index.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>花火調合ゲーム - 花火師の工房</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            user-select: none;
        }

        body {
            font-family: 'Arial', sans-serif;
            background: linear-gradient(135deg, #2c1810 0%, #4a2c1a 100%);
            min-height: 100vh;
            color: #fff;
            overflow-x: hidden;
        }

        .header {
            background: rgba(0,0,0,0.3);
            padding: 15px;
            text-align: center;
            border-bottom: 2px solid #ff6b35;
        }

        .header h1 {
            font-size: 24px;
            color: #ff6b35;
            text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
        }

        .countdown {
            font-size: 14px;
            color: #ffd700;
            margin-top: 5px;
        }

        .game-container {
            display: flex;
            flex-direction: column;
            padding: 20px;
            gap: 20px;
```

```
    max-width: 600px;
    margin: 0 auto;
}

.ingredients-section {
    background: rgba(0,0,0,0.4);
    border-radius: 15px;
    padding: 20px;
    border: 2px solid #8b4513;
}

.section-title {
    font-size: 18px;
    color: #ffd700;
    margin-bottom: 15px;
    text-align: center;
}

.ingredients-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(80px, 1fr));
    gap: 10px;
}

.ingredient {
    background: linear-gradient(145deg, #8b4513, #a0522d);
    border-radius: 10px;
    padding: 10px;
    text-align: center;
    cursor: pointer;
    transition: all 0.3s ease;
    border: 2px solid transparent;
    min-height: 80px;
    display: flex;
    flex-direction: column;
    justify-content: center;
}

.ingredient:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(255,107,53,0.3);
    border-color: #ff6b35;
}

.ingredient.selected {
    background: linear-gradient(145deg, #ff6b35, #ff8c42);
    border-color: #ffd700;
}
```

```
.ingredient-icon {
    font-size: 24px;
    margin-bottom: 5px;
}

.ingredient-name {
    font-size: 12px;
    font-weight: bold;
}

.mixing-bowl {
    background: radial-gradient(circle, #2c2c2c 0%, #1a1a1a 100%);
    border-radius: 50%;
    width: 200px;
    height: 200px;
    margin: 20px auto;
    border: 4px solid #8b4513;
    position: relative;
    display: flex;
    align-items: center;
    justify-content: center;
    box-shadow: inset 0 0 20px rgba(0,0,0,0.5);
}

.bowl-contents {
    display: flex;
    flex-wrap: wrap;
    gap: 5px;
    justify-content: center;
    align-items: center;
    width: 150px;
    height: 150px;
    border-radius: 50%;
}

.bowl-ingredient {
    background: rgba(255,255,255,0.2);
    border-radius: 50%;
    width: 30px;
    height: 30px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 16px;
    animation: float 2s ease-in-out infinite;
}
```

```
@keyframes float {
    0%, 100% { transform: translateY(0px); }
    50% { transform: translateY(-5px); }
}

.controls {
    display: flex;
    gap: 10px;
    justify-content: center;
    margin-top: 20px;
}

.btn {
    padding: 12px 24px;
    border: none;
    border-radius: 25px;
    font-size: 16px;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s ease;
    text-transform: uppercase;
}

.btn-craft {
    background: linear-gradient(45deg, #ff6b35, #ff8c42);
    color: white;
}

.btn-clear {
    background: linear-gradient(45deg, #666, #888);
    color: white;
}

.btn:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(0,0,0,0.3);
}

.btn:disabled {
    opacity: 0.5;
    cursor: not-allowed;
    transform: none;
}

.result-section {
    background: rgba(0,0,0,0.4);
    border-radius: 15px;
    padding: 20px;
```

```
    border: 2px solid #4169e1;
    text-align: center;
    min-height: 150px;
}

.firework-display {
    font-size: 48px;
    margin: 20px 0;
    animation: sparkle 1s ease-in-out infinite;
}

@keyframes sparkle {
    0%, 100% { transform: scale(1); }
    50% { transform: scale(1.1); }
}

.firework-name {
    font-size: 20px;
    color: #ffd700;
    margin-bottom: 10px;
}

