

BENEATH

Structure

- 1. Idea
- 2. Technical conception
- 3. Setting the scope
- 4. Work in progress
 - 4.1 Programming the visual part
 - 4.1.1 Raymarching part one
 - 4.1.2 Behavior part one
 - 4.1.3 Externals
 - 4.1.4 Behavior part two
 - 4.1.5 Connecting raymarching with flocking behavior
 - 4.1.6 Raymarching part two
 - 4.2 Reprogramming in 2D
 - 4.2.1 Building the base layer of the oil painting
 - 4.2.2 Building what is beneath
 - 4.2.3 Making it interactive
- 4.3 Making it sound
 - 4.3.1 Talking monster
 - 4.3.2 Subtle atmosphere
 - 4.3.3 Hands Sound
- 4.4 Communication
- 5. Summary

1. Idea

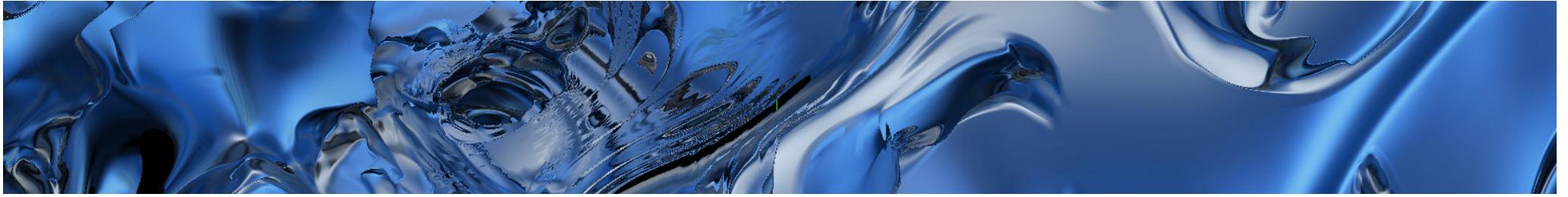
- images in exhibitions are being looked all the time
- they are passive and not alive
- but we give them meaning by looking at them and thinking about them
- could it be that they inherit some kind of life through that?
- if yes: how would life manifest itself in those images?
- would they be able to react and by that give something back to the viewer?
- how would they behave / could they have some kind of language to articulate themselves
- I wanted to translate this idea into an interactive installation of an abstract oil painting that reacts to the viewer of the installation
- the working title of this installation was: transcending oil painting in void
- but that got changed in the end into „beneath“

2. Technical conception

- visual part in vvvv
- audio part in VCV
- the installation should consist of a projection mapping in combination with tracking of users via a depthcam
- I would need a powerful pc with an even more powerful graphics card
- thank the uni that I got one from university :)

3. Setting the scope

- the painting should have a set of different emotions stages and behaviors towards the user
- calm /silent
- awakened
- unimpressed but reacting
- exited and approachable
- frightened and aggressive
- these emotions and behaviors should correlate to the users behavior and distance to the projection

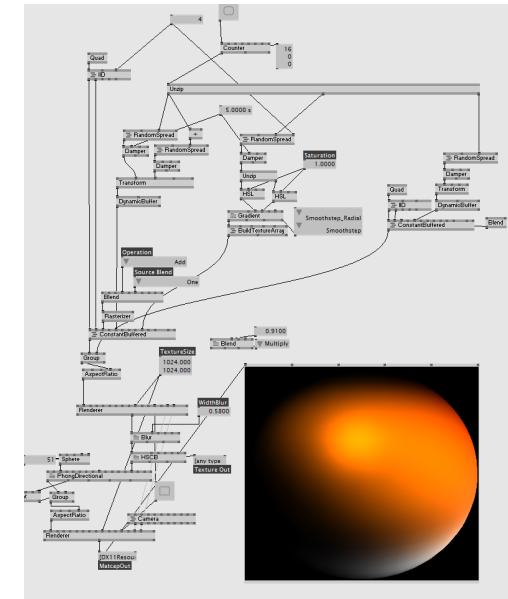


4. Work in progress

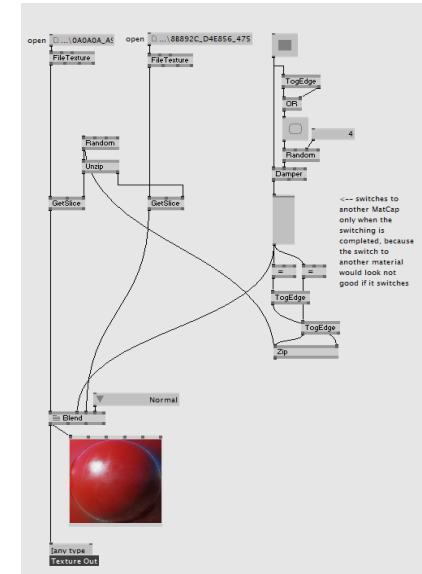
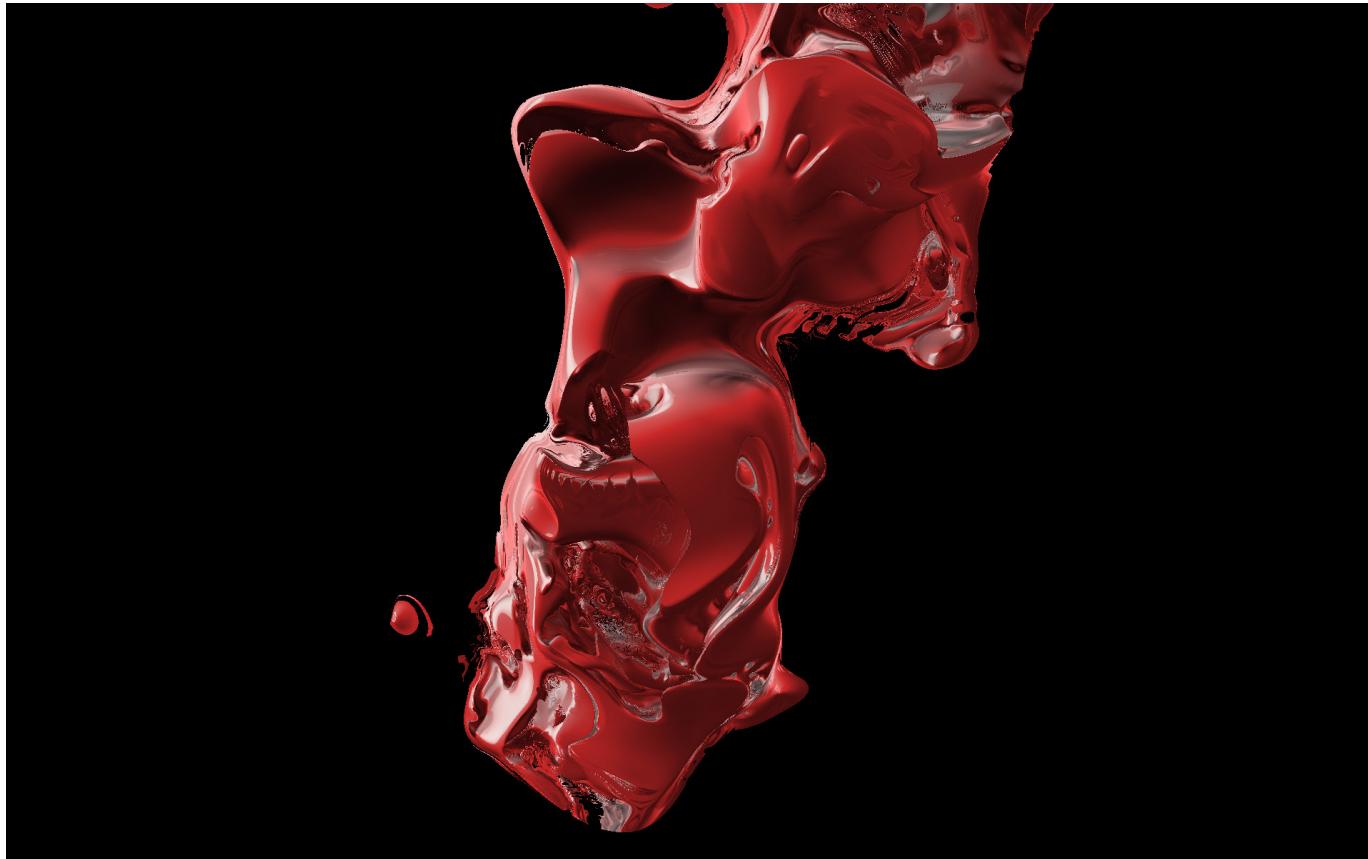


