<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no, viewport-fit=cover">

<meta name="theme-color" content="#0a0a2e">

<meta name="apple-mobile-web-app-capable" content="yes">

<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">

<title>Firefly Catcher</title>

<style>

body {

margin: 0;

padding: 0;

background: linear-gradient(to bottom, #0a0a2e, #16213e, #1a1a3a);

font-family: 'Arial', sans-serif;

color: #fff;

overflow: hidden;

cursor: crosshair;

height: 100vh;

height: 100dvh;

-webkit-user-select: none;

-webkit-tap-highlight-color: transparent;

-webkit-touch-callout: none;

}

.game-container {

position: relative;

width: 100vw;

height: 100vh;

height: 100dvh;

}

.firefly {

position: absolute;

border-radius: 50%;

cursor: pointer;

transition: all 0.3s ease;

padding: 20px;

margin: -20px;

}

.firefly::before {

content: '';

position: absolute;

top: -15px;

left: -15px;

right: -15px;

bottom: -15px;

background: transparent;

border-radius: 50%;

}

.firefly:hover {

transform: scale(1.3);

}

.firefly-common {

width: 8px;

height: 8px;

background: radial-gradient(circle, #ffff80, #ffff40, transparent);

box-shadow: 0 0 15px #ffff80, 0 0 25px #ffff40, 0 0 35px #ffff20;

animation: glow-common 2s ease-in-out infinite alternate;

}

.firefly-uncommon {

width: 10px;

height: 10px;

background: radial-gradient(circle, #80ff80, #40ff40, transparent);

box-shadow: 0 0 18px #80ff80, 0 0 28px #40ff40, 0 0 38px #20ff20;

animation: glow-uncommon 1.8s ease-in-out infinite alternate;

}

.firefly-rare {

width: 12px;

height: 12px;

background: radial-gradient(circle, #8080ff, #4040ff, transparent);

box-shadow: 0 0 20px #8080ff, 0 0 30px #4040ff, 0 0 40px #2020ff;

animation: glow-rare 1.5s ease-in-out infinite alternate;

}

.firefly-epic {

width: 14px;

height: 14px;

background: radial-gradient(circle, #ff80ff, #ff40ff, transparent);

box-shadow: 0 0 25px #ff80ff, 0 0 35px #ff40ff, 0 0 45px #ff20ff;

animation: glow-epic 1.2s ease-in-out infinite alternate;

}

.firefly-legendary {

width: 16px;

height: 16px;

background: radial-gradient(circle, #ffd700, #ffaa00, transparent);

box-shadow: 0 0 30px #ffd700, 0 0 40px #ffaa00, 0 0 50px #ff8800;

animation: glow-legendary 1s ease-in-out infinite alternate;

}

.firefly-bonus {

width: 24px;

height: 24px;

background: radial-gradient(circle, #ff0080, #ff4080, #ff0040, transparent);

box-shadow: 0 0 40px #ff0080, 0 0 60px #ff4080, 0 0 80px #ff0040;

animation: glow-bonus 0.3s ease-in-out infinite alternate;

border: 2px solid #fff;

z-index: 1000;

}

.firefly-super-legendary {

width: 20px;

height: 20px;

background: radial-gradient(circle, #ff69b4, #ff1493, #dc143c, transparent);

box-shadow: 0 0 35px #ff69b4, 0 0 50px #ff1493, 0 0 65px #dc143c, 0 0 80px #b22222;

animation: glow-super-legendary 0.8s ease-in-out infinite alternate;

border: 3px solid #ffffff;

z-index: 1001;

}

@keyframes glow-common {

0% { opacity: 0.6; transform: scale(1); }

100% { opacity: 1; transform: scale(1.1); }

}

@keyframes glow-uncommon {

0% { opacity: 0.6; transform: scale(1); }

100% { opacity: 1; transform: scale(1.15); }

}

@keyframes glow-rare {

0% { opacity: 0.6; transform: scale(1); }

100% { opacity: 1; transform: scale(1.2); }

}

@keyframes glow-epic {

0% { opacity: 0.6; transform: scale(1); }

100% { opacity: 1; transform: scale(1.25); }

}

@keyframes glow-legendary {

0% { opacity: 0.6; transform: scale(1) rotate(0deg); }

100% { opacity: 1; transform: scale(1.3) rotate(180deg); }

}

@keyframes glow-bonus {

0% { opacity: 0.8; transform: scale(1) rotate(0deg); }

100% { opacity: 1; transform: scale(1.5) rotate(360deg); }

}

@keyframes glow-super-legendary {

0% { opacity: 0.7; transform: scale(1) rotate(0deg); }

50% { opacity: 1; transform: scale(1.4) rotate(180deg); }

100% { opacity: 0.9; transform: scale(1.2) rotate(360deg); }

}

.caught-firefly {

animation: catch-animation 0.8s ease-out forwards;

}

@keyframes catch-animation {

0% { transform: scale(1); opacity: 1; }

50% { transform: scale(2); opacity: 0.8; }

100% { transform: scale(0); opacity: 0; }

}

.value-popup {

position: absolute;

font-size: 18px;

font-weight: bold;

color: #ffff80;

text-shadow: 2px 2px 4px rgba(0,0,0,0.8);

animation: popup 1s ease-out forwards;

pointer-events: none;

}

@keyframes popup {

0% { transform: translateY(0) scale(1); opacity: 1; }

100% { transform: translateY(-50px) scale(1.2); opacity: 0; }

}

.jar-container {

position: fixed;

bottom: calc(env(safe-area-inset-bottom, 20px) + 20px);

right: env(safe-area-inset-right, 20px);

width: 120px;

height: 140px;

}

.jar {

width: 100px;

height: 120px;

background: linear-gradient(to bottom, transparent 20%, rgba(200, 200, 255, 0.1) 25%, rgba(200, 200, 255, 0.2) 100%);

border: 3px solid #888;

border-radius: 0 0 40px 40px;

position: relative;

overflow: hidden;

}

.jar-lid {

width: 110px;

height: 15px;

background: #666;

border-radius: 10px;

position: absolute;

top: -8px;

left: -5px;

border: 2px solid #444;

}

.jar-firefly {

position: absolute;

width: 4px;

height: 4px;

border-radius: 50%;

animation: jar-glow 1.5s ease-in-out infinite alternate;

}

.jar-firefly.common {

background: #ffff80;

box-shadow: 0 0 8px #ffff80, 0 0 12px #ffff40;

}

.jar-firefly.uncommon {

background: #80ff80;

box-shadow: 0 0 8px #80ff80, 0 0 12px #40ff40;

}

.jar-firefly.rare {

background: #8080ff;

box-shadow: 0 0 8px #8080ff, 0 0 12px #4040ff;

}

.jar-firefly.epic {

background: #ff80ff;

box-shadow: 0 0 8px #ff80ff, 0 0 12px #ff40ff;

}

.jar-firefly.legendary {

background: #ffd700;

box-shadow: 0 0 8px #ffd700, 0 0 12px #ffaa00;

}

.jar-firefly.bonus {

background: #ff0080;

box-shadow: 0 0 12px #ff0080, 0 0 16px #ff4080;

width: 6px;

height: 6px;

}

.jar-firefly.super-legendary {

background: #ff69b4;

box-shadow: 0 0 15px #ff69b4, 0 0 20px #ff1493;

width: 7px;

height: 7px;

}

@keyframes jar-glow {

0% { opacity: 0.4; }

100% { opacity: 1; }

}

.ui-panel {

position: fixed;

top: env(safe-area-inset-top, 20px);

left: env(safe-area-inset-left, 20px);

background: rgba(0, 0, 0, 0.8);

padding: 12px;

border-radius: 10px;

border: 2px solid #555;

font-size: 14px;

backdrop-filter: blur(10px);

