STARTS Residency Public Report
Still

Natan Sinigaglia

I. INTRODUCTION

In the contemporary new-media art world and especially in the creative coding domain, the artist is not just someone that takes a digital tool and creates something with it. Quite often he also builds the tool and the technology that later will be used to create the artwork.
The residency "Still" has been a collaboration between me and the tech studio vvvv.
It started in May 2019 and ended in January 2020.
During this period, I stayed at vvvv studio, having full support from the development team.
Using the vvvv programming environment I created a collection of new software libraries; these libraries, now available as .NET open source packages, constituted a solid framework for the creation of the performance "Still".

II. ARTWORK

Still is an audio-visual work performed by electro-acoustic instruments, dancer and custom software. During the performance, these multiple components join together to generate real-time visuals projected in the space, together with an immersive soundscape. The sound of the musical instrument gets analyzed by a custom real-time analysis software which extracts live data used for the content creation.

The performance turns around the theme of contemplation; in this sense the abstraction of the visuals represents the possibility to stay on the threshold between what's known and what is still undefined, in a delicate and powerful state that allows the forms of comprehension to mutate. By entering into this contemplative state, we change completely the perception of time, its depth, vastness, flavour. As the philosopher Byung-Chul Han says, *"the value we attach today to the vita activa is producing a crisis in our sense of time. As a remedy to this, we should revitalize the vita contemplativa, our capacity for reflection and contemplation. When life regains this capacity, this art of lingering, it gains in time and space, in duration and vastness".*

The goal of the residency was to create the live performance using the vvvv programming environment. During the residency the priority shifted from the creation of the artwork towards the creation of a collection of software tools that open new possibilities in multiple creative scenarios.

III. METHODOLOGY

During the period of the residency, from May 2019 to end December 2019, I had the possibility to permanently stay at the vvvv group studio, becoming a beta tester for all the new technologies developed by the team during the year.
A crucial first step was the confrontation with the vvvv team about my usual workflow in vvvv and all the problematics encountered during the last years in creating artworks of multiple natures.
We analyzed my previous methodologies and my current technical and artistic needs, considering them in the context of the new vvvv software currently in development and all the possibilities offered by this new technology.

Very soon I realized that this residency was a great opportunity to realize a vast new set of software tools not only to facilitate the creation of the “Still” performance, but really to redefine the way I work, finding new methodologies for the artistic research.

For this reason, most of the residency period was spent on the technical development of this collection of tools, leaving just a very small period at the end to sketch a prototype of the live show, a first experiment made with the new tools created.

The outcome of the technical research is this collection of software:

- Kairos
- Model-Runtime-Editor software architecture
- Interpolation Engine
- Elementa
- ParticlesSystem
- Generators
- AudioAnalysis library

Kairos

In the new-media performative field we face two different and contradicting aspects in the creative methodology.

On one end we might probably want to create a narrative, defining certain states of the show and organizing them throughout the performance. A timeline is an old and great concept-tool that gives us this possibility: we can set some “keyframes” to define different values of some parameter in different time positions; in this way we can sample time (interpolating between the keyframes) producing a continuous flux of data as output.

On the other end the material of a live performance could be something that gets generated during the action and therefore can’t be predefined. In this case a standard timeline tool doesn’t help, since it forces us to define both the position in time and the value of our keyframe, but we don’t really know the value because it will be generated live.

Let’s make an example of the problematic: we want to control a certain property of our visuals in different ways during the show. At the beginning we want to set a specific numeric value to this property. So, we might set this numeric value as the first keyframe of a timeline. After 20 seconds from the start we might want to tune this property differently. To do so we create a new keyframe at time 20s and set it with the desired value.