4.1.1 Raymarching part one

- for the visual part I wanted to work with procedural noise and raymarching
- started experiments in that field to see what is possible
- There were different rendering options available for raymarching signed distance field (sdf's) I didn't know about before
- raymarching only with normals
- raymarching with matcap
- raymarching with physical based lighting
- decided for raymarching with matcap because of the balance between performance and how it looks
- explored further possibilities with raymarching
- scalar sdf operations, combination / blending methods, fractal multiplication and domain distortion
- these methods lead to more interesting and appealing results than only just using one sdf noise on its own
- for the different emotional stages of the painting I decided to assign these partly to the texture of the painting
- I tried to build my own matcap switcher by layering different colored quads onto the texture of a sphere

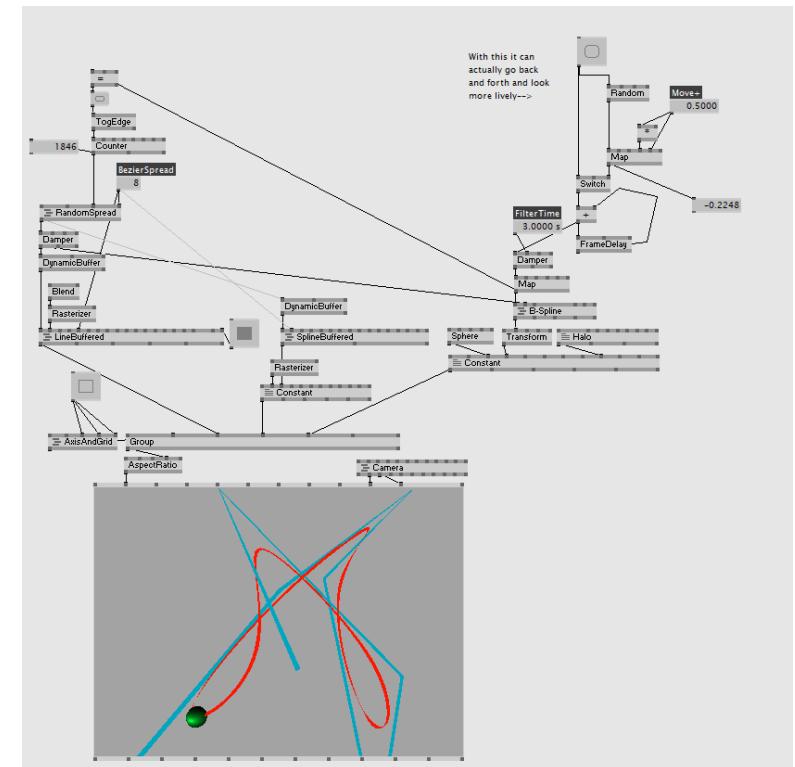


- but it didn't look as nice as I wanted it to be
- while researching what matcaps are and how they function I found a library of a ton of different matcaps online
- decided to build a matcap switcher that would change the texture of the oil painting as the emotions of the painting change
- turned out to look way better then the one I build before



4.1.2 Behavior part one

- the different emotional stages should also be represented in movement that could be seen as behavior
- I decided to start with the last emotion (aggressive) first because I thought it would be the hardest one to look natural
- build a first version of the mover that works with several random generated 3d vectors that are connected with a spline and form a path that way
- it is programmed so that an example sphere is moving back and forth when random triggers are being activated → those triggers should originate from the interaction which wasn't implemented at that stage

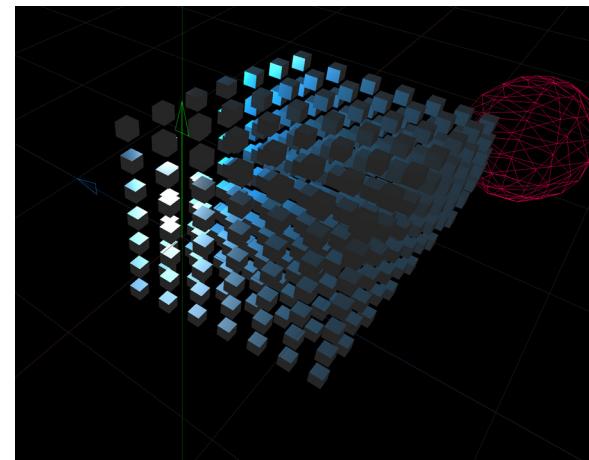
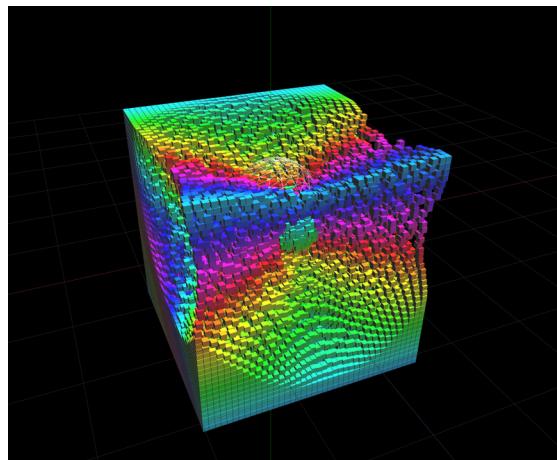


4.1.3 Externals

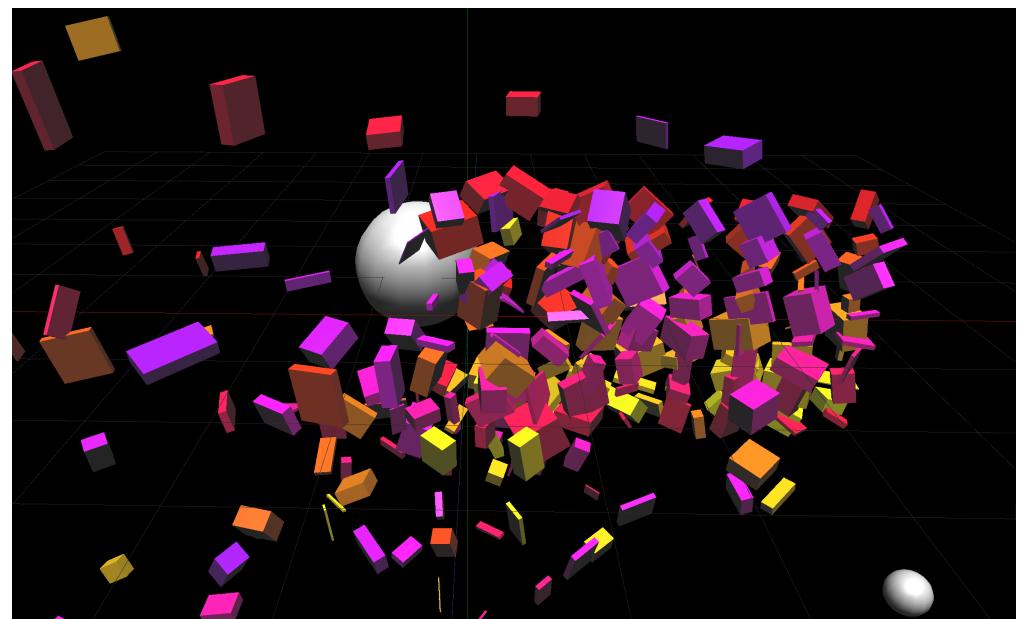
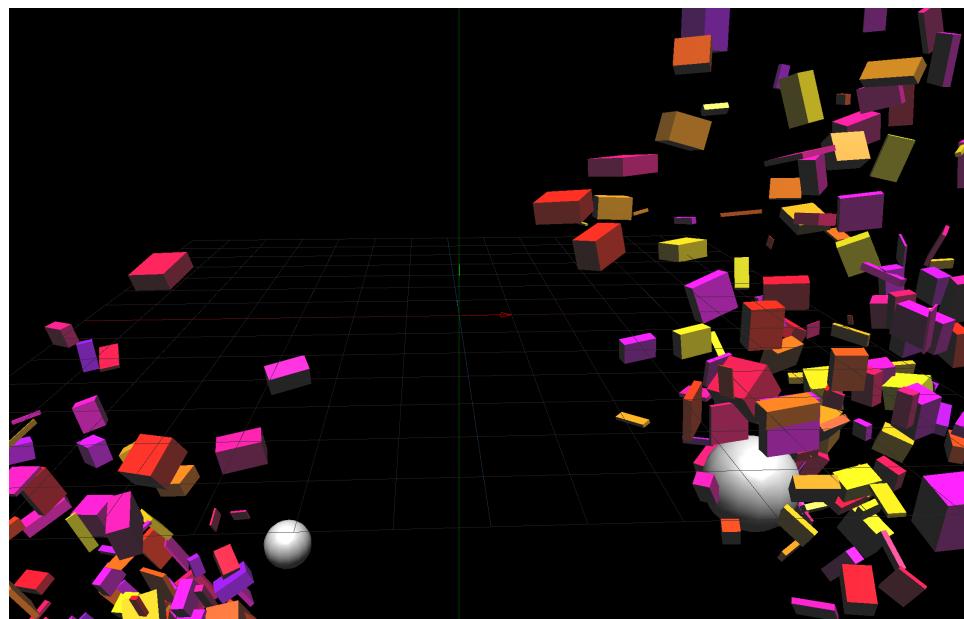
- early in january I got the Kinect and started testing how to implement it
- skeleton detection worked flawless → way better then with the Orbbec Astra
- decided to go with the Kinect
- build in the RTX 2080 → nearly twice the performance as with my 1070

