

```
/* TAG SPHERE v1.9.3 - FERNROHR + MOBILE FIX */
#tagSphere {
    position: fixed;
    top: 50%;
    left: 0px;
    transform: translateY(-50%);
    width: 250px;
    height: 250px;
    z-index: 999;
    cursor: move;
    cursor: -webkit-grab;
    cursor: -moz-grab;
    opacity: 0;
    visibility: hidden;
    transition: opacity 0.4s ease, visibility 0.4s ease, width 0.7s
cubic-bezier(0.68, -0.55, 0.265, 1.55), height 0.7s cubic-bezier(0.68,
-0.55, 0.265, 1.55), top 0.7s ease, left 0.7s ease, transform 0.7s ease,
z-index 0s;
    background: transparent;
    border: none;
    user-select: none;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
}
#tagSphere.visible {
    opacity: 1;
    visibility: visible;
}
#tagSphere.exploded {
    width: 96vw;
    height: 94vh;
    max-width: none;
    max-height: none;
    top: 50vh;
    left: 50vw;
    transform: translate(-50%, -50%);
    z-index: 2147483647;
}
@media (orientation: landscape) and (min-width: 1024px) {
    #tagSphere.exploded {
        width: 96vw;
        height: 94vh;
        max-width: none;
        max-height: none;
    }
}
#tagSphereBackdrop {
    position: fixed;
```

```

    top: 0;
    left: 0;
    width: 100vw;
    height: 100vh;
    background: rgba(0, 0, 0, 0.45);
    z-index: 2147483640;
    opacity: 0;
    visibility: hidden;
    pointer-events: none;
    transition: opacity 0.4s ease, visibility 0.4s ease;
}
#tagSphereBackdrop.visible {
    opacity: 1;
    visibility: visible;
    pointer-events: auto;
}
@media (max-width: 768px) {
    #tagSphere {
        width: 250px;
        height: 250px;
        left: 50vw;
        top: 50vh;
        transform: translate(-50%, -50%);
    }
    #tagSphere.exploded {
        width: 100vw !important;
        height: 100vh !important;
        max-width: none !important;
        max-height: none !important;
        top: 50vh !important;
        left: 50vw !important;
        transform: translate(-50%, -50%) !important;
        background: rgba(255, 255, 255, 0.95);
    }
    #tagSphereBackdrop {
        display: none;
    }
}
#tagSphere.dragging {
    cursor: -webkit-grabbing !important;
    cursor: -moz-grabbing !important;
}
/* === CLOSE BUTTON === */
#tagSphere .close-btn {
    position: absolute;
    top: 20px;
    right: 20px;
    width: 56px;
    height: 56px;
    background: rgba(255, 255, 255, 0.95);

```

```

backdrop-filter: blur(10px);
-webkit-backdrop-filter: blur(10px);
color: #CC0000;
border: 2px solid rgba(204, 0, 0, 0.2);
border-radius: 50%;
font-size: 32px;
font-weight: bold;
line-height: 52px;
text-align: center;
cursor: pointer;
z-index: 9999;
opacity: 0;
visibility: hidden;
transition: all 0.3s ease;
user-select: none;
box-shadow: 0 4px 16px rgba(0, 0, 0, 0.15);
}
#tagSphere.exploded .close-btn {
  opacity: 1 !important;
  visibility: visible !important;
}
#tagSphere .close-btn:hover {
  background: #CC0000;
  color: #FFFFFF;
  border-color: #CC0000;
  transform: scale(1.1);
  box-shadow: 0 6px 24px rgba(204, 0, 0, 0.4);
}
#tagSphere .close-btn:active {
  transform: scale(0.95);
}
/* === BACK BUTTON === */
#tagSphere .back-btn {
  position: absolute;
  top: 20px;
  left: 20px;
  width: 56px;
  height: 56px;
  background: rgba(255, 255, 255, 0.95);
  backdrop-filter: blur(10px);
  -webkit-backdrop-filter: blur(10px);
  color: #0066CC;
  border: 2px solid rgba(0, 102, 204, 0.2);
  border-radius: 50%;
  font-size: 28px;
  font-weight: bold;
  line-height: 52px;
  text-align: center;
  cursor: pointer;
  z-index: 9999;

