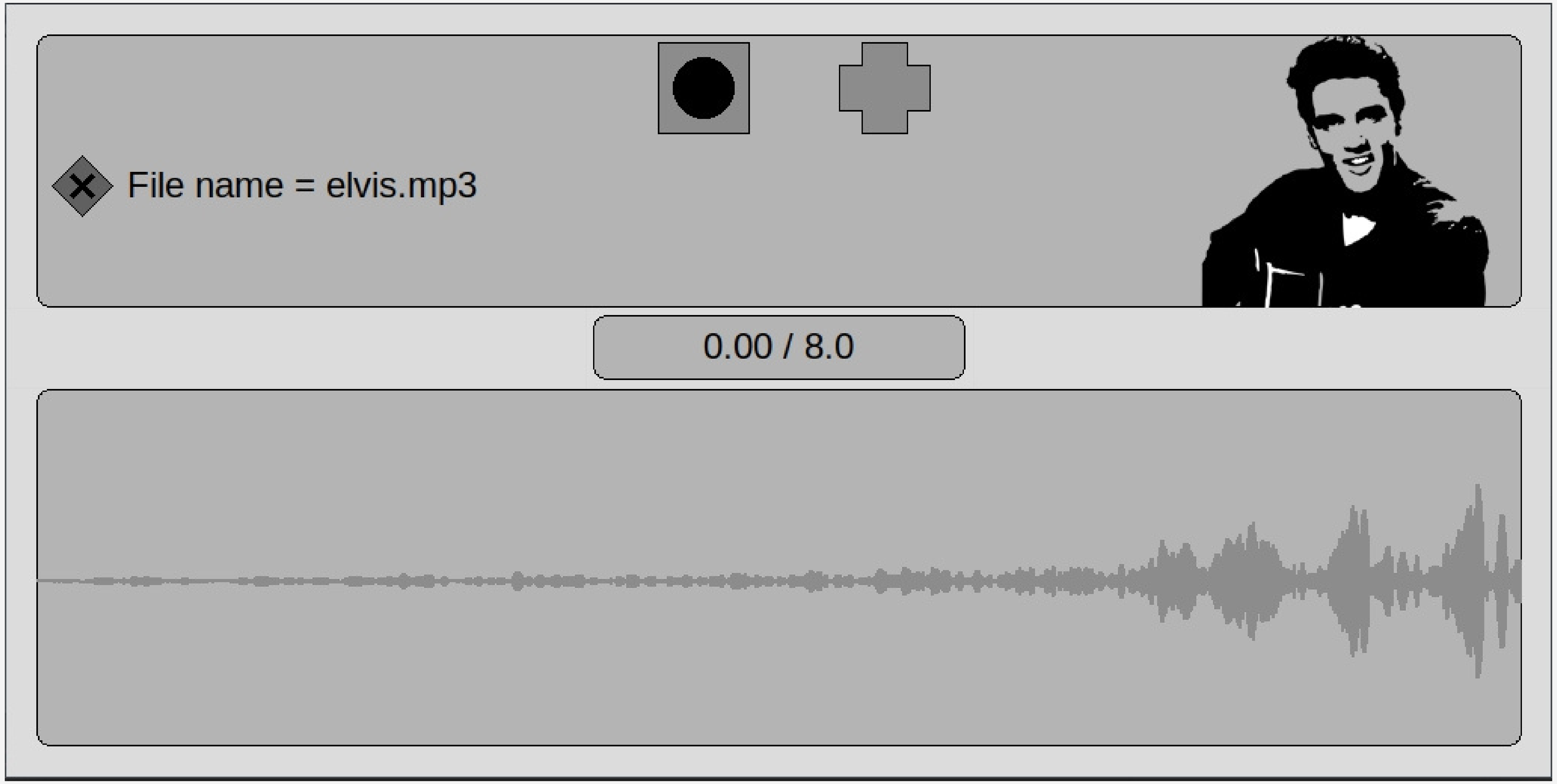
BPM & STEPS

INTRUMENTS

PLAY AND STOP

CONTROL AGENT

RECORDER & PLAYER

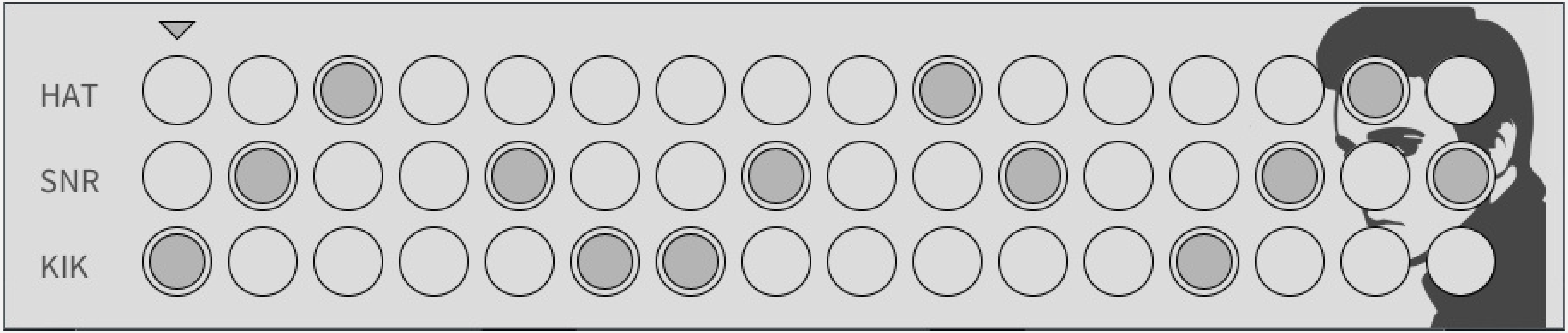


File name = elvis.mp3

0.00 / 8.0

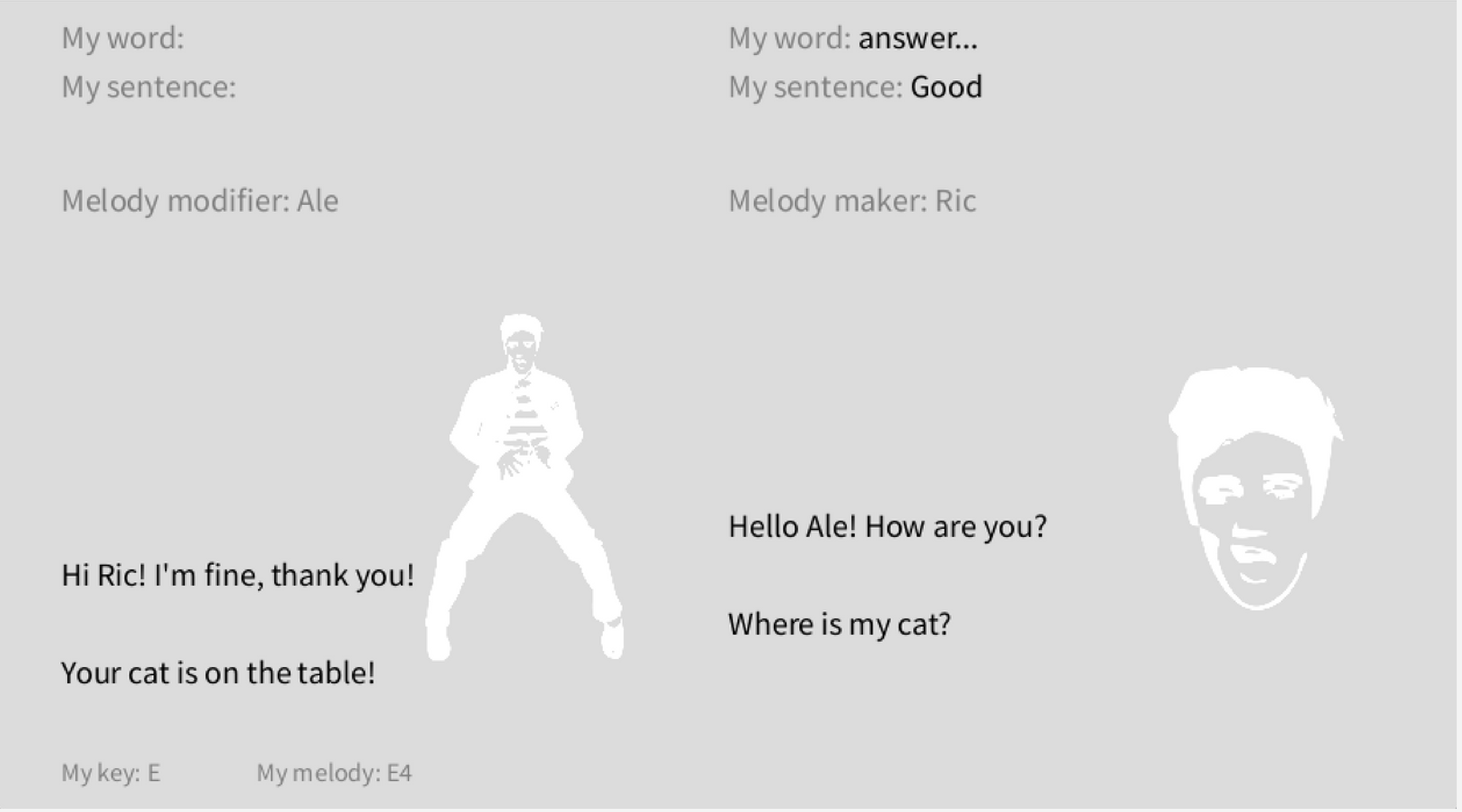
E.L.V.I.S.
Creative Programming and Computing

DRUM MACHINE



	▽															
HAT	○	○	●	○	○	○	○	○	○	●	○	○	○	○	●	○
SNR	○	●	○	○	●	○	○	●	○	○	●	○	○	●	○	●
KIK	●	○	○	○	○	●	●	○	○	○	○	○	●	○	○	○

MELODY CHAT



My word:

My sentence:

Melody modifier: Ale

Hi Ric! I'm fine, thank you!

