HOSEOB JEONG

SOFTWARE ENGINEER

do8437@gmail.com https://flexseob.github.io 206-571-4311 Redmond, WA hoseob-jeong flexseob

EDUCATION

Sept. 2021 to Apr. 2023 — Digipen Institute of Technology
Bachelor of Science in Computer Science in Real-Time Interactive Simulation

SKILLS

COMPUTER LANGUAGES C++, C, JavaScript, Lua, Python, GLSL, C#
OTHERS(GRAPHICS API, TOOLS) OpenGL, Vulkan, UnrealEngine, Unity, Git, ubuntu, VisualStudio, VisualStudioCode, SVN, Xcode, Maya, GameDevelopment, RenderDoc, CUDA-CUFFT, MacOS

PROJECTS

Graphics Programmer, Real-Time Ocean Rendering(Personal Project) Feb. 2023 to Mar. 2023
OpenGL Ocean Rendering(C++,CUDA,GLSL,GLFW,Imgui) reusd OpenGL Graphics Engine / Video : https://youtu.be/PtyeZioTXWY

- Implemented real-Time ocean rendering using the Fast Fourier Transform(FFT) method in OpenGL with CUDA-cufft library.
- For making height map, worked with the cufft library to efficiently perform FFT calculations on the GPU.
- Produced realistic ocean water simulation with reflection, and refraction effects.
- The project was based on the following research papers and video
  - Simulating Ocean Water – Jerry Tessendorf(2004)
  - RealTime GPGPU FFT Ocean Water Simulation – Fynn-Jorin Flugge(2017)

Graphics Programmer, Image Processing Demo(Personal Project) Sept. 2022 to Dec. 2022
OpenGL Imaging processing(C++,GLSL,GLFW,Imgui, compute shader) reused OpenGL Graphics Engine / Video : https://youtu.be/ZxpjpFKcgF0

- For fast computing image processing, Implemented compute shaders for processing method.
- Created Bi-linear interpolation and nearest neighbor method for imaging re-scaling.
- Implemented Imaging operation(addition, product, subtraction, negative, Log Transform, Gamma Transform)
- Implemented 2Pass 4,8 connected-CCL(Connected Component labeling) algorithm
- Added Histogram Equalization and histogram matching for adjust contrast adjustment.
- Created Gaussian Blur with N-kernel and the standard deviation, edge detection using the sobel operators.
- Added Unsharp-masking operation using gaussian blur.
- Implemented Discrete Fourier transform(DFT) and Inverse Fourier transform(IDFT)

Graphics Programmer, Ourborous Graphics Engine, 3 members May 2022 to Dec. 2022
Vulkan Graphics Engine(Vulkan api, C++, lua, Entity Component System) / Video : https://youtu.be/Z8Fgf065Ayk

- Implemented vulkan wrapper objects and class.
- Added Physics based Rendering(PBR) with texture mapping and multiple lights(Spot, Point, Dir light)
- Implemented shadow mapping with multiple lights using geometry shader
- Implemented deferred rendering and mipmapping generation in runtime
- Developed Shader include system(#include "shader.glsl" in glsl) for easier to make shader codes
- Created imGUI vulkan texture descriptor pool for drawing texture more friendly in imGUI.

Graphics Programmer, OpenGL Graphics Engine(Personal Project) Sept. 2021 to Apr. 2022
OpenGL Graphics Engine (C++,GLSL,GLFW,C,Imgui) / Video : https://youtu.be/oLizMwzHliM

- Implemented wrapper OpenGL API classes (buffer,shader, vertexarray, texture)
- Implemented manager class for resource handling (Object Manager, Mesh Manager, Light Manager, and Texture Manager) by using STL unordered map
- Created "obj" loader for drawing vertex normal rendering and face normal rendering
- Implemented Multiple Lights, Reflection, Refraction with Dynamic Cube mapping, and Deferred Shading.
- GUI supported to handle the objects, meshes, lights and shader uniform variables.

Gameplay Programmer, PinataPanic(GameProject), 13 members Aug. 2021 to Apr. 2022
Unreal Engine 4 (Gameplay programmer/Animation programmer) / Video : https://youtu.be/cOzKQyka0qo

- Implemented Pinata character movement, interactive objects, VFX and animation.
- Created Basic AI for chicken enemies, Delivered players more interesting.

Physics/Gameplay Programmer, Q(GameProject), 5 members Sept. 2020 to June 2021
2D platform Game Custom Engine (ECS) C++ (Physics/Gameplay Programmer) / Video : https://youtu.be/2qU1mYUu7Tw

- Implemented 2D physics, collision components(AABB, Circle, Ray) and event. Debug Collision Visualization System, and Offset system (for easily handle texture position), Built ray-casing algorithm for AI.
- For increasing game frame-rates, Upgrade spatial partitioning for collision with Quadtree data structure for collision optimization. Improve the O(n^2) to O(klogn)
- Implemented Player-movement state-machine.
- Designed and Built game weapons with Lua script.

AWARDS

Aug. 2021 — Digipen Institute of Technology · **DIT Merit Scholarship**
Academic scholarship (2021~2023)