HOME

"The MC Skooghall Sound n Sight Palette"

by Ashley Sagar

An interactive Graphical User Interface for "The MC Skooghall Sound n Sight Palette"

Introduction

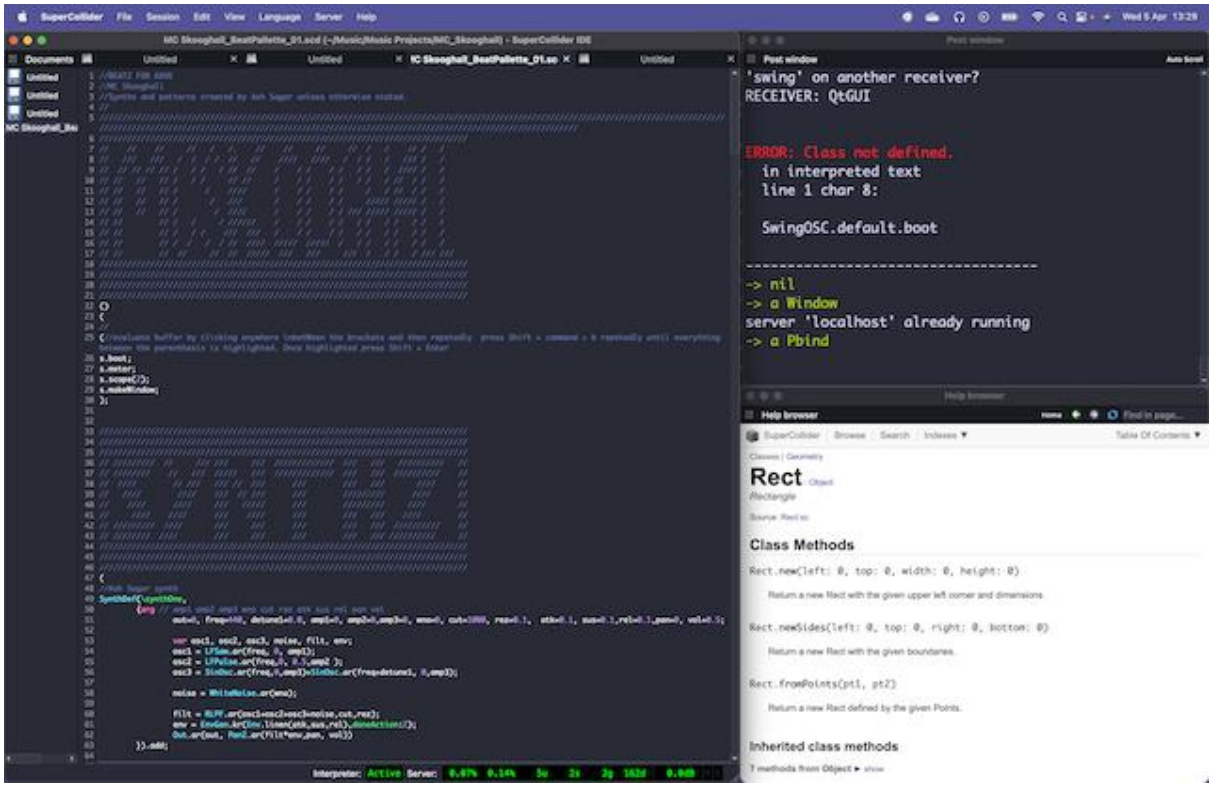
The MC Skooghall Sound n Sight Palette is aimed at allowing me to create an interactive GUI for the Swedish artist MC Skooghall.
Creating an application for live musical performance and visualizations.

Concept and Background Research

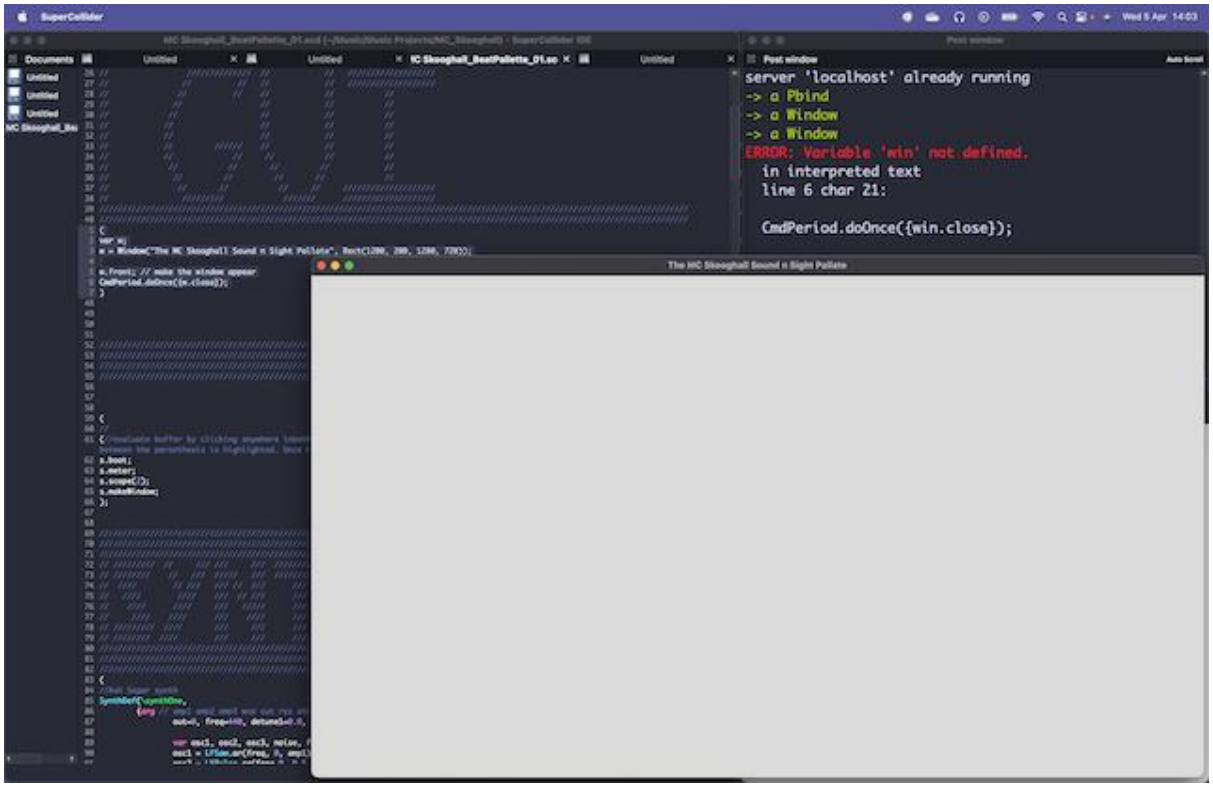
The MC Skooghall project is the brainchild of Swedish based noise artist Dave Proctor. MC Skooghall is an MC creating "lazy hip hop".
My part in the collaboration is to create the beats for MC Skooghall using the Open Source Audio Programming software SuperCollider.
I chose to use SuperCollider platform as the main sound and sequencing engine because of its power as a music producing tool.
Many experimental musicians and sound artists use SuperCollider for its ability to create music that doesn't necessarily sound like any other artist, it's lightweight use on the computer processor due to limited gui options and also the fact that the software is Open Source and free to use. The downside of this is the renowned, steep learning curve.
I first came across SuperCollider after discovering the music of Cylob (Chris Jeffs) around 2006 and was fascinated on how his music sounded like nothing else and found that he created the CMS (Cylob Music System) Environment whilst using SuperCollider. I thus began a very slow journey of programming music in SuperCollider.
The MC Skooghall Sound n Sight Palette would use the After discussions we have decided that since Dave is not a user of SuperCollider and has limited capability in programming, I would go ahead and attempt to make a GUI for him to use whilst performing live, thus making his performance less stressful and then he can just focus on triggering the loops.
The preliminary ideas are that he has a way to trigger pre-made loops with the possibility to also display visualizations for the duration of the loop being played.
Other ideas we explored are to possibly bundle the software as an app and also having some sort of access to it via a hand held device such as an ipad.