Your cat is on the table!

My key: E

My melody: E4

My word: answer...

My sentence: Good

Melody maker: Ric

Hello Ale! How are you?

Where is my cat?

Firebase

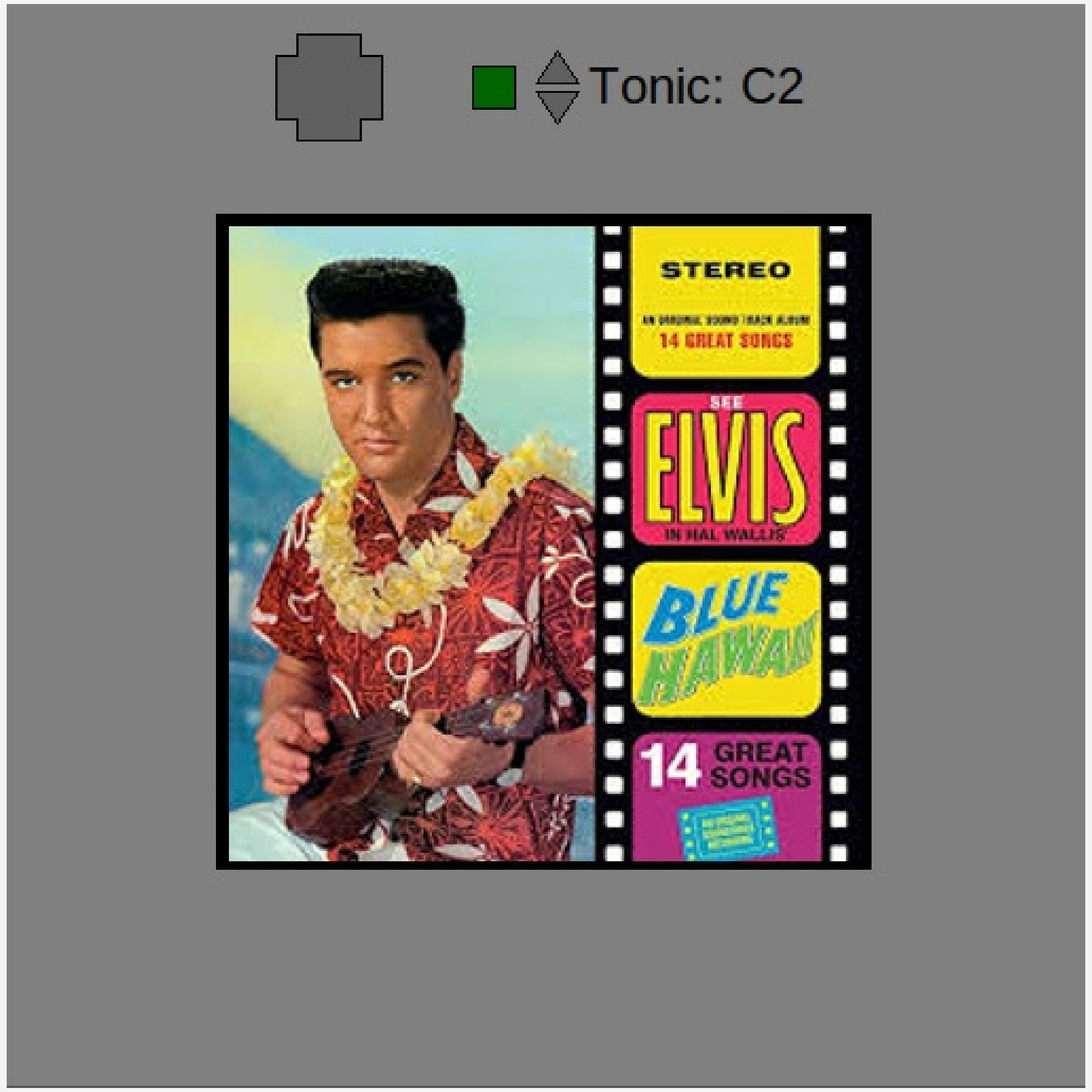
TABLE:

