GPU Shader Utilities (gpu.shader)

This module provides access to GPUShader internal functions.

Built-in shaders

All built-in shaders have the `mat4 ModelViewProjectionMatrix` uniform.

Its value must be modified using the `gpu.matrix` module.

`FLAT_COLOR`

Attributes:
vec3 pos, vec4 color

Uniforms:
none

`IMAGE`

Attributes:
vec3 pos, vec2 texCoord

Uniforms:
sampler2D image

`IMAGE_COLOR`

Attributes:
vec3 pos, vec2 texCoord

Uniforms:
sampler2D image, vec4 color

`SMOOTH_COLOR`

Attributes:
vec3 pos, vec4 color

Uniforms:
none

`UNIFORM_COLOR`

Attributes:
vec3 pos

Uniforms:
vec4 color

`POLYLINE_FLAT_COLOR`

Attributes:
vec3 pos, vec4 color

Uniforms:
vec2 viewportSize, float lineWidth

`POLYLINE_SMOOTH_COLOR`

Attributes:
vec3 pos, vec4 color

Uniforms:
vec2 viewportSize, float lineWidth

POLYLINE_UNIFORM_COLOR

Attributes:

vec3 pos

Uniforms:

vec2 viewportSize, float lineWidth

gpu.shader.**create_from_info(shader_info)**

Create shader from a GPUShaderCreateInfo.

PARAMETERS:

shader_info (`bpy.types.GPUShaderCreateInfo`) – GPUShaderCreateInfo

RETURNS:

Shader object corresponding to the given name.

RETURN TYPE:

`gpu.types.GPUShader`

gpu.shader.**from_builtin(shader_name, config='DEFAULT')**

Shaders that are embedded in the blender internal code (see Built-in shaders). They all read the uniform `mat4 ModelViewProjectionMatrix`, which can be edited by the `gpu.matrix` module.

You can also choose a shader configuration that uses clip_planes by setting the `CLIPPED` value to the config parameter. Note that in this case yc also need to manually set the value of `mat4 ModelMatrix`.

PARAMETERS:

- **shader_name** (*str*) – One of the builtin shader names.
- **config** (*str*) –

  One of these types of shader configuration:

  - `DEFAULT`
  - `CLIPPED`

RETURNS:

Shader object corresponding to the given name.

RETURN TYPE:

`gpu.types.GPUShader`

gpu.shader.**unbind()**

Unbind the bound shader object.