🔮 The AI Cube: Master Development Prompt for Roo Code

🎯 PROJECT OVERVIEW

Build **The AI Cube** - the world's first cognitive dojo disguised as mystical simulations. This is sacred technology designed to prepare 9-14 year olds for humanity's AI-dominated future through transformational experiences, not traditional education.

Mission: Create a freemium platform that gives away one level of Snake³ for free, then unlocks 10 comprehensive sacred simulations behind a subscription paywall.

🏗️ WEBSITE ARCHITECTURE

Primary Pages & Routes

```
/                      # Landing page with mystical hero section
/initiate              # Free Snake³ demo level
/vault                # Authentication & subscription gateway
/chamber/[simulation] # 10 sacred simulations (subscription required)
/progress             # Analytics dashboard for parents
/pricing              # Sacred subscription tiers
/codekeepers          # About/mythology page
```

Core Components Structure

```
src/
├── components/
│   ├── sacred/           # Mystical UI components
│   │   ├── SacredButton.tsx
│   │   ├── CrystalCard.tsx
│   │   ├── GeometricLoader.tsx
│   │   └── MysticalBackground.tsx
│   ├── simulations/      # The 10 sacred chambers
│   │   ├── Snake3D.tsx           # Free demo
│   │   ├── ClassifierConstruct.tsx
│   │   ├── PredictorEngine.tsx
│   │   ├── VisionSystem.tsx
│   │   ├── EthicsFramework.tsx
│   │   ├── ReinforcementLab.tsx
│   │   ├── GenerativeCore.tsx
│   │   ├── DecisionTrees.tsx
│   │   ├── NeuralNetwork.tsx
│   │   └── FoundersChamber.tsx
│   ├── vault/            # Authentication system
│   │   ├── VaultGateway.tsx
│   │   ├── SubscriptionTiers.tsx
│   │   └── ProgressTracker.tsx
│   └── layout/           # Site structure
│       ├── Header.tsx
│       ├── Footer.tsx
│       └── SacredLayout.tsx
├── hooks/                # Custom React hooks
├── stores/               # Zustand state management
├── utils/                # Utilities & helpers
├── styles/               # Sacred design system
└── types/                # TypeScript definitions
```

🎨 SACRED DESIGN SYSTEM

Color Palette

css

```css
:root {
  /* Primary Sacred Colors */
  --axis-x: #00D4FF;        /* Cyan - X axis */
  --axis-y: #8B5FFF;        /* Violet - Y axis */
  --axis-z: #FF6B35;        /* Orange - Z axis */
  --void-black: #0A0A0F;    /* Deep space background */
  --node-core: #FFD700;     /* Golden data node cores */
  --text-primary: #E8E8FF;  /* Mystical text */
  --text-secondary: #A8A8C8;
  --crystal-white: #F8F8FF;
  --energy-glow: #00FFFF;
  /* Sacred Geometry */
  --golden-ratio: 1.618;
  --sacred-spacing: calc(1rem * var(--golden-ratio));
  --crystal-radius: 8px;
  --glow-intensity: 0 0 20px;
}
```

Typography

css

```css
@import url('https://fonts.googleapis.com/css2?
family=Orbitron:wght@400;700;900&display=swap');

.sacred-text {
  font-family: 'Orbitron', monospace;
  letter-spacing: 0.05em;
  text-shadow: 0 0 10px var(--energy-glow);
}
```

