

Henry Orsagh

Portfolio:

<https://horsagh.github.io/portfolio/software-development/>

Phone: (585) 730-2838

Email: henry.orsagh@gmail.com

EXPERIENCE

Game Developer, RIT Cyber Games

May 2023 - May 2024

Grants from the NSA and ACI. Worked with students and faculty to develop resilience games & simulations in Unity.

Software Developer, RIT Software Design and Productivity Laboratory

November 2022 - May 2023

Grant from the DHS. Worked with a team to develop multiple tools related to software supply chain security. Focused on front-end web development and data analysis tools.

Event Staff, Rochester Foam Dart League

September 2022 - January 2023

Facilitated the deployment and design of game modes and rules.

EDUCATION

Rochester Institute of Technology, Rochester, NY, 14623

July 2020 - May 2024

Game Design and Development BS

- Magna Cum Laude
- Dean's List
- Creative Writing Minor
- 3.7 GPA

SKILLS

- **Programming Languages:** C#, C++, Java, Python, JS/TS
- **Team:** Agile/Scrum, Git, Trello, Google Suite
- **Web:** HTML/CSS, Angular, React
- **Design:** Photoshop, Axure, Figma

PROJECTS

HACKBOX

May 2023 - May 2024

Professional team project (Cybersecurity mobile game)

- Created in Unity
- Designed minigame framework for easy expandability
- Programmed entertaining minigames and reusable components
- taught important cybersecurity concepts
- Utilized Google Play Games Services for cloud save data
- Handled builds and deployment

Trials of Persephone

March 2023 - Present

Personal project (First Person Spellslinger)

- Created in Unreal Engine 5
- Programmed Enemy AI
- Created an engaging physics-based magic system
- Utilized vector math extensively
- Collaborated with 3D artist

Croak and Dagger

August 2022 - May 2022

Academic team project (Narrative driven Hack n Slash)

- Created in Unity
- Debugged combat system and movement
- Collaborated with team of 19
- Managed communication between programmers and artists

Clone!

April 2022 - May 2022

Academic solo project (Web puzzle game)

- Created with PixiJS
- Wrote collision detection and line trace algorithms
- Programmed recursive "cloning" mechanic