□ LLM CODING AGENT - EXECUTIVE SUMMARY

Snake Evolution v2.1 Complete Development Package

Created: November 5, 2025
For: LLM Code Generation Agents
Project: Snake Evolution v2.1 (Arcade Game)
Status: ✅ PRODUCTION READY

□ WHAT YOU'RE GETTING

✨ 11 COMPLETE DELIVERABLES

□ Professional Documentation (7 PDFs - 75 Pages)

1. **PRD-v2-1-SnakeEvolution-FINAL.pdf** (11pg) - Business requirements
2. **Analisi-Funzionale-v2-1-FINAL.pdf** (10pg) - Architecture design
3. **Analisi-Tecnica-v2-1-FINAL.pdf** (13pg) - Technical implementation
4. **DevOps-CI-CD-v2-SnakeEvolution.pdf** (10pg) - CI/CD pipeline
5. **Implementation-Guide-v2-SnakeEvolution.pdf** (10pg) - Development setup
6. **Logging-Configuration-v2-SnakeEvolution.pdf** (11pg) - Monitoring
7. **Deployment-Operations-v2-SnakeEvolution.pdf** (10pg) - Production ops

□ LLM Development Guides (4 Markdown Files)

8. LLM-CODING-MASTER-PROMPT.md - Complete development framework
9. **TASK_TRACKER-template.md** - Progress tracking system
10. **REPLIT_GITHUB_WORKFLOW-guide.md** - Daily workflow guide
11. QUICK-START-guide.md - Getting started guide

□ PROJECT OVERVIEW

What You're Building

Snake Evolution - A production-grade arcade game with:

- ✅ Classic Snake gameplay
- ✅ **5-stage evolution system** (unique feature)
- ✅ Persistent leaderboard

- ✅ Audio system
- ✅ 60 FPS performance
- ✅ 85%+ test coverage

Technology Stack

```
Frontend: Vanilla JavaScript (ES6+), Canvas 2D, Web Audio
Build: Webpack 5, Babel
Testing: Jest with property-based testing
Quality: ESLint, Prettier, Stylelint
DevOps: GitHub Actions, Netlify
```

Development Methodology

- **TDD** (Test-Driven Development)
- **Git workflow** (feature branches)
- **Incremental delivery** (31 tasks over 8 weeks)
- **Production-ready** (85%+ coverage, monitoring, CI/CD)

QUICK FACTS

Metric	Value
Total Tasks	31
Estimated Duration	8 weeks
Daily Commits	Expected
Test Coverage Target	85%+
Lighthouse Target	90+
FPS Target	60
Bundle Size Target	< 1MB (gzipped)
Deployment Target	Netlify (automatic)

8-WEEK TIMELINE

```
WEEK 1 ⚙  Setup (5 tasks)
├─ Replit initialization
├─ Webpack &amp; Babel config
├─ Jest testing setup
├─ ESLint &amp; Prettier config
└─ Project structure creation

WEEK 2  Core Game (5 tasks)
```

```
├─ Snake entity
├─ Collision detection (spatial hash O(1))
├─ Grid system
├─ Food spawning
└─ Game loop

WEEK 3 ⭐ Evolution &amp; Input (3 tasks)
├─ 5-stage evolution system
├─ Input validation pipeline
└─ Audio manager

WEEK 4 ▯ UI &amp; Persistence (4 tasks)
├─ Scene manager
├─ Canvas renderer
├─ Storage manager (with checksums)
└─ High score repository

WEEK 5 ▯ Quality (4 tasks)
├─ State machine
├─ Performance profiler
├─ Error boundary
└─ Logger

WEEK 6 ▯ Testing (4 tasks)
├─ Property-based testing
├─ Chaos scenarios
├─ Integration tests
└─ Performance benchmarks

WEEK 7 ▯ Deployment (4 tasks)
├─ Webpack optimization
├─ GitHub Actions CI/CD
├─ Netlify configuration
└─ Monitoring setup

WEEK 8 ▯ Launch (3 tasks)
├─ Final QA
├─ Production deployment
└─ Post-launch support
```