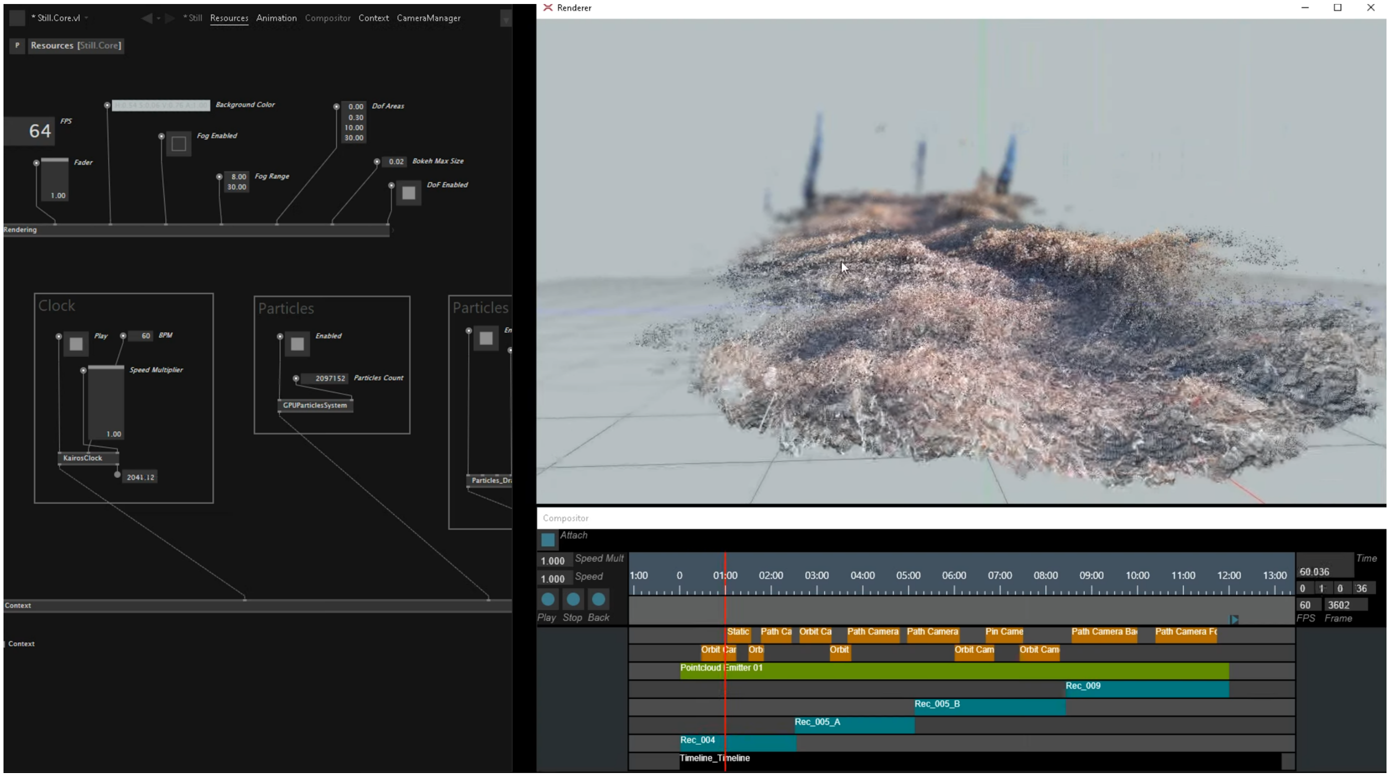
Now let’s consider this: at 30 seconds we want to control this property differently, we want it to be driven by the volume of one of the musical instruments on the stage (or by the hand position of a dancer). We can create a keyframe at time 30s but we don’t really know what to write in that keyframe as value! That’s something that will come live, that really depends on how the musician will play/improvise the music (or how the dancer will move in the space). It’s clear now how a standard timeline tool can’t handle this scenario.

This apparent contradiction between the need of fixing some states in time and the necessity of managing “live” states brought me to rethink the timeline tool (creating the Kairos timeline) and introduce the concept of “dynamic keyframe”.

A Dynamic keyframe is basically a placeholder for some live data; we can use it in our Kairos timeline, defining its position and duration. The value of this dynamic keyframe, which will be sampled to generate the output, is not contained in the keyframe itself, it will come from somewhere else, for example from a real-time audio analysis software.