- char1
- char2
- username1
- username2
- taken1
- taken2

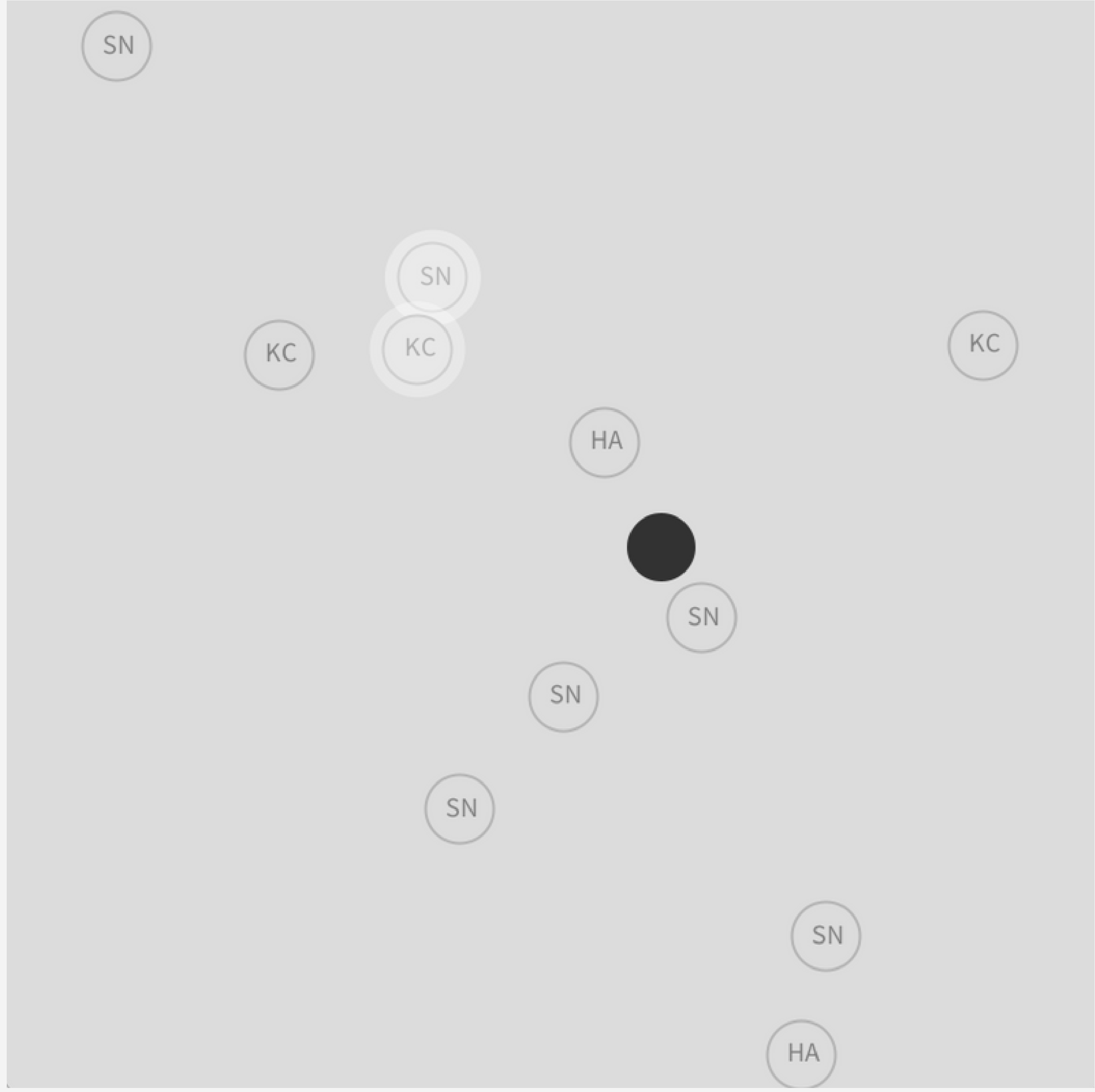
E.L.V.I.S.

