

# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
MEDIA

# Creative Simulation Technologies 2022

## Participant case studies

### 1. Agne Danieliute-Granickiene

2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina

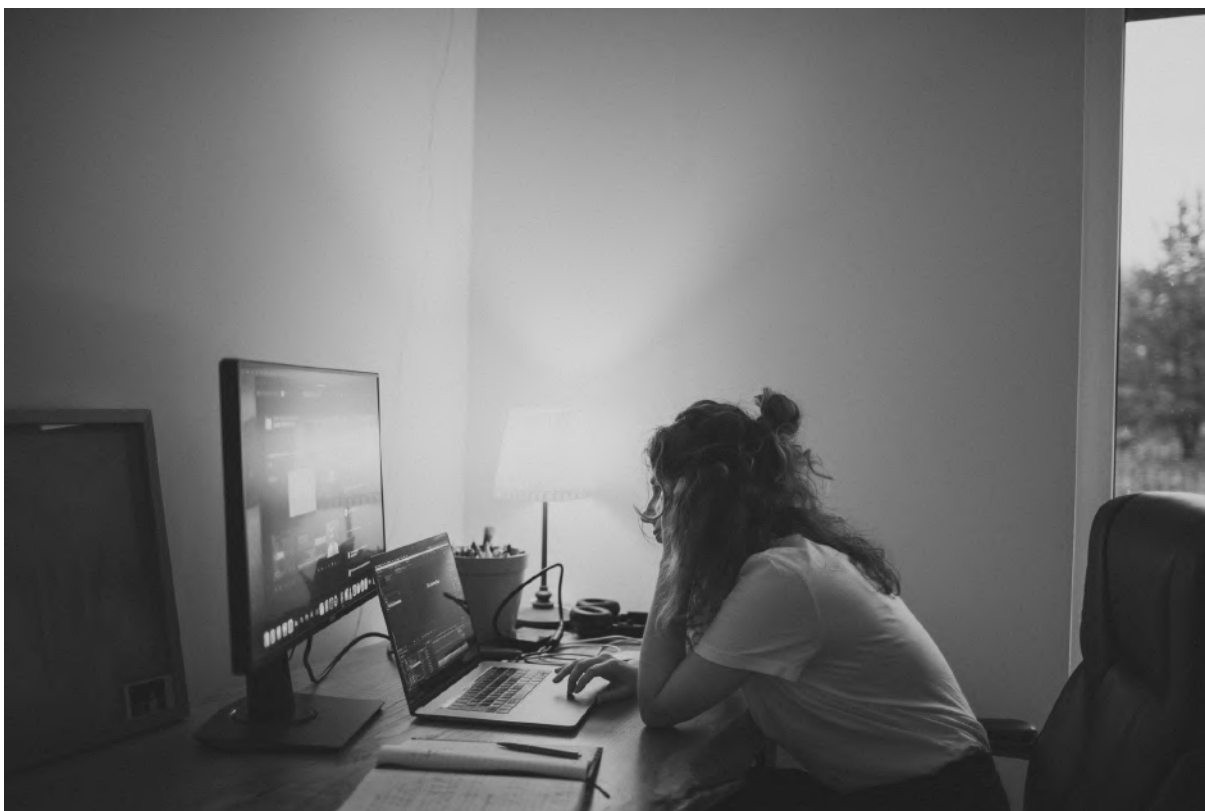


Creative  
Europe  
MEDIA

My name is Agne and a little bit more than a year ago I had a ceramics studio:



But then I decided to change my career or calling and to learn some kind of software. At first I started with After Effects.



And after few months by a super lucky chance I got an opportunity to study in this awesome Creative Simulation course. I had zero experience in 3D



This is me learning Houdini:)  
Just joking.

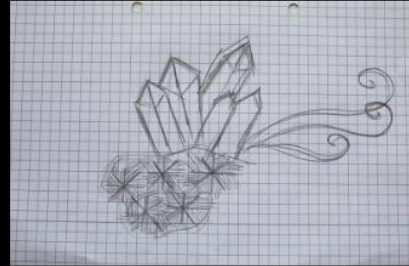
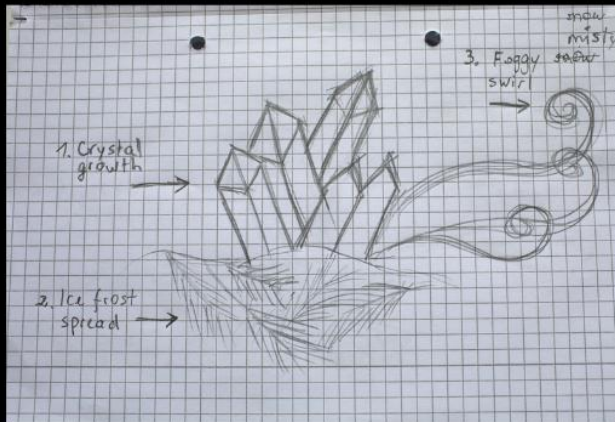
I knew that the main theme for the project should be somehow related to nature, especially winter and Arctic climates. I lived there for a while and I wanted to transmit that feeling by creating a digital nature scene. So after a week of Project Planning I got my half realistic / half magical looking plan for a landscape, containing snow, mist, frost and growing crystals.

Three simulations were planned:

- Crystal growth
- Misty snow
- Frost spread



# Sketches



16 SEPTEMBER 2022

PAGE 01

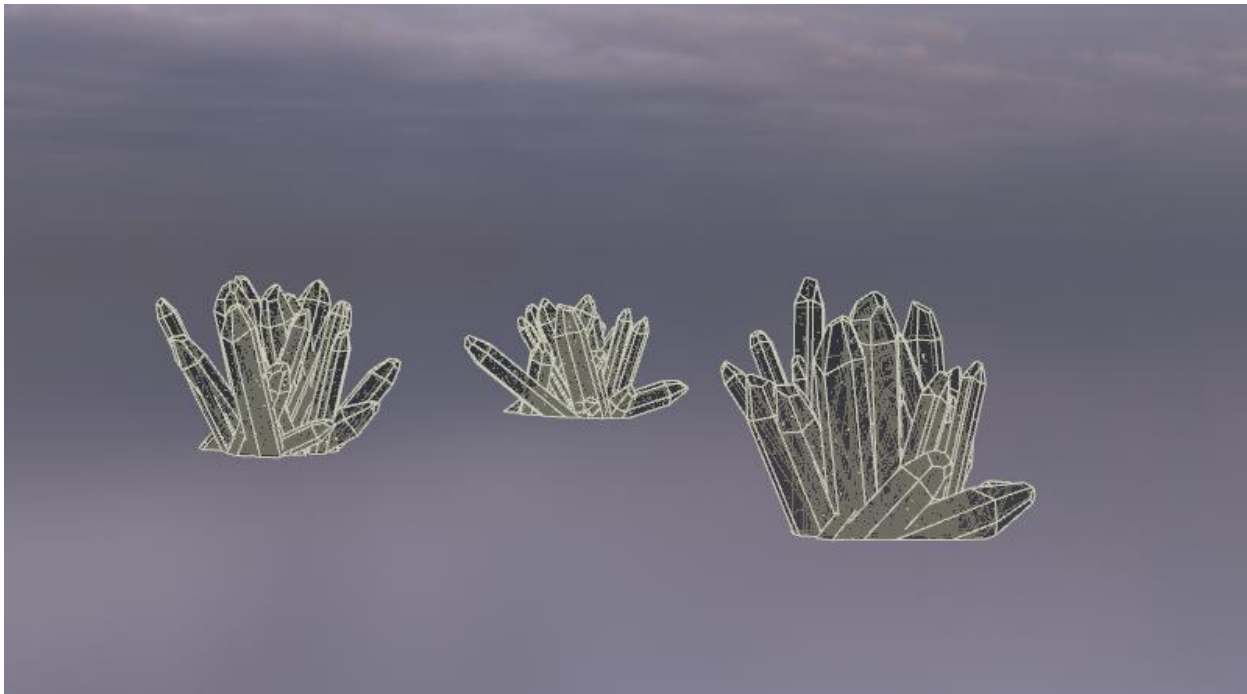
# Crystal cluster examples



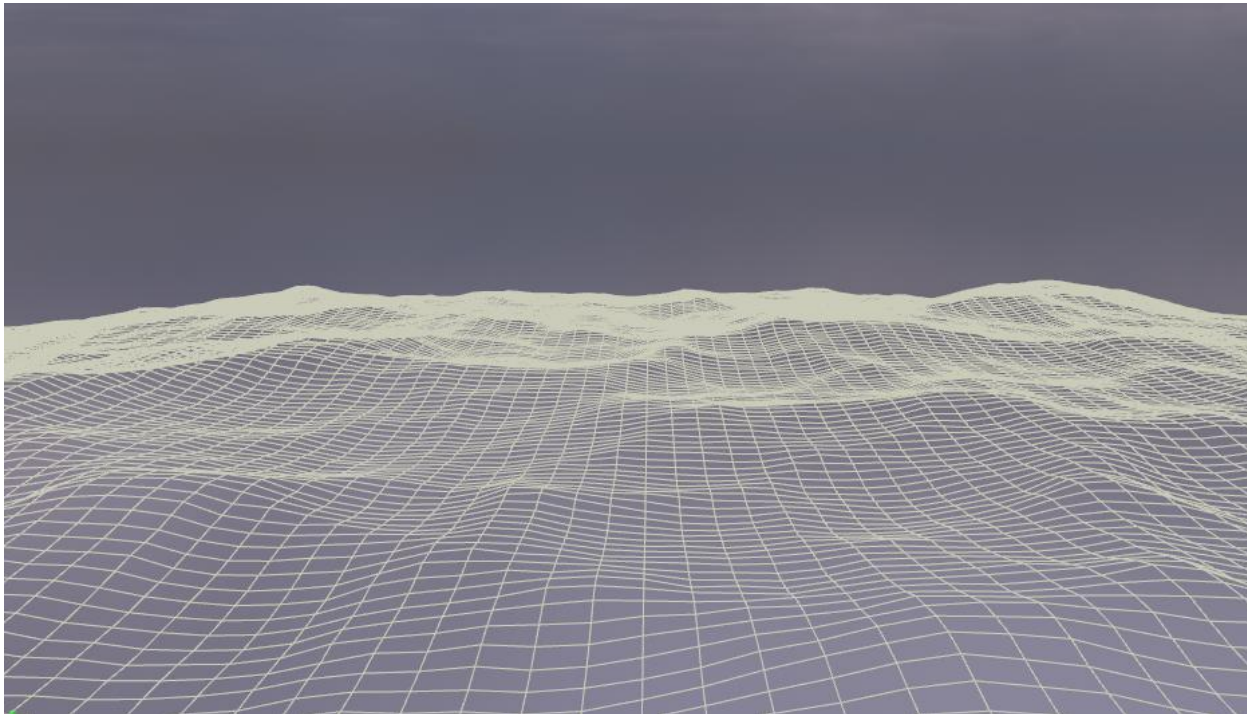
16 SEPTEMBER 2022

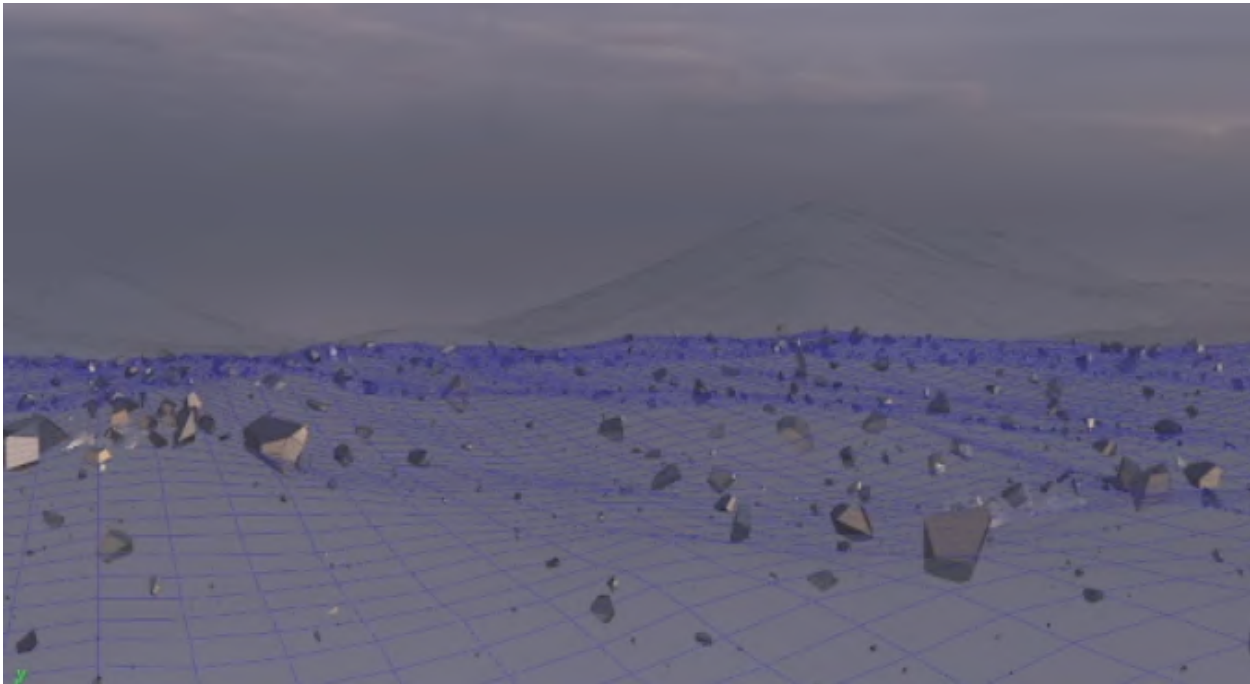
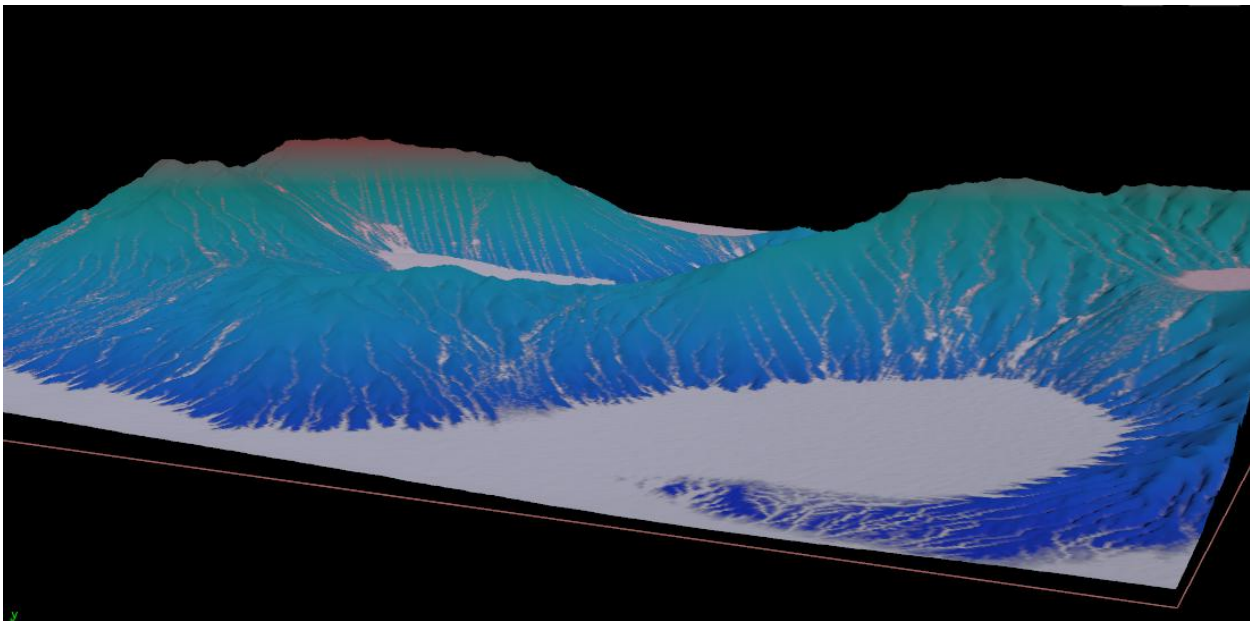
PAGE 03

Firstly I had to figure out how to grow crystals, with a help from our teacher Andrew the system was developed where crystals grow at different times and have different heights or thickness. I also modeled few options for the top parts.



Then I added the terrain, and modeled the mountains. Scattered rocks, which in the end had to look like ice pieces.





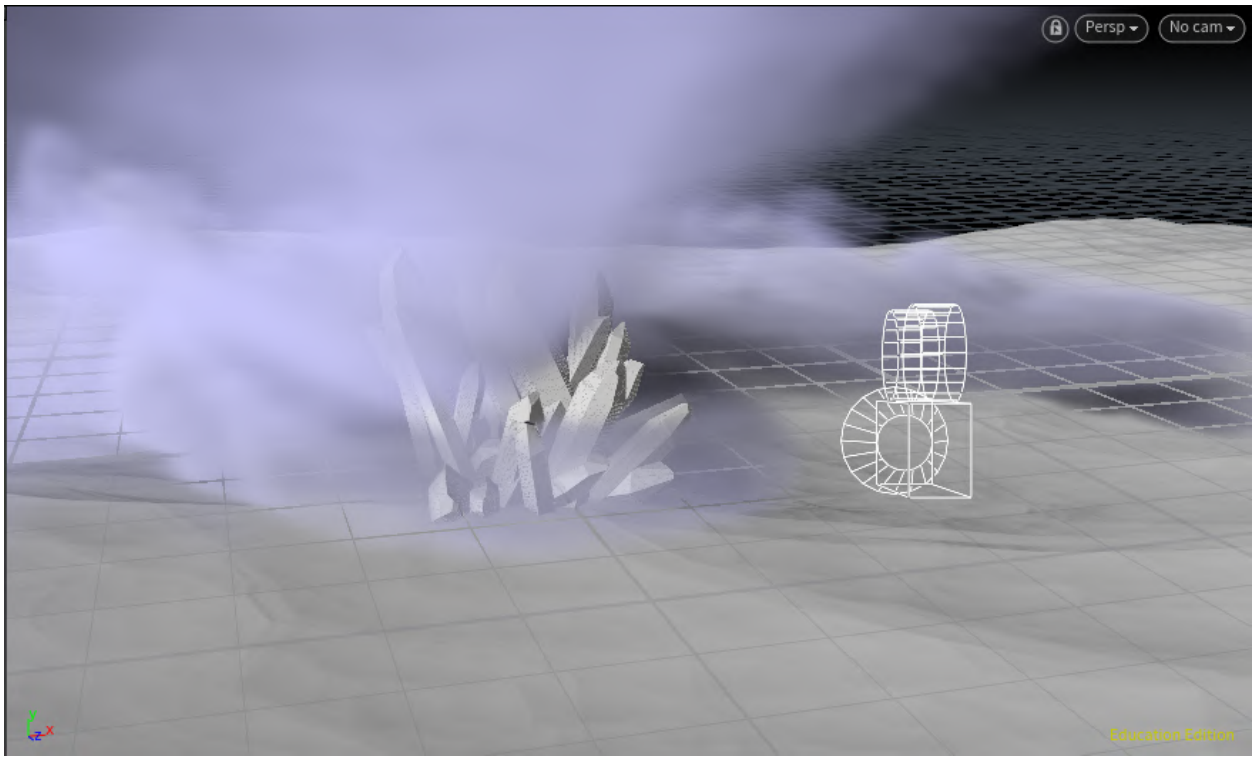
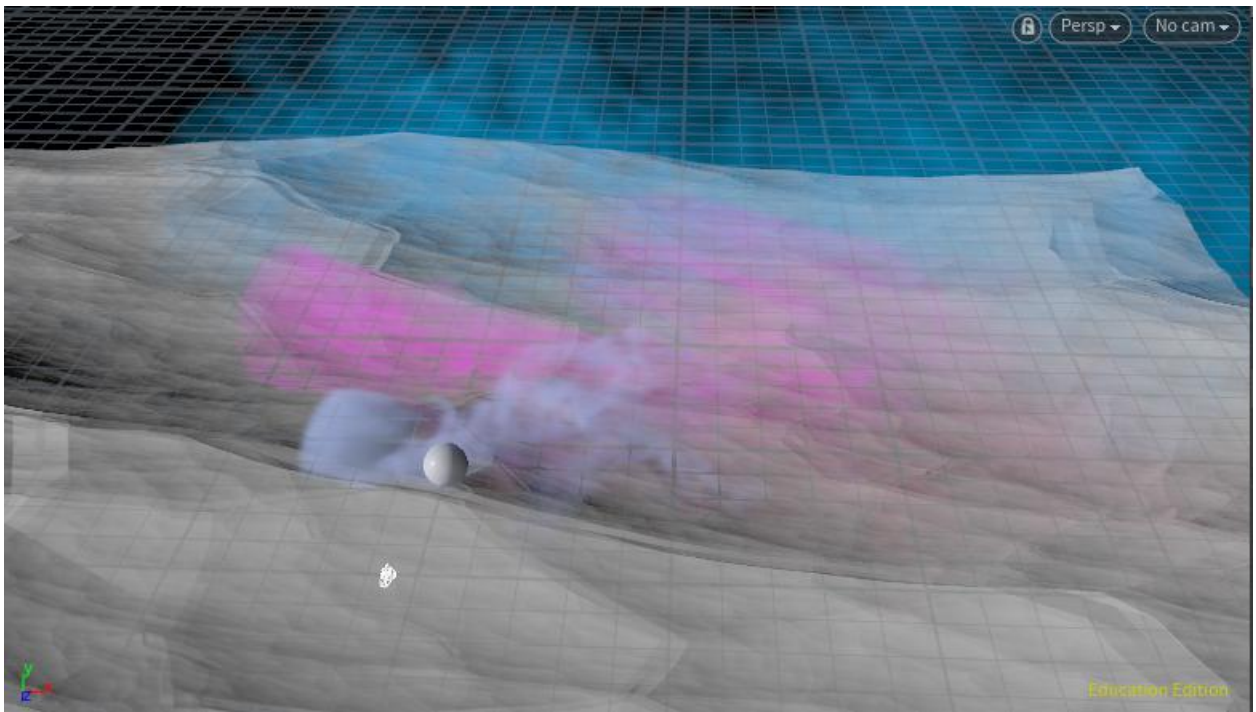
## SIMULATIONS

### FOG

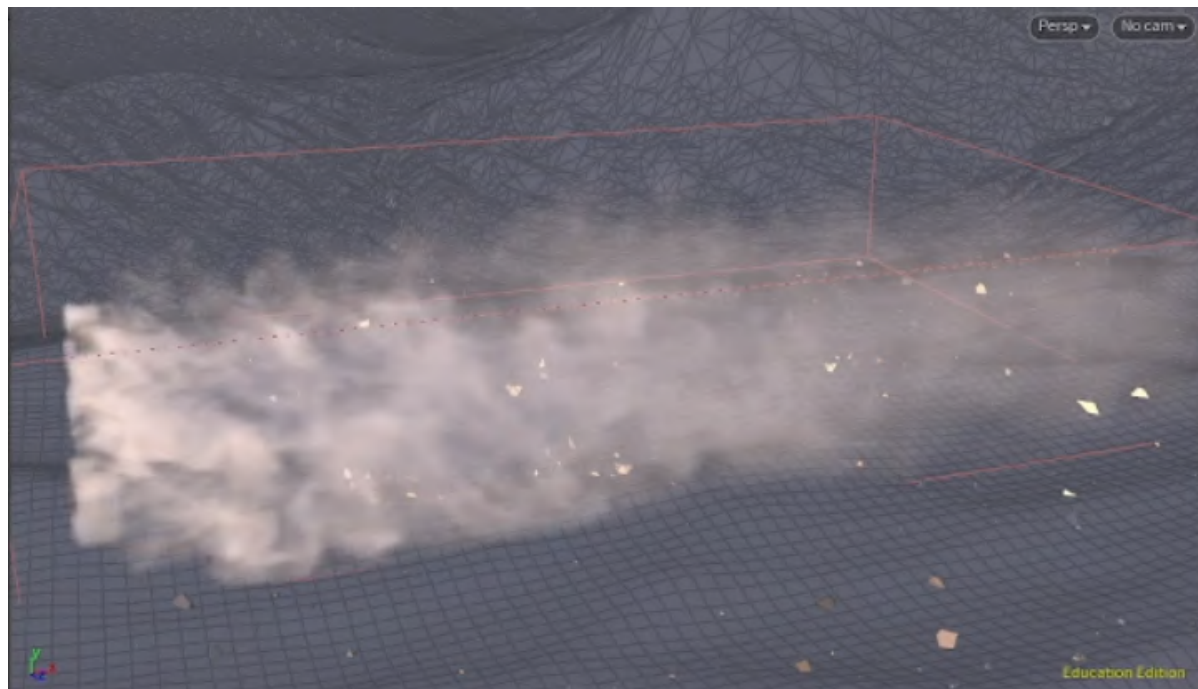
For the simulation part, I chose to have a misty fog and I think this part took the most time, because working with smoke can be challenging due to its specific formation characteristics.

Here are a few of my failed attempts:



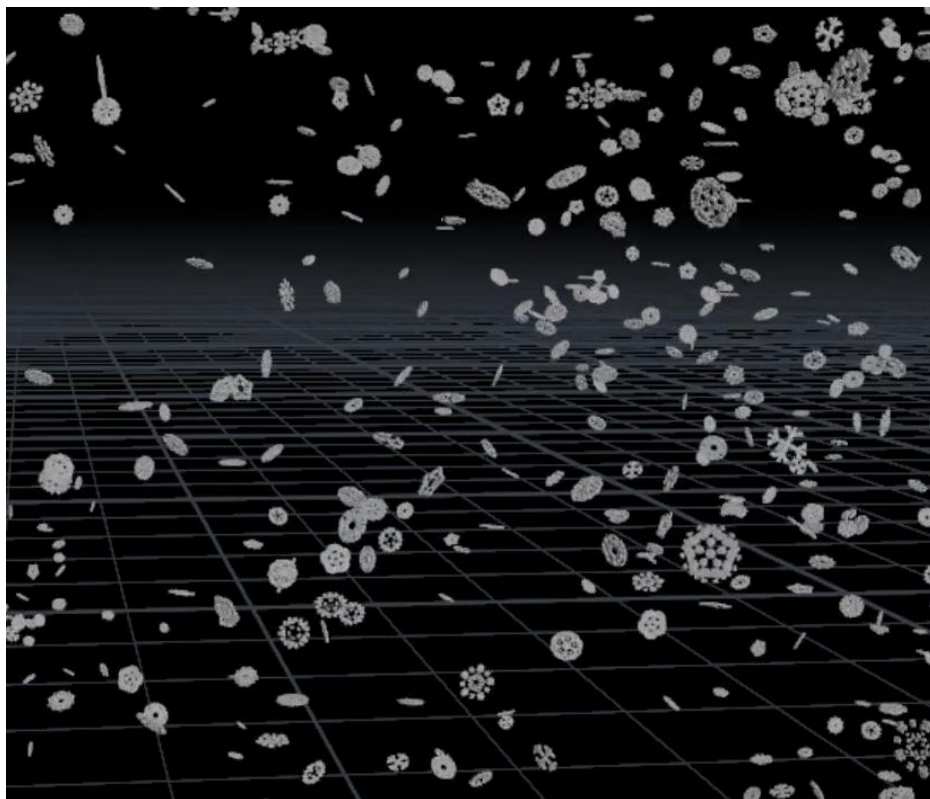


But finally with some help I was able to get this subtle yet a little bit stormy, snowy looking fog.

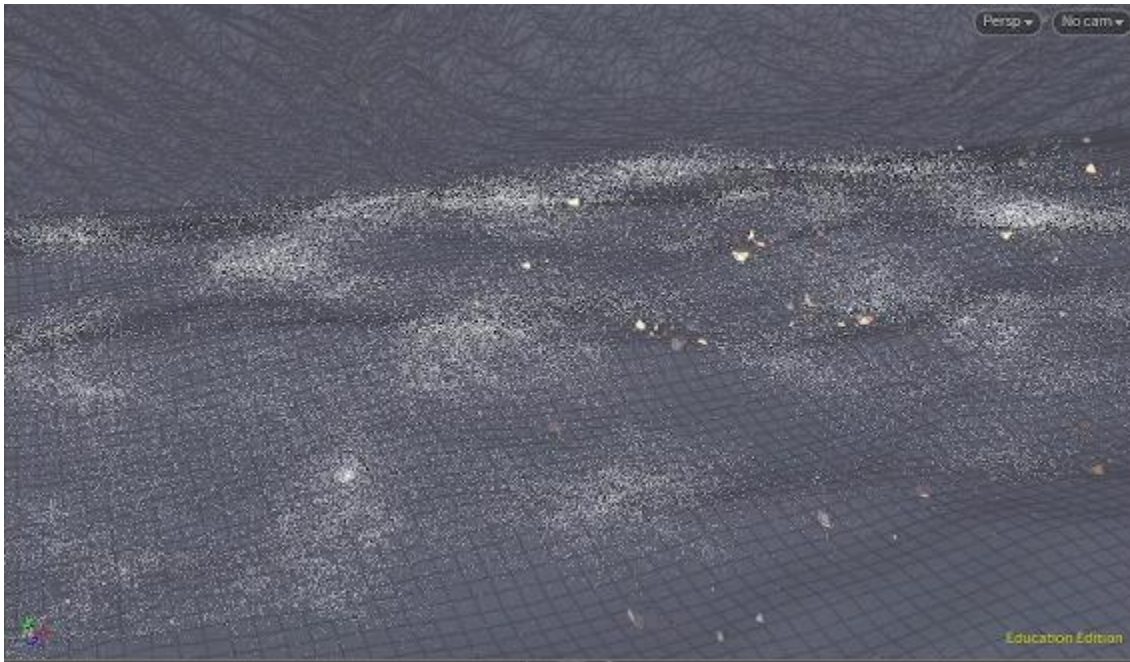


## SNOW

Then the particle system was added to get a snowy look, of course here as well, at first I had few attempts that were not exactly what I needed because these detailed ornaments would not be visible in a snowy fog.



But then finally the windy snowy particles were added that worked pretty well together with the fog.



#### Challenges:

Due to it's complexity I had to eliminate the frost spread sim from my project, but I do not feel sad about it, because I believe that these scattered rocks fill up the ground space quite nicely and correlates well with the crystals.





# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene

**2. Dimitris Kyriakou**

3. Jüri Lääs

4. Karol Wójcicki

5. Marisa Naemi Kristin Liebner

6. Neele Katharina Siemann

7. Paraskevi Malamatenia

8. Patrick Baitinger

9. Sergio Esteban Casilimas

10. Sylvia Rybak

11. Victor Fredrik Weijers

12. Yuliia Silkina

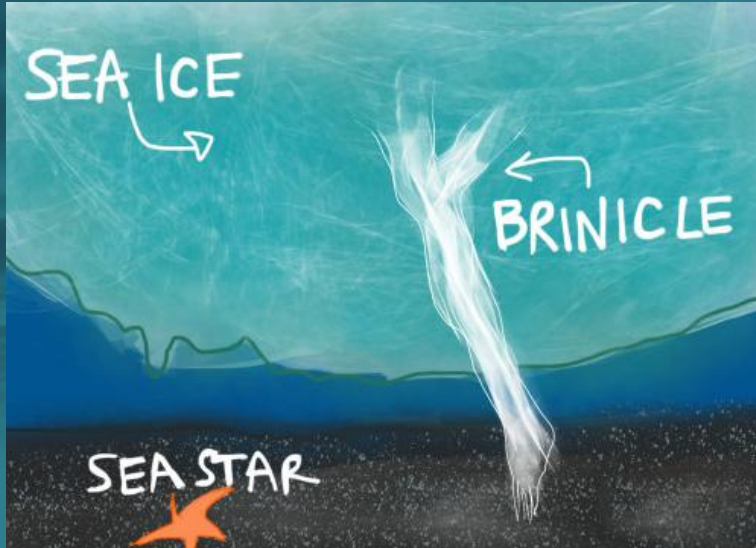


Creative  
Europe  
**MEDIA**

# Brinicle

Finger of death

Dimitris Kyriakou



# The Concept

## Simulations:

- Brinicle
- Starfish (Crowd Simulation)
- Frozen ground

## Environment:

- Ground
- Rocks
- Sea Ice



# The Simulations

## Brinicle:

- Particles
- Volume



Generate lines

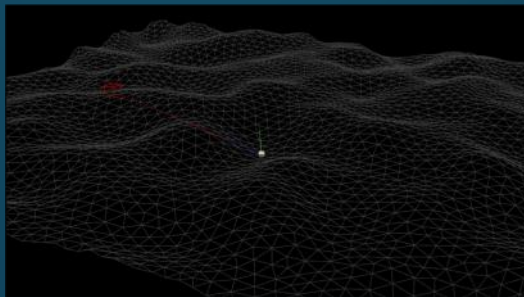
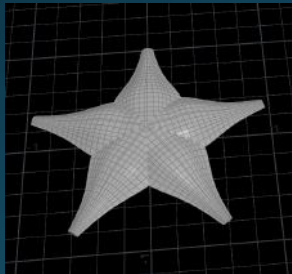
Particles following lines

Convert particles to volume

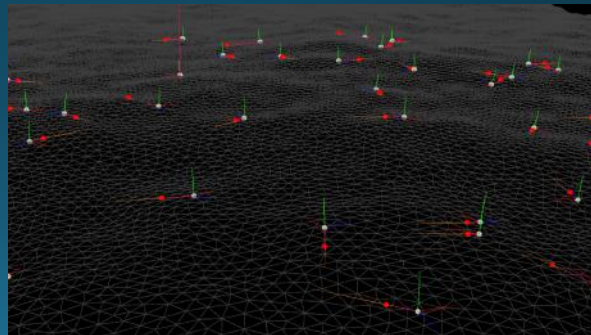
# The Simulations

## Starfish:

- Crowd Simulation
- Procedural Modeling



Calculating target distance and direction agent moving according to terrain

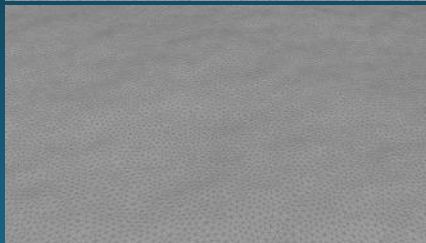
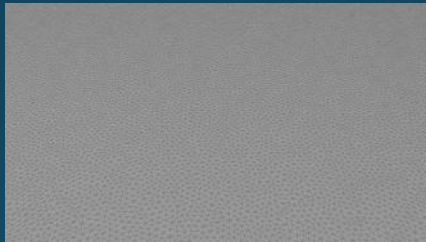
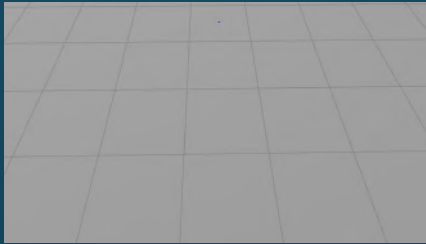


Multiple agents up vector adjusted to terrain  
agents spinning

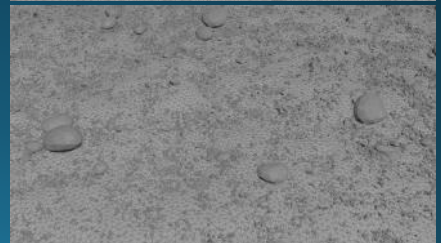
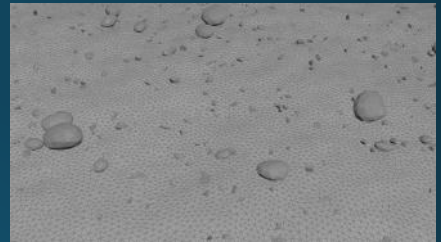
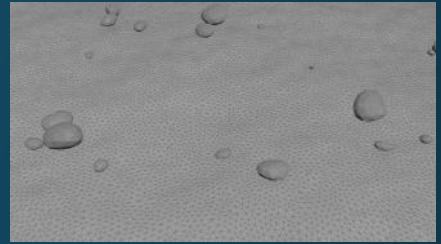
# The Simulations

## Rocks:

- Rigid Body



Terrain



Scatter rocks: large, medium, small



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene

2. Dimitris Kyriakou

**3. Jüri Lääs**

4. Karol Wójcicki

5. Marisa Naemi Kristin Liebner

6. Neele Katharina Siemann

7. Paraskevi Malamatenia

8. Patrick Baitinger

9. Sergio Esteban Casilimas

10. Sylvia Rybak

11. Victor Fredrik Weijers

12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

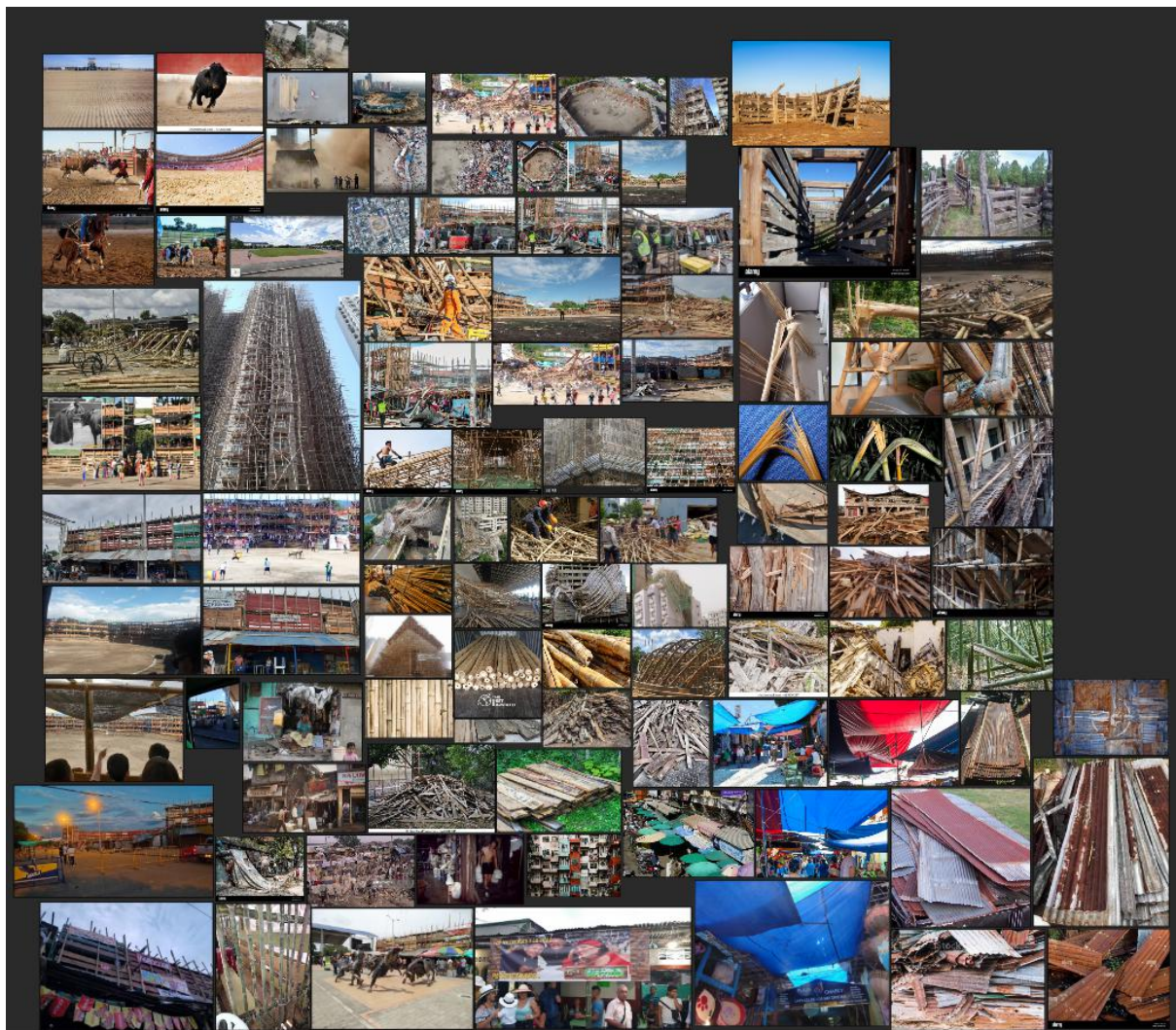
# Bull ring destruction in Houdini

## Case study

by

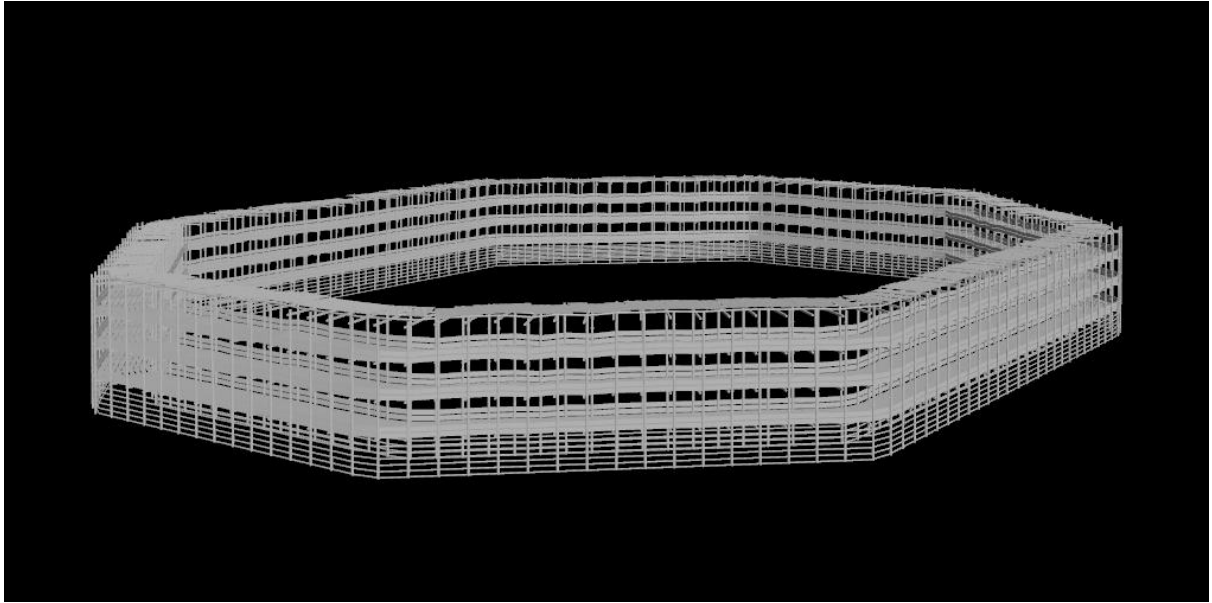
Jüri Lääs

Once you have figured out the concept you wish to realise, every visual effect begins with gathering references. Because I wanted to do a destruction shot, I began with spending a week looking through a substantial amount of videos and pictures of different structures falling apart.

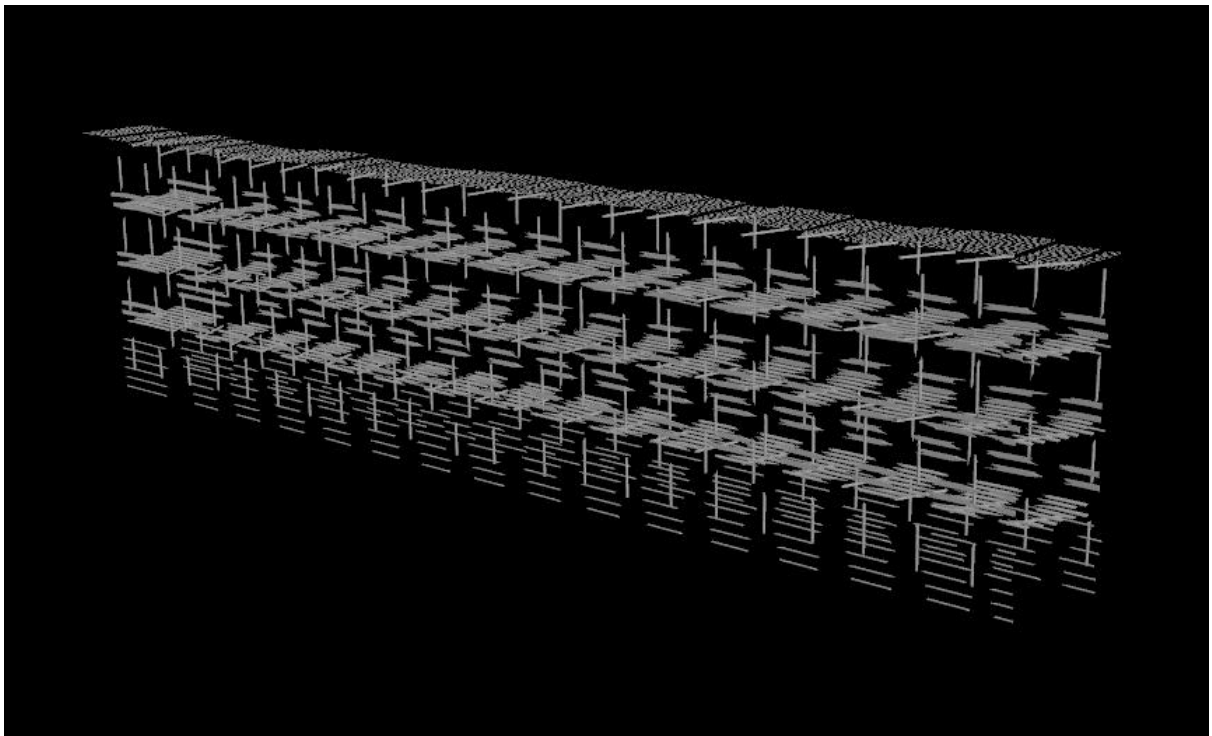


After collecting the right reference and wrapping my head around the objectives I realistically could achieve during the 12 weeks, I started to realise it in Houdini.

The first step was to produce a 3D model of the bull ring in order to destroy it later on.

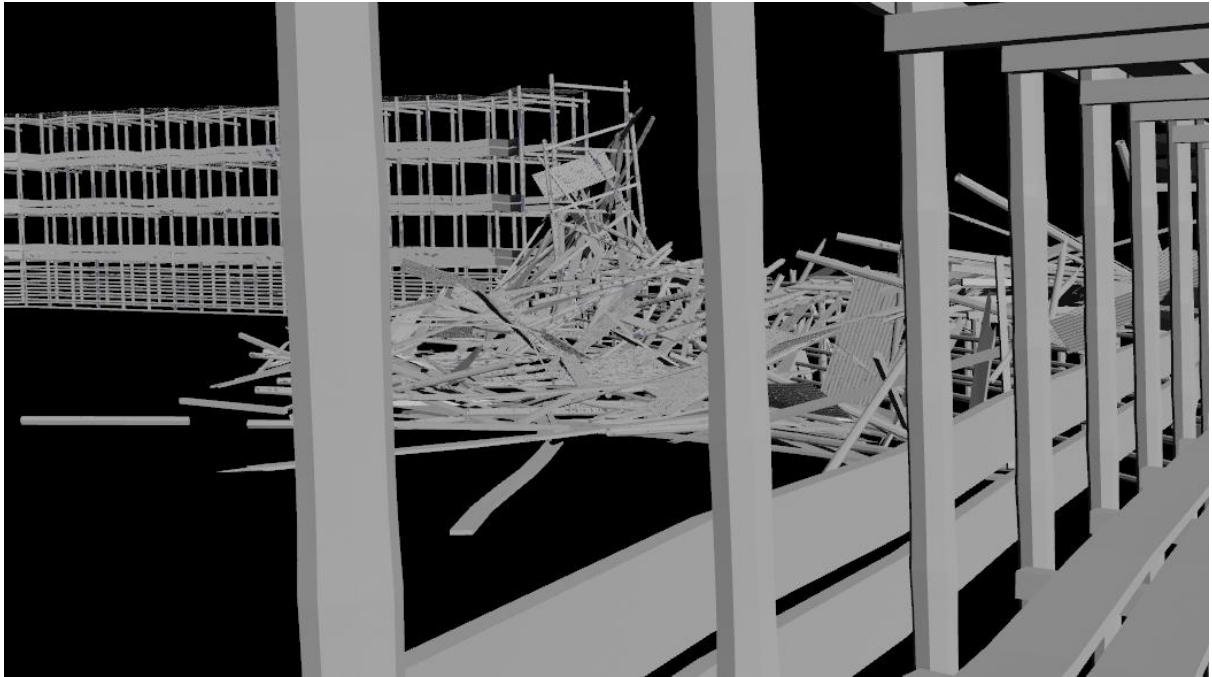


Then came the time to break it into pieces.





After that and a few other steps it was ready to be simulated/destroyed.



Then came the time to add in all the extra elements, such as small pieces of debris and dust.

The extra elements are better viewed as a rendered image and that's exactly what I have done.

Render without extras.



Render with small to medium debris.



Render with added flying sand from the impact with the ground.



All the layers together with added dust.



That is all, thank you for the opportunity to share my work with you!



