

Game Development Document System - Implementation Plan

MVP (Minimum Viable Product) Phase

Goal: Create a basic Chrome extension that can help users create game documents using Claude's conversation capabilities.

Timeline: 4-6 weeks

MVP Features

1. **Basic Chrome Extension Structure** (Week 1)
 - Extension manifest and permissions setup
 - Basic sidebar UI that appears on Claude.ai
 - Project and document list views
 - Storage infrastructure for saving documents
2. **Core Document Creation Flow** (Week 2)
 - Implement first document type (Game Vision)
 - Basic prompt injection for Claude
 - Document draft capture and saving
 - Simple document viewing
3. **Multi-Document Support** (Week 3)
 - Add support for Core Game Concept and Target Audience documents
 - Implement document relationships (previous doc context)
 - Document status tracking (not started, in progress, completed)
4. **Essential UI Improvements** (Week 4)
 - Document export to Markdown and PDF
 - Basic document editing/refinement flow
 - Improved navigation and document list UI
 - Error handling and recovery flows

MVP Testing Milestones

- **Internal Testing** (Week 5)
 - Bug fixes and performance improvements
 - User flow optimization

- Edge case handling
- **Limited User Testing** (Week 6)
 - Distribute to small group of game developers
 - Gather feedback on core functionality
 - Identify priority improvements for next phase

Phase 2: Enhanced Functionality

Goal: Expand document types and improve user experience with more seamless integration.

Timeline: 8 weeks after MVP

Phase 2 Features

1. **Complete Document Set** (Weeks 1-2)
 - Implement remaining document types
 - Improve context generation between documents
 - Add document templates and examples
2. **Google Drive Integration** (Weeks 3-4)
 - Authenticate with Google Drive
 - Save/sync documents to cloud storage
 - Share documents via Drive
3. **Improved Capture & Intelligence** (Weeks 5-6)
 - Better document content extraction from Claude conversations
 - Smarter detection of document completion
 - Content validation and suggestion features
4. **Enhanced UI/UX** (Weeks 7-8)
 - Improved document creation wizard
 - Visual styling enhancements
 - Onboarding experience
 - Keyboard shortcuts and productivity features

Phase 3: Advanced Features & Polish

Goal: Add advanced capabilities and prepare for wider distribution.

Timeline: 12 weeks after Phase 2

Phase 3 Features

1. **Document Versioning & History** (Weeks 1-3)
 - Track document versions
 - Compare and restore previous versions
 - Document change logs
2. **Export Enhancements** (Weeks 4-6)
 - Multiple export formats (HTML, Google Docs, etc.)
 - Custom styling and formatting options
 - Batch export capabilities
3. **Collaboration Features** (Weeks 7-9)
 - Share projects with team members
 - Comment and feedback system
 - Real-time updates and notifications
4. **Advanced Document Analysis** (Weeks 10-12)
 - Consistency checking across documents
 - Project health assessment
 - Suggestions for improvement
 - Document quality metrics

Tracking & Management

Key Metrics to Track

1. **Development Progress**
 - Features completed vs. planned
 - Bug count and resolution rate
 - Test coverage
2. **User Metrics** (Post-MVP)
 - Number of active users
 - Documents created per user
 - Completion rate of document suites
 - Export frequency
3. **Performance Metrics**

- Extension load time
- Response time for key actions
- Storage usage

Implementation Checklist - MVP

☐ Setup Development Environment

- ☐ Initialize project with TypeScript, Webpack
- ☐ Create Chrome extension manifest
- ☐ Setup GitHub repository with CI/CD

☐ Core Infrastructure

- ☐ Implement storage manager
- ☐ Create document type definitions
- ☐ Setup message passing between components

☐ UI Components

- ☐ Design and implement sidebar UI
- ☐ Create project management interface
- ☐ Build document creation flow
- ☐ Implement document viewing interface

☐ Claude Integration

- ☐ Build context injection system
- ☐ Create response monitoring and capture
- ☐ Implement document content extraction

☐ Testing & Deployment

- ☐ Create automated tests
- ☐ Perform manual testing on Claude.ai
- ☐ Package for distribution
- ☐ Create installation guide

Risk Assessment

1. Technical Risks

- **Claude.ai DOM Changes:** Continuous monitoring and quick updates if Claude's interface changes
- **Storage Limitations:** Plan for efficient storage and implement cloud sync early if needed
- **Context Window Limitations:** Optimize prompt size and document context sharing

2. User Experience Risks

- **Learning Curve:** Focus on intuitive design and clear onboarding

- **Workflow Disruption:** Ensure extension enhances rather than interrupts the conversation flow
- **Content Extraction Accuracy:** Develop robust algorithms for identifying document content

3. Business Risks

- **Chrome Extensions Policy Changes:** Monitor updates to Chrome Web Store policies
- **Claude API Changes:** Build flexible integration that can adapt to API updates
- **Similar Product Competition:** Focus on unique features and superior UX

Resources Required

1. Development Team

- 1 Frontend Developer (Chrome extension, UI)
- 1 Backend Developer (storage, Google Drive integration)
- 1 Game Design Consultant (document structure and prompts)

2. Testing Resources

- Small group of game developers for beta testing
- QA testing for extension functionality

3. Infrastructure

- Chrome Developer Account
- Google Cloud Platform (for Drive API)
- GitHub repository
- Testing devices

Success Criteria

The MVP will be considered successful if:

1. Users can successfully create, view, and export at least 3 document types
2. Document content is accurately captured from Claude conversations
3. Projects and documents are correctly saved and retrievable
4. The extension works reliably across Chrome versions
5. Initial user feedback indicates value and improved workflow

Next Steps To Begin Implementation

1. Immediate (Next 1-2 days)

- Create GitHub repository

- Setup development environment
- Create initial project structure

2. **Short-term (First week)**

- Implement basic DOM injection for sidebar
- Create storage infrastructure
- Begin UI component development

3. **Medium-term (First 2 weeks)**

- Complete basic document creation flow for Game Vision
- Implement Claude context injection
- Test document storage and retrieval