4.1.4 Behavior part two

- researched on how to implement the other two active stages
- for the unimpressed/reacting stage one should be able to move the hand through the painting and it would leave a little trail as when you wave your hand through water
- for the excited stage I thought of the „ink“ of the painting following the movement of your hands as you move them around like a swarm of birds would follow
- the painting also should develops more into a plastic thing then being a screen filling painting
- experimented with normal attractors
- behavior didn't look that natural at all
- tried out particles and attractors but the pulling in behavior was the same as with the normal attractors
- Problems with the particles lifetime → they do not live very long and want the particle to not die at some point
- they do not come back to their original position they spawned



- found out about boids and flocking behavior
- the boids can either have an enemy or an target they follow
- this principle I thought was very suitable
- the boids first form a screen filling wall of quads
- build it so you can move your hand through them and they try to avoid that by scattering
- the second behavior is programmed so they see the moving hands as their target → they follow the hands beautifully
- now I would only need to implement a transition between these stages and that would work out fine I thought

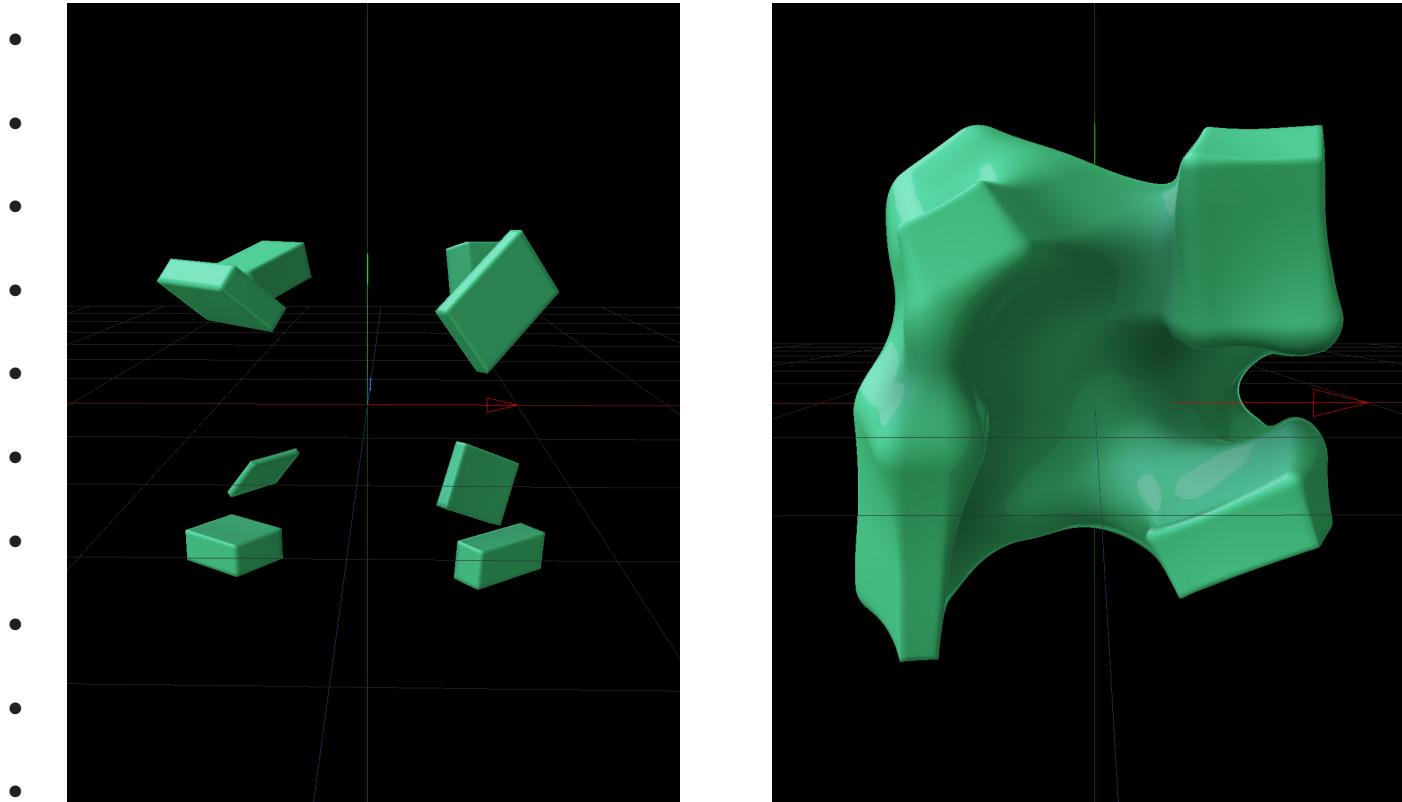


4.1.5 Connecting raymarching with the flocking behavior

- The overall idea was to make use of the flocking behaviot by taking their position and rotation vectors
- send the values to raymarched sdf boxes
- thes boxes are combined via union round to make smooth transition between them
- then intersect these boxes with the existing sdf noise
- this is where the problems came up

