

JavaScript Library for Mobile Touch Events

HAMMER.JS -

"Hammer is an open-source library that can recognize gestures made by touch, mouse, and pointer events." -
hammerjs.github.io

Enables us to make our own custom touch events & easy to code.

Make touch gestures to easy to Handle -
First. Install the HammerJs Library using

```
npm install Hammer  
  
// and just include it in your Project
```

And you are good to get your app Hammered 😁

Usage - we need to create a new instance of Hammer Class

```
var element = document.getElementById('something');  
var hammertime = new Hammer(element, myOptions);  
hammertime.on('<_gesture_name>', function(ev) {  
    console.log(ev);  
});
```

Lets Talk about and implement some of the default gestures -

1. Tap

Taps are similar to click events, however, the biggest difference is that a tap event is called earlier than standard click events. By using Hammer you can eliminate the famous 300 ms delay on mobile devices.

You can also implement multiple taps simply by configuring the taps option and specify the intervals

between them. Double tapping to like an image on Instagram is perhaps the best example of this.

2. Press

Press is recognized when we hold our finger or pointer for a certain amount of time. This is usually used to pop up a contextual menu or more options for some item. For example, if you press the saved articles in an app like Pocket or a chat in WhatsApp, it brings out more options for the saved article or the chat. You may implement something similar in your web application.

Again, you can configure the threshold and the amount of time in milliseconds that recognizes the event.

3. HorizontalPan -

Pan is a superset of Swipe as it may involve translational movement in any direction. It can be used to move across a zoomed in or a large view. Moving across the map in Google Maps is a perfect example of using Pan touch gesture.

You can alter the recognized direction which is set to `DIRECTION_ALL` by default and can also control the sensitivity by modifying its threshold.

4. Swipe

Swipe identifies translational movements in horizontal and vertical directions. This can be used to change or switch between different views such as moving across tabs, dismissing cards in Google Now or swiping right or left in Tinder.

By default, only horizontal swipes are recognized by Hammer. You may configure the direction, threshold and the velocity of the swipes.

ON Multi-Touch

1. Pinch

A pinch is a multi-touch gesture which involves at least two touch points moving towards or away from each other. Again, Google Maps is a perfect example of this. When we zoom in or out using our fingers, the pinch gesture is being utilized.

You can configure the number of touch points and their sensitivity by changing pointers and threshold options.

2. Rotat

Rotate is recognized when multiple touch points rotate around each other in a circular motion. Image gallery and Google Maps are the best examples of this.

This gesture is disabled by default in Hammer. You may configure the number of touch points and threshold.

Pinch and Rotate events are by **disabled by default**.

We can easily enable or disable any touch gesture easily -

```
hammer.get('pinch').set({ enable: true });  
hammer.get('rotate').set({ enable: true });
```

SOME ADVANTAGES OF USING HAMMER -

- Very Easy integration with JQuery.
- Load time is very Low because the script is very less in size (2kB) in minified version
- Library only for touch and multitouch.

- Enables using specific options for a plugin - like specifying different times for hold down (2sec / .5sec different for different elements) to do some functionality.
- Setting swipe time (Swipe Speed) for a functionality to occur.
- In tapping - we can specify the number of times user must tap for some functionality to occur.
- And other countless gestures can be designed as it can come in our broad imagination.

NOW what makes **Hammer most attractive** is that - we can custom create a new set of tough gestures and apply it to a specific element.

```
var mc = new Hammer.Manager(myElement, myOptions);
mc.add( new Hammer.Pan({ direction: Hammer.DIRECTION_ALL,
threshold: 0 }) );
mc.add( new Hammer.Tap({ event: 'quadrupletap', taps: 4
}) );
mc.add( new Hammer.Tap({ event: 'tripletap', taps: 3 })
);
mc.on("pan", handlePan);
mc.on("quadrupletap", handleQuadrupleTaps);
mc.on("tripletap", handleTripleTaps);
// we have default doubletap and singletap in Hammer so
no need to define new events for that
mc.on("doubletap", handleDoubleTaps);
mc.on("tap", handleSingleTap);

handlePan(){
    .... implement some functionality
}

handleQuadrupleTaps(){
    ....implement some functionality
}
```

```
}

handleTripleTaps(){
    ....implement some functionality
}
handleDoubleTaps(){
    ....implement some functionality
}
handleSingleTap(){
    ....implement some functionality
}
```

Now let's look at a demo project using this -

Using HAMMER in CircuitVerse -

- On long Press Zooming the area of the the circuit concerned . (We can customize the press time for better functionality)

A simple zoom in functionality i coded in Codepen

<https://codepen.io/Devartstar/pen/WNRGOJp>

- Swipe events to move the circuit Elements easily -

```

$(function () {
    var element =
document.getElementById("parent");
    var mc = new Hammer(element);
    mc.get('swipe').set({ direction:
Hammer.DIRECTION_ALL });

    mc.on("swipeleft", function () {
        alert('swipeleft');
    });

    mc.on("swiperight", function () {
        alert('swiperight');
    });

    mc.on("swipeup", function () {
        alert('swipeup');
    });

    mc.on("swipedown", function () {
        alert('swipedown');
    });
})

```

- **Tap events** : We can customize different event functionality with different number of taps -

```

var square = document.querySelector('.square');
// Create a manager to manage the element.
var manager = new Hammer.Manager(square);

```

```

// Create a recognizer.
var nTaps = new Hammer.Tap({
  event: 'ntaps',
  taps: <n> //give a value n
});

// Add the recognizer to the manager.
manager.add(nTaps);

// Subscribe to the event.
manager.on('ntaps', function(e) {
  e.target.classList...// doing something \
  console.log("You're n times tapping me!");
  console.log(e);
});

```

1 tap - select (highlight the circuit element)

2-tap - Zoom in and Zoom Out

3-tap - Rotate 90 degrees.

Long press - context-menu options for the circuit element

Press and move - to move circuit elements from one place to another

Rotate - rotate the circuit element

Swipe-left ||| Swipe-right - the widget option slides in and on choosing an element the context-menu slides out.

PROJECT --- Working on -

Hammer JS sidebar In code pen

<https://codepen.io/Devartstar/full/PoWGXVE>