The Kairos timeline is able to manage “dynamic keyframes” and “constant keyframes” (the classic ones, holding static-predefined values), solving the problem that we showed before in the example.

Thanks to the new version of vvvv that the studio is developing I was finally able to achieve something that I had in mind since years but I couldn’t realize due to some limitations of the previous versions of the vvvv language.

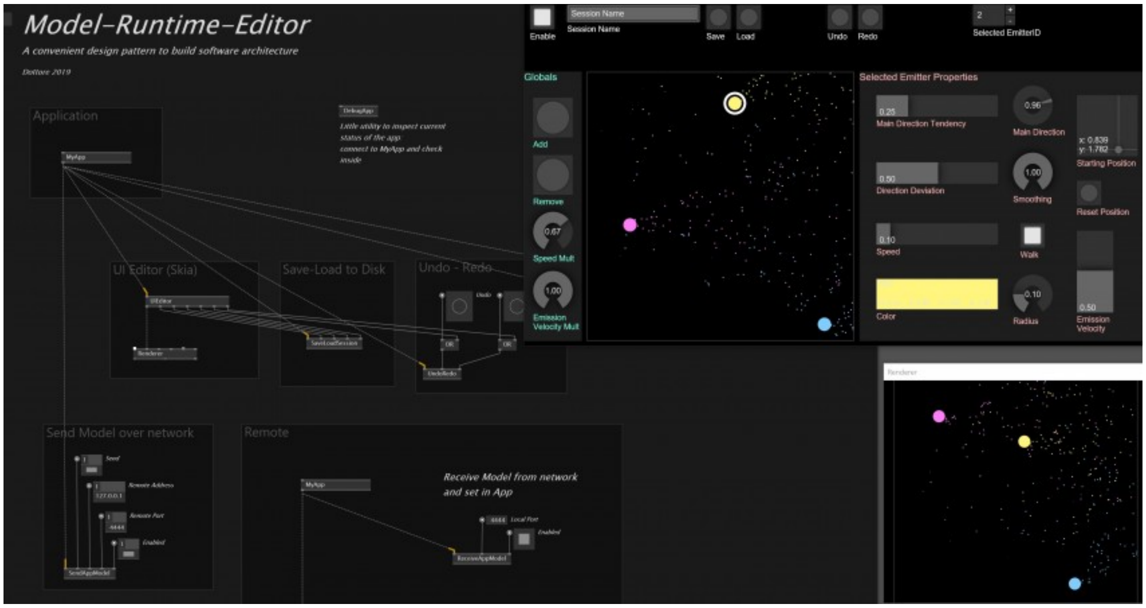


Model-Runtime-Editor software architecture implementation in vvvv

In order to create the Kairos software, I considered multiple programming pattern options for building the software architecture.

In the end I consolidated this research around the Model-Runtime-Editor programming pattern, creating a vvvv template for building software with it and producing a video tutorial for the vvvv community.

The material is available for download at this link: https://vvvv.org/contribution/model-runtime-editor-design-pattern

