Comparative Analysis Report

Archery Game Implementation Evaluation

Introduction

This report compares two JavaScript archery game implementations: an **Original Version** with structured, professional code architecture, and a **Simplified Version** designed for quick deployment. While the simplified version appears easier, it is functionally and architecturally inferior.

Key Comparison Areas

Aspect	Original (Better)	Simplified (Worse)
Code Structure	Clear function separation, predictable state management	Tightly coupled logic, global intervals
Game Area	Fixed container with boundary control	Full-screen, unpredictable layout behavior
Timing Control	Stored interval references, proper cleanup	No cleanup, relies on page reload
Difficulty Scaling	Gradual increase (every 3 hits)	Increases every hit, no upper limit
Restart Mechanism	Programmatic state reset	location.reload() - bad practice
User Interaction	Event listeners, logical separation	Inline onclick handlers, less flexible

Critical Issues with Simplified Version

• **Poor Resource Management:** setInterval() without cleanup causes memory leaks.

• **Unpredictable Behavior:** Objects may appear outside visible screen on different resolutions.

• **Non-scalable Difficulty:** Game becomes unplayable quickly with no speed cap.

• **Bad Restart Practice:** Page reload instead of proper state reset shows weak state management.

• **Limited Extensibility:** Tight coupling makes adding features (levels, sound, pause) difficult.

Conclusion

The **original implementation** demonstrates superior software engineering principles including proper separation of concerns, safe resource management, predictable gameplay, and high maintainability. The simplified version sacrifices **robustness, clarity, and best practices** for brevity, making it unsuitable for production or academic submissions.

Recommendation: Use the original structured implementation for any serious development, teaching, or portfolio work.