

MICHAEL CHERVENAK

Carteret, NJ 07008

[LinkedIn](#) | [GitHub](#) | [itch.io](#) | [Website](#)

ch3rv3@gmail.com

EMPLOYMENT

Tools/Systems Programmer **Small Loan Studio** **Sept 2024 – Current**

- Improved and maintaining a custom greyboxing tool plugin in Godot.
- Designing and building a serialization manager that loads and saves data and keeps track of persistent levels during gameplay.

Gameplay Programmer **Bitrate Games** **Jan 2024 – Aug 2025**

- Working on Roswell's Journey using Rider, Unreal Engine 5.4, Perforce, HacknPlan, Taiga, and Discord.
- Setup controller and Keyboard/mouse controls with custom character movement component for local multiplayer characters.
- Iterated with the team to find the best solution for handling AI characters and spearheaded the rework of the current AI to use State Trees instead of Behavior Trees.
- Created a world spline system with branching splines for all pawns to travel along in a level.

Game Developer **[P1] Virtual Civilization Initiative** **May 2023 – Jan 2024**

- Led the programming team as a servant leader, improved the onboarding and retention of new volunteers to [P1], simplified procedures with Trello and GitHub to eliminate bottlenecks in the programming team's workflow.
- Identified, documented, and fixed multiple bugs in the Sky Jellies prototype using Unity and Trello.
- Organized and improved all the programming team's documentation with Notice.
- Assisted in the creation of UML Diagrams for the existing systems of Sky Jellies with PlantUML and Miro.
- Improved and expanded the Sky Jellies prototype tutorial.
- Designed and implemented inventory system in the Sky Jellies production build.

QA Tester **DAQA & GBTN by Keywords Studios** **Oct 2014 – Current**

- Insured the functionality of systems for AAA games developed for PC, Xbox, PlayStation, mobile, and Stadia were functioning properly.
- Reported hundreds of bugs with detailed recreation steps.

EDUCATION

Newark, NJ **Rutgers University** **May 2021**

- B.A. in Computer Science with Minor in Mathematics, *Magna cum laude*

LANGUAGES AND TECHNOLOGIES

- C++, C, C#, Blueprints, GDScript, Java, Python, CMake, SDL 2/3
- Rider, Visual Studio, VS Code, Perforce, GitHub, Git Bash, Unreal Engine 5, Unity, Godot, custom engines
- HacknPlan, Trello, Miro, Taiga, PlantUML, Notice, Notion, Discord

TECHNICAL EXPERIENCE

- **Game Engine** (2025). An Initial game engine from scratch using C++, Cmake, Vulkan 1.4, and SDL 3. ([GitHub](#))
- **Shardrun** – [Portfolio](#) (2023). A multiplayer tag/infection game. Developed multiple abilities, shard item pickup, player ready logic for waiting rooms, and implemented various parts of the the UI. (*Unity, C#, PC*)
- **Jelly's Daydream** – [Portfolio](#) (2023). I developed a procedural level and "capture area" that is all randomized at runtime. Implemented the UI designer's assets and programmed the functionality. (*Unity C#, Android*)
- **Game Engines** (2021). Designed and implemented a 2D game engine ([GitHub](#)) ([Portfolio](#)) for an open world beat 'em up and a 3D game engine ([GitHub](#)) ([Portfolio](#)) for an open world RPG. Systems: entity, camera, UI, combat, animation, stats, random loot chests, quests, interaction, audio, and item pickup. Used Chipmunk for 2D physics. (*C, SDL 2, Vulkan*)

CERTIFICATES

Unreal Engine 5 C++ Developer: Learn C++ & Make Video Games

Jan 2023

- Learned how to start making games with Unreal Engine 5, focusing on Object-Oriented design and best practices.

Created six projects with increasing difficulty including the basics of the editor, when to use C++ and/or Blueprints, level design, gameplay systems, artificial intelligence, animation, UI, and more.

Electronic Arts Virtual Experience Program Participant

Nov 2022

- Participated in the open access Electronic Arts Virtual Experience Program with Forage.

Tasks – Write a feature proposal, create a game object class, improve an inventory system, character creators

Udemy: Beginning C++ Programmer – From Beginner to Beyond

July 2022

- Basic and advanced topics of modern C++, maintaining best practices and learning features up to C++20.