Yanmei Wang

Phone: (US+1) (734) 548-7207 | E-mail: wyanmei@umich.edu | Game Gellary | Github: Yanmeeei | LinkedIn: Yanmei

EDUCATION

University of Michigan, Ann Arbor, United States 04/2025
M.S. in Computer Science & Engineering, GPA: 3.9/4.0

University of Michigan, Ann Arbor, United States 04/2023
B.S. in Computer Science, GPA: 3.8/4.0

Shanghai Jiao Tong University, Shanghai, China 08/2023
BSE. in Electrical and Computer Engineering, GPA: 3.6/4.0

SKILLS

Languages: Python, C#, C++, HTML/CSS, Swift, GoLang, Javascript
Framework: XCTest, Numpy, Pandas, React.js, Django, Flask, SQLite, Alamofire, OpenAI API
Tools: Unity, Git, AWS, Jira, Confluence, XCode, VS Code, JetBrains IDEs
Other skills: Agile Development, Data Structures, Unit Testing, Debug, CI/CD, Relational databases, Web Design & System, Operating System, Software Development Cycle, Machine Learning, Delta Debug

GAME DEV PROJECTS

Web-First, Accessible Game Engine Development [Spec | Git Repo] Ann Arbor, MI | 01/2024 – Present
(Research directed by Professor Austin Yarger @ Arbor Interactive)

- Developing a web-based game engine for **RTS** + **tower defense** games.
- Implemented the in-game toast message system using **C# Godot** and **Eventbus library**, allowing in-game messaging and accelerating the development process.
- Created the generalizable tower and resource functionalities using **Godot** (**C#**), ensuring tower object animation and material adaptation.
- Adapting **AWS** servers to dynamically load resources into the game at runtime.

Soul of the Forest (**Wolverine Soft Studio**) [Steam Page] Ann Arbor, MI | 09/2023 – 04/2024

- Developed a dialogue-driven **Unity** game inspired by *Undertale*.
- Designed and Implemented multiple **cutscenes** using Unity built-in tools such as playable director.
- Programmed various **in-game interactions** between the player character and the environment.
- Created six level maps and applied layers of tilesets using **LDTK**; worked closely with the Art and TechAudio team.
- Utilized **Jira** for project management, and **Git** for code version control and cooperation.

BIO 452: Field Ecology of Snail-Fungus Interaction [Gamejolt | Portfolio] Ann Arbor, MI | 03/2023 – 05/2023

- Developed a two-player **RTS** + **tower defense** game ranked 6/30 on the UMich game showcase by player voting.
- Implemented several core mechanisms in **Unity** (**C#**): special ground blocks, the navigation and auto-attack features of little snail & mushroom RTS units, the overall damage-health system, etc.
- Led the art tasks and created in-game art assets using **ProCreate**, including menu, level design, sprites, CGs, etc.

WORK EXPERIENCE

Software Engineer Intern @ PreVeil, Inc. Boston, MA | 06/2024 - 08/2024

- Developed a prototype for mounting the online PreVeil Drive to **macOS** devices' file system using **GoLang** and **Swift**, enabling real-time management and synchronization of drive files across devices, and laying the groundwork for integration into end-user environments and future product deployment.
- Implemented and deployed 22 **RESTful API** endpoints using the **Swift Alamofire** library sending **HTTP** requests, ensuring efficient interactions with remote drives.
- Developed and deployed over 30 **unit tests** for the JSON decoder and the backend API endpoints using **XCTest**. Additionally, conducted API testing with **Postman**, ensuring code integrity.
- Iterated and improved **UI**/**UX** of PreVeil Desktop/Web App using **TypeScript, Javascript,** and **HTML**/**CSS**, improving performance and user experience, and contributing to a more intuitive interface for thousands of users.
- Engineered a diagnostic API endpoint for PreVeil Drive using **GoLang**, optimizing data pagination from the server backend, which reduced data flow overhead and improved response time by ~1.5x. Improved the corresponding diagnostic frontend webpage using **HTML** and **Javascript** to adapt to pagination.

Software Developer @ Arborsense, Inc. Ann Arbor, MI | 09/2022 - 02/2024

- Implemented a prototype using **Python Numpy**, **Pandas**, and **Scipy** libraries to analyze biological & environmental data, identifying over 72 types of patient events, powering the company's core product.
- Optimized data processing algorithm using a sliding window, **accelerating** computing overhead by ~1.6x.
- Generated chart reports using **Matplotlib**, enabling the engineering team to visually identify event periods on a timeline, accelerating the development and iteration of algorithms.
- Directed the **code version control** and **documentation** within the engineering team, resulting in a well-organized changelog over 78 iterations to ensure transparency and easy tracking of updates.