```

```

    opacity: 0;
    visibility: hidden;
    transition: all 0.3s ease;
    user-select: none;
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.15);
}
#tagSphere.mode-pages.exploded .back-btn {
    opacity: 1 !important;
    visibility: visible !important;
}
#tagSphere .back-btn:hover {
    background: #0066CC;
    color: #FFFFFF;
    border-color: #0066CC;
    transform: scale(1.1);
    box-shadow: 0 6px 24px rgba(0, 102, 204, 0.4);
}
#tagSphere .back-btn:active {
    transform: scale(0.95);
}
/* === ITEM COUNT === */
#tagSphere .item-count {
    position: absolute;
    bottom: 8px;
    left: 50%;
    transform: translateX(-50%);
    padding: 5px 16px;
    background: rgba(0, 0, 0, 0.05);
    border-radius: 20px;
    color: #999;
    font-size: 12px;
    font-weight: 600;
    z-index: 9999;
    opacity: 0;
    visibility: hidden;
    transition: all 0.3s ease;
    user-select: none;
    white-space: nowrap;
}
#tagSphere.exploded .item-count {
    opacity: 1;
    visibility: visible;
}
/* === RESPONSIVE BUTTONS === */
@media (orientation: landscape) and (min-width: 1024px) {
    #tagSphere.exploded .close-btn {
        top: 35px;
        right: 45px;
        width: 64px;
        height: 64px;
    }
}

```

```

        font-size: 38px;
        line-height: 60px;
    }
    #tagSphere.exploded .back-btn {
        top: 35px;
        left: 45px;
        width: 64px;
        height: 64px;
        font-size: 34px;
        line-height: 60px;
    }
}
@media (orientation: portrait) {
    #tagSphere .close-btn {
        width: 72px;
        height: 72px;
        font-size: 40px;
        line-height: 68px;
        top: 25px;
        right: 25px;
    }
    #tagSphere .back-btn {
        width: 72px;
        height: 72px;
        font-size: 36px;
        line-height: 68px;
        top: 25px;
        left: 25px;
    }
}
@media (max-width: 768px) and (orientation: portrait) {
    #tagSphere .close-btn {
        width: 80px;
        height: 80px;
        font-size: 48px;
        line-height: 76px;
        top: 20px;
        right: 20px;
        box-shadow: 0 6px 24px rgba(0, 0, 0, 0.25);
    }
    #tagSphere .back-btn {
        width: 80px;
        height: 80px;
        font-size: 42px;
        line-height: 76px;
        top: 20px;
        left: 20px;
        box-shadow: 0 6px 24px rgba(0, 0, 0, 0.25);
    }
}

```

```

@media (max-width: 768px) and (orientation: landscape) {
    #tagSphere .close-btn {
        width: 64px;
        height: 64px;
        font-size: 36px;
        line-height: 60px;
    }
    #tagSphere .back-btn {
        width: 64px;
        height: 64px;
        font-size: 32px;
        line-height: 60px;
    }
}
/* === TAG ITEMS === */
#tagSphere .tag-item {
    position: absolute;
    padding: 18px 32px;
    white-space: nowrap;
    font-size: 20px;
    font-weight: 700;
    text-decoration: none;
    background: #FFFFFF;
    border: none;
    border-radius: 25px;
    cursor: pointer;
    min-width: 70px;
    text-align: center;
    box-shadow: 0 3px 10px rgba(0, 0, 0, 0.15);
    transition: transform 0.2s ease, box-shadow 0.2s ease, color 0.2s
ease, opacity 0.3s ease, filter 0.3s ease;
    pointer-events: auto;
    z-index: 100;
    user-select: none;
}
#tagSphere.items-many .tag-item {
    padding: 12px 22px;
    min-width: 50px;
    border-radius: 20px;
}
@media (max-width: 768px) {
    #tagSphere .tag-item {
        padding: 22px 38px;
        font-size: 22px;
        min-width: 95px;
    }
    #tagSphere.items-many .tag-item {
        padding: 14px 26px;
        min-width: 60px;
    }
}

```

```

}
/* === FLOAT ANIMATIONS === */
#tagSphere .tag-item.atom-0 { animation: float-0 4.2s ease-in-out
infinite; }
#tagSphere .tag-item.atom-1 { animation: float-1 5.1s ease-in-out
infinite; }
#tagSphere .tag-item.atom-2 { animation: float-2 4.7s ease-in-out
infinite; }
#tagSphere .tag-item.atom-3 { animation: float-3 5.5s ease-in-out
infinite; }
#tagSphere .tag-item.atom-4 { animation: float-4 4.9s ease-in-out
infinite; }
#tagSphere .tag-item.atom-5 { animation: float-5 5.3s ease-in-out
infinite; }
#tagSphere .tag-item.atom-6 { animation: float-6 4.5s ease-in-out
infinite; }
#tagSphere .tag-item.atom-7 { animation: float-7 5.8s ease-in-out
infinite; }
@keyframes float-0 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(3px, -4px) rotate(1deg); }
    50% { transform: translate(-3px, -5px) rotate(-1deg); }
    75% { transform: translate(-4px, -2px) rotate(2deg); }
}
@keyframes float-1 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(-3px, -5px) rotate(-1deg); }
    50% { transform: translate(4px, -3px) rotate(1deg); }
    75% { transform: translate(3px, -6px) rotate(-2deg); }
}
@keyframes float-2 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(4px, -3px) rotate(2deg); }
    50% { transform: translate(-3px, -6px) rotate(-1deg); }
    75% { transform: translate(3px, -4px) rotate(1deg); }
}
@keyframes float-3 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(-4px, -6px) rotate(-2deg); }
    50% { transform: translate(3px, -4px) rotate(1deg); }
    75% { transform: translate(-3px, -5px) rotate(-1deg); }
}
@keyframes float-4 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(3px, -5px) rotate(1deg); }
    50% { transform: translate(-4px, -4px) rotate(-2deg); }
    75% { transform: translate(4px, -6px) rotate(2deg); }
}
@keyframes float-5 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }