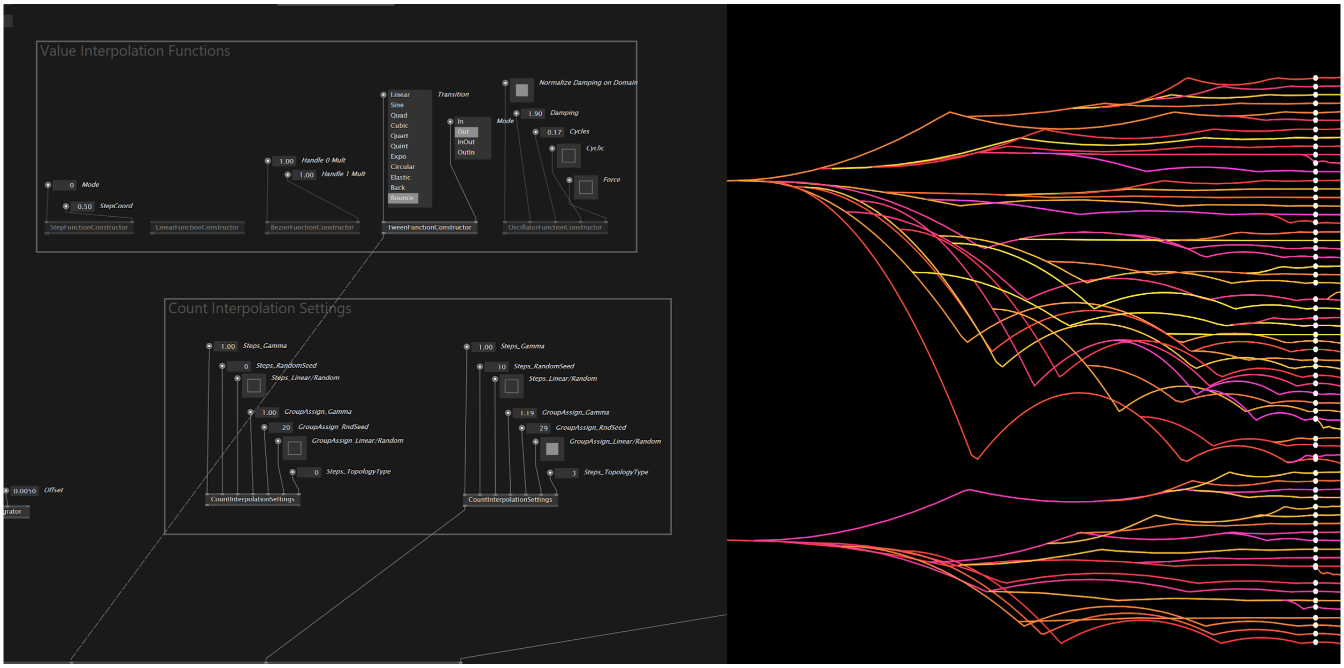


Interpolation Engine

Since one of the key components of Kairos is the timeline and one of the main features of a timeline is the possibility to interpolate between states, I made a research on how to handle interpolation for multiple kinds of data that we're used to working with in vvvv.

It turned out to be an incredible topic to explore, opening some creative possibilities that I couldn't think of before.

The outcome of this research is a library of tools that manage interpolation, which is at the core of Kairos software.


Value Interpolation Functions
StepFunctionConstructor
LinearFunctionConstructor
BezierFunctionConstructor
TweenFunctionConstructor
OscillatorFunctionConstructor
Linear
Sine
Quad
Cubic
Quart
Quint
Expo
Circular
Elastic
Back
Bounce
In
Out
InOut
OutIn
Count Interpolation Settings
CountInterpolationSettings
CountInterpolationSettings
grator

0 Mode
0.50 StepCoord
1.00 Handle 0 Mult
1.00 Handle 1 Mult
Transition: Bounce
Mode: Out
☐ Normalize Damping on Domain
1.90 Damping
0.17 Cycles
☐ Cyclic
☐ Force

1.00 Steps_Gamma
0 Steps_RandomSeed
☐ Steps_Linear/Random
1.00 GroupAssign_Gamma
20 GroupAssign_RndSeed
☐ GroupAssign_Linear/Random
0 Steps_TopologyType