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs

### **4. Karol Wójcicki**

5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**



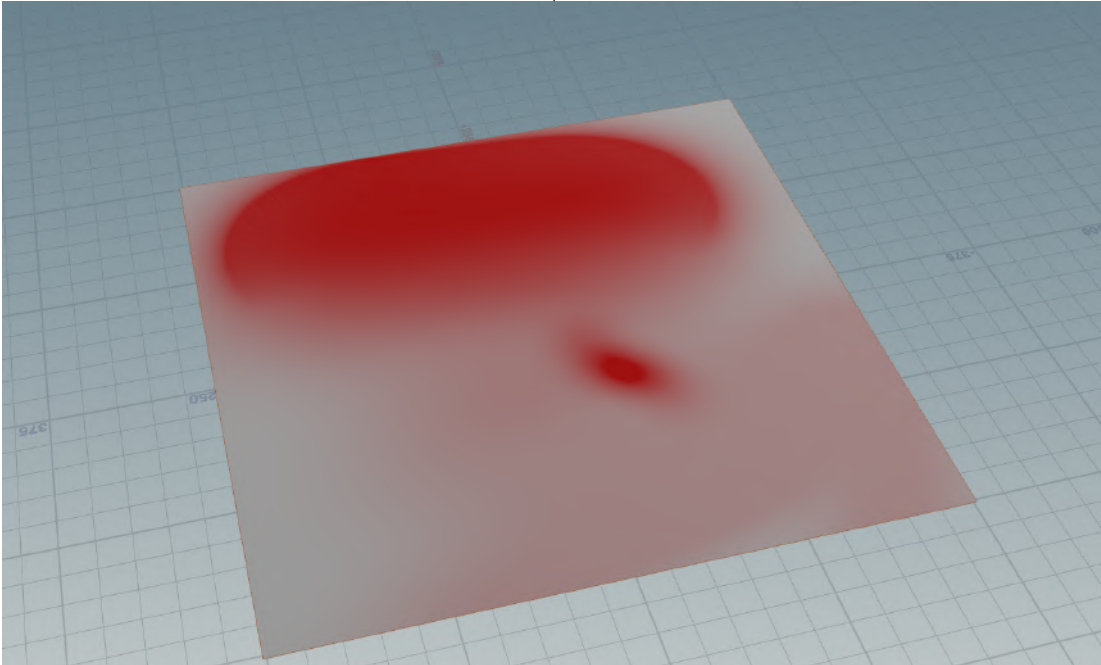
# Untitled sheep project

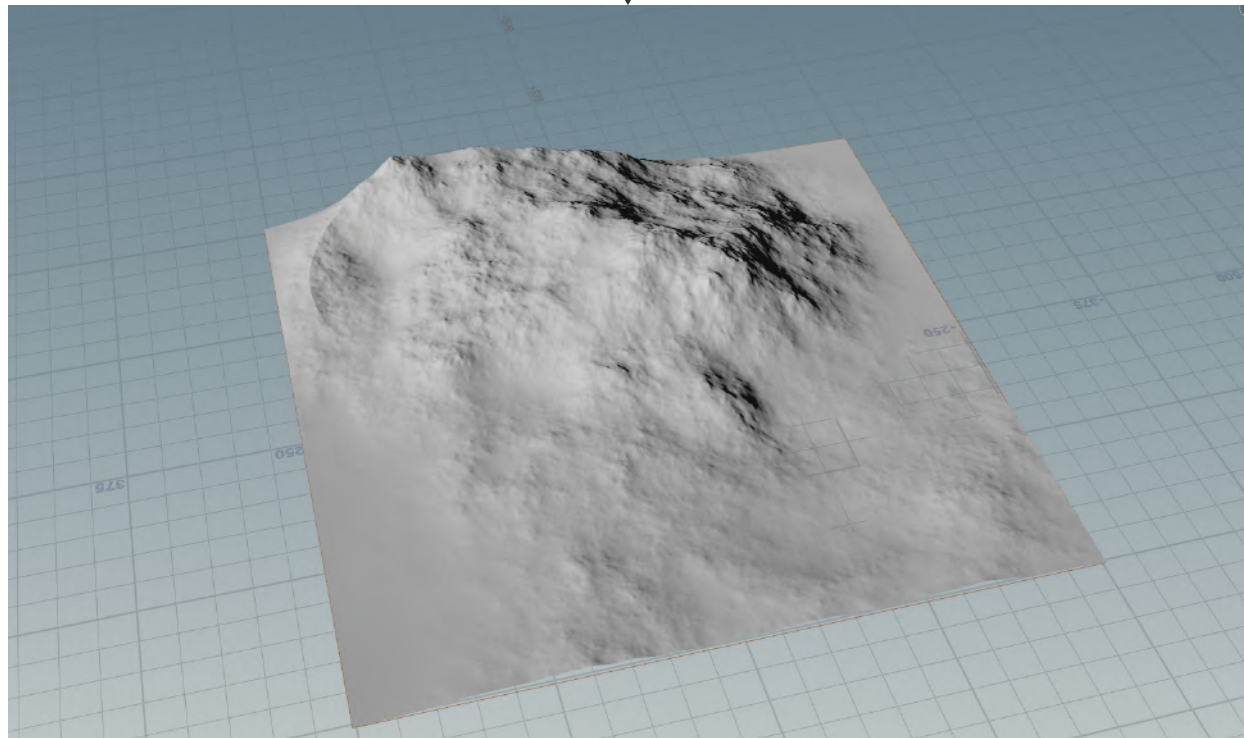


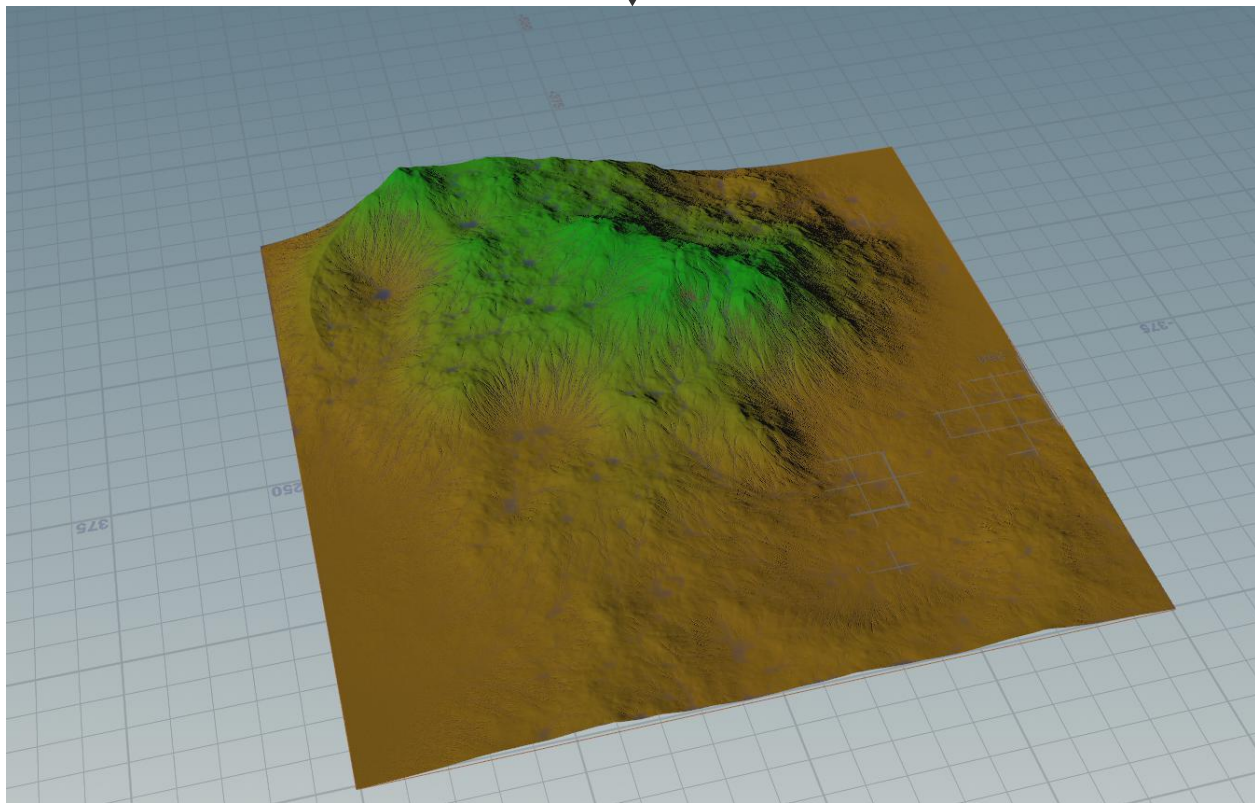
**Inspiration**



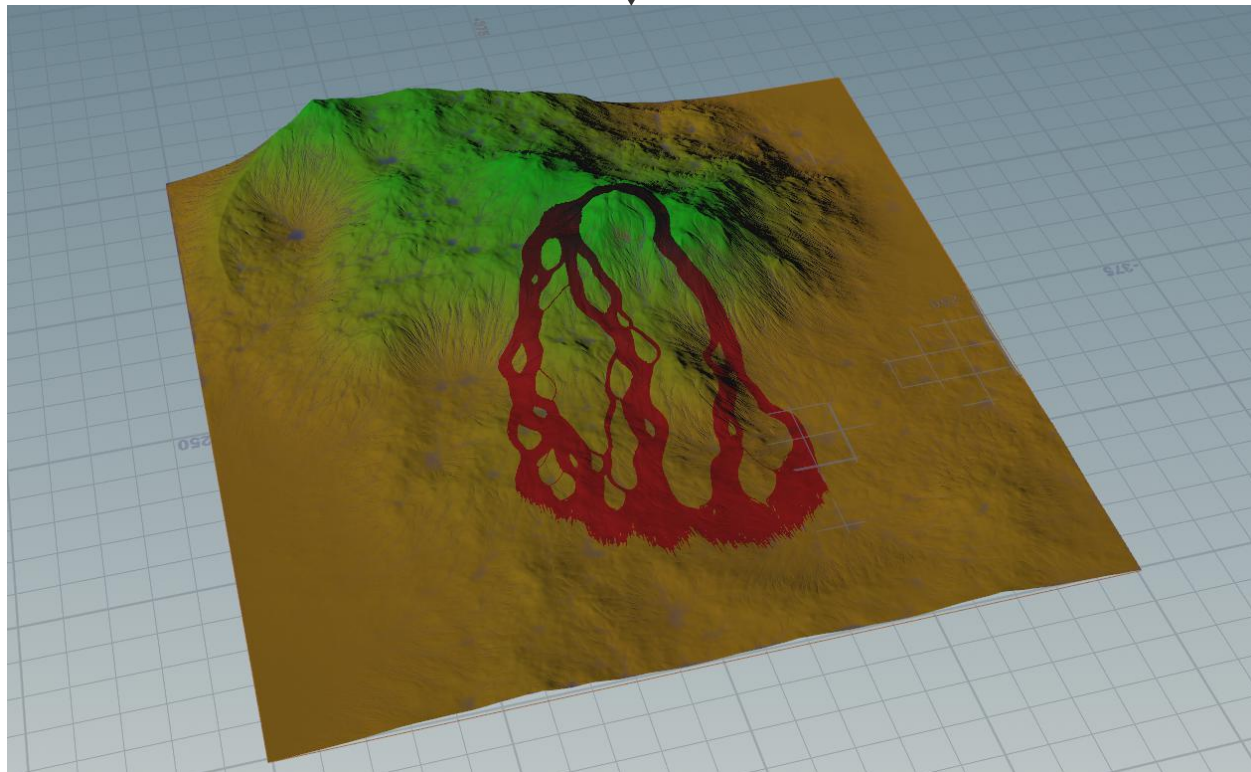
**Landscape Development**



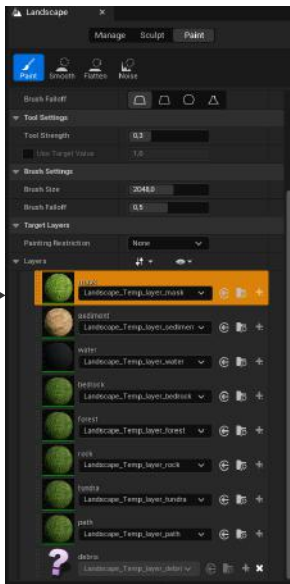




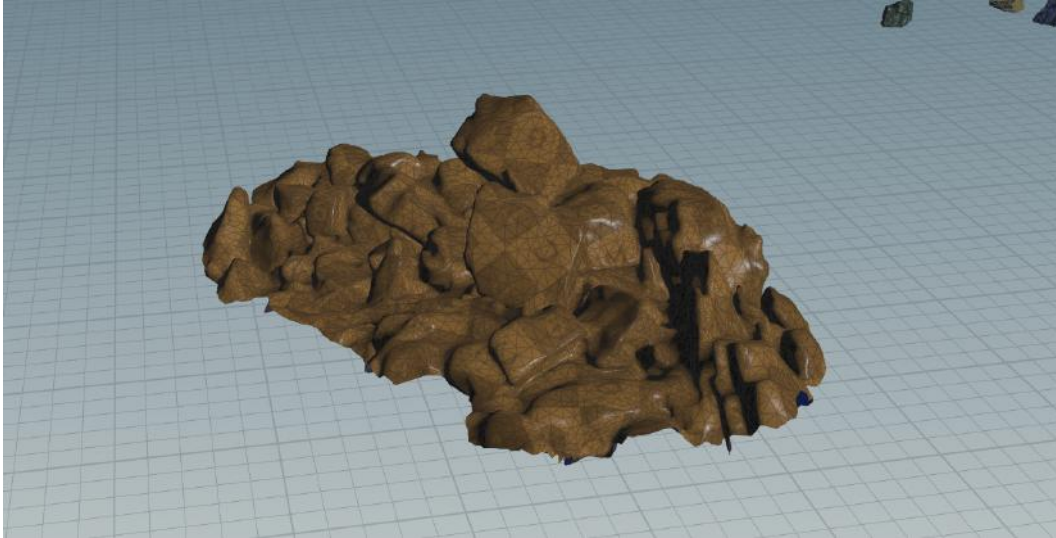


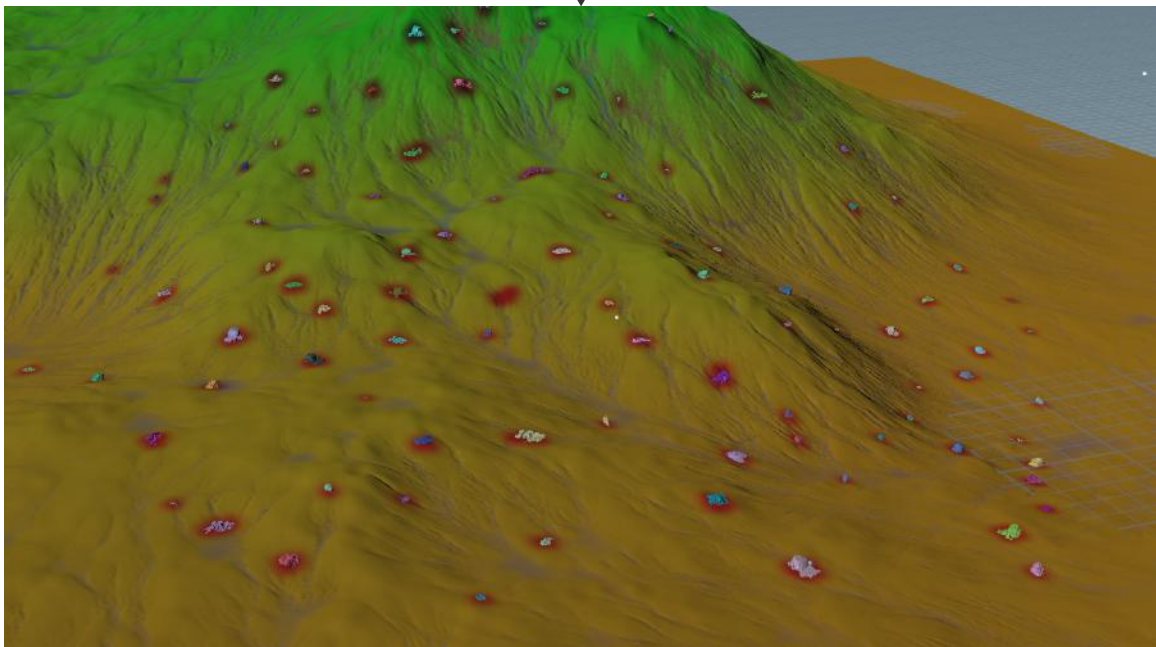


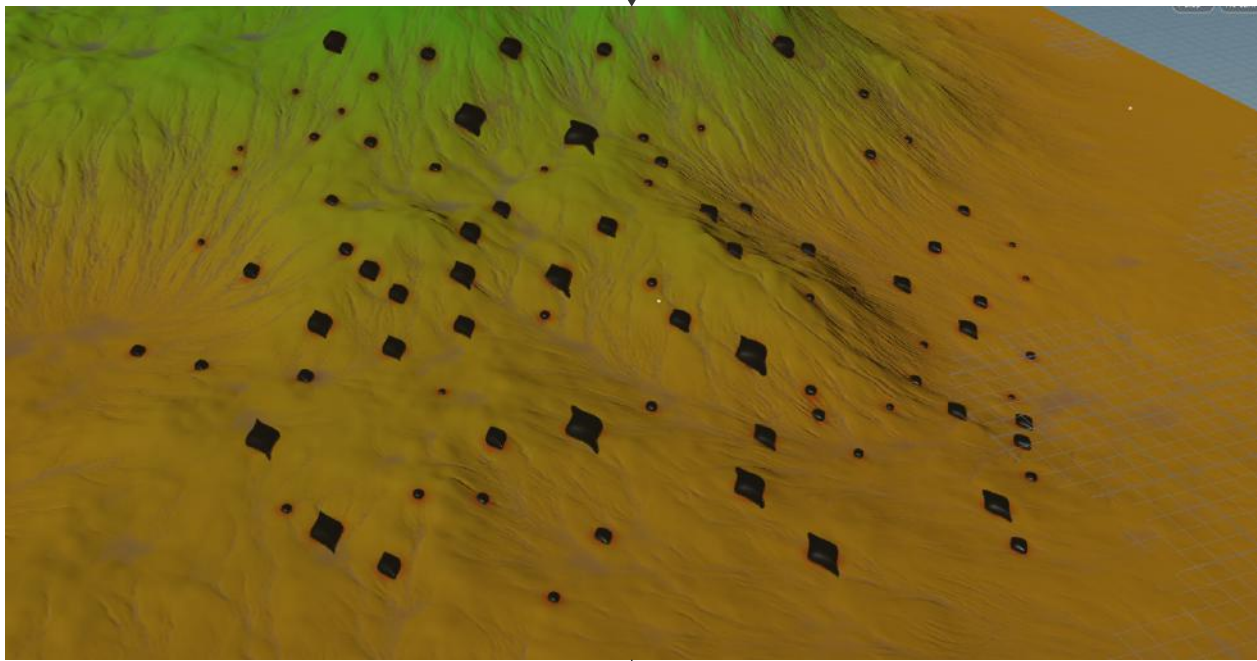




## Buildings

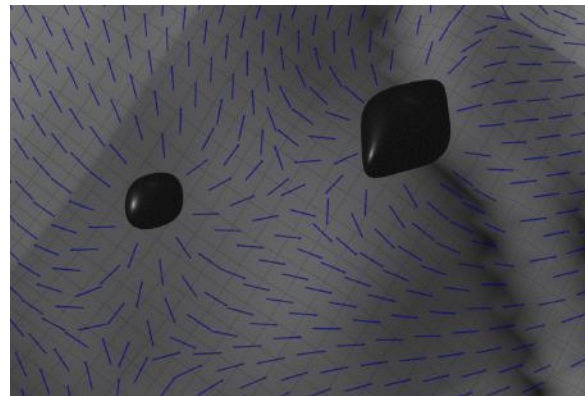
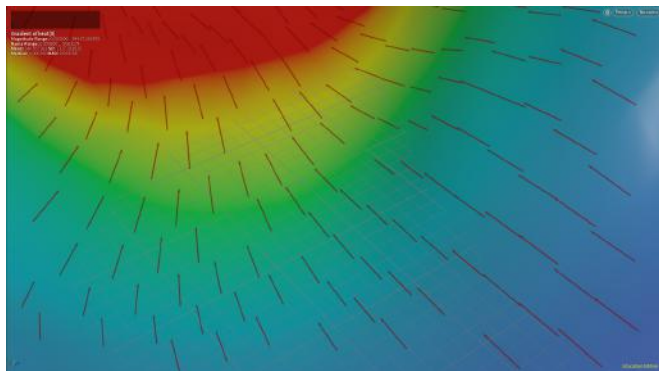
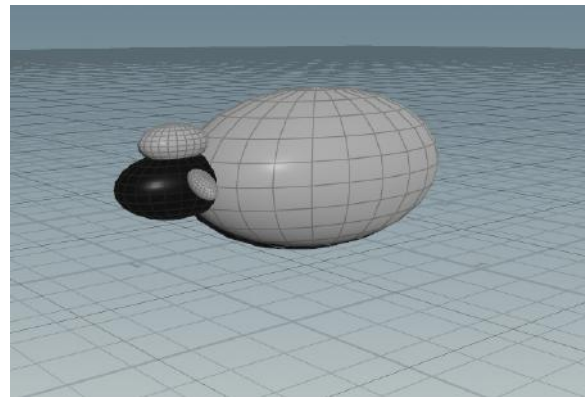
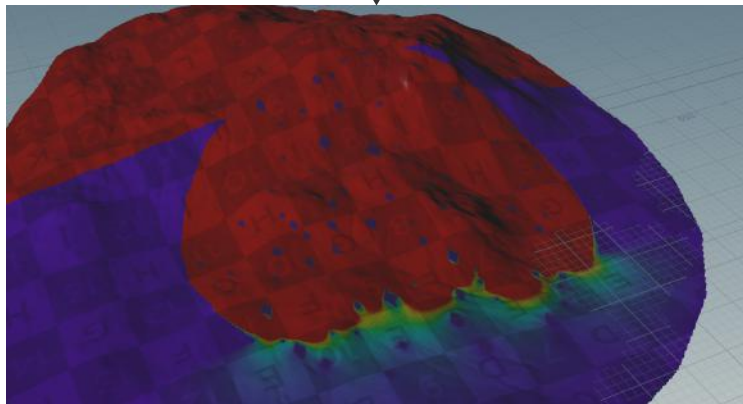


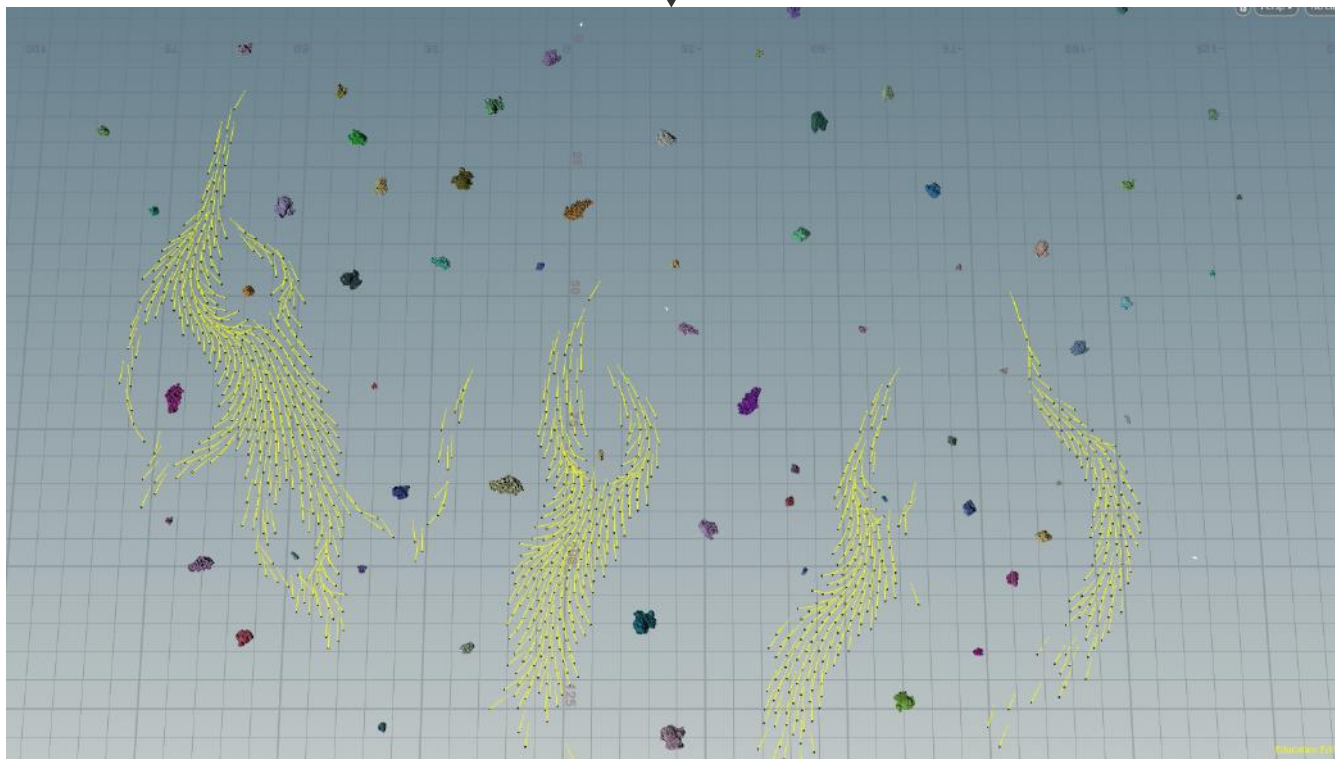




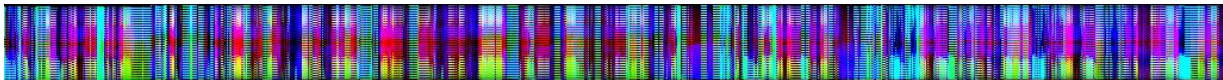


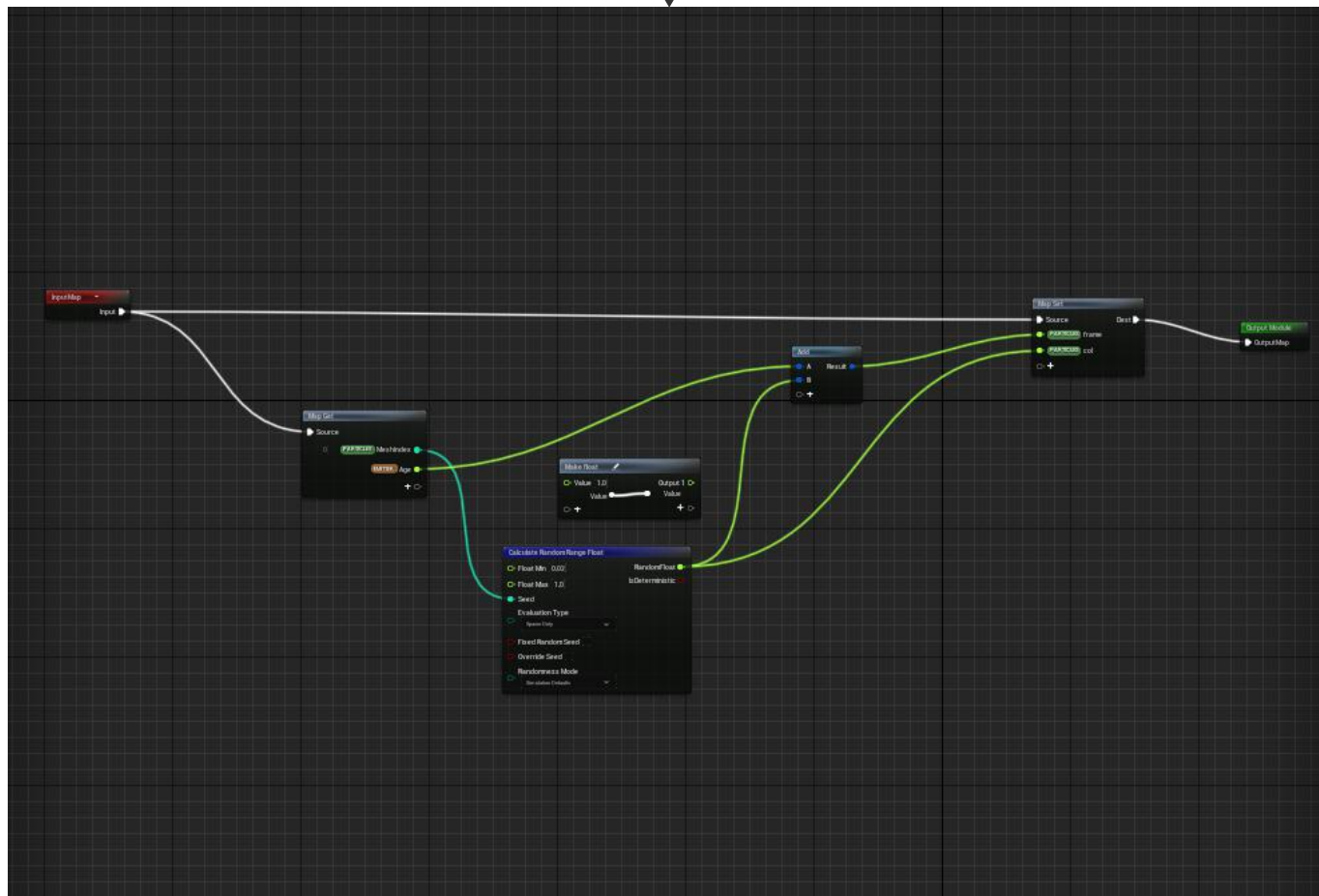
Sheep behavior





Bring it all into Unreal Engine











**Thank you!**

# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki

### **5. Marisa Naemi Kristin Liebner**

6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

# BUBBLE SAND FASHION

## BACKGROUND

I previously studied fashion design and during my studies became really interested in digital fashion as a sustainable alternative to physical fashion. I used digital toiling as a more sustainable alternative to physical design development and in this process became really interested in fabric simulation. My previous work focused on the interaction of humanity and technology in fashion and textiles and challenged the categories of the natural and the artificial.



Initial collection inspiration



## RESEARCH AND IDEA DEVELOPMENT

My initial idea was to create a garment based on one of my collection looks, that was made of sustainable yarn fabric and sand infused bioplastic. I invented a vegetable starch bioplastic that I later mixed with sand and created biodegradable and compostable jewellery. For this

project I combined the two elements of digital fashion and sustainable materials and added the sand falling out of the garment. For this I researched the falling behaviour of sand as well as its interaction with different materials such as cotton fabric, knitted wool



Sand interaction with materials



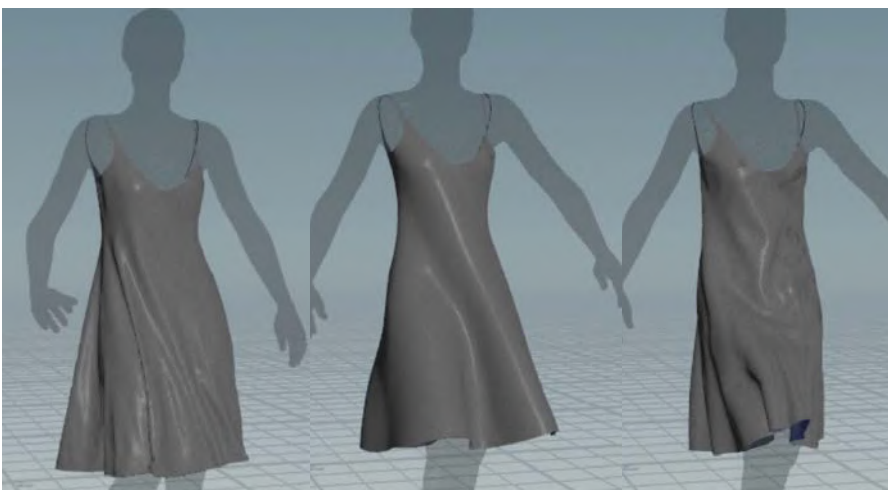
and see-through thin plastic sheets. The final look I created for this project was based on a purple yarn jacket and socks based on trousers from my collection look six as well as a skirt based on a corset skirt from look one.



Compostable bioplastic jewellery

## GARMENT AND PLASTIC

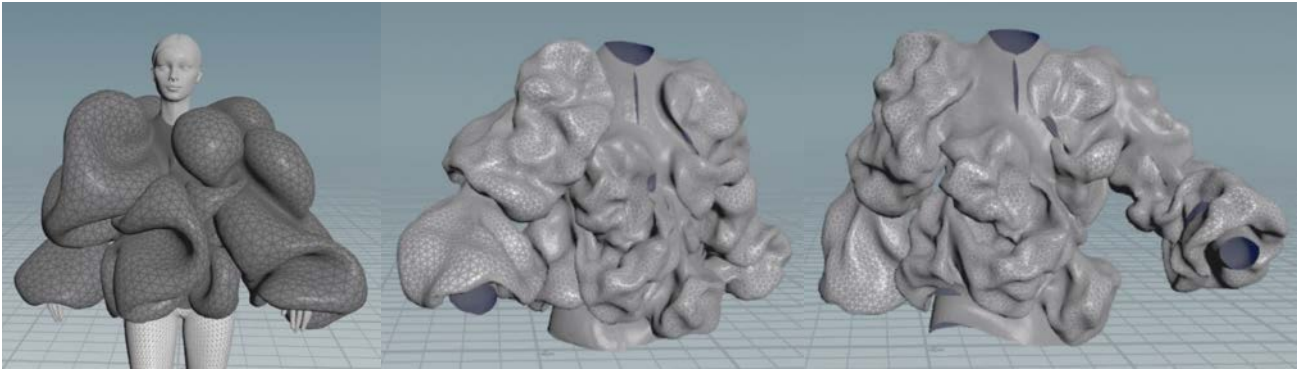
I started out this project looking into vellum in Houdini in general. I tried the vellum drape node and welding seams of the pattern pieces I created in Houdini this way. I had some trouble making this work as the edge lengths did not match up correctly and created lots of unwanted creases. I found that the better way for me was to create the basic garments in Clo and export them as welded objs into Houdini. I then developed the vellum cloth and solver settings to get the right vellum properties for different material settings such as silk-like, bioplastic/leather, conventional thin plastic and mid-weight denim. I used all those settings for the final shot as I had three different garments with two different settings each for the frames before and after the bubbles



Testing vellum material properties and settings: silk, leather, plastic

form so six settings in total were used. I found that some textile settings only worked on the less deformed geometry, as in the garments before the bubbles are starting to build and not on the bubbles forming with high velocity.

The bubbles forming itself is due to animating

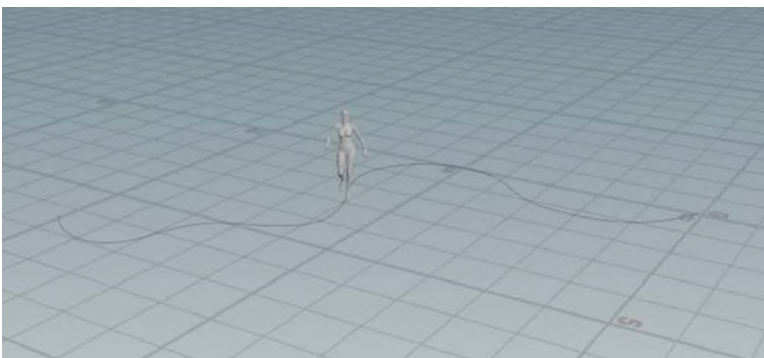


Leather and plastic vellum material on the first, basic garment

the restlength attribute that goes into the constraints input of the vellum solver based on a restlength amplitude attribute I created in a pointvop. I developed these settings because in the beginning I just used a simple aanoise as the rest\_length\_amp map, but found that this caused problems where the rest\_length\_amp was greater than zero on the seam edges of the garment. So I ended up using an attribute mask to paint on where and how the bubbles should form as this gave me more control over the placements of the bubbles over the seams as the inner seams seemed to appear working well with rest\_length\_amps of a lower value in relation to the higher total values, so if I used 4 on the outside of the bubbles the inner seams worked with values lower than 2.

Due to the way the bubbles on the jacket were created they were starting to become really low res. I tried to res up my garments before the solver to get a better look but the simulations were not looking good this way. In the end I used the clothdeform nodes to get higher resolution overall as I was unable to remesh the bubbles after forming due to the uvs and topology changing as well.

For my walking character I used kinefx to adjust the walk cycles and chops to smooth out the walking animation as well as trying to get the more plastic looking material vellum setting to blur enough to make them less jagged which worked in some parts.



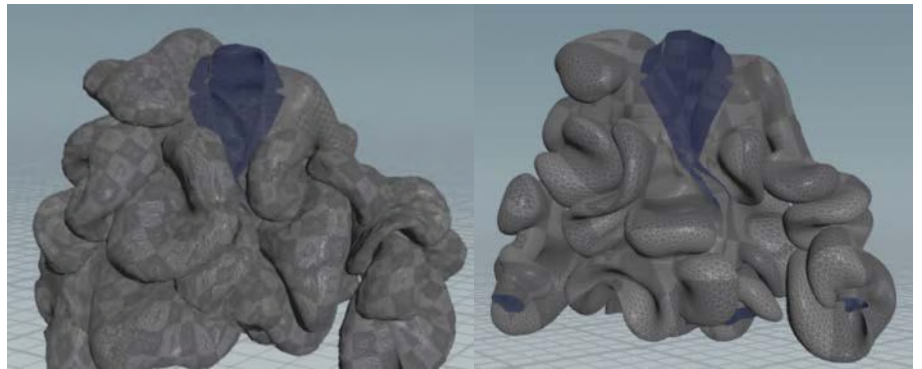
Path deformed walk animation

I used mixed mtlx shaders for the stripey garment and plastic

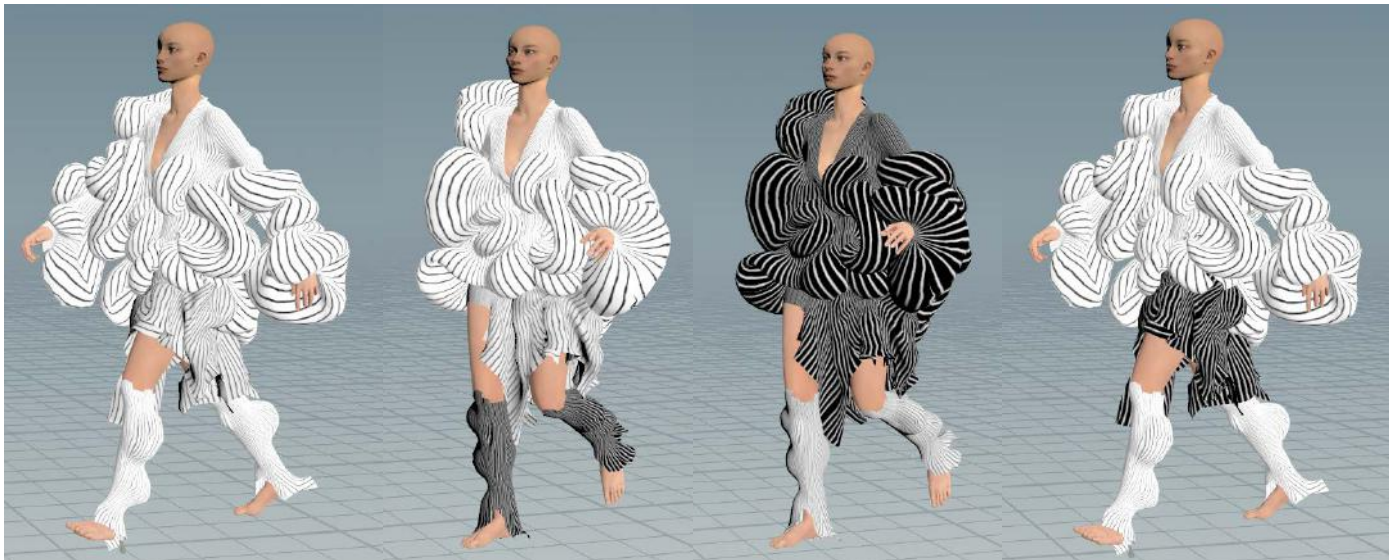
surfaces based on a painted attribute mask on the outside of the bubbles. I tested using the principled shader with different settings (on the specular roughness, reflectivity, transparency, and opacity) by bringing in mask attributes, but mtlx achieved better plastic looking materials while the principled shader was a bit better in displaying the fabric normal maps. I also did a colour test of the garments itself to put the character into the focus of the shot.



Issues with rest length amp on edge seams and intersecting geo

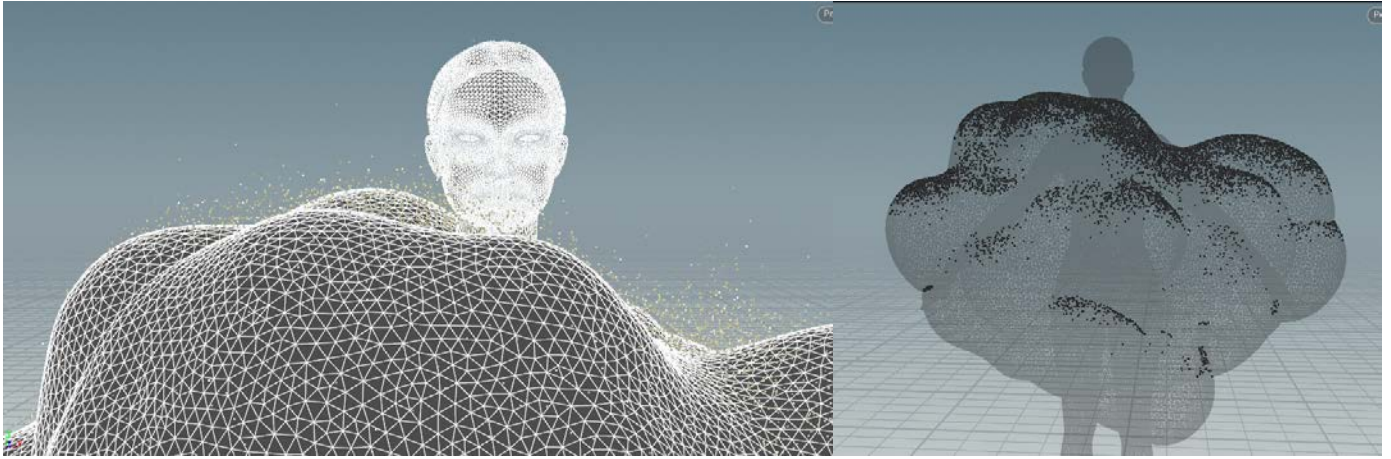


Chops jitter smooth



Colour tests

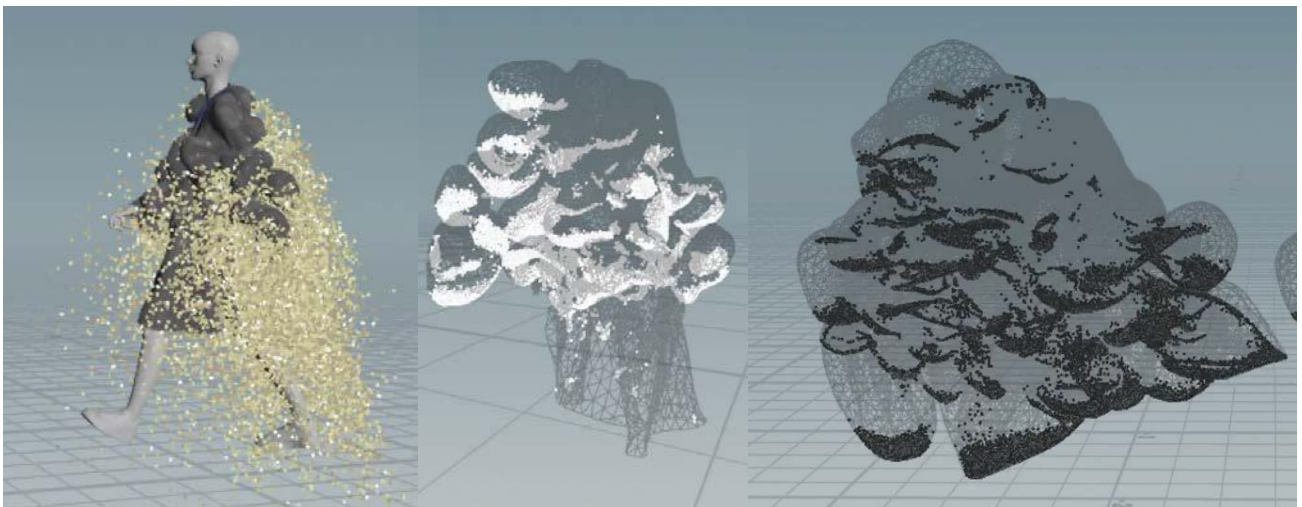




Initial sand falling and sand distribution on the upwards facing surfaces

## SAND

I developed the sand element of this project while working in the garments. I experimented with different types of sand like vellum grains for interactions and sand release from the garment and in the end I used particles as sand. I worked with a lot of different settings and parameters on particles by just dropping them on the ground and objects to get differently behaving particles that depict dry to very clumped sand.



