

Offline 2D Gamified Classroom - Project Documentation

1. Executive Summary

This document provides a comprehensive project plan for developing an offline 2D gamified classroom application. The system allows students to navigate a virtual school environment, access educational resources, complete quizzes, and communicate with peers - all while functioning completely offline with optional online synchronization.

2. Project Overview

2.1 Vision

Create an engaging, accessible educational platform that functions in low-connectivity environments while providing an immersive learning experience through gamification elements.

2.2 Core Features

- 2D top-down tile-based map of a school with multiple room types
- Player avatar movement with collision detection
- Interactive educational resources (books, videos, quizzes)
- Local chat system for offline messaging
- Complete offline functionality with local storage
- Optional synchronization when online connectivity is available

2.3 Target Users

- Students in areas with limited internet connectivity
- Educational institutions with bandwidth constraints
- Teachers looking for gamified classroom experiences
- Homeschooling environments

3. Technical Architecture

3.1 Technology Stack

- **Game Engine:** Phaser.js (JavaScript)
- **Offline Storage:** IndexedDB via localForage.js
- **Asset Packaging:** Service Worker for caching
- **Chat System:** Local queue (offline) + WebSocket (when online)
- **UI Components:** HTML/CSS overlays + Phaser UI plugins
- **Build Tools:** Node.js, npm, webpack/Vite

3.2 Component Architecture

- **Core Game Engine:** Manages scenes, physics, rendering
- **Data Manager:** Handles persistence, sync, and state
- **Resource System:** Loads and displays educational content
- **Player System:** Controls avatar, interactions, inventory
- **UI Layer:** Provides interface elements and overlays
- **Chat System:** Manages communication between users
- **Offline Manager:** Ensures functionality without connectivity

3.3 Data Flow

- User interactions are processed by the game engine
- Game state is persisted to IndexedDB via the Storage Manager
- Educational resources are cached locally via Service Worker
- Progress data is stored locally and synced when online
- Chat messages are queued locally and synced when online

4. Development Roadmap

4.1 Phase 1: Foundation (Weeks 1-2)

- Project setup and configuration
- Core game engine implementation
- Offline storage infrastructure
- Basic player movement system

4.2 Phase 2: Core Mechanics (Weeks 3-4)

- School map implementation
- Room transitions and navigation
- Interactive object framework
- Collision system and physics

4.3 Phase 3: Educational Elements (Weeks 5-6)

- Resource management system
- Quiz implementation
- Progress tracking
- Content authoring tools

4.4 Phase 4: User Experience (Weeks 7-8)

- UI design and implementation

- Chat system development
- Achievement system
- Tutorial and help system

4.5 Phase 5: Refinement (Weeks 9-10)

- Online synchronization
- Performance optimization
- User testing and feedback
- Bug fixes and improvements

4.6 Phase 6: Deployment (Weeks 11-12)

- Final testing and QA
- Documentation development
- Production build pipeline
- Release preparation

5. Technical Specifications

5.1 Game Map System

- Multiple tilemap layers for background, collision, and interactivity
- Room definitions with unique properties and resource associations
- Dynamic loading/unloading of areas for performance
- Navigation system with pathfinding for NPCs

5.2 Resource System

- Support for multiple resource types (text, video, quiz, interactive)
- Metadata structure for tracking completion and progress
- Caching strategy for offline availability
- Version control for content updates

5.3 Player System

- Avatar customization options
- Inventory management
- Progress tracking across multiple courses/subjects
- Achievement system with rewards

5.4 Data Storage

- User profile structure

- Resource metadata schema
- Progress tracking data model
- Chat message format
- Conflict resolution for sync operations

5.5 Offline Functionality

- Service Worker cache management
- Asset preloading strategy
- Storage quota management
- Sync queue for offline actions

6. User Experience Design

6.1 Navigation Flow

- Initial onboarding and tutorial
- Main menu to school map navigation
- Room selection and exploration
- Resource discovery and interaction
- Progress visualization and feedback

6.2 Interface Design

- Clean, intuitive UI suitable for educational context
- Age-appropriate visual styling
- Accessibility considerations
- Responsive design for various devices

6.3 Feedback Systems

- Visual cues for interactive elements
- Progress indicators for resource completion
- Achievement notifications
- Quiz results and feedback

7. Content Management

7.1 Content Structure

- Course organization hierarchy
- Lesson planning framework
- Resource categorization

- Quiz construction guidelines

7.2 Content Authoring

- Resource creation workflow
- Content validation rules
- Metadata requirements
- Import/export functionality

7.3 Content Deployment

- Packaging for offline distribution
- Update mechanism for new content
- Version control and staging

8. Testing Strategy

8.1 Functional Testing

- Core gameplay mechanics
- Educational resource functionality
- Storage and persistence
- Offline capabilities

8.2 Performance Testing

- Loading times and optimization
- Memory usage monitoring
- Battery impact assessment
- Storage utilization

8.3 User Testing

- Usability evaluation
- Educational effectiveness
- Engagement metrics
- Feedback collection

9. Deployment Strategy

9.1 Distribution Methods

- Web application (PWA)
- Packaged desktop application
- Mobile application options

- Offline distribution mechanisms

9.2 Update Management

- Version control system
- Update notification flow
- Minimal update packages
- Staged rollout strategy

9.3 Maintenance Plan

- Monitoring system
- Support infrastructure
- Bug tracking process
- Feature request handling

10. Risk Assessment

10.1 Technical Risks

- Browser compatibility issues
- Storage limitations on devices
- Performance on low-end hardware
- Sync conflicts with server

10.2 Project Risks

- Scope creep in educational features
- Content creation bottlenecks
- User adoption challenges
- Integration with existing systems

10.3 Mitigation Strategies

- Progressive enhancement approach
- Early prototype testing
- Regular stakeholder reviews
- Modular development approach