WORK SHOWCASE

ANTONIO ESPINOSA GARCÍA

Render engineer

Researcher

- 2024 -

Table of contents

General info

In this document, I will showcase some of the projects that exemplify my expertise in the field of computer graphics. While not all of these projects stem from my professional career, I consider them equally important due to the valuable lessons I have learned from them.

As evidenced in this document, my proficiency extends beyond rendering to incorporate various facets of real-time computer graphics, particularly focusing on solutions and tools applicable to video game technology, namely real-time interactive applications. I possess skill in utilizing OpenGL, WebGL, and Vulkan APIs, with professional experience in core graphic engine development utilizing WebGL frameworks such as ThreeJS and BabylonJS. Moreover, I am versed in game engine and renderer architectures, having personally implemented several renderers based on each of the aforementioned APIs.

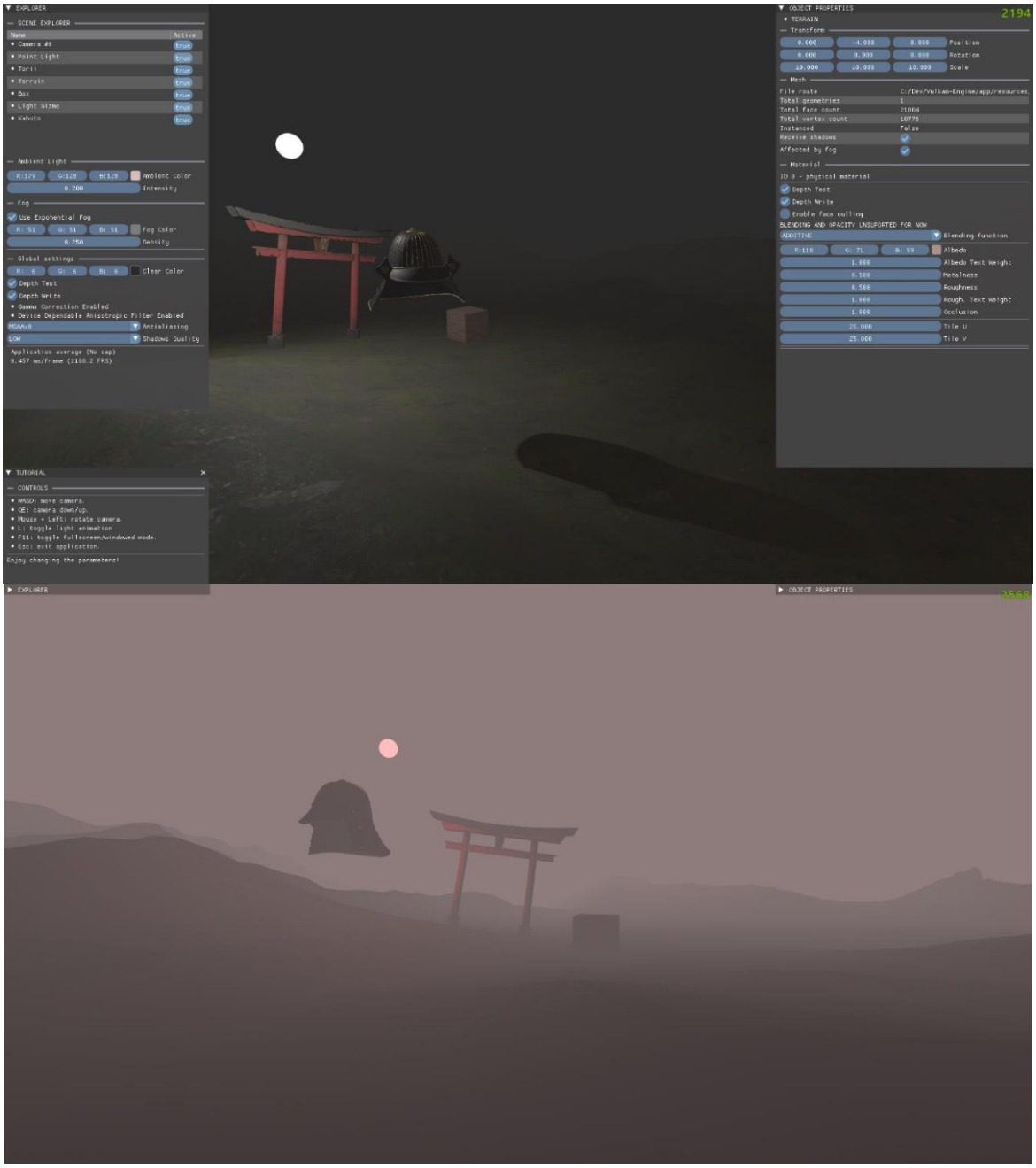
Furthermore, I have expertise in GPGPU --CUDA and graphic APIs compute stages-- and Machine Learning, particularly in neural radiance fields. One of my strengths is my knowledge of cutting-edge graphics research, which I have gained through my work in companies where advancing graphical capabilities is a primary goal. This experience has equipped me to research and implement rendering techniques outlined in white papers. While I am well-versed in raster real-time computer graphics, I also have a decent understanding, although more limited, of ray-traced and offline graphic techniques such as path-tracing, having personally completed the Wenzel Jacob's Nori path tracer course in my university master. Lastly, my expertise also extends to the fields of physics simulation, being familiar with mass spring implicit solving techniques and eulerian liquid specifications.