.firework-description {
    font-size: 14px;
    color: #ccc;
    line-height: 1.4;
}

.score-board {
    background: rgba(0,0,0,0.3);
    padding: 15px;
    border-radius: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.score-item {
    text-align: center;
}

.score-value {
    font-size: 24px;
    font-weight: bold;
    color: #ffd700;
}

.score-label {
```

```
            font-size: 12px;
            color: #ccc;
        }

        .recipes-discovered {
            margin-top: 20px;
            background: rgba(0,0,0,0.3);
            padding: 15px;
            border-radius: 10px;
        }

        .recipe-item {
            display: flex;
            align-items: center;
            justify-content: space-between;
            padding: 8px;
            margin: 5px 0;
            background: rgba(255,255,255,0.1);
            border-radius: 5px;
            font-size: 14px;
        }

        .recipe-discovered {
            color: #90EE90;
        }

        .recipe-unknown {
            color: #666;
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>🎆 花火師の工房 🎆</h1>
        <div class="countdown" id="countdown">花火大会まで: 計算中...</div>
    </div>

    <div class="game-container">
        <div class="score-board">
            <div class="score-item">
                <div class="score-value" id="score">0</div>
                <div class="score-label">スコア</div>
            </div>
            <div class="score-item">
                <div class="score-value" id="discoveries">0</div>
                <div class="score-label">新発見</div>
            </div>
            <div class="score-item">
```

```
            <div class="score-value" id="total-crafted">0</div>
            <div class="score-label">調合数</div>
        </div>
    </div>

    <div class="ingredients-section">
        <div class="section-title">🧪 調合材料</div>
        <div class="ingredients-grid" id="ingredients-grid">
            <!-- 材料がここに動的に追加される -->
        </div>
    </div>

    <div class="ingredients-section">
        <div class="section-title">⚗️ 調合釜</div>
        <div class="mixing-bowl">
            <div class="bowl-contents" id="bowl-contents">
                <div style="color: #666; font-size: 14px;">材料を選んで調合しよう</div>
            </div>
        </div>
        <div class="controls">
            <button class="btn btn-craft" id="craft-btn" onclick="craftFirework()">調合する
</button>
            <button class="btn btn-clear" onclick="clearBowl()">リセット</button>
        </div>
    </div>

    <div class="result-section">
        <div class="section-title">🎇 調合結果</div>
        <div id="result-display">
            <div style="color: #666; font-size: 16px; margin-top: 40px;">
                材料を調合して美しい花火を作ろう！
            </div>
        </div>
    </div>

    <div class="recipes-discovered">
        <div class="section-title">📚 レシピ図鑑</div>
        <div id="recipes-list">
            <!-- レシピがここに表示される -->
        </div>
    </div>
</div>

<script>
    // 游戏数据
    const ingredients = [
        { id: 'potassium', name: '硝酸カリウム', icon: '🧂', type: 'base' },
        { id: 'sulfur', name: '硫黄', icon: '🟡', type: 'base' },
```

```
    { id: 'charcoal', name: '木炭', icon: '⚫', type: 'base' },
    { id: 'copper', name: '銅', icon: '🔵', type: 'color' },
    { id: 'strontium', name: 'ストロンチウム', icon: '🔴', type: 'color' },
    { id: 'barium', name: 'バリウム', icon: '🟢', type: 'color' },
    { id: 'sodium', name: 'ナトリウム', icon: '🟠', type: 'color' },
    { id: 'magnesium', name: 'マグネシウム', icon: '⚪', type: 'special' },
    { id: 'titanium', name: 'チタン', icon: '✨', type: 'special' },
    { id: 'iron', name: '鉄粉', icon: '🟤', type: 'special' }
];

