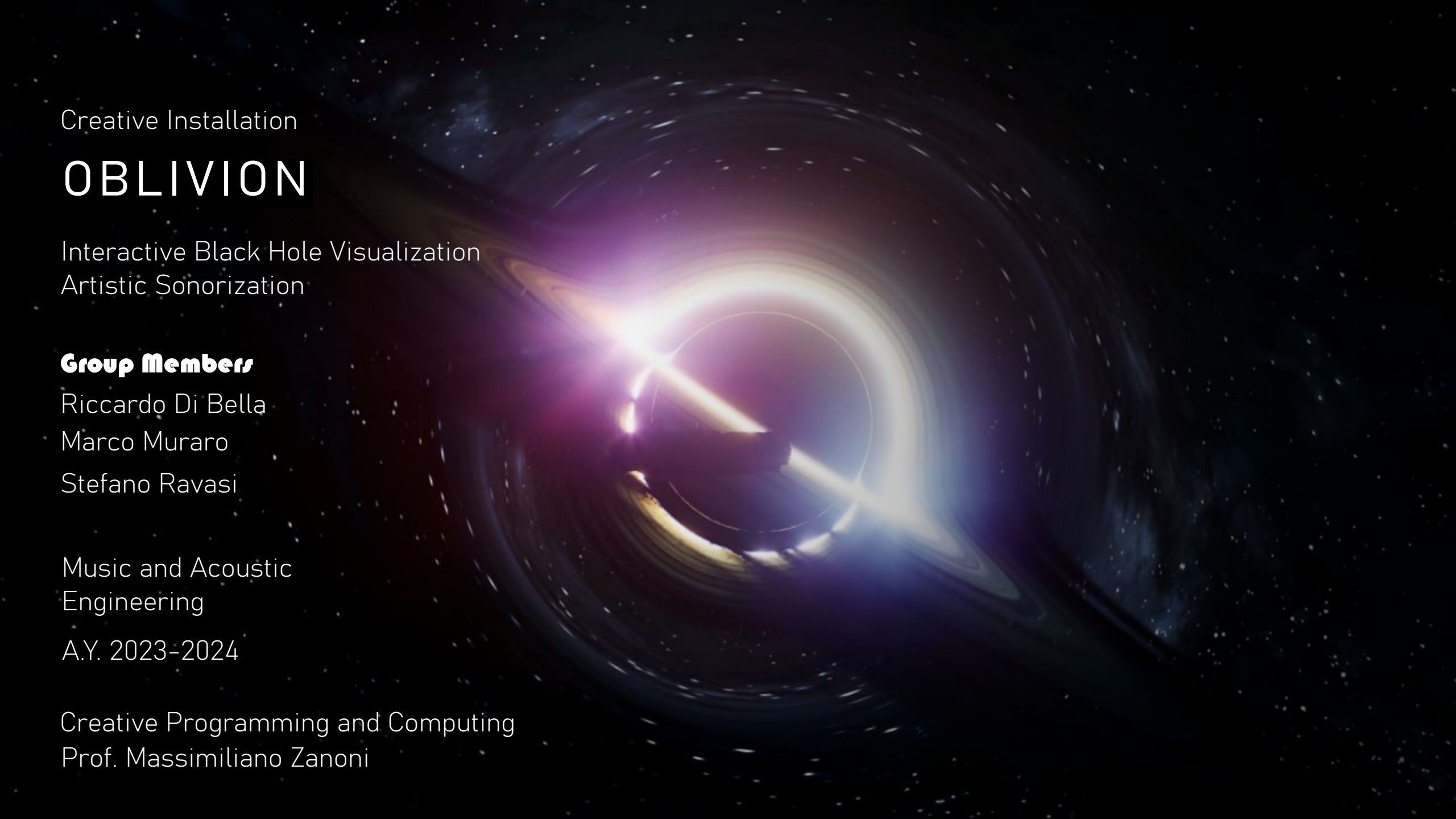


Creative Installation

OBLIVION

Interactive Black Hole Visualization
Artistic Sonorization

Group Members

Riccardo Di Bella

Marco Muraro

Stefano Ravasi

Music and Acoustic Engineering

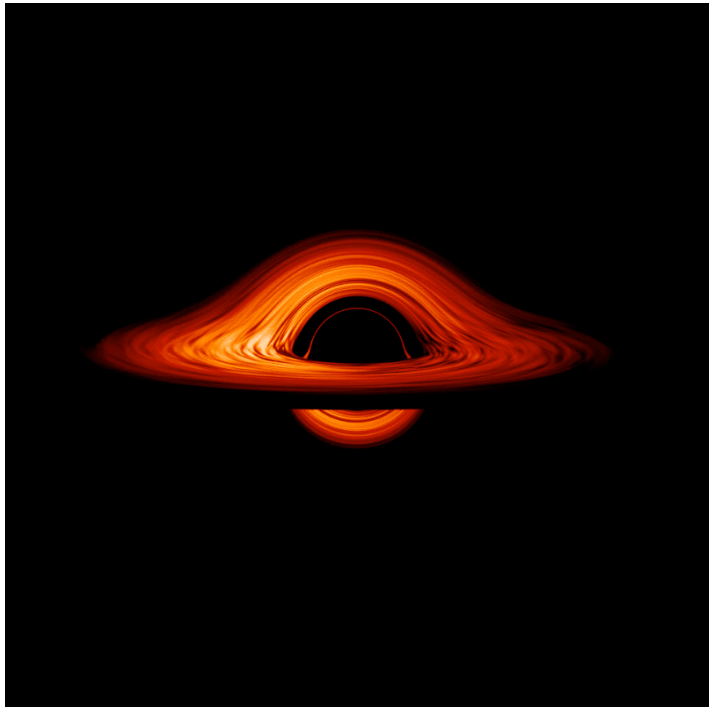
A.Y. 2023-2024

Creative Programming and Computing
Prof. Massimiliano Zanoni

What is a black hole?