As a hobbyist artist, I enjoy implementing and refining computer graphics applications with a keen focus on the end-user experience. I try to simplify the interface of these complex tools while retaining their power and functionality, recognizing this as one of the most challenging and important aspects of the field. I find myself drawn to communicating with both design and technical teams, as I bridge the gap between user needs and technical capabilities.

I believe this sensitivity to user experience makes me well-suited for roles within companies dedicated to developing visually interactive products for users, particularly within industries like videogames, which aligns closely with one of my primary career aspirations.

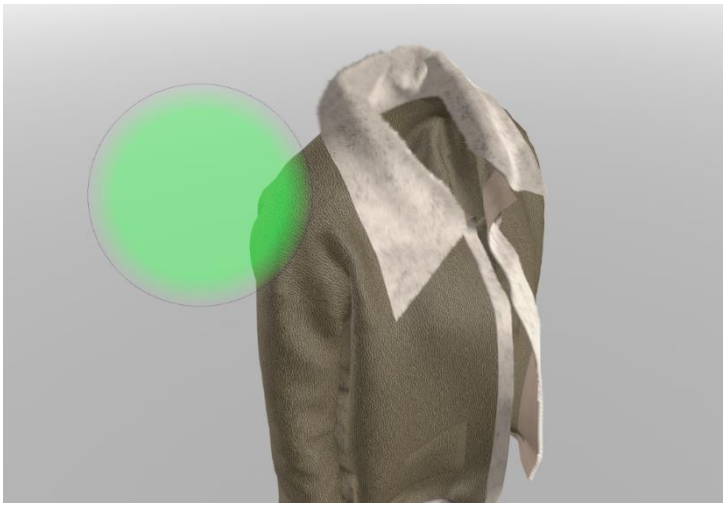
Vulkan based 3D Library

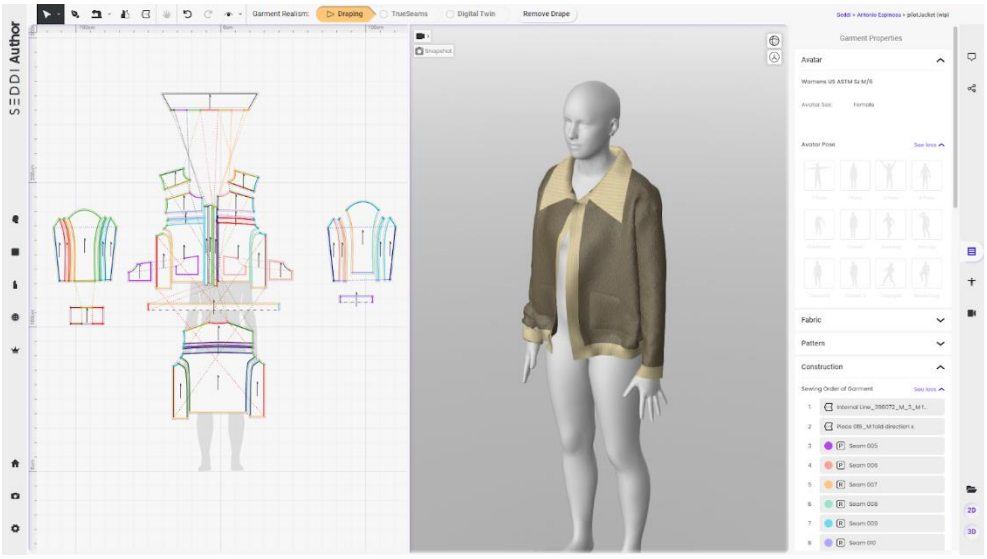
(link)

