

The image is a title screen for the video game "Death Stranding". It features a dramatic, high-contrast landscape. The foreground and middle ground are dominated by dark, silhouetted mountains and rolling hills, with some patches of green grass visible. The sky is filled with large, billowing white and grey clouds, with sunlight breaking through in several places, creating a hazy, atmospheric effect. A thin, faint rainbow is visible in the sky, arching over the mountains. The title "DEATH STRANDING" is centered in the middle of the image, written in a clean, white, sans-serif font. The overall mood is somber and mysterious.

DEATH STRANDING

Generic Sensor API

Koa

Koa

Backbone

Koa

React

Koa webpack

React

Koa webpack

babel

React

Koa webpack

babel

React

typescr

ipt

Redux

Koa

webpack

Babel

React

Typescript

ipt

Redux

Babel

Koa

Webpack

redux-saga

React

typescr

ipt

Kredux

babel

redux-saga

webpack

React

typescr

RxJS

ipt

Kredux
Koa webpack
redux-saga babel
React typescr
RxJS socket ioipt

redux babble
WASTED

redux-socket
React typescr
RxJS socket ioipt

Generic Sensor API

AbsoluteOrientationSensor

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Accelerometer

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Accelerometer

LinearAccelerationSensor

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Accelerometer

LinearAccelerationSensor

Gyroscope

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Accelerometer

LinearAccelerationSensor

Gyroscope

AmbientLightSensor

Generic Sensor API

AbsoluteOrientationSensor

RelativeOrientationSensor

Accelerometer

LinearAccelerationSensor

Gyroscope

AmbientLightSensor

Magnetometer



```
Promise.all([
  navigator.permissions.query({ name: 'accelerometer' }),
  navigator.permissions.query({ name: 'magnetometer' }),
  navigator.permissions.query({ name: 'gyroscope' }),
  navigator.permissions.query({ name: 'ambient-light-sensor' }),
  navigator.permissions.query({ name: 'geolocation' }),
]).then(results => {
  if (result.state == 'granted') {
    // Действуем
  } else if (result.state == 'prompt') {
    // Запрашиваем
  }
}).catch((error) => {
  // Что-то пошло не так
});
```



```
const options = { frequency: 60, referenceFrame: 'device' };

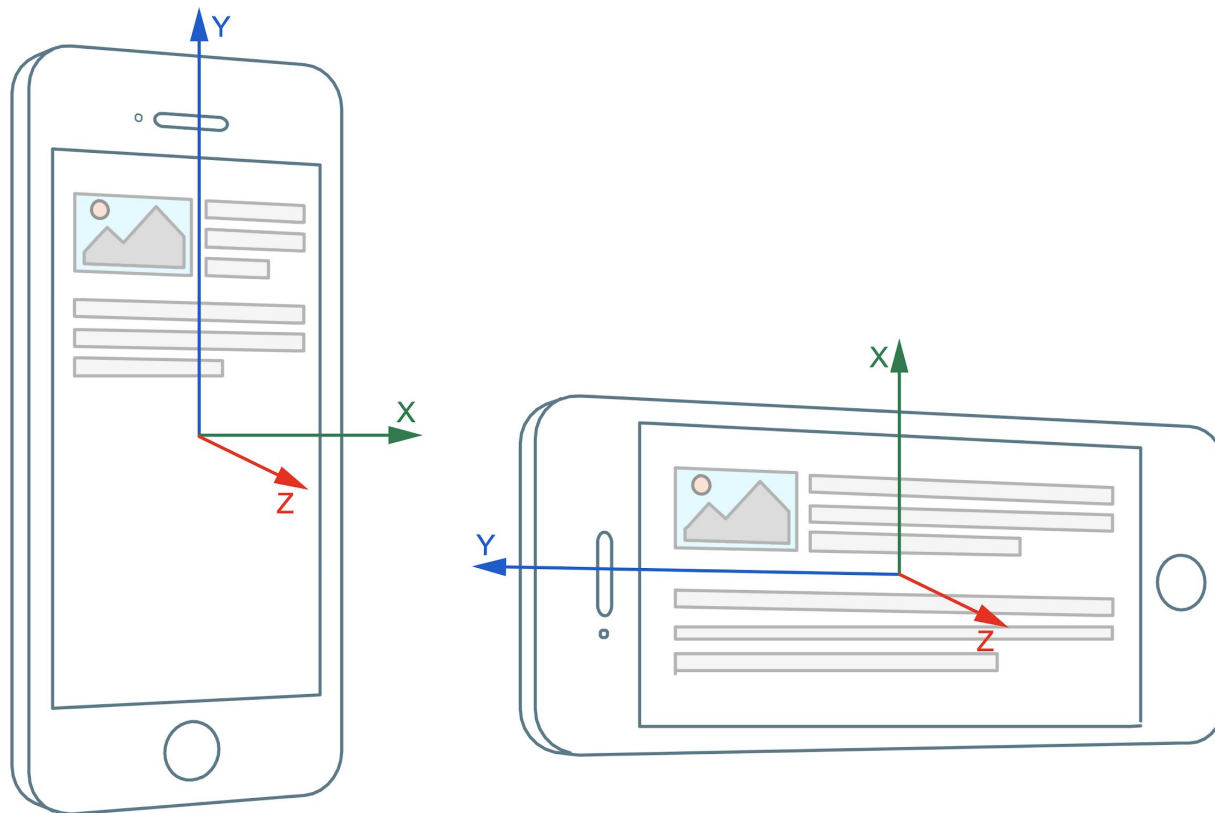
const sensor = new RelativeOrientationSensor(options);

sensor.addEventListener('reading', () => {
  sendSomehowSomewhere({
    quaternion: sensor.quaternion,
  });
});

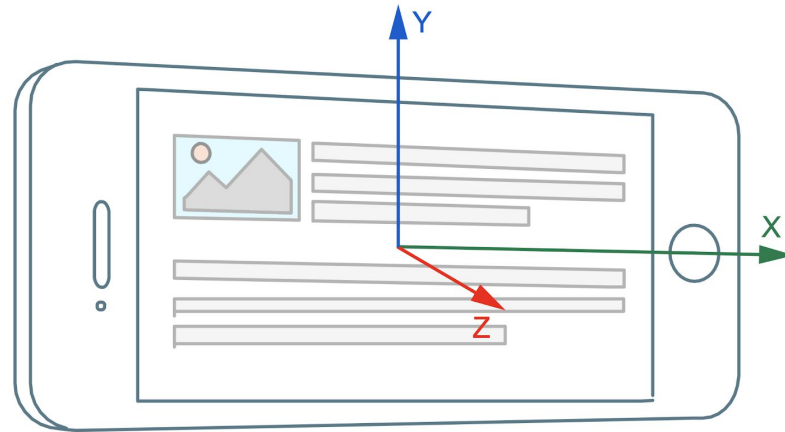
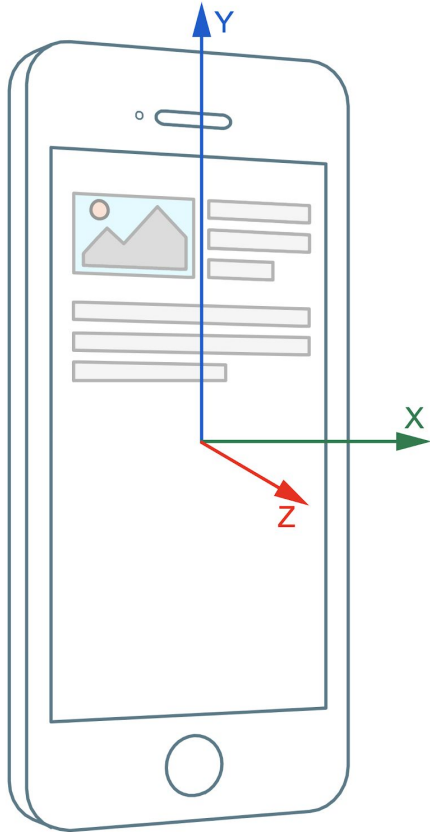
sensor.addEventListener('error', (error: SensorErrorEvent) => {
  console.log(error.error.message);
});

sensor.start();
```