Solver sand release and sand distribution scatter based on dot product

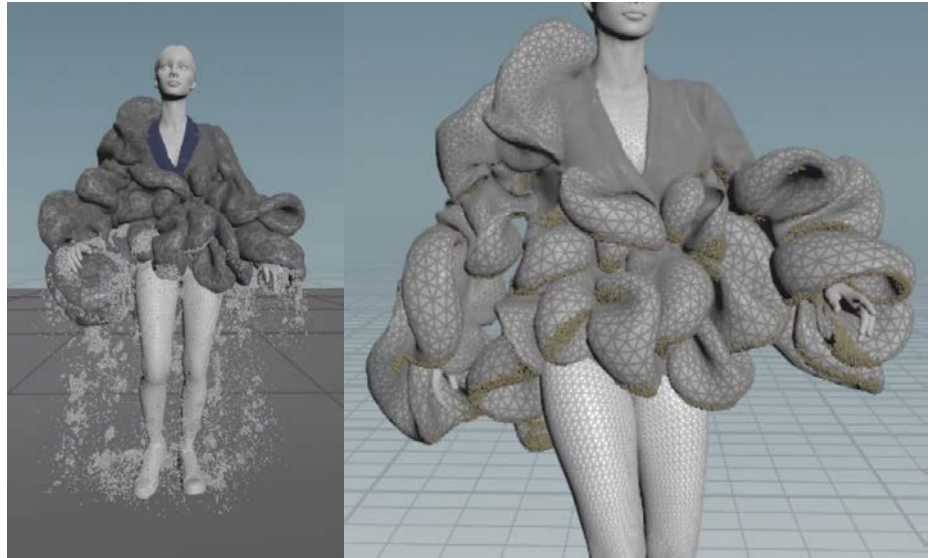
The sand used for the final render is made up of three elements: 1) sand that is built up in the bubbles on the outside where the plastic is formed as well, 2) sand that is released from these bubbles based on the velocity of the particle, 3) sand that is gathering on top of the bubbles and in the creases of the garment (dot product).

As mentioned before I used the clothdeform to deform the sand that was projected to gather on the jacket based on the dot product (3) and used the clothdeform method to

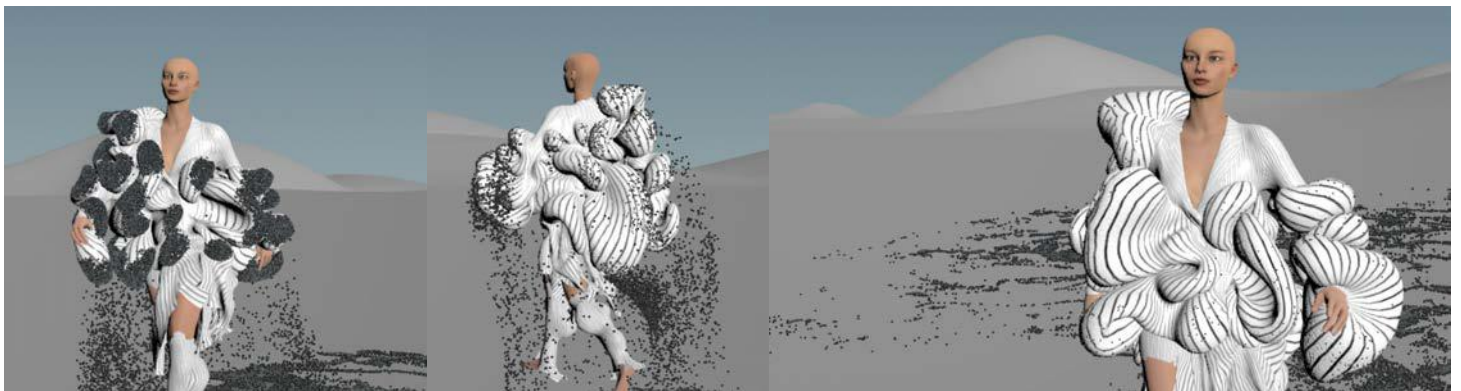


have the building up sand and slow release over the course of the walk (1). For this I used a time shifted particle scatter in a foreach loop, to only scatter this once and not have changing particles as this caused a lot of jittering issues.

I used a simple principled shader with a colour variance on the particles by creating a randomised `clo_var` attribute to get the colour distribution based on my colour ramp and used velocity blur on the final Karma render to get the particles looking realistic.



Vellum sand release from garment



Sand release tests



Render tests

# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner

### **6. Neele Katharina Siemann**

7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

# PROJECT: CLOUDSCAPE

Creative Simulation Course 2022 – Neele Katharina Siemann

## IDEA

Based on a scene of a personal project:

*"To escape from reality, a sick child imagines herself to be a superhero. She flies up into the clouds and breaks through the thick cloud layer, into a sunset, feeling completely free in her imagination."*

Since the whole sequence could not be realized within the given time, it was broken down into the main elements, which were:

- The creation of a sunset cloudscape consisting out of cumulus clouds
- The breakthrough of a character through the cloud layer
- A cape simulation for the flight

An additional challenge in this project was the usage of Unreal and real-time rendering. Real-time rendering with Game Engines is a fairly new and still developing technology in the film industry. How to implement volumetric simulations into the Engine, how to achieve a cinematic look and assess what the current state of the technology is, were some further goals.

Further steps of the pre-production:

- Assembling a lot of cloud references
- Gathering an understanding on how clouds form in reality

## SIMULATION

in HoudiniFX 19.5

The *clouds* were made by simulating multiple volumes with customized pyro solvers. For this, custom scalar and vector fields were used to represent real physical elements that influence the look of cumulus clouds e.g. wind, humidity, temperature, etc. By freezing and then stacking multiple pyro simulations over each other, a realistic cumulus cloud look could be achieved.

Since the goal was to render the scene with Unreal in real-time, it was really important to make the volumes as performant as possible. For this, OpenVDB Volumes were used, instead of the standard Houdini Volumes, which are more optimized.

Furthermore, the static clouds were packed as instanced before copying them on procedurally created points, creating the cloudscape.

The *breakthrough* was made by using one of the static clouds and creating a collision in an additional pyro solver. The collision object had to be animated to create the needed point velocity. Furthermore, it had also to be converted into a VDB. The build-in human-mesh "Tommy" was used as a placeholder collision object. After the

animation was finalized with the Unreal 5 character, it was exported as a FBX2016 file and the breakthrough re-simulated in Houdini.

The *cap*es was made using two vellum simulations. The cloak was modelled procedurally and using a drape node, fitted onto the character. With masks, constraints were made around the neck and shoulders, holding the cape in place. The animation of the FBX was used as the point velocity, making the cloak move correctly in the “wind”.

## IMPLEMENTATION

in Unreal 5.0.2

For the *background* a cloud layer was exported as a volume texture. It stores information into 3D space, rather than 2D space, by slicing the volume into layers on an additional spatial axis. In UE5 a volume material was created and used to customize the volumetric cloud component.

To recreate the cloudscape in the *foreground*, which was already made in Houdini, an experimental Plug-In had to be used. It allowed the import of the VDBs into Unreal and even optimized them further by converting them into NanoVDBs. Moreover, it was possible to put the breakthrough simulation into sequencer and control the material properties over it.

For the character animation, the UE5 character “Quinn” was used. By creating a start, as well as an end pose and blending between them in the Sequencer, the animation was realized rather quickly. As mentioned before, the animation was exported as a FBX2016 file, to be used in Houdini to (re-)create the breakthrough and the cape simulations.

To implement the *vellum simulations* in real-time, they were exported as Vertex Animation Textures with the Labs Tools. These textures allow to recreate non-traditional animations in real-time by storing the data of each vertex. In Unreal a material had to be made, using these textures and applied to the cloak mesh, re-creating movement of the simulation with textures.

After having all the elements assembled, it was time to add post-production to the cameras inside Unreal, like bloom, chromatic aberration, depth of field, etc. Additionally, the sequence was set up to be in cinemascope and in 24FPS, to get a cinematic look. During the whole process, the balance between quality and performance had to be constantly evaluated. This was also true for the rendering of the EXRs out of the Movie Render, especially for the usage of console commands.



# CLOUDSCAPE

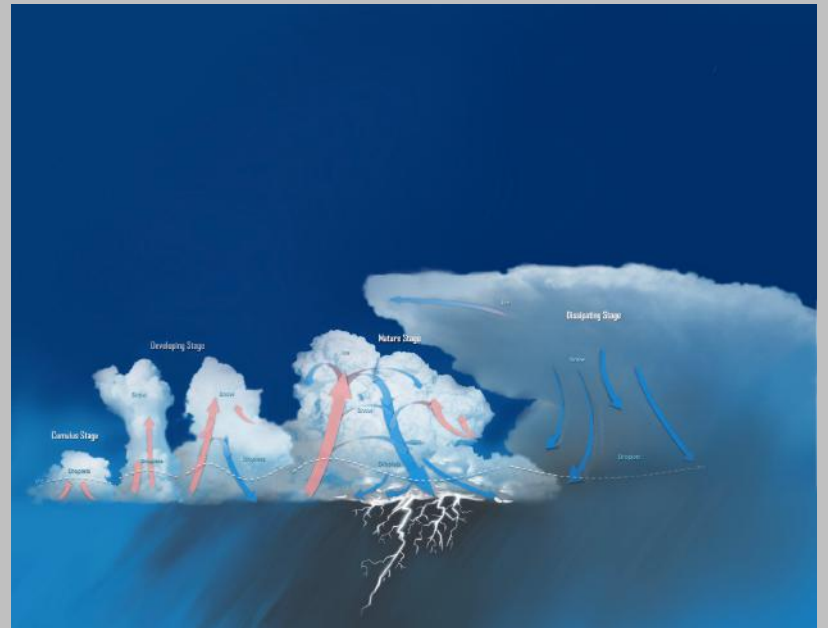
Creative Simulation Course 2022 – Neele Katharina Siemann



# the **IDEA**

## Preproduction

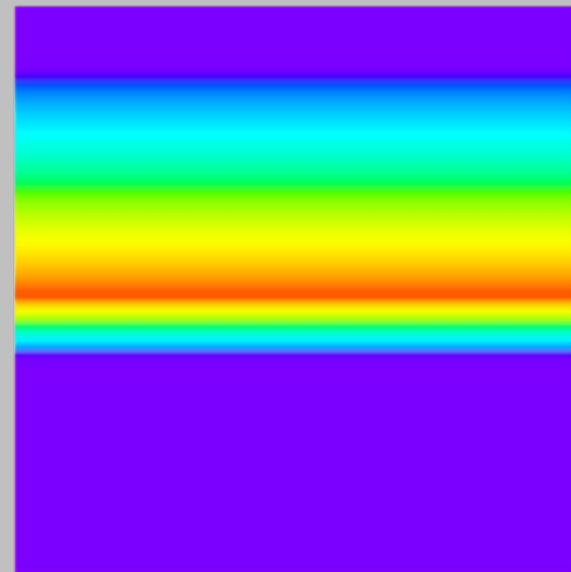
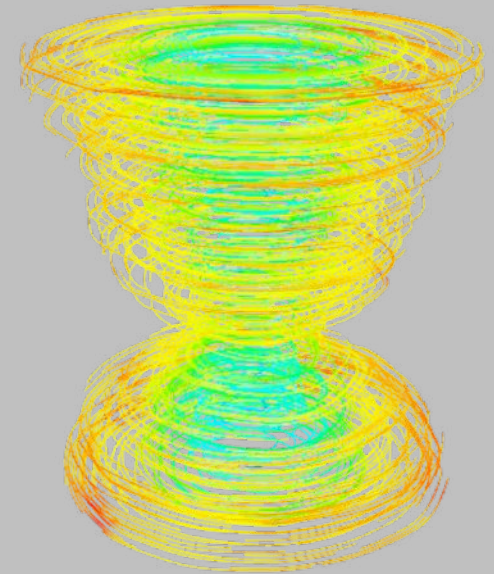
- sunset cloudscape
- breakthrough of the character
- cloak simulation (extra)
- real-time implementation
- simulation of the cloud



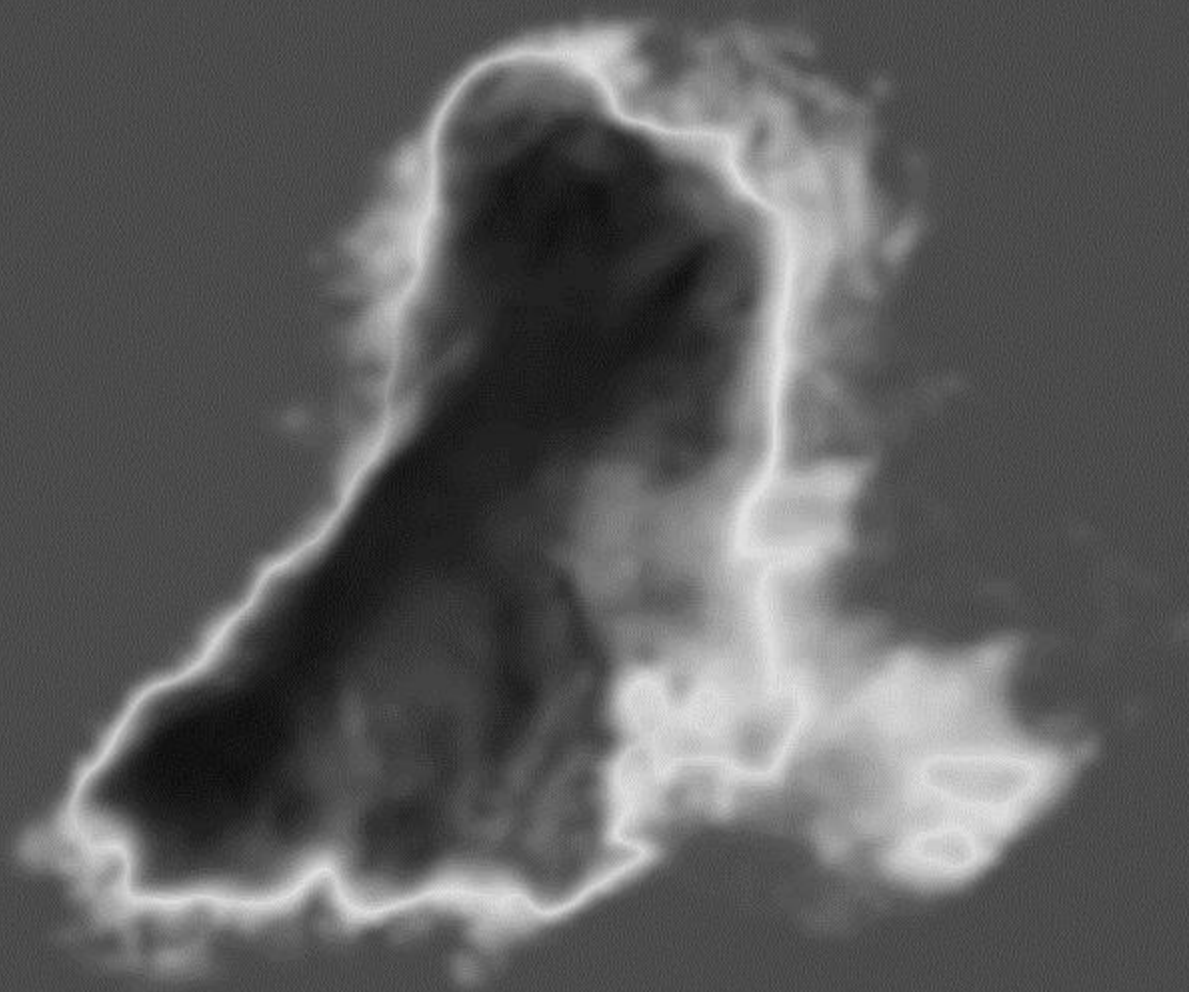
# the **SIMULATION**

in HoudiniFX 19.5

- VDBs/Houdini Volumes
- Custom Scalar/Vector Fields
- Custom Multisolver
- Breakthrough Simulation
- Vellum Simulation



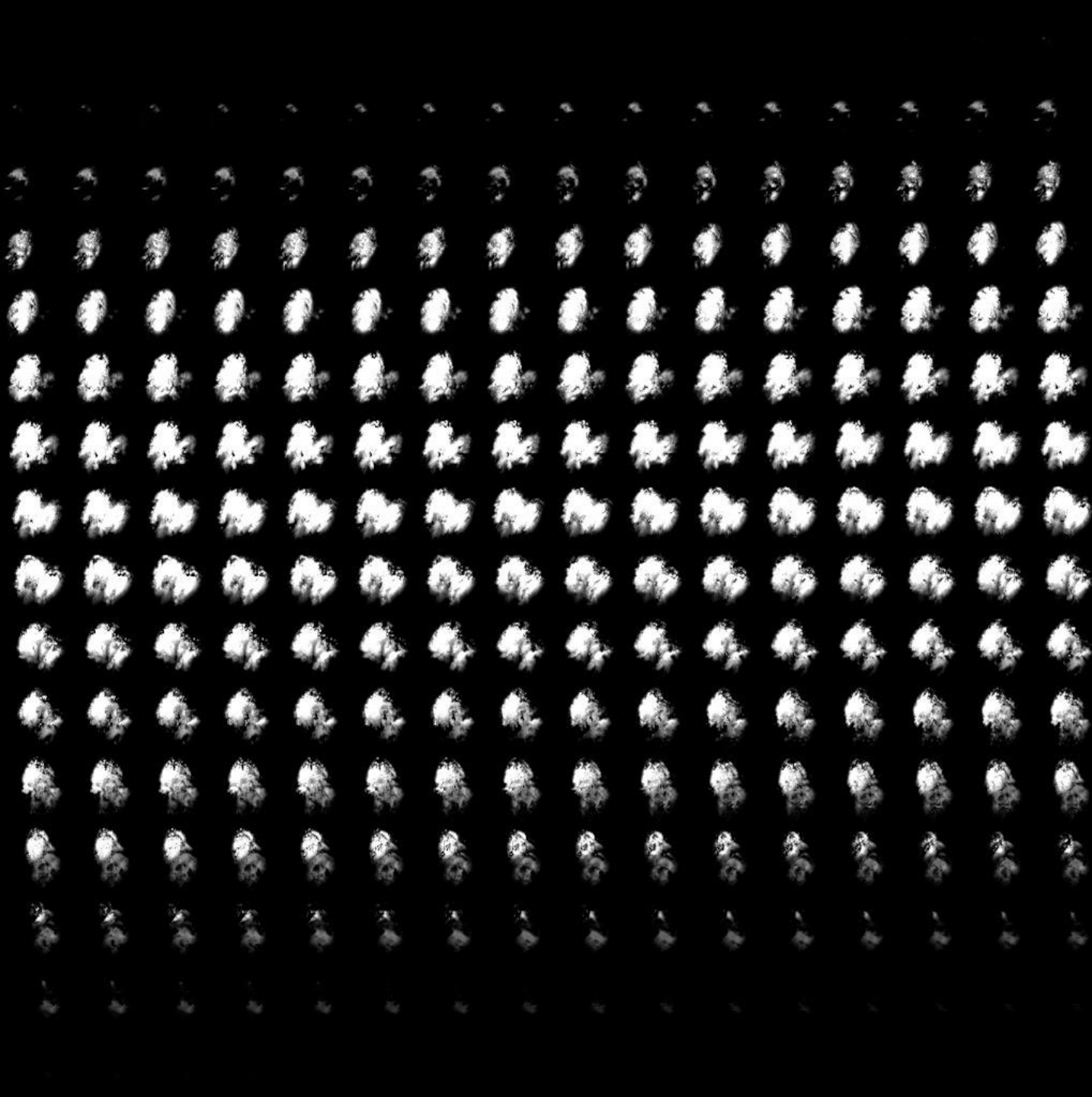




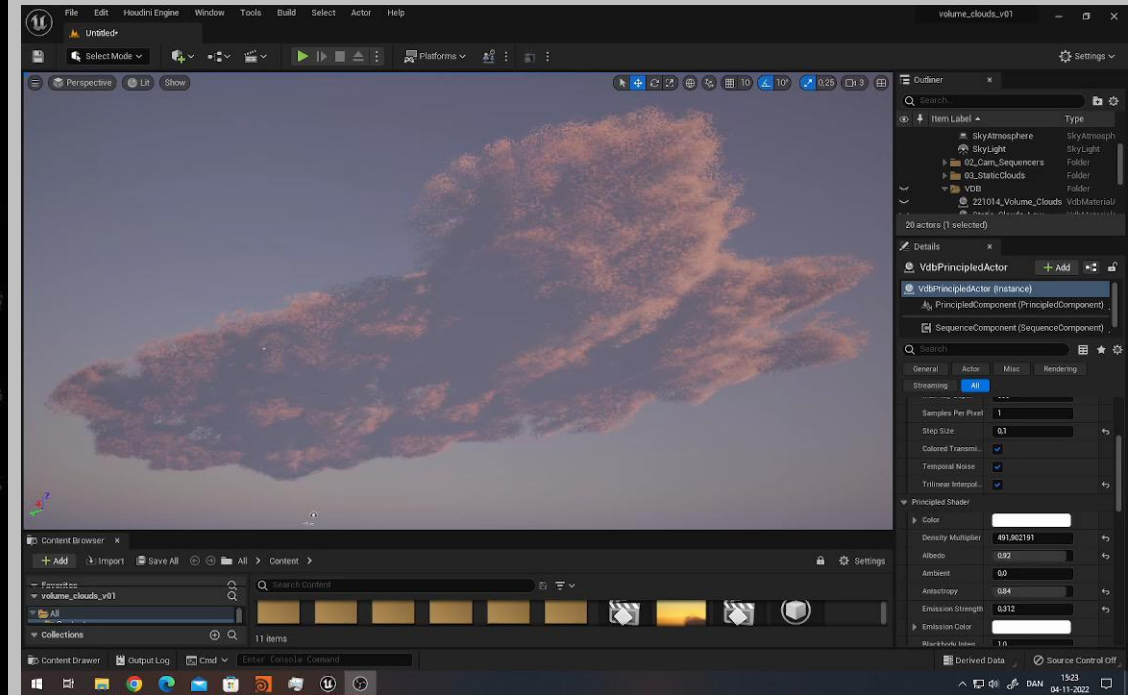
# the **IMPLEMENTATION**

in Unreal 5.0.3

- BG → Volume Textures/Material & Cloud Actor
- FG, Breakthrough → Experimental Plug-In VDBs to Nano VDBs
- Character → Poses, Animation Sequences, FBX for Houdini
- Cloak → Vertex Animation Texture



## Volume Texture



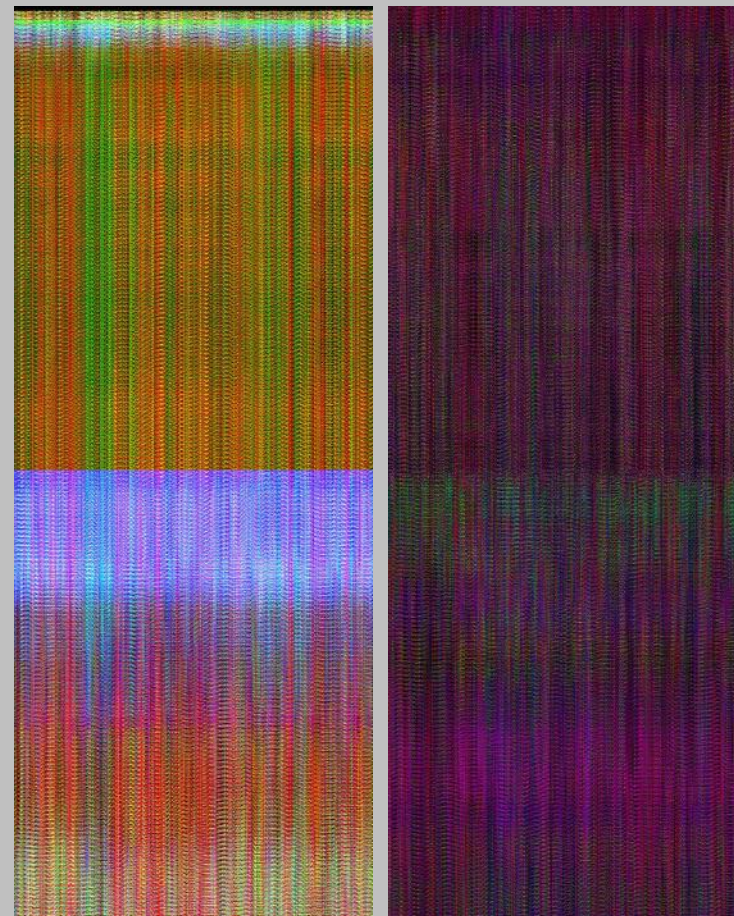


## Character Animation (Start-/Endpose)





## Vertex Animation Texture



















# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann

### **7. Paraskevi Malamatenia**

8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

# Case Study

Paraskevi Malamatenia

Creative Simulation Technologies 2022

The Animation Workshop

# Concept

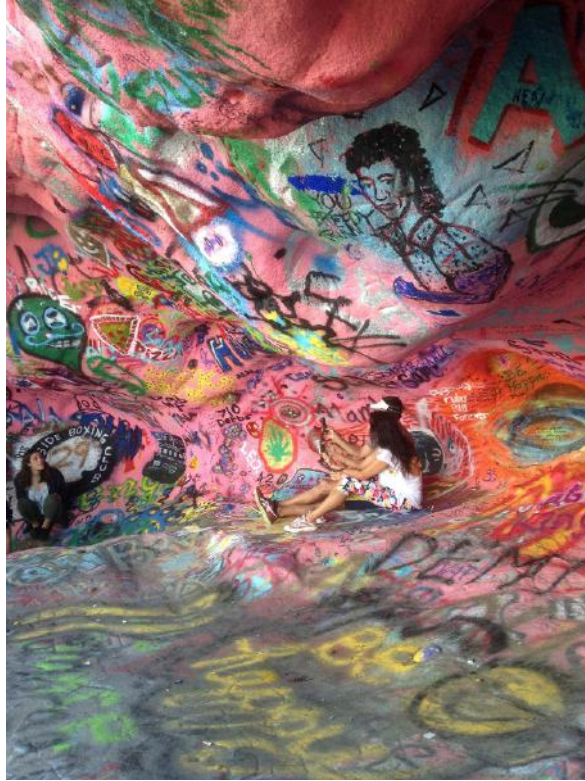
The main idea of the project was the creation of a night scene in a sea cave environment.

The basic elements were:

- Sea waves and bioluminescence simulation
- Neon graffiti material on the walls of the cave
- Cave collapse simulation



# References

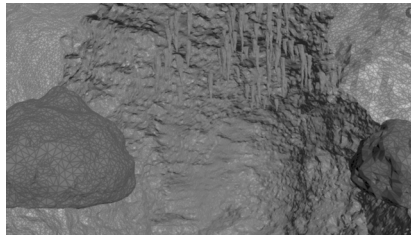
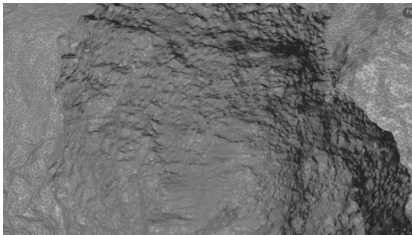


Paraskevi Malamatenia

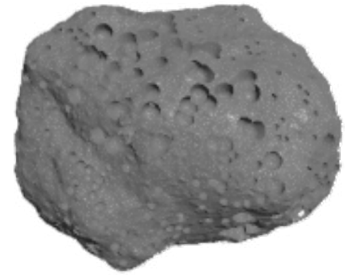
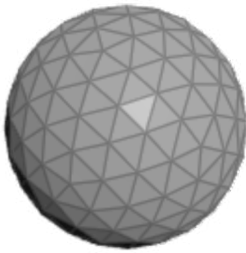
The Animation Workshop

# Procedural Modeling

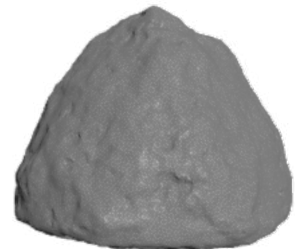
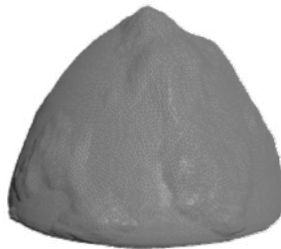
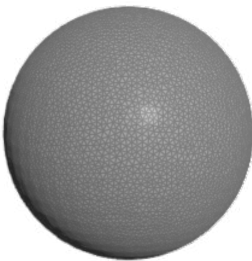
For the environment, multiple assets were created procedurally: cave, different types of rocks, stalactites, ground and pebbles.