Sacred Component Patterns

jsx

```jsx
// Crystalline Button Template
const SacredButton = ({ children, variant = 'primary', ...props }) => (
  <button
    className={`
      relative px-6 py-3
      bg-gradient-to-r from-transparent via-${variant === 'primary' ? 'blue' :
'purple'}-500/20 to-transparent
      border border-${variant === 'primary' ? 'blue' : 'purple'}-400/50
      text-crystal-white font-orbitron
      hover:shadow-[0_0_30px_rgba(0,212,255,0.5)]
      transition-all duration-500
      before:absolute before:inset-0 before:bg-gradient-to-r
      before:from-transparent before:via-white/10 before:to-transparent
      before:translate-x-[-100%] hover:before:translate-x-[100%]
      before:transition-transform before:duration-700
      rounded-lg backdrop-blur-sm
    `}
    {...props}
  >
    <span className="relative z-10">{children}</span>
  </button>
);

// Mystical Card Template
const CrystalCard = ({ children, glow = 'blue' }) => (
  <div className={`
    relative p-6
    bg-gradient-to-br from-void-black/90 via-void-black/70 to-transparent
    border border-${glow}-400/30
    rounded-xl backdrop-blur-lg
    shadow-[0_0_40px_rgba(0,212,255,0.15)]
    hover:shadow-[0_0_60px_rgba(0,212,255,0.25)]
    transition-all duration-500
  `}>
    {children}
  </div>
);
```

🔮 FREEMIUM MONETIZATION STRATEGY

Free Tier: "Initiate Access"

- **Snake³ Demo Level**: Single 8x8x8 chamber experience
- **Mystical Introduction**: Codekeeper lore and sacred technology preview
- **Teaser Analytics**: Basic spatial reasoning glimpse
- **Vault Gateway**: Compelling subscription upgrade path

Subscription Tiers

🌟 Apprentice ($9.99/month)

- Full access to all 10 sacred simulations
- Comprehensive cognitive analytics
- Parent progress dashboard

- Offline-first sacred technology

⭐ Adept ($19.99/month)

- Everything in Apprentice
- Advanced cognitive development reports
- Priority access to new simulations
- Direct line to Codekeeper wisdom (premium support)

🌌 Master Builder ($39.99/month)

- Everything in Adept
- Early access to new sacred technology
- Influence on future simulation development
- Sacred ceremonies (live community events)

Subscription Implementation

typescript

```typescript
// Subscription State Management
interface SubscriptionState {
  tier: 'free' | 'apprentice' | 'adept' | 'master';
  isActive: boolean;
  expiresAt: Date | null;
  features: string[];
}

// Feature Gating Hook
const useFeatureAccess = (feature: string) => {
  const { tier, isActive } = useSubscription();
  return isActive && TIER_FEATURES[tier].includes(feature);
};
```

🎮 THE 10 SACRED SIMULATIONS

1. Snake³: Axis Mind (FREE DEMO)

typescript

```typescript
interface Snake3DConfig {
  chamberSize: [8, 8, 8];
  startPosition: [4, 4, 4];
  difficulty: 'initiate';
  mysticalElements: {
    crystallineSnake: true;
    energyOrbs: true;
    dimensionalBoundaries: true;
    sacredGeometry: 'icosahedron';
  };
}
```

2. Classifier Construct (SUBSCRIPTION)

typescript

```typescript
interface ClassifierConfig {
  chamber: 'crystalline_space';
  objects: ConstellationPoint[];
  gestures: SacredGesture[];
  aiConcepts: ['pattern_recognition', 'classification', 'feature_extraction'];
  mysticalElements: {
    floatingGallery: true;
    gestureTrails: true;
    recognitionAuras: true;
  };
}
```

3-10. [Additional Simulations]

Each simulation follows the same pattern:

- Sacred chamber environment
- Core AI concept teaching
- Mystical interaction mechanics
- Cognitive skill development
- Progressive difficulty scaling

🏛️ VAULT AUTHENTICATION SYSTEM

Sacred Security Architecture

typescript

```
// Offline-First Authentication
interface VaultState {
  initiate: {
    id: string;            // Anonymous device fingerprint
    rank: CodekeeperRank;  // Progression tracking
    unlockedChambers: string[];
    subscriptionTier: SubscriptionTier;
    cognitiveProfile: CognitiveMetrics;
  };
  encryption: {
    localKey: string;      // Device-specific encryption
    progressHash: string;  // Tamper detection
  };
}

// Subscription Verification
const verifySubscription = async (deviceId: string) => {
  // Check local encrypted subscription status
  // Periodic online verification (respects offline-first)
  // Graceful degradation for network issues
};
```