Technical Implementation

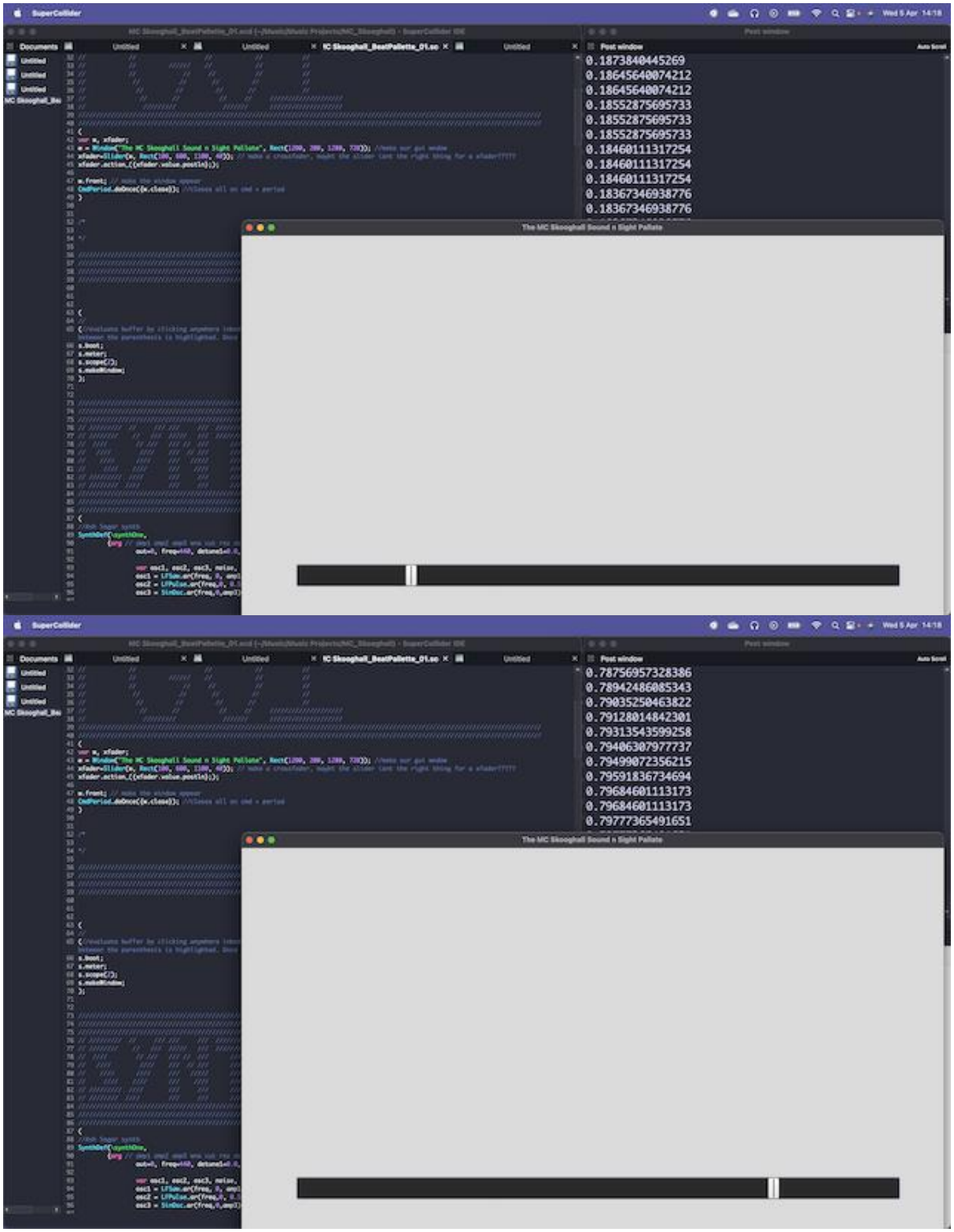
The basic SuperCollider script which includes all the synthdefs and patterns for the beat palette