Huge concentration of matter packed into a very tiny space, called **singularity**.

A black hole is so dense that the gravitation pull just beneath its surface, the **event horizon**, is so strong that nothing, not even light, can escape.

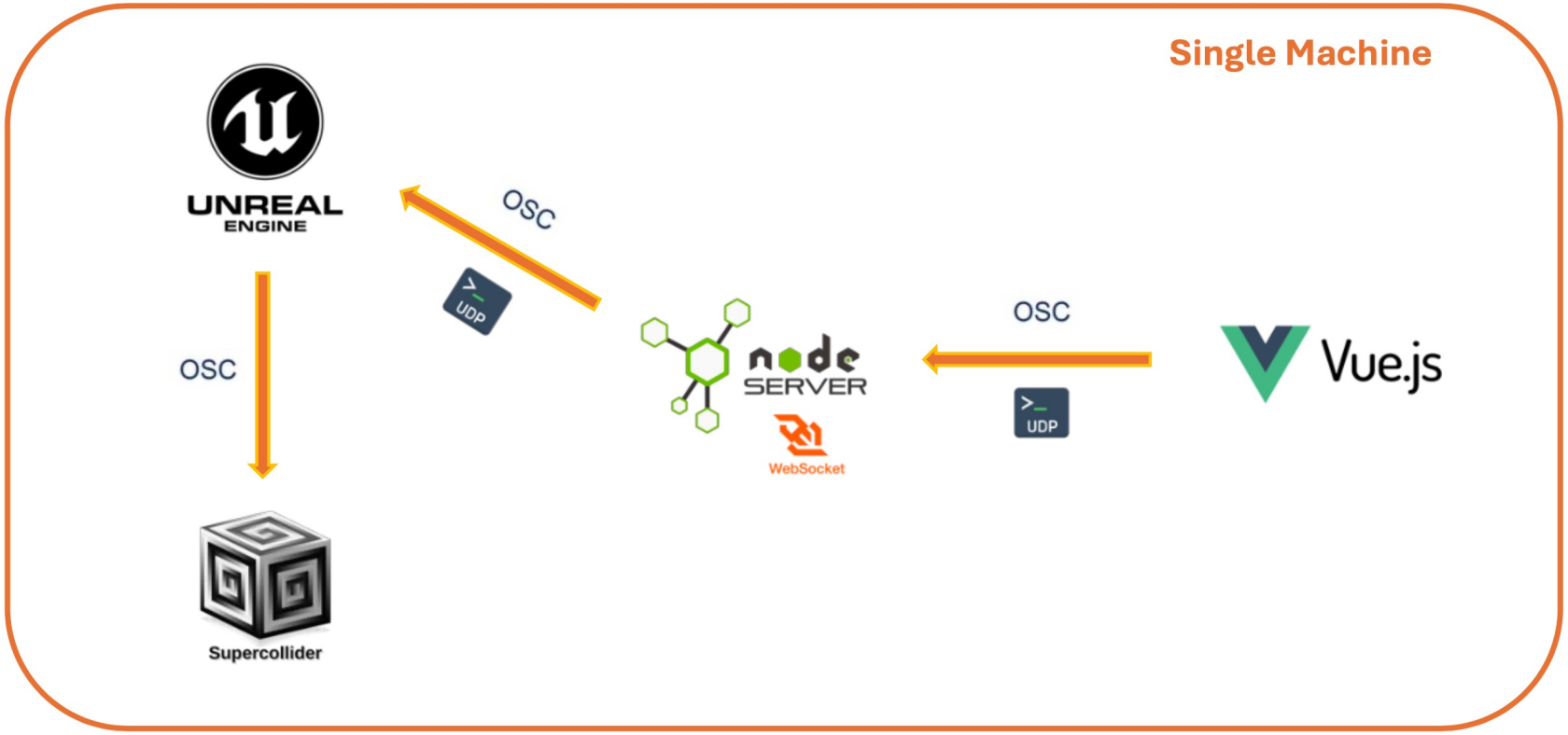


Accretion Disk

Some black holes are surrounded by **hot, swirling matter progressively falling inward towards the event horizon**.