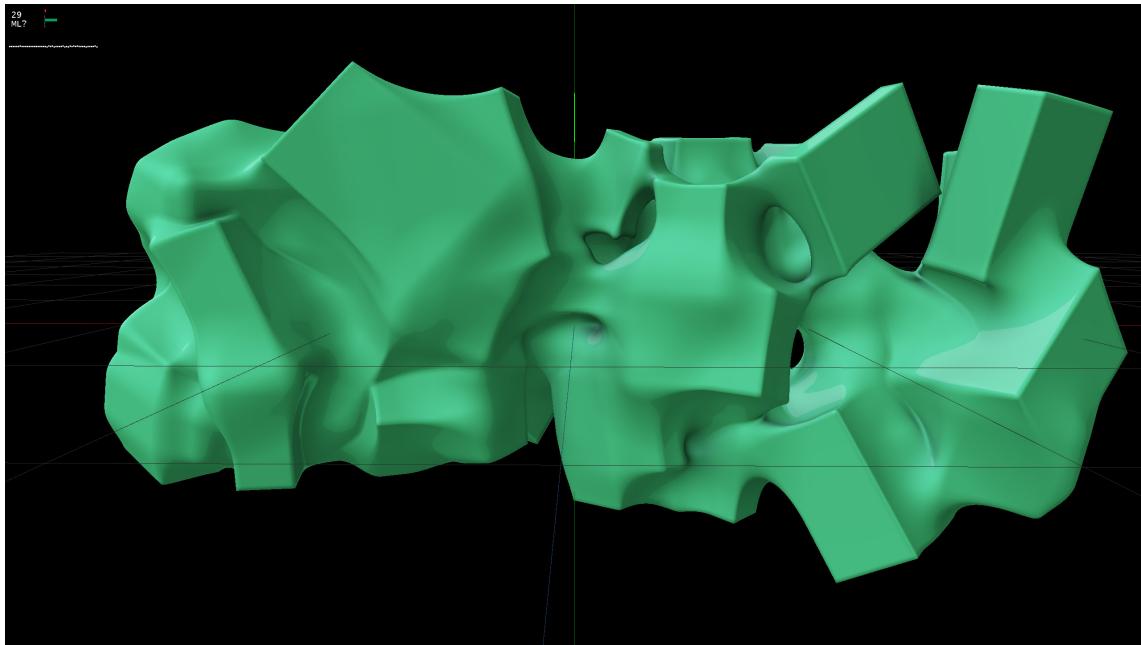
4.1.6 Raymarching part two

- first I tried to build this wall with sdf boxes by only using a 2x2 grid of boxes
- had to process each box on its own to make it compare / union round to each other box

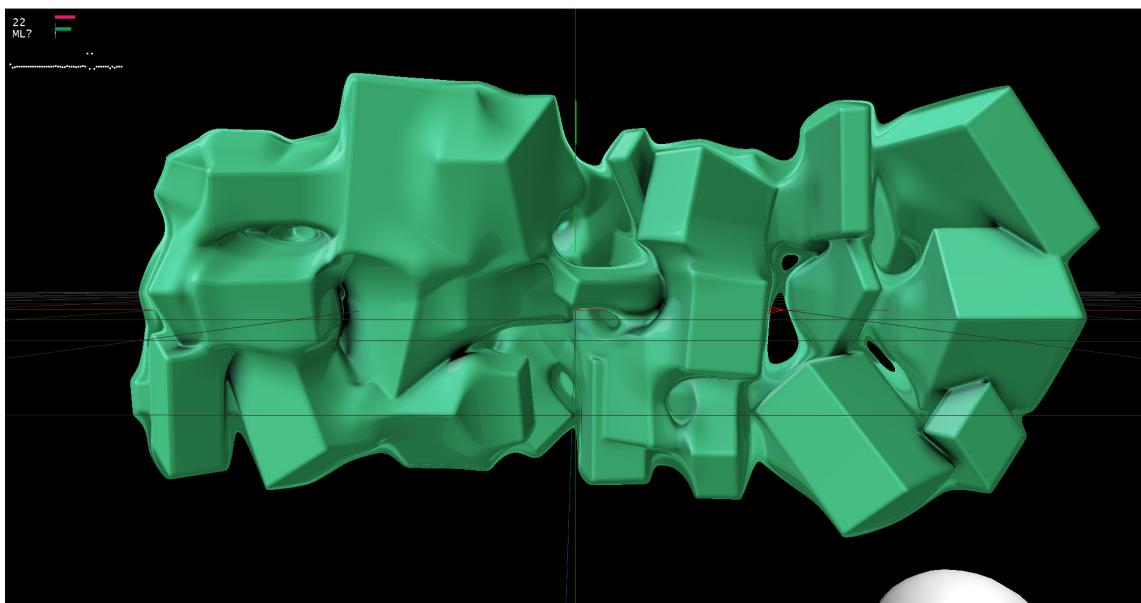


- so I had 8 separate combination processes for 8 boxes → not very performant if there I would use a bigger grid size
- to make it more performant I stayed with 8 combination processes and randomized the indices of grid spread of the boxes
- this should make it look like actually every box compares to each other box of the grid

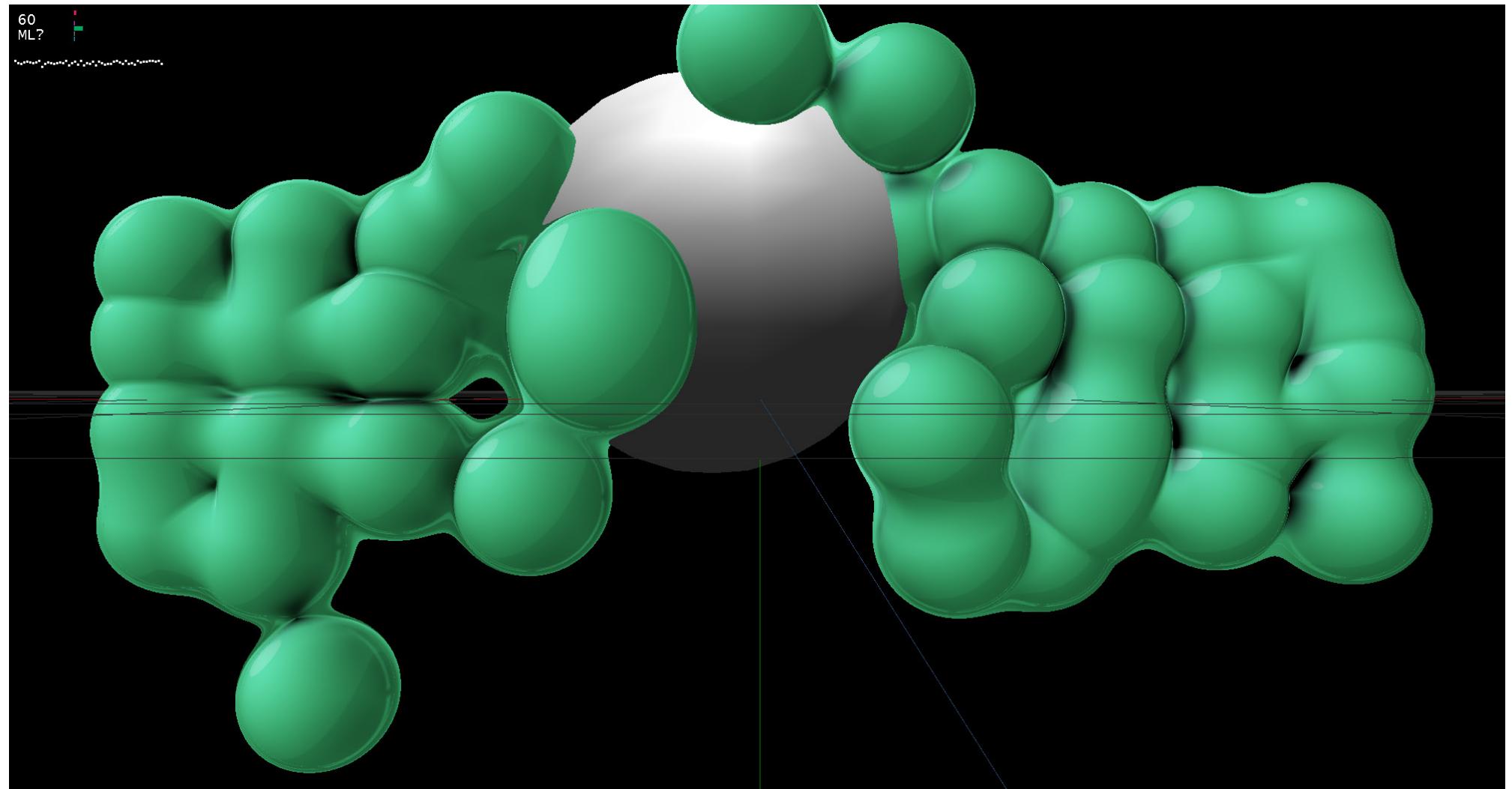
- worked fine but the performance did go down very drastically → 30fps...



