

orbitControl()

Allows the user to orbit around a 3D sketch using a mouse, trackpad, or touchscreen.

3D sketches are viewed through an imaginary camera. Calling `orbitControl()` within the `draw()` function allows the user to change the camera's position:

```
function draw() {
  background(200);

  // Enable orbiting with the mouse.
  orbitControl();

  // Rest of sketch.
}
```

Left-clicking and dragging or swipe motion will rotate the camera position about the center of the sketch. Right-clicking and dragging or multi-swipe will pan the camera position without rotation. Using the mouse wheel (scrolling) or pinch in/out will move the camera further or closer from the center of the sketch.

The first three parameters, `sensitivityX`, `sensitivityY`, and `sensitivityZ`, are optional. They're numbers that set the sketch's sensitivity to movement along each axis. For example, calling `orbitControl(1, 2, -1)` keeps movement along the x-axis at its default value, makes the sketch twice as sensitive to movement along the y-axis, and reverses motion along the z-axis. By default, all sensitivity values are 1.

The fourth parameter, `options`, is also optional. It's an object that changes the behavior of orbiting. For example, calling `orbitControl(1, 1, 1, options)` keeps the default sensitivity values while changing the behaviors set with `options`. The object can have the following properties:

```
let options = {
  // Setting this to false makes mobile interactions smoother by
  // preventing accidental interactions with the page while orbiting.
  // By default, it's true.
  disableTouchActions: true,

  // Setting this to true makes the camera always rotate in the
  // direction the mouse/touch is moving.
  // By default, it's false.
  freeRotation: false
};

orbitControl(1, 1, 1, options);
```

Examples

```
// Click and drag the mouse to view the scene from different angles.

function setup() {
  createCanvas(100, 100, WEBGL);

  describe('A multicolor box on a gray background. The camera angle changes when the user interacts using a mouse, trackpad, or touchscreen.');
```

```
function draw() {
  background(200);

  // Enable orbiting with the mouse.
  orbitControl();

  // Style the box.
  normalMaterial();

  // Draw the box.
  box(30, 50);
}
```

```
// Click and drag the mouse to view the scene from different angles.

function setup() {
  createCanvas(100, 100, WEBGL);

  describe('A multicolor box on a gray background. The camera angle changes when the user interacts using a mouse, trackpad, or touchscreen.');
```

```
function draw() {
  background(200);

  // Enable orbiting with the mouse.
  // Make the interactions 3X sensitive.
  orbitControl(3, 3, 3);

  // Style the box.
  normalMaterial();

  // Draw the box.
  box(30, 50);
}
```

```
// Click and drag the mouse to view the scene from different angles.

function setup() {
  createCanvas(100, 100, WEBGL);

  describe('A multicolor box on a gray background. The camera angle changes when the user interacts using a mouse, trackpad, or touchscreen.');
```

```
function draw() {
  background(200);

  // Create an options object.
  let options = {
    disableTouchActions: false,
    freeRotation: true
  };

  // Enable orbiting with the mouse.
  // Prevent accidental touch actions on touchscreen devices
  // and enable free rotation.
  orbitControl(1, 1, 1, options);

  // Style the box.
  normalMaterial();

  // Draw the box.
  box(30, 50);
}
```

Syntax

```
orbitControl([sensitivityX], [sensitivityY], [sensitivityZ], [options])
```

Parameters

<code>sensitivityX</code>	Number: sensitivity to movement along the x-axis. Defaults to 1.
<code>sensitivityY</code>	Number: sensitivity to movement along the y-axis. Defaults to 1.
<code>sensitivityZ</code>	Number: sensitivity to movement along the z-axis. Defaults to 1.
<code>options</code>	Object: object with two optional properties, <code>disableTouchActions</code> and <code>freeRotation</code> . Both are Booleans. <code>disableTouchActions</code> defaults to <code>true</code> and <code>freeRotation</code> defaults to <code>false</code> .

This page is generated from the comments in `src/webgl/interaction.js`. Please feel free to edit it and submit a pull request!

Related References

debugMode Adds a grid and an axes icon to clarify orientation in 3D sketches.	noDebugMode Turns off <code>debugMode()</code> in a 3D sketch.	orbitControl Allows the user to orbit around a 3D sketch using a mouse, trackpad, or touchscreen.
---	--	---

