

Chanhaeng Lee

Seoul · South Korea · lch32111@gmail.com / [github](#) / [blog](#)

BACKGROUND

I am a game engine programmer interested in computer graphics. I've started my programming career to be a graphics programmer because I've wanted to make my own world.

WORK EXPERIENCE

UBISOFT LA FORGE

2022.11–2024.08

Generalist Programmer

I worked in the internship program of Ubisoft La Forge Montreal as a R&D generalist programmer. I researched neural cloth synthesis to apply realistic cloth deformation in real-time applications such as games. This research was done together with my master's degree program. The research paper, Real-Time Neural Cloth Deformation using a Compact Latent Space and a Latent Vector Predictor, is accepted at ECCV 2024 CV2 Workshop. Refer to my thesis about the paper [here](#) before ECCV uploads the workshop paper.

LG AI RESEARCH

2022.05–2022.08

Freelancer Software Engineer

I implemented an application that a user can view chemical files, edit rendering results, and get labeled data for the results in order to generate a number of training datasets for the company's machine learning project. The program is based on one famous chemical rendering library. The program targets cross-platforms including Windows and Ubuntu. The project is built with the cmake build system and C++.

- Implemented a command line interface program for Windows and Ubuntu that can generate loads of training datasets in parallel.
- Implemented a GUI program for Windows that a user can edit rendering results and get labeled data with custom rendering options. The program also support multi-threaded processing of chemical files and multiple window rendering. DirectX 11 is used for rendering.
- Analyzed and customized one of the famous chemical rendering libraries to generate training datasets

COM2US

2019.03–2021.06

Game Engine Programmer

I implemented a renderer in a game engine and other systems for the engine. This project targeted cross-platforms including Windows, Mac, Android, and iOS. The project was built with the cmake build system and C++ 11. I mainly handled not only various aspects related to rendering, but also useful facilities for the engine to make it work robustly.

- Implemented **an integrated Graphics API** using OpenGL ES, Vulkan, Metal.
- Implemented **a material/shader system** which enables the engine to do the shader permutation and a user to do **shader programming** on the engine. The user shader programming feature is inspired by Godot engine. This system mainly uses glslang for shader code conversion.
- Implemented **a forward shading renderer**, which renders game scenes and editor viewports. The renderer also supports **multiple window rendering** for the editor. The renderer architecture refers to the [mesh drawing pipeline](#) from Unreal Engine 4 (UE4)
- Co-planned **an asset system** of the game engine, and implemented the core part of the asset system.
- Co-planned **a serialization system** of game engine files.
- Implemented **an automation test framework** inspired by UE4 by using Socket API to communicate test requests/results between processes on a local computer.

PROJECTS

CHLIB ([BLOG](#))

I start to develop my second personal library. I have been developing this project using C99, unity build, 4coder editor and raddebugger. I implemented my STL with allocators, rendering api with Vulkan 1.3, PNG/BMP decoder, Font Decoder and Rasterizer, Immediate mode GUI, one draw call GUI rendering, Obj File Parser, and etc.

MESHOPTIMIZATION ([GITHUB](#))

2023

An old Mesh Simplification algorithm by Hoppe et al. (SIGGRAPH 1993). I studied linear least squares problem in terms of mesh simplification problem and practiced mesh topological operations such as edge collapse, edge split, and edge swap with HalfEdge data structure of a mesh.

MARCHINGCUBESDF ([GITHUB](#))

2022

I studied how to calculate SDF values from a mesh and render it with marching cube algorithm by OpenGL API.

PERSONAL GAME ENGINE ([GITHUB](#), [YOUTUBE](#))

2018–2021

This is a toy game engine project, where I studied computer graphics and physics. I implemented **a range of graphics features** such as deferred shading, lighting, shadow cast, post-processing (High Dynamic Range, Tone Mapping, Bloom, Screen Space Ambient Occlusion), terrain rendering with Perlin noise, and so on. As for the physics part, I implemented **a simple physics engine** which detects collisions and resolves contacts. I studied **Rigidbody Dynamics** and **Sequential Impulse Constraint Solver** for the physics engine. After this work, I implemented **a simple multi-threaded ray tracer** with a bounding volume hierarchy.

GJK COLLISION DETECTION ([GITHUB](#))

2019

I studied and implemented a frequent-used algorithm (Gilbert-Johnson-Keerthi distance algorithm, GJK) in collision detection with a Game Developer Conference(GDC) presentation by Erin Catto, Box2D developer.

EDUCATION

MASTER OF COMPUTER SCIENCE

2022.09– 2024.09

Concordia University, Montreal, Canada
GPA: 4.23 / 4.3

I researched neural cloth synthesis for my master's degree of computer science at Concordia University, supervised by [Dr. Tiberiu Popa](#). Our paper, Real-Time Neural Cloth Deformation using a Compact Latent Space and a Latent Vector Predictor, is accepted at ECCV 2024 CV2 Workshop. I worked as a TA for COMP 352 Data Structures and Algorithms in 2022 Fall semester.

BACHELOR OF BUSINESS ADMINISTRATION

2012–2018

Chonnam National University, Gwangju, South Korea
GPA : 4.0 / 4.5

I majored in business administration, and minored in computer science from 2016. I studied computer graphics at Northumbria University in the UK as an exchange student for one semester in 2017. I served in the Korean military from 2013 to 2015.

SKILLS

Programming

Language : C++, C, GLSL, HLSL, Python
Graphics API : OpenGL, Vulkan, Metal, DirectX
Build Tools : CMake, Batch
IDE : 4coder, Visual Studio, XCode.

Communication

Korean, English