

```

/**
 * Created by IntelliJ IDEA.
 *
 * User: phil
 * Date: 15/11/12
 * Time: 11:04 AM
 *
 */
(function ($) {
    var self = this, container, running=false, currentY = 0, targetY = 0, oldY = 0, maxScrollTop= 0, minScrollTop, direction,
    onRenderCallback=null,
        fricton = 0.95, // higher value for slower deceleration
        vy = 0,
        stepAmt = 1,
        minMovement= 0.1,
        ts=0.1;

    var updateScrollTarget = function (amt) {
        targetY += amt;
        vy += (targetY - oldY) * stepAmt;

        oldY = targetY;
    }
    var render = function () {
        if (vy < -(minMovement) || vy > minMovement) {

```

```

    currentY = (currentY + vy);
    if (currentY > maxScrollTop) {
        currentY = vy = 0;
    } else if (currentY < minScrollTop) {
        vy = 0;
        currentY = minScrollTop;
    }

    container.scrollTop(-currentY);

    vy *= fricton;

    //  vy += ts * (currentY-targetY);
    //  scrollTopTweened += settings.tweenSpeed * (scrollTop - scrollTopTweened);
    //  currentY += ts * (targetY - currentY);

    if(onRenderCallback){
        onRenderCallback();
    }
}

var animateLoop = function () {
    if(! running)return;
    requestAnimFrame(animateLoop);
    render();
    //log("45","animateLoop","animateLoop", "",stop);

```

```

}
var onWheel = function (e) {
    e.preventDefault();
    var evt = e.originalEvent;

    var delta = evt.detail ? evt.detail * -1 : evt.wheelDelta / 40;
    var dir = delta < 0 ? -1 : 1;
    if (dir != direction) {
        vy = 0;
        direction = dir;
    }

    //reset currentY in case non-wheel scroll has occurred (scrollbar drag, etc.)
    currentY = -container.scrollTop();

    updateScrollTarget(delta);
}

/*
 * http://paulirish.com/2011/requestanimationframe-for-smart-animating/
 */
window.requestAnimationFrame = (function () {
    return window.requestAnimationFrame ||
        window.webkitRequestAnimationFrame ||
        window.mozRequestAnimationFrame ||
        window.oRequestAnimationFrame ||

```

```

        window.msRequestAnimationFrame ||
        function (callback) {
            window.setTimeout(callback, 1000 / 60);
        };

    })();

    /*
     * http://jsbin.com/iqafek/2/edit
     */
    var normalizeWheelDelta = function () {
        // Keep a distribution of observed values, and scale by the
        // 33rd percentile.
        var distribution = [], done = null, scale = 30;
        return function (n) {
            // Zeroes don't count.
            if (n == 0) return n;
            // After 500 samples, we stop sampling and keep current factor.
            if (done != null) return n * done;
            var abs = Math.abs(n);
            // Insert value (sorted in ascending order).
            outer: do { // Just used for break goto
                for (var i = 0; i < distribution.length; ++i) {
                    if (abs <= distribution[i]) {
                        distribution.splice(i, 0, abs);
                    }
                }
            } while (distribution.length < 500);
            done = scale / distribution[33];
        };
    };

```

```

        break outer;
    }
}
distribution.push(abs);
} while (false);
// Factor is scale divided by 33rd percentile.
var factor = scale / distribution[Math.floor(distribution.length / 3)];
if (distribution.length == 500) done = factor;
return n * factor;
};
}());

```

```

$.fn.smoothWheel = function () {
    // var args = [].splice.call(arguments, 0);
    var options = jQuery.extend({}, arguments[0]);
    return this.each(function (index, elm) {

        if(!('ontouchstart' in window)){
            container = $(this);
            container.bind("mousewheel", onWheel);
            container.bind("DOMMouseScroll", onWheel);

            //set target/old/current Y to match current scroll position to prevent jump to top of container
            targetY = oldY = container.get(0).scrollTop;
            currentY = -targetY;

```

```
minScrollTop = container.get(0).clientHeight - container.get(0).scrollHeight;
if(options.onRender){
    onRenderCallback = options.onRender;
}
if(options.remove){
    log("122","smoothWheel","remove", "");
    running=false;
    container.unbind("mousewheel", onWheel);
    container.unbind("DOMMouseScroll", onWheel);
}else if(!running){
    running=true;
    animateLoop();
}

}

});

};

})(jQuery);
```