

# Sahil Chawla

[sahilchawla2312@gmail.com](mailto:sahilchawla2312@gmail.com) | [www.linkedin.com/in/sahil-r-chawla](https://www.linkedin.com/in/sahil-r-chawla) | <https://sahilchawla01.github.io/> | 6786704960

Atlanta, Georgia, United States

## EDUCATION

**Georgia Institute of Technology** | Atlanta, Georgia, United States

August 2024 – May 2026

Masters in Computer Science

Coursework: Procedural Content Generation, GPU Hardware & Software, Computer Animation, and Video Game Design

**MIT World Peace University** | Pune, India

July 2019 – June 2023

Bachelor of Technology in Computer Science & Engineering

CGPA: 9.44/10

## SKILLS

- **Programming Languages:** C++, C#, C, Typescript, Javascript, HTML, CSS, Python, YAML, GLSL, Java
- **Technologies:** Unreal Engine 4/5, Unity, Cocos Creator, Colyseus, GameAnalytics, React, Electron, NodeJs, GLFW, OpenGL, Visual Studio, Visual Studio Code, Git, Blender, Figma
- **Concepts:** Data Structures and Algorithms, Linear Algebra, Operating Systems, Memory Optimization, Object-Oriented Programming, 3D Maths, Rendering/Graphics Pipeline, Game Engine Architecture, Project Management

## WORK EXPERIENCE

**Bobble Head Studios**, Founder and Lead Game Developer | Pune, India  
(Unreal Engine 5, C++)

August 2023 - July 2024

- Solely led the programming of an unreleased project, alongside 2 designers; responsible for building the user interface, enemy AI, interaction system, UI event system, dynamic cinematics architecture, and player movement.
- Coded an event system to spawn temporary actors that trigger UI prompts, sound effects, volume changes, and damage events; removing redundancy, and streamlined the process to construct gameplay events.
- Exposed interactable actors' events to blueprints that fire upon and after interaction with the actor, enabling designer-friendly tools that allowed rapid iteration without code-level changes, reducing iteration time by **50%**.
- Designed a blackboard tree for a non-playable character (NPC); integrated idle and attack animations, player searching, and hitbox detection, creating an NPC that chases and attacks the player.
- Built the UI and the functionality for the main and in-game menus along with video, audio, and control settings.
- Implemented a save system for menu settings that would save data in real-time when a setting would change.

**Wega Labs**, Gameplay and UI Programmer | Pune, India

July 2022 - July 2023

(Cocos Creator Engine, Typescript, Unity, React, GameAnalytics, YAML, and GLSL)

- Collaborated with a team of over 10 co-workers, to help develop and launch the mobile video game "[Cricinshots](#)" (now World Cricket Premier League) currently with more than **100,000** users.
- Led the user interface's bug fixing and new feature integration tasks during the game's live-ops period; promptly fixed over 50 bugs reported by users.
- Simulated multiple mobile devices using Google Chrome's DevTools, and performed memory profiling to identify bottlenecks. Upon optimization, rendering time was decreased by **40%**.
- Iterated upon and polished the user interface, leading to a peak of over **11,000** daily active users.
- Implemented a campaign system that acted as a first-time user experience, facilitating a peak "day 1" retention of **25%**.
- Used GameAnalytics to record player behavior during gameplay; the data helped decrease the player onboarding time by **35%**.
- Supervised game developer interns, assigned tasks, and promptly solved any documentation or game engine questions.

**Wega Labs**, Game Developer Intern | Pune, India

March 2022 - July 2022

(Cocos Creator Engine, Colyseus, Typescript, NodeJs, and React)

- Created a cricket game prototype implementing interactable stadiums, map maneuverability, and stadium inspection animations, allowing the team to execute consecutive sprints on time and iterate upon new mechanics swiftly.
- Prototyped and built the UI system to display the round's information, and integrated Colyseus to visually reflect real-time changes.
- Implemented pre-downloading of assets and platform-specific texture compression, which helped reduce scene load times by **50%**.

## PROJECTS

**CANNON CLASH!** (C#, HLSL, Unity) | [Project Description Link](#)

- Led development of the game in a team of 5, spearheading sprint meetings, conducting playtests and programming a multitude of gameplay systems.
- Designed and developed AI behaviour including patrolling around the map, chasing, and engaging the player.

**PROTOX GAME ENGINE** (OpenGL, GLFW, ImGui, GLM) | [Project Description Link](#)

- Solely developing a game engine to recreate the basic elements of the game, "Minecraft".
- Used a Model-View-Project matrix to transform objects from their model view to the perspective view.
- Utilized Phong's lighting model to provide lighting from multiple sources to multiple objects in the scene.

**PROCEDURAL CONTENT GENERATION SERIES** (C#, Unity) | [Project Description Link](#)

- Developed a series of projects that created 3D content procedurally and programmatically
- These projects used the following techniques - Flood Fill Algorithm, Perlin Noise, Cellular Automata, Random Numbers, Implicit Surfaces, Marching Cube Algorithm, and Reynold's Flocking Principles