Andrew Hansen

(757) 355-2871 | Vancouver, WA
andrewhansen.dev@gmail.com | https://dtc-wsuv.org/ahansen20/portfolio/
https://www.linkedin.com/in/andrewhansen-dev/ | https://github.com/axolotliterature

Technical Skills
- Unreal Engine 5
- Unity Engine
- Phaser.JS
- C++
- C#
- Javascript
- Visual Studio
- Wampserver64
- HTML5
- CSS3
- Command Line

Collaboration and Version Control
- Github
- Perforce
- Slack
- Basecamp

Education

Bachelor of Arts in Digital Technology and Culture, Graduating Spring 2023
Creative Media and Digital Culture
Minor: English
Washington State University Vancouver

Certifications

Game Studies and Design – *Washington State University Vancouver*
- Spring 2023

Projects

"Portal Reimagined VR" (In Progress)
[URL not yet available]
Lead Programmer, Game Developer, January-May 2023
A gamified VR experience built in Unreal Engine 5 that reimagines the 1986 hypertext game and novel *Portal* by Rob Swigart.
- Creating original source code in C++
- Utilizing Unreal Engine's Blueprint system to efficiently test new concepts
- Creating interactive levels from scratch
- Designing player-manipulable 3D logic puzzles
- Implementing player interactions and locomotion

"Generative Ship Shooter Game"
https://dtc-wsuv.org/ahansen20/final477/
Game Developer, May 2021
A miniature linear ship shooter game that is procedurally generated, built using Javascript in p5.js.
- Authored an infinitely generative game with Javascript and p5.js processing
- Applied industry standards for Object Oriented Programming models
- Programmed collision detection, projectile tracking, and a score counter
- Used zero external assets, instead generating each geometrical model procedurally

"Blogging Application Backend"
https://github.com/axolotliterature/blog_C
Programmer, June 2022
A text based blogging application built entirely in C that allows users to create, count, search, print, and delete blog entries.
- Programmed complete text based blogging program in C
- Utilized Dynamic memory allocation for creation and alteration of linked lists
- Implemented the creation, traversal, and removal of structure data types held in linked lists
- Applied successful garbage collection to avoid memory leaks