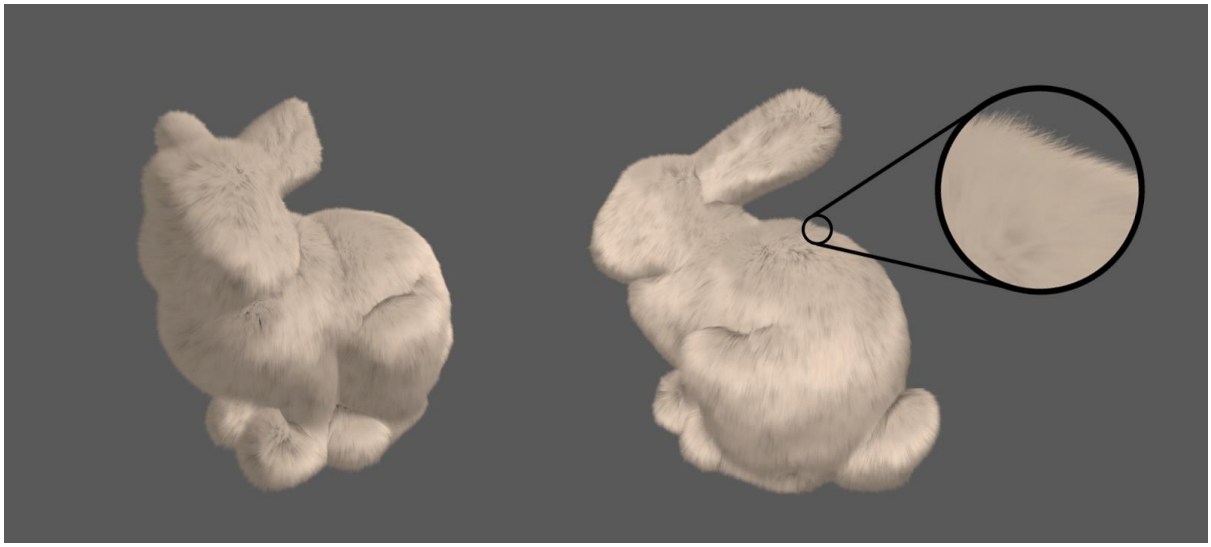


Cook-Torrance PBR, Phong and other types of material abstractions. Vulkan main objects and functionality abstraction. Multipass and custom pass. On the fly shader compiling. Simple to use interface. Concurrent resource loading. Working on forward plus. By implementing this library, I have learned the basics for working with the Vulkan API.

ThreeJS fur edition web application (Author SEDDI)