The first beginnings of adding a window gui into SuperCollider



Adding a crossfader slider

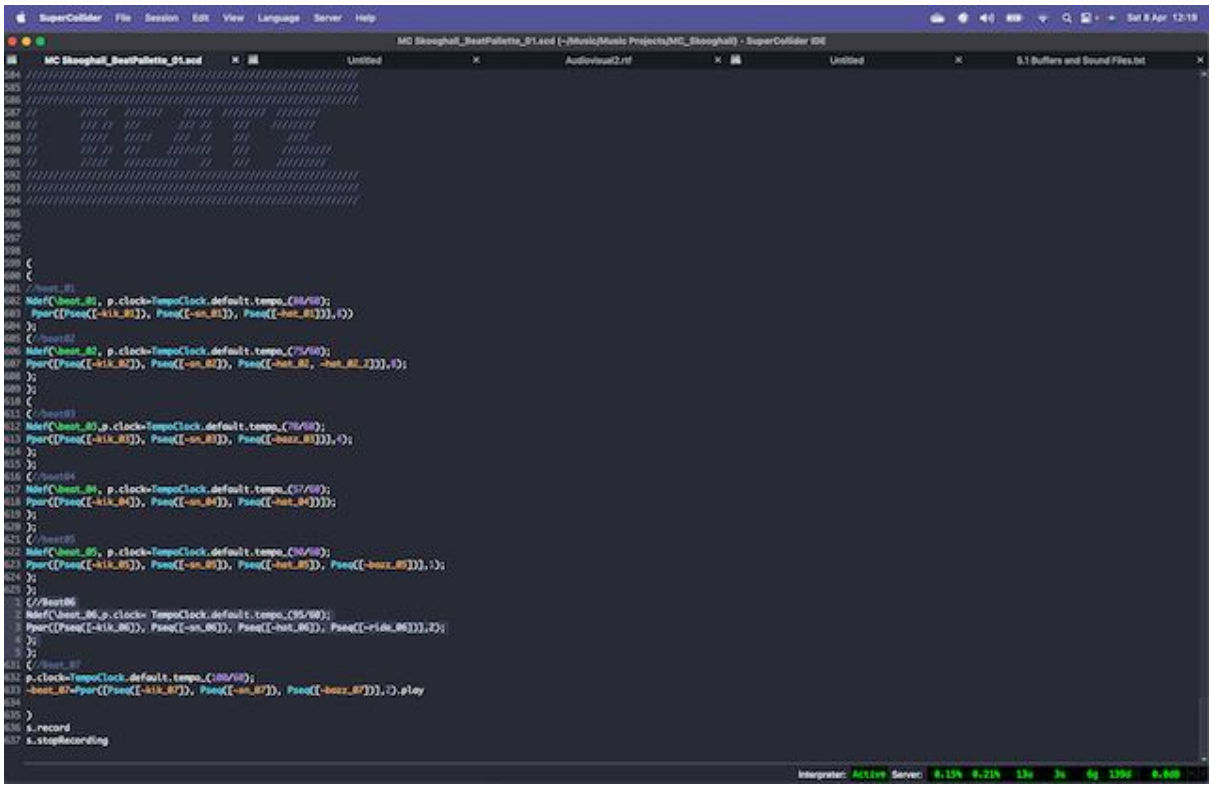


What technical elements are involved?

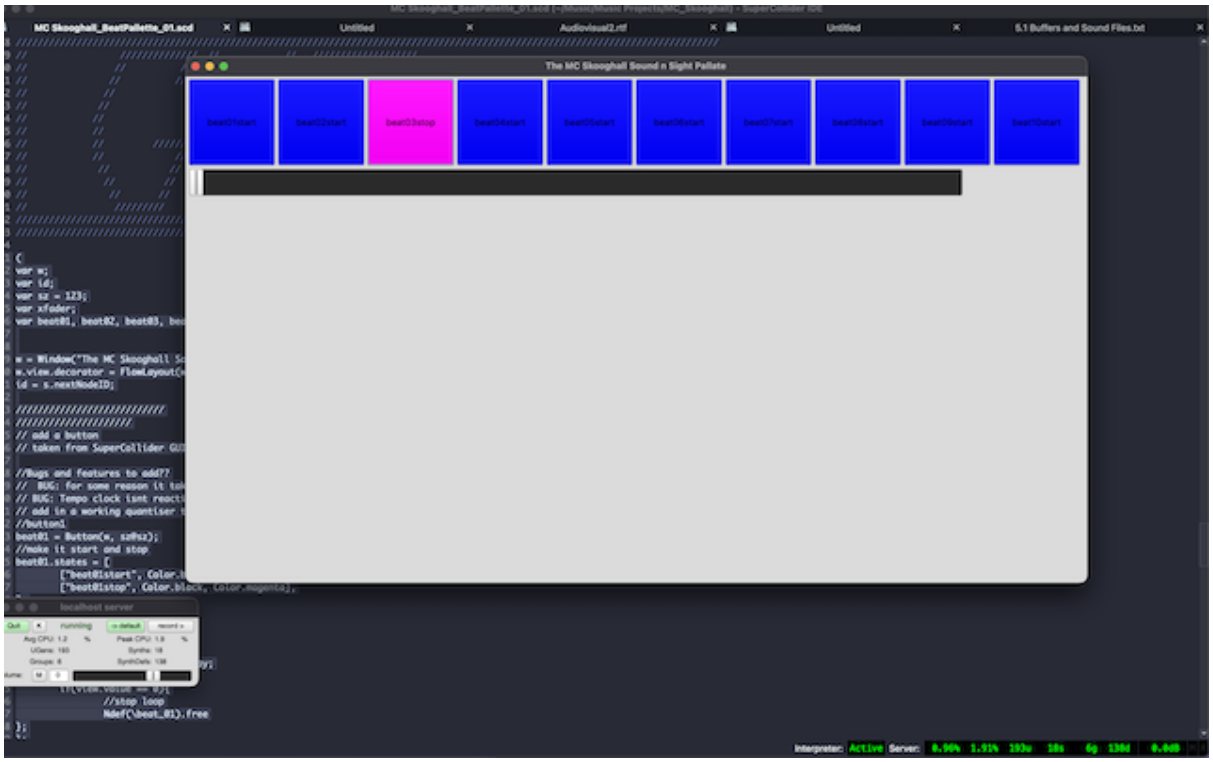
The process of this project has been a steady one. As a sometimes user of SuperCollider I didn't really have much in the way of problems whilst creating the music for this project, as that is the part of the software I'm familiar with. Work slowed down when it came to creating functionality within the gui, such as triggering loops with buttons and making the visualization windows open and close on the same button press. After figuring this out I moves on to trying to control the app via TouchOSC. Here is where I began running into real problems. I did get TouchOSC talking to SuperCollider

You can find my ongoing TODO org file documenting my "daily" process with working on the app

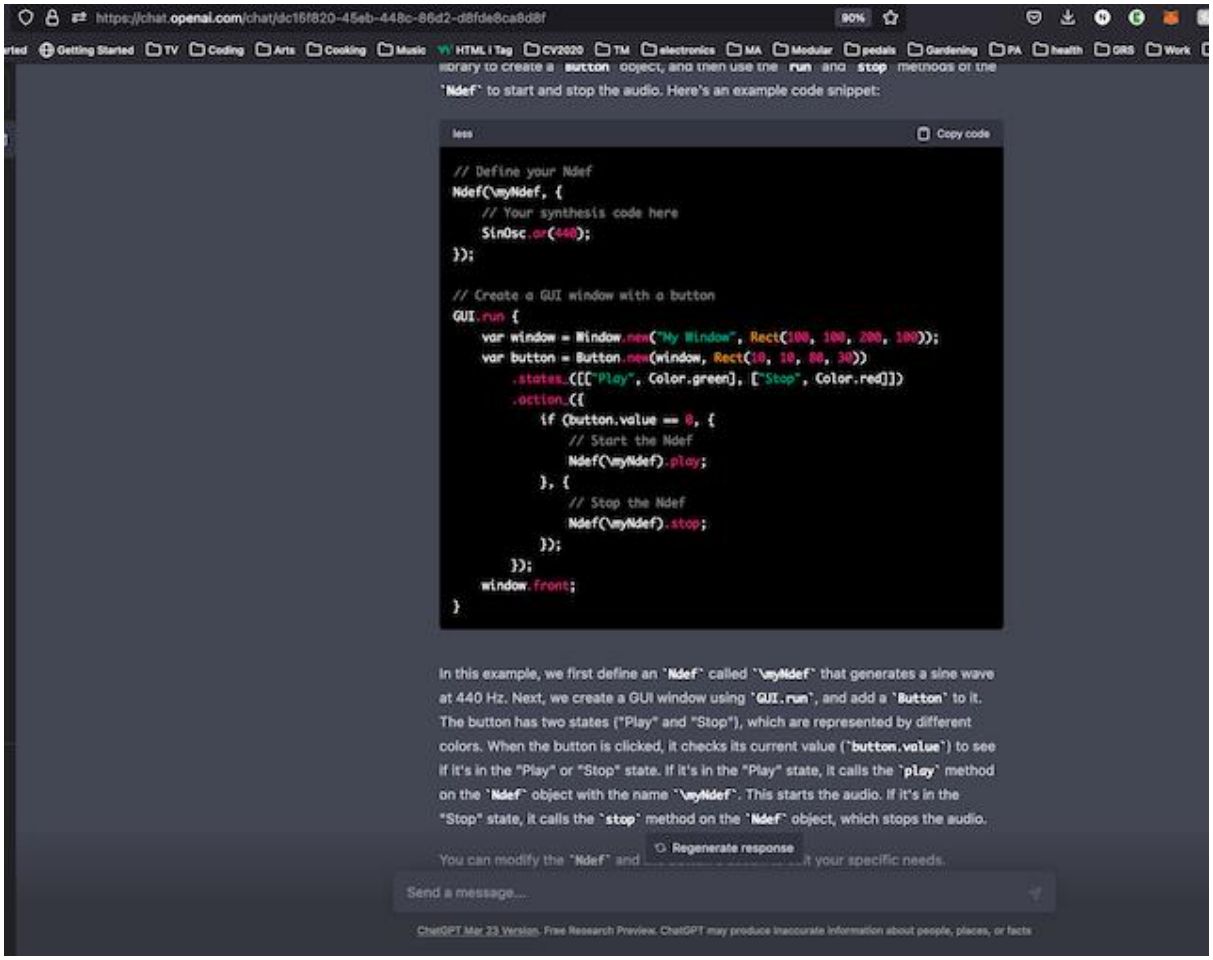
the beats section which needs to be added to buttons



