FENIL SHINGALA

(+1)425-351-1074 | shingala.fenil@gmail.com | https://www.linkedin.com/in/fenilshingala/ | https://github.com/fenilshingala

SKILLS

APIs and Game Engine:	OpenGL, Vulkan, Win32, Unreal Engine 4 Virtual Reality
Languages:	C++, C, Lua, Java, Python, R, HTML, CSS, JavaScript, jQuery, PHP
IDE and Tools:	Microsoft Visual Studio, Git, Jenkins, RenderDoc, Eclipse, Anaconda Navigator

EDUCATION

MASTER OF SCIENCE: COMPUTER SCIENCE — April 2020

DigiPen Institute of Technology

BACHELOR OF ENGINEERING: COMPUTER ENGINEERING — July 2018

Gujarat Technological University

PROFESSIONAL EXPERIENCE

GRAPHICS PROGRAMMER — August 2020 – Current

The Forge Interactive, Inc.

- Introduced a scripting (Lua) layer in UI widgets which registers functions to avoid manual registration for every app
- Fixed and optimized window management functionalities for better window handling written in Win32 api
- Worked on performance analysis on Oculus Quest to find out cause of the heavy GPU times using tools provided in UE4 such as OVR metric tools, stat commands, shader complexity viewer, UE4 Insights and RenderDoc
- Applied fixes for performance optimization using tools in UE4 such as reducing landscape resolution, make pixel shader heavy materials rough, merge static meshes to reduce draw calls and simplifying material blueprints
- Performed analysis of visual artifacts on Oculus caused by high Fixed Foveated Rendering and low MSAA

GRAPHICS PROGRAMMER- INTERN — May 2019 – August 2019

CONFETTI Interactive, Inc.

- Worked on a shipped game to replace OpenGL renderer with cross-platform existing renderer (Vulkan and DX12)
- Added support for renderer framework (Vulkan API), file system and GUI on Android platform for client project
- Improved existing Entity Component System performance and made component creation dynamic in The-Forge

ACADEMIC PROJECTS

GRAPHICS FRAMEWORK (Phoenix) — September 2019 – Current

- Used Nvidia's Turing Extension for Vulkan Ray Tracing(VK_NV_ray_tracing) and recursion inside of the ray tracing shaders to implement real time reflections on a scene with reflective surfaces
- Extended and optimized my graphics framework to include features like:
  - Deferred Rendering
  - Physically Based Rendering (PBR) pipeline
  - Ambient Occlusion

3D GAME PROJECT — September 2019 – December 2019

Producer, Engine and Gameplay Programmer

- Led a team of 3 to make a game engine with ECS architecture and 3 game prototypes
- Implemented basic 3D physics which included collision detection, resolution, impulse and friction
- Designed one of the prototypes- Chameleon Run -to test our engine with 3D platformer game
- Wrote a stack-based scene manager to have smoother scene transitions

2.5D GAME PROJECT — January 2018 – March 2019

Engine and Tools Programmer

- Built an Entity Component System (ECS) in C++ for handling vast number of entities per frame
- Used Run Time Type Reflections (rttr library) to support deserialization, serialization from JSON and level editor
- Created Memory Manager with pools of memory to avoid runtime system calls for new memory allocation
- Worked in an agile development environment with SCRUM methodology to coordinate team of five people

CUSTOM ENGINE — September 2018 – November 2018

- Programmed a Component Based Game Engine in C++ with Data Driven Architecture for deserialization
- Designed vector, math and matrix libraries to support game physics of Axis Aligned Bounding Box and Circles
- Added textures, text renderer, sprite animations, power ups, sound, multiple levels, pause-continue-restart modes