}

.ui-row {

display: flex;

justify-content: space-between;

margin: 5px 0;

min-width: 200px;

}

.coins {

color: #ffd700;

font-weight: bold;

}

.collected {

color: #9C27B0;

font-weight: bold;

}

.sell-button {

background: #4CAF50;

color: white;

border: none;

padding: 8px 16px;

border-radius: 5px;

cursor: pointer;

font-size: 14px;

margin-top: 10px;

transition: background 0.3s ease;

}

.sell-button:hover {

background: #45a049;

}

.sell-button:disabled {

background: #666;

cursor: not-allowed;

}

.scrapbook-button {

background: #9C27B0;

color: white;

border: none;

padding: 8px 16px;

border-radius: 5px;

cursor: pointer;

font-size: 14px;

margin-top: 5px;

transition: background 0.3s ease;

width: 100%;

}

.scrapbook-button:hover {

background: #7B1FA2;

}

.save-button {

background: #4CAF50;

color: white;

border: none;

padding: 6px 12px;

border-radius: 5px;

cursor: pointer;

font-size: 12px;

margin-top: 3px;

transition: background 0.3s ease;

width: 100%;

}

.save-button:hover {

background: #45a049;

}

.export-button {

background: #FF9800;

color: white;

border: none;

padding: 6px 12px;

border-radius: 5px;

cursor: pointer;

font-size: 12px;

margin-top: 3px;

transition: background 0.3s ease;

width: 100%;

}

.export-button:hover {

background: #F57C00;

}

.save-status {

font-size: 11px;

text-align: center;

margin: 5px 0;

padding: 4px;

border-radius: 4px;

background: rgba(76, 175, 80, 0.2);

border: 1px solid #4CAF50;

transition: all 0.3s ease;

}

.save-status.saving {

background: rgba(255, 193, 7, 0.2);

border-color: #FFC107;

color: #FFC107;

}

.save-status.error {

background: rgba(244, 67, 54, 0.2);

border-color: #f44336;

color: #f44336;

}

.upgrade-panel {

position: fixed;

top: calc(env(safe-area-inset-top, 20px) + 200px);

left: env(safe-area-inset-left, 20px);

background: rgba(0, 0, 0, 0.8);

padding: 12px;

border-radius: 10px;

border: 2px solid #555;

font-size: 12px;

max-width: 200px;

backdrop-filter: blur(10px);

}

.timer-panel {

position: fixed;

top: calc(env(safe-area-inset-top, 20px) + 400px);

left: env(safe-area-inset-left, 20px);

background: rgba(0, 0, 0, 0.8);

padding: 12px;

border-radius: 10px;

border: 2px solid #555;

font-size: 12px;

max-width: 200px;

backdrop-filter: blur(10px);

}

.timer-row {

display: flex;

justify-content: space-between;

margin: 5px 0;

min-width: 150px;

}

.upgrade-button {

background: #2196F3;

color: white;

border: none;

padding: 6px 12px;

border-radius: 5px;

cursor: pointer;

font-size: 12px;

margin: 2px;

transition: background 0.3s ease;

width: 100%;

}

.upgrade-button:hover {

background: #1976D2;

}

.upgrade-button:disabled {

background: #666;

cursor: not-allowed;

}

.stars {

position: absolute;

width: 2px;

height: 2px;

background: white;

border-radius: 50%;

animation: twinkle 3s ease-in-out infinite;

}

@keyframes twinkle {

0%, 100% { opacity: 0.3; }

50% { opacity: 1; }

}

.grass {

position: absolute;

bottom: 0;

width: 100%;

height: 60px;

background: linear-gradient(to top, #1a4c1a, #2d5a2d, transparent);

}

.rarity-info {

font-size: 11px;

margin-top: 10px;

opacity: 0.8;

}

.scrapbook-modal {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.9);

z-index: 10000;

overflow-y: auto;

}

.scrapbook-content {

max-width: 900px;

margin: 20px auto;

background: linear-gradient(135deg, #1a1a3a, #2a2a5a);

border-radius: 15px;

border: 3px solid #9C27B0;

padding: 20px;

color: white;

}

.scrapbook-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

border-bottom: 2px solid #9C27B0;

padding-bottom: 10px;

}

.scrapbook-header h2 {

margin: 0;

color: #9C27B0;

text-shadow: 2px 2px 4px rgba(0,0,0,0.8);

}

.scrapbook-header div {

display: flex;

gap: 10px;

}

.import-button {

background: #2196F3;

color: white;

border: none;

padding: 8px 12px;

border-radius: 5px;

cursor: pointer;

font-size: 14px;

font-weight: bold;

}

.import-button:hover {

background: #1976D2;

}

.close-button {

background: #f44336;

color: white;

border: none;

padding: 8px 12px;

border-radius: 50%;

cursor: pointer;

font-size: 16px;

font-weight: bold;

}

.close-button:hover {

background: #d32f2f;

}

.scrapbook-stats {

display: flex;

justify-content: space-around;

margin-bottom: 20px;

font-size: 18px;

font-weight: bold;

color: #ffd700;

}

.scrapbook-tabs {

display: flex;

flex-wrap: wrap;

gap: 5px;

margin-bottom: 15px;

justify-content: center;

}

.tab-button {

background: rgba(255, 255, 255, 0.1);

color: white;

border: 2px solid #9C27B0;

padding: 8px 12px;

border-radius: 5px;

cursor: pointer;

font-size: 12px;

transition: all 0.3s ease;

}

.tab-button:hover {

background: rgba(156, 39, 176, 0.3);

}

.tab-button.active {

background: #9C27B0;

color: white;

font-weight: bold;

}

.scrapbook-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));

gap: 15px;

}

.firefly-card {

background: rgba(255, 255, 255, 0.1);

border: 2px solid #9C27B0;

border-radius: 10px;

padding: 15px;

backdrop-filter: blur(10px);

transition: transform 0.3s ease;

}

.firefly-card:hover {

transform: scale(1.05);

border-color: #ffd700;

}

.firefly-card-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 10px;

}

.firefly-name {

font-size: 20px;

font-weight: bold;

color: #ffd700;

}

.firefly-rarity {

padding: 4px 8px;

border-radius: 15px;

font-size: 12px;

font-weight: bold;

}

.rarity-common { background: #ffff80; color: #333; }

.rarity-uncommon { background: #80ff80; color: #333; }

.rarity-rare { background: #8080ff; color: white; }

.rarity-epic { background: #ff80ff; color: white; }

.rarity-legendary { background: #ffd700; color: #333; }

.rarity-super-legendary { background: #ff69b4; color: white; }

.rarity-bonus { background: #ff0080; color: white; }

.firefly-theme {

font-style: italic;

color: #9C27B0;

margin-bottom: 8px;

}

.firefly-bio {

font-size: 14px;

line-height: 1.4;

margin-bottom: 8px;

}

.firefly-caught {

font-size: 12px;

color: #888;

text-align: right;

}

/\* Mobile optimizations \*/

@media (max-width: 768px) {

.ui-panel, .upgrade-panel, .timer-panel {

font-size: 12px;

padding: 10px;

}

.ui-row, .timer-row {

min-width: 150px;

}

.rarity-info {

font-size: 10px;