adding buttons to the gui



I found that I had trouble with triggering the loops so I had to change the loop section into an Ndef to make it trigger.
I became stuck with this so I had to ask Chat GPT for a little help in hat to write for actioning the buttons



```
// Define your Ndef
Ndef(\myNdef, {
    // Your synthesis code here
    SinOsc.ar(440);
});

// Create a GUI window with a button
GUI.run {
    var window = Window.new("My Window", Rect(100, 100, 200, 100));
    var button = Button.new(window, Rect(10, 10, 80, 30))
        .states_([["Play", Color.green], ["Stop", Color.red]])
        .action_({
            if (button.value == 0, {
                // Start the Ndef
                Ndef(\myNdef).play;
            }, {
                // Stop the Ndef
                Ndef(\myNdef).stop;
            });
        });
    window.front;
}
```

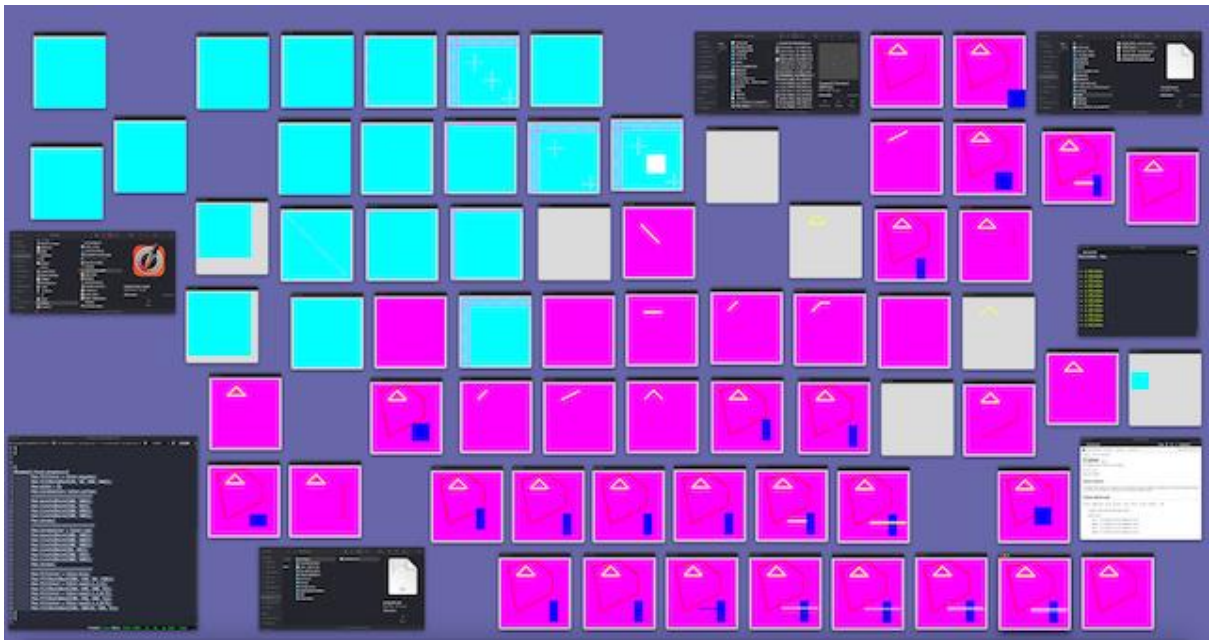
changing the code in SuperCollider to reflect this

