// 新しいグローバル変数

let itemWidthWithMargin = 0;

let initialScrollOffsetValue = 0;

// ... (existing code) ...

/\*\*

 \* ページの読み込み時とリサイズ時にルーレットを初期化します。

 \* アイテムを生成し、リストを中央にスクロールします。

 \*/

function initializeRoulette() {

    createRouletteItems(); // アイテムを生成

    updateButtonStates(); // 初期状態のボタン設定

    const itemElement = document.querySelector('.item');

    if (itemElement) {

        // ここで計算し、グローバル変数にキャッシュ

        itemWidthWithMargin = itemElement.offsetWidth + parseFloat(getComputedStyle(itemElement).marginLeft) + parseFloat(getComputedStyle(itemElement).marginRight);

        initialScrollOffsetValue = (Math.floor(itemListContainer.children.length / items.length / 2)) \* items.length \* itemWidthWithMargin;

        itemListContainer.scrollLeft = initialScrollOffsetValue;

    }

}

/\*\*

 \* スピンアニメーションを開始します。

 \*/

function spin() {

    if (spinning) return;

    // ... (既存のavailableItemsと全アイテム選ばれた時のチェック) ...

    spinning = true;

    spinButton.disabled = true;

    resetButton.disabled = true;

    itemListWrapper.style.pointerEvents = 'none';

    resultDisplay.style.opacity = '0';

    // キャッシュされた値を使用

    itemListContainer.scrollLeft = initialScrollOffsetValue;

    let startScrollLeft = itemListContainer.scrollLeft;

    // ... (selectedOriginalIndexの取得) ...

    const targetSetIndex = (Math.floor(itemListContainer.children.length / items.length) - 1);

    const targetIndexInFullList = targetSetIndex \* items.length + selectedOriginalIndex;

    const targetItem = itemListContainer.children[targetIndexInFullList];

    // ... (targetItemのnullチェックとエラーハンドリング) ...

    const itemStyle = getComputedStyle(targetItem);

    const itemTotalWidth = parseFloat(itemStyle.width) + parseFloat(itemStyle.marginLeft) + parseFloat(itemStyle.marginRight);

    const targetScrollLeft = targetItem.offsetLeft - (itemListWrapper.offsetWidth / 2) + (itemTotalWidth / 2);

    let startTime = null;

    // console.log("スピン開始時のデータ:", { startScrollLeft, targetScrollLeft, itemWidthWithMargin, selectedOriginalIndex, selectedItemName: availableItems[0].name, targetSetIndex, targetItemOffset: targetItem.offsetLeft, selectedItemIndices: [...selectedItemIndices] });

    function animateSpin(currentTime) {

        // ... (アニメーションループのロジックは変更なし) ...

    }

    requestAnimationFrame(animateSpin);

}

// ... (既存のコード) ...