



# **RESONATE 2015**

## **BURSARY APPLICATION**

**Bojan Čoka**

[metropolis.lights@gmail.com](mailto:metropolis.lights@gmail.com)

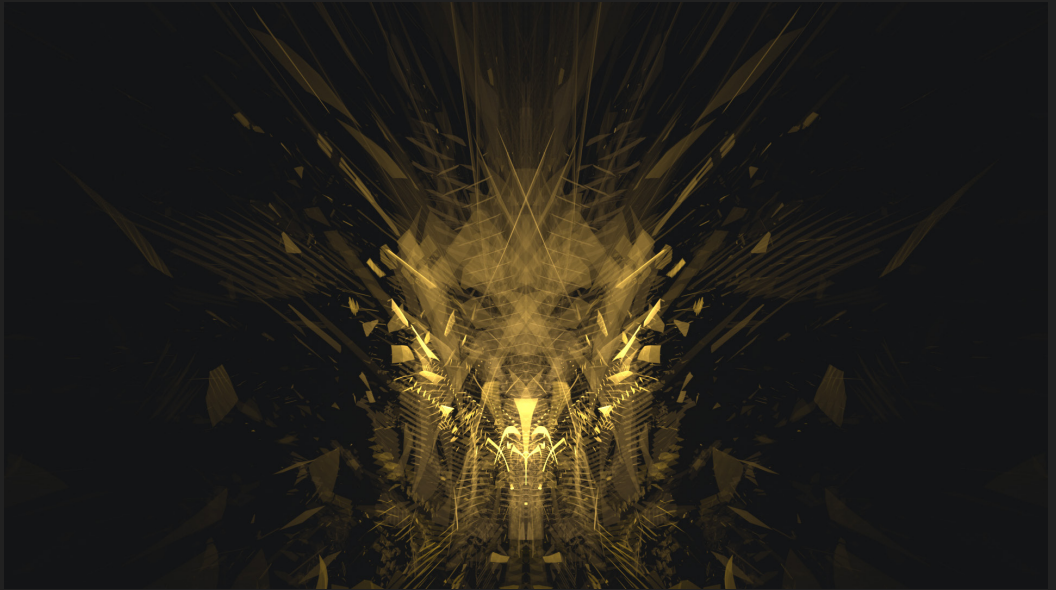
This page is intentionally left blank.

Hello there. My name is Bojan. I am 22 years old, and currently in pursuit of a degree in Computer Graphics at the University of Novi Sad. Last year, I attended Resonate for the first time, rather casually heading out to see what all the fuss was about. In hindsight, I can safely claim that I was completely oblivious to what I was getting myself into. Little did I know how prominent of an effect on me would being introduced to this hidden gem of a creative art form and creative community have. With the pleasure of meeting some of the most positively awe-inspiring people with their downright bonkers projects, and an opportunity to experience it all again, my decision is to not just talk about how flabbergasted I was, but to also show it. So, without any further ado...

# # 01

## Exploring caustics

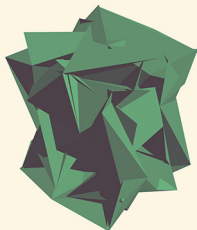
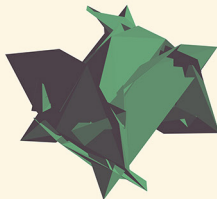
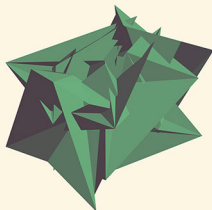
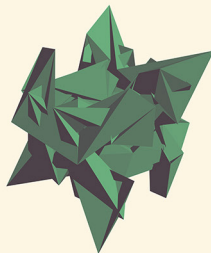
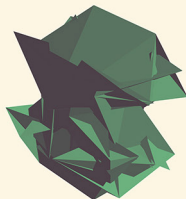
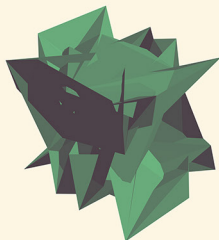
More than anything, what genuinely excites me about this field is how spontaneous creation can be. 'Twas a rather drab session of learning how to simulate caustics in digital content creation software, staring at the render output, until I was suddenly greeted by the face of a lion staring right back at me. The whole session took a turn for the more interesting as I pushed the idea further. The jist of the entire process is almost too simple to dare say - in goes some simple glass geometry, a couple of point lights, and a healthy dose of wishful thinking, and out comes imagery over which there is no real control. And that is exciting. Perhaps when privy to the modelled scene in it's entirety, the patterns might make some sense, but isolated, they can represent a space, or a face, or both, tucked away until light shines through at just the right angle. And so it comes to pass that a bent tube reveals the face of a regal lion, or a couple of star-shaped fragments of glass send a ship sailing through an infinite expanse. In the end, the artist is left the with the task of making a conscious choice of what is aesthetically pleasing.



# 02

# Cryptography visualisation

With the headlines drowning in talks of internet security, privacy, hacks, breaks, and heists, cryptography is in the spotlight. All this high-profile mathematics, promising indecipherable ways of manipulating data. Now, I don't fancy myself a mathematician, and I needed a more visceral way to see what encryption looks like. The algorithm I chose to visualise is RSA, a public-key cryptography method which relies on the difficulty of factoring prime numbers. The math behind it is genius in its elegance. But, just how difficult would it be for anything, man or machine, to glean from an encrypted representation any part of the original message? Understanding the public part of the algorithm reveals what is fundamentally an iterative process of transforming point coordinates by repeatedly applying the same rule, generating limitless variations. Oh, and in case anyone was wondering, somewhere in there hides a sphere.