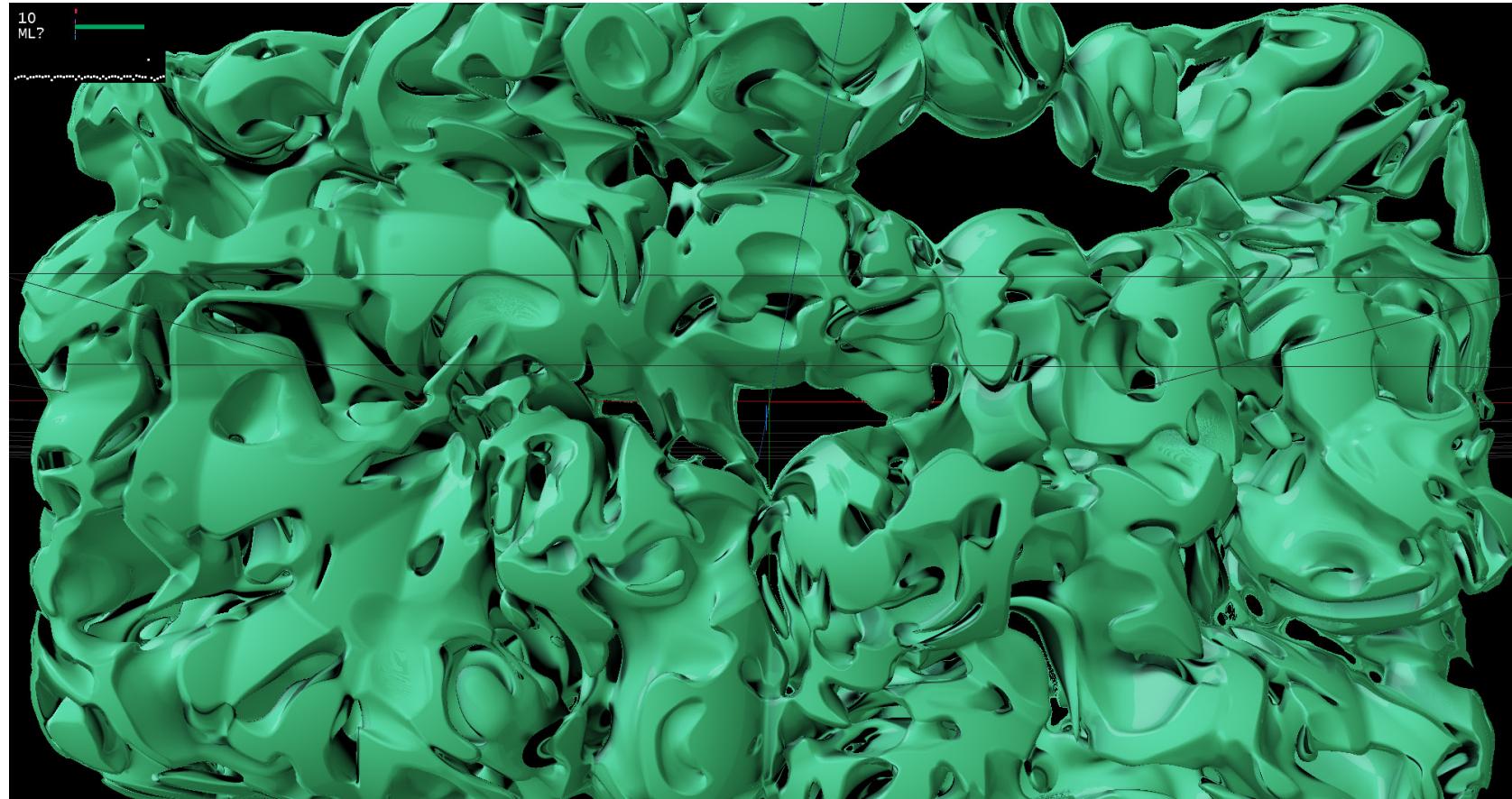
- decided to lower the comparing processes to two but there was still a big problem with the performance and it looked worse



- for the installation to really look good I would have to take a lot more sdf boxes to actually make
- decided to try sdf spheres because Stefan told me they would process easier then boxes
- got better performance out of it but it didnt look at all how I wanted it to



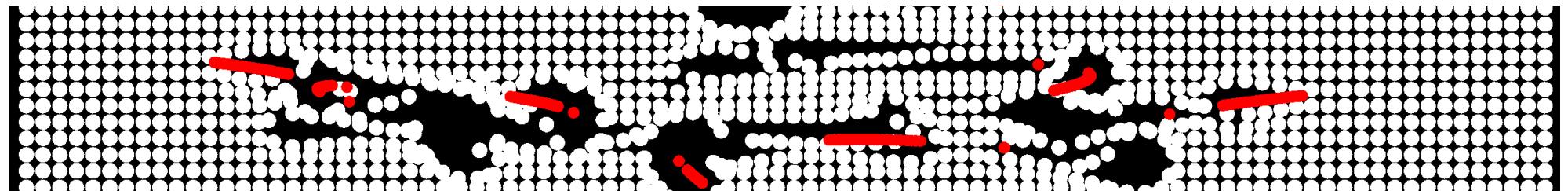
- by intersecting the sdf spheres with the noise the performance got down to 10fps
- with a more complex noise the performance got even worse



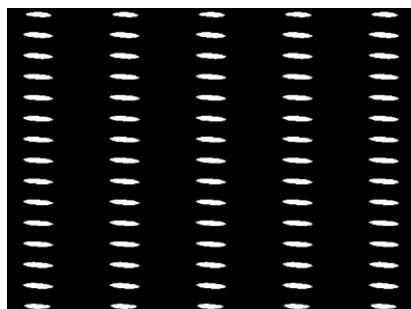
- this is not acceptable at all
- I tried to change the parameters of the raymarching but it didnt help alt all
- I had to rethink my approach
- The solution would may be to programm the whole thing again in 2d...

4.2 Reprogramming in 2D

- tried to port flocking behavior into 2d
- the behavior I was getting didn't really matched my expectations
- the boids changed their positioning too much when they were coming back to their original positions after they tried to avoid the „enemy“ hand
- got back to the normal attractors of vvvv
- with only one attractor per moving hand I couldn't get this trail effect that I was looking for
- to get this „wave through water“ behavior I build it with queued circles
- each circle is its own attractor and pushes away the white circle around him and leaves a trail like this



- I programmed again the union round processing in combination with sdf circles to have this „glued together look“ of the overall base texture



- but again when porting the position data of the white circles to the sdf circle I had severe performance issues
- the final solution I found was to only use the former attractors and combine them into a mask
- I experimented with remapping the values of a sdf circle to make it have sharp corners
- distort the sdf circle with a vector noise field and I have a perfect mask to look beneath the the painting and see the actual soul of the painting

•

•

•

•

•

•

•

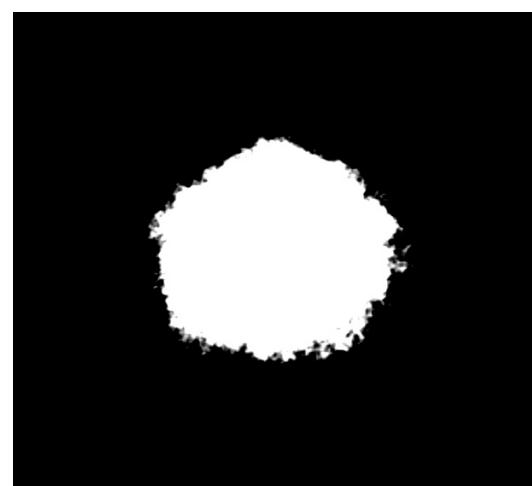
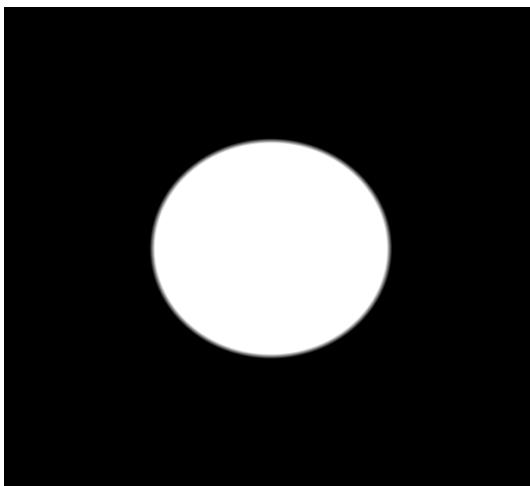
•