}

}

</style>

</head>

<body>

<div class="game-container">

<!-- Stars background -->

<div class="stars" style="top: 10%; left: 15%; animation-delay: 0s;"></div>

<div class="stars" style="top: 20%; left: 80%; animation-delay: 1s;"></div>

<div class="stars" style="top: 30%; left: 25%; animation-delay: 2s;"></div>

<div class="stars" style="top: 15%; left: 60%; animation-delay: 0.5s;"></div>

<div class="stars" style="top: 40%; left: 70%; animation-delay: 1.5s;"></div>

<div class="stars" style="top: 25%; left: 40%; animation-delay: 0.8s;"></div>

<div class="stars" style="top: 35%; left: 85%; animation-delay: 2.2s;"></div>

<div class="stars" style="top: 45%; left: 15%; animation-delay: 1.8s;"></div>

<div class="grass"></div>

<div class="ui-panel">

<div class="ui-row">

<span>Fireflies:</span>

<span id="firefly-count">0</span>

</div>

<div class="ui-row">

<span>Coins:</span>

<span class="coins" id="coins">0</span>

</div>

<div class="ui-row">

<span>Jar:</span>

<span id="jar-count">0/50</span>

</div>

<div class="ui-row">

<span>Collected:</span>

<span class="collected" id="total-caught">0</span>

</div>

<div class="save-status" id="save-status">

💾 Auto-Save: Ready

</div>

<button class="sell-button" id="sell-button" onclick="sellFireflies()">

Sell All Fireflies

</button>

<button class="scrapbook-button" id="scrapbook-button" onclick="toggleScrapbook()">

📚 View Scrapbook

</button>

<button class="save-button" id="manual-save-button" onclick="manualSave()">

💾 Save Game

</button>

<button class="export-button" id="export-button" onclick="exportSave()">

📤 Export Data

</button>

<div class="rarity-info">

<div>💛 Common: 5 coins</div>

<div>💚 Uncommon: 15 coins</div>

<div>💙 Rare: 35 coins</div>

<div>💜 Epic: 75 coins</div>

<div>💛 Legendary: 150 coins</div>

<div style="color: #ff69b4;">🌟 Super Legendary: 500 coins</div>

<div style="color: #ff0080;">💎 Bonus: 5000 coins!</div>

<div style="font-size: 10px; margin-top: 5px; opacity: 0.7;" id="multiplier-info">

Coin Multiplier: x<span id="current-multiplier">1.0</span>

</div>

</div>

</div>

<div class="upgrade-panel">

<h4 style="margin-top: 0;">Upgrades</h4>

<button class="upgrade-button" id="upgrade-bigger-net" onclick="buyUpgrade('bigger-net')">

🥅 Bigger Net Lv.1 (50 coins)

</button>

<button class="upgrade-button" id="upgrade-lucky-charm" onclick="buyUpgrade('lucky-charm')">

🍀 Lucky Charm Lv.1 (100 coins)

</button>

<button class="upgrade-button" id="upgrade-speed-boost" onclick="buyUpgrade('speed-boost')">

⚡ Slow Motion Lv.1 (150 coins)

</button>

<button class="upgrade-button" id="upgrade-bigger-jar" onclick="buyUpgrade('bigger-jar')">

🏺 Bigger Jar Lv.1 (75 coins)

</button>

<button class="upgrade-button" id="upgrade-auto-catcher" onclick="buyUpgrade('auto-catcher')">

🤖 Auto-Catcher Lv.1 (500 coins)

</button>

<button class="upgrade-button" id="upgrade-bonus-magnet" onclick="buyUpgrade('bonus-magnet')">

🧲 Bonus Magnet Lv.1 (1000 coins)

</button>

<button class="upgrade-button" id="upgrade-coin-multiplier" onclick="buyUpgrade('coin-multiplier')">

💎 Coin Multiplier Lv.1 (2000 coins)

</button>

</div>

<div class="timer-panel">

<h4 style="margin-top: 0;">Timers</h4>

<div class="timer-row">

<span>Next Bonus:</span>

<span id="bonus-timer">6:45</span>

</div>

<div class="timer-row">

<span>Max Fireflies:</span>

<span id="max-fireflies">60</span>

</div>

</div>

<div class="jar-container">

<div class="jar-lid"></div>

<div class="jar" id="jar">

<!-- Caught fireflies will appear here -->

</div>

</div>

<!-- Scrapbook Modal -->

<div class="scrapbook-modal" id="scrapbook-modal" style="display: none;">

<div class="scrapbook-content">

<div class="scrapbook-header">

<h2>🦋 Firefly Scrapbook 🦋</h2>

<div>

<button class="import-button" id="import-button" onclick="document.getElementById('import-input').click()">

📥 Import

</button>

<button class="close-button" onclick="toggleScrapbook()">✖</button>

</div>

</div>

<div class="scrapbook-tabs">

<button class="tab-button active" onclick="switchScrapbookTab('all')">All (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('common')">Common (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('uncommon')">Uncommon (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('rare')">Rare (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('epic')">Epic (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('legendary')">Legendary (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('super-legendary')">Super Legendary (0)</button>

<button class="tab-button" onclick="switchScrapbookTab('bonus')">Bonus (0)</button>

</div>

<div class="scrapbook-stats">

<div>Total Collected: <span id="scrapbook-total">0</span></div>

<div>Unique Species: <span id="unique-count">0</span></div>

<div>Current Page: <span id="current-page-count">0</span></div>

</div>

<div class="scrapbook-grid" id="scrapbook-grid">

<!-- Firefly cards will be added here -->

</div>

</div>

</div>

<input type="file" id="import-input" accept=".json" style="display: none;">

</div>

<script>

let coins = 0;

let fireflies = [];

let jarFireflies = [];

let jarCapacity = 50;

let catchRadius = 1;

let luckBonus = 0;

let speedMultiplier = 1;

let maxFireflies = 60;

let bonusFireflyActive = false;

let gameStartTime = Date.now();

let nextBonusTime = 6 \* 60 \* 1000 + 45 \* 1000;

let customFireflyImages = {};

let autoCatcherLevel = 0;

let bonusMagnetLevel = 0;

let coinMultiplier = 1;

let fireflyScrapbook = [];

let totalCaught = 0;

let lastSaveTime = Date.now();

let autoSaveInterval = 30000; // Auto-save every 30 seconds

let lastLegendarySpawn = Date.now();

let lastSuperLegendarySpawn = Date.now();

let legendarySpawnInterval = 45 \* 60 \* 1000; // 45 minutes base

let superLegendarySpawnInterval = 120 \* 60 \* 1000; // 120 minutes

// Firefly personality system

const fireflyThemes = {

'country-cousins': {

names: ['Cletus', 'Dolly', 'Jebediah', 'Bessie', 'Clyde', 'Magnolia', 'Rufus', 'Maybelle', 'Homer', 'Possum', 'Cricket', 'Moonbeam', 'Buttercup', 'Banjo', 'Petunia'],

traits: ['loves porch sittin\'', 'makes the best firefly jam', 'knows every creek for miles', 'can call rain with a whistle', 'tells tall tales', 'grows prize-winning tomatoes', 'plays mean banjo', 'has seventeen cousins'],

locations: ['the old oak tree', 'grandpappy\'s farm', 'the fishing hole', 'behind the barn', 'near the cornfield', 'by the creek', 'the front porch'],

bios: [

'Born and raised {location}, {name} {trait} and dreams of firefly rodeos.',

'{name} hails from {location} where they {trait}. Local legend says they once out-glowed the moon!',

'Sweet as honey and twice as bright, {name} {trait} while living {location}.',

'This country firefly {trait} and calls {location} home. Yee-haw!'

