specularMaterial()

Sets the specular color of shapes' surface material.

The `specularMaterial()` color sets the components of light color that glossy coats on shapes will reflect. For example, calling `specularMaterial(255, 255, 0)` would cause a shape to reflect red and green light, but not blue light.

Unlike ambientMaterial(), `specularMaterial()` will reflect the full color of light sources including directionalLight(), pointLight(), and spotLight(). This is what gives it shapes their "shiny" appearance. The material's shininess can be controlled by the shininess() function.

`specularMaterial()` can be called three ways with different parameters to set the material's color.

The first way to call `specularMaterial()` has one parameter, `gray`. Grayscale values between 0 and 255, as in `specularMaterial(50)`, can be passed to set the material's color. Higher grayscale values make shapes appear brighter.

The second way to call `specularMaterial()` has one parameter, `color`. A p5.Color object, an array of color values, or a CSS color string, as in `specularMaterial('magenta')`, can be passed to set the material's color.

The third way to call `specularMaterial()` has four parameters, `v1`, `v2`, `v3`, and `alpha`. `alpha` is optional. RGBA, HSBA, or HSLA values can be passed to set the material's colors, as in `specularMaterial(255, 0, 0)` or `specularMaterial(255, 0, 0, 30)`. Color values will be interpreted using the current colorMode().

Examples

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to apply a specular material.

let isGlossy = false;

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

  describe('A red torus drawn on a gray background. It becomes
glossy when the user double-clicks.');
}

function draw() {
  background(200);

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

  // Turn on a white point light at the top-right.
  pointLight(255, 255, 255, 30, -40, 30);

  // Add a glossy coat if the user has double-clicked.
  if (isGlossy === true) {
    specularMaterial(255);
    shininess(50);
  }
```

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to apply a specular material.

let isGlossy = false;

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

  describe(
    'A red torus drawn on a gray background. It becomes glossy
and reflects green light when the user double-clicks.'
  );
}

function draw() {
  background(200);

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

  // Turn on a white point light at the top-right.
  pointLight(255, 255, 255, 30, -40, 30);

  // Add a glossy green coat if the user has double-clicked.
  if (isGlossy === true) {
    specularMaterial(0, 255, 0);
```

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to apply a specular material.

let isGlossy = false;

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

  describe(
    'A red torus drawn on a gray background. It becomes glossy
and reflects green light when the user double-clicks.'
  );
}

function draw() {
  background(200);

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

  // Turn on a white point light at the top-right.
  pointLight(255, 255, 255, 30, -40, 30);

  // Add a glossy green coat if the user has double-clicked.
  if (isGlossy === true) {
    // Create a p5.Color object.
```

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to apply a specular material.

let isGlossy = false;

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

  describe(
    'A red torus drawn on a gray background. It becomes glossy
and reflects green light when the user double-clicks.'
  );
}

function draw() {
  background(200);

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

  // Turn on a white point light at the top-right.
  pointLight(255, 255, 255, 30, -40, 30);

  // Add a glossy green coat if the user has double-clicked.
  if (isGlossy === true) {
    specularMaterial('#00FF00');
```

Syntax

```
specularMaterial(gray, [alpha])
```

```
specularMaterial(v1, v2, v3, [alpha])
```

```
specularMaterial(color)
```

Parameters

gray	Number: grayscale value between 0 (black) and 255 (white).
alpha	Number: alpha value in the current current colorMode().
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().
color	p5.Color|Number|[]String: color as a p5.Color object, an array of color values, or a CSS string.

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

Related References

copyToContext
Copies the shader from one drawing context to another.

inspectHooks
Logs the hooks available in this shader, and their current implementation.

modify
Returns a new shader, based on the original, but with custom snippets of shader code replacing default behaviour.

setUniform
Sets the shader's uniform (global) variables.