Spencer Melnick

Software Engineer

1016 Hemlock Lane

Enola PA 17025

Email: [smelnick97@gmail.com](mailto:smelnick97@gmail.com)

Portfolio: [spencermelnick.net](https://www.spencermelnick.net)

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

Skills

* ***Graphics Programming:*** Volumetric raymarching, 3D noise generation, Blinn-Phong shading, normal mapping, compute shader programming, post processing, general graphics pipeline, realtime fast-fourier transformations
* ***Mathematics:***Linear algebra, vector mathematics, signal processing, collision detection
* ***Programming Languages:***
  + **Advanced:** C++, C, HLSL, GLSL
  + **Proficient:** Python, Java, Javascript, C#, HTML, CSS
* ***Development Tools:***Visual Studio, GCC, GDB, CMake, Git
* ***APIs and Libraries:***OpenGL, Vulkan, SDL, GLEW, STL
* ***Game Engines:***Unreal Engine 4, Unity 3D, Godot
* ***General Programming:*** Data structures (linked lists, queues, stacks, object pools), multithreaded programming (Mutexes, condition variables, atomics, producer/consumer queues), network programming (replication, remote procedure calls, transport layer programming), object-oriented design

Projects

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

**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

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

**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

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

**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

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

**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

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

**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

* 2015 - 2020 *BSE in Electrical Engineering (Computer Engineering Concentration)*

**Temple University**

Professional Experience

* 2018 - 2019 *Undergraduate Research Assistant (C++ Programming)*

**Temple University Computer Science Department**

* Improved performance on algorithms by multithreading existing code and utilizing sparse data structures where appropriate