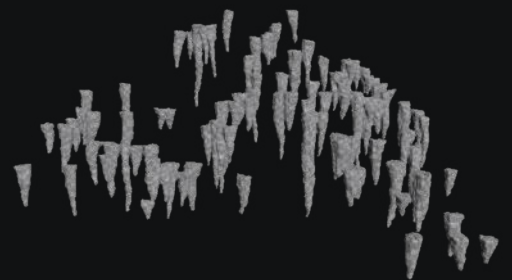
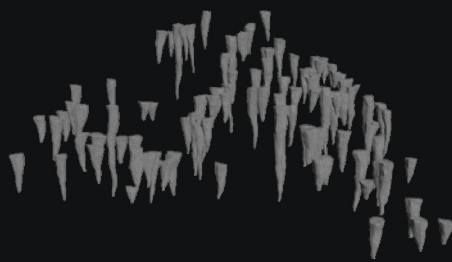
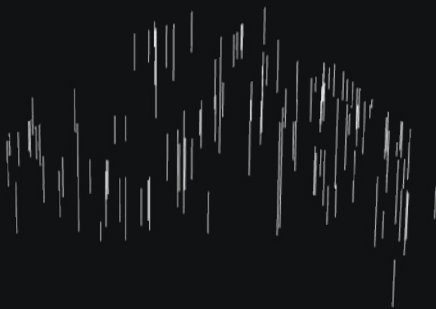
Laying multiple noises the rocks and the stalactites were created.



*Process for the eroded rocks*



*Process for the ground rocks*

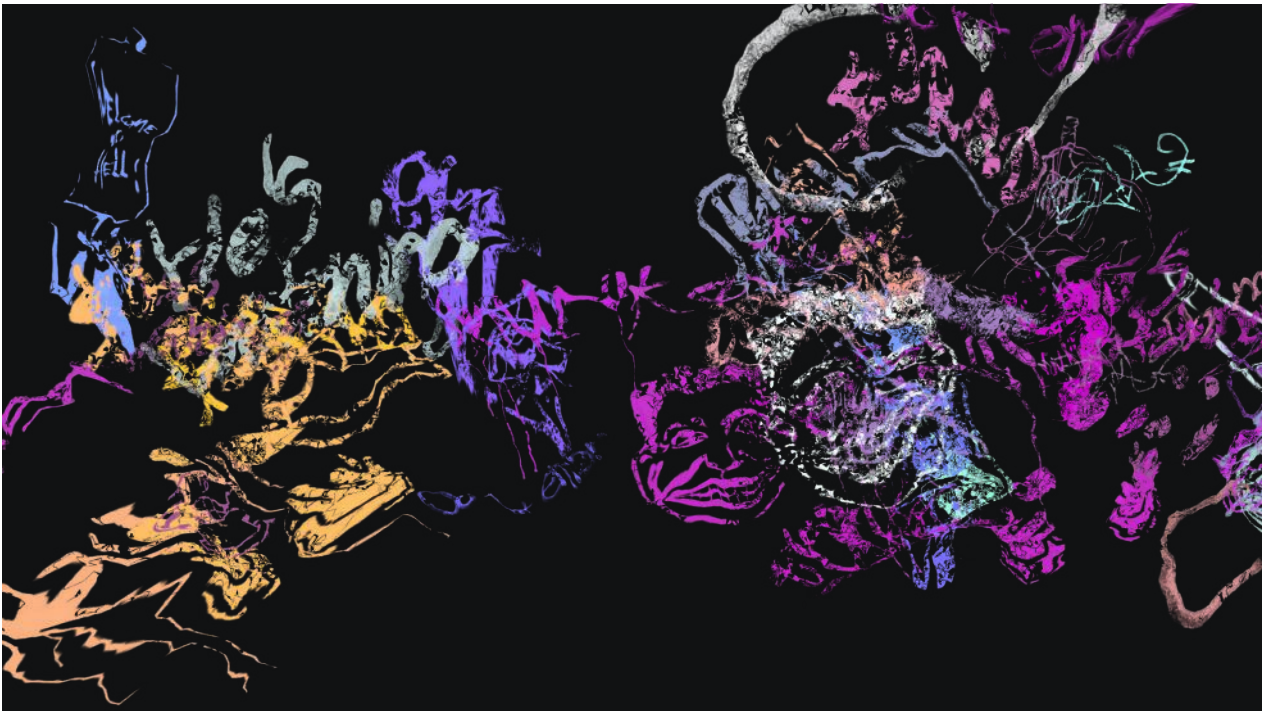
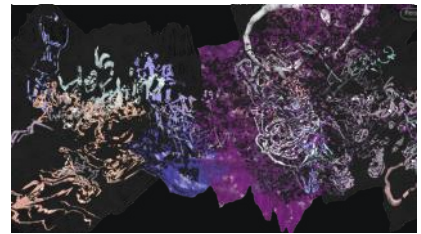
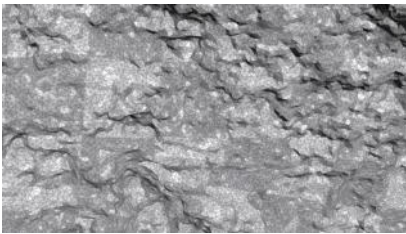


*Process for the stalactites*



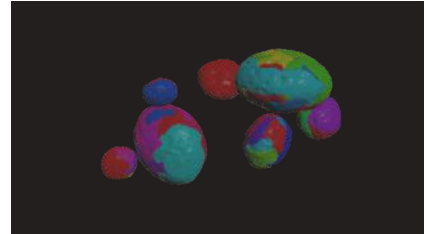
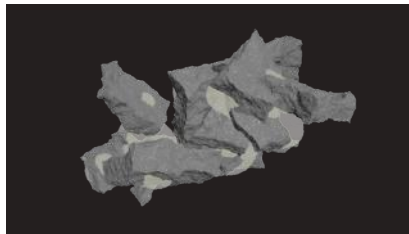
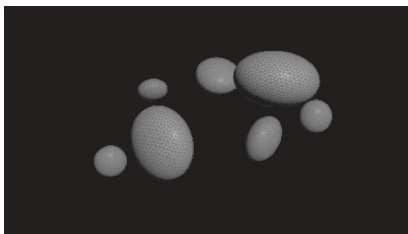
# Graffiti Material

After downloading some graffiti images from the internet, they were randomly assigned to points in the cave's walls. Afterwards, they were assigned a different color and emission from a color ramp and then they got rayed back in the cave's walls.



# Rocks Simulation

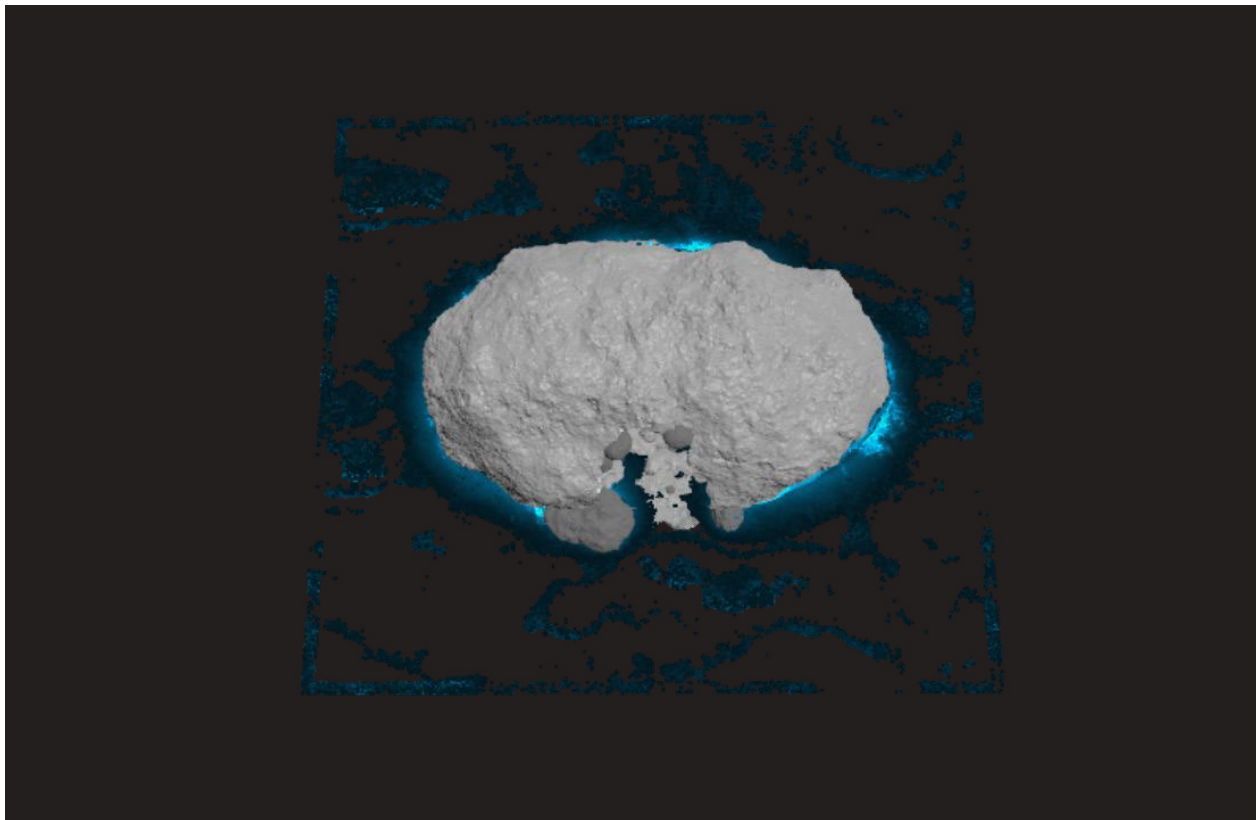
Using noise to boxes, i managed to create more interesting fracture shapes. With the rigid body solver, this fractures “crack” once they hit the ground.





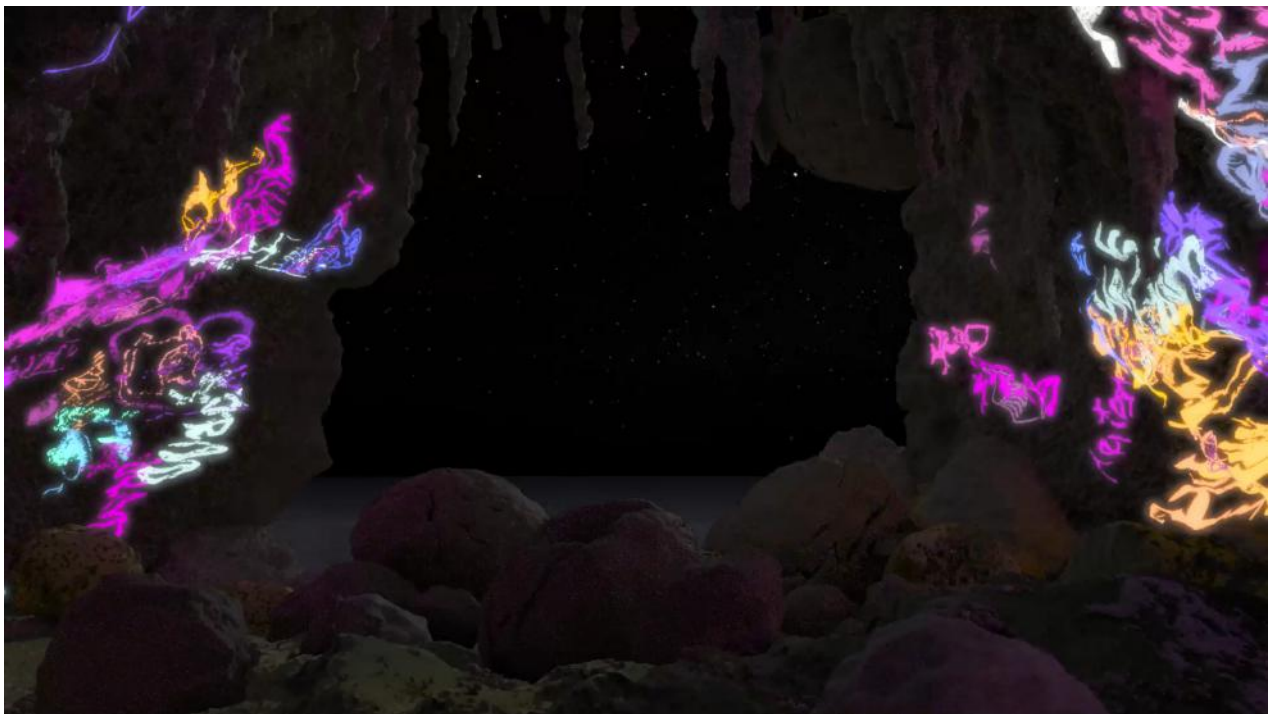
# Bioluminescence Simulation

The Wave Tank Shelf Tool was used to create the ocean. For the bioluminescence effect, the change in the velocity was calculated in order to select points from the wavetank. Afterwards, depending on the collisions, the bioluminescent particles became neon once they hit objects and then slowly faded away.



# Final Project

*The project was rendered with Karma*



Paraskevi Malamatenia

The Animation Workshop

# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia

### **8. Patrick Baitinger**

9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina

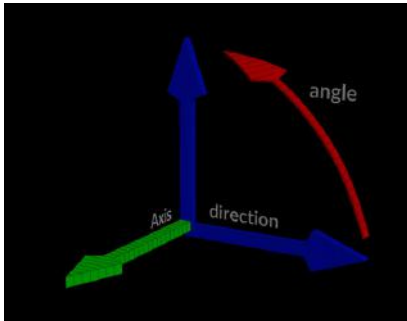


Creative  
Europe  
**MEDIA**

## Rotations with Vectors:

Seems easy and straight forward, but it isn't.

A Vector rotation is defined by an Axis, which the vector is rotated about and the angle by which the Vector is rotated about.

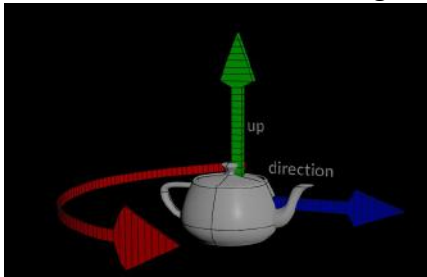


## Stable Rotation

For a stable rotation in 3D Space, we need two defined Axis:

An up-vector which give use information about which direction in 3D Space is up on our object.

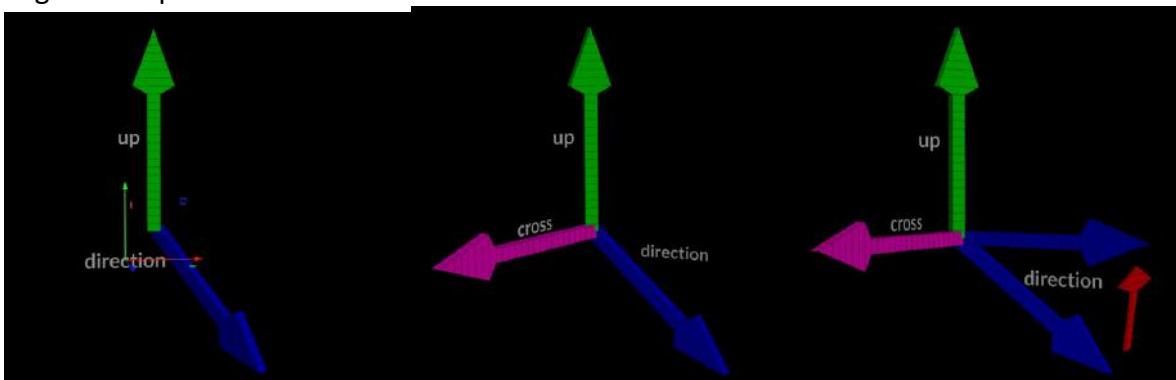
And a forward vector which gives us the direction we are facing towards.



## Align Direction and Up

To align (get an angle of  $90^\circ$ ) between up and direction, we do a double cross product of Both.

A double cross is first a cross product between direction and up the cross product of those two vectors will then be crossed again with up to get the initial direction again but now aligned to up.



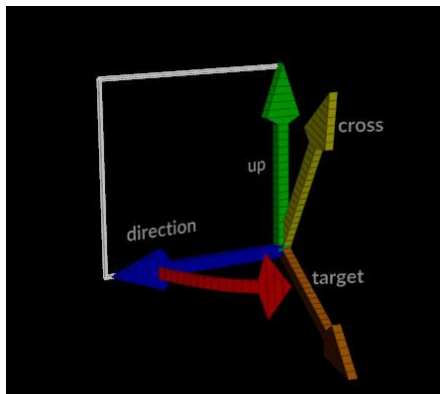


## Rotation Direction

We have two rotation directions, positive and negative.

To define which is used for the rotation towards sth. We need to figure out on which side of the Up and direction Plane the object is, we want to rotate towards.

For this we calculate the dot product between the cross product of the Target and the direction and the up workout. After that we compare the dot product against 0 we have the rotation direction.



## Rotation seeking

This is the process of doing the calculations above based on the feedback from the last frame. First, we need to calculate the direction between the point and the target point. Then we calculate the alignment of the target direction towards the direction up plane, which defines in which direction the vectors will rotate by. For now the angular velocity is hardcoded to a value.

## Angle of Rotation (level one abstraction)

To get the angle between two Vectors we need to calculate the dot product between those vectors and use the result as input for the arccosine function.

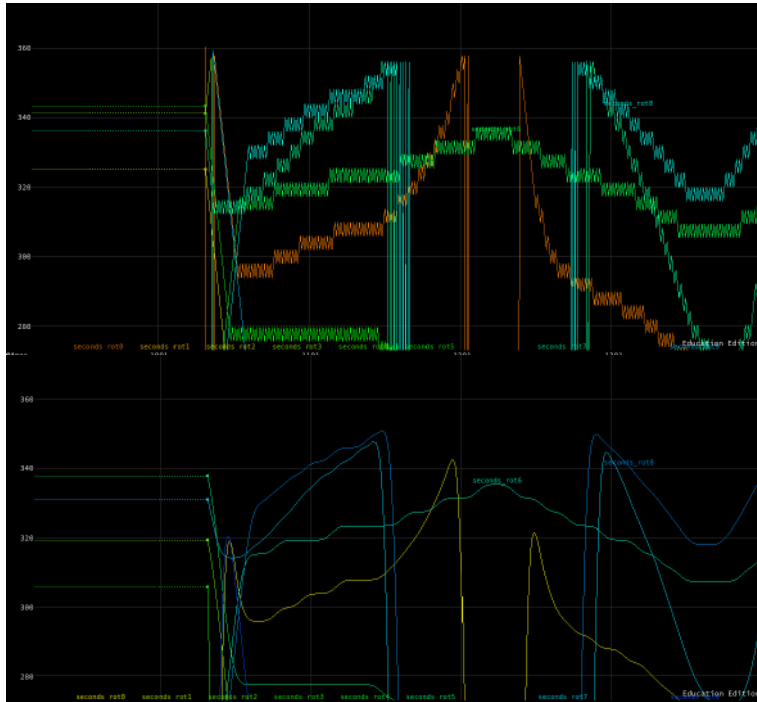
To get the degree of the Hand moved over the Watch we need to calculate the angle between the Rest position of the Hands at noon and the current position of the Hand. This methods flaw is that it can only calculate angles inside a triangle so  $0^\circ$  to  $180^\circ$  as everything over  $180^\circ$  is like an angle smaller  $180^\circ$  from the other side.

To solve this we can reuse the alignment calculation and switch between the ranges of  $0^\circ$  -  $180^\circ$  and  $180^\circ$  to  $360^\circ$ . Now we turned the vector rotation into a single numeric value.



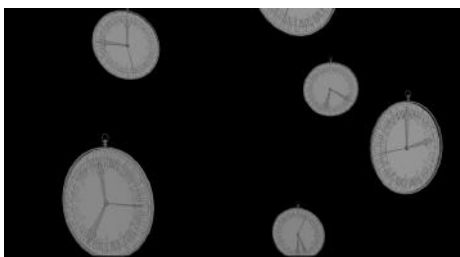
## Data Smoothing

As it is impossible to hit the target vector exactly the rotated vector will jitter around the target. To get rid of this jitter the data needs to be smoothed over time. The result is a smooth output data which can be used for the final Hand rotation.



## Level Two Abstraction

The smoothed numeric value can now be used to rotate a new vector around it's up Vector. This is now the data used to rotate the geometry of the Hand.



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger

### **9. Sergio Esteban Casilimas**

10. Sylvia Rybak
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

# Sunflower growth

overview



## Project objective

Create a digital simulation that mimics a sunflower development from seed to maturity.

## Visual component

Create aesthetically pleasing shot of a Sunflower development

## Technical component

Create a scalable and performant plant growth system that is responsive and interactive to its environment.



## Developed systems

Basic artificial vision system to guide plant growth, sun tracking and collision avoidance.

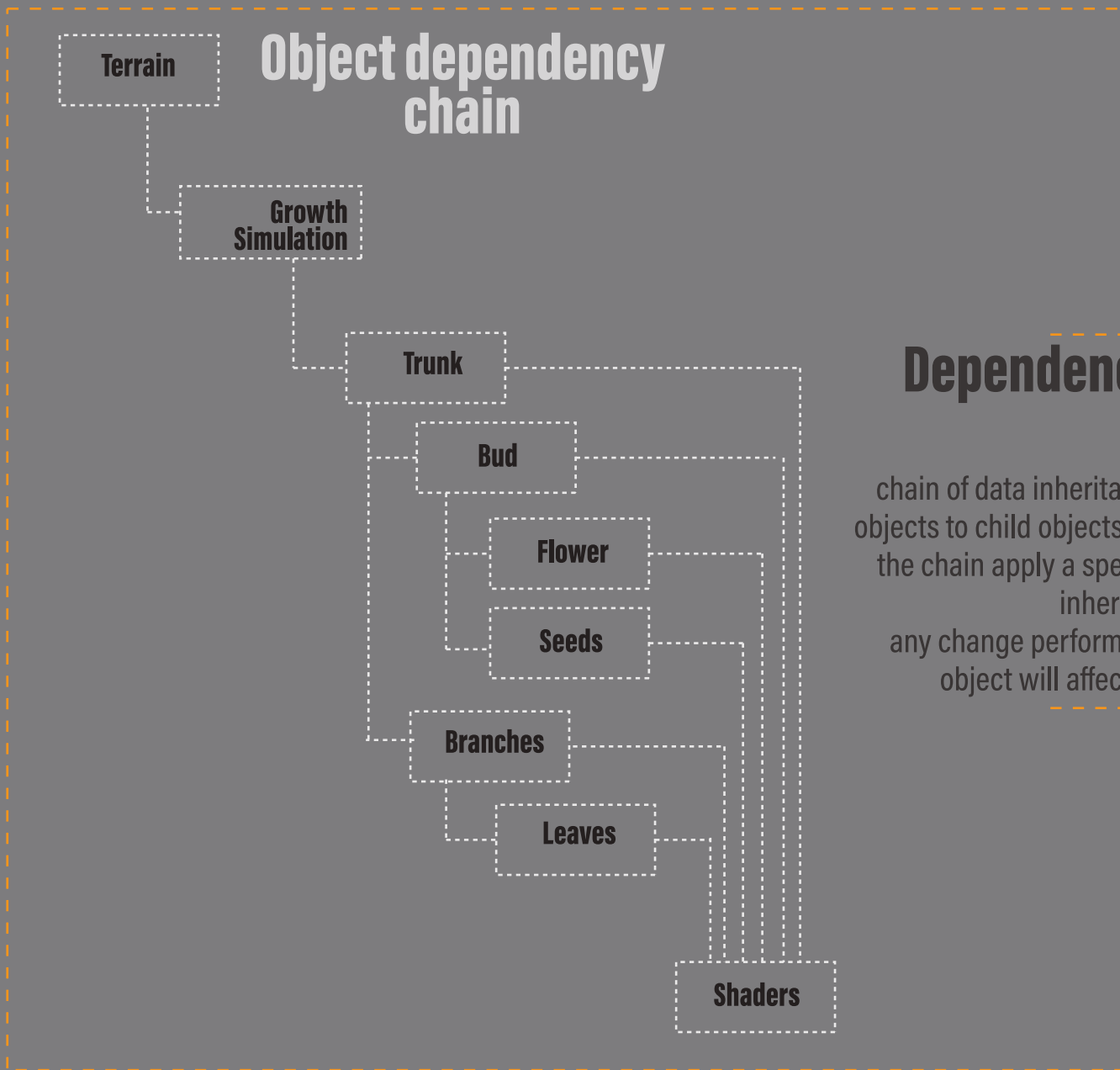
Self replicating growth system using Vellum PBD solver inside of Houdini.

Procedural modeling of detailed sunflower features.



**Sergio Casilimas**

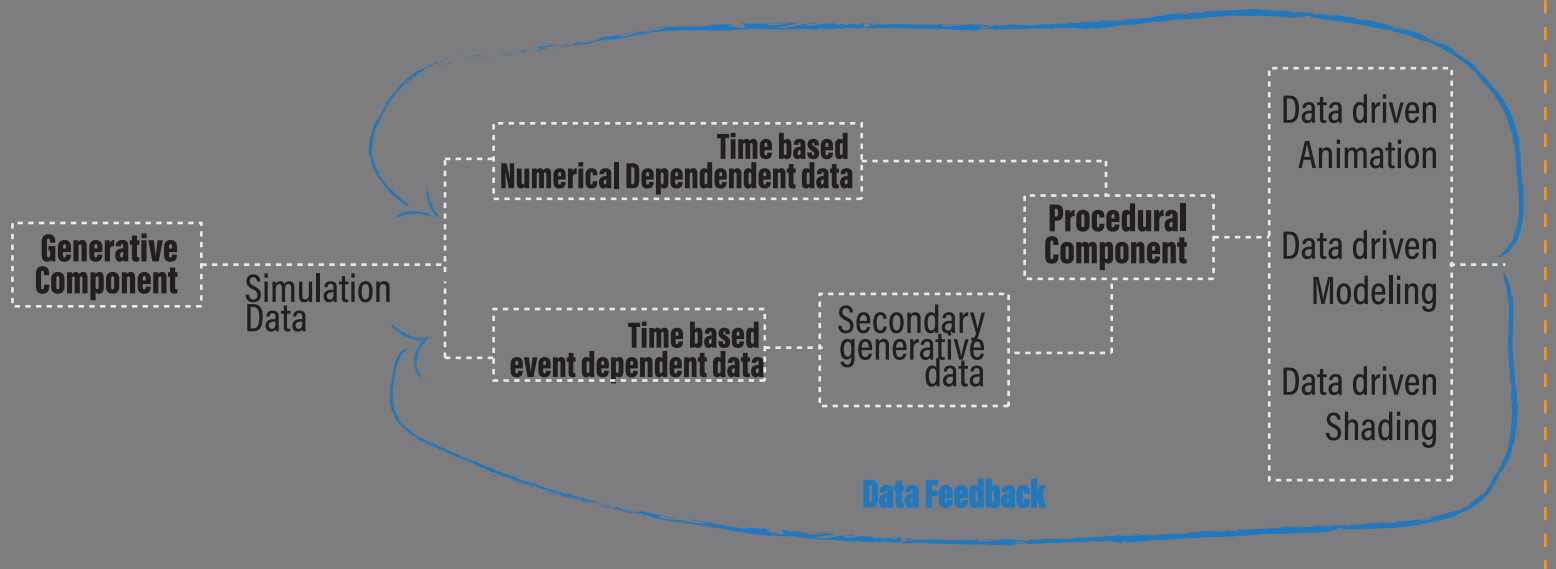


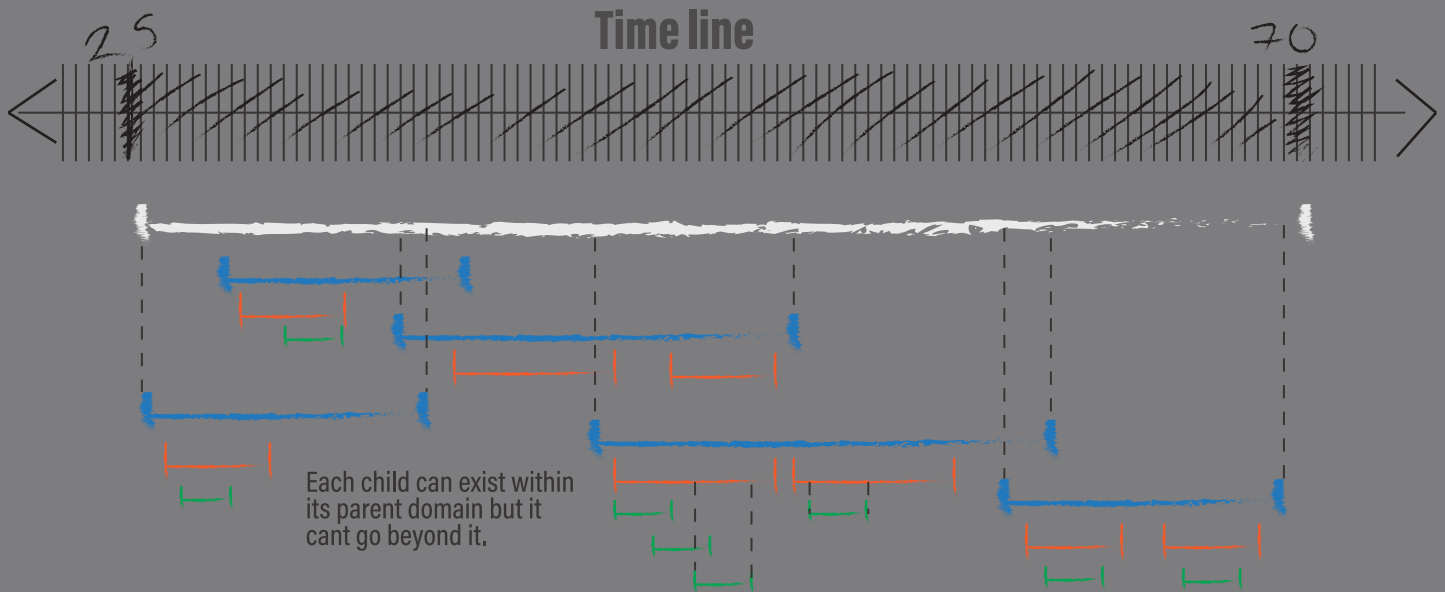


## Dependency-based system

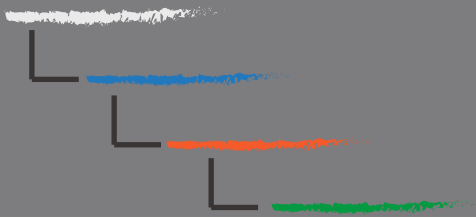
chain of data inheritance from parent objects to child objects. Each object on the chain apply a specific logic to the inherited data so that any change performed to the parent object will affect the hole chain.