Device coordinate system



Screen coordinate system





```
const options = { frequency: 60, referenceFrame: 'device' };

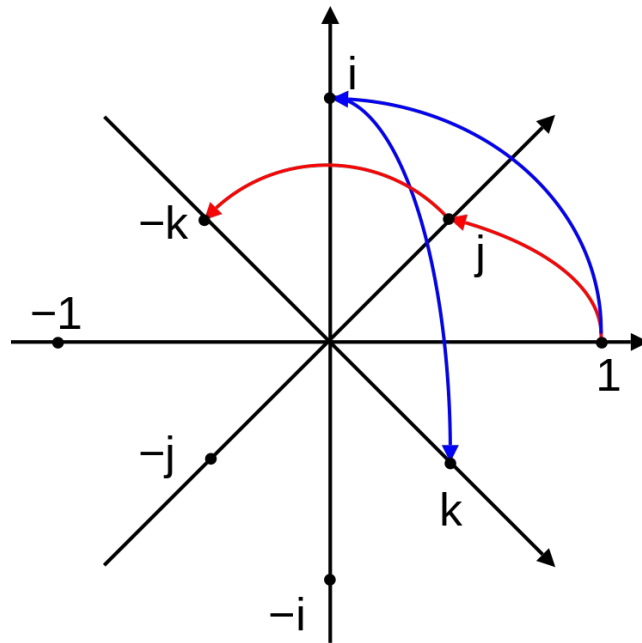
const sensor = new RelativeOrientationSensor(options);

sensor.addEventListener('reading', () => {
  sendSomehowSomewhere({
    quaternion: sensor.quaternion,
  });
});

sensor.addEventListener('error', (error: SensorErrorEvent) => {
  console.log(error.error.message);
});

sensor.start();
```

Кватернион



$$ij = k$$

$$ji = -k$$

$$ij = -ji$$



```
const options = { frequency: 60, referenceFrame: 'device' };

const sensor = new Accelerometer(options);

sensor.addEventListener('reading', () => {
  sendSomehowSomewhere({
    x: sensor.x,
    y: sensor.y,
    z: sensor.z,
  });
});

sensor.addEventListener('error', (error: SensorErrorEvent) => {
  console.log(error.error.message);
});

sensor.start();
```



```
const options = { frequency: 60 };

const sensor = new Gyroscope(options);

sensor.addEventListener('reading', () => {
  sendSomehowSomewhere({
    x: sensor.x,
    y: sensor.y,
    z: sensor.z,
  });
});

sensor.addEventListener('error', (error: SensorErrorEvent) => {
  console.log(error.error.message);
});

sensor.start();
```



Experiments

78.0.3904.108

Available

Unavailable

Generic Sensor Extra Classes

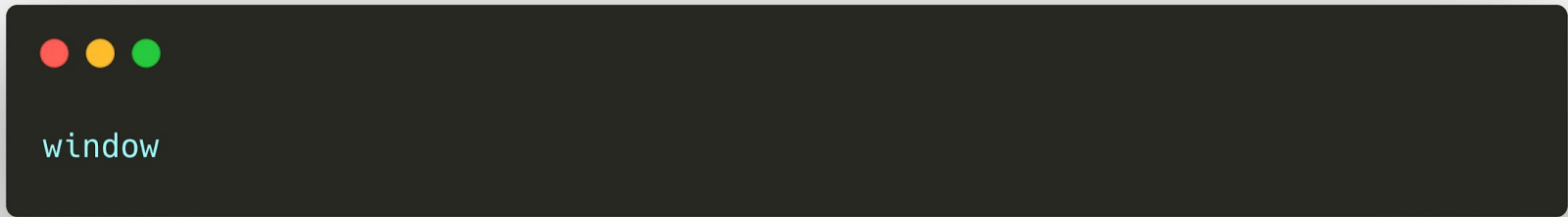
Enables an extra set of sensor classes based on Generic Sensor API, which expose previously unavailable platform features, i.e. AmbientLightSensor and Magnetometer interfaces. – Mac, Windows, Linux, Chrome OS, Android

[#enable-generic-sensor-extra-classes](#)

Default

Your changes will take effect the next time you relaunch Google Chrome.

Relaunch





```
window.addEventListener("deviceorientation", handleOrientation, true);
```

```
function handleOrientation(event) {  
  const absolute = event.absolute;  
  const alpha    = event.alpha;  
  const beta     = event.beta;  
  const gamma    = event.gamma;  
}
```




```
window.addEventListener("devicemotion", handleMotionEvent, true);
```

```
function handleMotionEvent(event) {  
    const x = event.accelerationIncludingGravity.x;  
    const y = event.accelerationIncludingGravity.y;  
    const z = event.accelerationIncludingGravity.z;  
}
```


Вопросы?