Skip to content

# GPU State Utilities (gpu.state)

This module provides access to the gpu state.

gpu.state.**active_framebuffer_get(enable)**

Return the active frame-buffer in context.

gpu.state.**blend_get()**

Current blending equation.

gpu.state.**blend_set(mode)**

Defines the fixed pipeline blending equation.

**PARAMETERS:**

**mode** (*str*) –

The type of blend mode.

- `NONE` No blending.
- `ALPHA` The original color channels are interpolated according to the alpha value.
- `ALPHA_PREMULT` The original color channels are interpolated according to the alpha value with the new colors pre-multiplied by this valu
- `ADDITIVE` The original color channels are added by the corresponding ones.
- `ADDITIVE_PREMULT` The original color channels are added by the corresponding ones that are pre-multiplied by the alpha value.
- `MULTIPLY` The original color channels are multiplied by the corresponding ones.
- `SUBTRACT` The original color channels are subtracted by the corresponding ones.
- `INVERT` The original color channels are replaced by its complementary color.

gpu.state.**clip_distances_set(distances_enabled)**

Sets the number of *gl_ClipDistance* planes used for clip geometry.

**PARAMETERS:**

**distances_enabled** (*int*) – Number of clip distances enabled.

gpu.state.**color_mask_set(r, g, b, a)**

Enable or disable writing of frame buffer color components.

**PARAMETERS:**

**a** (*r, g, b,*) – components red, green, blue, and alpha.

gpu.state.**depth_mask_get()**

Writing status in the depth component.

gpu.state.**depth_mask_set(value)**

Write to depth component.

**PARAMETERS:**

**value** – True for writing to the depth component.

gpu.state.**depth_test_get()**

Current depth_test equation.

gpu.state.**depth_test_set(mode)**

Defines the depth_test equation.

**PARAMETERS:**

**mode** (*str*) – The depth test equation name. Possible values are *NONE*, *ALWAYS*, *LESS*, *LESS_EQUAL*, *EQUAL*, *GREATER* and *GREATER_EQUAL*.

gpu.state.**face_culling_set(culling)**

Specify whether none, front-facing or back-facing facets can be culled.

**PARAMETERS:**

**mode** (*str*) – *NONE*, *FRONT* or *BACK*.

gpu.state.**front_facing_set(invert)**

Specifies the orientation of front-facing polygons.

**PARAMETERS:**

**invert** – True for clockwise polygons as front-facing.

gpu.state.**line_width_get()**

Current width of rasterized lines.

gpu.state.**line_width_set(width)**

Specify the width of rasterized lines.

**PARAMETERS:**

**size** – New width.

gpu.state.**point_size_set(size)**

Specify the diameter of rasterized points.

**PARAMETERS:**

**size** – New diameter.

gpu.state.**program_point_size_set(enable)**

If enabled, the derived point size is taken from the (potentially clipped) shader builtin gl_PointSize.

**PARAMETERS:**

**enable** (*bool*) – True for shader builtin gl_PointSize.

gpu.state.**scissor_get()**

Retrieve the scissors of the active framebuffer. Note: Only valid between 'scissor_set' and a framebuffer rebind.

**RETURNS:**

The scissor of the active framebuffer as a tuple (x, y, xsize, ysize). x, y: lower left corner of the scissor rectangle, in pixels. xsize, ysize: width a height of the scissor rectangle.

**RETURN TYPE:**

tuple[int, int, int, int]

gpu.state.**scissor_set(x, y, xsize, ysize)**

Specifies the scissor area of the active framebuffer. Note: The scissor state is not saved upon framebuffer rebind.

**PARAMETERS:**

- **y** (*x,*) – lower left corner of the scissor rectangle, in pixels.
- **ysize** (*xsize,*) – width and height of the scissor rectangle.

gpu.state.**scissor_test_set(enable)**

Enable/disable scissor testing on the active framebuffer.

**PARAMETERS:**

**enable** (*bool*) – True - enable scissor testing. False - disable scissor testing.

gpu.state.**viewport_get()**

Viewport of the active framebuffer.

gpu.state.**viewport_set(x, y, xsize, ysize)**

Specifies the viewport of the active framebuffer. Note: The viewport state is not saved upon framebuffer rebind.

**PARAMETERS:**

- **y** (*x,*) – lower left corner of the viewport_set rectangle, in pixels.
- **ysize** (*xsize,*) – width and height of the viewport_set.