**Truffle Hunter Tycoon - Project Handoff Document**

**Project Overview**

**Truffle Hunter Tycoon** is a web-based idle/auto-clicker game where players manage pigs that hunt for truffles in various forest locations. The game features progression through upgrades, location unlocks, pig retirement, and achievement systems.

**Target Audience**: Casual gamers who enjoy idle/incremental games  
**Platform**: Web browser (HTML5/CSS/JavaScript)  
**Development Goal**: Rapid prototyping with clean, maintainable code

**Game Concept Summary**

Players control a pig that automatically walks through side-scrolling forest environments. Dirt mounds appear every 3-5 seconds that players can click to make their pig dig for truffles. Found truffles auto-sell for gold, which funds upgrades and location unlocks. The core progression loop involves upgrading pig abilities, unlocking new locations with rarer truffles, and retiring pigs for permanent bonuses.

**Victory Condition**: Accumulate $1,000,000 (estimated 90-120 minutes gameplay)

**Core Mechanics**

**Primary Gameplay Loop**

1. **Auto-Walking Pig**: Pig walks automatically through scrolling forest background
2. **Click to Dig**: Click dirt mounds (spawn every 3-5 seconds) to search for truffles
3. **Auto-Collection**: Found truffles float to collection basket and auto-sell for gold
4. **Upgrade System**: Spend gold on pig improvements (sniffing power, luck, speed)
5. **Location Progression**: Unlock new areas with better truffles using gold + reputation
6. **Retirement System**: Retire pigs (100 truffles + 1 legendary) for permanent bonuses

**Resource System**

* **Gold**: Primary currency from truffle sales + passive income from retired pigs
* **Reputation**: Secondary currency from finding truffles, required for location unlocks

**Progression Mechanics**

* **Pig Upgrades**: Sniffing Power (success rate), Luck (rare truffle chance), Speed (walking speed)
* **Location Unlocks**: 4 areas with increasingly valuable and rare truffles
* **Retirement Benefits**: Each retired pig provides 2% price bonus + 60 gold/minute passive income

**Technical Architecture**

**Technology Stack**

* **Frontend**: HTML5, CSS3, JavaScript (ES6 modules)
* **Animations**: GreenSock Animation Platform (GSAP)
* **Architecture**: MVC pattern with component-based design
* **Storage**: Browser localStorage for save data

**File Structure Philosophy**

* **Single Responsibility**: Each class in its own file
* **Separation of Concerns**: Models (data), Views (UI), Controllers (logic)
* **Component-Based CSS**: Modular stylesheets matching JavaScript components
* **Configuration-Driven**: Game balance in centralized config files

**Current Development Status**

**Completed Design Phase**

✅ **Game Design Document**: Complete mechanics specification  
✅ **Balancing Tables**: All numerical values defined and tested via simulation  
✅ **File Structure**: Complete OOP architecture planned  
✅ **Architecture Guide**: Detailed implementation strategy

**Ready for Implementation**

* Core game loop and mechanics fully specified
* All upgrade costs, truffle values, and spawn rates calculated
* UI/UX flow completely designed
* Technical architecture documented

**Key Design Decisions Made**

**Game Balance**

* **Progression Curve**: Smooth 90-120 minute experience to victory
* **First Retirement**: Achievable at ~15 minutes to introduce prestige system early
* **Location Unlocks**: New area every 30 minutes to maintain engagement
* **No Monetization**: Pure gameplay experience without pay-to-win mechanics

**Technical Choices**

* **No Weather System**: Removed to simplify initial implementation
* **No Offline Progress**: Game pauses when closed (can be added later)
* **Single Platform**: Web-first development, mobile-responsive design
* **Modular Design**: Easy to add/modify features without major refactoring

**User Experience**

* **Visual Style**: Side-scrolling with charming pig animations
* **Audio Design**: Forest ambiance + satisfying sound effects for actions
* **Accessibility**: Clear visual feedback, keyboard support planned
* **Tutorial**: Text-based initial tutorial, interactive tutorial planned for later

**Implementation Priority Order**

**Phase 1: Core Mechanics (MVP)**

1. Basic pig walking and dirt mound clicking
2. Truffle spawning and collection system
3. Gold accumulation and upgrade purchasing
4. Simple upgrade effects (sniffing power, speed)

**Phase 2: Progression Systems**

1. Location unlocking and switching
2. Reputation system and achievement tracking
3. Pig retirement and prestige benefits
4. Save/load functionality

**Phase 3: Polish and Enhancement**

1. GreenSock animations and visual effects
2. Sound system and audio feedback
3. UI polish and responsive design
4. Achievement notifications and celebrations

**Critical Implementation Notes**

**Must-Have Features**

* **Retirement System**: Essential for long-term progression and retention
* **Achievement Tracking**: Drives engagement and provides goals
* **Save System**: Players expect progress persistence
* **Responsive Design**: Mobile-friendly interface crucial

**Performance Considerations**

* **Efficient Rendering**: Use CSS transforms for animations when possible
* **Memory Management**: Clean up unused DOM elements and animations
* **Asset Loading**: Optimize images and implement progressive loading

**Extensibility Hooks**

* **Event System**: Implement custom events for easy feature additions
* **Plugin Architecture**: Design upgrade system to easily add new upgrade types
* **Data-Driven**: Keep all game balance in configuration files for easy tweaking

**Known Technical Challenges**

**Animation Complexity**

* **Pig State Management**: Walking, digging, celebrating animations need smooth transitions
* **Particle Effects**: Truffle discovery effects require efficient particle systems
* **Background Scrolling**: Parallax scrolling with performance optimization

**Game Balance Iteration**

* **Playtesting Required**: Numerical balance needs real-world validation
* **Progression Tuning**: May need adjustment based on actual play patterns
* **Achievement Difficulty**: Balance between too easy and too hard

**Next Steps for Implementation**

1. **Set Up Project Structure**: Create file hierarchy and basic HTML/CSS framework
2. **Implement Core Game Class**: Start with Game.js and basic state management
3. **Create Pig Entity**: Basic pig with movement and upgrade tracking
4. **Build Click Handler**: Dirt mound detection and truffle generation
5. **Add UI Components**: Gold display, upgrade buttons, basic interface

**Success Metrics**

**Technical Success**

* **Code Quality**: Clean, maintainable, well-documented code
* **Performance**: 60fps gameplay with smooth animations
* **Compatibility**: Works across modern browsers (Chrome, Firefox, Safari, Edge)

**Gameplay Success**

* **Progression Feel**: Smooth, rewarding advancement curve
* **Engagement**: Players reach first retirement within 15 minutes
* **Retention**: Compelling enough to play to victory condition (90-120 min)

**Additional Resources**

The following documents should be referenced during development:

1. **Truffle Hunter Tycoon - Complete Game Design Document**: Full gameplay mechanics
2. **Truffle Hunter Tycoon - Balancing Numbers Table**: All numerical values and progression curves
3. **Truffle Hunter Tycoon - File Structure**: Complete technical architecture
4. **Key Files and Architecture Explanation**: Implementation guidance and code examples

**Contact and Iteration**

This design represents a complete, implementable game specification. However, game development is iterative - expect to refine mechanics, adjust balance, and enhance features based on playtesting and development discoveries.

The modular architecture is specifically designed to support this iteration process while maintaining code quality and development velocity.