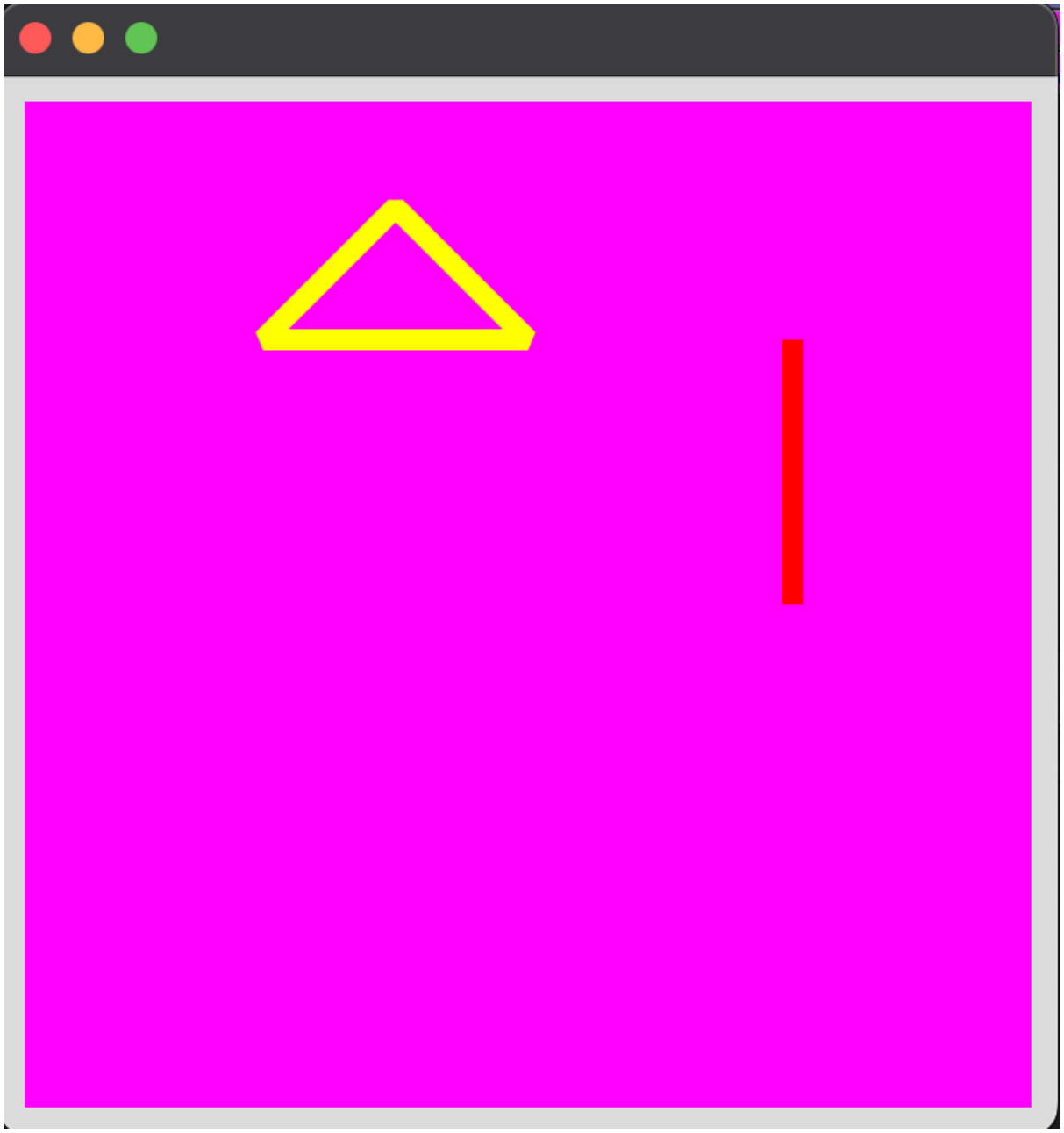
```
632
633 (//Beat_07 old code
634 p.clock=TempoClock.default.tempo_(100/60);
635 ~beat_07=Ppar([Pseq([~kik_07]), Pseq([~sn_07]), Pseq([~bazz_07])],2).play
636 )
637
638 (//Beat_07 new code
639 Ndef(\beat_07, p.clock=TempoClock.default.tempo_(100/60);
640 Ppar([Pseq([~kik_07]), Pseq([~sn_07]), Pseq([~bazz_07])],2);
641 );
642
643
```

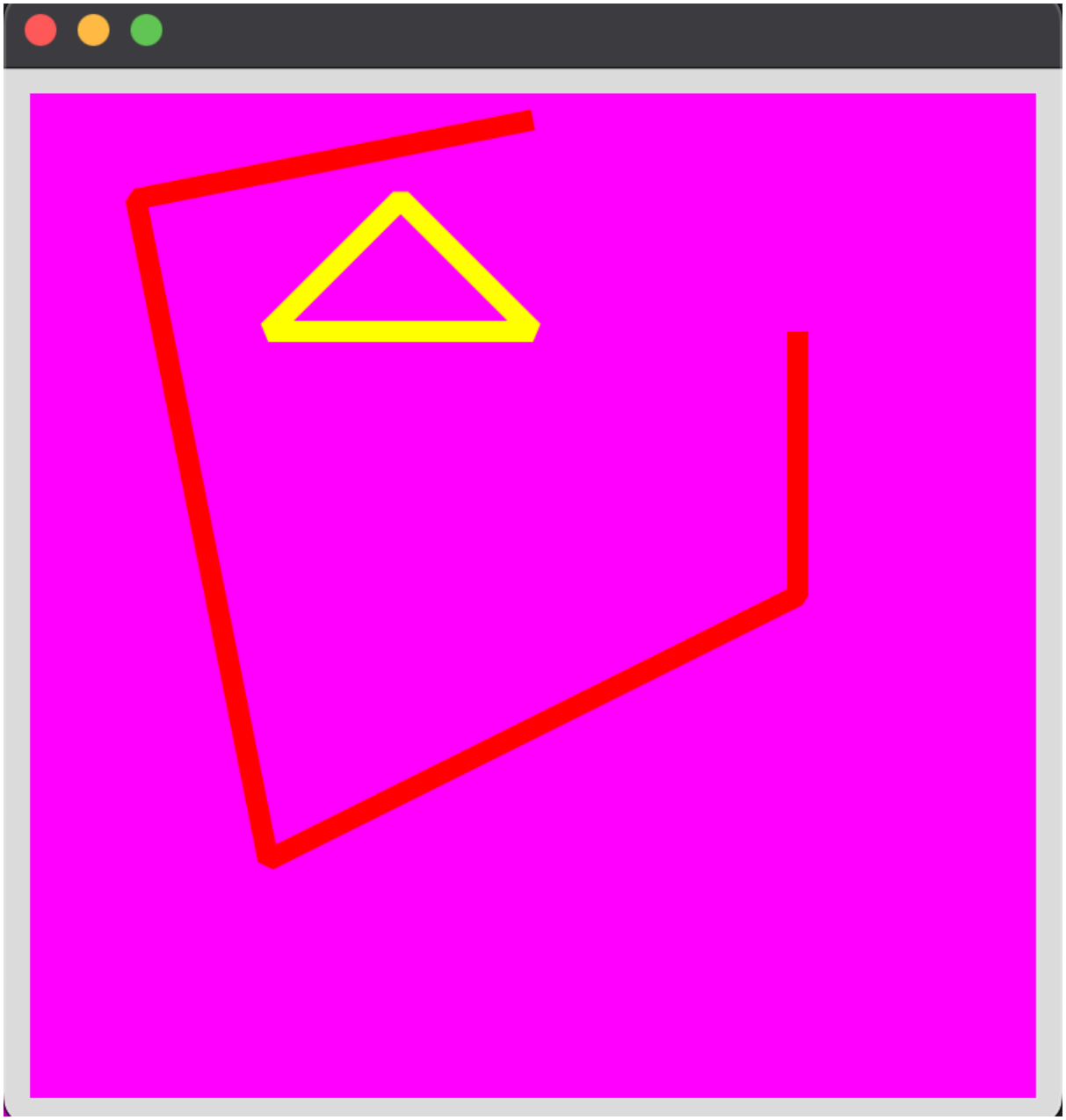
I still had problems with the triggering of the tempo for different loops. When triggering a loop, the tempo would remain the same, instead of changing for the called loop. After some movement of code in various places, I moved the p.TempoClock code into the if statement for the button inside the GUI code as shown below

