specularColor()

Sets the specular color for lights.

`specularColor()` affects lights that bounce off a surface in a preferred direction. These lights include directionalLight(), pointLight(), and spotLight(). The function helps to create highlights on p5.Geometry objects that are styled with specularMaterial(). If a geometry does not use specularMaterial(), then `specularColor()` will have no effect.

Note: `specularColor()` doesn't affect lights that bounce in all directions, including ambientLight() and imageLight().

There are three ways to call `specularColor()` with optional parameters to set the specular highlight color.

The first way to call `specularColor()` has two optional parameters, `gray` and `alpha`. Grayscale and alpha values between 0 and 255, as in `specularColor(50)` or `specularColor(50, 80)`, can be passed to set the specular highlight color.

The second way to call `specularColor()` has one optional parameter, `color`. A p5.Color object, an array of color values, or a CSS color string can be passed to set the specular highlight color.

The third way to call `specularColor()` has four optional parameters, `v1`, `v2`, `v3`, and `alpha`. RGBA, HSBA, or HSLA values, as in `specularColor(255, 0, 0, 80)`, can be passed to set the specular highlight color. Color values will be interpreted using the current colorMode().

Examples

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

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

  describe('A white sphere drawn on a gray background.');
}

function draw() {
  background(200);

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

  // No specular color.
  // Draw the sphere.
  sphere(30);
}
```

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to add a point light.

let isLit = false;

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

  describe('A sphere drawn on a gray background. A spotlight
starts shining when the user double-clicks.');
}

function draw() {
  background(200);

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

  // Style the sphere.
  noStroke();
  specularColor(100);
  specularMaterial(255, 255, 255);

  // Control the light.
  if (isLit === true) {
    // Add a white point light from the top-right.
    pointLight(255, 255, 255, 30, -20, 40);
```

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

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

  describe('A black sphere drawn on a gray background. An area
on the surface of the sphere is highlighted in blue.');
}

function draw() {
  background(200);

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

  // Add a specular highlight.
  // Use a p5.Color object.
  let c = color('dodgerblue');
  specularColor(c);

  // Add a white point light from the top-right.
  pointLight(255, 255, 255, 30, -20, 40);

  // Style the sphere.
  noStroke();

  // Add a white specular material.
  specularMaterial(255, 255, 255);
```

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

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

  describe('A black sphere drawn on a gray background. An area
on the surface of the sphere is highlighted in blue.');
}

function draw() {
  background(200);

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

  // Add a specular highlight.
  // Use a CSS color string.
  specularColor('#1E90FF');

  // Add a white point light from the top-right.
  pointLight(255, 255, 255, 30, -20, 40);

  // Style the sphere.
  noStroke();

  // Add a white specular material.
  specularMaterial(255, 255, 255);
```

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

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

  describe('A black sphere drawn on a gray background. An area
on the surface of the sphere is highlighted in blue.');
}

function draw() {
  background(200);

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

  // Add a specular highlight.
  // Use RGB values.
  specularColor(30, 144, 255);

  // Add a white point light from the top-right.
  pointLight(255, 255, 255, 30, -20, 40);

  // Style the sphere.
  noStroke();

  // Add a white specular material.
  specularMaterial(255, 255, 255);
```

Syntax

```
specularColor(v1, v2, v3)
```

```
specularColor(gray)
```

```
specularColor(value)
```

```
specularColor(values)
```

```
specularColor(color)
```

Parameters

v1	Number: red or hue value in the current colorMode().
v2	Number: green or saturation value in the current colorMode().
v3	Number: blue, brightness, or lightness value in the current colorMode().
gray	Number: grayscale value between 0 and 255.
value	String: color as a CSS string.
values	Number[]: color as an array of RGBA, HSBA, or HSLA values.
color	p5.Color: color as a p5.Color object.

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

Related References

ambientLight
Creates a light that shines from all directions.

directionalLight
Creates a light that shines in one direction.

imageLight
Creates an ambient light from an image.

lightFalloff
Sets the falloff rate for pointLight() and spotLight().