

# ✦ Crystalline

## *Elemental Knowledge Puzzle*

---

Programming Project — Full Technical Summary

<https://crystal-line.vercel.app>

**Fabiana Fazio**

March 2026

# 1. What Is Crystalline?

Crystalline is a browser-based puzzle game that combines match-3 gem gameplay with a real-time AI trivia quiz system. The player swaps colorful gems on an 8×8 board to create lines of 3 or more matching elements, earning points as they explode. As score builds, special gold-marked Rune Gems appear — match one and an AI-generated quiz question pops up about your chosen topic. Answer correctly and earn bonus points and extra moves. Answer wrong and lose moves.

The game has 8 levels of increasing difficulty. Each level has a high score target the player must reach before running out of moves. Because score targets are large and base gem scores are small, players must chain combos, trigger elemental reactions, and — most importantly — answer many Rune questions to advance.









**Design goal:**

Levels are intentionally long so players get asked many questions per level. Rune Gems spawn every 75–120 points and up to 5 can be on the board at once — meaning a full level will have 8–15+ quiz moments.

# 2. How to Play

## Step 1 — Choose Your Topic

When the game loads, you see 8 topic cards. Click one to choose the subject your AI quiz questions will be about, then click the gold Begin button.

Topic	Subject
 Science	Physics, chemistry & biology
 Mathematics	Algebra, geometry & calculus
 History	World events & civilizations
 Coding	Programming & computer science
 Biology	Life, cells & ecosystems
 Geography	Countries, capitals & nature
 Art & Music	Artists, movements & composers
 Space	Astronomy & the cosmos

## Step 2 — Match Gems on the Board

The game board is an 8×8 grid of colorful hexagonal gems. Each gem has one of 6 elements:

Gem	Element & Color
	Fire — red / orange
	Water — blue
	Ice — light blue
	Lightning — yellow
	Earth — green
	Shadow — purple

Click one gem, then click an adjacent gem to swap them. **If the swap creates a line of 3 or more matching gems going horizontally or vertically**, those gems explode and you earn points. Diagonal matches do not count. After gems disappear, gems above fall down to fill gaps and new gems drop from the top — sometimes creating automatic chain reactions that multiply your score.

## Step 3 — Watch for Rune Gems ✦



Once you earn enough points, Rune Gems start appearing. They look like a normal gem of their element color, but have a gold border, a gold ✦ symbol, and a quiz abbreviation (like "DNA" or "WWI") printed on them.

**Important:**

Rune Gems match exactly like regular gems. If a Rune Gem is part of a line of 3 or more matching gems — at the end, the start, or in the middle — it explodes with the others and triggers a quiz question. You cannot skip a rune; it will be matched eventually as the board refills.

When a Rune Gem is matched, a quiz popup appears with:

- A multiple-choice question about your chosen topic
- A generous countdown timer — plenty of time to read and think
- 4 answer choices labeled A, B, C, D
- The correct answer and an explanation shown after you answer






Result	What happens
 Correct answer	Bonus points + 2 extra moves added to your total
 Wrong answer	Lose moves as a penalty — the correct answer is still shown

Result	What happens
 Timer runs out	Treated as wrong — lose moves, but you see the answer anyway

## Step 4 — Reach the Score Target

Each level has a score target displayed in the HUD. Reach it before you run out of moves to advance to the next level. Run out of moves first and it is Game Over. The game always shows your current score, target, and remaining moves so you know exactly where you stand.

## Controls

Action	How to do it
Swap gems	Click a gem, then click an adjacent gem
Pause the game	Click  Pause in the top bar, or press P or Escape
Resume	Click  Resume in the pause menu
Go to home screen	Click  Home in the top bar (confirms before leaving)
View scoring guide (mobile)	Tap  Info in the top bar — slides down a full scoring panel
Restart current level	Click  Restart in the pause menu

## 3. How to Win

Beat all 8 levels by reaching each level's score target. The game ends in one of two ways:

1. Reach the score target before moves run out — advance to the next level
2. Run out of moves before the target — Game Over

### The key insight:

Basic gem matches alone are worth only 50–150 base points. Level targets range from 3,000 to 55,000 points. You must combine all three scoring systems — combos, elemental reactions, and Rune question bonuses — to realistically clear any level.

