RYAN DAVIS

(425) 681-0338
rdavis.dev@gmail.com
github.com/rdavisdev
rdavisdev.wixsite.com/helloplayer

Graphics Programmer

Languages – C++, C, C#, GLSL
Libraries – OpenGL4, STL, ImGUI, Spine C++ Runtime
Platforms – Unity3D, Windows7/8/10, Linux Ubuntu/Mint
Concepts – Deferred Rendering, Realtime Lighting and Shadow Mapping, Normal Mapped Textures, Object Oriented Development, Design Patterns, Algorithm Analysis, Multi-Thread Programming

EDUCATION

DigiPen Institute of Technology – Bachelor of Science in Real-time Interactive Simulation **Summer 2019 – Current**
Set to Graduate Summer 2022

University of Colorado – Bachelor of Science in Computer Science **Fall 2014 – Summer 2018**
Graduated 2018

SCHOOL PROJECTS

Graphics / Engine Programmer, C++ **Fall 2019 – Current**
Isles of Limbo – *2D Hack and Slash, Team of 11*
- Developed OpenGL rendering pipeline and interface.
- Overhauled particle system using instanced and deterministic particles. Up to 10,000 per second.
- Expanded ImGUI in-engine editor viewports using deferred rendering for debug drawing.
- Implemented Spine 2D C++ runtime library for animation.
- Designed asset archetype deserialization and runtime loading for ease of use.
- Consulted team in Entity Component System engine design patterns.

Graphics / Gameplay Programmer, C++ **Summer 2019**
Chromatic Split – *Grid-based Multitasking Puzzle Game, Team of 4*
- Developed render pipeline through Alpha Engine library.
- Refined game-feel through player/camera movement intricacies.
- Implemented object-oriented multi-typed tile design.
- Designed system for constructing levels from text file map layouts.
- Established color blending and color collision system.

Graphics / Engine Programmer, Java **Spring 2018**
Untitled Project – *2D survival shooter, Team of 4*
- Incorporated Android SDK's MVC library to render graphics and register input.
- Developed random level generation and enemy behavior.
- Built collision detection and resolution system.
- Authored Unified Modeling Language diagrams for engine design.

Graphics Programmer, C **Spring 2018**
Water Surface Quad Shader – *GLSL Project, Solo*
- Utilized framebuffers to render multiple angles of scene.
- Simulated Fresnel effect by combining refraction and reflection based on viewing angle.
- Replicated ripples of water using normal mapped texture and Phong lighting model.

Graphics Programmer, C **Summer 2017**
Etch – *3D Object Modeler, Solo*
- Implemented object construction using real-time mesh editing.
- Incorporated texture mapping for seamless surface detailing.
- Generated dynamic shadow maps to light created objects.

INDEPENDENT PROJECTS

Unity3D Games – *Oculus VR SDK, Solo* **Fall 2015 – Summer 2019**
- First Person hack and slash
- Tabletop-VR Tower Defense
- Anti-Gravity VR Climbing Experience