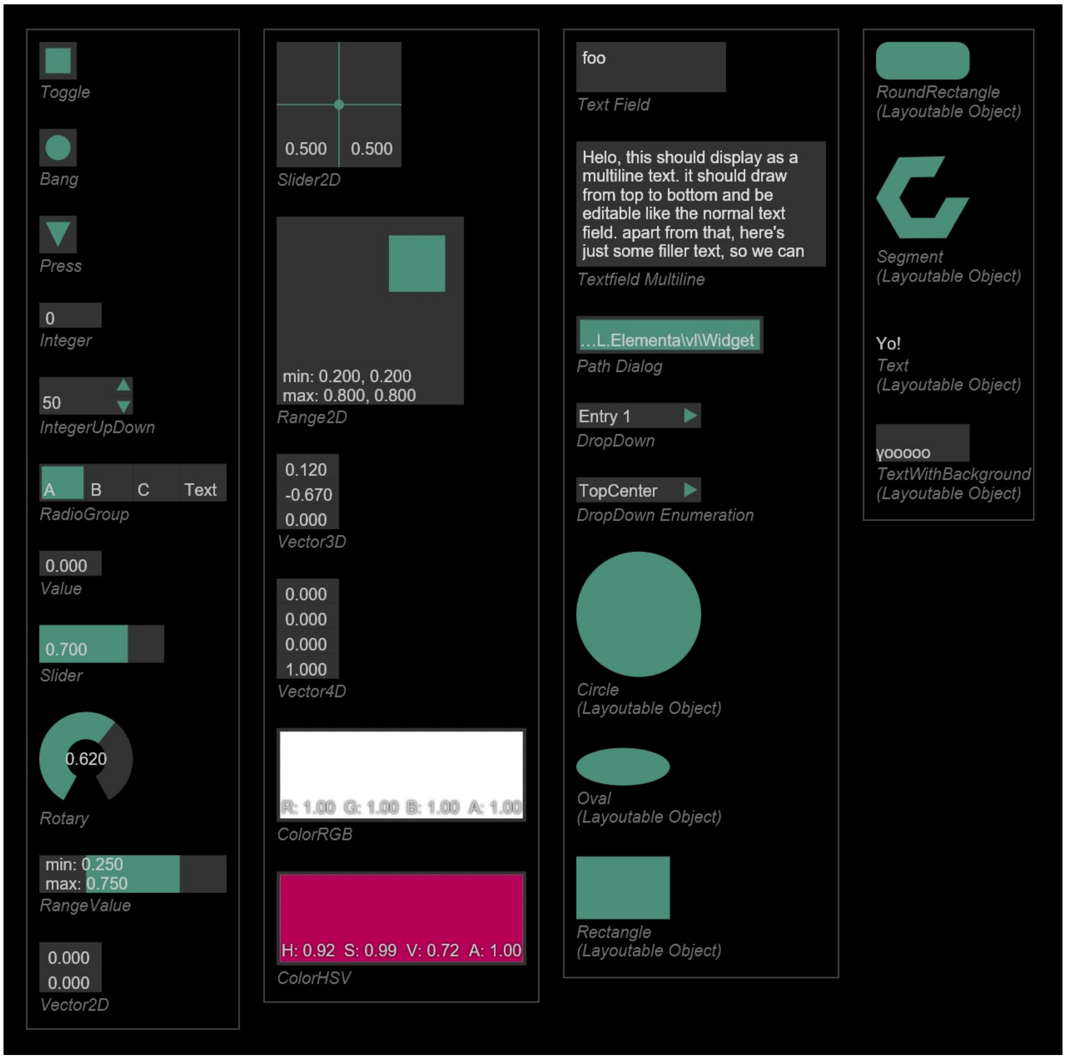
1.00 Steps_Gamma
10 Steps_RandomSeed
☐ Steps_Linear/Random
1.19 GroupAssign_Gamma
29 GroupAssign_RndSeed
☐ GroupAssign_Linear/Random
3 Steps_TopologyType

0.0050 Offset

Elementa

When we want to create a software, we need obviously to consider how to interact with it and make use of all its features in a convenient way. To do so we need a GUI (Graphical User Interface) that visually represents each part of the software and allows us to interact with it. In order to create the GUI for Kairos, I built a library of UI Widgets called Elementa. This library comes with a vast collection of building blocks to create a custom GUI. Elementa is not specific to my needs for making Kairos software: it is conceived to be a generic library, that gives to other vvvv users the possibility to quickly prototype a GUI and control their software. Elementa is now available as an open-source extension pack for vvvv and is being used and already maintained by many people in the community.

