```
Core', () => {
  test('Simulation teaches intended cognitive skill', async () => {

    const beforeSkill = await measureCognitiveBaseline();

    await simulateGameplay(300); // 5 minutes

    const afterSkill = await measureCognitiveGrowth();



expect(afterSkill.spatialReasoning).toBeGreaterThan(beforeSkill.spatialReasoning);

  });

});
```

💳 PRIORITY 4: SUBSCRIPTION FLOW TESTS

Freemium Flow Validator

typescript

```
describe('Subscription Core', () => {

  test('Free to paid conversion works smoothly', async () => {

    // Test free Snake³ experience

    const freeExperience = await playFreeSnake3D();

    expect(freeExperience.completionRate).toBeGreaterThan(0.7);


    // Test subscription gateway

    const subscriptionFlow = await testUpgradeFlow();

    expect(subscriptionFlow.conversionFriction).toBeLow();

    expect(subscriptionFlow.paymentSuccess).toBe(true);

  });

});
```

🔒 PRIORITY 5: PRIVACY & SECURITY TESTS

Sacred Data Protection Validator

typescript

```typescript
describe('Privacy & Security Core', () => {
  test('User data encrypted and offline-first', async () => {
    const dataStorage = await analyzeDataHandling();
    expect(dataStorage.localEncryption).toBe(true);
    expect(dataStorage.offlineCapability).toBe(true);
    expect(dataStorage.personalDataTransmission).toBe(false);
  });
});
```

🤖 ROO CODE COMPLIANCE CHECKLIST

Simple Validation for Each Component

typescript

```typescript
// Add this single test to each component Roo Code generates
const validateSacredComponent = (component) => {
  return {
    // Visual (5 checks)
    usesIcosahedronNotSphere: component.geometry === 'icosahedron',
    hasGoldenRatioSpacing: component.spacing === 'calc(1rem * 1.618)',
    materialIsShimmering: component.material.includes('shader'),
    animationIsOrganic: component.animation.easing !== 'linear',
    colorsAreSacred: component.colors.includes('#FFD700'),

    // Voice (3 checks)
    voiceIsMystical: !component.voice.includes('Good job'),
    noEducationalJargon: !component.voice.includes('Click here'),
    respectsIntelligence: component.voice.includes('consciousness'),

    // Performance (3 checks)
    under512MB: component.memoryUsage < 512,
    loads3Seconds: component.loadTime < 3000,
    maintains60FPS: component.averageFPS >= 58,

    // Sacred Score (must be > 8/11)
    sacredScore: function() {
      return Object.values(this).filter(v => v === true).length;
    }
  };
};

// Usage in each component test:
test('Component meets sacred standards', () => {
  const validation = validateSacredComponent(component);
  expect(validation.sacredScore()).toBeGreaterThan(8);
});
```

🎮 SIMULATION-SPECIFIC QUICK TESTS

Snake³: Axis Mind

typescript

```
test('Snake³ develops spatial reasoning', async () => {
  const game = await loadSnake3D();
  expect(game.requires3DThinking).toBe(true);
  expect(game.hasFadingTail).toBe(true); // Working memory challenge

  expect(game.feelsMystical).toBe(true);

});
```

Classifier Construct

typescript

```
test('Classifier teaches pattern recognition', async () => {

  const classifier = await loadClassifier();

  expect(classifier.hasGestureInteraction).toBe(true);

  expect(classifier.teachesAIvsNatural).toBe(true);

  expect(classifier.feelsMystical).toBe(true);

});
```

Template for All 10 Simulations

typescript

```
// Copy this pattern for each simulation

test('${SimulationName} achieves cognitive objective', async () => {

  const simulation = await load${SimulationName}();

  expect(simulation.teaches${CognitiveSkill}).toBe(true);

  expect(simulation.feelsMystical).toBe(true);

  expect(simulation.maintainsPerformance).toBe(true);

});
```

🚀 AUTOMATED TESTING PIPELINE (Minimal)

yaml

```
# .github/workflows/sacred-validation.yml

name: Sacred Technology Validation


on: [push, pull_request]
```

```yaml
jobs:
  sacred-core:
    runs-on: ubuntu-latest
    steps:
      - name: Sacred Experience Test
        run: npm test -- sacred-experience.test.js
      - name: Performance Test
        run: npm test -- performance.test.js
      - name: Cognitive Development Test
        run: npm test -- cognitive.test.js
      - name: Subscription Flow Test
        run: npm test -- subscription.test.js
      - name: Privacy Test
        run: npm test -- privacy.test.js

  # Only 5 test files total - runs in under 5 minutes
```

📋 DEVELOPMENT WORKFLOW

For Each New Component/Feature:

1. **Build** (80% of time)
   - Code the sacred technology feature
   - Make it visually mystical
   - Ensure 60fps performance
2. **Quick Test** (15% of time)
   - Run the 11-point sacred validation
   - Fix any failures immediately
   - Move on when score > 8
3. **Integration** (5% of time)
   - Ensure works with subscription gating
   - Test offline capability

        - Deploy

Anti-Pattern: What NOT to Do

❌ Writing 100+ tests before building anything
❌ Testing every edge case before core functionality works
❌ Analysis paralysis from over-specification
❌ Perfectionist testing that prevents shipping

Sacred Pattern: What TO Do

✅ Build core functionality first
✅ Test sacred compliance quickly
✅ Ship and iterate based on user feedback
✅ Add tests only when bugs appear

🎯 SUCCESS METRICS

Development Velocity

- **New simulation**: 1-2 weeks max
- **Sacred compliance**: < 1 hour testing per component
- **Bug fixes**: Same day turnaround
- **Feature additions**: Days not weeks

Quality Gates

- **Sacred Score**: > 8/11 for every component
- **Performance**: 60fps on iPhone 12
- **Learning**: Measurable cognitive improvement
- **Conversion**: Free to paid > 10%

🔮 FINAL DIRECTIVE FOR ROO CODE

Keep it simple. Build fast. Stay sacred.

1. **Generate working code first** - prioritize functionality
2. **Apply sacred aesthetics** - crystalline materials, mystical voice
3. **Run the 11-point check** - ensure sacred compliance
4. **Ship when score > 8** - don't perfect endlessly
5. **Iterate based on user feedback** - real users > theoretical tests

The goal is **sacred technology that works**, not perfect tests that prevent shipping. Trust the vision, build rapidly, validate quickly, and let the transformational experience speak for itself.

🔮 **"Perfect is the enemy of sacred. Ship the magic."** 🔮