TUNG LE

(+01) 413-210-2127 • tungle@umass.edu • linkedin.com/in/tungvle

SUMMARY

Self-learn developer & artist enjoying Programming and Mathematical challenges. Guided by Project-oriented learning.
Portfolio: https://tungvietle.github.io/portfolio

EDUCATION

Bachelor of Science in Computer Science Graduating Spring 2027
University of Massachusetts Amherst
Coursework: *OOP (Data structure, Algorithm), Calculus II, Linear Algebra, *Game Programming (*lecture attend only).*
Activities: NASA Space Apps, CICSOFT, Recreational Math, ACM Machine Learning, HackUMass XI, Hackher413.

EXPERIENCE

Software Developer Present
BUILD UMass

- Achieved 43% code reduction resulting in faster API by implementing backend integration with database.
- Worked closely with 3000 lines in codebase, resolve team merge conflicts, translate client's objectives into product.
- Built and debugged Serialization System for saving. Optimized loading time by 9%. Analyzed file formats & common data types to design data structure for safely encoded/decoded from disk via JSON or Scriptable Object.

Graphic Designer May - June 2021
Autonomous Inc

- Created motion designs, banners, advertisements, identites using Photoshop & Illustrator for 20+ blog entries.
- Adapted to real-world operations between technical and graphical teams. Coordinated with team members.

PROJECTS

Blimp
Personal project published on Google Play. Currently on Beta (Open) Testing.

- Wrote scalable code using MVC pattern, achieving loose coupling, reducing component-dependency & code duplication in codebase. Minimized checks and draw calls using events system in Observer Pattern, faster 28 FPS compared to unnecessary property checks in game-loop.
- Implemented Singleton, State Machine, Factory patterns in architecture. Programmed state machines for key in-game systems to modularize over 4000 lines of code.

ECS Game Engine
Basic game engine as a project after finishing self-learned course COMP 4300 by Professor David Churchill.

- Prototyped 2D game engine from the bottom using C++ and SFML. Achieved comprehensive understanding of game engine internals, optimizing main loop structure and tick rates, implemented Entity-Component-System (ECS) architecture, data-oriented designs for scalable, reusable game development.
- Engineered C++ fundamental game engine elements, including scenes, asset loading, memory management, level editor, sprite animations, basic shaders, input handling, collision detection, event systems, windows, drag&drop systems, AI for entity behaviors, maintaining stable 60 FPS frame rate.

Noxious Night
Team of 6 - GMTK Game Jam 2023. Collaborated for 48 hours effectively among strangers.

- Managed effective version control using Github, optimized pull requests & code reviews to coordinate work between 3 programmers, reducing merge conflicts by 32%, avoiding premature optimization.
- Integrated easy-to-use & scalable path editor for other programmers. Implemented AI, movement, collision for Granny using custom ray casting logic, reducing visual detection errors by over 50%.

TECHNICAL SKILLS

Programming: C#, **C++**, **Python**, Java, HTML, CSS, JavaScript, React/Native, MERN Stack.
Tools: Unity, Git, Github, OpenGL, SFML, Photoshop, Illustrator, XD, Blender, Figma, Asesprite, Firebase, Sourcetree.
Courses/Cert: COMP 4300-Game Programming, Forage EA Software Engineering.