vgl Documentation

Release 0.1

vgl

February 25, 2014

Contents

The documentation for the site is organized into a few different sections below:

- *User Documentation*
- *Developer Documentation*
- *Designer Documentation*
- *About VGL*

CHAPTER 1

User Documentation

1.1 Getting Started

VGL is a lightweight rendering library for scientific visualization and applications. It provides a object level API for WebGL. The high level API provided by the VGL is essential for building sophisticated visualization applications over the web.

For information on how to build VGL on your system please refer to *Build*. VGL is developed as part of ClimatePipes project which is funded by Department of Energy (DOE).

1.2 Build

You will need cmake to build VGL. Follow documentation on _cmake on how to get it installed on your system. Once installed, follow these steps:

```
$ git clone git//github.com:OpenGeoscience/vgl.git
$ mkdir build
$ cd build
$ ccmake ../vgl
```

If you have installed uglifyjs, then set UglifyJS_EXECUTABLE to the uglify executable, and then *configure* again. Finally, when you are satisfied with all the options, then you can run make to generate the library.:

```
$ make
```

1.3 Support

For all your general questions on how to use VGL, please send an email to opengeoscience-users@public.kitware.com. If you want to discuss on VGL API and other code level issues, then send an email to opengeoscience-developers@public.kitware.com.

1.4 FAQ

CHAPTER 2

Developer Documentation

2.1 API

2.1.1 Object (data)

Object

Object.**size**(*obj*)

Arguments

- **obj** –

Convenient function to get size of an object

Returns

-

Return type number

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.2 _global_ (data)

global

global.**gl**

global.**TraversalMode**
Type of traverse modes

global.**vertexAttributeKeys**
Vertex attribute keys

global.**vgl**
ogs.vgl namespace

global.**VisitorType**
Types of visitor type

modelViewMatrixStack()
Helper function to create stack for matrices

pushMatrix(*mat*)

Arguments

- **mat** –

Push new matrix to the stack

popMatrix()

Pop matrix from the stack

Returns

Return type

-

vglAttributeData()

Attribute data for the source

inherit(*C*, *P*)

Arguments

- **C** –
- **P** –

Convenient function to define JS inheritance

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.3 m_globalModifiedTime (class)

class m_globalModifiedTime()

Create a new instance of class timestamp

Returns

rtype vgl.timestamp

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.4 vgl.GL (class)

class vgl.**GL**()

Wrap GL enums. Currently to get values of the enums we need to create or access the context.

Using enums from here: https://github.com/toji/dart-gl-enums/blob/master/lib/gl_enums.dart

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.5 vgl.actor (class)

class vgl.**actor**()

Create a new instance of class actor

Returns

rtype vgl.actor

`vgl.actor.`**`matrix`**`()`

Get transformation matrix used by the actor

Returns

Return type mat4

`vgl.actor.`**`setMatrix`**`(`*4X4*`)`

Arguments

- **4X4** (*mat4*) – transformation matrix

Set transformation matrix for the actor

`vgl.actor.`**`referenceFrame`**`()`

Get reference frame for the transformations

Returns Possible values are Absolute or Relative

Return type String

`vgl.actor.`**`setReferenceFrame`**`(`*referenceFrame*`)`

Arguments

- **referenceFrame** (*vgl.boundingObject.ReferenceFrame*) – Possible values are (Absolute | Relative)

Set reference frame for the transformations

`vgl.actor.`**`mapper`**`()`

Return mapper where actor gets it behavior and data

Returns

Return type vgl.mapper

`vgl.actor.`**`setMapper`**`()`

:param vgl.mapper :

Connect an actor to its data source

`vgl.actor.`**`accept`**`(`*visitor*`)`

Arguments

- **visitor** –

`vgl.actor.`**`ascend`**`(`*visitor*`)`

Arguments

- **visitor** –

`vgl.actor.`**`computeLocalToWorldMatrix`**`(`*matrix*, *visitor*`)`

Arguments

- **matrix** –
- **visitor** –

Compute object space to world space matrix

`vgl.actor.`**`computeWorldToLocalMatrix`**`(`*matrix*, *visitor*`)`

Arguments

- **matrix** –

- **visitor** –

Compute world space to object space matrix

`vgl.actor.`**`computeBounds`**`()`
Compute actor bounds

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.6 vgl.blendFunction (class)

class `vgl.`**`blendFunction`**`(`*source*, *destination*`)`
Create a new instance of clas blendFunction

Arguments

- **source** –
- **destination** –

Returns

rtype vgl.blendFunction

`vgl.blendFunction.`**`apply`**`()`
:param vgl.renderState :

Apply blend function to the current state

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.7 vgl.boundingObject (class)

class `vgl.`**`boundingObject`**`()`
Create a new instance of class boundingObject

Returns

rtype vgl.boundingObject

`vgl.boundingObject.`**`bounds`**`()`
Get current bounds of the object

`vgl.boundingObject.`**`setBounds`**`(`*minX*, *maxX*, *minY*, *maxY*, *minZ*, *maxZ*`)`

Arguments

- **minX** –
- **maxX** –
- **minY** –
- **maxY** –
- **minZ** –
- **maxZ** –

Set current bounds of the object

`vgl.boundingObject.`**`resetBounds`**`()`
Reset bounds to default values

`vgl.boundingObject.`**`computeBounds`**`()`
Compute bounds of the object

Should be implemented by the concrete class

`vgl.boundingObject.`**`computeBoundsTimestamp`**`()`
Return bounds computation modification time

Returns

Return type vgl.timestamp

`vgl.boundingObject.`**`boundsDirtyTimestamp`**`()`
Return bounds dirty timestamp

Returns

Return type vgl.timestamp

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.8 vgl.camera (class)

class `vgl.`**`camera`**`()`
Create a new instance of class camera

Returns

rtype vgl.camera

`vgl.camera.`**`viewAngle`**`()`
Get view angle of the camera

`vgl.camera.`**`setViewAngleDegrees`**`(`*a*`)`

Arguments

- **a** –

Set view angle of the camera in degrees, which is converted to radians.