## Task dependency





### Dependency chain

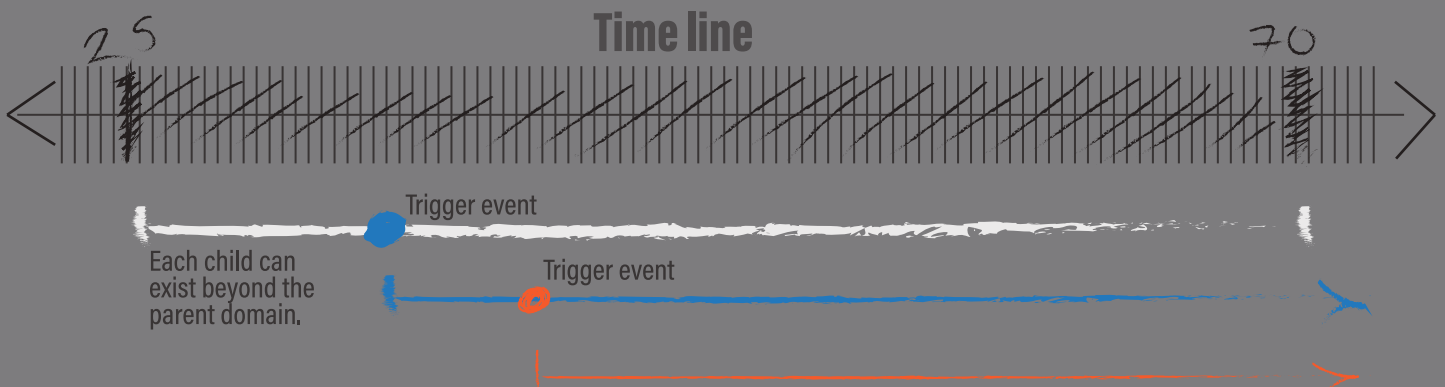


### Time based Numerical Dependent data

Hard coded time dependant piece of data.

Allows for a very controlable math based handling over data.

Constrained by the domain of the inherited data



### Dependency chain



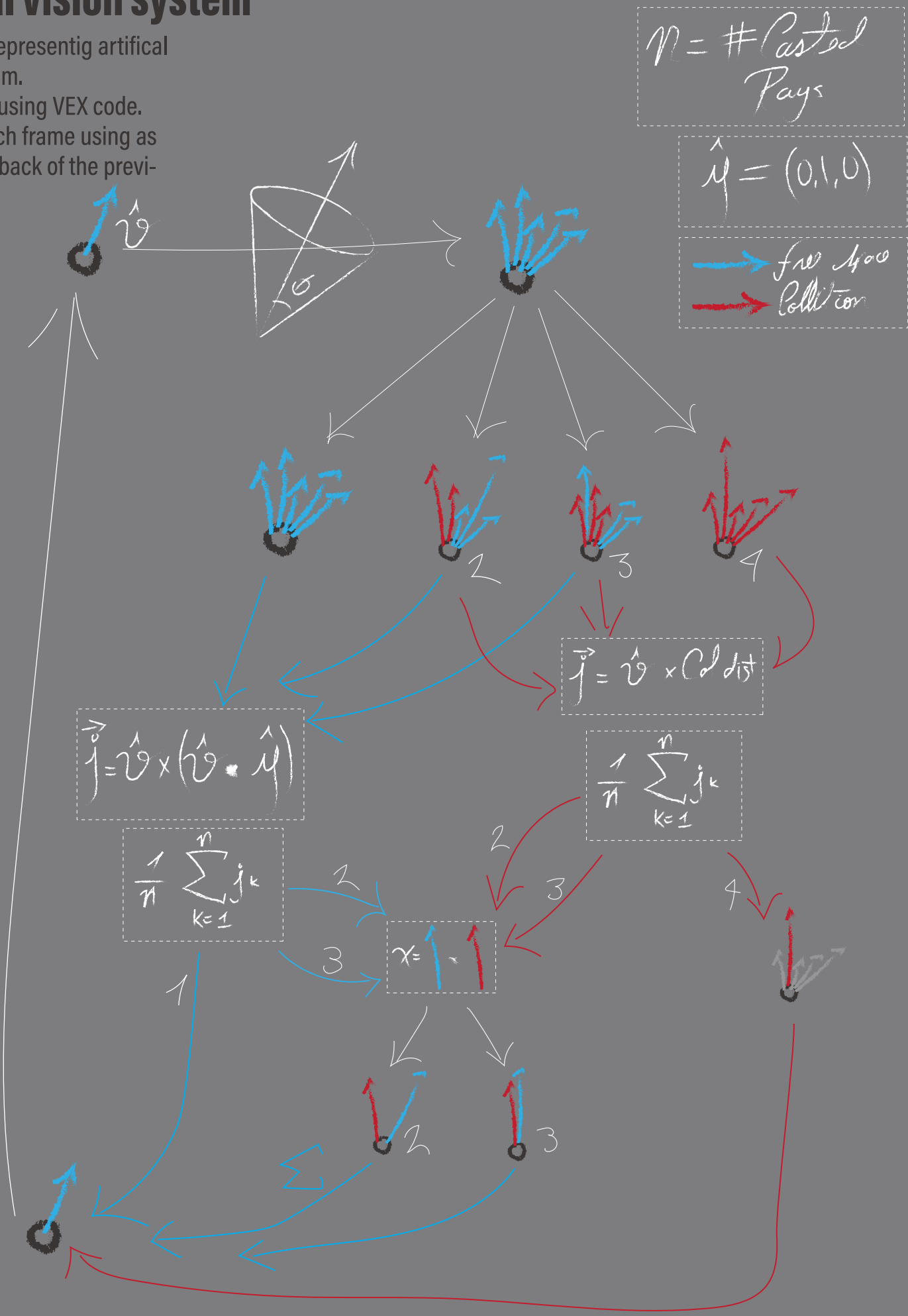
### Time based event dependent data

Inherited Data will trigger independent actions in the chain.

Allows for a dynamic creation of new sets of data.

# Artificial vision system

Visual guide representig artifical vision algorithm.  
Implemented using VEX code.  
Calculated each frame using as input the feedback of the previ-  
ous frame.



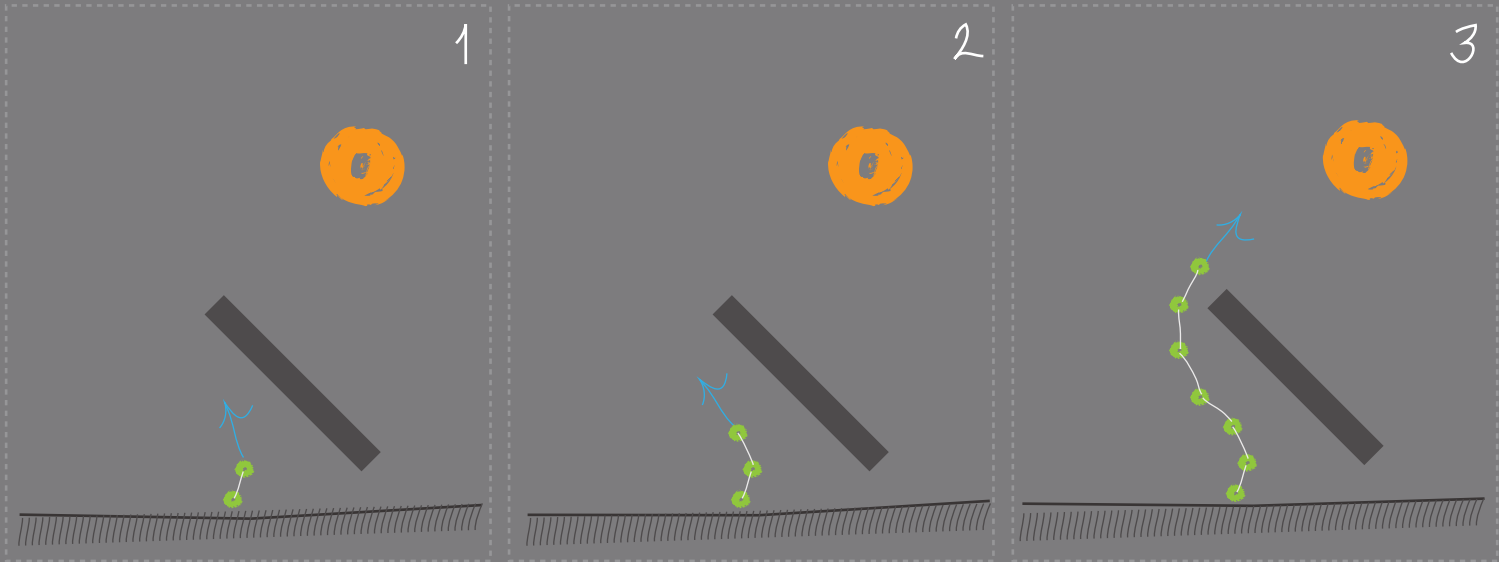
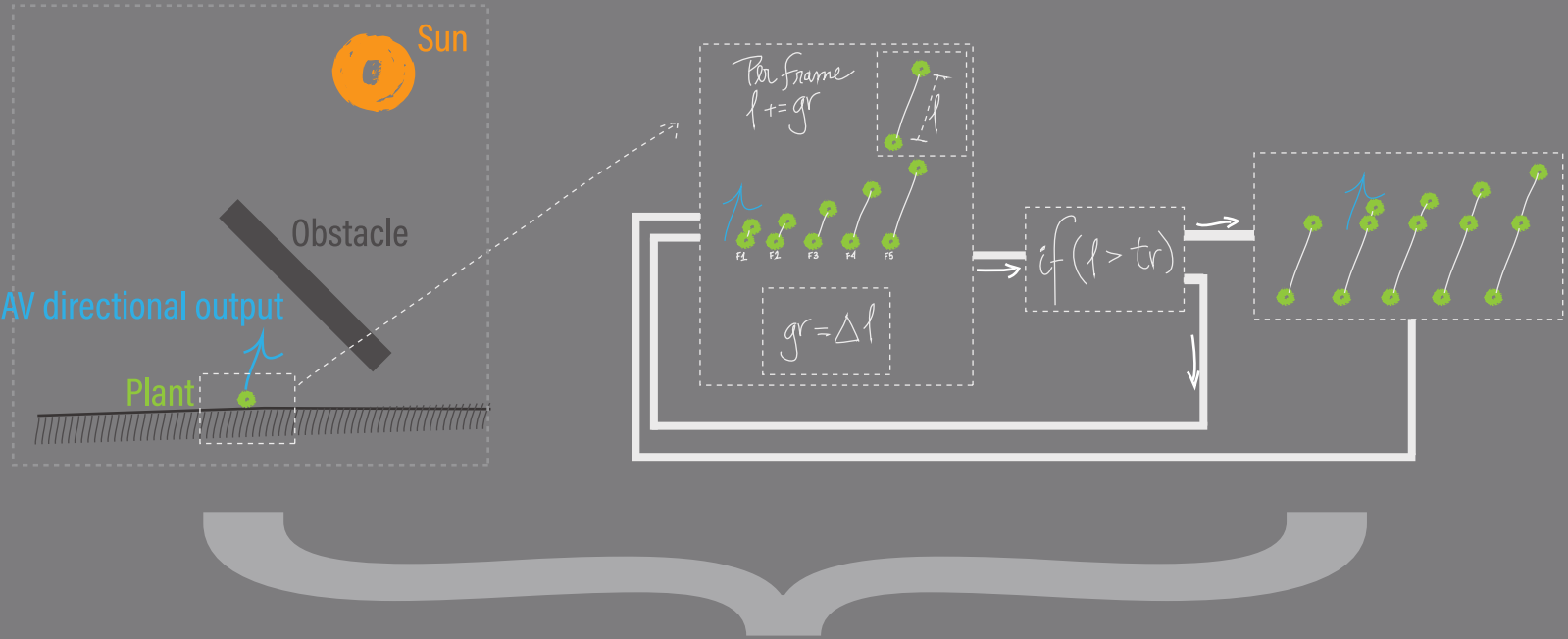
# Self replicating growth system

Growth is driven by two factors, terrain attributes and the artificial vision system output.

Growth rate is defined by a mix of terrain growth attributes.

Growth direction is defined by AV system

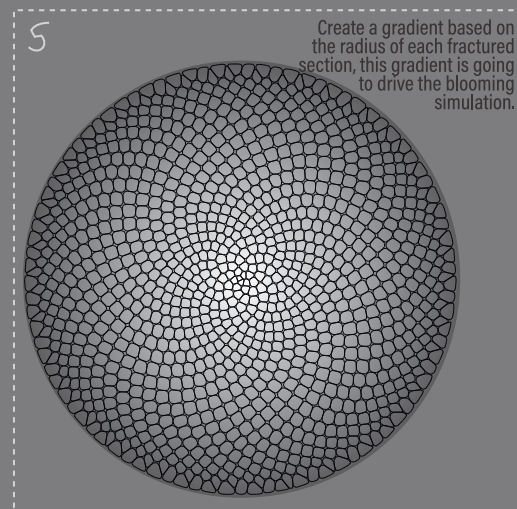
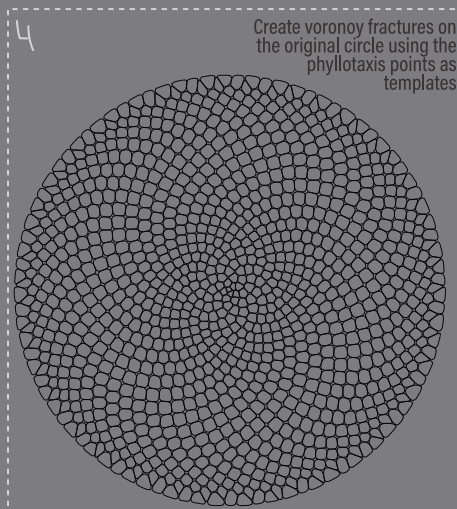
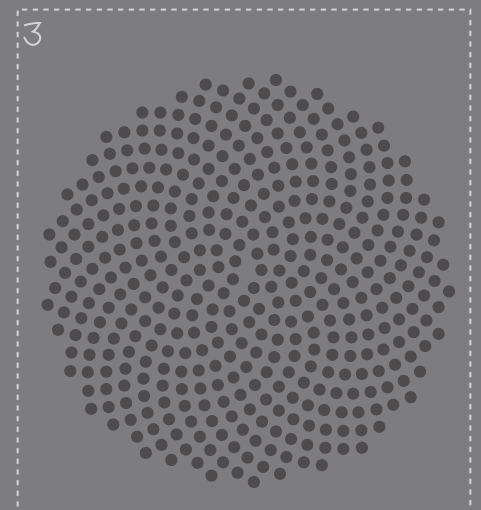
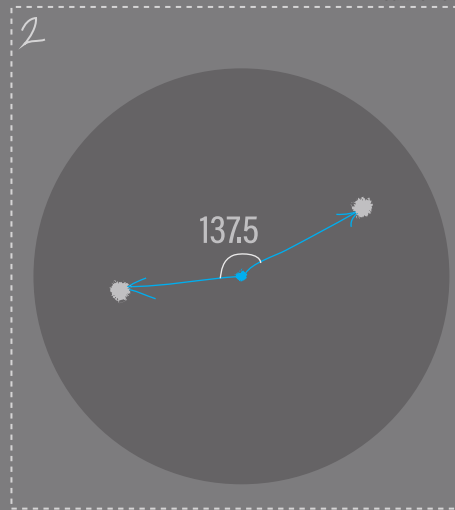
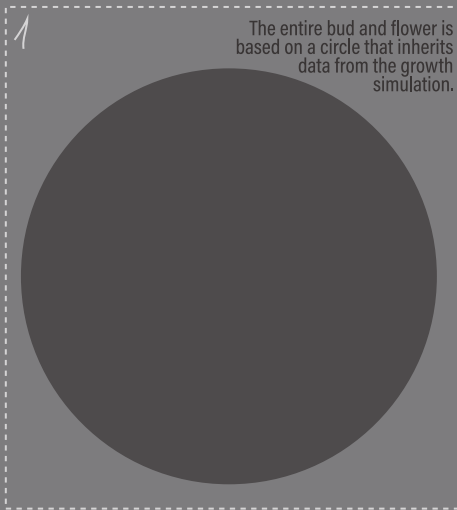
$gr$  = Growth rate  
 $tr$  = Max section length





# Plant features

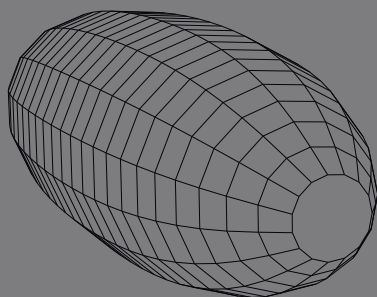
## Disk florets



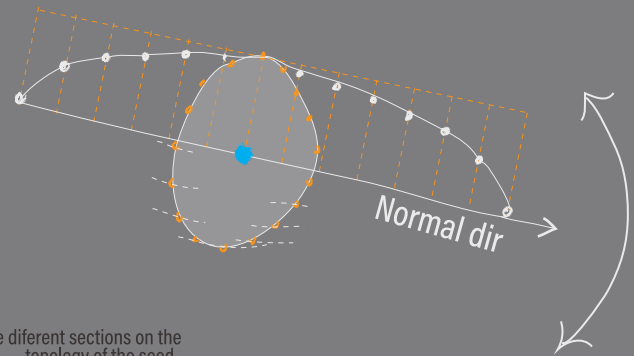
## Animation method

Time based Numerical  
Dependent data

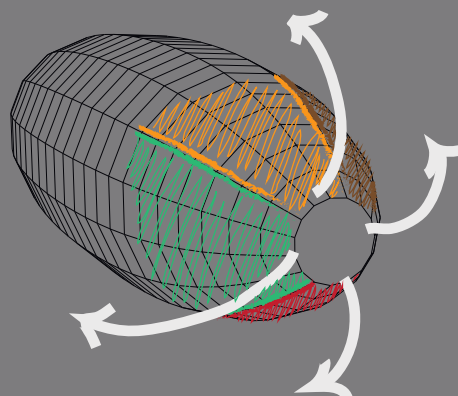
Time based  
event dependent data



Foreach fractured section:  
displace the points that made them up along the normal direction and interpolate between the displaced centroid to get a seed like shape.

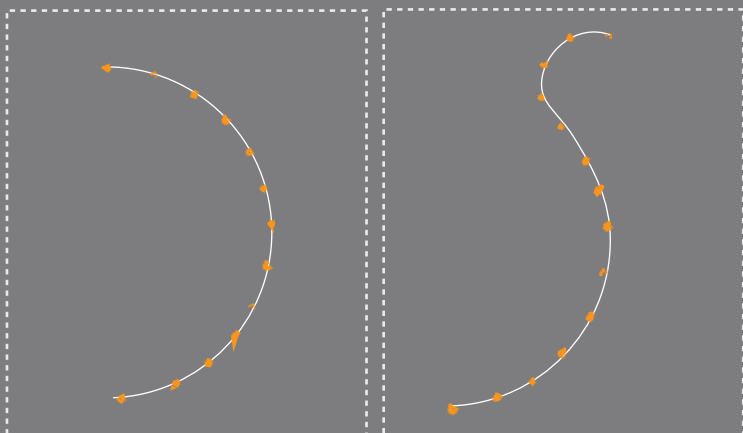


define different sections on the topology of the seed .

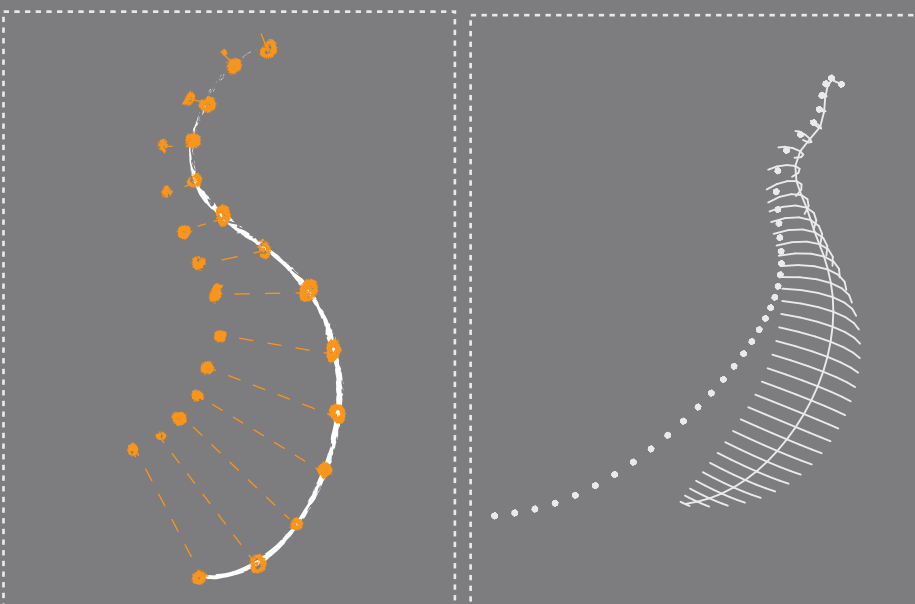


# Plant features

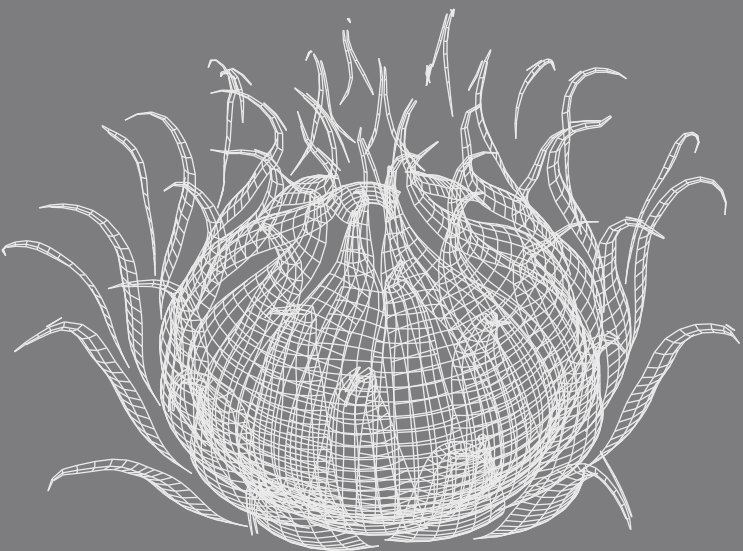
## Sunflower bud leaves



An additional phyllotaxis pattern is created on the circumference and concentric arcs are created on them, with a chain of matrices the tips get curled.



To create the surface of the leaf I use the displaced position of the points that made up the curve to act as center of a set of arcs that will serve as the cross section of the leaves. This modeling method will render cartesian like surfaces with implicit normalized UVs. To correct the UV dimension its necessary to scale them by the length of the cross section .



Implicit UVs  
For procedural  
texturing



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
- 10. Sylvia Rybak**
11. Victor Fredrik Weijers
12. Yuliia Silkina



Creative  
Europe  
**MEDIA**

## Case Study

Ninemia

Sylvia Rybak

Creative Simulation Technologies 2022

Concept:

- \* based on previous research into water, fungi and Hydrofeminism
- \* spa-like environment
- \* both eerie and ethereal

Simulation Types:

### 1. Pyro

Learnings:

- How to create pyro simulations with a mushroom like shape and modify the parameters to receive the effect I was looking for.
- Shading the pyro to achieve the desired shape and texture
- Properly caching and instancing volumes





## 2. Liquids

Learnings:

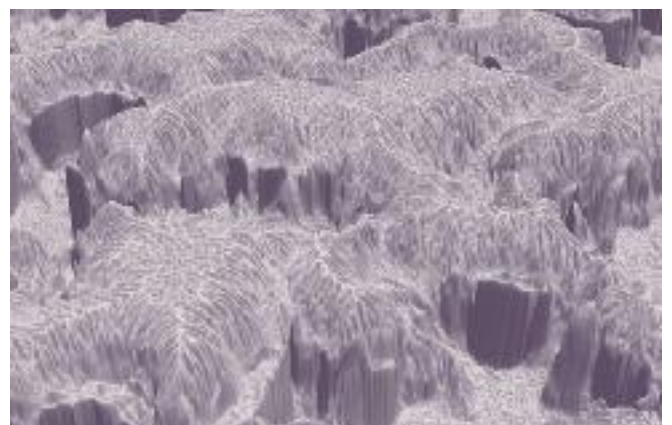
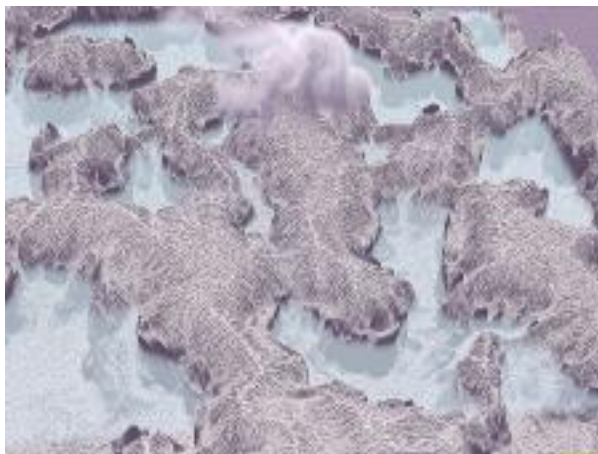
- How to create liquid simulations, which follow a spline shape.
- Refining liquid simulations for smooth movement
- Optimizing the simulations to make them more light-weight
- Shading water
- Creating and refining ripple simulations on a plane and tying them to the location of particles



## 3. Terrain

Learnings:

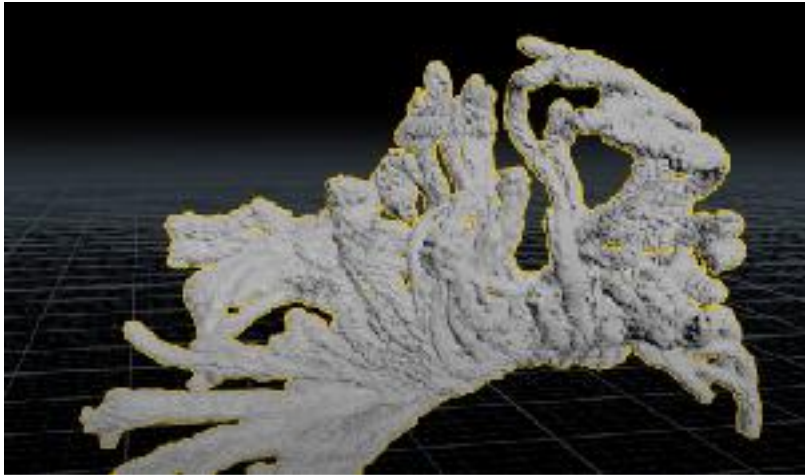
- How to use heightfields
- Sculpting the desired shape
- Using noise and erosion and other modifiers in order to achieve a more varied surface
- Scattering geometry and volumes on the terrain surface



## 6. Corals

### Learnings:

- How to procedurally create organic shapes
- How to use vellum for creating movement
- Meshing volumes



## 5. Rendering

### Learnings:

- How to optimize renders and make them as lightweight and fast as possible
- How to use Houdini's Karma as a render engine
- Refining shaders and simulations for proper rendering
- How to create render layers and composite them in Nuke
- Effectively using a render farm

### Test Renders:



Final Result:



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak

**11. Victor Fredrik Weijers**

12. Yuliia Silkina

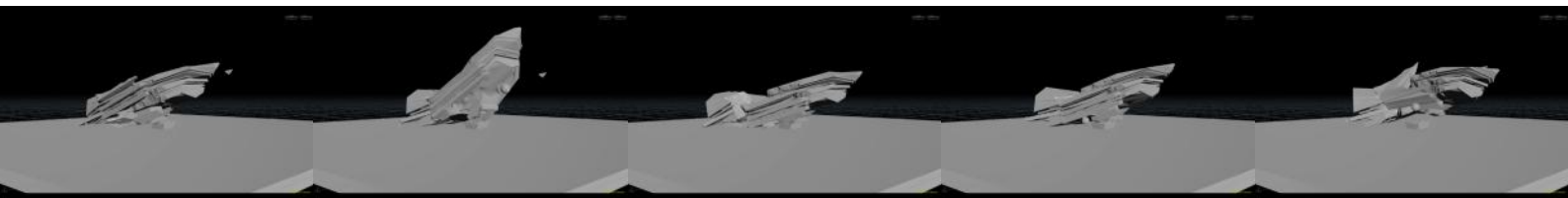


Creative  
Europe  
**MEDIA**

# Case study

## Jungle destruction game

### Victor Weijers



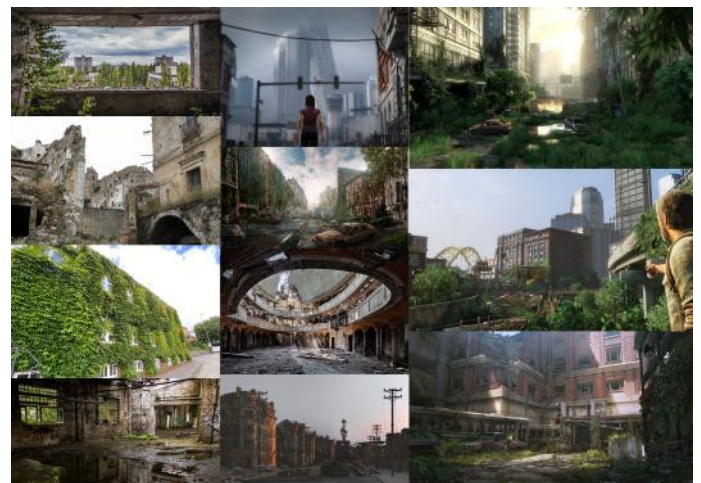


# Concept and drafting

Before the start of the course I already knew I wanted to do something with destruction and maybe something with plants.

These jungle ruins are a perfect combination of these two.

In video games destruction often doesn't look good. In this project I want to make a video game with good destruction and beautiful fractures.



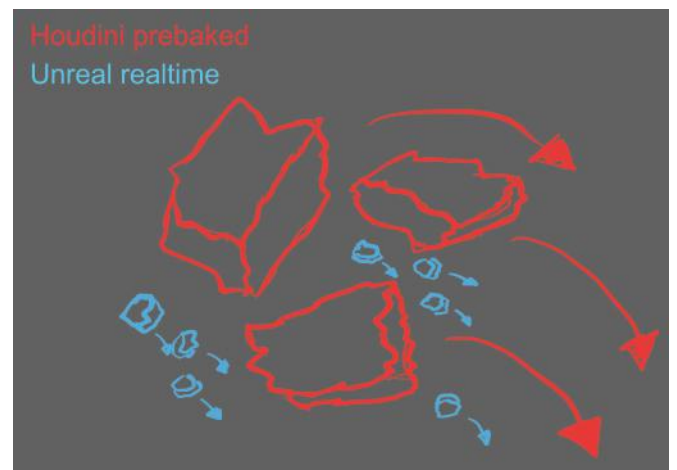
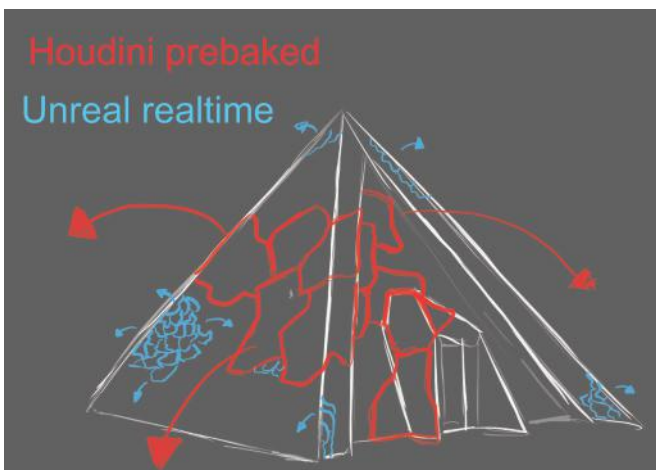


I didn't want to make jungle ruins as we know them. I wanted it to be a bit different. I took a lot of inspiration from alien fantasy architecture and old soviet monuments.