Creative Programming and Computing

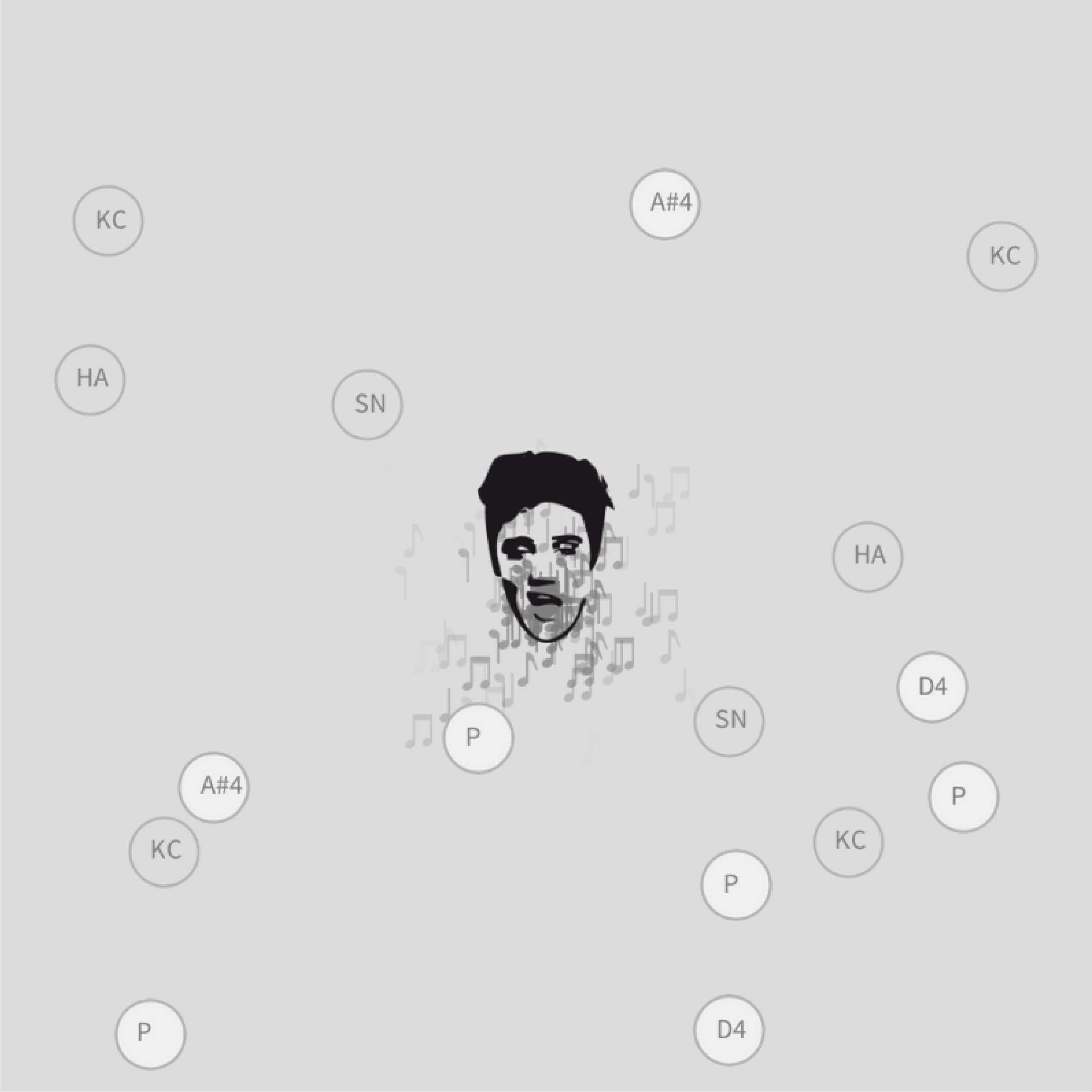
IMAGE SONIFICATOR



Tonic: C2

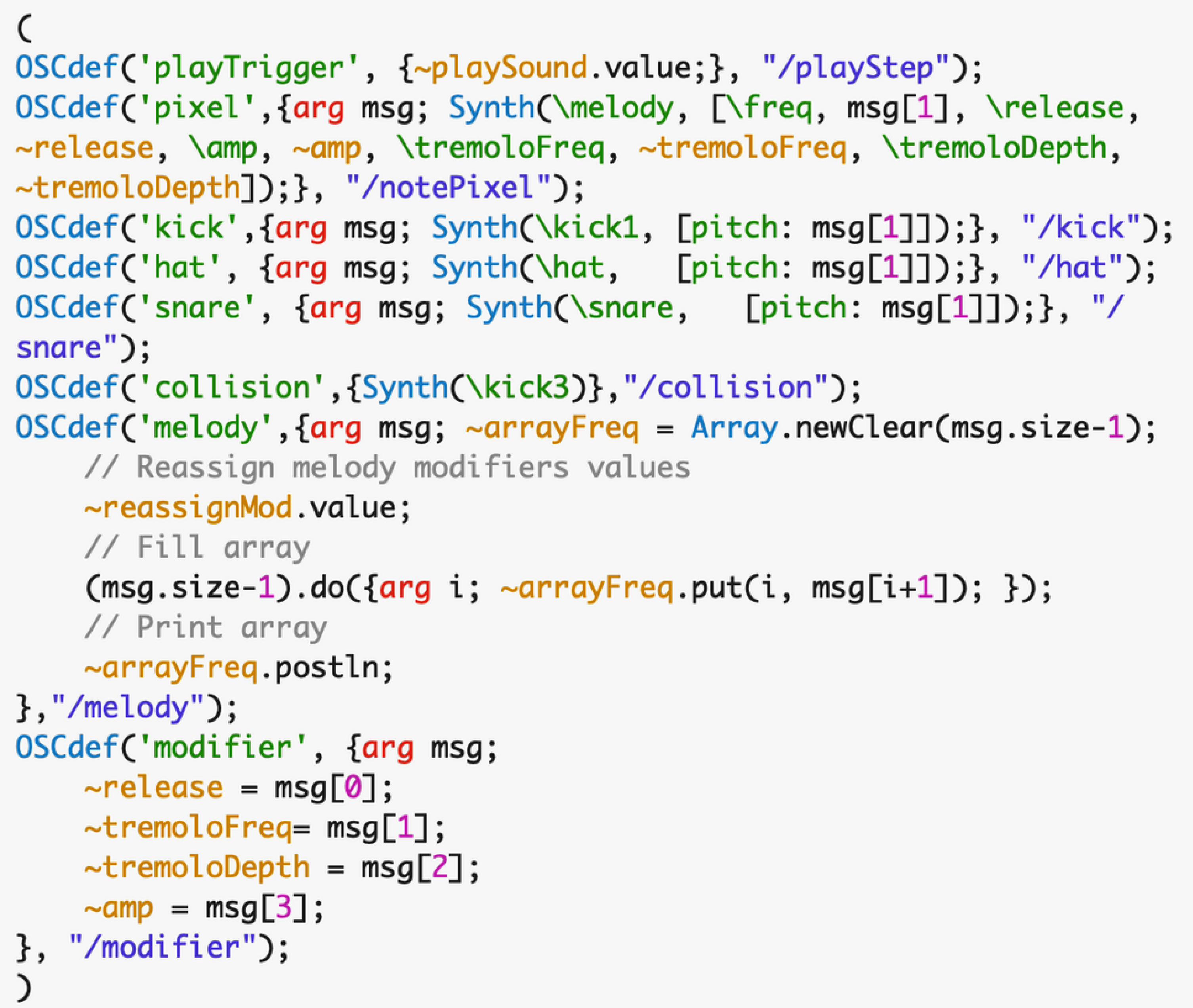


LAYER INTERACTION