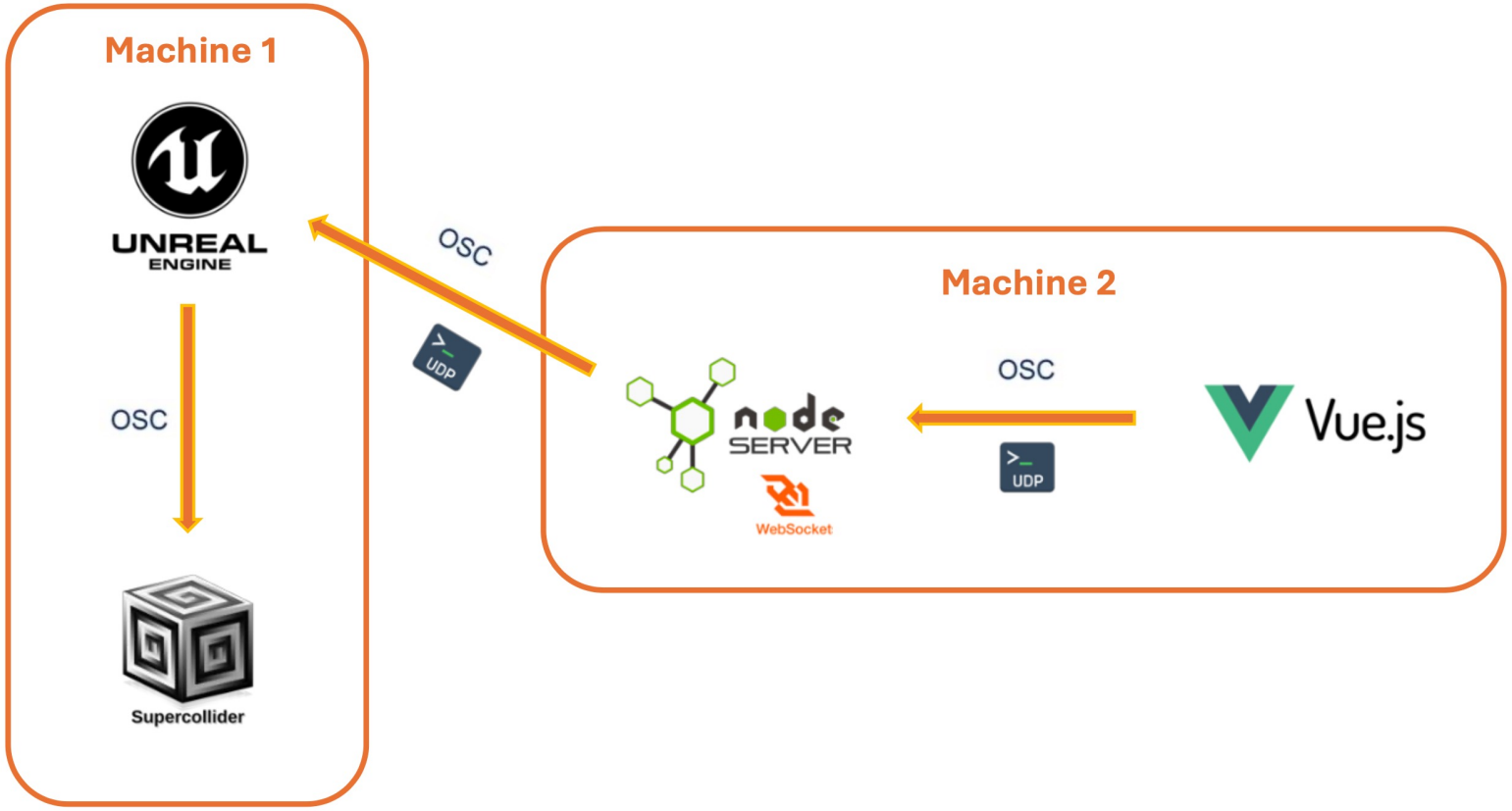
Architecture

Single Machine



Architecture

Separate Machines



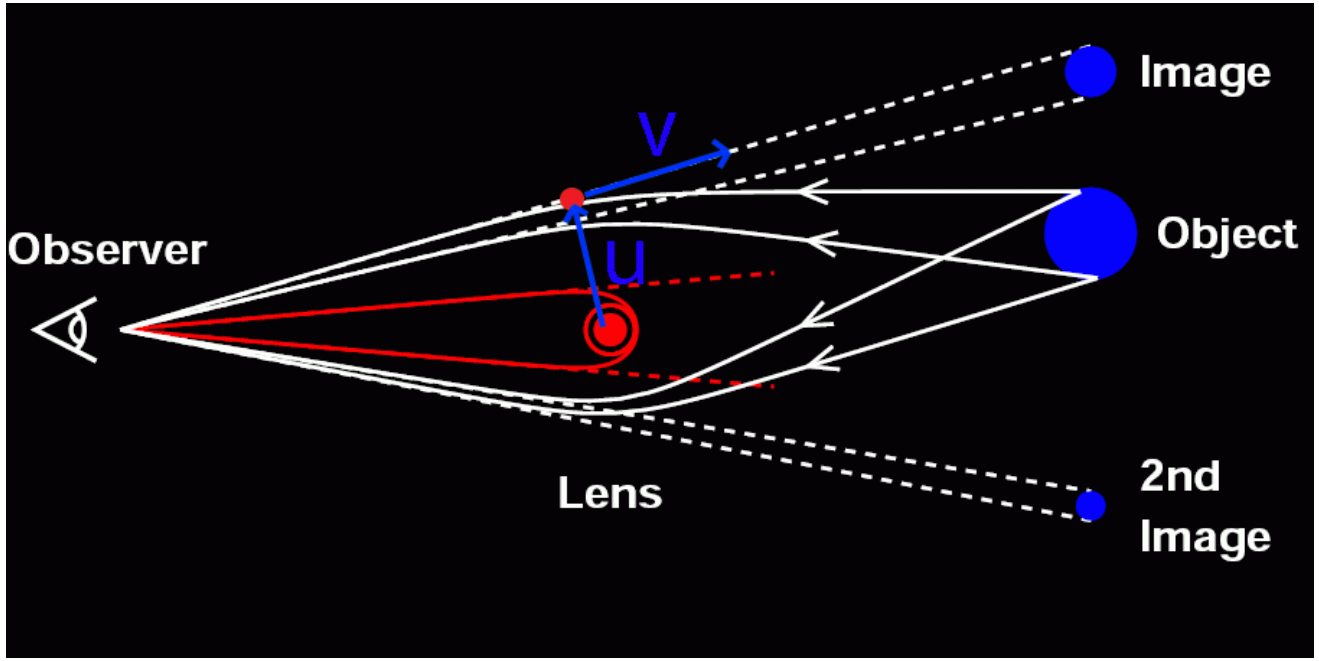
Black Hole Rendering

Implemented as a custom **shader** inside Unreal Engine

- **Ray Marching** technique
- Scene inside a sphere covered by sky space texture

Gravitational Lensing – simplified model

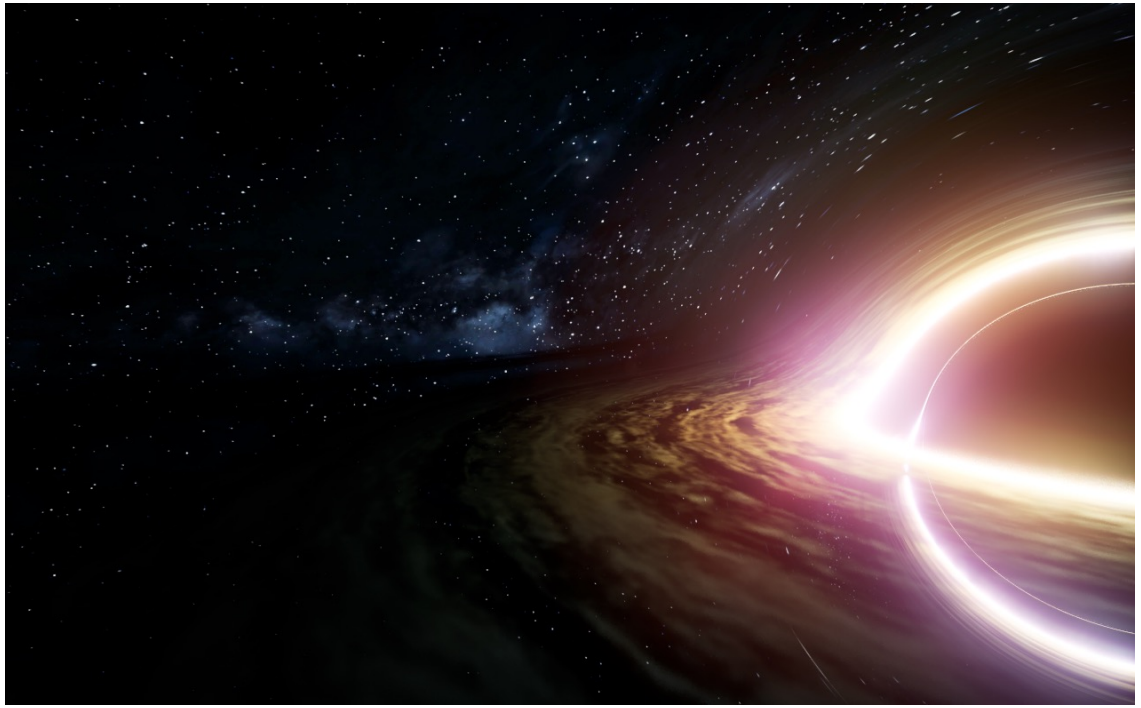
