

Aviral Goel

aviralgo@usc.edu | www.aviralgoel.xyz | linkedin.com/in/goelaviral | github.com/aviralgoel

EDUCATION

University of Southern California

Master's in Computer Science, Game Development

Los Angeles, CA

Aug. 2022 – May 2024

Birla Institute of Technology

Bachelor's of Engineering, Computer Science GPA: 3.68

Ranchi, India

Aug. 2016 – May 2020

TECHNICAL SKILLS

Languages: C++, C, C#, Python, Java, Solidity, JavaScript, Lua, Golang, Assembly Language

Developer Tools: Unity, Unreal, Maya, Visual Studio, RenderDoc, Git, CLion, Perforce, Ryder

Libraries: OpenGL, Vulkan, GLFW, SDL, GLAD, ImGui, Mpi4py, Sockets, Photon Networking, upng

PROJECTS

Mini Vulkan Renderer | *C++*, *Vulkan* | [GitHub](#)

Jan 2022 – Feb 2022

- Setting up Vulkan development environment in sync with validation layers, GLFW, GLM, and GLSL shader modules, and their compiler in Visual Studio complete with git source control.
- Familiarity with Vulkan pipeline creation, Vulkan device, command buffers, swap chains, and descriptor sets
- Implemented texture mapping, mipmapping, and multi-sampling anti-aliasing in Vulkan API for a 3D scene.

Avi(ral's) Graphics Library | *C*, *Graphics Programming* | [GitHub](#)

Aug 2022 – Jan 2023

- Built a graphics library capable of loading and rendering 45K+ triangles at 60FPS using just C and SDL library.
- Accomplished linear transformation, backface culling, perspective projection, z-buffer algorithm, directional lighting, Barycentric coordinates for interpolation, flat shading, and Gouraud for rasterization of 3D models.
- Coded third-person camera and PNG texture mapping with perspective correctness for 45K+ triangle faces.
- Performed view frustum culling, loading multiple meshes per scene, and managed version control using git CLI.

Try Again, Team Game Project | *Unity*, *C#* | [Youtube](#)

Aug 2022 – Jan 2023

- Extended Unity's tag system to support more than one tag, such as breakable, throw-able, pick-able, etc., for a single gameObject, allowing the player to interact with objects in multiple ways.
- Worked with Cinemachine to have third-person and side-view cameras switch back and forth based on map locations. Also, allowing the player to adjust camera orientation.
- Wrote a script for altering camera rotation, thereby overriding the restriction set by the Cinemachine Transposer.

OpenGL Batch Rendering | *C++*, *OpenGL*, *GLSL* | [GitHub](#)

Jul 2022 – Sep 2022

- Learnt the basics of OpenGL in C++, including vertex buffer, index buffer, vertex array objects, and textures.
- Implemented fragment and vertex shader with the ability to batch render quads having multiple textures or colors.
- Abstracted separate classes to maintain buffer layouts, vertex array objects, textures, and shaders.
- Familiarized with ImGui, STBI, GLAD, and GLFW helper libraries in sync with OpenGL to help render on screen

Supreme Physics Engine | *C++*, *Git* | [GitHub](#)

Jan 2022 – Jul 2022

- Created a 2D physics simulation engine capable of performing particle physics and rendering on screen.
- Developed skills in creating and managing Vulkan resources such as buffers, images, and pipelines, including vertex and index buffers, shader development, and texture mapping.

EXPERIENCE

Unity VR Engineering Intern

Jun 2019 – Dec 2019

Sar Centre for Virtual Reality

Muscat, Oman

- Developed a virtual reality training simulation to train heavy crane operators at shipping ports.
- Worked with Oculus SDK for Unity to build an application for Oculus Quest with six degrees of freedom.
- Engineered a virtual joystick to be grabbed and moved in the simulation just as a real joystick would.

LEADERSHIP & INVOLVEMENT

Contributor

Aug 2022 – Present

ScratchAPixel.com

Los Angeles, CA

- Edited articles on ScratchAPixel website for graphics programming with 120K visitor per month to the website.