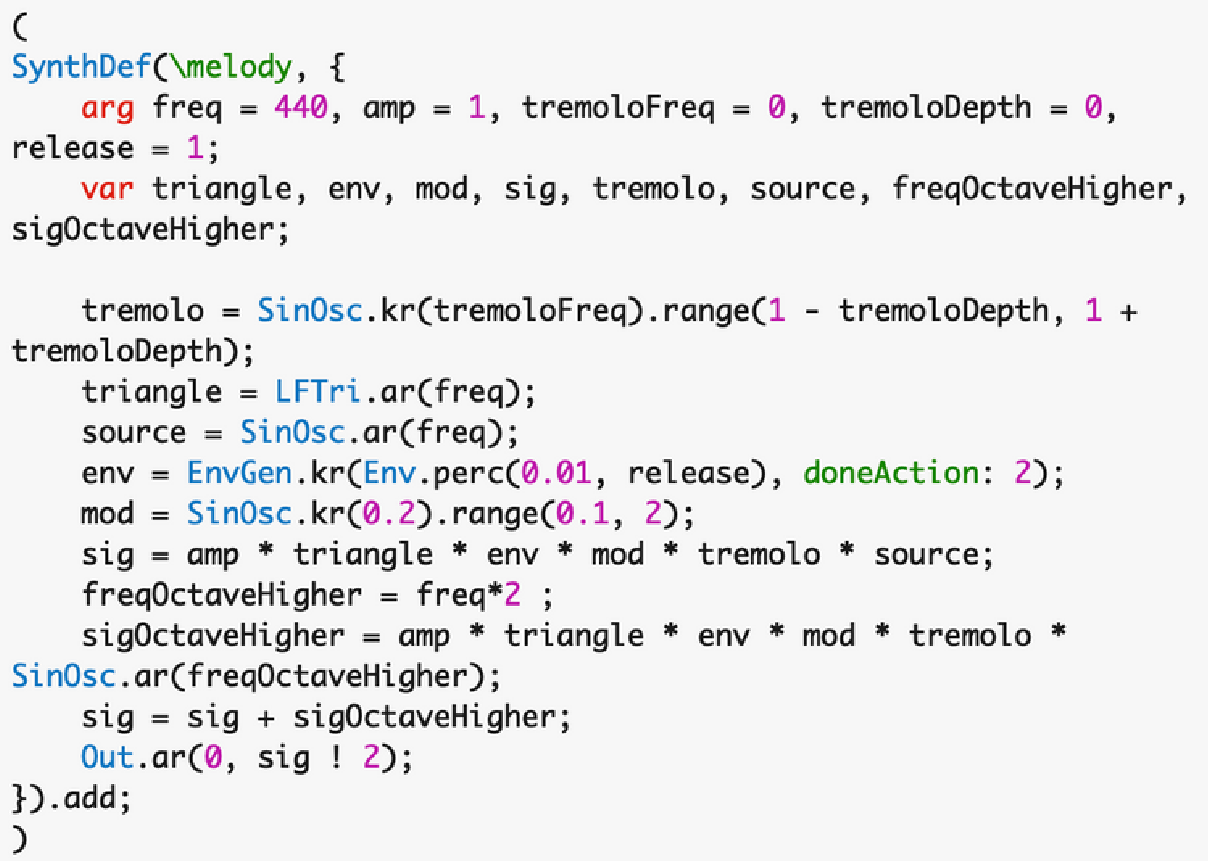


- Drum Machine Balls
- Melody Cht Balls
- Collisions
- Control Agent (Elvis face)
- Image Sonificator (black hole)

