

Zhasheken de Guia

zhashekendeguia@gmail.com | zhashe.dev | github.com/zhasheken | linkedin.com/in/zhashe

SUMMARY

Computing Science student specializing in AI algorithms, game design, and user experience, with hands-on experience in UX design and designing gameplay loops and game features. Draws on professional development experience and having built several games to create player-centered experiences that are accessible, intuitive, and engaging. Collaborative team player with strong communication skills and experience working across technical disciplines.

EDUCATION

University of Alberta

Edmonton, AB

Bachelor of Science, Specialization in Computing Science

Expected December 2027

Dean's Honor Roll

Relevant Coursework: Data Structures, Database Systems, Information Retrieval, AI Algorithms, Software Engineering, UI/UX Design, Game Development, Game Theory, Game Engine Programming

WORK EXPERIENCE

Loretta Health

Berlin, Germany

Software Engineer

September 2025 - August 2026

- Led mobile application prototype through the full software development life cycle, from planning and requirements to deployment and maintenance, building 5+ core features in React Native with backend logic, data flows, and responsive iOS and Android interfaces
- Led and onboarded multiple interns within an agile development process, managing tasks and sprints in Jira
- Performed integration testing across the app's frontend, database, and API, alongside functional testing of core features, within a CI/CD workflow
- Gathered feedback from user trial groups, surveys, investors, and external partner organizations to shape product requirements and direction, coordinating delivery of external components to keep production on track
- Built secure web application prototype with authentication, deployed machine learning model API to production, and designed dashboard interfaces and website pages to enhance user experience

TECHNICAL SKILLS

Languages:	C#, C++, GDScript, Python, Java, JavaScript, C, SQL, HTML, CSS
Game Engines:	Unity, Godot
Game Development:	Game design, game engine development, gameplay systems, level and puzzle design, AI pathfinding, playtesting
Frameworks & Libraries:	React, React Native, Firebase, Firestore
Tools:	CMake, SFML, Jira, Expo Go, Android Studio

PROJECTS

2D Game Engine & Arcade Shooter | C++20, SFML, CMake

Object-oriented 2D game engine built from scratch, with a Galaga-style arcade shooter built on top of it

- Architected the engine's frame loop, game object hierarchy, backend-agnostic rendering interface, and publish/subscribe messaging system, managing memory with shared and weak pointers to keep ownership acyclic

- Built a two-phase collision system (bounding-box broad phase, shape-level narrow phase across lines, circles, and rectangles), using swept-line projectiles so fast bullets cannot pass through targets between frames
- Implemented gameplay including a state machine for game flow, constant-speed Bezier flight paths for enemy waves, and multiple enemy types with distinct attack patterns, health, and scoring

Petrichor | Unity, C#

Game of the Year, University of Alberta 2025

Post-apocalyptic puzzle platformer where players wake on a farm with no memory, uncover the truth of what happened, and choose their fate

- Designed and implemented platformer levels and solar panel light ray puzzles with moving objects, shaping the core puzzle-solving experience for players
- Added accessibility and quality-of-life features, including brightness and sound settings and level resets, and functionally tested them to ensure a smooth player experience
- Collaborated with team on level design, puzzle design, and core game mechanics with a focus on playability

Factoroom | Godot Engine, GDScript

Isometric factory tycoon game featuring resource management and machine upgrade mechanics

- Designed and implemented the core gameplay loop of buying resources, fulfilling orders, and upgrading machines, plus in-game email notifications and an endless mode to extend replayability
- Built intuitive UI components that make game information easy to read and act on, refining mechanics through functional testing and playtesting feedback
- Developed scalable data structures to reliably track game state and player progression

CodeCatchers | Java, Android Studio, Firebase

Location-based Android app game for scanning QR codes to collect procedurally-generated monsters

- Architected UI structure and implemented seamless navigation flow across multiple app screens
- Developed account system with user profiles, monster inventories, and optimized data structures for efficient retrieval, validated with unit tests
- Integrated real-time Firebase database synchronization to enable persistent data storage and multi-device access