const recipes = [
    {
        name: '基本花火',
        ingredients: ['potassium', 'sulfur', 'charcoal'],
        icon: '🎆',
        description: '伝統的な花火。シンプルな美しさ。',
        points: 10
    },
    {
        name: '青い星',
        ingredients: ['potassium', 'sulfur', 'copper'],
        icon: '💙',
        description: '夜空に輝く青い星のような花火。',
        points: 15
    },
    {
        name: '紅花火',
        ingredients: ['potassium', 'charcoal', 'strontium'],
        icon: '❤️',
        description: '情熱的な赤い花火。愛を込めて。',
        points: 15
    },
    {
        name: '緑の風',
        ingredients: ['potassium', 'sulfur', 'barium'],
        icon: '💚',
        description: '自然の風を感じる緑の花火。',
        points: 15
    },
    {
        name: '黄金花',
        ingredients: ['potassium', 'charcoal', 'sodium'],
        icon: '💛',
        description: '豪華絢爛な黄金色の花火。',
        points: 15
    },
    {
        name: '白銀の輝き',
```

```
        ingredients: ['potassium', 'magnesium', 'titanium'],
        icon: '🤍',
        description: '清楚で美しい白銀の花火。',
        points: 20
    },
    {
        name: '虹色花火',
        ingredients: ['copper', 'strontium', 'barium', 'sodium'],
        icon: '🌈',
        description: '全ての色が混ざった虹色の花火。',
        points: 25
    },
    {
        name: '流星花火',
        ingredients: ['magnesium', 'iron', 'titanium'],
        icon: '☄️',
        description: '流れ星のような軌跡を描く花火。',
        points: 30
    },
    {
        name: '花火玉',
        ingredients: ['potassium', 'sulfur', 'charcoal', 'copper', 'strontium'],
        icon: '🎇',
        description: '大輪の花を咲かせる豪華な花火。',
        points: 35
    },
    {
        name: '究極花火',
        ingredients: ['potassium', 'magnesium', 'copper', 'strontium', 'barium', 'titanium'],
        icon: '🌟',
        description: '全ての技術を結集した究極の花火。',
        points: 50
    }
];

// ゲーム状態
let selectedIngredients = [];
let discoveredRecipes = new Set();
let score = 0;
let totalCrafted = 0;

// 初期化
function initGame() {
    renderIngredients();
    renderRecipesList();
    updateCountdown();
    setInterval(updateCountdown, 1000);
}
```

```
// 材料表示
function renderIngredients() {
    const grid = document.getElementById('ingredients-grid');
    grid.innerHTML = '';

    ingredients.forEach(ingredient => {
        const div = document.createElement('div');
        div.className = 'ingredient';
        div.innerHTML = `
            <div class="ingredient-icon">${ingredient.icon}</div>
            <div class="ingredient-name">${ingredient.name}</div>
        `;
        div.onclick = () => selectIngredient(ingredient.id);
        grid.appendChild(div);
    });
}

// 材料選択
function selectIngredient(ingredientId) {
    if (selectedIngredients.includes(ingredientId)) {
        selectedIngredients = selectedIngredients.filter(id => id !== ingredientId);
    } else if (selectedIngredients.length < 6) {
        selectedIngredients.push(ingredientId);
    }

    updateIngredientSelection();
    updateBowlContents();
}

// 材料選択状態更新
function updateIngredientSelection() {
    const ingredientElements = document.querySelectorAll('.ingredient');
    ingredientElements.forEach((element, index) => {
        const ingredientId = ingredients[index].id;
        if (selectedIngredients.includes(ingredientId)) {
            element.classList.add('selected');
        } else {
            element.classList.remove('selected');
        }
    });
}

// 調合釜内容更新
function updateBowlContents() {
    const bowlContents = document.getElementById('bowl-contents');

    if (selectedIngredients.length === 0) {
```

```
            bowlContents.innerHTML = '<div style="color: #666; font-size: 14px;">材料を選んで調合しよう</div>';
            return;
        }

        bowlContents.innerHTML = '';
        selectedIngredients.forEach(ingredientId => {
            const ingredient = ingredients.find(ing => ing.id === ingredientId);
            const div = document.createElement('div');
            div.className = 'bowl-ingredient';
            div.innerHTML = ingredient.icon;
            div.style.animationDelay = Math.random() * 2 + 's';
            bowlContents.appendChild(div);
        });
    }