```

```

    25% { transform: translate(-3px, -4px) rotate(-1deg); }
    50% { transform: translate(4px, -6px) rotate(2deg); }
    75% { transform: translate(-4px, -3px) rotate(-2deg); }
}
@keyframes float-6 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(4px, -6px) rotate(2deg); }
    50% { transform: translate(-3px, -5px) rotate(-1deg); }
    75% { transform: translate(3px, -3px) rotate(1deg); }
}
@keyframes float-7 {
    0%, 100% { transform: translate(0, 0) rotate(0deg); }
    25% { transform: translate(-4px, -4px) rotate(-2deg); }
    50% { transform: translate(3px, -6px) rotate(1deg); }
    75% { transform: translate(-3px, -5px) rotate(-1deg); }
}
/* === TAG COLORS === */
#tagSphere.mode-tags .tag-item {
    color: #0066CC;
}
#tagSphere.mode-tags .tag-item:hover {
    color: #0044AA;
    box-shadow: 0 6px 20px rgba(0, 102, 204, 0.5);
    transform: scale(1.3) !important;
    filter: blur(0px) !important;
    z-index: 9998 !important;
}
#tagSphere.mode-pages .tag-item:hover {
    box-shadow: 0 6px 20px rgba(204, 0, 0, 0.5);
    transform: scale(1.3) !important;
    filter: blur(0px) !important;
    z-index: 9998 !important;
}

(function(){
/* ===== CONFIG ===== */
var CFG = new Object();
CFG.baseRadius = 110;
CFG.smallSize = 250;
CFG.smallSizeMobile = 250;
CFG.baseFontSize = 20;
CFG.baseFontSizeMobile = 22;
CFG.minFontPages = 18;
CFG.maxFontPages = 38;
CFG.minFontPagesMobile = 20;
CFG.maxFontPagesMobile = 40;
CFG.xFactorNormal = 0.4;
CFG.yFactorNormal = 1.5;
CFG.inactiveOpacity = 0.4;
CFG.verticalGradient = 0.6;

```



```

CFG.autoExplodeTrigger = 0.3;
CFG.scrollThreshold = 1000;
CFG.mobileBreak = 768;
CFG.landscapeBreak = 1024;
CFG.dragSense = 0.008;
CFG.moveThreshold = 15;
CFG.clickMaxTime = 500;
CFG.manyItemsThreshold = 20;
/* Fernrohr: Schaerfe-Radius (0-1, Anteil vom halben Viewport) */
CFG.focusRadius = 0.3;
/* Fernrohr: Max Blur in px fuer Items am Rand */
CFG.maxBlur = 3.5;
/* Fernrohr: Scale-Boost fuer Items im Zentrum */
CFG.focusScaleBoost = 1.4;
/* Fernrohr: Scale fuer Items am Rand */
CFG.edgeScale = 0.7;
/* ===== STATE ===== */
var c = null;
var backdrop = null;
var closeBtn = null;
var backBtn = null;
var countLabel = null;
var items = [];
var angleX = 0;
var angleY = 0;
var exploded = false;
var drag = false;
var lastX = 0;
var lastY = 0;
var visible = false;
var loading = false;
var mode = 'tags';
var originalTags = [];
var isMobile = false;
var startX = 0;
var startY = 0;
var hasMoved = false;
var isActive = false;
var isLandscape = false;
var clickStartX = 0;
var clickStartTime = 0;
var navigating = false;
/* ===== HELPERS ===== */
function detectMobile() {
    isMobile = window.innerWidth <= CFG.mobileBreak;
}
function detectLandscape() {
    isLandscape = (window.innerWidth > window.innerHeight) &&
(window.innerWidth >= CFG.landscapeBreak);
}

