Code Review Report: Web Casual Games

Date: 2026-02-03 **Reviewer:** AI Assistant

1. Executive Summary

The project contains three games: **Cat Café Tycoon**, **Neon Snake 2077**, and **Space Invaders**.Overall, the code is structured well for prototyping (single-file components) but has room for improved modularity.
Critical issues regarding mobile responsiveness and collision logic in Space Invaders and Neon Snake have been fixed during this review.

2. Changes Implemented (Fixes)

Space Invaders

Fixed Mobile Responsiveness:

Changed width from fixed to (max 600px).

Set Canvas to scale automatically () while maintaining internal resolution.

Improved button layout for mobile touch targets.

Fixed Logic Mismatch:

Standardized invader dimensions. Previously, logic used hardcoded for collision/draw, while spawn logic used .

Updated and functions to use consistent (40px) and (30px).

Unified collision detection to use these properties instead of magic numbers.

Neon Snake 2077

Fixed Layout Scaling:

Updated CSS from (square based on width) to (square based on smallest viewport dimension).

This prevents the game container from overflowing or getting cropped on landscape mobile screens.

3. Detailed Review & Suggestions

Cat Café Tycoon

Quality: Good use of for state persistence.

Performance: DOM element pooling (limiting cats/staff) is a good optimization.

Suggestion:

Move large SVG data or image paths to a configuration object or separate file to clean up the code.

The runs every frame but logic only triggers every 15s. Consider throttling the loop or using a separate timer for income to save battery on mobile.

Neon Snake 2077

Quality: Clean implementation of the game loop with fixed time step.

Visuals: "Phase Dash" mechanic and particle effects are well implemented.

Suggestion:

Add a "Pause" button.

The function is currently static (600x600). While CSS scaling handles the display, allowing dynamic resolution could make the game sharper on high-DPI displays.

Space Invaders

Quality: Functional classic arcade logic.

Audio: Good use of HTML5 Audio.

Suggestion:

Preloading: The game starts immediately, but images might not be loaded. Added safeguards in , but a proper "Loading..." screen would be better.

Code Structure: There is significant repetition in collision detection loops (Player Bullet -> Invader, Player Bullet -> UFO, Invader Bullet -> Player). A helper function would reduce code size and errors.

4. Next Steps

Refactor: Separate CSS, JS, and HTML into distinct files for better maintainability.

Asset Management: Implement a simple asset loader to ensure all images/sounds are ready before game start.

PWA: Add a and service worker to allow these games to be installed as offline apps on mobile.