## Winning Strategy

### 1 — Chain Combos

Every consecutive match in one turn multiplies your score by +25%. Chaining 4 or 5 matches in a row makes even small 3-gem matches worth far more. Always look for a move that will cause new gems to fall and create another match automatically.

## 2 — Trigger Elemental Reactions

When your match touches two different element types, a reaction fires and adds bonus points on top of the match score. Plan your swaps to hit multiple reactions at once.

Reaction	Elements
Melt!	🔥 Fire + ❄️ Ice — +60 pts
Shock!	💧 Water + ⚡ Lightning — +75 pts
Ignite!	⚡ Lightning + 🔥 Fire — +70 pts
Absorb!	🌿 Earth + 💧 Water — +55 pts
Wildfire!	🔥 Fire + 🌿 Earth — +100 pts
Phantom!	💜 Shadow + any element — +80 pts

## 3 — Answer Rune Questions Correctly

This is the most important part of the game. Rune Gems fire frequently — every 75–120 points earned — and up to 5 can be on the board at once. Each correct answer gives bonus points AND adds 2 extra moves, which keeps you alive longer to earn even more score. Players who answer consistently will almost always beat the level. Players who get them wrong will run out of moves.

# 4. Scoring Reference

### Base Gem Match Points

Match size	Base points
3 gems in a row	50 pts
4 gems in a row	90 pts
5 gems in a row	150 pts

**Combo multiplier:** each consecutive match in one turn adds +25% to your base score.

**Chain reaction multiplier:** each automatic chain after gems fall adds +35%.

## Rune Question Bonus by Level

Level	Correct Answer Bonus / Timer / Max Runes
1 — Novice	+200 pts + 2 moves   40 sec timer   3 runes max
2 — Apprentice	+220 pts + 2 moves   36 sec timer   3 runes max
3 — Scholar	+250 pts + 2 moves   32 sec timer   4 runes max
4 — Adept	+280 pts + 2 moves   28 sec timer   4 runes max
5 — Veteran	+310 pts + 2 moves   24 sec timer   4 runes max
6 — Expert	+340 pts + 2 moves   20 sec timer   5 runes max
7 — Elite	+380 pts + 2 moves   17 sec timer   5 runes max
8 — Master	+420 pts + 2 moves   14 sec timer   5 runes max

## Level Score Targets

Level	Score Target / Moves Available
1 — Novice	3,000 pts   40 moves
2 — Apprentice	6,000 pts   38 moves
3 — Scholar	10,000 pts   36 moves
4 — Adept	15,000 pts   34 moves
5 — Veteran	22,000 pts   32 moves
6 — Expert	30,000 pts   30 moves
7 — Elite	40,000 pts   28 moves
8 — Master	55,000 pts   26 moves

## 5. Technology Stack

---

The game is built with three core web technologies — no game engine or external libraries were used for gameplay.

### 5.1 HTML (index.html)

Defines every element the player sees — topic selection screen, game board, score display, question popup, pause button, home button, and mobile info drawer. All screens live in the same file and are shown or hidden using CSS classes controlled by JavaScript.

## 5.2 CSS (inside index.html)

All visual styling is written in a `<style>` block. This includes the dark space theme, starfield background, hexagonal gem board, responsive top bar, slide-down mobile info drawer, and all overlay screens (pause, question, level complete, game over).