```

```

function getViewportsSize() {
    if (window.visualViewport) {
        return { width: window.visualViewport.width, height:
window.visualViewport.height };
    }
    return { width: window.innerWidth, height: window.innerHeight };
}
function fixUrl(href) {
    if (!href) return href;
    /* IMMER relative Pfade fuer interne Links zurueckgeben.
    So nutzt der Browser automatisch das aktuelle Protokoll.
    Kein HTTP/HTTPS Mismatch moeglich = keine 304er. */
    href = href.replace(/https?:\/\/\/docs\.trialanderror\.tech/g, '');
    href = href.replace(/https?:\/\/\/172\.22\.84\.184/g, '');
    href = href.replace(/https?:\/\/\/172\.14\.67\.184/g, '');
    /* Falls noch ein anderer Host drin ist, Pfad extrahieren */
    if (href.indexOf('http') === 0) {
        var slashPos = href.indexOf('/', 8);
        if (slashPos > 0) {
            href = href.substring(slashPos);
        }
    }
    /* Sicherstellen dass mit / beginnt */
    if (href.length > 0 && href.charAt(0) !== '/') {
        href = '/' + href;
    }
    return href;
}
function calcExplodedRadius(count) {
    var vp = getViewportsSize();
    var smaller = Math.min(vp.width, vp.height);
    var pct = 0.30 + (Math.sqrt(count) * 0.015);
    if (pct > 0.46) pct = 0.46;
    return smaller * pct;
}
function updateItemCountClass() {
    var cn = c.className;
    cn = cn.replace(/s*items-many/g, '');
    if (items.length > CFG.manyItemsThreshold) {
        cn += ' items-many';
    }
    c.className = cn;
}
function updateCountLabel() {
    if (!countLabel) return;
    if (mode === 'pages') {
        countLabel.textContent = items.length + ' Seiten';
    } else {
        countLabel.textContent = items.length + ' Tags';
    }
}

```

```

}
/* ===== CREATE DOM ===== */
function createContainer() {
    backdrop = document.createElement('div');
    backdrop.id = 'tagSphereBackdrop';
    c = document.createElement('div');
    c.id = 'tagSphere';
    c.className = 'mode-tags';
    closeBtn = document.createElement('button');
    closeBtn.className = 'close-btn';
    closeBtn.textContent = '\u00D7';
    closeBtn.title = 'Close';
    backBtn = document.createElement('button');
    backBtn.className = 'back-btn';
    backBtn.textContent = '\u25C4';
    backBtn.title = 'Back to Tags';
    countLabel = document.createElement('div');
    countLabel.className = 'item-count';
    c.appendChild(closeBtn);
    c.appendChild(backBtn);
    c.appendChild(countLabel);
    var body = document.body || document.getElementsByTagName('body')[0];
    body.appendChild(backdrop);
    body.appendChild(c);
}

/* ===== EXTRACT TAGS ===== */
function extractTags() {
    var box = document.querySelector('.pageTags');
    if (!box) return;
    var links = box.getElementsByTagName('a');
    for (var i = 0; i < links.length; i++) {
        var link = links[i];
        var txt = (link.textContent || link.innerText ||
'').replace(/^\\s+|\\s$/g, '');
        if (txt.indexOf('bearbeiten') !== -1 || txt.indexOf('edit') !==
-1) continue;
        if (txt.length > 0 && link.href) {
            items.push({
                text: txt,
                url: fixUrl(link.href),
                isTag: true,
                views: 0,
                phi: 0,
                theta: 0
            });
        }
    }
    originalTags = items.slice();
    distributeItems();
}

```

```

/* ===== FIBONACCI SPHERE ===== */
function distributeItems() {
    var count = items.length;
    if (count === 0) return;
    if (count === 1) {
        items[0].phi = 0;
        items[0].theta = Math.PI / 2;
        return;
    }
    var goldenAngle = Math.PI * (3 - Math.sqrt(5));
    for (var i = 0; i < count; i++) {
        var y = 1 - (i / (count - 1)) * 2;
        var theta = goldenAngle * i;
        items[i].phi = theta;
        items[i].theta = Math.acos(y);
    }
}

/* ===== CLEAR / CREATE ELEMENTS ===== */
function clearItems() {
    var elements = c.getElementsByClassName('tag-item');
    while (elements.length > 0) {
        c.removeChild(elements[0]);
    }
    items = [];
}

function createElements() {
    for (var i = 0; i < items.length; i++) {
        var item = items[i];
        var span = document.createElement('span');
        span.className = 'tag-item atom-' + (i % 8);
        var icon = item.isTag ? '\uD83D\uDD17 ' : '\uD83D\uDCC4 ';
        span.textContent = icon + item.text;
        span.title = item.text + (item.views > 0 ? ' (' + item.views + '
views)' : '');
        span.setAttribute('data-text', item.text);
        span.setAttribute('data-url', item.url);
        span.setAttribute('data-istag', item.isTag ? '1' : '0');
        span.setAttribute('data-views', item.views || 0);
        if (mode === 'pages') {
            var maxV = 0;
            var minV = 999999999;
            var hasV = false;
            for (var j = 0; j < items.length; j++) {
                if (items[j].views > 0) {
                    hasV = true;
                    if (items[j].views > maxV) maxV = items[j].views;
                    if (items[j].views < minV) minV = items[j].views;
                }
            }
            if (hasV && item.views > 0) {

