Ryan Davis

(425) 681-0338
rdavis.dev@gmail.com
linkedin.com/in/rdavisdev
rdavisdev.github.io

Programmer – Gameplay, Graphics, Engine

Languages – C++ | C | C# | GLSL | HLSL
APIs – OpenGL | ImGUI | Spine | Eigen | Assimp
Software – Unreal Engine 4 | Unity | Visual Studio | Maya | Git
Skills – Multi-Thread Programming | Performance Profiling | Algorithm Analysis | x86 Assembly Debugging
Capabilities – Path Tracing | Micro-Facet BRDF | Ray Marching | Image Based Lighting | Ambient Occlusion
Moment Shadow Mapping | Deferred Rendering | Bounding Volume Hierarchies

EDUCATION

Bachelor of Science in Real-Time Interactive Simulation **Graduating April 2022**
DigiPen Institute of Technology – Redmond, WA
Bachelor of Science in Computer Science **May 2018**
University of Colorado Boulder – Boulder, CO

ACADEMIC PROJECTS

Gameplay Programmer / Cinematics Designer [C++ / UE4] **April 2021**
Project Neon – *High Mobility FPS* *Team of 15*
- Scripted campaign and challenge level cinematics.
- Programmed dynamic script system to improve speed and quality of cinematic implementation.
- Repurposed scripting system to drive events during champaign gameplay.

Graphics Programmer / Engine Programmer [C# / C++ / GLSL] **July 2020**
Isles of Limbo –*Pseudo-3D Hack and Slash* *Team of 11*
- Developed and optimized engine's OpenGL graphics and visual effects pipeline.
- Overhauled particle system instancing deterministic particles allowing ~10,000 particles per draw.
- Built profiling toolset for finding bottlenecks and improving engine performance.
- Implemented Spine 2D C++ runtime library to run dynamic animations.
- Applied archetype deserialization and runtime loading to decrease asset load times.

Producer / Gameplay Programmer / Game Designer [C++] **July 2019**
Chromatic Split – *Multitask Puzzle Game* *Team of 4*
- Designed game mechanics and stages to engage and naturally teach players.
- Developed render pipeline for game's principal color mixing mechanic.
- Refined game-feel with satisfying player/camera movements.
- Implemented hierarchical tile class design for easy mechanic prototyping.
- Designed serialization system for constructing and designing levels from text files.

INDEPENDENT PROJECTS

Fog of War w/Line of Sight [C++ / UE4] **June 2021**
- Scene gets mapped to a grid, sampling tile heights and slopes from geometry.
- Accurately identifies grid locations obscured from a unit's sight. Darkens scene to reflect.
- Multi-threaded for 8 core parallel processing and delayed result gathering.
- Can update a radius of 20 (~2.5k tiles) per frame at 120fps, but usually runs once on new tile entry.
- Cost scales linearly with minimal overhead, allowing massive visibility range.

Zipline Locomotion [UE4] **August 2020**
- Swing physics react to player and rope movements realistically.
- Smart attach system allows attaching to and between ropes at any point along length.
- Letting go or jumping detaches, but momentum can launch the player for distant jumps.

Volumetric Clouds [UE4 / HLSL] **June 2020**
- Produces real-time traversable cloudscapes at 60fps.
- Ray-marching shader samples generated 3d-texture to accumulate cloud density and shadow.
- Dozens of settings allow for wide range of weather effects.

Slotted Combat Crowd AI [Unity / C#] **April 2019**
- 100+ Enemies surround player and wait their turn to move closer to an attack slot.
- Occupancy offload system assures even crowd distribution and no slot starvation.
- Crowd and slot positions/links react dynamically to scene geometry.