Available for download at this link: https://vvvv.org/contribution/vl.elementa

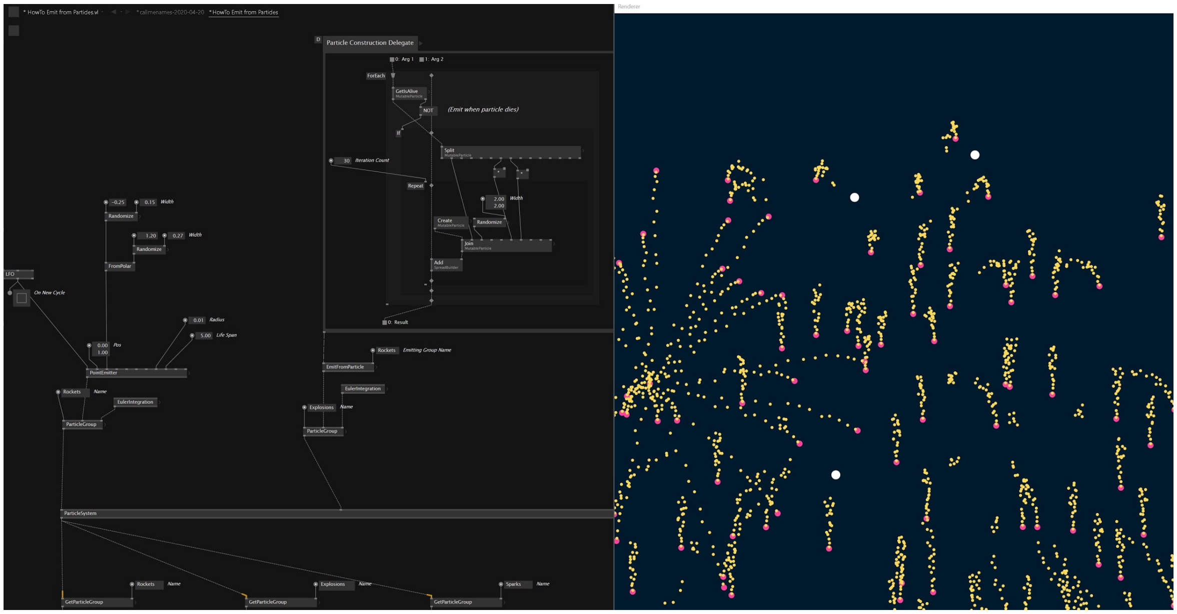


Helo, this should display as a multiline text. it should draw from top to bottom and be editable like the normal text field. apart from that, here's just some filler text, so we can