]

},

'citified-dignified': {

names: ['Reginald', 'Vivienne', 'Bartholomew', 'Penelope', 'Maximilian', 'Cordelia', 'Theodore', 'Beatrice', 'Montgomery', 'Priscilla', 'Fitzgerald', 'Evangeline'],

traits: ['reads financial newspapers', 'attends opera every Tuesday', 'speaks four languages fluently', 'collects vintage wine', 'never misses afternoon tea', 'owns a penthouse', 'graduated summa cum laude', 'has impeccable manners'],

locations: ['the metropolitan district', 'a high-rise apartment', 'the cultural center', 'an exclusive club', 'the university library', 'a luxury hotel', 'the business district'],

bios: [

'Sophisticated {name} {trait} from their residence {location}. Simply divine!',

'Urban elite {name} {trait} while maintaining an elegant lifestyle {location}.',

'Distinguished and refined, {name} {trait} and frequents {location}.',

'Metropolitan firefly {name} {trait}. You\'ll find them {location} discussing philosophy.'

]

},

'mystical-magical': {

names: ['Starwhisper', 'Moonweaver', 'Crystalwing', 'Shadowdancer', 'Dreamspell', 'Nighthaven', 'Ethermist', 'Spellbound', 'Mysticglow', 'Enchanta', 'Lumina', 'Celestine'],

traits: ['weaves moonlight into dreams', 'guards ancient secrets', 'can predict the future in star patterns', 'speaks to the forest spirits', 'collects fallen stars', 'brews wisdom potions', 'dances with shadows', 'sings lullabies to sleepy flowers'],

locations: ['the enchanted grove', 'a crystal cave', 'the fairy ring', 'among the ancient runes', 'the mystical library', 'beside the wishing well', 'the ethereal meadow'],

bios: [

'Mystical {name} {trait} while dwelling {location}. Magic flows through their very essence.',

'Born under a shooting star, {name} {trait} and makes their home {location}.',

'Ethereal {name} {trait}. Legends say they appear {location} during the full moon.',

'Wise and wonderful, {name} {trait} from their sacred sanctuary {location}.'

]

},

'adventurous-explorers': {

names: ['Scout', 'Journey', 'Compass', 'Atlas', 'Ranger', 'Quest', 'Explorer', 'Trek', 'Nomad', 'Wanderer', 'Pathfinder', 'Roamer', 'Voyager', 'Pioneer'],

traits: ['has visited seven continents', 'climbed the highest peaks', 'discovered three new species', 'speaks ancient map language', 'never travels without a compass', 'collects passport stamps', 'survived in the wilderness for months', 'finds treasure everywhere'],

locations: ['base camp', 'the expedition tent', 'a remote outpost', 'the ranger station', 'an unexplored territory', 'the adventure guild', 'a mountain lodge'],

bios: [

'Intrepid explorer {name} {trait} and currently camps {location}.',

'Adventure-seeking {name} {trait} between expeditions from {location}.',

'Bold and brave, {name} {trait} while planning the next journey from {location}.',

'World-traveling {name} {trait} and makes temporary headquarters {location}.'

]

},

'scholarly-bookworms': {

names: ['Professor', 'Sage', 'Quill', 'Wisdom', 'Scholar', 'Newton', 'Einstein', 'Aristotle', 'Bookworm', 'Thesis', 'Research', 'Logic', 'Theory', 'Academia'],

traits: ['has memorized 1,247 books', 'speaks in scientific theorems', 'discovers new math formulas', 'reads three languages simultaneously', 'organizes by dewey decimal', 'fact-checks everything twice', 'loves philosophical debates', 'collects rare manuscripts'],

locations: ['the university library', 'a private study', 'the research laboratory', 'among the ancient texts', 'the observatory', 'a quiet reading nook', 'the academy'],

bios: [

'Brilliant {name} {trait} while conducting research {location}.',

'Intellectual {name} {trait} and spends countless hours {location}.',

'Scholarly {name} {trait} when not busy studying {location}.',

'Academic firefly {name} {trait} from their prestigious position {location}.'

]

},

'artistic-dreamers': {

names: ['Canvas', 'Melody', 'Palette', 'Rhythm', 'Sketch', 'Harmony', 'Brushstroke', 'Symphony', 'Muse', 'Sonnet', 'Fresco', 'Ballad', 'Sculpt', 'Verse'],

traits: ['paints with pure light', 'composes lullabies for baby stars', 'sculpts clouds into masterpieces', 'writes poetry on flower petals', 'dances to colors only they can see', 'creates music from dewdrops', 'dreams in watercolors', 'turns imagination into reality'],

locations: ['the art studio', 'a bohemian café', 'the music conservatory', 'among the gallery walls', 'the creative workshop', 'an inspiring garden', 'the poetry corner'],

bios: [

'Creative {name} {trait} from their inspiring workspace {location}.',

'Artistic soul {name} {trait} while finding inspiration {location}.',

'Imaginative {name} {trait} and creates beauty {location}.',

'Visionary {name} {trait} between masterpieces at {location}.'

]

},

'royal-court': {

names: ['Duke', 'Duchess', 'Baron', 'Countess', 'Lord', 'Lady', 'Sir', 'Princess', 'Majesty', 'Noble', 'Royal', 'Crown', 'Scepter', 'Jewel', 'Regal'],

traits: ['wears a tiny crown', 'holds court every evening', 'speaks only in royal proclamations', 'has a personal herald', 'collects precious gems', 'rules with wisdom and grace', 'hosts magnificent balls', 'grants wishes to loyal subjects'],

locations: ['the royal palace', 'the throne room', 'the royal gardens', 'a majestic castle', 'the court chambers', 'among the crown jewels', 'the royal library'],

bios: [

'His/Her Royal Brightness {name} {trait} while presiding over {location}.',

'Noble {name} {trait} from their magnificent domain {location}.',

'Majestic {name} {trait} and maintains royal residence {location}.',

'Regal firefly {name} {trait} while governing wisely from {location}.'

]

},

'mischievous-tricksters': {

names: ['Giggles', 'Prank', 'Mischief', 'Tickle', 'Joke', 'Rascal', 'Trouble', 'Scamp', 'Fidget', 'Wiggle', 'Chaos', 'Pickle', 'Hijinks', 'Shenanigan'],

traits: ['hides socks for fun', 'switches sugar with salt', 'makes funny faces in mirrors', 'tickles sleeping caterpillars', 'rearranges garden gnomes', 'plays practical jokes', 'giggles at everything', 'turns serious moments silly'],

locations: ['behind the cookie jar', 'in the toy box', 'under the stairs', 'near the playground', 'in the joke shop', 'around the corner', 'the comedy club'],

bios: [

'Playful prankster {name} {trait} while plotting mischief {location}.',

'Silly {name} {trait} and spreads laughter wherever they go, especially {location}.',

'Mischievous {name} {trait} between shenanigans {location}.',

'Giggly troublemaker {name} {trait} and brings joy to everyone {location}.'