```

```

        var vRange = maxV - minV;
        if (vRange > 0) {
            var norm = (item.views - minV) / vRange;
            var r = Math.floor(norm * 204);
            span.style.color = 'rgb(' + r + ', 0, 0)';
        } else {
            span.style.color = '#CC0000';
        }
    } else {
        span.style.color = '#000000';
    }
}
c.appendChild(span);
}
updateItemCountClass();
updateCountLabel();
}
/* ===== POSITIONING + FERNROHR EFFEKT ===== */
function position() {
    var elements = c.getElementsByClassName('tag-item');
    if (elements.length === 0) return;
    var vp = getViewportSize();
    var count = items.length;
    var radius, centerX, centerY;
    if (exploded) {
        radius = calcExplodedRadius(count);
        centerX = vp.width / 2;
        centerY = vp.height / 2;
    } else {
        radius = CFG.baseRadius;
        var smallS = isMobile ? CFG.smallSizeMobile : CFG.smallSize;
        centerX = smallS / 2;
        centerY = smallS / 2;
    }
    var xF, yF;
    if (exploded) {
        if (isLandscape) {
            xF = (vp.width / vp.height) * 0.7;
            yF = 0.85;
        } else {
            xF = 0.55;
            yF = (vp.height / vp.width) * 0.55;
        }
    } else {
        xF = CFG.xFactorNormal;
        yF = CFG.yFactorNormal;
    }
    /* Fernrohr: max Distanz = halbe Diagonale des Viewports */
    var maxDist = Math.sqrt(centerX * centerX + centerY * centerY);
    /* Fokus-Zone in Pixeln */

```

```

var focusZone = maxDist * CFG.focusRadius;
for (var i = 0; i < elements.length; i++) {
    if (i >= items.length) break;
    var elem = elements[i];
    var item = items[i];
    var phi = item.phi + angleX;
    var theta = item.theta + (angleY * 0.3);
    var x = radius * Math.sin(theta) * Math.cos(phi) * xF;
    var y = radius * Math.sin(theta) * Math.sin(phi) * yF;
    var z = radius * Math.cos(theta);
    var scale = (z + radius) / (radius * 2);
    scale = Math.max(0.7, Math.min(1.4, scale));
    var baseFS = isMobile ? CFG.baseFontSizeMobile : CFG.baseFontSize;
    var fs = baseFS;
    if (count > CFG.manyItemsThreshold) {
        var shrink = 1.0 - (Math.min(count, 80) -
CFG.manyItemsThreshold) * 0.008;
        if (shrink < 0.65) shrink = 0.65;
        fs = Math.floor(fs * shrink);
    }
    if (mode === 'pages' && item.views > 0) {
        var maxV2 = 0;
        var minV2 = 999999999;
        for (var j = 0; j < items.length; j++) {
            if (items[j].views > 0) {
                if (items[j].views > maxV2) maxV2 = items[j].views;
                if (items[j].views < minV2) minV2 = items[j].views;
            }
        }
        var vR = maxV2 - minV2;
        if (vR > 0) {
            var norm2 = (item.views - minV2) / vR;
            var minF = isMobile ? CFG.minFontPagesMobile :
CFG.minFontPages;
            var maxF = isMobile ? CFG.maxFontPagesMobile :
CFG.maxFontPages;
            fs = Math.floor(minF + (norm2 * (maxF - minF)));
        }
    }
    fs = Math.floor(fs * scale);
    /* === FERNROHR EFFEKT (nur im exploded Modus) === */
    var blurVal = 0;
    var focusScale = 1.0;
    if (exploded) {
        /* Distanz vom Sphere-Zentrum (nicht Viewport-Zentrum) */
        var dist = Math.sqrt(x * x + y * y);
        /* Normalisiert: 0 = Zentrum, 1 = maximale Distanz */
        var normDist = dist / (radius * Math.max(xF, yF));
        if (normDist > 1.0) normDist = 1.0;
        if (normDist < CFG.focusRadius) {