▯ CORE FEATURES

Gameplay Features

- ✅ 4-direction movement (keyboard + touch)
- ✅ Food collection & growth
- ✅ Collision detection (walls + self)
- ✅ **5-stage evolution** with speed multipliers
- ✅ Score tracking & persistence
- ✅ Audio feedback (BGM + SFX)

- ✅ Leaderboard (top 10 local)

Technical Features

- ✅ Spatial hash collision O(1) performance
- ✅ State machine with race condition prevention
- ✅ Input validation pipeline (4 stages)
- ✅ Data integrity (checksums + recovery)
- ✅ Error handling (boundary pattern)
- ✅ Performance profiling (FPS tracking)
- ✅ Structured logging

Quality Features

- ✅ 85%+ test coverage (target)
- ✅ TDD methodology
- ✅ CI/CD automation (GitHub Actions)
- ✅ Automatic deployment (Netlify)
- ✅ 90+ Lighthouse score
- ✅ < 1MB bundle (gzipped)
- ✅ 60 FPS on target devices

HOW TO USE THIS PACKAGE

Step 1: Read the Guides (30 minutes)

1. QUICK-START-guide.md - 5-minute overview
2. LLM-CODING-MASTER-PROMPT.md - Full context
3. **REPLIT_GITHUB_WORKFLOW-guide.md** - Daily routine

Step 2: Setup Environment (15 minutes)

```
git clone https://github.com/yourusername/snake-evolution.git
cd snake-evolution
npm install
npm run dev    # Verify it works
npm test       # Verify tests run
```

Step 3: Start Development (Daily)

- Follow **TDD cycle**: RED → GREEN → REFACTOR
- Reference **TASK_TRACKER.md** for task breakdown
- Update progress daily
- Commit meaningful changes
- Push to GitHub

Step 4: Reference Documentation (As Needed)

- **Architecture questions** → Check Analisi-Funzionale PDF
- **Implementation questions** → Check Analisi-Tecnica PDF
- **Build/DevOps questions** → Check DevOps guide
- **Workflow questions** → Check REPLIT_GITHUB_WORKFLOW guide

✨ HIGHLIGHTS

Innovation

▯ **5-Stage Evolution System** - Unique gameplay mechanic not in original Snake
▯ **Spatial Hash Collision** - O(1) performance for 100+ segment snakes
▯ **Production Architecture** - Enterprise-grade error handling, monitoring, recovery

Quality

⭐ **85%+ Test Coverage** - Comprehensive testing from day 1
⭐ **TDD Methodology** - RED → GREEN → REFACTOR for every feature
⭐ **Enterprise Standards** - Professional software development practices

Automation

▯ **GitHub Actions CI/CD** - Automated testing, building, deployment
▯ **Netlify Deployment** - Automatic deploys on every commit
▯ **Performance Monitoring** - Real-time FPS tracking & alerts

▯ SUCCESS METRICS

By End of Week 8

- ✅ 31/31 tasks completed (100%)
- ✅ 85%+ test coverage
- ✅ 0 blocking bugs
- ✅ Lighthouse score 90+

- ✅ Bundle < 1MB (gzipped)
- ✅ 60 FPS achievable
- ✅ Deployed to production
- ✅ CI/CD fully automated

FILE REFERENCE

Read First

- QUICK-START-guide.md - Overview & immediate setup

Read Before Starting Development

- LLM-CODING-MASTER-PROMPT.md - Complete development framework

Reference During Development

- **TASK_TRACKER.md** - What to build (daily reference)
- **REPLIT_GITHUB_WORKFLOW-guide.md** - How to work (daily reference)
- DELIVERABLES-SUMMARY.md - This summary

Reference for Technical Details

- **Analisi-Funzionale-v2-1-FINAL.pdf** - Architecture & design
- **Analisi-Tecnica-v2-1-FINAL.pdf** - Implementation & algorithms
- **DevOps-CI-CD-v2-SnakeEvolution.pdf** - CI/CD pipeline
- **Other PDFs** - As needed for specific topics