•

•

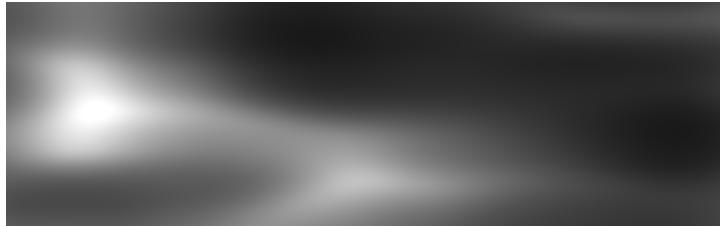
•

•

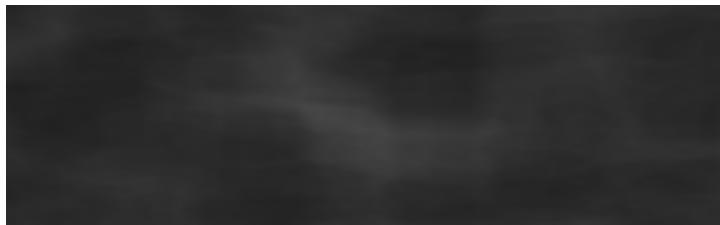


4.2.1 Building the base layer of the oil painting

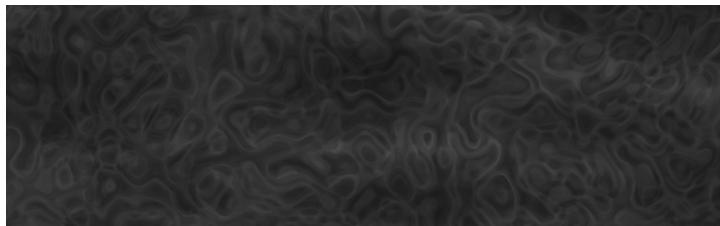
- I took a sdf perlin noise with low frequency and low amplitude



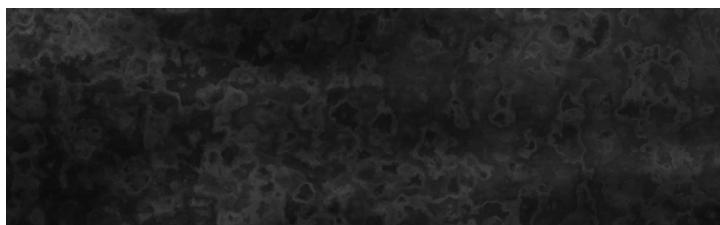
- then fractal multiplied it



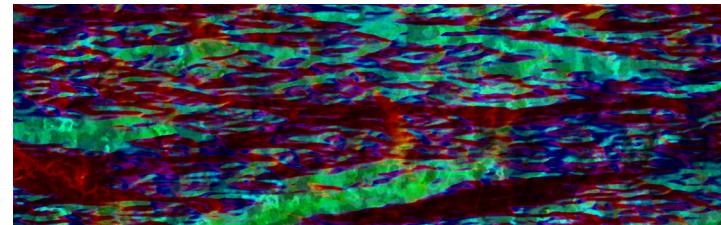
- distortion is the perlin noise with a noise vectorfield



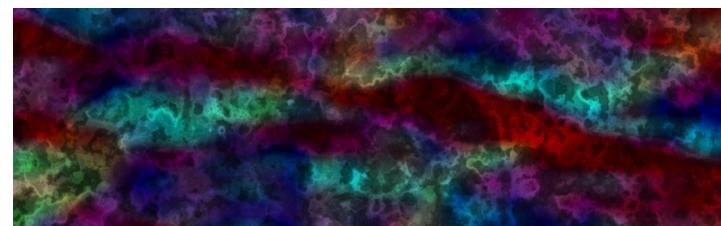
- the vectorfield then is also fractal multiplied



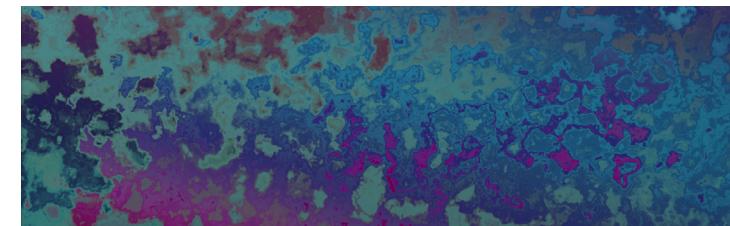
- for the coloring the noisefield I multiplied it first with the normal map of the first fractal multiplied sdf perlin noise



- blurred it



- then displaced the colored texture by the normal map of the original black and white noise texture



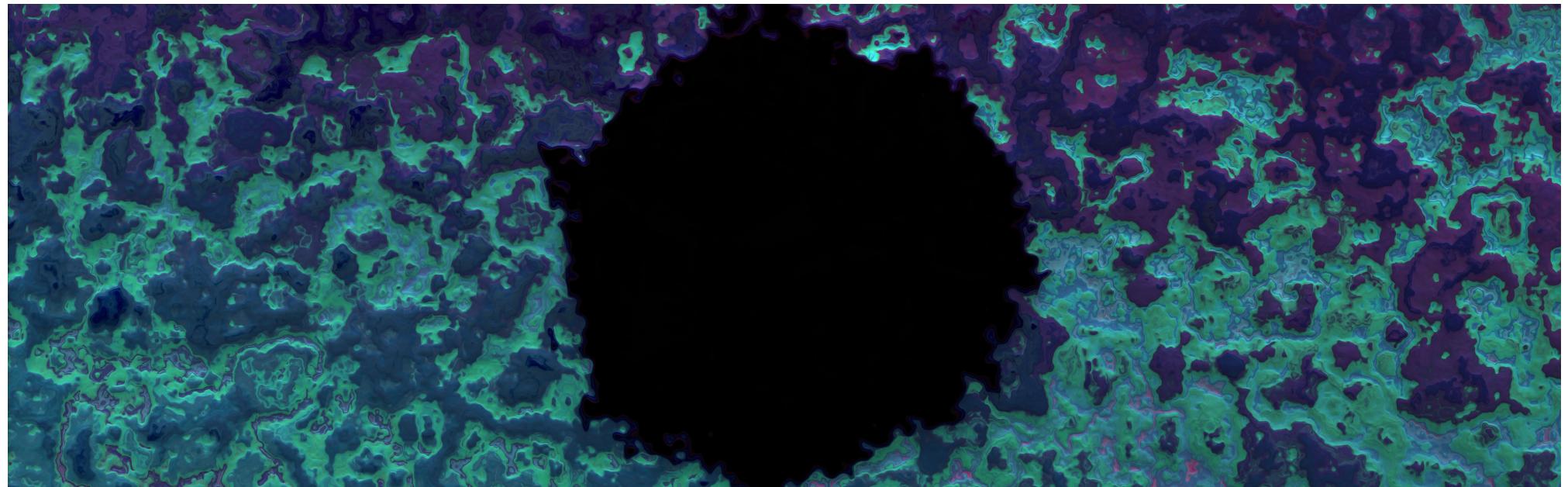
- this way the image is coloring itself fully synthetically
- another advantage is that the color moves according to the base layer of the perlin noise which makes it look more lively

