

## SKILLS

### Backend/Systems

- Unity's Jobs + DOTS
- Multithreaded delayed result gathering
- Large project management with assembly definitions

### Editor tooling

- Unity editor extensions
- ShaderGraph and VFX Graph
- Editor scripting and inspectors

### Specific applications

- Procedural mesh generation
- Behavior tree NPC AI
- Story driven narratives
- Optimized real time simulations

## EDUCATION

### BS Software Engineering

Milwaukee School of Engineering | 2017

## EXPERIENCE

**Seeb Defender** made with Unity3D [fraculation.com/blog/seeb-defender-project/](https://fraculation.com/blog/seeb-defender-project/)  
12/26/2020 – Current

- Farming simulation tower defense roguelite
- Roguelite leveling is breeding procedurally generated plants, with l-system extension
- Developed in-editor tooling to design genetics ([github.com/dsmiller95/Genetics](https://github.com/dsmiller95/Genetics))
- Simple quest and narrative system
- Skills: C#, Unity Jobs/DOTS, 3D modeling

**L-System language extension** made with Unity3D

[fraculation.com/blog/lindenmayer-implementation](https://fraculation.com/blog/lindenmayer-implementation)

12/26/2020 – Current

- Highly optimized procedural mesh generation language
- Extended the Unity editor to compile custom language files into mesh generators.
- Used Unity's multithreading and mesh editing tools to increase performance.
- Skills: C#, Unity Jobs, Unity library management, Unit and Performance testing

**AWS developer** @ Milwaukee tool

12/14/2020 – 06/18/2021

- Near real-time data processing architecture
- Skills: C#, Python, Kinesis, Lambda, ECS, SQL, dynamodb, terraform

**Spideyboi** made with Unity2D playable at [fraculation.com/Spideyboi/](https://fraculation.com/Spideyboi/)

07/31/2020 – 08/03/2020

- Player uses a step-based visual programming scheme to build instructions telling a Spider how to build its force-directed-graph “web.”

**Dimensional Rifter** made with Unity2D: [github.com/dsmiller95/Dimensional-rifter](https://github.com/dsmiller95/Dimensional-rifter)  
08/08/2020 – 12/25/2020

- 2D colony simulation game in which The Player designates build orders, and the Colonists gather resources and build the structures.
- Leverage Unity’s DOTS system to maintain high performance when updating many agents at the same time.
- Create a custom Behavior Tree system as the AI of colony members.
- Skills: C#, DOTS, Unity’s Jobs, 2D sprite design.

**Greebles Economy** made with Unity3D: [github.com/dsmiller95/greebles-economy](https://github.com/dsmiller95/greebles-economy)  
04/27/2020 – 07/31/2020

- 3D animations used to visualize interaction between agents in a hex grid.
- Agents will trade resources with each other, optimizing for max utility.
- Player designs trade routes to extract profit from the simulated economy.
- Skills: C#, Unit Testing
-