## 5.3 JavaScript (game.js)

The brain of the game. Handles everything that moves or reacts:

- Drawing the 8×8 gem board using the HTML Canvas API at 60 frames per second
- Detecting horizontal and vertical matches of 3+ same-color gems
- Animating gem swaps, explosions, gravity (gems falling to fill gaps), and particles
- Tracking score, moves, combos, level progress, and Rune spawn timing
- Spawning Rune Gems and triggering quiz questions when they are matched
- Communicating with the AI API to fetch fresh questions before each level
- Handling all user interaction — clicks, P/Escape keyboard shortcuts, home/pause/restart

## 5.4 Vercel (Hosting + Serverless Functions)

The game is deployed on Vercel, a free cloud hosting platform. Vercel also handles API key security through a serverless function. A serverless function is a small piece of backend code that runs in the cloud only when called — no traditional server needed. The file `api/questions.js` is that function. When the game needs questions, it calls `/api/questions`, Vercel runs the function in their cloud, contacts OpenAI with the key, and returns the questions. The API key never reaches the browser.

## 5.5 OpenAI API

Quiz questions are generated by OpenAI's GPT model. The game sends a request with a prompt like "Generate 8 trivia questions about Science at Scholar difficulty." OpenAI returns the questions in JSON format, and the game displays them. Every session generates completely fresh, unique questions.

# 6. API Key Security

---

## The Problem

An API key is a secret password for a paid service. If someone sees the key, they can use it billed to your account. Browser JavaScript files are publicly readable by anyone in DevTools — so the key cannot live in game.js or index.html.

#### Rule #1:

Never put an API key in any JavaScript file sent to the browser. Anyone can read those files using browser DevTools.

## The Solution — Serverless Function

The API key lives in only two places:

3. A local .env file on the developer's computer — never pushed to GitHub (listed in .gitignore)
4. Vercel's Environment Variables dashboard — set once, stored encrypted, never visible in any file

api/questions.js runs in Vercel's cloud. When the game calls /api/questions, Vercel runs the function, which reads the key from process.env.OPENAI\_API\_KEY (injected securely by Vercel), calls OpenAI, gets the questions, and returns them. The key is never exposed.

## Request Flow

Step	What happens
1	Player starts a level in the browser
2	game.js calls fetch('/api/questions?topic=science&level=0')
3	Browser sends that request to Vercel's servers
4	Vercel runs api/questions.js in a secure cloud environment
5	Function reads OPENAI_API_KEY from secure environment variables
6	Sends a request to OpenAI with the key and a prompt describing the questions needed
7	OpenAI responds with 8 quiz questions in JSON format
8	Serverless function returns those questions to the browser
9	game.js caches them for the current level — API key never touched the browser

## Fallback System

If the API call fails for any reason — no key, network error, rate limit — game.js automatically uses the built-in FALLBACK question bank. The game never crashes or shows an error. It silently keeps playing with pre-written questions.



### Resilience design:

The game always works even without an API key. AI questions are an enhancement, not a requirement. This means the game can be demoed anywhere, even offline, using the built-in question bank.

## 7. Project File Structure

File	What it does
public/index.html	All screens, CSS styling, and HTML structure
public/game.js	All game logic, canvas rendering, scoring, AI calls, pause/home system
api/questions.js	Vercel serverless function — holds API key, contacts OpenAI, returns questions
vercel.json	Routes /api/questions to the function; everything else serves from /public
.env	Local API key (never pushed to GitHub)
.env.example	Template showing required environment variables with placeholder values
.gitignore	Blocks .env, node_modules, and logs from being committed to GitHub

## 8. Quick Reference

Component	Technology / Approach
Game UI & screens	HTML + CSS — no external framework
Game logic & rendering	Vanilla JavaScript + HTML Canvas API
Hosting	Vercel (free tier, auto-deploys from GitHub)
API key security	Vercel serverless function + encrypted Environment Variables
AI questions	OpenAI GPT — fresh unique questions generated every session
Fallback questions	Built-in FALLBACK object in game.js — works without internet

Component	Technology / Approach
Version control	Git + GitHub — .env gitignored, key never committed
Match detection	Horizontal + vertical only — custom scan of 8×8 2D array
Rune gems	Normal gems flagged isRune=true — match like regular gems
Score progression	8 levels, targets 3,000 → 55,000 pts, combo + reaction multipliers
Rune frequency	3–5 runes on board at once, spawning every 75–120 pts earned