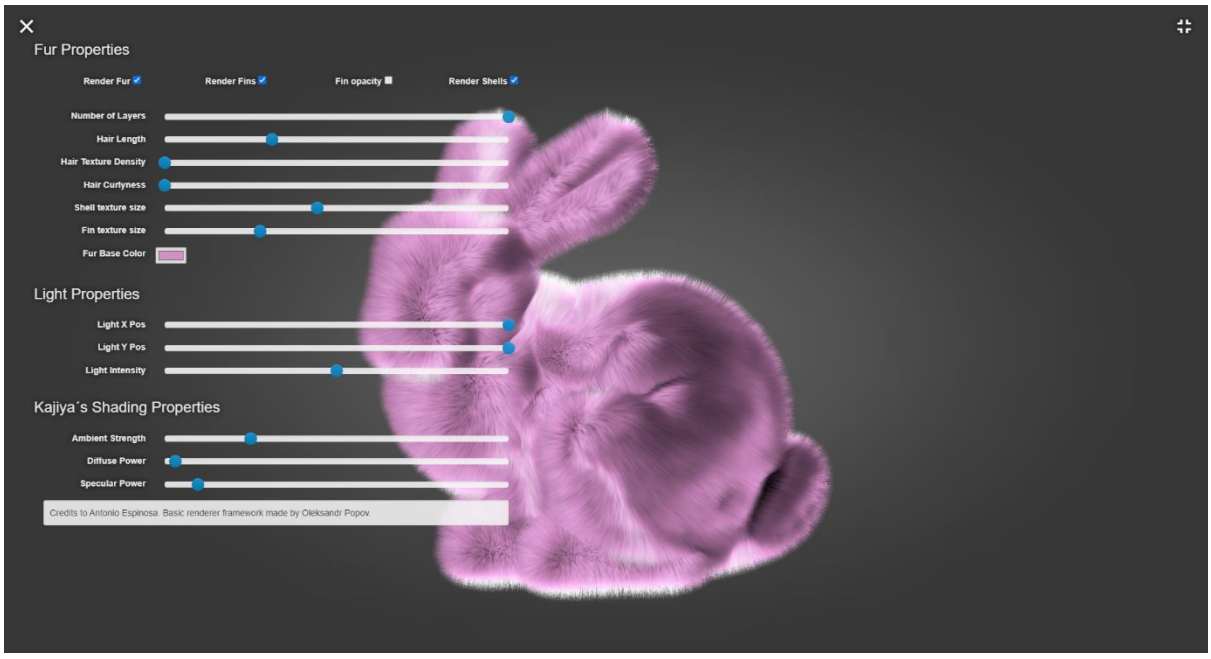
One of my latests contributions at SEDDI was integrating my developed fur rendering pipeline into an existing human avatar and fabric visualizer, directly linked to the commercial clothing editing application Author. The aim of this milestone was to equip the visualizer with a representation of furry fabric materials, enabling their addition to the avatar's clothing in Author. Finally, it's worth noting that, to achieve this goal, I had to modify the source code of the open-source commercial graphics engine ThreeJS, adding a WebGL functionality called Transform Feedback.

To represent hair fibers, I decided to implement the shells technique, proposed by Lengyel et al. For hair lighting, he used an enhanced method by Kayija and Kay, whose main contributions stem from Scheuerman's work. These techniques ensure high performance on any device, including low-end mobile devices, due to their low computational cost.

Because it is a private rep, I cannot provide you with a link.

WebGL fur edition web application

(link)



For my first task at SEDDI I managed to develop a graphic pipeline for rendering short hair/fur that is easily integrable into any web application regardless of the graphics engine used, as well as fully customizable. As a demonstration, I implemented a series of applications, each using a different level of abstraction of the WebGL graphics API.

These applications are primarily object viewers, allowing users to load any desired mesh, add short hair to it, and edit it in real-time. Among the developed editing functionalities, the brushing feature stands out. In essence, these applications are primarily designed to represent any surface susceptible to being furry, such as fabric or an animal.

WebGL hair card web renderer

(link)

In these demos, you can observe the results of transparency using the Weighted Blended method, which is order-independent and greatly improves refresh rate. The advantage of this method is that it eliminates the need for primitive sorting, which can be very costly for hair rendering. Additionally, you can observe the self-occlusion shadows using the Deep Opacity Maps technique in the second image, which is subject to improvement in terms of shadow map filtering and precision.

OpenGL strand based hair renderer

My last work, my idea is to bring all the already implemented hair rendering techniques to a standalone demo, using more powerful APIs than WebGL and thus create a more advanced hair rendering pipeline through polyline tessellation, to train a NN and use it as an optimization phase. This has been my current work on board the MSLab Research Group the King Juan Carlos University. NeuralHaircut project functionality is integrated in this application, in order to showcased real hair from real persons. Scheuerman model is being updated to a real time version of Marchnerr, based on Tafuri's work.

OpenGL based 3D Library

(link)

This library wraps all basic functionality needed for a renderer to be user-friendly and look good. It has all things my Vulkan library has plus multi omnidirectional shadow maps using geometry shaders, stencil buffer usage for object highlighting and a sprite pipeline.

Unity Cloth Physics Tool

(*link*)

As a hobby, I also like to implement modules and tools for game engines, one of my most successful is this particular one. It uses a mass-spring approach with Euler implicit collisions.

Videogame (Gears of Hell)

(*link*)

Videogame (Rocket Yan)

(*link*)

Lastly, although it is not strictly connected to rendering, I would also like to point out two of the videogames I have designed and implemented I feel most proud of, in order to showcase my skills and sensibility for the videogame industry overall.

I encourage you to check my github for more info about my projects!

All works showcased in this document are authored by Antonio Espinosa García.