ParticlesSystem

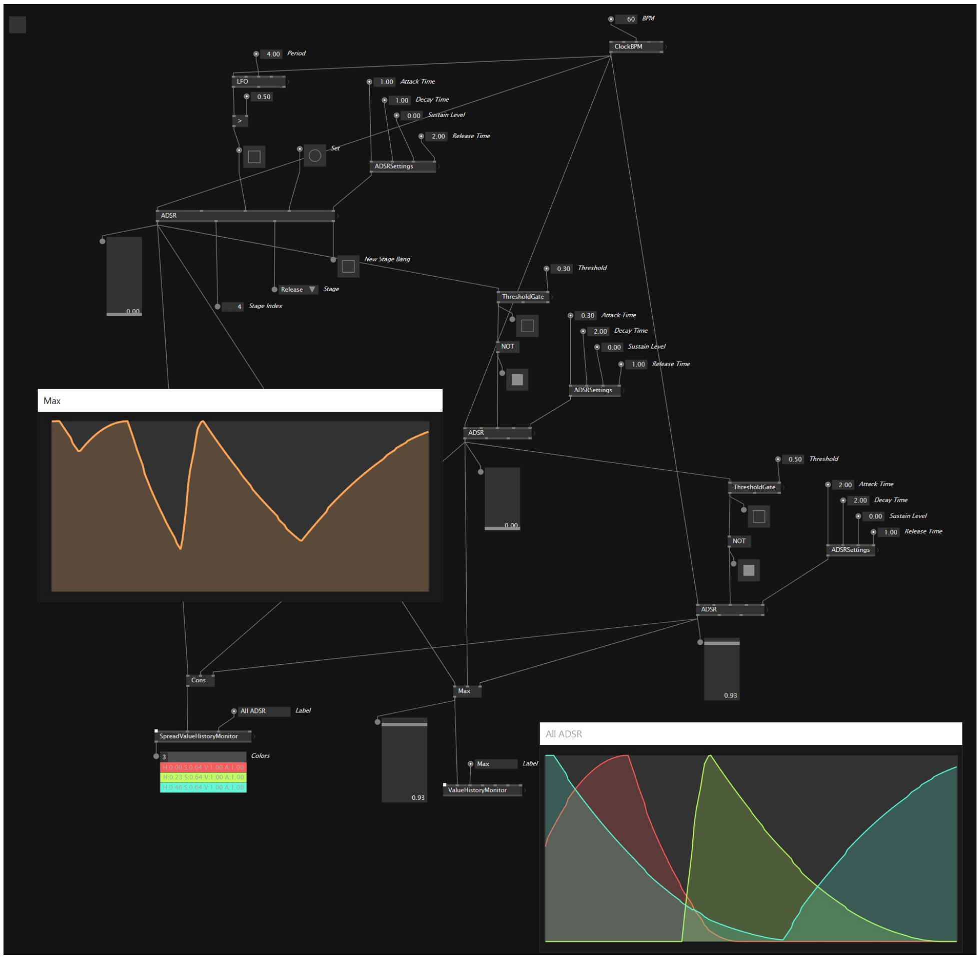
For the production of the live show "Still", which was the final goal of the residency, I needed to control particles in the visuals. For this reason, I created a Particles System library for vvvv. The library is a collection of tools to create, animate and draw particles. I shared it as an open-source extension pack for vvvv, also producing a series of video tutorials about it.

This is the link to the open-source library: https://vvvv.org/contribution/vl.animation.particlesystem



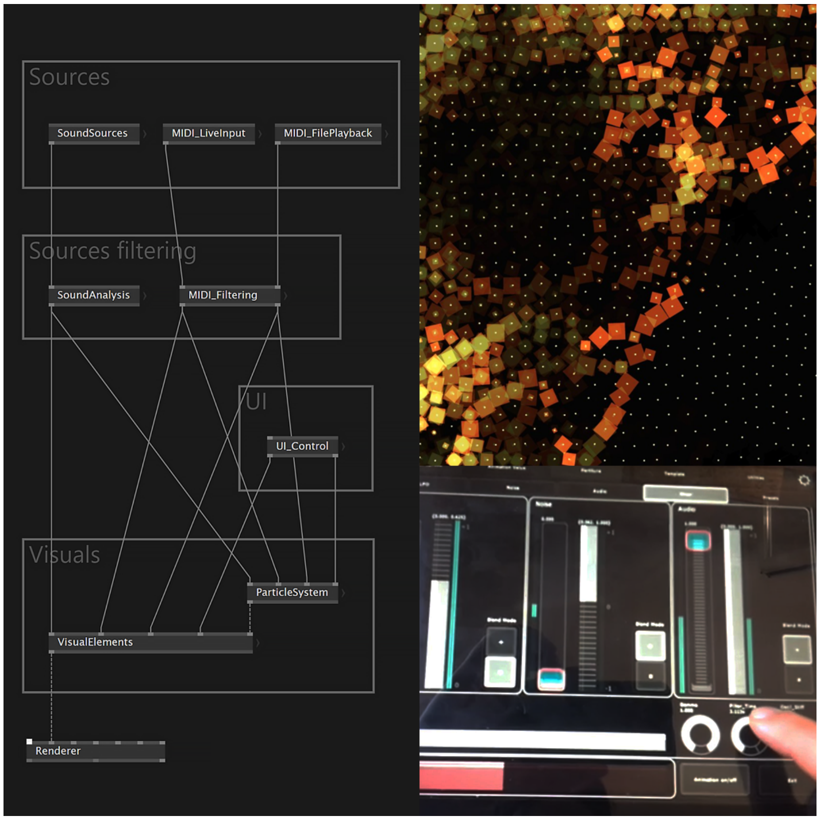
Generators

Thanks to Kairos I was now able to control certain aspects of the content of the live show using live generated data. I started to create a library called Generators as a collection of operators that generate procedurally animated values. Now I'll be also able to use the richness of generative algorithms to animate the content together with the sensor data from the sound analysis or from the body tracking.