```

```

        /* Im Fokus: scharf + gross */
        focusScale = CFG.focusScaleBoost;
        blurVal = 0;
    } else {
        /* Ausserhalb Fokus: progressiv unscharf + kleiner */
        var edgeFactor = (normDist - CFG.focusRadius) / (1.0 -
CFG.focusRadius);
        if (edgeFactor > 1.0) edgeFactor = 1.0;
        /* Smooth curve: ease-in */
        edgeFactor = edgeFactor * edgeFactor;
        focusScale = CFG.focusScaleBoost - (edgeFactor *
(CFG.focusScaleBoost - CFG.edgeScale));
        blurVal = edgeFactor * CFG.maxBlur;
    }
    fs = Math.floor(fs * focusScale);
}
elem.style.fontSize = fs + 'px';
var approxW = fs * item.text.length * 0.45 + 50;
var approxH = fs + 30;
if (exploded) {
    elem.style.left = Math.floor(centerX + x - approxW / 2) +
'px';
    elem.style.top = Math.floor(centerY + y - approxH / 2) + 'px';
} else {
    elem.style.left = (centerX + x - 40) + 'px';
    elem.style.top = (centerY + y - 20) + 'px';
}
var zI = 100 + Math.floor(scale * 200);
/* Fernrohr: Items im Fokus bekommen hoehere z-index */
if (exploded) {
    zI = zI + Math.floor((1.0 - (blurVal / CFG.maxBlur)) * 500);
}
elem.style.zIndex = zI;
/* Blur anwenden */
if (blurVal > 0.1) {
    elem.style.filter = 'blur(' + blurVal.toFixed(1) + 'px)';
    elem.style.webkitFilter = 'blur(' + blurVal.toFixed(1) +
'px)';
} else {
    elem.style.filter = 'none';
    elem.style.webkitFilter = 'none';
}
var baseO = 0.7 + (scale * 0.3);
var posF = 1.0;
if (!exploded) {
    var cH = isMobile ? CFG.smallSizeMobile : CFG.smallSize;
    var yPos = parseFloat(elem.style.top);
    var relY = yPos / cH;
    posF = 1.0 - (Math.abs(relY - 0.5) * CFG.verticalGradient);
    if (posF < 0.6) posF = 0.6;
}

```

```

        if (posF > 1.0) posF = 1.0;
    }
    var finalO = baseO * posF;
    /* Fernrohr: Rand-Items etwas transparenter */
    if (exploded && blurVal > 0.1) {
        var blurFade = 1.0 - ((blurVal / CFG.maxBlur) * 0.3);
        finalO = finalO * blurFade;
    }
    if (!isActive && !exploded) {
        finalO = finalO * CFG.inactiveOpacity;
    }
    elem.style.opacity = finalO;
}
}
/* ===== EXPLODE / COLLAPSE ===== */
function toggleExplode() {
    exploded = !exploded;
    if (exploded) {
        c.className = c.className + ' exploded';
        backdrop.className = 'visible';
    } else {
        c.className = c.className.replace(/\s*exploded/g, '');
        backdrop.className = '';
        /* Blur zuruecksetzen */
        var elements = c.getElementsByClassName('tag-item');
        for (var i = 0; i < elements.length; i++) {
            elements[i].style.filter = 'none';
            elements[i].style.webkitFilter = 'none';
        }
        if (mode === 'pages') {
            backToTags();
        }
    }
    position();
}
function backToTags() {
    clearItems();
    items = originalTags.slice();
    mode = 'tags';
    c.className = c.className.replace(/mode-\w+/g, '') + ' mode-tags';
    if (exploded) c.className += ' exploded';
    if (visible) c.className += ' visible';
    distributeItems();
    createElements();
    position();
}
/* ===== CLICK HANDLER ===== */
function handleClick(e) {
    e = e || window.event;
    var target = e.target || e.srcElement;