$$\vec{F} = \frac{-3M\left|\left|\hat{u} \times \hat{v}\right|\right|^2}{\left|\left|\vec{u}\right|\right|^2}\hat{u}$$



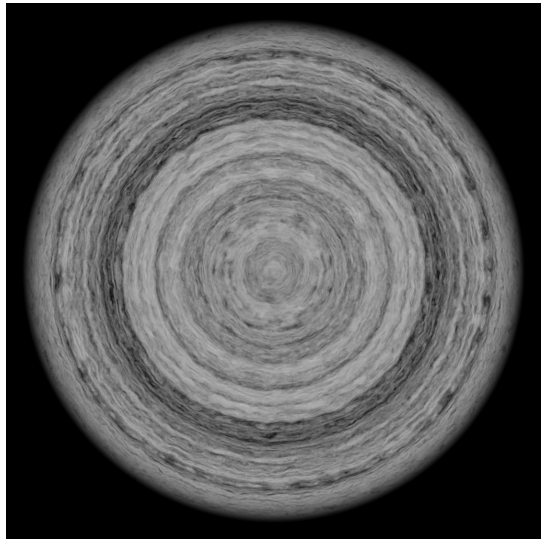
Black Hole Rendering

Accretion Disk – Rendering Process

- Check intersection with disk region
- Compute **density** using pre-defined texture
- Compute light intensity using **square roll-off**
- Compute shadow (light absorption) using **volumetric ray-marching**
- Sum contribution of **disk** and **fog layer**



