



Struct HolovizOp::InputSpec

Table of contents

Nested Relationships

Struct Documentation

- Defined in [File holoviz.hpp](#)

Nested Relationships

This struct is a nested type of [Class HolovizOp](#).

Nested Types

- [Struct InputSpec::View](#)

Struct Documentation

struct InputSpec

Input specification

Public Functions

InputSpec() = default

inline InputSpec(const std::string &tensor_name, [InputType](#) type)

InputSpec(const std::string &tensor_name, const std::string &type_str)

explicit InputSpec(const std::string &yaml_description)

Returns

an [InputSpec](#) from the YAML form output by [description\(\)](#).

inline explicit operator bool() const noexcept

Returns

true if the input spec is valid

std::string description() const

Returns

a YAML string representation of the [InputSpec](#)

Public Members

`std::string tensor_name_`

name of the tensor containing the input data

`InputType type_ = InputType::UNKNOWN`

input type

`float opacity_ = 1.f`

layer opacity, 1.0 is fully opaque, 0.0 is fully transparent

`int32_t priority_ = 0`

layer priority, determines the render order, layers with higher priority values are rendered on top of layers with lower priority values

`std::vector<float> color_ = {1.f, 1.f, 1.f, 1.f}`

color of rendered geometry

`float line_width_ = 1.f`

line width for geometry made of lines

`float point_size_ = 1.f`

point size for geometry made of points

`std::vector<std::string> text_`

array of text strings, used when `type_` is TEXT.

`DepthMapRenderMode depth_map_render_mode_ = DepthMapRenderMode::POINTS`

depth map render mode, used if `type_` is DEPTH_MAP or DEPTH_MAP_COLOR.

`std::vector<View> views_`

struct View

Layer view.

By default a layer will fill the whole window. When using a view the layer can be placed freely within the window.

Layers can also be placed in 3D space by specifying a 3D transformation matrix. Note that for geometry layers there is a default matrix which allows coordinates in the range of [0 ... 1] instead of the Vulkan [-1 ... 1] range. When specifying a matrix for a geometry layer, this default matrix is overwritten.

When multiple views are specified the layer is drawn multiple times using the specified layer views.

It's possible to specify a negative term for height, which flips the image. When using a negative height, one should also adjust the y value to point to the lower left corner of the viewport instead of the upper left corner.

Public Members

float offset_x_ = 0.f

float offset_y_ = 0.f

offset of top-left corner of the view. Top left coordinate of the window area is (0, 0) bottom right coordinate is (1, 1).

float width_ = 1.f

float height_ = 1.f

width and height of the view in normalized range. 1.0 is full size.

std::optional<std::array<float, 16>> matrix_

row major 4x4 transform matrix (optional, can be nullptr)