```



```

if (loading) return;
if (hasMoved) return;
if (navigating) return;
if (target === closeBtn) {
    if (exploded) toggleExplode();
    if (e.preventDefault) e.preventDefault();
    return false;
}
if (target === backBtn) {
    backToTags();
    if (e.preventDefault) e.preventDefault();
    return false;
}
if (target === backdrop) {
    if (exploded) toggleExplode();
    if (e.preventDefault) e.preventDefault();
    return false;
}
var clickedTag = false;
var tempT = target;
while (tempT && tempT !== c) {
    if (tempT.className && tempT.className.indexOf('tag-item') !==
-1) {
        clickedTag = true;
        var isTag = tempT.getAttribute('data-istag') === '1';
        var text = tempT.getAttribute('data-text');
        var url = tempT.getAttribute('data-url');
        if (isTag) {
            if (!exploded) {
                /* Minimiert: immer erst Tag-Uebersicht aufklappen */
                toggleExplode();
            } else {
                /* Exploded: Pages fuer diesen Tag laden */
                loadPagesForTag(text);
            }
        } else {
            /* PAGE CLICK: setTimeout verhindert
Touch-Event-Konflikte auf Mobile */
            navigating = true;
            var targetUrl = fixUrl(url);
            setTimeout(function() {
                window.location.href = targetUrl;
            }, 80);
        }
        if (e.preventDefault) e.preventDefault();
        return false;
    }
    tempT = tempT.parentNode;
}
if (!exploded && !clickedTag) {

```

```

        var triggerPos = window.innerWidth * CFG.autoExplodeTrigger;
        if (clickStartX < triggerPos) {
            toggleExplode();
        }
        if (e.preventDefault) e.preventDefault();
        return false;
    }
}
/* ===== LOAD PAGES FOR TAG ===== */
function loadPagesForTag(tagName) {
    loading = true;
    c.className = c.className.replace(/mode-\w+/g, '') + ' loading';
    if (exploded) c.className += ' exploded';
    if (visible) c.className += ' visible';
    var url = '/Special:Tags?tag=' + encodeURIComponent(tagName);
    var xhr = window.XMLHttpRequest ? new XMLHttpRequest() : new
ActiveXObject('Microsoft.XMLHTTP');
    xhr.onreadystatechange = function() {
        if (xhr.readyState === 4) {
            if (xhr.status === 200) {
                parsePages(xhr.responseText);
            } else {
                loading = false;
                c.className = c.className.replace(/loading/g, '') + '
mode-tags';
                if (visible) c.className += ' visible';
            }
        }
    };
    try {
        xhr.open('GET', url, true);
        xhr.send();
    } catch(err) {
        loading = false;
    }
}
function parsePages(html) {
    var tempDiv = document.createElement('div');
    tempDiv.innerHTML = html;
    var specialT = tempDiv.querySelector('#SpecialTags');
    if (!specialT) {
        specialT = tempDiv.querySelector('.wiki-directory') ||
tempDiv.querySelector('table') || tempDiv;
    }
    var newItems = [];
    var links = specialT.getElementsByTagName('a');
    var seen = new Object();
    for (var i = 0; i < links.length; i++) {
        var link = links[i];
        var txt = (link.textContent || link.innerText ||

```

```

    '').replace(/^\s+|\s+$/g, '');
    var href = link.getAttribute('href');
    if (txt.length > 0 && href && !seen[txt]) {
        seen[txt] = true;
        newItem.push({
            text: txt,
            url: fixUrl(href),
            isTag: false,
            views: 0,
            phi: 0,
            theta: 0
        });
    }
}
clearItems();
items = newItem;
distributeItems();
mode = 'pages';
loadPopularity();
}
function loadPopularity() {
    var xhr = window.XMLHttpRequest ? new XMLHttpRequest() : new
ActiveXObject('Microsoft.XMLHTTP');
    xhr.onreadystatechange = function() {
        if (xhr.readyState === 4) {
            if (xhr.status === 200) {
                parsePopularity(xhr.responseText);
            } else {
                finishPageLoad();
            }
        }
    };
    try {
        xhr.open('GET', '/Special:Popularpages', true);
        xhr.send();
    } catch(err) {
        finishPageLoad();
    }
}
function parsePopularity(html) {
    var tempDiv = document.createElement('div');
    tempDiv.innerHTML = html;
    var rows = tempDiv.getElementsByTagName('tr');
    var viewData = new Object();
    for (var i = 0; i < rows.length; i++) {
        var cells = rows[i].getElementsByTagName('td');
        if (cells.length >= 2) {
            var link = cells[0].getElementsByTagName('a')[0];
            if (link) {
                var pageName = (link.textContent || link.innerText ||