I'm planning on using Houdini and Unreal for the destruction. First I will fracture and simulate the big pieces that make up the geometry in Houdini to bake them out and import them into Unreal.

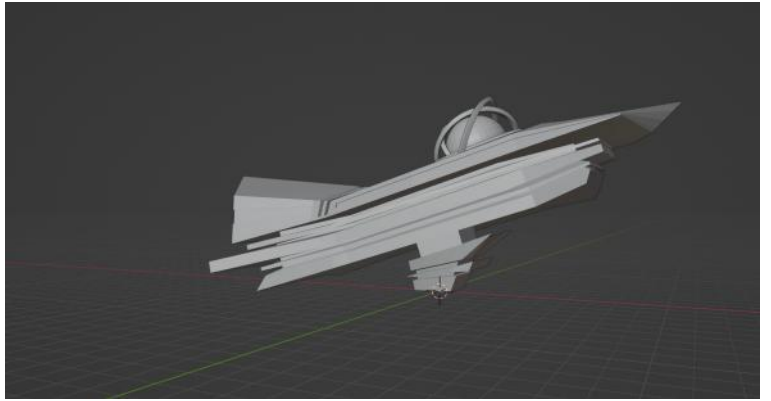
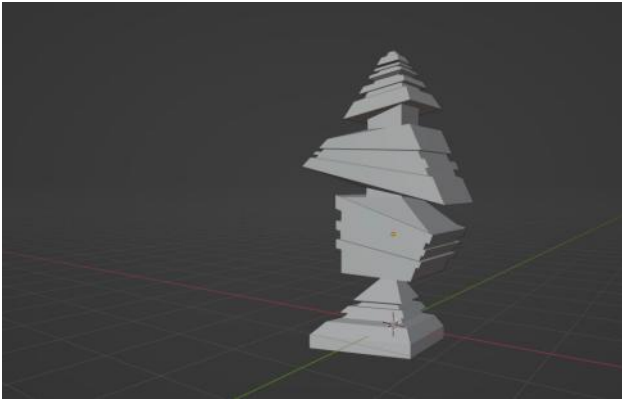
For the little chunks and details I will use Unreal to make realtime simulations.



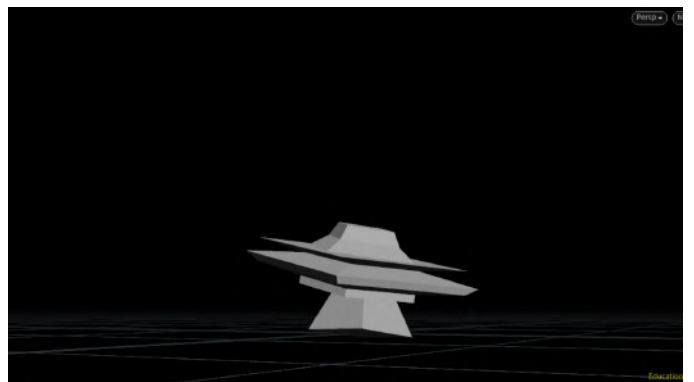
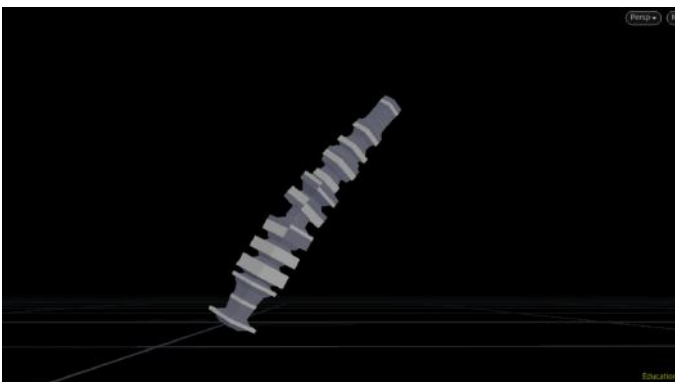
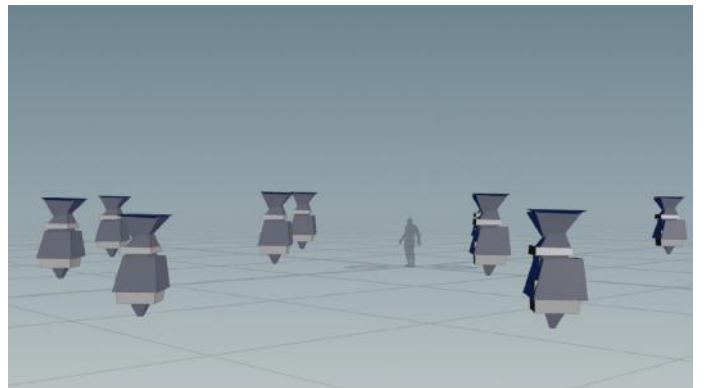
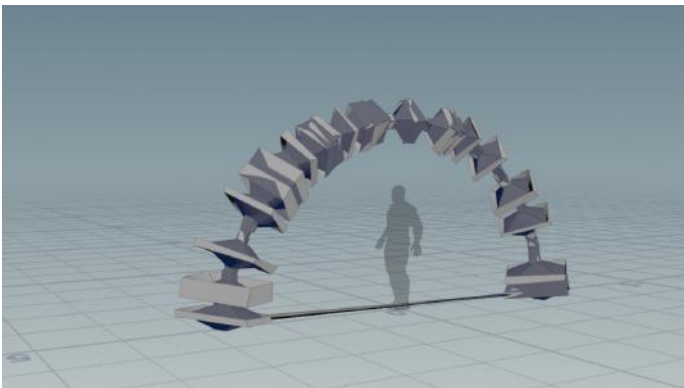
# Procedural models

## Week 4

To start modeling I first made model sketches in Blender to use as reference.

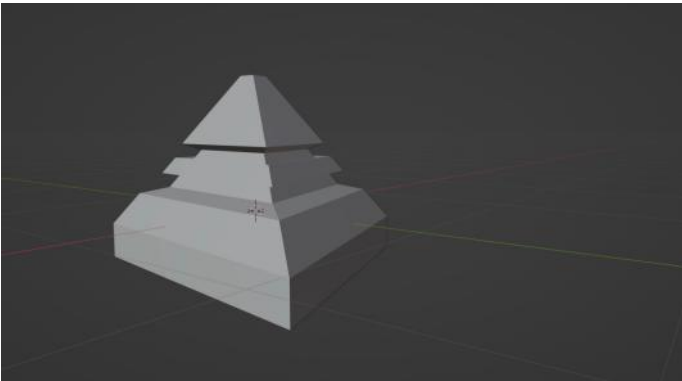


Then I started making procedural models using spline modeling. I played around with pscale, random seed and noise to change the look.

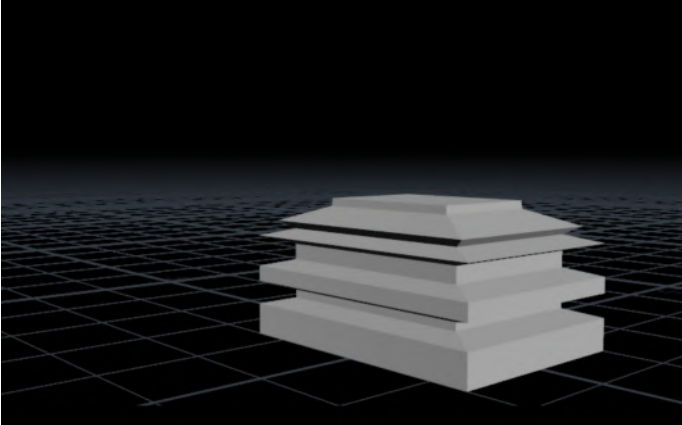
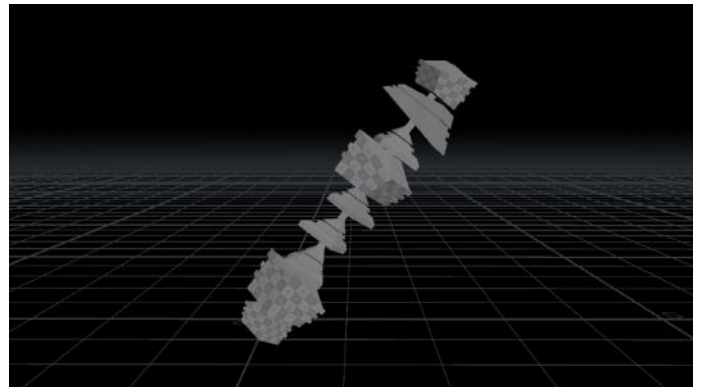
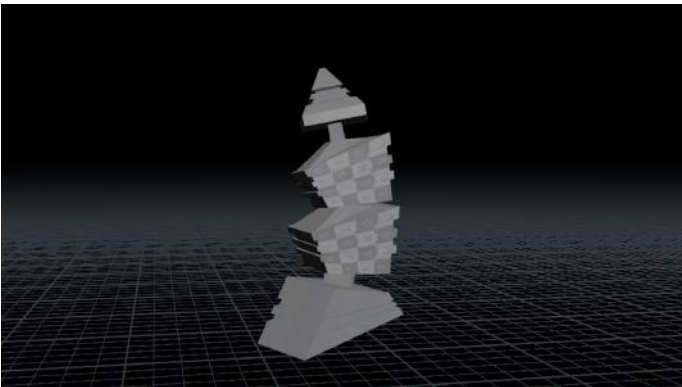


## Week 5

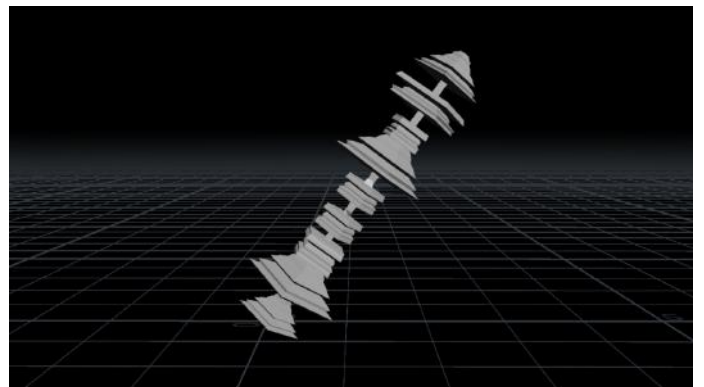
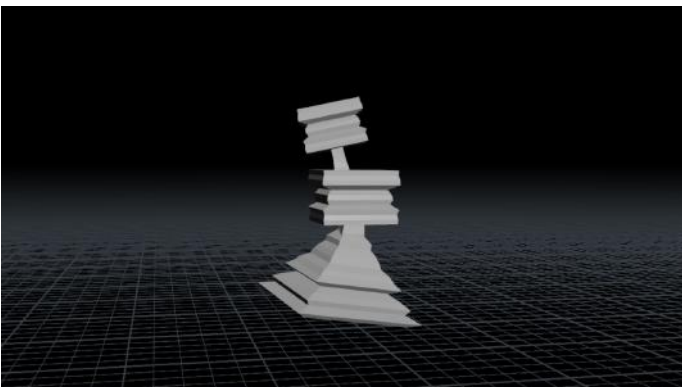
Here I tried out two different ways to make simulair models.



First I made some non procedural chunks in Blender to later copy them on a procedural spline model.



After that I made some spline models to use as chunks. These I copied on the same spline model.

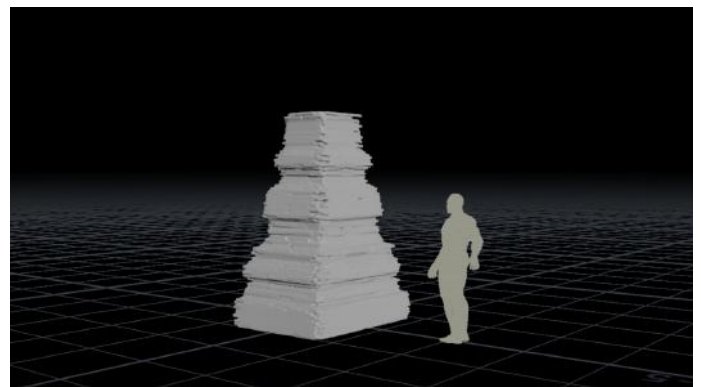
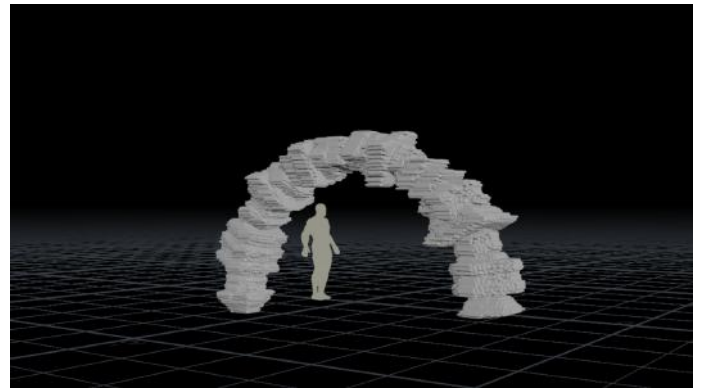
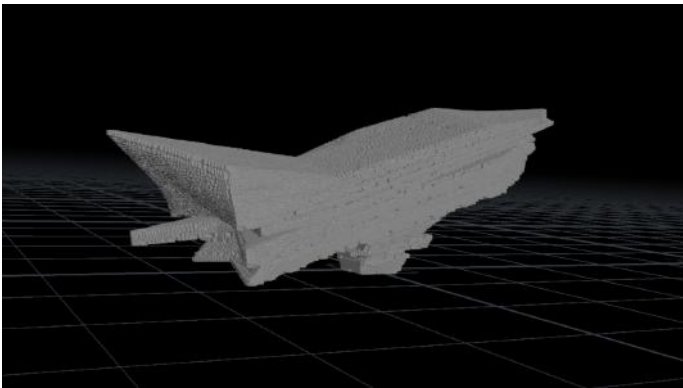
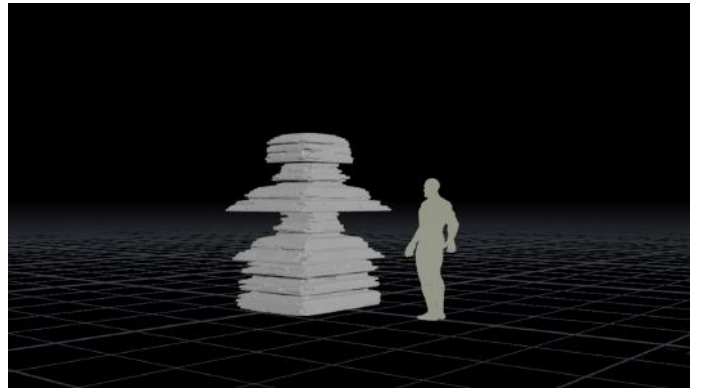
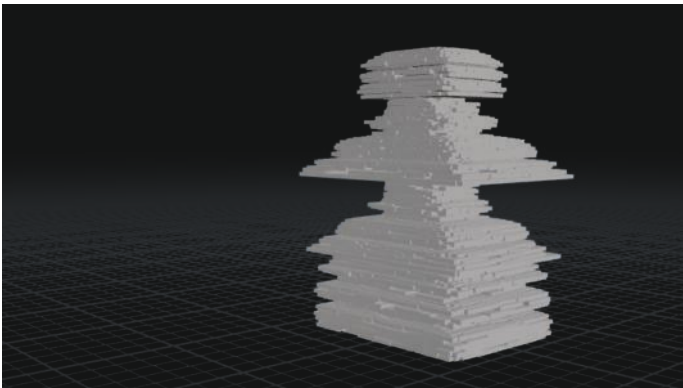
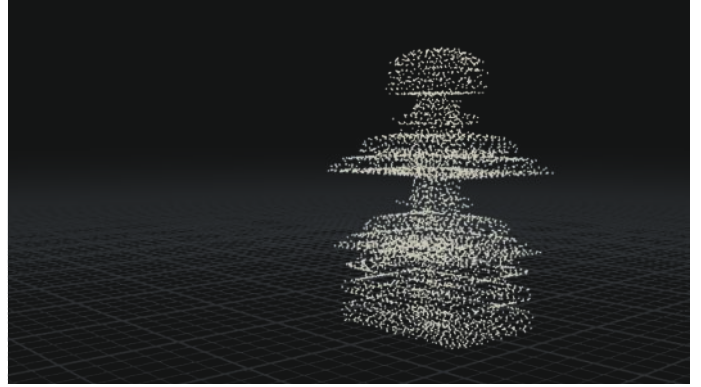
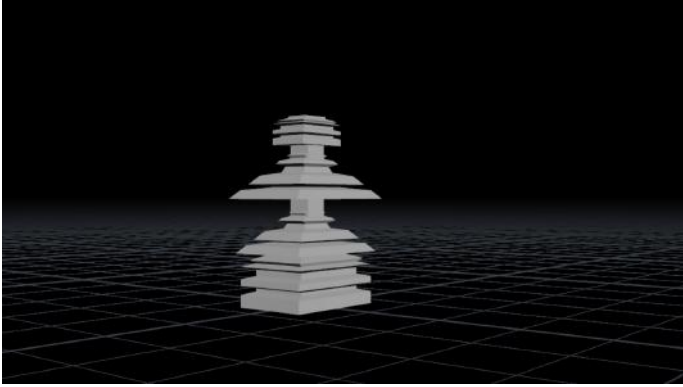


I ended up liking the procedural spline modeld chunks best. They came with more possibilities and freedom.



## Week 6

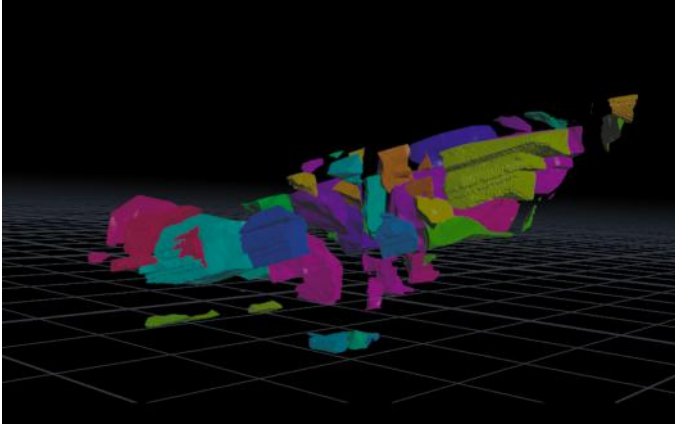
After some time I ended up with four models I really liked. To finish them I scattered points on the surfaces and copied boxes on them to give it a less clean and older look.





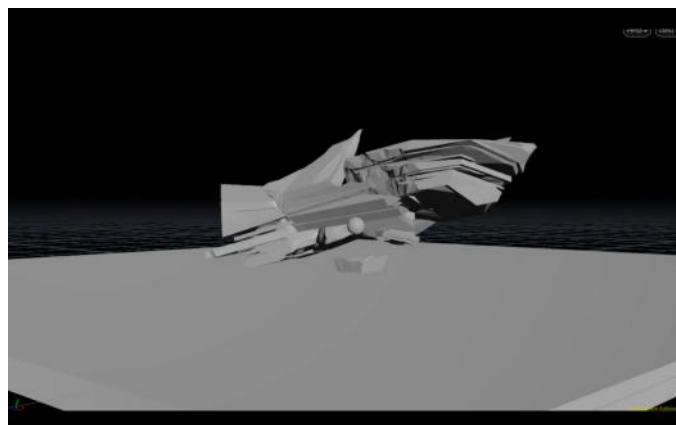
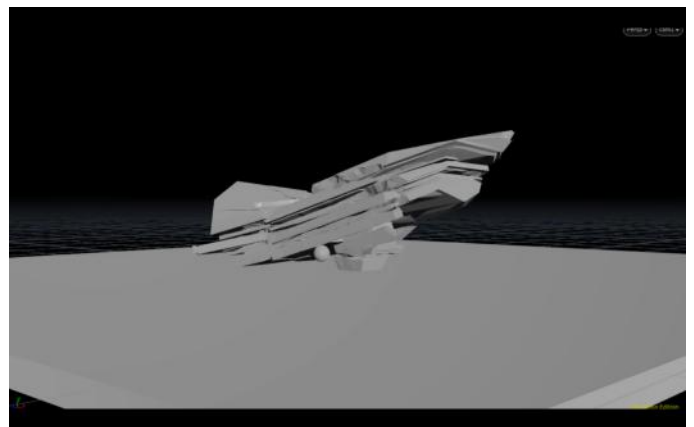
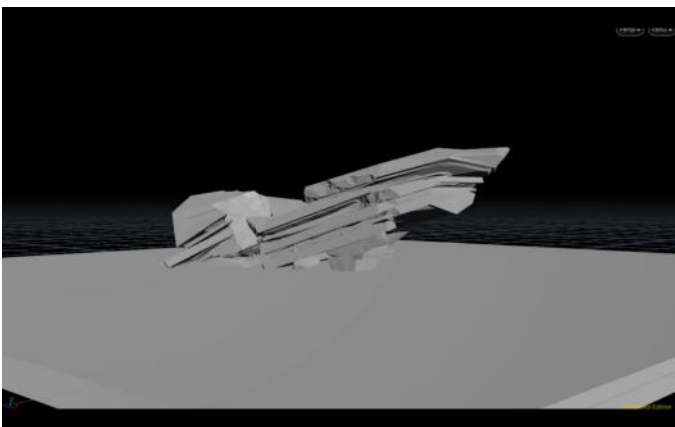
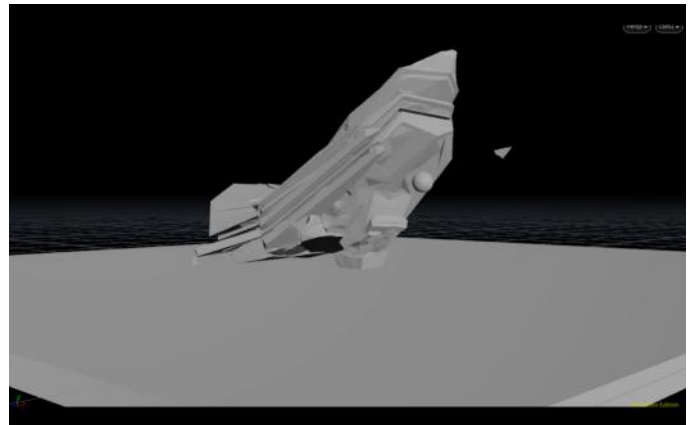
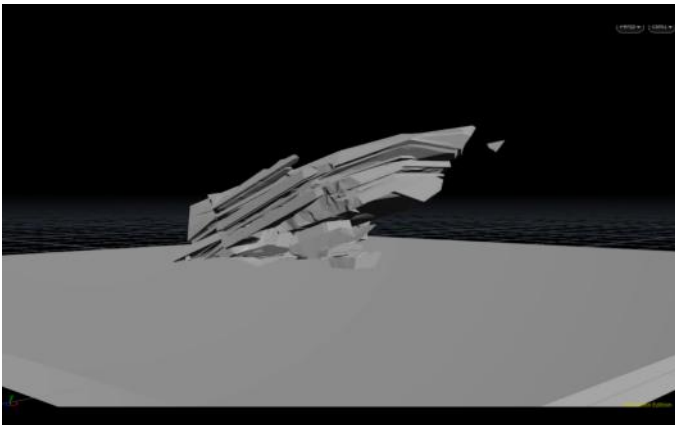
# Destruction

## Week 6



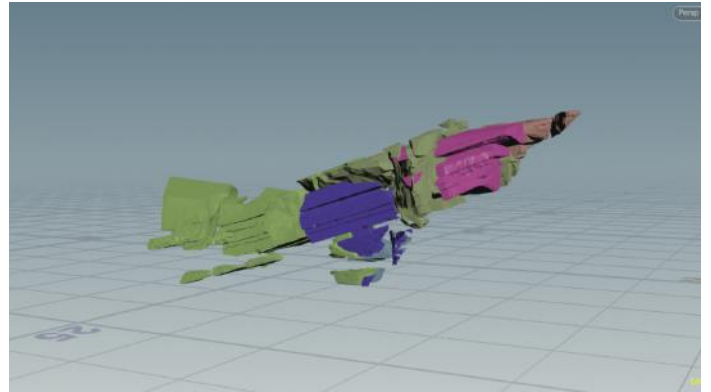
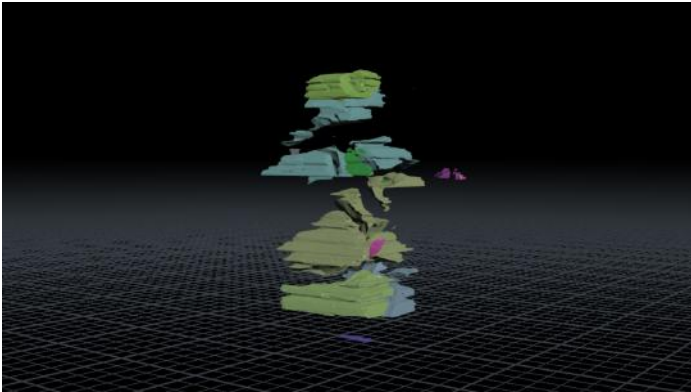
For the destruction I first made custom fracture geometry to fracture the ruin.

After making the constraints and setting up the Bullet solver I had my first version of the destruction.

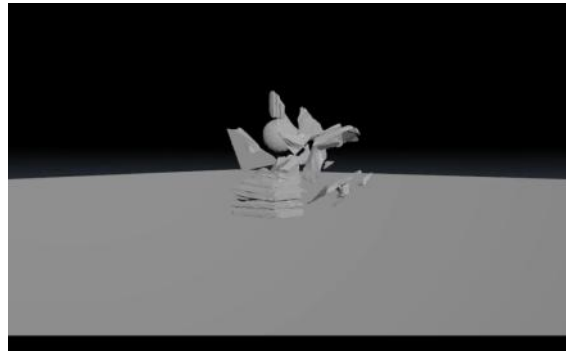
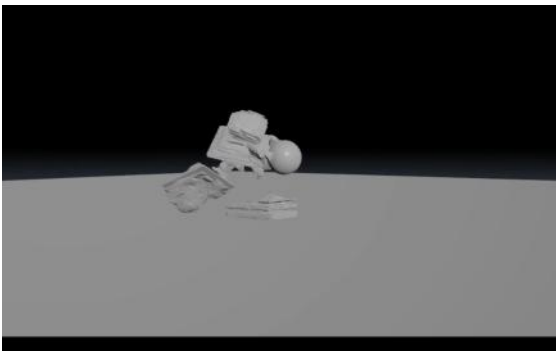
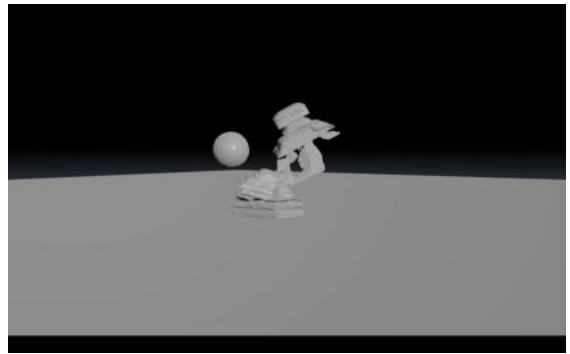
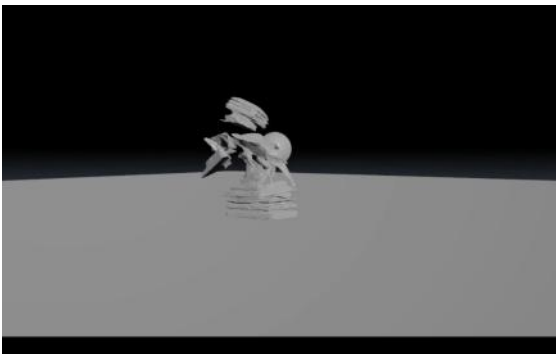


## Week 7

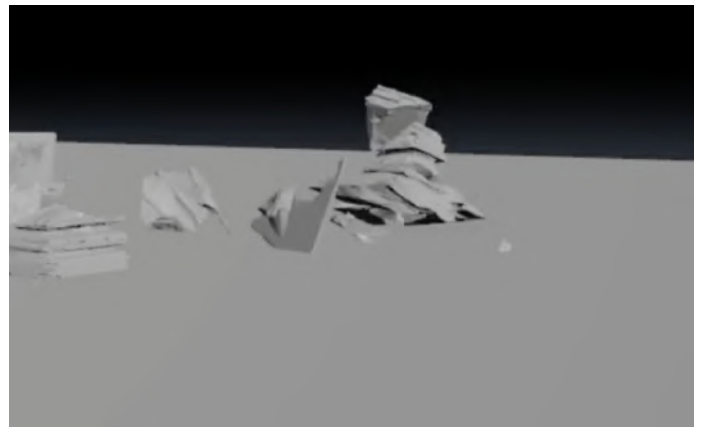
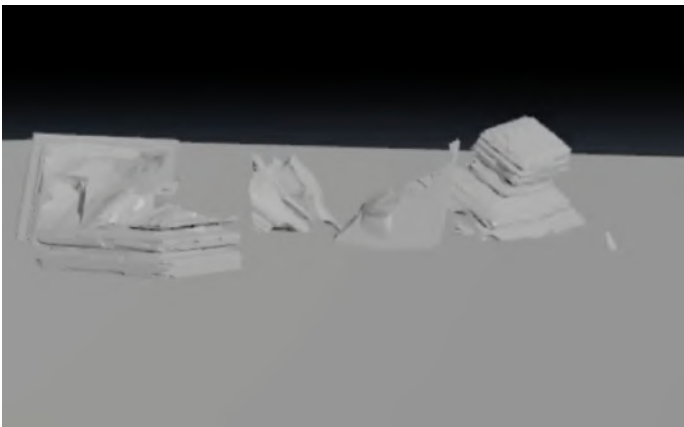
To make the destruction more realistic I made more fractures and clustered them (wich is visualized with color in the pictures below). With this we can make two different constraints with different strengths. The constraints within the clusters will be strong. The constraints that hold the different clusters together will be weaker.