Disk texture



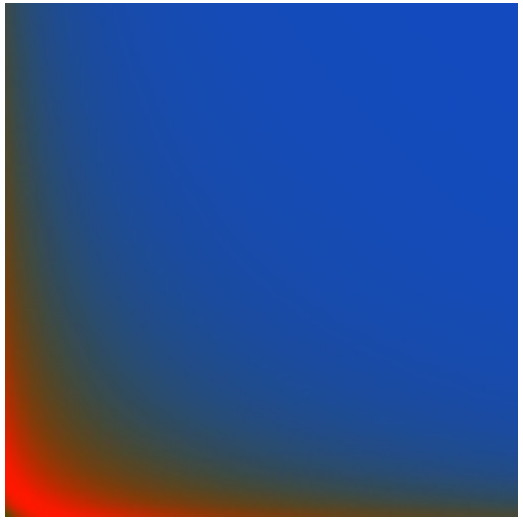
Black Hole Rendering

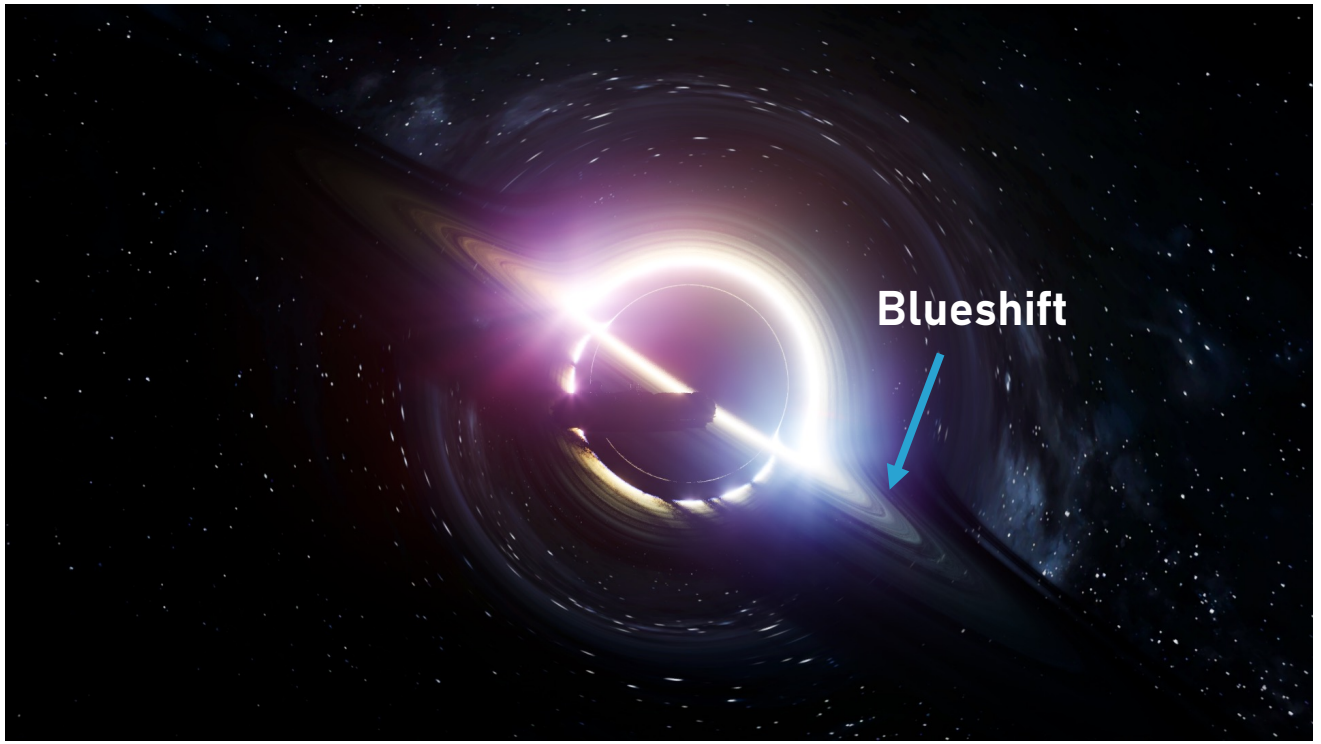
Accretion Disk – Color

Color is determined using a **black body radiation** map

- Sample according to temperature and frequency shift
- Temperature decreases with distance
- Frequency shift due to motion (**doppler effect**)

Blackbody radiation map





Generative Music

Everything implemented using **SuperCollider**. The **OSC** communication protocol makes it communicating with Unreal Engine.

The main goal is to **emulate sci-fi music in a generative fashion** while maintaining a certain grade of control on the algorithm.