```

```

').replace(/^\s+|\s+$/g, '');
        var viewText = (cells[1].textContent ||
cells[1].innerText || '').replace(/^\s+|\s+$/g, '');
        var views = parseInt(viewText.replace(/^[^0-9]/g, ''), 10);
        if (pageName && !isNaN(views)) {
            viewData[pageName] = views;
        }
    }
}
for (var j = 0; j < items.length; j++) {
    if (viewData[items[j].text]) {
        items[j].views = viewData[items[j].text];
    }
}
finishPageLoad();
}
function finishPageLoad() {
    createElements();
    position();
    loading = false;
    var cn = 'mode-pages exploded';
    if (visible) cn += ' visible';
    c.className = cn;
    updateItemCountClass();
}
/* ===== SCROLL VISIBILITY ===== */
function updateVisibility() {
    var scrollTop = window.pageYOffset ||
document.documentElement.scrollTop || document.body.scrollTop || 0;
    if (scrollTop > CFG.scrollThreshold && !visible) {
        c.className += ' visible';
        visible = true;
    } else if (scrollTop <= CFG.scrollThreshold && visible) {
        c.className = c.className.replace(/\s*visible/g, '');
        visible = false;
    }
}
/* ===== ACTIVATE / DEACTIVATE ===== */
function activate() {
    isActive = true;
    position();
}
function deactivate() {
    isActive = false;
    position();
}
/* ===== DRAG HANDLERS ===== */
function down(e) {
    if (loading) return;

```

```

    activate();
    var touch = e.touches ? e.touches[0] : e;
    startX = touch.clientX || touch.pageX || 0;
    startY = touch.clientY || touch.pageY || 0;
    clickStartX = startX;
    clickStartTime = new Date().getTime();
    lastX = startX;
    lastY = startY;
    hasMoved = false;
    drag = true;
    if (e.preventDefault) e.preventDefault();
    return false;
}
function move(e) {
    if (!drag) return;
    var touch = e.touches ? e.touches[0] : e;
    var x = touch.clientX || touch.pageX || 0;
    var y = touch.clientY || touch.pageY || 0;
    var deltaX = Math.abs(x - startX);
    var deltaY = Math.abs(y - startY);
    if (deltaX > CFG.moveThreshold || deltaY > CFG.moveThreshold) {
        hasMoved = true;
        var cn = c.className;
        if (cn.indexOf('dragging') === -1) {
            c.className = cn + ' dragging';
        }
        angleX += (x - lastX) * CFG.dragSense;
        angleY += (y - lastY) * CFG.dragSense;
        position();
    }
    lastX = x;
    lastY = y;
    if (e.preventDefault) e.preventDefault();
    return false;
}
function up(e) {
    if (!drag) return;
    drag = false;
    c.className = c.className.replace(/\s*dragging/g, '');
    var elapsed = new Date().getTime() - clickStartTime;
    if (!hasMoved && elapsed < CFG.clickMaxTime) {
        handleClick(e);
    }
}
function handleMouseEnter() {
    activate();
}
function handleMouseLeave() {
    if (!drag) {
        deactivate();
    }
}

```

```

    }
}
/* ===== RESIZE ===== */
function handleResize() {
    detectMobile();
    detectLandscape();
    position();
}
/* ===== BIND EVENTS ===== */
function bind() {
    if (backdrop.addEventListener) {
        backdrop.addEventListener('click', handleClick, false);
    } else if (backdrop.attachEvent) {
        backdrop.attachEvent('onclick', handleClick);
    }
    if (c.addEventListener) {
        c.addEventListener('mousedown', down, false);
        c.addEventListener('touchstart', down, false);
        c.addEventListener('mouseenter', handleMouseEnter, false);
        c.addEventListener('mouseleave', handleMouseLeave, false);
        document.addEventListener('mousemove', move, false);
        document.addEventListener('touchmove', move, false);
        document.addEventListener('mouseup', up, false);
        document.addEventListener('touchend', up, false);
    } else if (c.attachEvent) {
        c.attachEvent('onmousedown', down);
        c.attachEvent('onmouseenter', handleMouseEnter);
        c.attachEvent('onmouseleave', handleMouseLeave);
        document.attachEvent('onmousemove', move);
        document.attachEvent('onmouseup', up);
    }
    if (window.addEventListener) {
        window.addEventListener('scroll', updateVisibility, false);
        window.addEventListener('resize', handleResize, false);
        if (window.visualViewport) {
            window.visualViewport.addEventListener('resize',
handleResize, false);
        }
    } else if (window.attachEvent) {
        window.attachEvent('onscroll', updateVisibility);
        window.attachEvent('onresize', handleResize);
    }
}
/* ===== INIT ===== */
function init() {
    try {
        detectMobile();
        detectLandscape();
        createContainer();
        extractTags();
    }
}

```

```

        if (items.length === 0) {
            c.style.display = 'none';
            backdrop.style.display = 'none';
            return;
        }
        createElements();
        position();
        bind();
        updateVisibility();
    } catch(err) {
        if (window.console && console.error) {
            console.error('TagSphere init error:', err);
        }
    }
}
if (document.readyState === 'complete') {
    init();
} else if (document.addEventListener) {
    document.addEventListener('DOMContentLoaded', init, false);
} else {
    window.onload = init;
}
})();

```