CoderQuest

Complete System Submission

Full Stack Architecture

Database • Phaser Game Engine • React Components • SCSS Styling

Date: November 28, 2025

Status: Production Ready

Version: 1.0

CoderQuest - Complete System Submission

Date: November 28, 2025 **Project Status:** Ready for Submission (Phaser with Pseudocode)

Table of Contents

1. **Database Schema (SQL)** 2. **Phaser Game Engine Layer (Pseudocode)** 3. **React Event Processors (Logic)** 4. **React Components Hierarchy** 5. **SCSS Styling**

Database Schema (SQL)

Architecture: Normalized 3NF (Third Normal Form)

CODE*BLOCK*0

ER Diagram (Relationships)

CODE*BLOCK*1

Phaser Game Engine Layer (Pseudocode)

Layer 1: Game Scene Initialization

CODE*BLOCK*2

Layer 2: NPC Meeting & Dialog Emission

CODE*BLOCK*3

Layer 3: Camera & Player Movement

CODE*BLOCK*4

Layer 4: Map Transitions

CODE*BLOCK*5

React Event Processors (Logic)

Event Processing Flow

CODE*BLOCK*6

Processor 1: Game Event Emitter (Hook)

CODE*BLOCK*7

Processor 2: Game UI Event Handler

CODE*BLOCK*8

React Components Hierarchy

CODE*BLOCK*9

Component Details

Parent: GameUI

- **Props:** None
- **State:** dialogOpen, questOpen, dialogData, questData
- **Events Listen:** showDialog, showQuest, closePopup
- **Events Emit:** dialogClosed, questClosed, questCompleted

  #### Child 1: DialogBox
- **Props:** npcData, onClose, isOpen
- **State:** isAnimating
- **Events:** None
- **Purpose:** Display NPC dialog with animation

  #### Child 2: QuestPopup

- **Props:** questData, isOpen, onClose
- **State:** mode (lesson|quiz), isAnimating
- **Events:** None
- **Children:** Lesson OR Quiz (conditionally rendered)

  #### Grandchild 2a: Lesson
- **Props:** lessonData, onStartQuiz, onClose
- **State:** None
- **Purpose:** Display lesson content with code examples

  #### Grandchild 2b: Quiz
- **Props:** quizData, onComplete, onBack
- **State:** currentQuestionIndex, score, answers, showResults
- **Children:** MultipleChoiceQuestion OR FillInBlanksQuestion OR QuizResults

  #### Great-grandchild 2b1: MultipleChoiceQuestion
- **Props:** question, onAnswer, disabled
- **State:** selected, answered, feedback
- **Purpose:** Single multiple choice question

  #### Great-grandchild 2b2: FillInBlanksQuestion
- **Props:** question, onAnswer, disabled
- **State:** answers, answered, feedback
- **Purpose:** Single fill-in-the-blanks question

  #### Great-grandchild 2b3: QuizResults
- **Props:** score, totalQuestions, answers, onRetry, onBack
- **State:** None
- **Purpose:** Display quiz score and performance

SCSS Styling

Architecture: Component-Scoped Styles

Global Variables & Mixins **CODE***BLOCK*10 **CODE***BLOCK*11 #### Component Styles: DialogBox **CODE***BLOCK*12 #### Component Styles: Quiz **CODE***BLOCK*13 #### Component Styles: MultipleChoice Question **CODE***BLOCK*14

Summary Table

Layer	Component	Responsibility	Events
Phaser	GameScene	Game loop, rendering, physics	`showDialog`, `showQuest`
Phaser	NPCSystem	NPC behavior, proximity detection	`npcInRange`, `dialogClosed`
Event Bus	window.gameEvents	Event mediation	All events pass through
React Hook	useGameEvents	Create event emitter	N/A
React Hook	useGameEventListener	Subscribe to events	All events
React Hook	useGameEventEmitter	Emit to Phaser	All events
React	GameUI	Main processor	Listens & emits all
React	DialogBox	NPC dialog display	Emits `dialogClosed`
React	QuestPopup	Quest container	Routes to Lesson/Quiz
React	Lesson	Lesson content	Emits `startQuiz`
React	Quiz	Quiz manager	Displays questions
React	MultipleChoiceQuestion	MCQ display	Submits answers
React	FillInBlanksQuestion	Fill-in display	Submits answers
React	QuizResults	Results display	Emits `questCompleted`
CSS	Dialog Styles	Visual appearance	N/A
CSS	Quiz Styles	Visual appearance	N/A

Implementation Status

✅ **COMPLETE:**

- Database schema (normalized 3NF)
- React component hierarchy with full source code
- Event communication system (useGameEvents hooks)
- SCSS styling with design tokens
- Component prop documentation

🟡 **PSEUDOCODE (Ready for Implementation):**

- Phaser game scene initialization
- NPC interaction system
- Player movement and camera
- Map transitions and events
- Database integration

Next Steps for Implementation

1. **Phaser Layer:** Implement pseudocode in Phaser v3 2. **Backend:** Create REST API endpoints for database operations 3. **Integration:** Connect React components to backend APIs 4. **Testing:** Unit and integration testing for all layers 5. **Deployment:** Deploy to production environment