SOUNDS

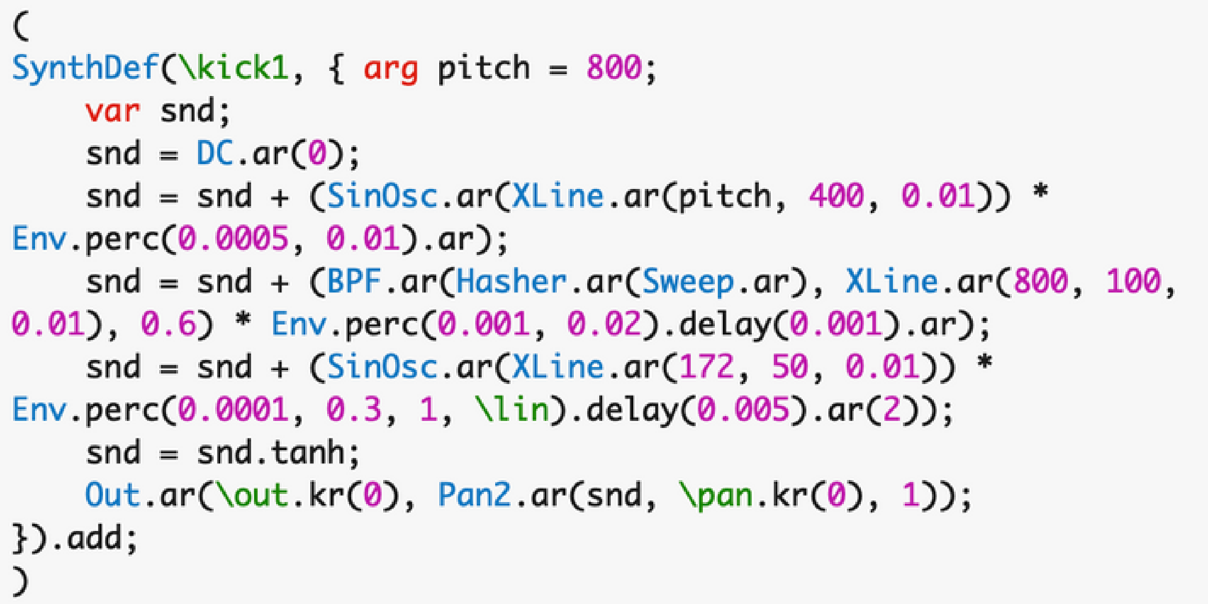


```
(
OSCdef('playTrigger', {~playSound.value;}, "/playStep");
OSCdef('pixel',{arg msg; Synth(\melody, [\freq, msg[1], \release,
~release, \amp, ~amp, \tremoloFreq, ~tremoloFreq, \tremoloDepth,
~tremoloDepth]);}, "/notePixel");
OSCdef('kick',{arg msg; Synth(\kick1, [pitch: msg[1]]);}, "/kick");
OSCdef('hat', {arg msg; Synth(\hat,   [pitch: msg[1]]);}, "/hat");
OSCdef('snare', {arg msg; Synth(\snare,   [pitch: msg[1]]);}, "/
snare");
OSCdef('collision',{Synth(\kick3)},"/collision");
OSCdef('melody',{arg msg; ~arrayFreq = Array.newClear(msg.size-1);
    // Reassign melody modifiers values
    ~reassignMod.value;
    // Fill array
    (msg.size-1).do({arg i; ~arrayFreq.put(i, msg[i+1]); });
    // Print array
    ~arrayFreq.postln;
},"/melody");
OSCdef('modifier', {arg msg;
    ~release = msg[0];
    ~tremoloFreq= msg[1];
    ~tremoloDepth = msg[2];
    ~amp = msg[3];
}, "/modifier");
)
```



