createFilterShader()

Creates a p5.Shader object to be used with the filter() function.

`createFilterShader()` works like createShader() but has a default vertex shader included. `createFilterShader()` is intended to be used along with filter() for filtering the contents of a canvas. A filter shader will be applied to the whole canvas instead of just p5.Geometry objects.

The parameter, `fragSrc`, sets the fragment shader. It's a string that contains the fragment shader program written in GLSL.

The p5.Shader object that's created has some uniforms that can be set:

- `sampler2D tex0`, which contains the canvas contents as a texture.
- `vec2 canvasSize`, which is the width and height of the canvas, not including pixel density.
- `vec2 texelSize`, which is the size of a physical pixel including pixel density. This is calculated as `1.0 / (width * density)` for the pixel width and `1.0 / (height * density)` for the pixel height.

The p5.Shader that's created also provides `varying vec2 vTexCoord`, a coordinate with values between 0 and 1. `vTexCoord` describes where on the canvas the pixel will be drawn.

For more info about filters and shaders, see Adam Ferriss' repo of shader examples or the Introduction to Shaders tutorial.

Examples

```
function setup() {
  let fragSrc = `precision highp float;
  void main() {
    gl_FragColor = vec4(1.0, 1.0, 0.0, 1.0);
  }`;

  createCanvas(100, 100, WEBGL);
  let s = createFilterShader(fragSrc);
  filter(s);
  describe('a yellow canvas');
}
```

```
let img, s;
function preload() {
  img = loadImage('/assets/bricks.jpg');
}
function setup() {
  let fragSrc = `precision highp float;

  // x,y coordinates, given from the vertex shader
  varying vec2 vTexCoord;

  // the canvas contents, given from filter()
  uniform sampler2D tex0;
  // other useful information from the canvas
  uniform vec2 texelSize;
  uniform vec2 canvasSize;
  // a custom variable from this sketch
  uniform float darkness;

  void main() {
    // get the color at current pixel
    vec4 color = texture2D(tex0, vTexCoord);
    // set the output color
    color.b = 1.0;
    color *= darkness;
    gl_FragColor = vec4(color.rgb, 1.0);
```

Syntax

```
createFilterShader(fragSrc)
```

Parameters

fragSrc	String: source code for the fragment shader.

Returns

p5.Shader: new shader object created from the fragment shader.

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.