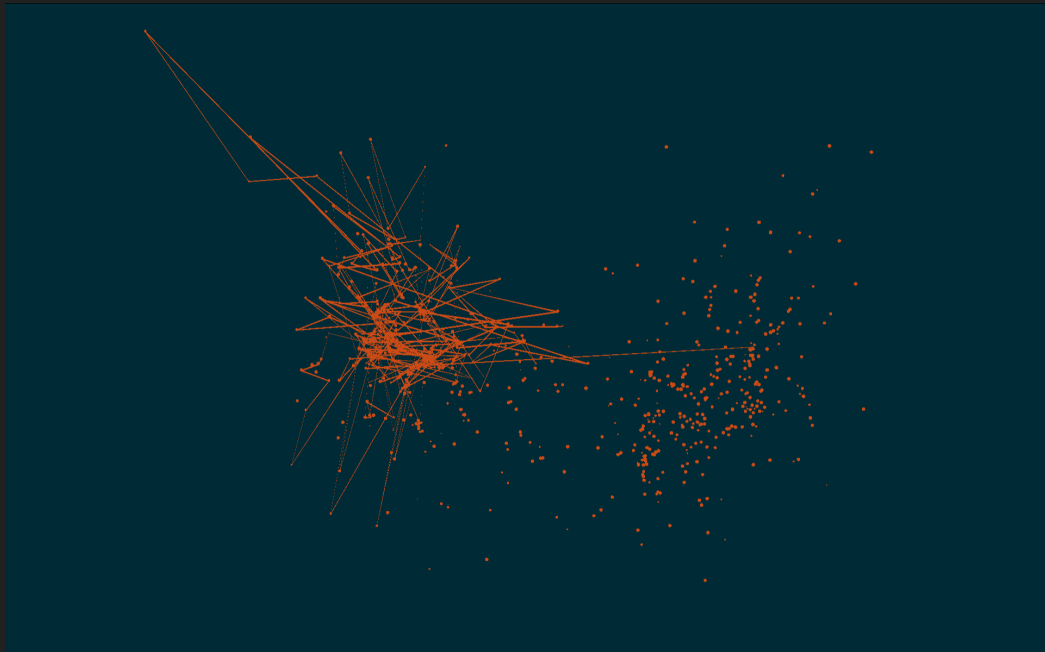
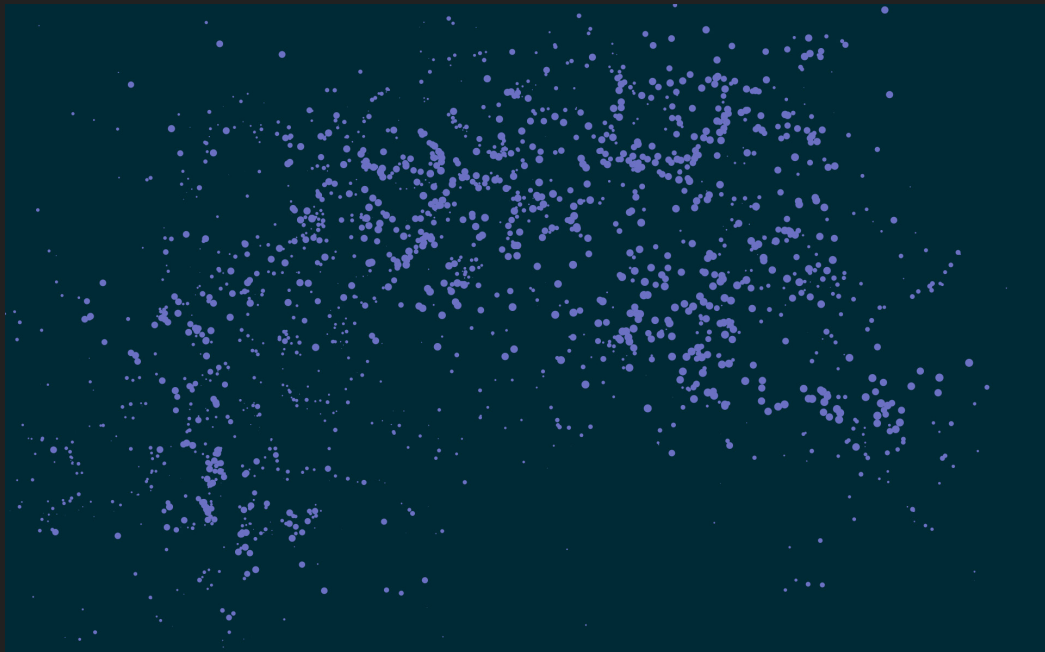


## # 03

# The mandatory particles

I wouldn't dare apply to any festival in the vein of this one if my portfolio wasn't worth its weight in particles. This little snippet of JavaScript is based on the fantastic "Hello, Cinder!" tutorial series penned by the even more fantastic Robert Hodgkin. It is almost a direct port of it, save for utilising pixi.js instead of the C++ framework, and a couple of personal tweaks. Switching to the WebGL renderer produces an exciting glitch, the root cause of which I cannot, for the life of me, determine. However, I think it has a certain appeal, echoing the motif of spontaneity mentioned before. And speaking of glitches...





## # 04

### **a.out > test.ppm**

I leave you with a bit (no pun intended) of programmer art. The driving force behind these pieces was wanting to see how much could be done with a single line of code expressing the relationship of colour values. The awfully primitive C++ programs cathartically shout curse numbers at the console, which are then piped into the simplest and niftiest of all the image formats - .ppm files. The vulgarity of it all leaves no room for an adequate title for the project, so may it be named after the shell command that created it all.