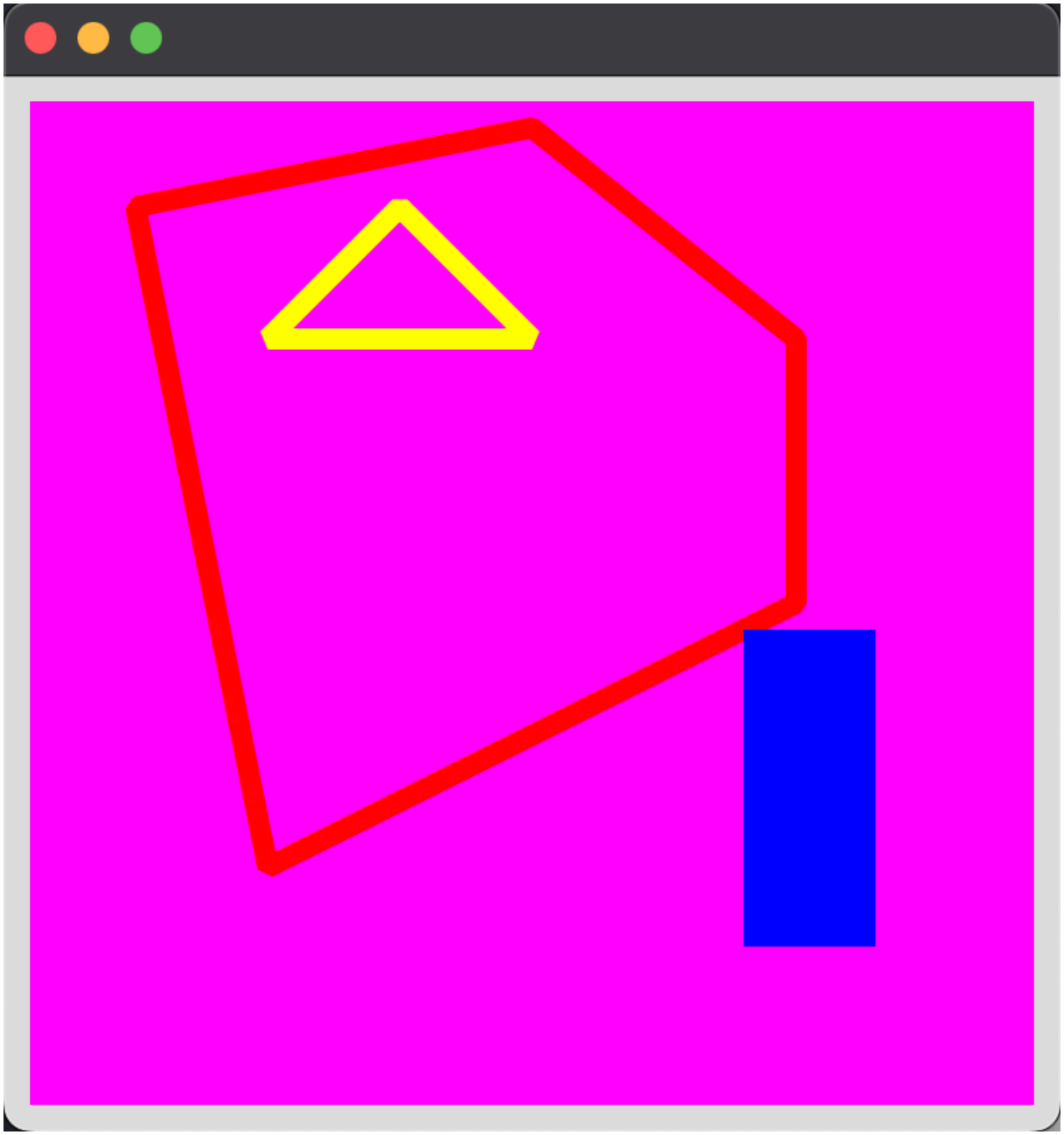
```
//button02
beat02 = Button(w, sz@sz);
//make it start/stop
beat02.states = [
        ["beat02start", Color.black, Color.blue],
        ["beat02stop", Color.black, Color.magenta],
];
// assign action to the buttons
beat02.action = {|view|
        if(view.value == 1){
                //start loop
                Ndef(\beat_02).play;
                p.clock=TempoClock.default.tempo_(75/60);

        };
        if(view.value == 0){
                //stop loop
                Ndef(\beat_02).free
        };
};

//button03
```

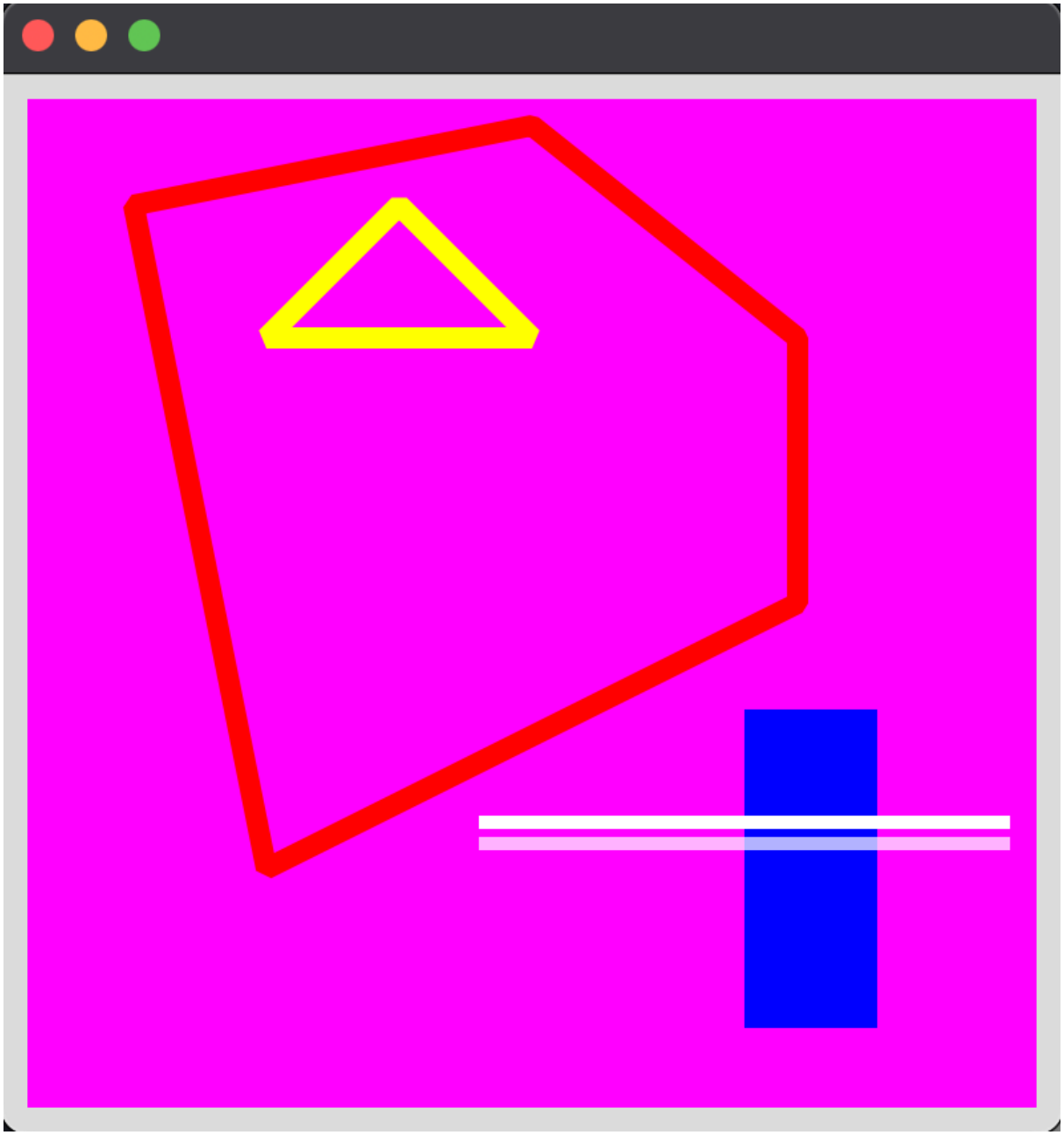
Here is the start to teach myself and experiment with making visuals inside of SuperCollider using Oloffson's Audio-visuals in SuperCollider tutorials I experimented with basic drawings