    // 花火調合
    function craftFirework() {
        if (selectedIngredients.length === 0) return;

        const sortedIngredients = [...selectedIngredients].sort();
        const recipe = recipes.find(r =>
            r.ingredients.length === sortedIngredients.length &&
            r.ingredients.every(ing => sortedIngredients.includes(ing))
        );

        totalCrafted++;

        if (recipe) {
            displayResult(recipe);
            score += recipe.points;

            if (!discoveredRecipes.has(recipe.name)) {
                discoveredRecipes.add(recipe.name);
                score += recipe.points; // ボーナスポイント
            }
        } else {
            displayFailure();
            score += 5; // 失敗でも少しポイント
        }

        updateScoreBoard();
        renderRecipesList();
        clearBowl();
    }

    // 結果表示（成功）
    function displayResult(recipe) {
```

```
        const resultDisplay = document.getElementById('result-display');
        resultDisplay.innerHTML = `
            <div class="firework-display">${recipe.icon}</div>
            <div class="firework-name">${recipe.name}</div>
            <div class="firework-description">${recipe.description}</div>
            <div style="color: #ffd700; margin-top:
10px;">+${recipe.points}pt${!discoveredRecipes.has(recipe.name) ? ' (新発見ボーナス!)' :
''}</div>
        `;
    }

    // 結果表示（失敗）
    function displayFailure() {
        const resultDisplay = document.getElementById('result-display');
        const failures = [
            { icon: '💨', text: '煙だけが出ました...', desc: '配合を見直してみましょう' },
            { icon: '💥', text: '小爆発！', desc: '危険ですが経験になりました' },
            { icon: '🌫️', text: '霧のような煙', desc: '何かが足りないようです' },
            { icon: '✨', text: '小さな火花', desc: 'もう少しで成功しそうです' }
        ];

        const failure = failures[Math.floor(Math.random() * failures.length)];
        resultDisplay.innerHTML = `
            <div class="firework-display">${failure.icon}</div>
            <div class="firework-name">${failure.text}</div>
            <div class="firework-description">${failure.desc}</div>
            <div style="color: #ffd700; margin-top: 10px;">+5pt (経験値)</div>
        `;
    }

    // 調合釜リセット
    function clearBowl() {
        selectedIngredients = [];
        updateIngredientSelection();
        updateBowlContents();
    }

    // スコアボード更新
    function updateScoreBoard() {
        document.getElementById('score').textContent = score;
        document.getElementById('discoveries').textContent = discoveredRecipes.size;
        document.getElementById('total-crafted').textContent = totalCrafted;
    }

    // レシピリスト表示
    function renderRecipesList() {
        const recipesList = document.getElementById('recipes-list');
        recipesList.innerHTML = '';
```

```
            recipes.forEach(recipe => {
                const div = document.createElement('div');
                div.className = 'recipe-item';

                if (discoveredRecipes.has(recipe.name)) {
                    div.className += ' recipe-discovered';
                    div.innerHTML = `
                        <span>${recipe.icon} ${recipe.name}</span>
                        <span>${recipe.points}pt</span>
                    `;
                } else {
                    div.className += ' recipe-unknown';
                    div.innerHTML = `
                        <span>❓ ???</span>
                        <span>???pt</span>
                    `;
                }

                recipesList.appendChild(div);
            });
        }

        // カウントダウン更新
        function updateCountdown() {
            const now = new Date();
            const today = new Date(now.getFullYear(), now.getMonth(), now.getDate());
            const fireworkTime = new Date(today.getTime() + 19 * 60 * 60 * 1000); // 19:00

            if (now > fireworkTime) {
                fireworkTime.setDate(fireworkTime.getDate() + 1);
            }

            const diff = fireworkTime - now;
            const hours = Math.floor(diff / (1000 * 60 * 60));
            const minutes = Math.floor((diff % (1000 * 60 * 60)) / (1000 * 60));
            const seconds = Math.floor((diff % (1000 * 60)) / 1000);

            document.getElementById('countdown').textContent =
                `花火大会まで: ${hours}時間${minutes}分${seconds}秒`;
        }

        // ゲーム開始
        initGame();
    </script>
</body>
</html>
```