- to give the painting at least some kind of 3D touch I applied a bump map



4.2.2 Building what is beneath

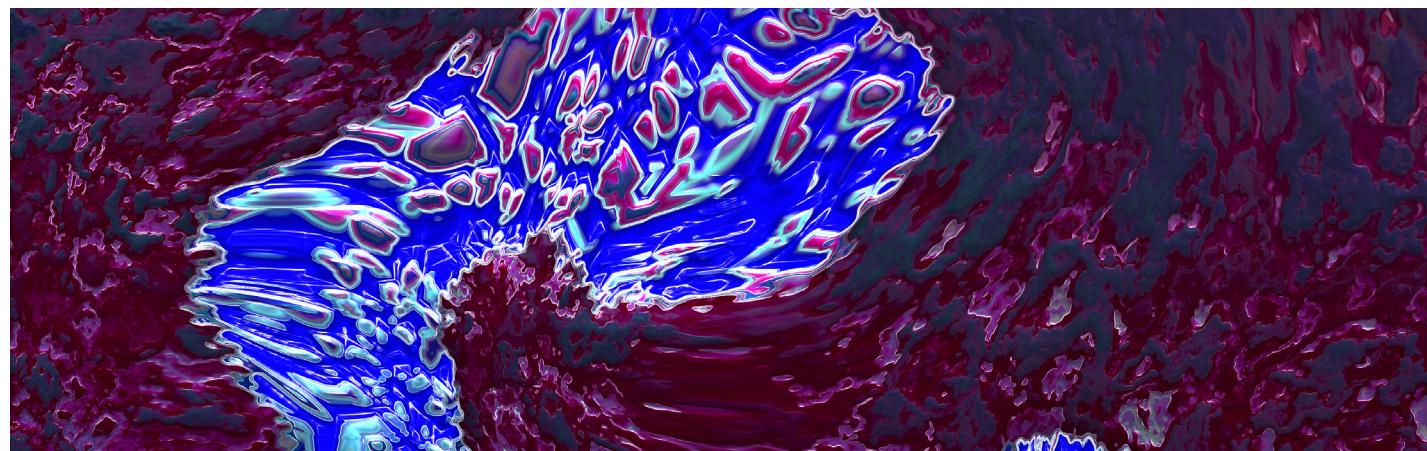
- I had to reconsider my approach a bit in terms of time
- decided to make some kind of hidden soul of the painting that lies beneath it
- this soul could be uncovered when coming too close to the painting and moving your hands too fast
- with the prebuild sdf circle mask you can now look through the painting



- I decided to make the soul or monster of the painting contrasting in its structure to what lies above it
- therefore I used worley noise

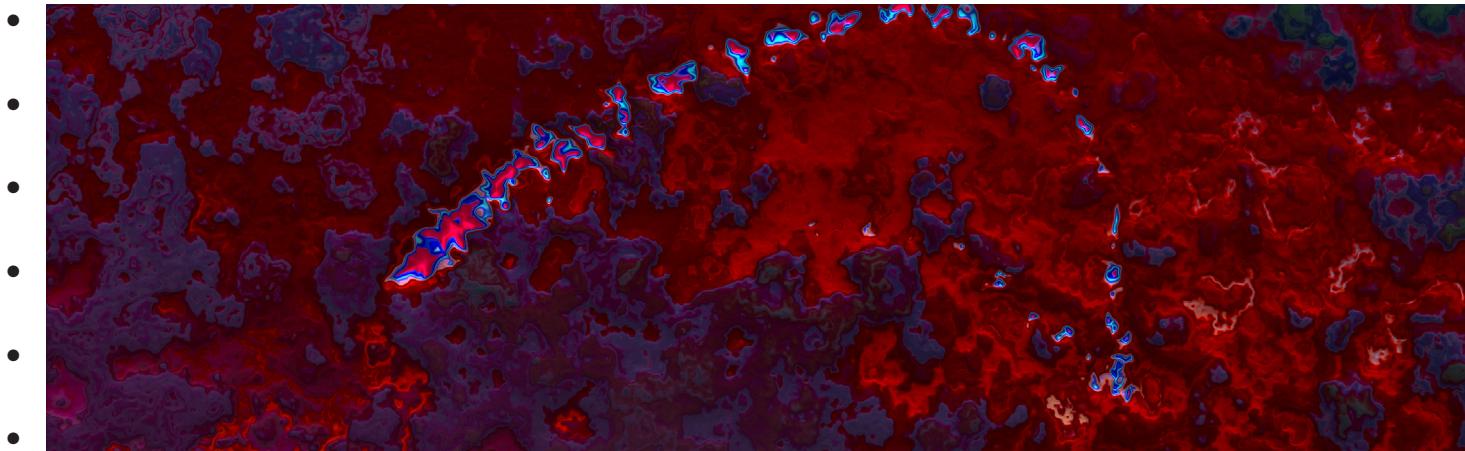


- then the noise is again fractal multiplied and distorted
- to give it more impact when the monster is reacting and aggressive, I detected the edges of the general texture and max multiplied a black and white version of it back on itself



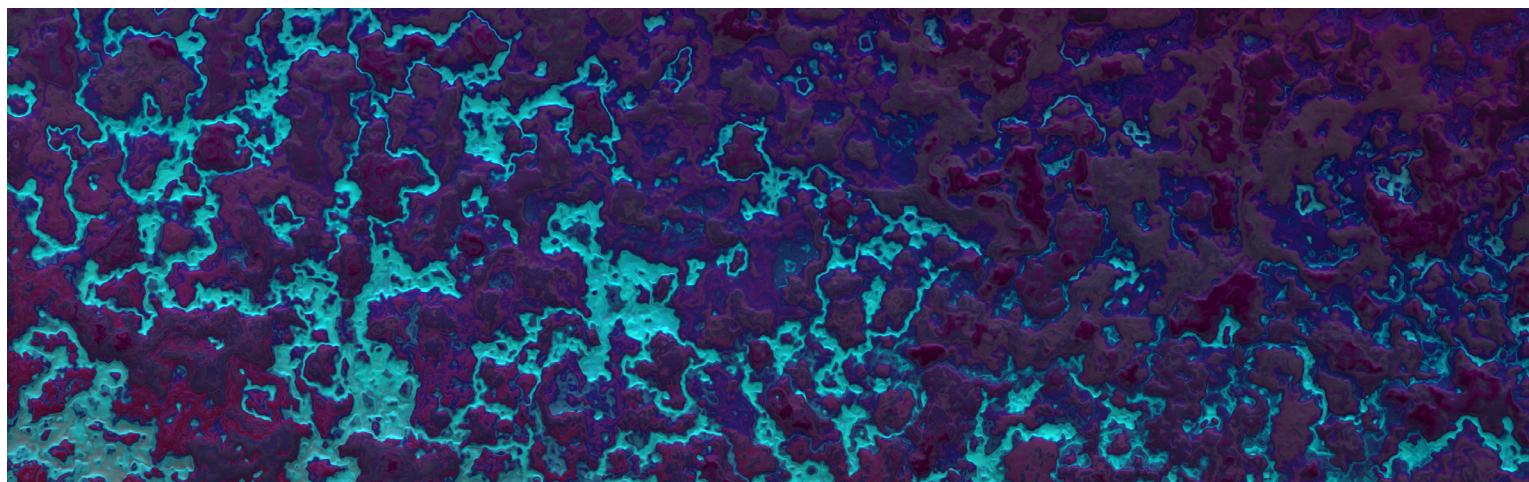
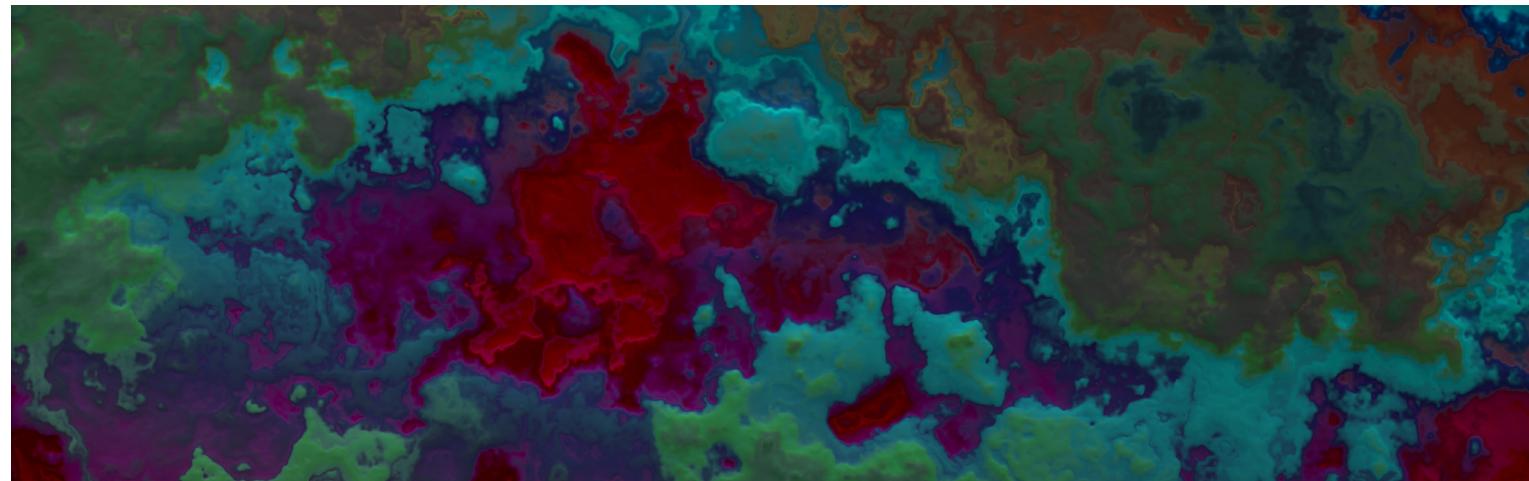
4.2.3 Making it interactive