]

}

};

function generateFireflyPersonality(rarity) {

const themes = Object.keys(fireflyThemes);

const theme = themes[Math.floor(Math.random() \* themes.length)];

const themeData = fireflyThemes[theme];

const name = themeData.names[Math.floor(Math.random() \* themeData.names.length)];

const trait = themeData.traits[Math.floor(Math.random() \* themeData.traits.length)];

const location = themeData.locations[Math.floor(Math.random() \* themeData.locations.length)];

const bioTemplate = themeData.bios[Math.floor(Math.random() \* themeData.bios.length)];

const bio = bioTemplate

.replace(/{name}/g, name)

.replace(/{trait}/g, trait)

.replace(/{location}/g, location);

return {

name,

theme: theme.replace('-', ' ').replace(/\b\w/g, l => l.toUpperCase()),

bio,

rarity,

caughtAt: new Date().toLocaleString(),

id: Date.now() + Math.random()

};

}

// Multi-level upgrade system

let upgradelevels = {

'bigger-net': 0,

'lucky-charm': 0,

'speed-boost': 0,

'bigger-jar': 0,

'auto-catcher': 0,

'bonus-magnet': 0,

'coin-multiplier': 0

};

const gameContainer = document.querySelector('.game-container');

const coinsElement = document.getElementById('coins');

const fireflyCountElement = document.getElementById('firefly-count');

const jarCountElement = document.getElementById('jar-count');

const jar = document.getElementById('jar');

const sellButton = document.getElementById('sell-button');

const bonusTimerElement = document.getElementById('bonus-timer');

const maxFirefliesElement = document.getElementById('max-fireflies');

// Check for uploaded firefly images

async function loadCustomImages() {

try {

const imageFiles = [

'image\_1.png', 'image\_2.png', 'image\_3.png', 'image\_4.png', 'image\_5.png',

'image\_6.png', 'image\_7.png', 'image\_8.png', 'image\_9.png', 'image\_10.png'

];

const fireflyTypeImages = ['common', 'uncommon', 'rare', 'epic', 'legendary', 'bonus'];

for (let i = 0; i < imageFiles.length && i < 6; i++) {

try {

const imageData = await window.fs.readFile(imageFiles[i]);

const blob = new Blob([imageData]);

const imageUrl = URL.createObjectURL(blob);

const type = fireflyTypeImages[i] || 'common';

customFireflyImages[type] = imageUrl;

console.log(`Loaded custom image ${imageFiles[i]} for ${type} fireflies`);

} catch (e) {

console.log(`Could not load ${imageFiles[i]}, using default design`);

}

}

for (let i = 6; i < imageFiles.length; i++) {

try {

const imageData = await window.fs.readFile(imageFiles[i]);

const blob = new Blob([imageData]);

const imageUrl = URL.createObjectURL(blob);

customFireflyImages[`variant${i}`] = imageUrl;

console.log(`Loaded variant image ${imageFiles[i]}`);

} catch (e) {

console.log(`Could not load ${imageFiles[i]}`);

}

}

} catch (error) {

console.log('Using default firefly designs');

}

}

const fireflyTypes = {

common: { value: 5, chance: 50, className: 'firefly-common', jarClass: 'common' },

uncommon: { value: 15, chance: 30, className: 'firefly-uncommon', jarClass: 'uncommon' },

rare: { value: 35, chance: 15, className: 'firefly-rare', jarClass: 'rare' },

epic: { value: 75, chance: 4.5, className: 'firefly-epic', jarClass: 'epic' },

legendary: { value: 150, chance: 0.5, className: 'firefly-legendary', jarClass: 'legendary' },

'super-legendary': { value: 500, chance: 0, className: 'firefly-super-legendary', jarClass: 'super-legendary' },

bonus: { value: 5000, chance: 0, className: 'firefly-bonus', jarClass: 'bonus' }

};

const upgradeConfigs = {

'bigger-net': {

baseCost: 50,

costMultiplier: 2,

maxLevel: 10,

effect: (level) => 1 + (level \* 0.5),

name: '🥅 Bigger Net'

},

'lucky-charm': {

baseCost: 100,

costMultiplier: 2.5,

maxLevel: 8,

effect: (level) => level \* 15,

name: '🍀 Lucky Charm'

},

'speed-boost': {

baseCost: 150,

costMultiplier: 3,

maxLevel: 6,

effect: (level) => 1 + (level \* 0.8),

name: '⚡ Slow Motion'

},

'bigger-jar': {

baseCost: 75,

costMultiplier: 1.8,

maxLevel: 20,

effect: (level) => 50 + (level \* 25),

name: '🏺 Bigger Jar'

},

'auto-catcher': {

baseCost: 500,

costMultiplier: 3.5,

maxLevel: 5,

effect: (level) => level,

name: '🤖 Auto-Catcher'

},

'bonus-magnet': {

baseCost: 1000,

costMultiplier: 4,

maxLevel: 3,

effect: (level) => level,

name: '🧲 Bonus Magnet'

},

'coin-multiplier': {

baseCost: 2000,

costMultiplier: 5,

maxLevel: 10,

effect: (level) => 1 + (level \* 0.25),

name: '💎 Coin Multiplier'

}

};

function getUpgradeCost(upgradeType, level) {

const config = upgradeConfigs[upgradeType];

return Math.floor(config.baseCost \* Math.pow(config.costMultiplier, level));

}

function getRandomFireflyType() {

const totalChance = 100 + luckBonus;

let random = Math.random() \* totalChance;

for (const [type, data] of Object.entries(fireflyTypes)) {

if (random < data.chance) {

return { type, ...data };

}

random -= data.chance;

}

return { type: 'common', ...fireflyTypes.common };

}

function createFirefly(forceType = null) {

if (fireflies.length >= maxFireflies && !forceType) return;

const firefly = document.createElement('div');

let fireflyData;

// Handle special rare spawns

if (forceType) {

fireflyData = { type: forceType, ...fireflyTypes[forceType] };

} else {

// Check for rare legendary spawns (1-2 per hour)

const currentTime = Date.now();

const timeSinceLegendary = currentTime - lastLegendarySpawn;

const timeSinceSuperLegendary = currentTime - lastSuperLegendarySpawn;

// Random chance for legendary within the time window (45-75 minutes)

if (timeSinceLegendary > legendarySpawnInterval && Math.random() < 0.003) { // 0.3% chance per spawn

fireflyData = { type: 'legendary', ...fireflyTypes.legendary };

lastLegendarySpawn = currentTime;

showRareSpawnNotification('🌟 LEGENDARY FIREFLY APPEARED! 🌟');

}

// Super legendary spawn (every 120+ minutes)

else if (timeSinceSuperLegendary > superLegendarySpawnInterval && Math.random() < 0.001) { // 0.1% chance per spawn

fireflyData = { type: 'super-legendary', ...fireflyTypes['super-legendary'] };

lastSuperLegendarySpawn = currentTime;

showRareSpawnNotification('💫 SUPER LEGENDARY FIREFLY! ULTRA RARE! 💫');

} else {

fireflyData = getRandomFireflyType();

}

}

firefly.className = `firefly ${fireflyData.className}`;

firefly.fireflyType = fireflyData.type;

firefly.fireflyValue = fireflyData.value;

firefly.fireflyJarClass = fireflyData.jarClass;

const padding = 20 + (upgradelevels['bigger-net'] \* 10);

firefly.style.padding = `${padding}px`;

firefly.style.margin = `-${padding}px`;

let imageUrl = null;

if (customFireflyImages[fireflyData.type]) {

imageUrl = customFireflyImages[fireflyData.type];

} else {

const variantKeys = Object.keys(customFireflyImages).filter(key => key.startsWith('variant'));

if (variantKeys.length > 0) {

const randomVariant = variantKeys[Math.floor(Math.random() \* variantKeys.length)];

imageUrl = customFireflyImages[randomVariant];

}

}

if (imageUrl) {

firefly.style.background = 'none';

firefly.style.boxShadow = 'none';

firefly.style.backgroundImage = `url(${imageUrl})`;

firefly.style.backgroundSize = 'contain';

firefly.style.backgroundRepeat = 'no-repeat';

firefly.style.backgroundPosition = 'center';

const baseSize = fireflyData.type === 'bonus' ? 32 :

fireflyData.type === 'super-legendary' ? 28 :

fireflyData.type === 'legendary' ? 24 :

fireflyData.type === 'epic' ? 20 :

fireflyData.type === 'rare' ? 16 :

fireflyData.type === 'uncommon' ? 14 : 12;

firefly.style.width = `${baseSize}px`;

firefly.style.height = `${baseSize}px`;

firefly.style.filter = `drop-shadow(0 0 10px rgba(255, 255, 255, 0.8))`;