Afterwards I tested the result again in four different angles

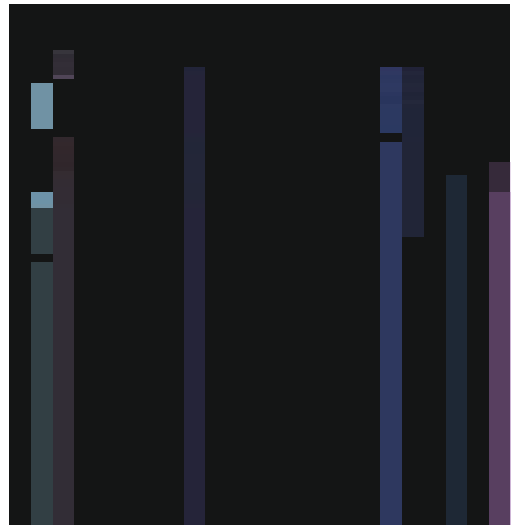
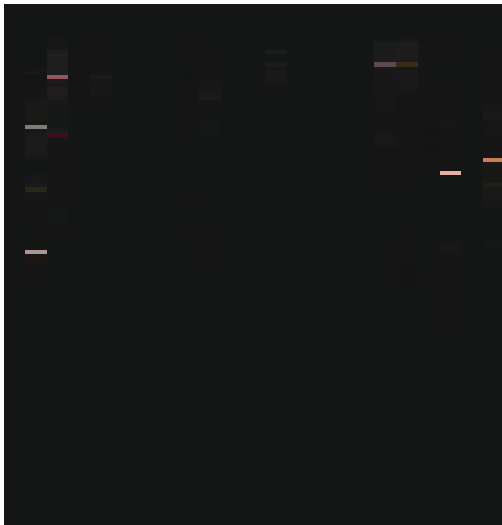


In these pictures the effect of the clusters is very visible. As the ruin gets hit it breaks into big chunks, but when these big chunks hit the ground they break again into smaller pieces.



## Week 8

The animations are exported using Vertex Animation Textures. This new technic is really amazing. It saves the animation data in color textures wich are very light weight and small in size, unlike Alambics wich can take an exteme amount of storage.



After making the Blueprint for the destruction in Unreal I had my first version of my destruction game.

Every ruin has eight different animations. Wich animation gets playd depents on what angle it's getting hit from.



# Final destruction

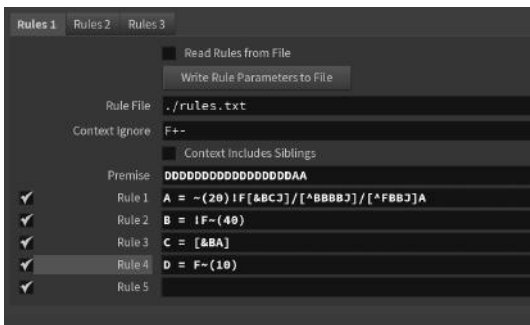
The little chunks, dust and the explosion is made in Unreal using Niagara particles.

After UV mapping and texturing I had the final destruction. The pre-baked animation from Houdini and the realtime Niagara particles make a great destruction together.



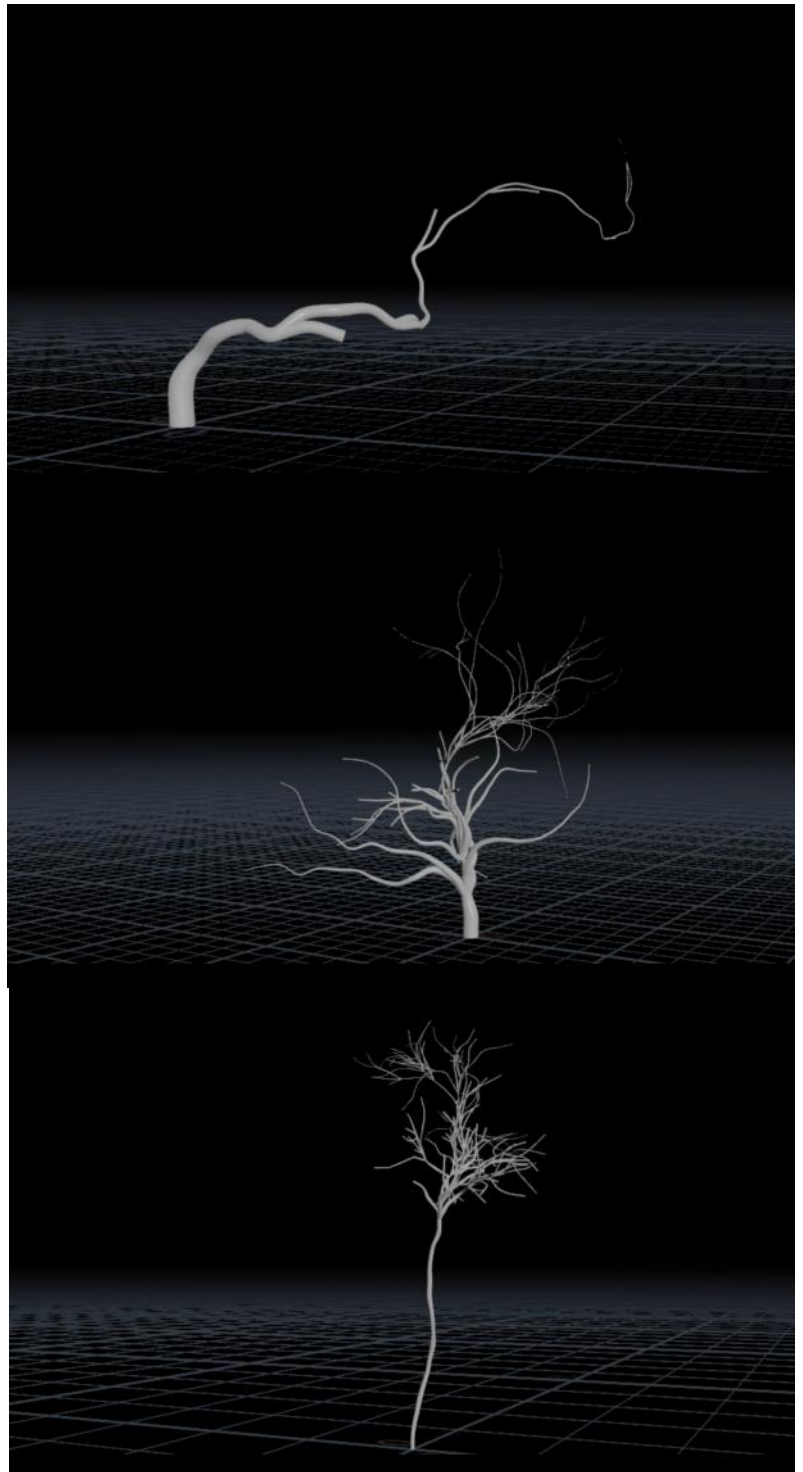
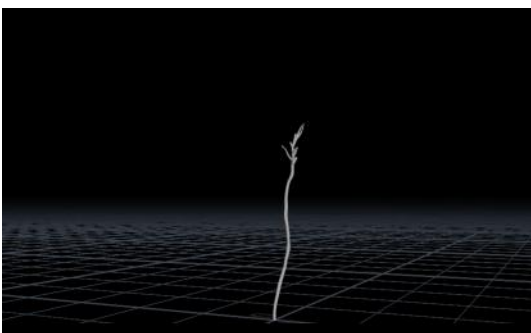
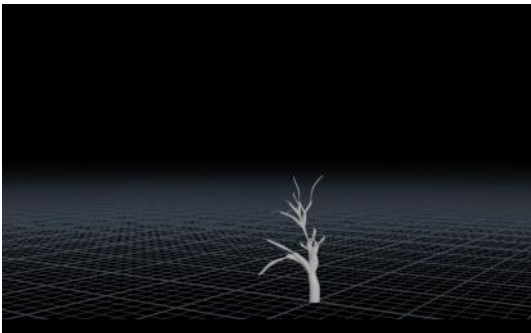
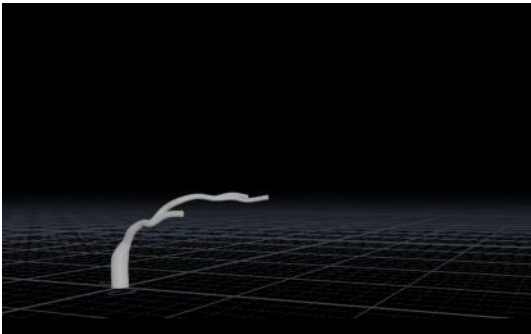


# L-System trees



The trees are made with the L-System node in Houdini. With this node you can write a very simple kind of code with instructions of how the plant must grow.

The powerful thing is that you can adjust the generations (growth size) and random seed of the plants so you can create an almost infinite amount of plants with only one system.



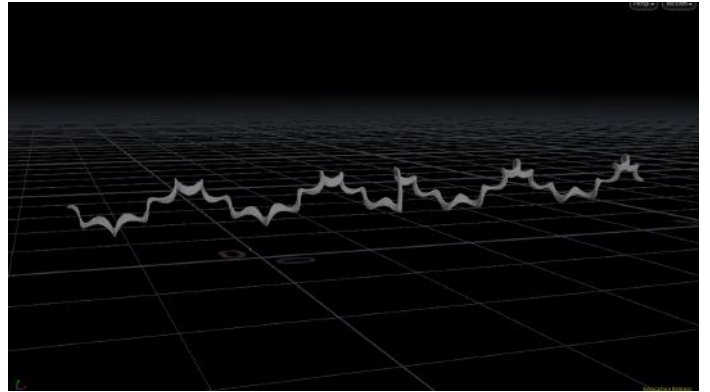
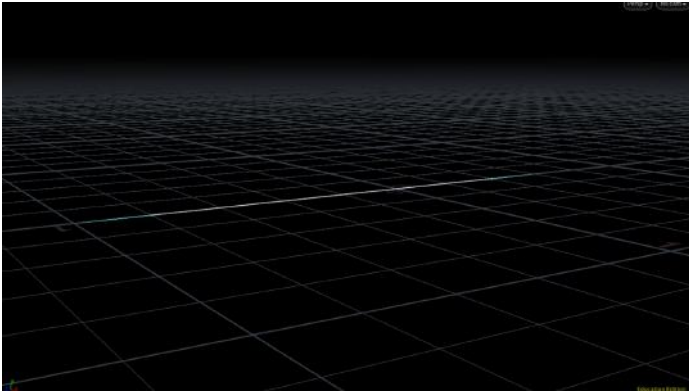
In the game I made the L-System trees look very different from the Unreal marketplace trees. Since the marketplace trees are made by professionals it's very hard to compete with its quality. The trees are also not the focus of this project, so spending an incredible amount of time on them was no option.

The thought behind the look of the trees is that they're some kind of poisonous plant. The fleshy texture and Niagara particles that float around give it a dangerous look.

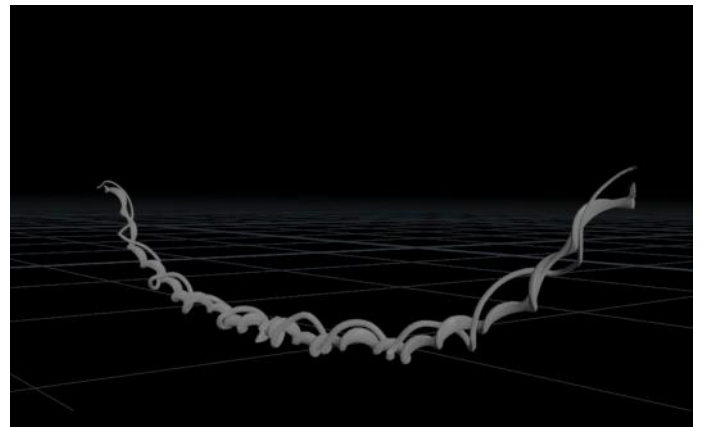
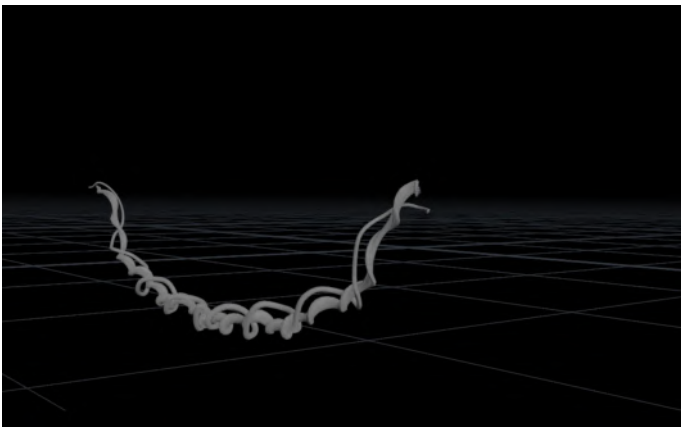
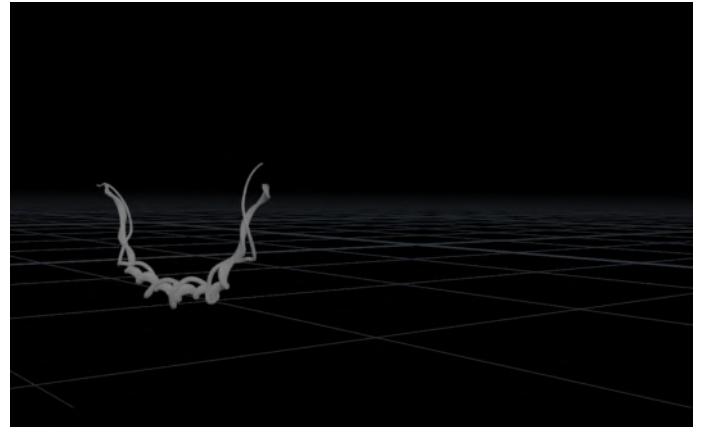
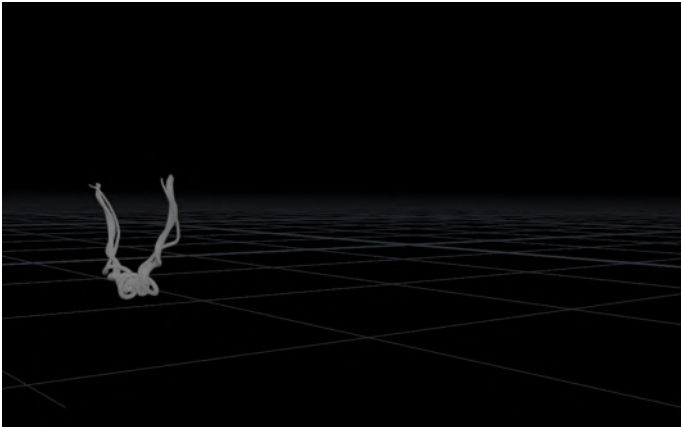


# Vine generator

For the vine generator I started out with a line with one point on each side. With some clever vector math I created twirls and noise in the line so it has the look of a vine.



To edit the length of the vine you only need to increase the distance of the points that makup the initial line.





After making the generator into an HDA I imported it into Unreal. By using parameters, like the distance I spoke of earlier and thickness, I can now create custom vines in my game level.





# Conclusion

It's incredible how much I've learned in such a short time. Before the course I had zero knowlege of Houdini, procedural systems and Unreal.

Now, three mounts later, I feel like I have quite some knowledge about all these things in regards to destruction.

The following time and the final 1,5 years of my beachlor I will focus on improving on this field and implementing it in my projects and future jobs.

A big thank you to all my teachers and class mates.



# Creative Simulation Technologies 2022

## Participant case studies

1. Agne Danieliute-Granickiene
2. Dimitris Kyriakou
3. Jüri Lääs
4. Karol Wójcicki
5. Marisa Naemi Kristin Liebner
6. Neele Katharina Siemann
7. Paraskevi Malamatenia
8. Patrick Baitinger
9. Sergio Esteban Casilimas
10. Sylvia Rybak
11. Victor Fredrik Weijers

**12. Yuliia Silkina**



Creative  
Europe  
**MEDIA**

## Meeting the Giantess



# Case study

## Concept/Storyboard

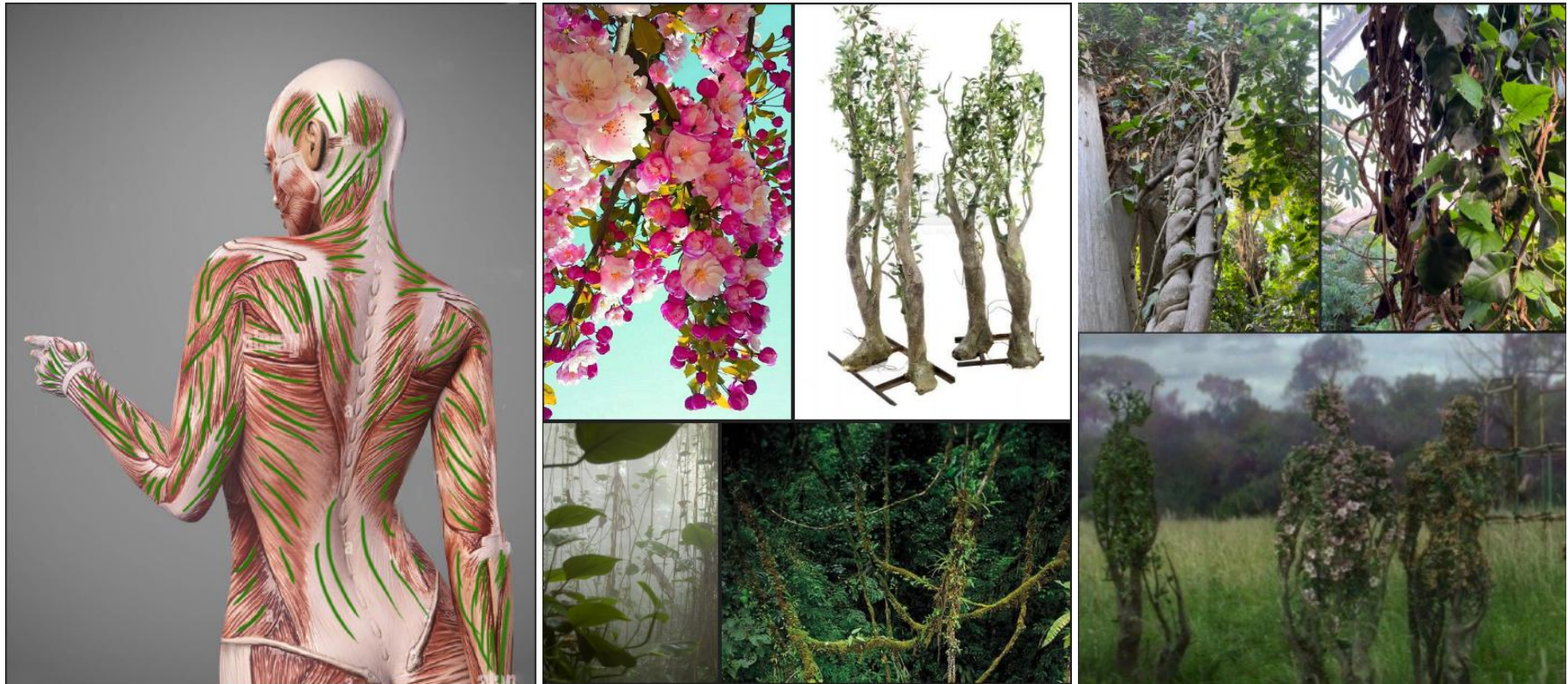


The feeling I wanted to bring to the viewer is the power of authenticity, majesty of love and nature.

The main character is made out of liana veins and spring flowers, and acts like a goddess. She is all in one: sacred, scary, thrilling, flirting, dancing, blooming.



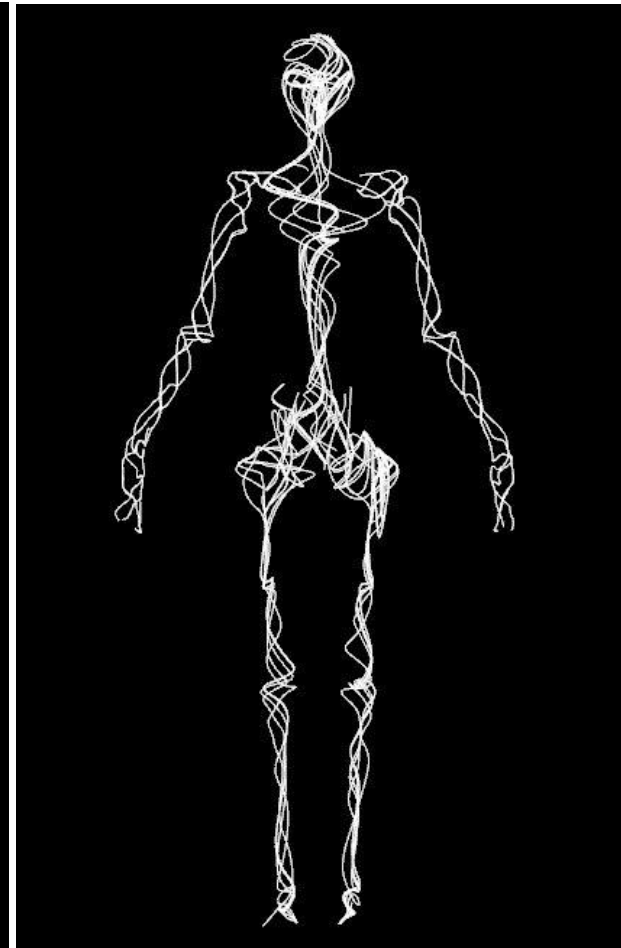
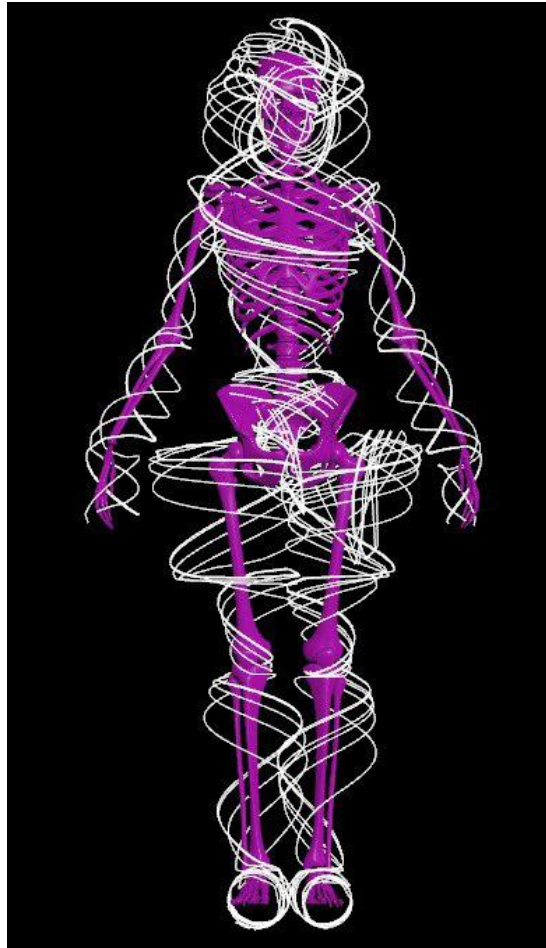
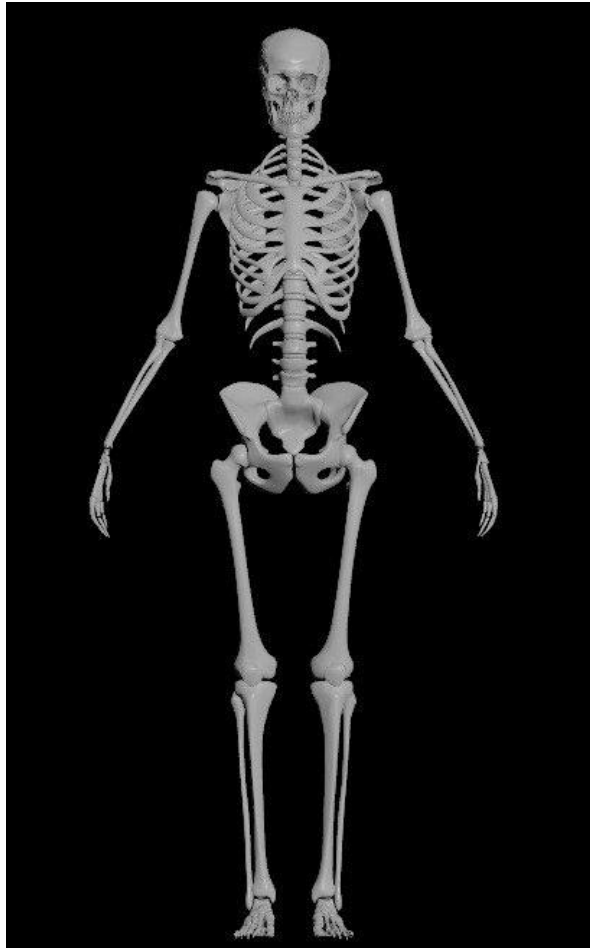
## Reference board. Idea development



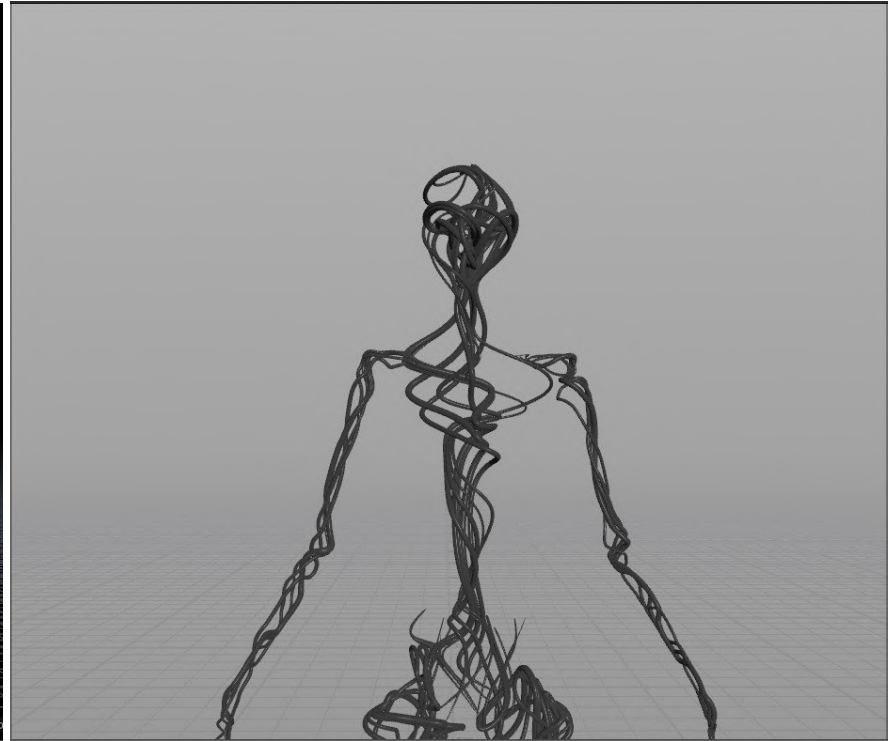
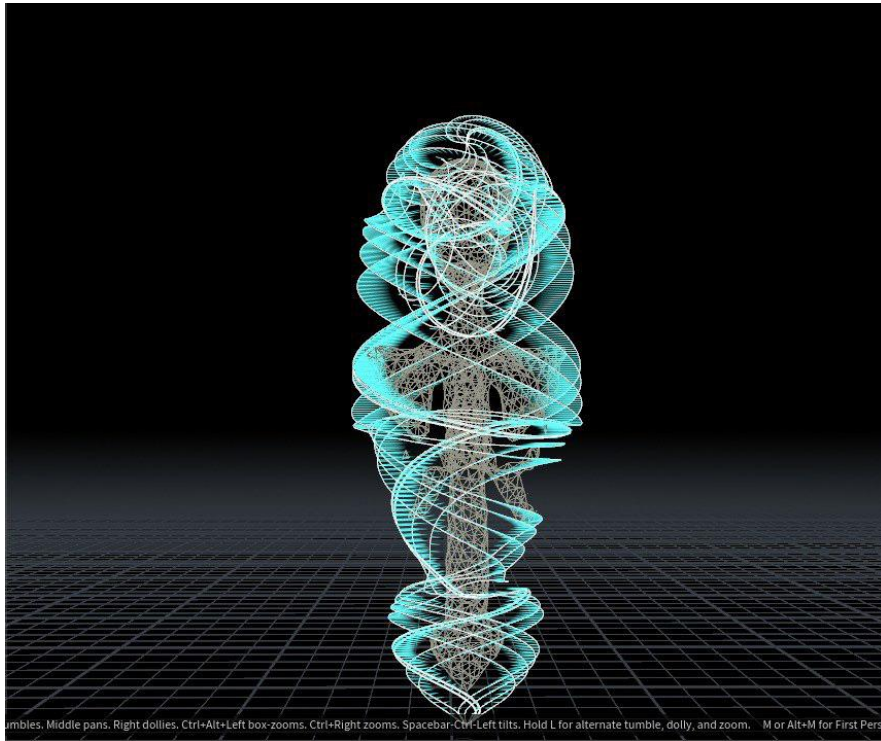
The plan was to grow veins following each muscle's fiber. The movement of arms split into multiple arms has to be done by literally tearing apart existing arms and regrowing the thickness of each arm.

## Houdini project development phase

### Skeleton System



I started with anatomically correct female skeleton. I've created twisted lines that are easily adjustable and replicated this approach per each limb and torso with the skull.



Once satisfied with the lines, I've rayed them on the skeleton and added the variety of thickness to the veins.  
The blue lines on the top image show the direction of lines' projection onto geometry.



## Muscles System



The system was created based on one muscle. I've scattered starting points (yellow dots) on the intersection of the muscle with skeleton veins and ending points (red arrow) at the start of the muscle for it to inherit the anatomical flow. I smoothed the result and added natural thickness. Then I ran all the body muscles through this system.





## Procedural adjustments

- Since I can procedurally adjust the skeleton lines twist and offset, and have the muscle system reacting to the changing incoming data. I've generated 4 other sets of arms slightly different from each other for an organic look.

## Simulations

- You can see vellum simulation done to show the ripping effect of additional veins on the arms split.
- The particles simulation for leaves and debris to fall where it's appropriate.

## Animation

- The character was rigged and animated using KineFX in Houdini. I've used mixamo animation as the basis and layered keyframe animation to make it custom.

## Export preparation into Unreal Engine

- The colors on the image above are Houdini attributes data represented as vertex colors to drive the textures animations inside Unreal Engine.

### Additional assets created in Houdini



- The Knight's cape simulation.
- An animation of a blooming flower.

Both assets were baked and exported as Vertex Animated Textures in Unreal Engine.

## Unreal Engine project development phase

### Character setup



- Imported animated character with all hands
- Created animated materials: the ending of each hand has wave-like movement, skeleton and muscle veins are pulsing with different rhythm all over her body.
- Niagara particle systems: two types of flowers are spawning, blooming and sliding along the veins, two colors of leaves also spawning and growing.



## Composition, landscape and lighting setup



Here I'm using Metahuman as a placeholder to set up the composition. I've bought landscape and the Knight to speed up the development process.



I've added tearing veins and falling leaves simulations from Houdini. Adjusted lighting, atmosphere and rendered in Unreal Engine.

Thank you!