IMMEDIATE NEXT STEPS

Today (30 min)

1. Read QUICK-START-guide.md
2. Read LLM-CODING-MASTER-PROMPT.md overview
3. Setup Replit environment (npm install, npm run dev)

Week 1 (8 hours)

1. Follow Phase 1 tasks
2. Implement project setup
3. Create project structure
4. Make 5 meaningful commits

Weeks 2-8 (50 hours)

1. Follow TDD cycle for each task
2. Write tests first (RED)
3. Implement code (GREEN)
4. Refactor & optimize (REFACTOR)
5. Commit work
6. Update progress tracker
7. Push to GitHub daily

▯ KEY PRINCIPLES

✅ **Always TDD** - Never skip tests
✅ **Always Commit** - After each passing test
✅ **Always Update TASK_TRACKER** - Track progress daily
✅ **Always Push to GitHub** - Daily backup
✅ **Always Test Locally** - Before committing
✅ **Always Review Docs** - When stuck

▯ WHAT YOU'LL LEARN

- ✅ TDD methodology in practice
- ✅ Git workflow with feature branches
- ✅ Professional JavaScript development
- ✅ Game development (physics, collision, rendering)
- ✅ Web Audio API
- ✅ Canvas 2D graphics
- ✅ Testing frameworks (Jest)
- ✅ Build tools (Webpack)
- ✅ CI/CD automation
- ✅ Performance optimization
- ✅ Error handling & recovery
- ✅ Production monitoring

▯ ESTIMATED EFFORT

Phase	Tasks	Hours	Notes
1: Setup	5	8	Foundational setup
2: Core	5	12	Core gameplay
3: Evolution	3	8	Unique features
4: UI	4	10	User interface
5: Quality	4	10	Robustness
6: Testing	4	8	Comprehensive testing
7: Deploy	4	8	DevOps & deployment
8: Launch	3	6	Final polish & launch
Total	**31**	**~70 hours**	**8-10 hours/week**

✅ CHECKLIST FOR SUCCESS

Before Starting:

- [ ] Read QUICK-START-guide.md
- [ ] Read LLM-CODING-MASTER-PROMPT.md
- [ ] Setup Replit environment
- [ ] Clone GitHub repo
- [ ] npm install completed
- [ ] npm run dev works
- [ ] npm test works
- [ ] TASK_TRACKER.md created

During Development (Daily):

- [ ] Follow TDD cycle
- [ ] Write tests first
- [ ] Implement code
- [ ] Run tests locally
- [ ] Commit changes
- [ ] Update TASK_TRACKER
- [ ] Push to GitHub
- [ ] CI/CD passes (green checkmark)

After Each Phase:

- [ ] All tasks completed

- [ ] 85%+ coverage on modules
- [ ] All tests passing
- [ ] 0 linting errors
- [ ] Updated TASK_TRACKER
- [ ] Pushed to GitHub
- [ ] CI/CD pipeline passing

YOU'RE READY!

You now have:
✅ Complete documentation (75+ pages)
✅ Clear 8-week plan (31 tasks)
✅ Development framework (TDD + Git)
✅ Progress tracking (TASK_TRACKER)
✅ Daily guides (Workflow reference)
✅ Code examples (In all docs)

Everything you need is here. Follow the plan, trust the process, implement consistently, and you'll build a professional game in 8 weeks.

Let's go!

SUPPORT RESOURCES

If you're stuck:

1. Check LLM-CODING-MASTER-PROMPT.md - Comprehensive guide
2. Check **REPLIT_GITHUB_WORKFLOW-guide.md** - Common pitfalls
3. Check relevant PDF - Technical details
4. Search code examples in documentation

If you need clarification:

1. Re-read TASK_TRACKER description
2. Check TDD cycle in guides
3. Review similar completed tests
4. Check architectural diagrams in PDFs

Version: 1.0
Created: November 5, 2025
Status: Complete & Ready for Development
Next Step: Read QUICK-START-guide.md and start Phase 1!

 Good luck building Snake Evolution!