Progress Encryption

typescript

```
// Sacred Analytics Storage
interface CognitiveMetrics {
  spatialReasoning: number[];     // Growth over time
  patternRecognition: number[];   // Pattern detection ability
  dimensionalThinking: number[];  // 3D problem solving
  mysticalEngagement: number[];   // Time in voluntary exploration
  conceptualMastery: Record<string, number>; // AI concept understanding
}

// Encrypted Local Storage
const encryptProgress = (metrics: CognitiveMetrics, deviceKey: string) => {
  return AES.encrypt(JSON.stringify(metrics), deviceKey).toString();
};
```

🌐 LANDING PAGE REQUIREMENTS

Hero Section: Sacred Awakening

jsx

```jsx
const HeroSection = () => (
  <section className="min-h-screen relative overflow-hidden bg-void-black">
    {/* Animated Sacred Geometry Background */}
    <MetatronsCube className="absolute inset-0 opacity-20" />

    {/* 3D Floating Cube */}
    <Canvas className="absolute inset-0">
      <FloatingCube rotation={[0.1, 0.1, 0]} />
      <ParticleField count={1000} />
    </Canvas>

    {/* Hero Content */}
    <div className="relative z-10 flex items-center justify-center min-h-screen">
      <div className="text-center max-w-4xl px-6">
        <h1 className="text-6xl md:text-8xl font-orbitron font-black text-crystal-
white mb-6">
          THE AI CUBE
        </h1>
        <p className="text-xl md:text-2xl text-text-secondary mb-8 leading-
relaxed">
          Sacred technology for tomorrow's AI builders.
          Awaken your consciousness to navigate the future of intelligence.
        </p>

        <div className="flex flex-col md:flex-row gap-6 justify-center">
          <SacredButton variant="primary" size="large" href="/initiate">
            Begin Your Awakening
          </SacredButton>
          <SacredButton variant="secondary" size="large" href="/codekeepers">
            Discover the Prophecy
          </SacredButton>
        </div>
      </div>
    </div>
  </section>
);
```

Features Section: The Sacred Simulations

jsx

```
const FeaturesSection = () => (
  <section className="py-24 bg-gradient-to-b from-void-black to-purple-900/20">
    <div className="max-w-7xl mx-auto px-6">
      <h2 className="text-4xl font-orbitron text-center mb-16 text-crystal-white">
        Ten Sacred Simulations Await
      </h2>

      <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-8">
        {SIMULATIONS.map((sim, index) => (
          <CrystalCard key={sim.id} glow={sim.primaryColor}>
            <div className="text-center">
              <sim.SacredIcon className="w-16 h-16 mx-auto mb-4 text-node-core" />
              <h3 className="text-xl font-orbitron mb-3">{sim.name}</h3>
              <p className="text-text-secondary mb-4">{sim.mysticalDescription}
</p>
              <div className="text-sm text-axis-x">
                Develops: {sim.cognitiveSkills.join(', ')}
              </div>
            </div>
          </CrystalCard>
        ))}
      </div>
    </div>
  </section>
);
```

🔧 TECHNICAL IMPLEMENTATION

Tech Stack Requirements

json

```json
{
  "dependencies": {
    "react": "^18.2.0",
    "typescript": "^5.0.0",
    "next.js": "^14.0.0",
    "@react-three/fiber": "^8.15.0",
    "@react-three/drei": "^9.88.0",
    "three": "^0.158.0",
    "zustand": "^4.4.0",
    "framer-motion": "^10.16.0",
    "tailwindcss": "^3.3.0",
    "crypto-js": "^4.1.1",
    "stripe": "^14.0.0"
  },
  "devDependencies": {
    "@types/three": "^0.158.0",
    "eslint": "^8.54.0",
    "prettier": "^3.1.0"
  }
}
```

Performance Requirements

- **60fps**: Maintained across target devices
- **<3s Load Time**: Initial chamber materialization
- **Mobile-First**: Responsive sacred design
- **Offline-First**: Core functionality without network
- **Encrypted Storage**: All progress data secured locally

