

slerp()

Sets the camera's position and orientation to values that are in-between those of two other cameras.

`myCamera.slerp()` uses spherical linear interpolation to calculate a position and orientation that's in-between two other cameras. Doing so is helpful for transitioning smoothly between two perspectives.

The first two parameters, `cam0` and `cam1`, are the `p5.Camera` objects that should be used to set the current camera.

The third parameter, `amt`, is the amount to interpolate between `cam0` and `cam1`. 0.0 keeps the camera's position and orientation equal to `cam0`'s, 0.5 sets them halfway between `cam0`'s and `cam1`'s, and 1.0 sets the position and orientation equal to `cam1`'s.

For example, calling `myCamera.slerp(cam0, cam1, 0.1)` sets cam's position and orientation very close to `cam0`'s. Calling `myCamera.slerp(cam0, cam1, 0.9)` sets cam's position and orientation very close to `cam1`'s.

Note: All of the cameras must use the same projection.

Examples



```
let cam;
let cam0;
let cam1;

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

  // Create the main camera.
  // Keep its default settings.
  cam = createCamera();

  // Create the first camera.
  // Keep its default settings.
  cam0 = createCamera();

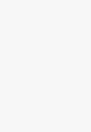
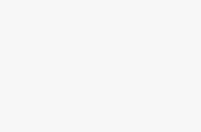
  // Create the second camera.
  cam1 = createCamera();

  // Place it at the top-right.
  cam1.setPosition(400, -400, 800);

  // Point it at the origin.
  cam1.lookAt(0, 0, 0);

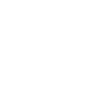
  // Set the current camera to cam.
  setCamera(cam);

  describe('A white cube drawn against a gray background. The camera slowly oscillates between a frontal view and an aerial view.');
}
```



Syntax

```
slerp(cam0, cam1, amt)
```



Parameters

<code>cam0</code>	<code>p5.Camera</code> : first camera.
<code>cam1</code>	<code>p5.Camera</code> : second camera.
<code>amt</code>	Number: amount of interpolation between 0.0 (<code>cam0</code>) and 1.0 (<code>cam1</code>).

This page is generated from the comments in [src/webgl/p5.Camera.js](#). Please feel free to edit it and submit a pull request!

Related References

`camera`
Sets the position and orientation of the camera.

`centerX`
The x-coordinate of the place where the camera looks.

`centerY`
The y-coordinate of the place where the camera looks.

`centerZ`
The z-coordinate of the place where the camera looks.