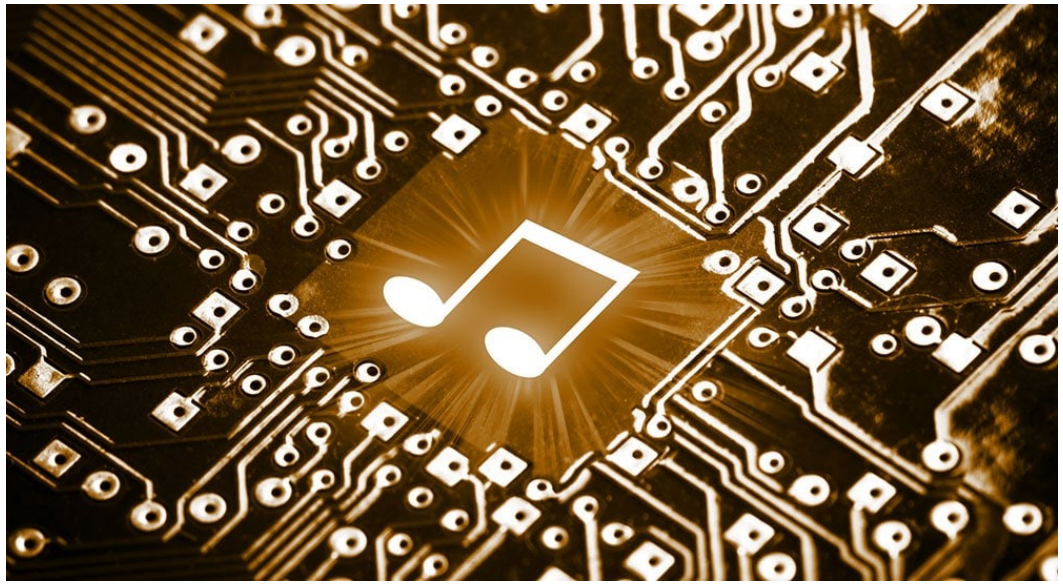
The aim of this algorithm is to **enlighten the vastity of the universe and the immense power of a black hole**.

Generative Music

Different kind of synthesis:

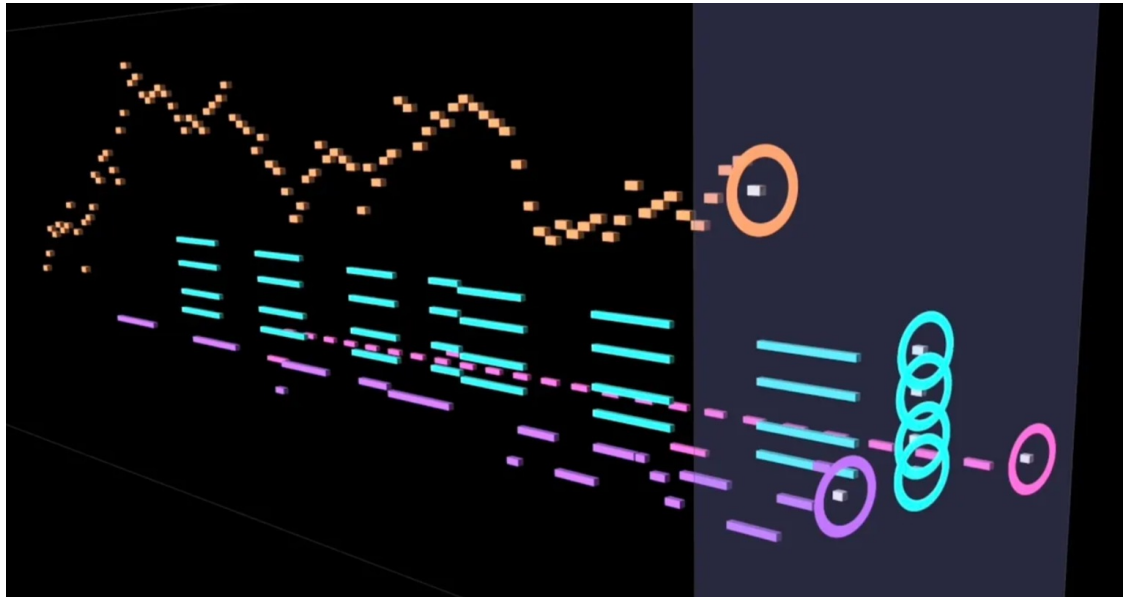
- **Subtractive** (main background)
- **FM** (modulated FX and "particle noise")
- **Wavetable** (Ringing sounds)
- **Digital Waveguide** ("gusts of wind")

All the synths are processed by a **reverb** effect which also provides **companding** and **limiting functionalities**.



Generative Music

Fundamental frequencies are picked up from a more restricted set of values while all the other parameters are randomly generated within a pre-established range. All the sounds are generated regularly in time with a minimum grade of randomness.