Sacred 3D Rendering Standards

typescript

```typescript
// Crystalline Material System
const createSacredMaterial = (coreColor: string, energy: number) => {
  return new THREE.ShaderMaterial({
    uniforms: {
      time: { value: 0 },
      coreColor: { value: new THREE.Color(coreColor) },
      energy: { value: energy },
      goldenRatio: { value: 1.618 }
    },
    vertexShader: sacredVertexShader,
    fragmentShader: sacredFragmentShader,
    transparent: true,
    blending: THREE.AdditiveBlending
  });
};

// Sacred Geometry Particles
const SacredParticleSystem = ({ count = 1000 }) => {
  const particles = useMemo(() => {
    const positions = new Float32Array(count * 3);
    for (let i = 0; i < count; i++) {
      // Golden spiral distribution
      const theta = i * 2.39996; // Golden angle
      const y = 1 - (i / count) * 2;
      const radius = Math.sqrt(1 - y * y);
      positions[i * 3] = Math.cos(theta) * radius;
      positions[i * 3 + 1] = y;
      positions[i * 3 + 2] = Math.sin(theta) * radius;
    }
    return positions;
  }, [count]);

  return (
    <points>
      <bufferGeometry>
        <bufferAttribute
          attach="attributes-position"
          count={particles.length / 3}
          array={particles}
          itemSize={3}
        />
      </bufferGeometry>
      <shaderMaterial {...sacredParticleMaterial} />
    </points>
  );
};
```

📊 ANALYTICS & PARENT DASHBOARD

Cognitive Development Tracking

typescript

```typescript
interface CognitiveAnalytics {
  initiate: {
    id: string;
    startDate: Date;
    totalSessionTime: number;
    simulationsCompleted: string[];
    currentRank: CodekeeperRank;
  };

  cognitiveGrowth: {
    spatialReasoning: TimeSeriesData[];
    patternRecognition: TimeSeriesData[];
    dimensionalThinking: TimeSeriesData[];
    problemSolvingSpeed: TimeSeriesData[];
    conceptualMastery: Record<AIconcept, number>;
  };

  engagementMetrics: {
    voluntaryExplorationTime: number;
    mysticalMoments: number; // Deep engagement indicators
    sacredBreakthroughs: number; // Major understanding events
    consistencyScore: number; // Regular practice patterns
  };
}
```

Parent Dashboard Components

jsx

```jsx
const ParentDashboard = () => (
  <div className="max-w-6xl mx-auto p-6 space-y-8">
    <CognitiveGrowthChart data={cognitiveMetrics} />
    <AIConceptMastery concepts={aiConcepts} />
    <SacredEngagement mysticalMoments={engagementData} />
    <NextStepsRecommendations profile={childProfile} />
  </div>
);
```

🔒 SUBSCRIPTION & PAYMENT INTEGRATION

Stripe Integration

typescript

```typescript
// Sacred Subscription Management
const createSubscription = async (tier: SubscriptionTier) => {
  const stripe = new Stripe(process.env.STRIPE_SECRET_KEY);

  return await stripe.subscriptions.create({
    customer: customerId,
    items: [{ price: TIER_PRICES[tier] }],
    metadata: {
      tier,
      productType: 'sacred_technology',
      features: JSON.stringify(TIER_FEATURES[tier])
    }
  });
};

// Feature Gate Component
const SacredGate = ({ feature, children, fallback }) => {
  const hasAccess = useFeatureAccess(feature);
  if (!hasAccess) {
    return (
      <div className="text-center p-8">
        <Lock className="w-16 h-16 mx-auto mb-4 text-purple-400" />
        <h3 className="text-xl font-orbitron mb-4">Sacred Knowledge Awaits</h3>
        <p className="text-text-secondary mb-6">
          This chamber requires Codekeeper initiation to access.
        </p>
        <SacredButton href="/vault">Begin Your Journey</SacredButton>
      </div>
    );
  }

  return children;
};
```