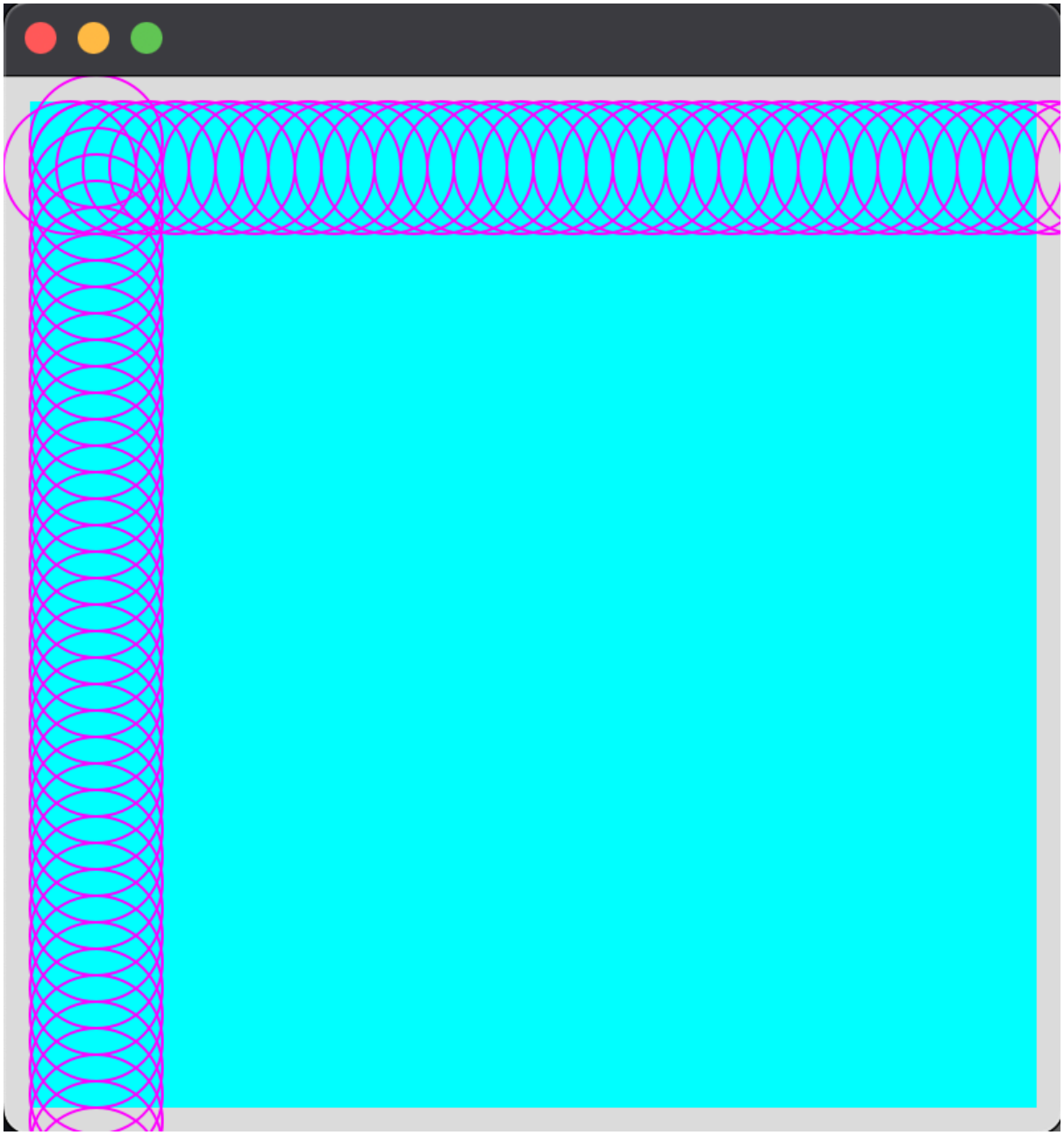


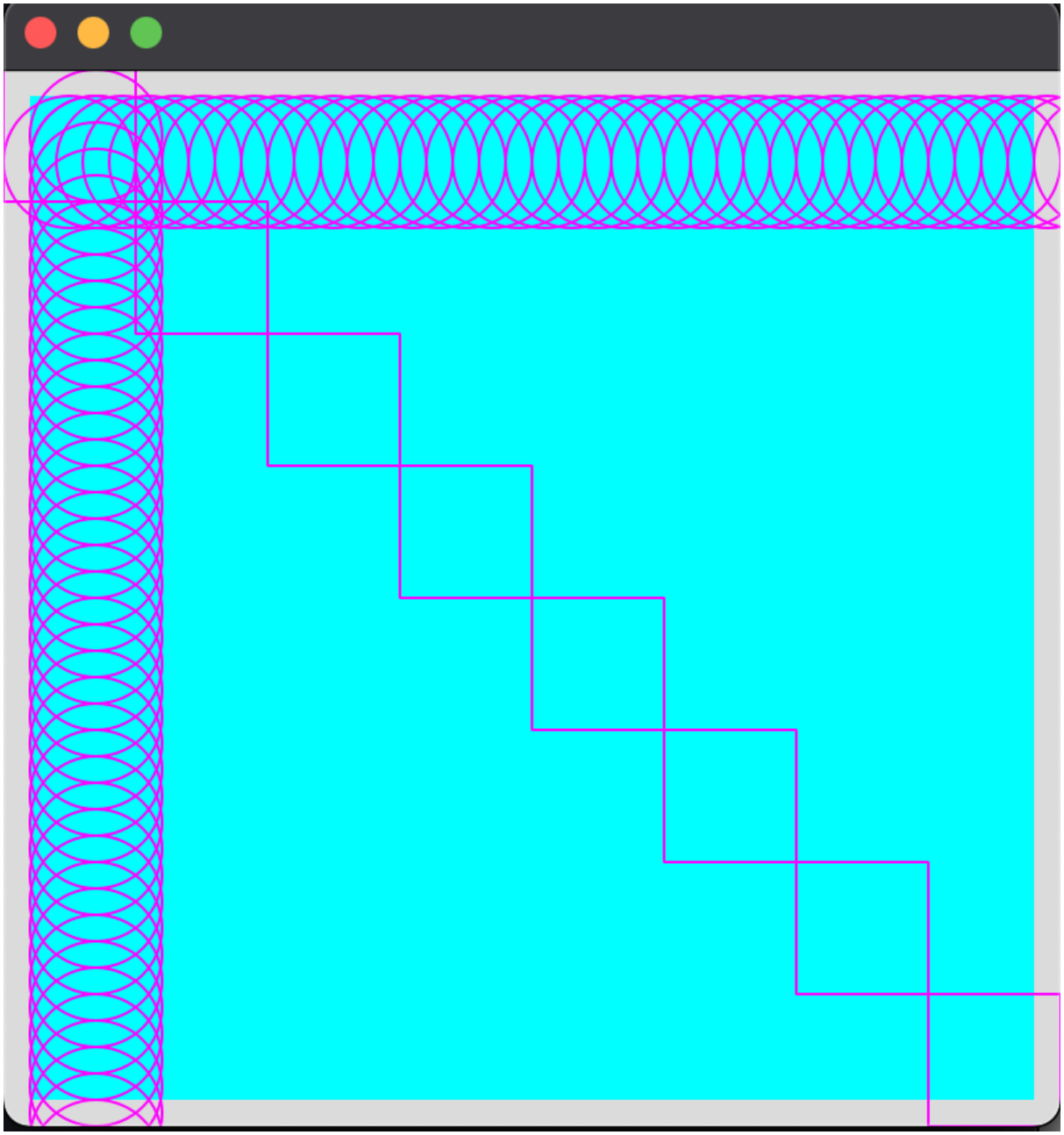


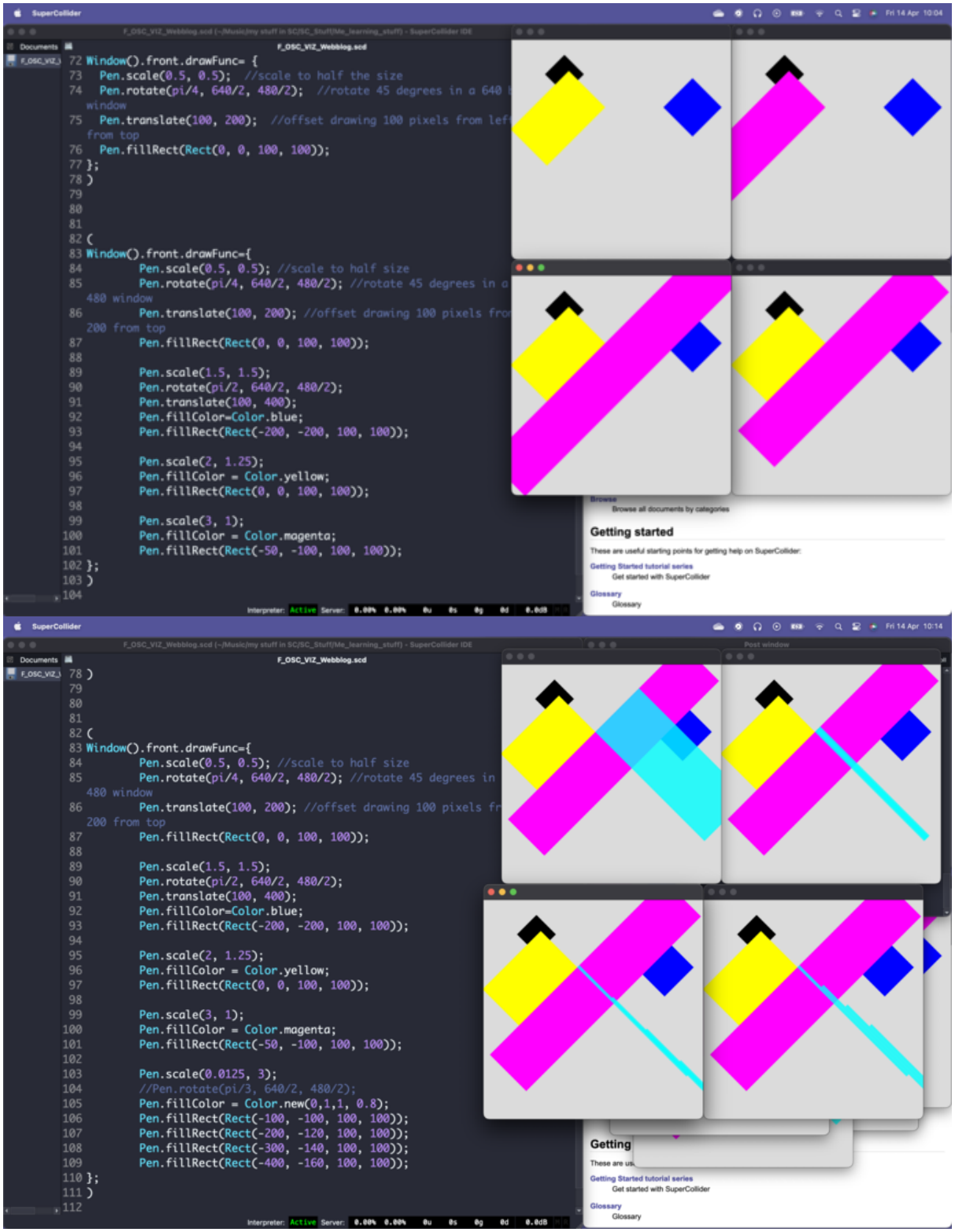












I'm now at the point of making a visualizer window appear and close on a button press. I do have an issue about having the window open on a different display though.
I made some video documentation for this which you can find below

Here I've begun to add that the window opens in fullscreen on a separate display. Allowing the gui to be used on a laptop and the visualisation to be projected

Reflection and Future Development

So far with working on this project I felt that everything didn't go so bad. I refreshed my knowledge of SuperCollider with running through some tutorial files each day in both the audio and visualization parts of it. Unfortunately, SuperCollider does not have much in the way of documentation and tutorials for visuals and GUI as many people use the application for experimental composition. I did work through Fredrik Oloffson's tutorials and workshop documents whilst working on this project which will assist me in the ongoing work with this project.
As far as I can see the project is a long way from being complete and is only in a beta stage at the moment. Many things need to be completed such as the TouchOSC button triggering correctly and adding more visualisations to the app as well as an ongoing addition of loops and visualizations for as long as this application works for the project in question. Features to be added include and could include
- random beat generator
- audio metering to the gui
- scope to the gui
- tempo shift sliders like on a turntable
- crossfader to fade between tracks

References

- Technical ref ex. Include link where possible.
- Fredrik Olofsson AudioVisuals in SuperCollider
- conceptual ref example. Include link where possible.

  CMS Enviornment