App Summary - One Page

Evidence scope: README.md, index.html, galaga.html, src/galaga/*, src/snake/*

What it is

A static web app for System Academy with a Korean marketing homepage and a separate browser Galaga game page.
The repo evidence shows client-side HTML/CSS/JS modules only; backend services are not present.

Who it is for

Primary persona: middle and high school students plus parents evaluating course offerings and how to contact the academy.

What it does

- Shows academy branding, hero message, subject cards (English, Math, Science), operating hours, and contact methods.
- Uses responsive layout rules so the homepage shifts from multi-column grids to single-column on small screens.
- Provides a playable Galaga screen with canvas rendering and a live HUD for score, lives, level, and status.
- Supports keyboard and pointer controls for movement, shooting, pause/resume, and restart.
- Runs a real-time loop with requestAnimationFrame and dt-based state updates for smoother gameplay.
- Includes isolated game logic modules with Node tests covering collisions, score, boundaries, and wave progression.

How it works (repo evidence only)

- Entry points: index.html (academy site) and galaga.html (game UI and controls).
- UI/controller: src/galaga/game.mjs binds DOM input, calls step(), and re-renders canvas plus HUD each frame.
- Domain logic: src/galaga/logic.mjs and src/snake/logic.mjs implement pure state transitions.
- Validation: src/galaga/logic.test.mjs and src/snake/logic.test.mjs run with node:test.
- Data flow: user input -> input flags -> step(state, input) -> next state -> canvas and DOM render.
- Backend API, database, auth, deployment pipeline: Not found in repo.
- Snake page route/launcher HTML that uses src/snake/game.mjs: Not found in repo.

How to run (minimal)

- From repo root: python3 -m http.server 4173
- Open homepage: http://localhost:4173/index.html
- Open game page: http://localhost:4173/galaga.html