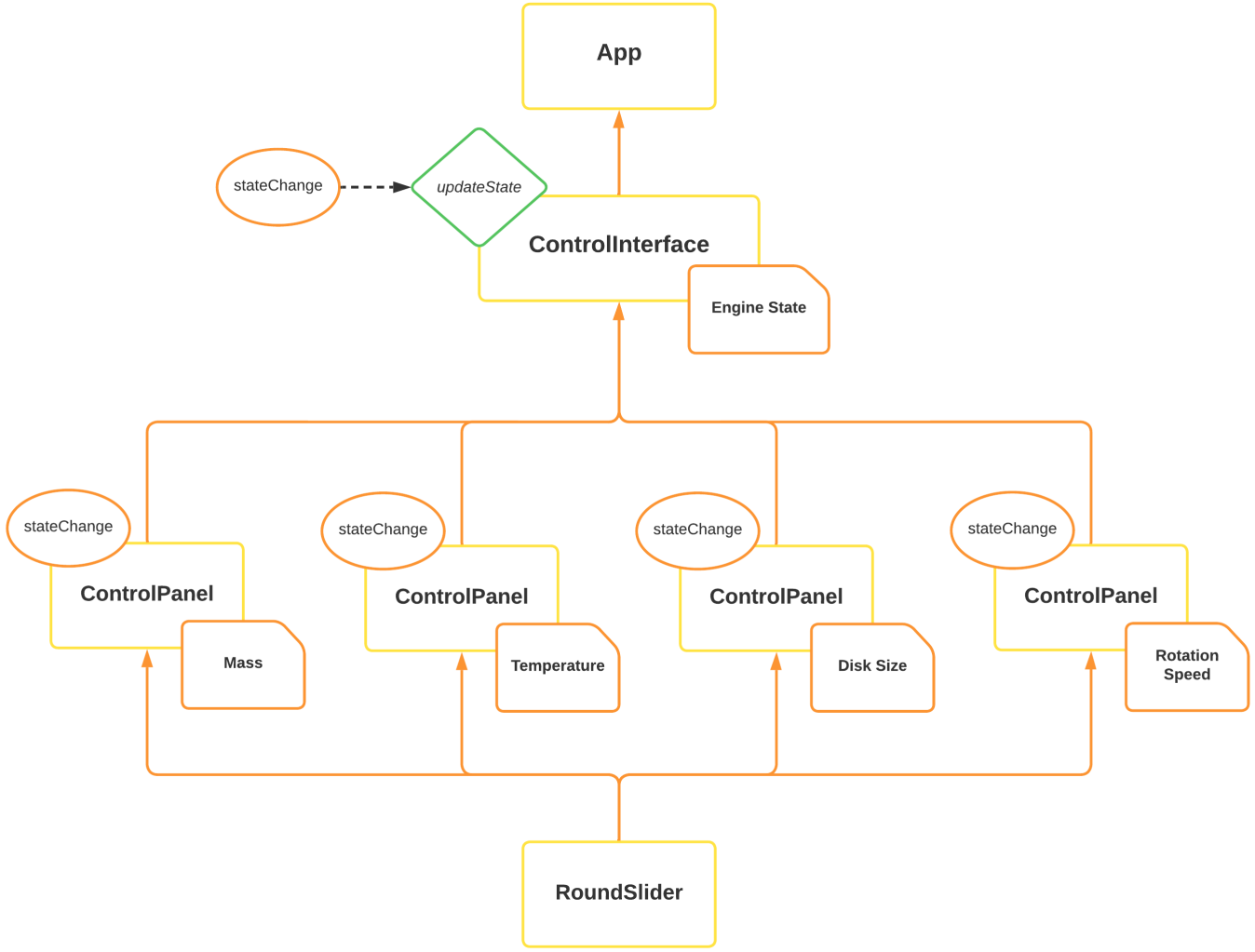
Parameters from the physical simulation influence the soundscape:

- **Mass**: low frequencies amplitude
- **Temperature**: "particle noise" amplitude
- **Disk Size**: mid frequencies amplitude
- **Rotation Speed**: high frequencies amplitude
- **Distance**: FXs amplitude and rate of occurrence

Web Application

Vue App

Web App Architecture



Engine State

```
14  export default {
15      name: 'ControlInterface',
16      components: {
17          ControlPanel
18      },
19      data() :{engineState:  {
20          return {
21              title: 'OBLIVION',
22              engineState: {
23                  mass: {
24                      name: 'Mass',
25                      label: 'mass',
26                      unrealLabel: 'Mass',
27                      value: 5,
28                      min: 1,
29                      max: 20,
30                      step: 1,
31                      display: 'up'
32                  },
33  >               temperature: {name: 'Temperature'...},
43  >               diskSize: {name: 'Disk Size'...},
53  >               rotationSpeed: {name: 'Rotation Speed'...}
63              }
64          }
65      },
```

The global state of available parameters is defined as an object, named **engineState**, in the **ControlInterface** component within its data function.

This object holds on its turn multiple objects, one for each parameter control displayed within the GUI.

Each one of these objects is passed as a property to the corresponding ControlPanel instance (child component) for initialization of its round slider.

POLITECNICO MILANO 1863

Engine State Update

v-model directive automatically updates the **internal state** of the single **ControlPanel** component.

A **Vue Watcher** is employed for each instance to emit **stateChange** custom event that is dispatched to the parent whenever the user modifies the single parameter value.