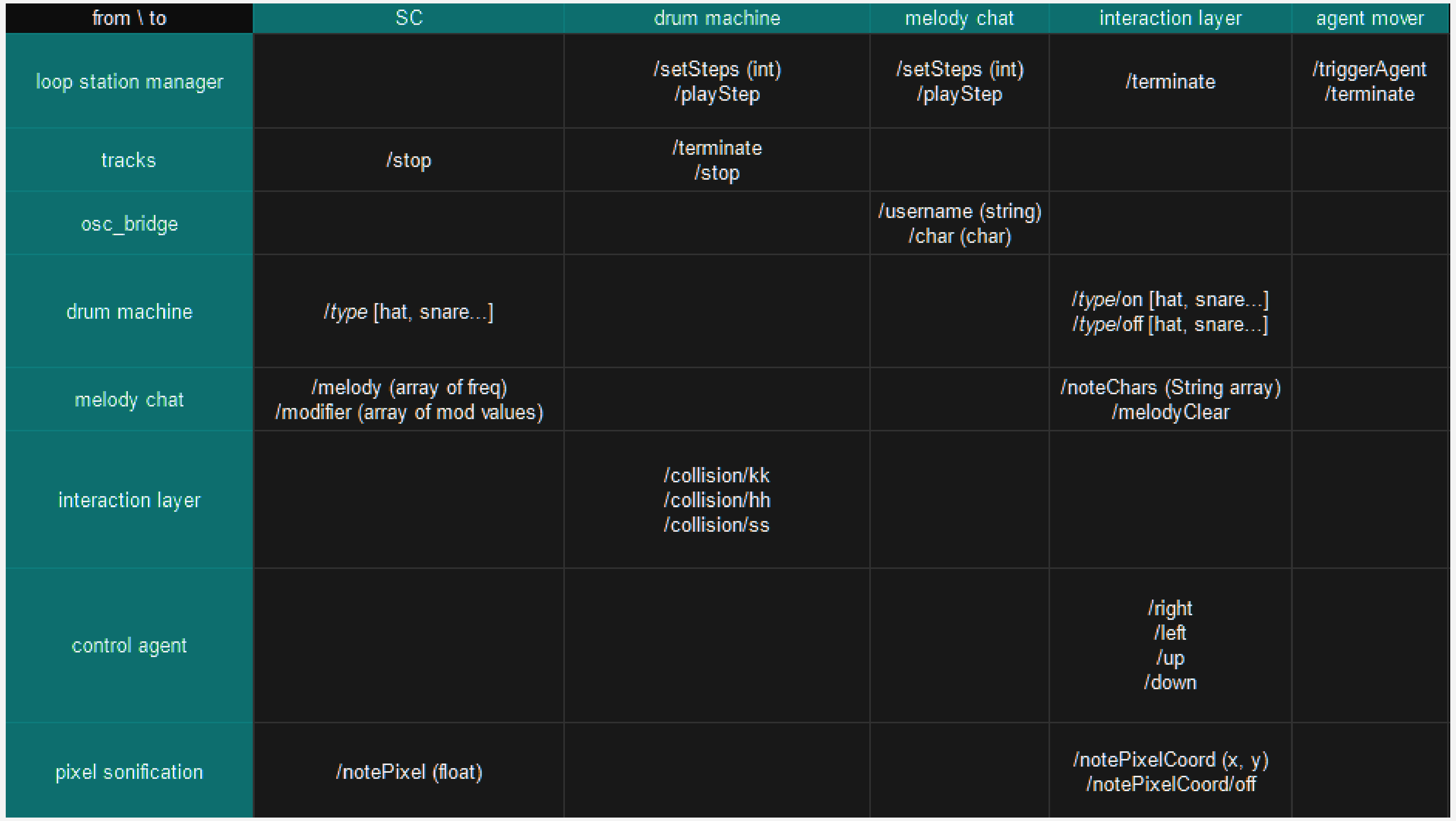
```
(
SynthDef(\melody, {
    arg freq = 440, amp = 1, tremoloFreq = 0, tremoloDepth = 0,
release = 1;
    var triangle, env, mod, sig, tremolo, source, freqOctaveHigher,
sigOctaveHigher;

    tremolo = SinOsc.kr(tremoloFreq).range(1 - tremoloDepth, 1 +
tremoloDepth);
    triangle = LFTri.ar(freq);
    source = SinOsc.ar(freq);
    env = EnvGen.kr(Env.perc(0.01, release), doneAction: 2);
    mod = SinOsc.kr(0.2).range(0.1, 2);
    sig = amp * triangle * env * mod * tremolo * source;
    freqOctaveHigher = freq*2 ;
    sigOctaveHigher = amp * triangle * env * mod * tremolo *
SinOsc.ar(freqOctaveHigher);
    sig = sig + sigOctaveHigher;
    Out.ar(0, sig ! 2);
}).add;
)
```



```
(
SynthDef(\kick1, { arg pitch = 800;
    var snd;
    snd = DC.ar(0);
    snd = snd + (SinOsc.ar(XLine.ar(pitch, 400, 0.01)) *
Env.perc(0.0005, 0.01).ar);
    snd = snd + (BPF.ar(Hasher.ar(Sweep.ar), XLine.ar(800, 100,
0.01), 0.6) * Env.perc(0.001, 0.02).delay(0.001).ar);
    snd = snd + (SinOsc.ar(XLine.ar(172, 50, 0.01)) *
Env.perc(0.0001, 0.3, 1, \lin).delay(0.005).ar(2));
    snd = snd.tanh;
    Out.ar(\out.kr(0), Pan2.ar(snd, \pan.kr(0), 1));
}).add;
)
```

OSC STRUCTURES



from \ to	SC	drum machine	melody chat	interaction layer	agent mover
loop station manager		/setSteps (int) /playStep	/setSteps (int) /playStep	/terminate	/triggerAgent /terminate
tracks	/stop	/terminate /stop			
osc_bridge			/username (string) /char (char)		
drum machine	/*type* [hat, snare...]			/*type*/on [hat, snare...] /*type*/off [hat, snare...]	
melody chat	/melody (array of freq) /modifier (array of mod values)			/noteChars (String array) /melodyClear	
interaction layer		/collision/kk /collision/hh /collision/ss			
control agent				/right /left /up /down	
pixel sonification	/notePixel (float)			/notePixelCoord (x, y) /notePixelCoord/off	

FUTURE IMPROVEMENTS

- Other Instruments
- Method of interaction user-machine
- Other graphic-sound layer
- Real-time Interaction

THANK'S FOR ATTENTION!

GITHUB	https://github.com/Rhapsodizer/CC_Project
OTHER LINKS	YouTube demo -> soon online

CONTACTS

Attolini Silvio	silvio.attolini@mail.polimi.it
Gorni Alessandro	alessandro.gorni@mail.polimi.it
Martinelli Riccardo	riccardo1.martinelli@mail.polimi.it

E.L.V.I.S.
Creative Programming and Computing