}

const x = Math.random() \* (window.innerWidth - 40) + 20;

const y = Math.random() \* (window.innerHeight - 120) + 60;

firefly.style.left = x + 'px';

firefly.style.top = y + 'px';

const baseSpeed = 0.7 / speedMultiplier;

firefly.dx = (Math.random() - 0.5) \* baseSpeed;

firefly.dy = (Math.random() - 0.5) \* baseSpeed;

firefly.x = x;

firefly.y = y;

if (fireflyData.type === 'bonus') {

bonusFireflyActive = true;

firefly.bonusTimer = 120000;

firefly.style.filter = `drop-shadow(0 0 20px rgba(255, 0, 128, 1)) drop-shadow(0 0 40px rgba(255, 0, 128, 0.5))`;

setTimeout(() => {

if (firefly.parentNode && bonusFireflyActive) {

removeFirefly(firefly);

bonusFireflyActive = false;

}

}, 120000);

}

// Special handling for super legendary - stays longer

if (fireflyData.type === 'super-legendary') {

firefly.style.filter = `drop-shadow(0 0 25px rgba(255, 105, 180, 1)) drop-shadow(0 0 50px rgba(255, 20, 147, 0.8))`;

// Super legendary stays for 5 minutes

setTimeout(() => {

if (firefly.parentNode) {

removeFirefly(firefly);

}

}, 300000);

}

firefly.addEventListener('click', catchFirefly);

firefly.addEventListener('touchstart', catchFirefly);

gameContainer.appendChild(firefly);

fireflies.push(firefly);

updateUI();

}

function removeFirefly(firefly) {

const index = fireflies.indexOf(firefly);

if (index > -1) {

fireflies.splice(index, 1);

}

if (firefly.parentNode) {

firefly.parentNode.removeChild(firefly);

}

}

function catchFirefly(event) {

event.preventDefault();

const firefly = event.target;

const personality = generateFireflyPersonality(firefly.fireflyType);

fireflyScrapbook.push(personality);

totalCaught++;

if (firefly.fireflyType === 'bonus') {

bonusFireflyActive = false;

}

if (navigator.vibrate) {

navigator.vibrate(firefly.fireflyType === 'bonus' ? 200 : 50);

}

const finalValue = Math.floor(firefly.fireflyValue \* coinMultiplier);

showValuePopup(firefly.x, firefly.y, firefly.fireflyValue, firefly.fireflyType === 'bonus', personality.name);

firefly.classList.add('caught-firefly');

firefly.removeEventListener('click', catchFirefly);

firefly.removeEventListener('touchstart', catchFirefly);

removeFirefly(firefly);

if (jarFireflies.length < jarCapacity) {

addFireflyToJar(firefly.fireflyJarClass, firefly.fireflyValue);

} else {

coins += finalValue;

}

setTimeout(() => {

if (firefly.parentNode) {

firefly.parentNode.removeChild(firefly);

}

}, 800);

updateUI();

}

function showValuePopup(x, y, value, isBonus = false, fireflyName = '') {

const popup = document.createElement('div');

popup.className = 'value-popup';

const finalValue = Math.floor(value \* coinMultiplier);

let text = coinMultiplier > 1 ? `+${value} x${coinMultiplier.toFixed(1)} = ${finalValue}` : `+${finalValue}`;

if (fireflyName) {

text = `${fireflyName}: ${text}`;

}

popup.textContent = text;

popup.style.left = x + 'px';

popup.style.top = y + 'px';

if (isBonus) {

popup.style.fontSize = '28px';

popup.style.color = '#ff0080';

popup.style.fontWeight = 'bold';

popup.textContent = `💎 ${fireflyName}: MEGA BONUS ${finalValue}! 💎`;

}

gameContainer.appendChild(popup);

setTimeout(() => {

if (popup.parentNode) {

popup.parentNode.removeChild(popup);

}

}, isBonus ? 2000 : 1000);

}

function addFireflyToJar(jarClass, value) {

const multipliedValue = Math.floor(value \* coinMultiplier);

if (jarFireflies.length >= jarCapacity) {

coins += multipliedValue;

return;

}

const jarFirefly = document.createElement('div');

jarFirefly.className = `jar-firefly ${jarClass}`;

jarFirefly.fireflyValue = multipliedValue;

const x = Math.random() \* 85 + 5;

const y = Math.random() \* 100 + 15;

jarFirefly.style.left = x + '%';

jarFirefly.style.top = y + '%';

jarFirefly.style.animationDelay = Math.random() \* 2 + 's';

jar.appendChild(jarFirefly);

jarFireflies.push(jarFirefly);

}

function sellFireflies() {

let totalValue = 0;

jarFireflies.forEach(firefly => {

totalValue += firefly.fireflyValue;

});

coins += totalValue;

jarFireflies.forEach(firefly => {

jar.removeChild(firefly);

});

jarFireflies = [];

updateUI();

}

function buyUpgrade(upgradeType) {

const config = upgradeConfigs[upgradeType];

const currentLevel = upgradelevels[upgradeType];

const cost = getUpgradeCost(upgradeType, currentLevel);

if (currentLevel >= config.maxLevel || coins < cost) return;

coins -= cost;

upgradelevels[upgradeType]++;

switch(upgradeType) {

case 'bigger-net':

catchRadius = config.effect(upgradelevels[upgradeType]);

document.querySelectorAll('.firefly').forEach(f => {

const padding = 20 + (upgradelevels[upgradeType] \* 10);

f.style.padding = `${padding}px`;

f.style.margin = `-${padding}px`;

});

break;

case 'lucky-charm':

luckBonus = config.effect(upgradelevels[upgradeType]);

break;

case 'speed-boost':

speedMultiplier = config.effect(upgradelevels[upgradeType]);

break;

case 'bigger-jar':

jarCapacity = config.effect(upgradelevels[upgradeType]);

break;

case 'auto-catcher':

autoCatcherLevel = upgradelevels[upgradeType];

break;

case 'bonus-magnet':

bonusMagnetLevel = upgradelevels[upgradeType];

if (bonusMagnetLevel > 0) {

nextBonusTime = Math.max(nextBonusTime - (bonusMagnetLevel \* 60000), 60000);

}

break;

case 'coin-multiplier':

coinMultiplier = config.effect(upgradelevels[upgradeType]);

break;

}

updateUI();

}

function updateUI() {

coinsElement.textContent = coins;

fireflyCountElement.textContent = fireflies.length;

jarCountElement.textContent = `${jarFireflies.length}/${jarCapacity}`;

maxFirefliesElement.textContent = maxFireflies;

const totalCaughtElement = document.getElementById('total-caught');

if (totalCaughtElement) {

totalCaughtElement.textContent = totalCaught;

}

const multiplierElement = document.getElementById('current-multiplier');

if (multiplierElement) {

multiplierElement.textContent = coinMultiplier.toFixed(1);

}

const currentTime = Date.now() - gameStartTime;

let timeUntilNextBonus = nextBonusTime - currentTime;

if (bonusMagnetLevel > 0) {

timeUntilNextBonus = Math.max(timeUntilNextBonus - (bonusMagnetLevel \* 60000), 1000);