- I connected the queued points with the hand tracking of the Kinect
- made it so that the points get smaller to the end of the queue
- was a bit problematic to make it work for multiple user inputs
- found the solution by spectral adding two values of the spread together



- to awake the monster there are certain requirements to be met
- first you have to be very close to it (the distance can be tuned to the rooms measurements)
- and you have to wave your hands to fast
- then the monster is triggered

- it is programmed in a way that if there are multiple users interacting with the painting there is an average calculated over the general distance of all users and the general speed of their hands movement
- I decided to programm it that way because I dont want the monster to be awaken too fast
- also correlating to the average distance of the users is the general harshness of the structure of the painting
- it changes with the amplitude of the base perlin noise



4.3 Making it sound

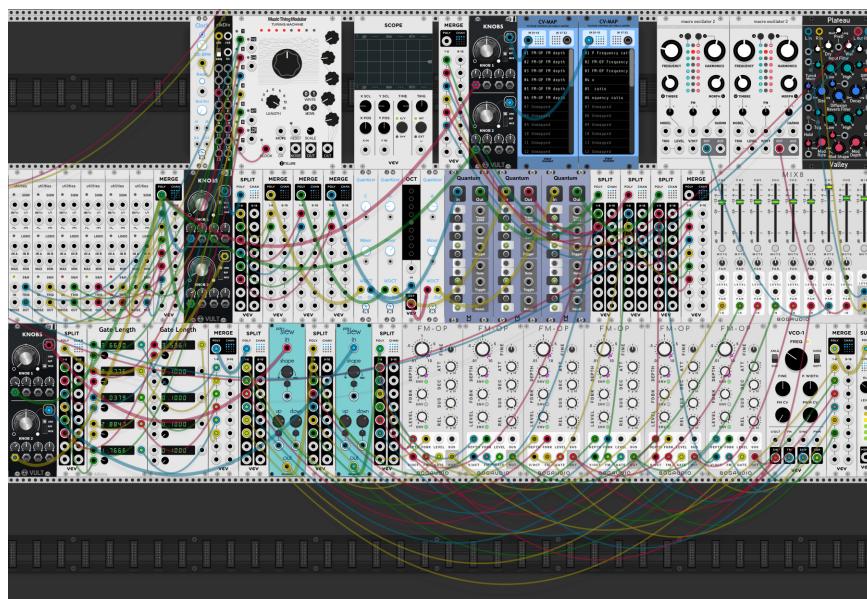
4.3.1 Talking monster

- I decided to build the soundscape of the installation in VCV because I am interested in modular synthesis a lot and wanted to really learn more about it
- making a good and realistic sounding behavior of the oil painting I thought was very dependant on a lot of modulation
- I first started to build again the aggressive behavior of the painting
- the monster consists of three different oscillators
- two macro oscillators from audible instruments and a normal vco with a low basefrequency to give the oil monster some weight
- the pitch of these two oscillators is beeing modulated by various chained, random triggered envelopes
- the signal is then processed via a granular module whose texture is also being modulated randomly
- the processed signal is then again mixed with the dry signals and send into a reverb



4.3.2 Subtle atmosphere

- it consists of six frequency modulation synthesizers
- they are tuned in pairs of two to three scales which are layer over each other
- C minor / D minor 6 / A minor 7
- I chose this layering because these chords do layer quite nice over each other because of they are relatives via the fifth or the dominant
- the 6th in D and the 7th in A are there to give the chords some friction
- The atmosphere reacts in the same way it does in the visual section too
- when the visual structure gets rougher because of the users distance the atmosphere starts to detune itself
- if the users are even closer it starts to modulate its fm-depth which results in a more rusty sound



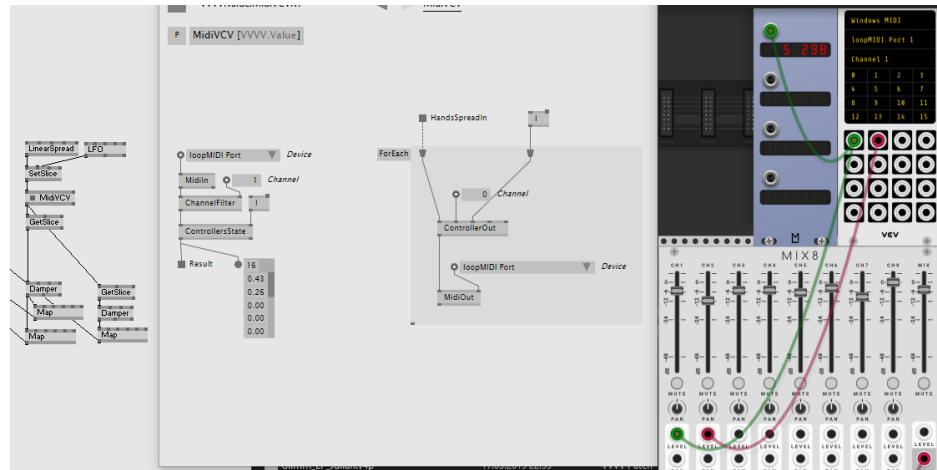
4.3.3 Hands sound

- to support the animation of the hands movement in the audio section i decided to make a bubbly or drop like sound
- I tried to do that from the ground up but the result was not very satisfying
- instead I found out that the macro oscillator module I was using before has exactly such a sound synthesis build in
- I had to use 8 separate macro oscillators for an eventual 4 person usecase because this module is not capable of being polyphonic
- they all go into a mixer where the level of each module is adjusted according to the velocity of the hands movement which is send to VCV from vvvv

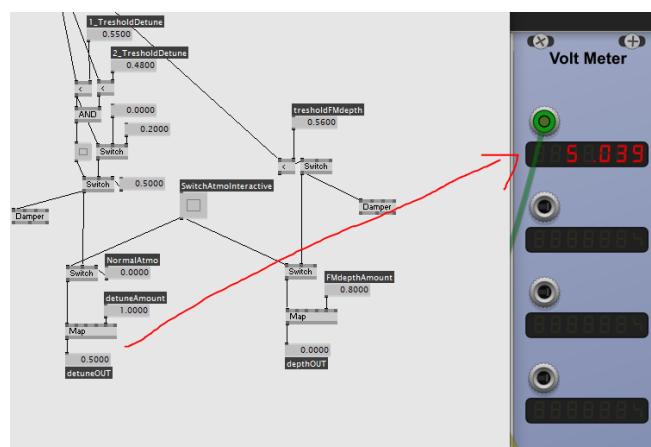


4.4 Communication between vvvv and VCV

- I programmed a bidirectional communication between vvvv and VCV
- therefore the external program „loop Midi“ is needed because Windows isn't really good at routing Midi internally between programs



- fun thing to note: there is a small offset when sending 0.5 from vvvv to VCV / normally that should be 5V
- corrected the offset in VCV because the atmo was constant out of tune



5. Summary

- the interactive painting works, even at higher framerates and way higher resolution than 1920x1080
- you can interact with it by coming closer and moving your hands
- moving your hands will uncover a bit of what is beneath the painting
- coming closer will make the overall structure in visual and audio rougher and harsher
- when coming too close and moving hands too fast, the painting is showing its true self and is reacting aggressive towards the user in picture and in sound