Audio Analysis library

I often use live audio inputs as sources of data for controlling parts of the visuals.
It was obvious to me that I needed a set of tools to handle the sound sources, filter and analyze them to get the information I need from the sound. Together with the vvvv team we created a library to do so. It's already available for the vvvv community as an extension pack.
Using this library I also gave a full-day workshop about music-visualization, here in Berlin.



IV. CO-CREATION PROCESS

I had the possibility to stay at vvvv studio for the entire period of the residency. The tech team helped me in learning the new visual programming language they are developing. At the beginning of the residency we defined a schedule of weekly meetings between me and the team: during these meetings I was able to show the progresses, the encountered problematics and study with them the solutions to these problematics, finding ways to implement the features I had in mind. The schedule with weekly meetings that we initially planned didn't work much, especially at the beginning of the residency I was familiarizing with the features of the new language and I needed more support. I started to ask questions more frequently when I was stuck in the learning process, trying at the same time to find a compromise to not distract the team members too much. The team was very supportive, even if very busy in a crucial moment of the development of the Software.

In the second part of the residency I managed to be more independent, putting less pressure on the tech team, having their support in confined moments, with more focused topics and problematics.

V. IMPACT

A. Research Impact

I wasn't just using the programming language that is in development at vvvv studio, I was digging deeper, exploring, adding to it with my requests and ideas. This is, for a tech company, the best-case scenario, in which there is someone who is thinking within the system, making more out of it. Seeing the enthusiasm of someone that takes what you do and creates something out of it is also an important psychological support for a technical team, especially when you're developing a completely new technology which not always gets welcomed by the user base which is not used to it. Given the fact that some of the developers time was spent in communicating with me and considering my requests and observations, it's also true that most of these feedbacks were immediately useful for the development: either a report of something that used to work and didn't anymore with the latest updates, or some suggestion about features while they get implemented.

B. Artistic Impact

The innovative software tools that were developed during the residency are an incredible improvement in the way I create digital artworks, representing a new generation of methodologies that facilitate the artistic research. These tools will also affect a large number of creators, since they will be available for all the vvvv users in the next future.

VI. ART-SCIENCE INTER-RELATIONSHIPS

Art and science share a very important aspect which is the technique. They both apply some technique to achieve their goals, whether it is an instrument to observe reality, or a way to produce certain sounds that become music.
So, technique (technology) is a crucial part of the methodology of both fields and therefore represents an interesting common ground where to dialogue-inspire-interrogate each other.

VII. FUTURE DIRECTION AND ACTIONS

After this residency program I will continue to work on the performance "Still", which I presented as a work in progress at the end of the residency. My plan is to finish it before September, starting to perform it afterwards.
I will also continue to work on the software tools that I started to develop at vvvv studio, since they will be the basement for all my new artistic projects.
Since the experience was incredibly productive, my collaboration with the vvvv studio will continue after the end of the residency. I'll also be part of the Node20 event, a festival dedicated to vvvv that occurs every 2 years, giving a series of workshops and doing live performances.
I will also search for new opportunities like STARTS residency to support this research.

VIII. CONCLUSION

The research has benefited and inspired both myself and the tech team.
I had the chance to present a prototype of the artwork at the STARTS Residencies Days in February.
The work that has been done is giving already great results also in the artistic community: the open-source software that I developed have been already incorporated in other artists' projects and is already growing thanks to the contribution of new developers.
I see a great potential in continuing this research and sharing the results on the way.