

# HELLO,

I love algorithms, puzzles, and complex systems. I want to be on a strong team with passionate, driven members. I have programmed with peers for over a decade and have been a professional developer since the start of 2016.

**- Jacob Gantt**

# RESUME

gantt42@gmail.com

(423) 269-5617

[github.com/JGantts](https://github.com/JGantts)

[linkedin.com/in/jgantts](https://linkedin.com/in/jgantts)

[JGantts.com](https://JGantts.com)

Jacob Gantt  
121 Niles Wheelock Road  
Jonesborough, TN

## EDUCATION

East Tennessee State  
University  
B.S. in Computing,  
Computer Science  
Concentration  
Dec 2016

## PROFILE

Experienced and self-motivated programmer and project owner. Both self-taught and educated. Loves learning. Wants to work on a passionate team.

## EXPERIENCE

Developer, Various Clients

Current, 6 Months (July 2022 - now), node.js, iOS

Smaller projects. Implemented a web server and improved a SwiftUI app.

System Administrator, TVA — 4 Months (March 2022 - July 2022), SQL

Founder/Developer, LemurWare LLC

3 Years (June 2018 - February 2022), Swift watchOS iOS

Envisioned and implemented apps for iOS/watchOS. Taught self how to work in environment for same from documentation and online tutorials.

Developer, Siemens, Johnson City, TN

2.5 Years (Jan 2016 - June 2018), C# .NET

Assigned to SCRUM-based team of approximately 8 developers responsible for maintaining and testing the .NET API of Siemens' flagship physical-automation product, TIA Portal. Coordinated the required features list with project owners and project clients, both on my team and in other offices.

Project Coordinator/Lead Developer, Storyteller

Ongoing part-time (May 2020 - current), ElectronJS

Wrote Storyteller, a suite of developer tools for the *Creatures* game, making developing mods much more fluid and seamless. Coordinates code changes from various volunteers. Developed server with RESTful API. Server hosted a MySQL database of in-game assets. API is under redevelopment.

Cashier/Gas n' Go Attendant, Food City — 7.5 Years

## SKILLS

Through Education and Personal Experience

Linux/Unix, CLI, VMware, Python, Java, C++, C, x86 assembly, Unity, Java-Script, PHP, ASP.NET, SQL, Electron, Node.js

1 to 2.5 Years' Professional Experience Each

Swift, watchOS, Xcode, C#, .NET, Visual Studio, NUnit, Automated Testing, WPF, TFS, Git/Github

# PORTFOLIO

## JGantts

### **Credits Maker — Designed and Developed**

App for iPhone. Makes scrolling credits for the end of a video. Has “Pro” version which removes certain restrictions. Generates video output using Core Animation.

Features include an optional sign-in, a small store of features for purchase, a handful of settings for each video ending generated, and a drag-n-drop credits name entry methodology.

**Technologies:** Swift, SwiftUI, iOS, Google Firebase

### **A NSW Games Site — Developed**

A game host site, similar to Steam. The difference is that this site caters to “adult” games. I implemented the site backend and coordinated with a worker 6 timezones ahead of me who implemented the frontend.

**Technologies:** node.js, javascript, pug templating, css

# Lemurware

## **Tiny Fish by LemurWare — Designed and Developed**

Fishing game for Apple Watch. Set on an original world. Includes different regions, boss fish, achievements, and high scores. Each region featuring art from a different artist.

Players cast their line using the Watch's digital crown (a rotating knob on the side of the watch) and reel in using the same. Difficulty increases as players progress through the region. Difficulty is controlled via several variables such as fish weight, fish strength, and tension sensitivity, and more.

Also features a binary weather system consisting of rainy and sunny. This state is controlled by a complex algorithm designed to make the weather unpredictable but consistent in ways that players would expect. Using ideas similar to song shuffling, the weather at any given time is entirely random, but the weather at the next time slot (each 64 seconds) is predictable.

**Technology:** Swift, watchOS

## **Calculator by LemurWare — Designed and Developed**

Basic calculator for Apple Watch. Features tan, sin, and cos, all in degrees or radians, X to power of Y, X to root Y, addition, subtraction, multiplications, division, and a tip calculator. All with large buttons.

**Technology:** Swift, watchOS

**Source code:** [github.com/JGantts/Calculator-by-LemurWare](https://github.com/JGantts/Calculator-by-LemurWare)

# Siemens

## **TIA Portal — Wrote Applications Supporting Development of API**

Siemens' flagship physical-automation product. An IDE for physical automation programs. Used by customers to write their physical-automation programs and deploy these programs to PLC's. These PLC's were used to automate physical tasks, such as controlling robots in factories, or monitoring safety lockouts on roller coasters.

The TIA Portal API was used by Siemens' customers to programmatically perform tasks on TIA Portal. For example, BMW used the API to perform automated testing of the programs that ran the robots in their factories. The API took the form a .NET assembly which then interacted with TIA Portal, securely exposing the API to any .NET environment and language, e.g. C#.

**Technology of API: C#, .Net**

## **Explorer — Maintained and Improved**

Internal tool used to browse, debug, and execute the TIA API.

Greatly improved Explorer's responsiveness, stability, and feature list. Added bookmarking feature which allowed users to save a batch of API actions and execute them later.

**Technology: C#, .Net, WPF**

## **Generator — Maintained and Improved**

Fuzz testing tool which generated random physical-automation programs used for automated testing of the logic for the deserialization and compilation of customers' physical-automation programs.

**Technology: C#, .Net**

## **Support for Legacy API — Implemented**

Support for legacy TIA API and automated testing framework for same. In practice this allowed customers to select which version of the API to use when running programatic tasks against TIA Portal.

**Technology: C#, .Net**

# Creatures Game

## Storyteller, Sorcerer, Cartographer — Designed and Developed

IDE for *Creatures*. Under development. Current beta features a working code editor and a working map editor. Coordinated contributions from various volunteers.

Storyteller is a launcher that also uses a RESTful API to pull down creature histories for display.

Sorcerer's Table is a code editor featuring syntax highlighting, simple error checking, and injection of code into the running game.

Cartographer's Table is a *Creatures*-specific 2D side-scrolling map editor featuring 20 fps animation, panning, and zooming.

**Technology:** node (Electron), HTML, CSS, JavaScript

**Source code:** [github.com/JGantts/Storyteller](https://github.com/JGantts/Storyteller)

## Albian Warp — Developed C# API

Reimplementation of game's old server.

Entirely rewrote C# code to communicate with the game engine.

**Technology:** CAOS, C#

**Source code:** [github.com/AlbianWarp](https://github.com/JGantts/AlbianWarp)

## Children of Capillata — Coder

A moderately-sized mod of the *Creatures* game. My first team project.

**Technology:** CAOS, PRAY (builder)

**Source code:** [github.com/JGantts/Children-of-Capillata](https://github.com/JGantts/Children-of-Capillata)