Spencer Melnick

Software Engineer

1016 Hemlock Lane

Enola PA 17025

Email: smelnick97@gmail.com

Portfolio: [spencermelnick.net](http://spencermelnick.net)

Github: [spencer-melnick](https://github.com/spencer-melnick)

## Professional Experience

[Fortnite Battle Royale *(Epic Games, Inc.)*](https://www.epicgames.com/fortnite)

**Gameplay Programming Intern**

2020-2021

* Developed new features to help ship several high profile cosmetics, including integrations with iconic third-party IP
* Collaborated with international teams specializing in a variety of disciplines such as art, design, quality assurance, and production
* Assessed, tested, and fixed numerous high-impact bugs in a breadth of gameplay systems
* Improved and created editor tools to facilitate faster development, based on designer specifications
* Learned the ins-and-outs of game development on one of the most well-known and fastest evolving games in the world

## Skills

* **Gameplay Programming**: 3D mathematics, object hierarchies, memory management, collision, physics, asset management, networking, multithreading, data structures, UI
* ***Graphics Programming:*** Volumetric raymarching, 3D noise generation, Blinn-Phong shading, normal mapping, compute shader programming, post processing, general graphics pipeline, realtime fast-fourier transformations
* ***Programming Languages:***
	+ **Advanced:** C++, C, HLSL, GLSL
	+ **Proficient:** Python, Java, Javascript, C#, HTML, CSS
* ***Development Tools:***Visual Studio, Perforce Helix Core, Jira, Git, CMake
* ***Game Engines:***Unreal Engine 4, Unity 3D, Godot

## Personal Projects

[**GPU Ocean Wave Simulation (Unreal Engine)**](https://github.com/spencer-melnick/VoidRoom/blob/OceanSim/README.md)2020

**Solo Developer**

* Researched academic papers on various techniques for computing wave displacement data
* Researched different algorithms for fast-fourier transformations
* Developed proof-of-concept project in Python before porting to engine and compute shader code
* Analyzed complex engine code to utilize cutting-edge/undocumented Unreal Engine features
* Developed custom method for precomputing butterfly operations, while minimizing memory cost
* Parallelized algorithm to take full advantage of GPU processing power

[**Volumetric Cloud Renderer (Unity 3D)**](https://github.com/spencer-melnick/Chalice)2019

**Solo Developer**

* Studied technical presentations by developers to understand emerging rendering techniques
* Developed external plugin for 3D Perlin and Worley noise generation to improve iteration time
* Implemented volumetric raymarcher as a material shader in HLSL
* Created custom depth blend function to enable early exit on half/quarter resolution rendering

[**Global Game Jam 2020 Entry - Meritocracy Train (Unity 3D)**](https://github.com/BuiltInParris/The-Meritocracy-Train)2020

**Gameplay Programmer**

* Worked with artists, musicians, designers, and programmers to determine minimal viable product
* Programmed main input system with keyboard and multiple gamepad support
* Developed platforming character dynamics with finely tuned collision resolution, acceleration, friction, variable air control, and jump extension to ensure core controls were fun and responsive
* Trained new team members on version control principles and resolved catastrophic merge failures
* Programmed state machines to drive character animation based on gameplay data

[**Global Game Jam 2019 Entry - Hearth (Godot Engine)**](https://github.com/spencer-melnick/hearth) 2019

**Lead Programmer**

* Programmed main character controls and core heat and health systems
* Programmed animations, particle effects, lighting effects, and sound effects
* Trained other programmers on how to develop games with the Godot Engine

[**Temple Robotics Frontend Mission Control Software**](https://gitlab.com/templerobotics1718/RUIN)2018

**Lead Programmer**

* Programmed simple OpenGL renderer for 3D visualization of telemetry data
* Created basic Blinn-Phong lighting system in GLSL
* Created custom .OBJ model importer
* Developed protocol for control and telemetry data with limited bandwidth on top of TCP/UDP
* Utilized multithreading to ensure other subsystems would not interfere with robot control
* Created cross-platform build process using CMake

## Education

**Temple University** 2015-2020

*BSE in Electrical Engineering (Computer Engineering Concentration)*