ControlInterface.vue | ControlInterface.js | ControlInterface.css | ControlPanel.js

```
32      watch: {
33          value() : void  {
34              //console.log(this.name + ' changed:', this.value);
35              this.$emit( event: "stateChange",  args: {label: this.label, value: this.
36          }
37      }
```

ControlInterface.vue | ControlInterface.js | ControlInterface.css | ControlPanel.js

```
66      methods: {
67          sendOSCMessage(update) : void  {
68              osc.open();
69              const message : Message  = new OSC.Message( address: '/' + update.unrealLabel, update.value);
70              osc.on( eventName: 'open',  callback: () : void  => {
71                  osc.send(message);
72                  //console.log('Message Sent!');
73              });
74          },
75          updateState(update) : void  {
76              this.engineState[update.label].value = update.value;
77              update.unrealLabel = this.engineState[update.label].unrealLabel;
78              //console.log(this.engineState[update.label].name + " updated: ", update.value);
79              this.sendOSCMessage(update);
80          }
81      }
```

ControlInterface.vue | ControlInterface.js | ControlInterface.css | ControlPanel.js

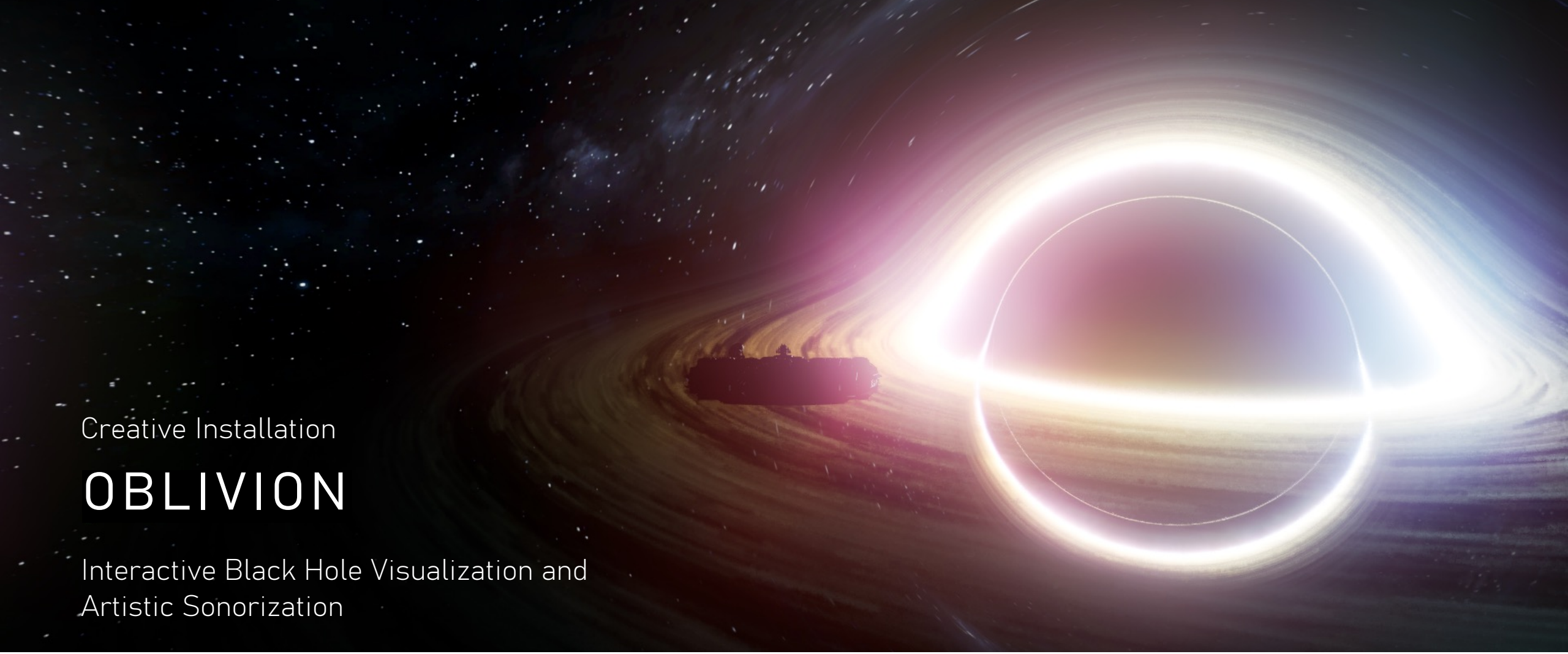
```
4       <div id="control-panel" class="flex-container-column">
5         <div class="controls-wrapper flex-container-row">
6           <div class="control-box">
7             <ControlPanel
8                 :controlParam="engineState.mass"
9                 @stateChange="updateState"
10            />
11          </div>
12          <div class="control-box">
13            <ControlPanel
14                :controlParam="engineState.temperature"
15                @stateChange="updateState"
16            />
17          </div>
18        </div>
```

OSC Communication over UDP

The **OSC Communication** between the **Web App** and **Unreal Engine** was implemented through a **Node WebSocket Server** listening to OSC Messages coming from an **OSC WebSocket Client** running within the Vue App.

OSC Messages are then **broadcasted over UDP** to the receiver, which could run on the same machine or on a remote one.


let's jump into Space!



Creative Installation

OBLIVION

Interactive Black Hole Visualization and
Artistic Sonorization

Group Members

Riccardo Di Bella
Marco Muraro
Stefano Ravasi

THANK YOU!

MAE
MASTER OF SCIENCE
MUSIC AND ACOUSTIC
ENGINEERING

POLITECNICO
MILANO 1863