Milestone List

Milestone 1: Project setup and environment configuration

Deliverables: A working development environment with TypeScript, Phaser 3, Node.js, Colyseus, and PostgreSQL set up. Repository initialized with proper structure, build scripts, and version control.

This milestone ensures all core technologies are properly installed, configured, and connected. It is foundational for development and prevents blockers later.

Milestone 2: Core front-end prototype client

Deliverables: A simple playable prototype where a player can move a character around an arena rendered in Phaser 3. Input is captured and displayed locally, with placeholder assets.

This milestone demonstrates basic game loop and rendering functionality on the client side, confirming that Phaser 3 is running correctly and ready for integration with networking.

Milestone 3: Realtime game server and network integration

Deliverables: A functioning server (Colyseus) that handles connections, accepts player inputs over WebSockets, and broadcasts state updates back to clients.

This milestone validates the multiplayer foundation of the project. Multiple clients should be able to connect, move around, and see each other in real time.

Milestone 4: Gameplay logic implementation

Deliverables: Authoritative server-side game logic including movement, collision detection, basic combat (attack/damage), and win/loss conditions.

With this milestone, the server controls the authoritative game state, preventing cheating and ensuring consistent gameplay across clients.

Milestone 5: AI opponent integration

Deliverables: AI-controlled bots that can perceive the game state, make decisions (chase, attack, evade, or seek pickups), and participate in matches.

Adds meaningful single-player functionality and demonstrates application of decision-making algorithms from AI coursework.

Milestone 6: Database and persistence layer

Deliverables: PostgreSQL schema implemented with tables for players, matches, match statistics, and leaderboards. Server integration to write match results and query player stats.

This milestone provides long-term data persistence for the game, allowing features like career stats and leaderboards.

Milestone 7: In-game chat and UI enhancements

Deliverables: A minimal in-game chat system for connected players and a more polished HUD with health bars, cooldowns, and player names.

Adds communication and usability features, improving the overall game experience.

Milestone 8: Metrics and observability

Deliverables: Basic telemetry collection for tick times, latency (RTT), and room statistics. Metrics available for debugging and tuning.

Ensures performance can be measured and verified during testing, supporting scalability and reliability.

Milestone 9: Testing and refinement

Deliverables: System-wide testing of networking stability, AI behavior, database integrity, and UI responsiveness. Identified issues addressed through refinements.

This milestone ensures the system is robust and all features work correctly before the final release.

Milestone 10: Final project delivery

Deliverables: Fully functional Pixel Coliseum game with online multiplayer, AI opponents, persistent stats, in-game chat, and documentation. Submission package including source code, setup instructions, and final report.

Marks the completion of the project, delivering all planned features and demonstrating them in a polished, playable state.

Timeline

Task/Milestone	Start Date	End Date	Responsible
Research requirements for Phaser/Colyseus	Jan 12	Jan 16	Cameron Estridge
Design overall project architecture	Jan 16	Jan 19	Cameron Estridge
Milestone 1: Project setup and environment configuration	---------	Jan 19	Cameron Estridge
Develop front-end input handler	Jan 20	Jan 24	Cameron Estridge
Implement front-end network client	Jan 25	Jan 29	Cameron Estridge
Develop front-end renderer and HUD	Jan 30	Feb 5	Cameron Estridge
Milestone 2: Core front-end prototype client	---------	Feb 5	Cameron Estridge
Design and implement game loop/tick system	Feb 6	Feb 11	Cameron Estridge
Implement server gameplay logic	Feb 12	Feb 18	Cameron Estridge
Implement server room/session manager	Feb 19	Feb 22	Cameron Estridge
Milestone 3: Realtime game server and network integration	---------	Feb 22	Cameron Estridge
Finalize server gameplay logic	Feb 23	Feb 27	Cameron Estridge
Milestone 4: Gameplay logic implementation	---------	Feb 27	Cameron Estridge
Design and implement AI perception, decision, and action	Feb 28	Mar 6	Cameron Estridge

Milestone 5: AI opponent integration	---------	Mar 6	Cameron Estridge
Design and finalize PostgreSQL schema	Mar 7	Mar 11	Cameron Estridge
Implement DB layer	Mar 12	Mar 18	Cameron Estridge
Milestone 6: Database and persistence layer	---------	Mar 18	Cameron Estridge
Develop in-game chat relay (server)	Mar 19	Mar 22	Cameron Estridge
Develop chat overlay (client)	Mar 23	Mar 26	Cameron Estridge
Refine HUD	Mar 27	Mar 31	Cameron Estridge
Milestone 7: In-game chat and UI enhancements	---------	Mar 31	Cameron Estridge
Implement metrics collections (tick times, RTT, player stats)	Apr 1	Apr 4	Cameron Estridge
Milestone 8: Metrics and observability	---------	Apr 4	Cameron Estridge
Test front-end/server integration	Apr 5	Apr 8	Cameron Estridge
Test AI opponent behavior	Apr 9	Apr 12	Cameron Estridge
Test DB persistence	Apr 13	Apr 15	Cameron Estridge
Refine UI based on playtests	Apr 16	Apr 18	Cameron Estridge
Validate full system integration	Apr 19	Apr 21	Cameron Estridge
Milestone 9: Testing and Refinement	---------	Apr 21	Cameron Estridge
Document system design and setup instructions	Apr 22	Apr 24	Cameron Estridge
Milestone 10: Final project delivery	---------	Apr 24	Cameron Estridge

Effort Matrix

Task	Hours (Cameron Estridge)
Research requirements	8
Design architecture	10
Develop front-end input handler	15
Implement front-end network client	15
Develop front-end renderer and HUD	20
Design and implement game loop	25
Implement gameplay logic	30
Develop chat relay	10
Implement room/session manager	15
AI module	30
Develop DB Layer match result writer	12
Implement DB Layer player profile service	12
Implement DB Layer leaderboard service	11
Configure DB access	10
PostgreSQL schema design	12
Metrics system	15
Testing (front-end/server)	20
Testing (AI)	15
Testing (DB persistence)	15
Documentation	12
UI refinement	10
System validation	15
Total	332