}

if (timeUntilNextBonus <= 0 && !bonusFireflyActive) {

createFirefly('bonus');

nextBonusTime = currentTime + (6 \* 60 \* 1000 + 45 \* 1000);

showBonusNotification();

}

const minutes = Math.floor(Math.abs(timeUntilNextBonus) / 60000);

const seconds = Math.floor((Math.abs(timeUntilNextBonus) % 60000) / 1000);

bonusTimerElement.textContent = bonusFireflyActive ? 'ACTIVE!' : `${minutes}:${seconds.toString().padStart(2, '0')}`;

const fourMinutes = 4 \* 60 \* 1000;

const difficultyLevel = Math.floor(currentTime / fourMinutes);

const newMaxFireflies = 60 + (difficultyLevel \* 100);

if (newMaxFireflies !== maxFireflies) {

maxFireflies = newMaxFireflies;

console.log(`Difficulty increased! Max fireflies now: ${maxFireflies}`);

}

sellButton.disabled = jarFireflies.length === 0;

Object.keys(upgradeConfigs).forEach((upgradeType, index) => {

const button = document.getElementById(`upgrade-${upgradeType}`);

if (!button) return;

const config = upgradeConfigs[upgradeType];

const currentLevel = upgradelevels[upgradeType];

const cost = getUpgradeCost(upgradeType, currentLevel);

const maxLevel = config.maxLevel;

if (currentLevel >= maxLevel) {

button.textContent = `${config.name} (MAX)`;

button.disabled = true;

} else {

button.textContent = `${config.name} Lv.${currentLevel + 1} (${cost} coins)`;

button.disabled = coins < cost;

}

});

}

function toggleScrapbook() {

const modal = document.getElementById('scrapbook-modal');

if (modal.style.display === 'none') {

modal.style.display = 'block';

updateScrapbook();

} else {

modal.style.display = 'none';

}

}

function updateScrapbook() {

const scrapbookGrid = document.getElementById('scrapbook-grid');

const scrapbookTotal = document.getElementById('scrapbook-total');

const uniqueCount = document.getElementById('unique-count');

const currentPageCount = document.getElementById('current-page-count');

// Count unique themes

const uniqueThemes = new Set(fireflyScrapbook.map(f => f.theme)).size;

// Get fireflies for current tab

const displayFireflies = getFirefliesByRarity(currentScrapbookTab);

scrapbookTotal.textContent = fireflyScrapbook.length;

uniqueCount.textContent = uniqueThemes;

currentPageCount.textContent = displayFireflies.length;

// Update tab counts

updateScrapbookTabCounts();

// Clear and rebuild grid

scrapbookGrid.innerHTML = '';

// Show most recent catches first

const sortedFireflies = [...displayFireflies].reverse();

if (sortedFireflies.length > 0) {

sortedFireflies.forEach(firefly => {

const card = document.createElement('div');

card.className = 'firefly-card';

card.innerHTML = `

<div class="firefly-card-header">

<div class="firefly-name">${firefly.name}</div>

<div class="firefly-rarity rarity-${firefly.rarity}">${firefly.rarity.toUpperCase().replace('-', ' ')}</div>

</div>

<div class="firefly-theme">${firefly.theme}</div>

<div class="firefly-bio">${firefly.bio}</div>

<div class="firefly-caught">Caught: ${firefly.caughtAt}</div>

`;

scrapbookGrid.appendChild(card);

});

} else {

const emptyMessage = currentScrapbookTab === 'all'

? 'No fireflies caught yet! Start catching to build your collection!'

: `No ${currentScrapbookTab.replace('-', ' ')} fireflies caught yet! Keep playing to find them!`;

scrapbookGrid.innerHTML = `<div style="text-align: center; color: #888; font-style: italic; grid-column: 1 / -1;">${emptyMessage}</div>`;

}

}

// Save/Load System

function saveGameData() {

const saveData = {

version: '1.1', // Updated version for new features

timestamp: Date.now(),

gameData: {

coins,

totalCaught,

fireflyScrapbook,

upgradelevels,

jarCapacity,

catchRadius,

luckBonus,

speedMultiplier,

maxFireflies,

autoCatcherLevel,

bonusMagnetLevel,

coinMultiplier,

gameStartTime,

playTime: Date.now() - gameStartTime,

lastLegendarySpawn,

lastSuperLegendarySpawn

}

};

try {

localStorage.setItem('firefly-catcher-save', JSON.stringify(saveData));

return true;

} catch (error) {

console.warn('LocalStorage not available, save failed:', error);

return false;

}

}

function loadGameData() {

try {

const saveString = localStorage.getItem('firefly-catcher-save');

if (!saveString) return false;

const saveData = JSON.parse(saveString);

if (!saveData.gameData) return false;

const data = saveData.gameData;

coins = data.coins || 0;

totalCaught = data.totalCaught || 0;

fireflyScrapbook = data.fireflyScrapbook || [];

upgradelevels = data.upgradelevels || {

'bigger-net': 0, 'lucky-charm': 0, 'speed-boost': 0,

'bigger-jar': 0, 'auto-catcher': 0, 'bonus-magnet': 0, 'coin-multiplier': 0

};

jarCapacity = data.jarCapacity || 50;

catchRadius = data.catchRadius || 1;

luckBonus = data.luckBonus || 0;

speedMultiplier = data.speedMultiplier || 1;

maxFireflies = data.maxFireflies || 60;

autoCatcherLevel = data.autoCatcherLevel || 0;

bonusMagnetLevel = data.bonusMagnetLevel || 0;

coinMultiplier = data.coinMultiplier || 1;

// Load rare spawn timers

lastLegendarySpawn = data.lastLegendarySpawn || Date.now();

lastSuperLegendarySpawn = data.lastSuperLegendarySpawn || Date.now();

const playTime = data.playTime || 0;

gameStartTime = Date.now() - playTime;

console.log(`Game loaded! Welcome back! You've caught ${totalCaught} fireflies.`);

updateSaveStatus('💾 Game Loaded!', 'loaded');

return true;

} catch (error) {

console.warn('Failed to load game data:', error);

return false;

}

}

function autoSave() {

const success = saveGameData();

if (success) {

updateSaveStatus('💾 Auto-Saved', 'success');

lastSaveTime = Date.now();

} else {

updateSaveStatus('❌ Save Failed', 'error');

}

}

function manualSave() {

const success = saveGameData();

if (success) {

updateSaveStatus('💾 Game Saved!', 'success');

} else {

updateSaveStatus('❌ Save Failed', 'error');

}

}

function updateSaveStatus(message, type = 'default') {

const statusElement = document.getElementById('save-status');

if (!statusElement) return;

statusElement.textContent = message;

statusElement.className = 'save-status';

if (type === 'saving') {

statusElement.classList.add('saving');

} else if (type === 'error') {

statusElement.classList.add('error');

}

if (type !== 'default') {

setTimeout(() => {

statusElement.textContent = '💾 Auto-Save: Ready';

statusElement.className = 'save-status';

}, 3000);

}

}

function exportSave() {

const saveData = {

version: '1.1',

timestamp: Date.now(),

gameData: {

coins,

totalCaught,

fireflyScrapbook,

upgradelevels,

jarCapacity,

catchRadius,

luckBonus,

speedMultiplier,

maxFireflies,

autoCatcherLevel,

bonusMagnetLevel,

coinMultiplier,

gameStartTime,

playTime: Date.now() - gameStartTime,

lastLegendarySpawn,

lastSuperLegendarySpawn

}

};

const dataStr = JSON.stringify(saveData, null, 2);

const blob = new Blob([dataStr], { type: 'application/json' });

const url = URL.createObjectURL(blob);

const link = document.createElement('a');

link.href = url;

link.download = `firefly-catcher-save-${new Date().toISOString().split('T')[0]}.json`;

link.click();

URL.revokeObjectURL(url);

updateSaveStatus('📤 Data Exported!', 'success');

}

function showBonusNotification() {

const notification = document.createElement('div');

notification.style.position = 'fixed';

notification.style.top = '50%';

notification.style.left = '50%';

notification.style.transform = 'translate(-50%, -50%)';

notification.style.background = 'rgba(255, 0, 128, 0.9)';

notification.style.color = 'white';

notification.style.padding = '20px';

notification.style.borderRadius = '15px';

notification.style.fontSize = '24px';

notification.style.fontWeight = 'bold';

notification.style.textAlign = 'center';

notification.style.zIndex = '10000';

notification.style.animation = 'pulse 0.5s ease-in-out';

notification.textContent = '🎯 BONUS FIREFLY APPEARED! 5000 COINS! 🎯';

gameContainer.appendChild(notification);

setTimeout(() => {

if (notification.parentNode) {

notification.parentNode.removeChild(notification);

}

}, 3000);

}

function showRareSpawnNotification(message) {

const notification = document.createElement('div');

notification.style.position = 'fixed';

notification.style.top = '50%';

notification.style.left = '50%';

notification.style.transform = 'translate(-50%, -50%)';

notification.style.background = 'linear-gradient(45deg, #ff69b4, #ff1493, #dc143c)';

notification.style.color = 'white';

notification.style.padding = '25px';

notification.style.borderRadius = '20px';

notification.style.fontSize = '28px';

notification.style.fontWeight = 'bold';

notification.style.textAlign = 'center';

notification.style.zIndex = '10001';

notification.style.border = '3px solid #ffffff';

notification.style.boxShadow = '0 0 30px rgba(255, 105, 180, 0.8)';

notification.style.animation = 'pulse 0.3s ease-in-out infinite';

notification.textContent = message;

gameContainer.appendChild(notification);

setTimeout(() => {

if (notification.parentNode) {

notification.parentNode.removeChild(notification);

}

}, 5000); // Show longer for rare spawns

}

let currentScrapbookTab = 'all';

function switchScrapbookTab(tab) {

currentScrapbookTab = tab;

// Update tab buttons

document.querySelectorAll('.tab-button').forEach(btn => {

btn.classList.remove('active');

});

event.target.classList.add('active');

updateScrapbook();

}

function getFirefliesByRarity(rarity) {

if (rarity === 'all') return fireflyScrapbook;

return fireflyScrapbook.filter(firefly => firefly.rarity === rarity);

}

function updateScrapbookTabCounts() {

const counts = {

all: fireflyScrapbook.length,

common: getFirefliesByRarity('common').length,

uncommon: getFirefliesByRarity('uncommon').length,

rare: getFirefliesByRarity('rare').length,

epic: getFirefliesByRarity('epic').length,

legendary: getFirefliesByRarity('legendary').length,

'super-legendary': getFirefliesByRarity('super-legendary').length,

bonus: getFirefliesByRarity('bonus').length

};

document.querySelectorAll('.tab-button').forEach((button, index) => {

const rarities = ['all', 'common', 'uncommon', 'rare', 'epic', 'legendary', 'super-legendary', 'bonus'];

const rarity = rarities[index];

if (rarity && counts[rarity] !== undefined) {

const originalText = button.textContent.split(' (')[0];

button.textContent = `${originalText} (${counts[rarity]})`;

}

});

}

function moveFireflies() {

fireflies.forEach(firefly => {

firefly.x += firefly.dx;

firefly.y += firefly.dy;

if (firefly.x <= 20 || firefly.x >= window.innerWidth - 40) {

firefly.dx = -firefly.dx;

}

if (firefly.y <= 60 || firefly.y >= window.innerHeight - 120) {

firefly.dy = -firefly.dy;

}

firefly.x = Math.max(20, Math.min(window.innerWidth - 40, firefly.x));

firefly.y = Math.max(60, Math.min(window.innerHeight - 120, firefly.y));

firefly.style.left = firefly.x + 'px';

firefly.style.top = firefly.y + 'px';

if (Math.random() < 0.02) {

const baseSpeed = 0.7 / speedMultiplier;

firefly.dx += (Math.random() - 0.5) \* (0.15 / speedMultiplier);

firefly.dy += (Math.random() - 0.5) \* (0.15 / speedMultiplier);

firefly.dx = Math.max(-baseSpeed, Math.min(baseSpeed, firefly.dx));

firefly.dy = Math.max(-baseSpeed, Math.min(baseSpeed, firefly.dy));

}

});

}

function animateJarFireflies() {

jarFireflies.forEach(jarFirefly => {

if (Math.random() < 0.05) {

const x = Math.random() \* 85 + 5;

const y = Math.random() \* 100 + 15;

jarFirefly.style.left = x + '%';

jarFirefly.style.top = y + '%';

}

});

}

function gameLoop() {

moveFireflies();

animateJarFireflies();

requestAnimationFrame(gameLoop);

}

async function initGame() {

const saveLoaded = loadGameData();

await loadCustomImages();

for (let i = 0; i < 32; i++) {

setTimeout(() => createFirefly(), i \* 200);

}

gameLoop();

setInterval(() => {

if (fireflies.length < maxFireflies && jarFireflies.length < jarCapacity) {

createFirefly();

}

}, 1500);

setInterval(() => {

if (autoCatcherLevel > 0 && fireflies.length > 0) {

for (let i = 0; i < autoCatcherLevel && fireflies.length > 0; i++) {

const randomFirefly = fireflies[Math.floor(Math.random() \* fireflies.length)];

// Don't auto-catch bonus, legendary, or super legendary fireflies

if (randomFirefly && !['bonus', 'legendary', 'super-legendary'].includes(randomFirefly.fireflyType)) {

const personality = generateFireflyPersonality(randomFirefly.fireflyType);

fireflyScrapbook.push(personality);

totalCaught++;

const finalValue = Math.floor(randomFirefly.fireflyValue \* coinMultiplier);

if (jarFireflies.length < jarCapacity) {

addFireflyToJar(randomFirefly.fireflyJarClass, randomFirefly.fireflyValue);

} else {

coins += finalValue;

}

removeFirefly(randomFirefly);

}

}

updateUI();

}

}, 2000);

setInterval(() => {

autoSave();

}, autoSaveInterval);

setInterval(updateUI, 1000);

setupImportInput();

if (saveLoaded) {

setTimeout(() => {

updateSaveStatus('💾 Welcome Back!', 'loaded');

}, 1000);

}

updateUI();

}

function setupImportInput() {

const importInput = document.getElementById('import-input');

if (importInput) {

importInput.addEventListener('change', (e) => {

const file = e.target.files[0];

if (file) {

const reader = new FileReader();

reader.onload = (e) => {

try {

const saveData = JSON.parse(e.target.result);

if (!saveData.gameData) throw new Error('Invalid save file');

const confirm = window.confirm('This will overwrite your current progress. Are you sure?');

if (confirm) {

localStorage.setItem('firefly-catcher-save', JSON.stringify(saveData));

location.reload();

}

} catch (error) {

alert('Failed to import save file. Please check the file format.');

console.error('Import error:', error);

}

};

reader.readAsText(file);

}

importInput.value = '';

});

}

}

initGame();

window.addEventListener('beforeunload', (e) => {

saveGameData();

});

document.addEventListener('visibilitychange', () => {

if (document.hidden) {

saveGameData();

}

});

</script>

</body>

</html>