`vgl.camera.`**`setViewAngle`**`(`*a*`)`

Arguments

- **a** –

Set view angle of the camera in degrees, which is converted to radians.

`vgl.camera.`**`position`**`()`
Get position of the camera

`vgl.camera.`**`setPosition`**`(`*x*, *y*, *z*`)`

Arguments

- **x** –
- **y** –
- **z** –

Set position of the camera

`vgl.camera.`**`focalPoint`**`()`
Get focal point of the camera

`vgl.camera.`**`setFocalPoint`**(*x*, *y*, *z*)

Arguments

- **x** –
- **y** –
- **z** –

Set focal point of the camera

`vgl.camera.`**`viewUpDirection`**()
Get view-up direction of camera

`vgl.camera.`**`setViewUpDirection`**(*x*, *y*, *z*)

Arguments

- **x** –
- **y** –
- **z** –

Set view-up direction of the camera

`vgl.camera.`**`centerOfRotation`**()
Get center of rotation for camera

`vgl.camera.`**`setCenterOfRotation`**(*centerOfRotation*)

Arguments

- **centerOfRotation** –

Set center of rotation for camera

`vgl.camera.`**`getClippingRange`**()
Get clipping range of the camera

`vgl.camera.`**`setClippingRange`**(*near*, *far*)

Arguments

- **near** –
- **far** –

Set clipping range of the camera

`vgl.camera.`**`viewAspect`**()
Get view aspect

`vgl.camera.`**`setViewAspect`**(*aspect*)

Arguments

- **aspect** –

Set view aspect

`vgl.camera.`**`isEnabledParallelProjection`**()
Return if parallel projection is enabled

`vgl.camera.`**`enableParallelProjection`**(*flag*)

Arguments

- **flag** –

Enable / disable parallel projection

`vgl.camera.`**`setParallelProjection`**`(`*left*, *right*, *top*, *bottom*`)`

Arguments

- **left** –
- **right** –
- **top** –
- **bottom** –

Set parallel projection parameters

`vgl.camera.`**`directionOfProjection`**`()`
Return direction of projection

`vgl.camera.`**`viewPlaneNormal`**`()`
Return view plane normal direction

`vgl.camera.`**`viewMatrix`**`()`
Return view-matrix for the camera This method does not compute the view-matrix for the camera. It is assumed that a call to computeViewMatrix has been made earlier.

Returns

Return type mat4

`vgl.camera.`**`projectionMatrix`**`()`
Return camera projection matrix This method does not compute the projection-matrix for the camera. It is assumed that a call to computeProjectionMatrix has been made earlier.

Returns

Return type mat4

`vgl.camera.`**`clearMask`**`()`
Return clear mask used by this camera

Returns

Return type number

`vgl.camera.`**`setClearMask`**`(`*mask*`)`

Arguments

- **mask** –

Set clear mask for camera

`vgl.camera.`**`clearColor`**`()`
Get clear color (background color) of the camera

Returns

Return type Array

`vgl.camera.`**`setClearColor`**`(`*color*, *g*, *b*, *a*`)`

Arguments

- **color** – RGBA
- **g** –
- **b** –

- **a** –

Set clear color (background color) for the camera

`vgl.camera.`**`clearDepth`**`()`

Returns

Return type {1.0: null}

`vgl.camera.`**`setClearDepth`**`(`*depth*`)`

Arguments

- **depth** –

`vgl.camera.`**`computeDirectionOfProjection`**`()`
Compute direction of projection

`vgl.camera.`**`computeViewPlaneNormal`**`()`
Compute view plane normal

`vgl.camera.`**`zoom`**`(`*d*`)`

Arguments

- **d** –

Move camera closer or further away from the scene

`vgl.camera.`**`pan`**`(`*dx*, *dy*, *dz*`)`

Arguments

- **dx** –
- **dy** –
- **dz** –

Move camera sideways

`vgl.camera.`**`computeOrthogonalAxes`**`()`
Compute camera coordinate axes

`vgl.camera.`**`rotate`**`(`*dx*, *dy*`)`

Arguments

- **dx** – Rotation around vertical axis in degrees
- **dy** – Rotation around horizontal axis in degrees

Rotate camera around center of rotation

`vgl.camera.`**`computeViewMatrix`**`()`
Compute camera view matrix

`vgl.camera.`**`computeProjectionMatrix`**`()`
Compute camera projection matrix

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.9 vgl.command (class)

command

class `vgl.command()`

Create a new instance of class command

Returns

rtype vgl.command

`command.keyPressEvent`

vgl.command

Event types

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.10 vgl.defaultValue (class)

class `vgl.defaultValue(`*a*, *b*`)`

Returns the first parameter if not undefined, otherwise the second parameter.

Arguments

- **a** –
- **b** –

Returns

rtype vgl.defaultValue

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.11 vgl.geojsonReader (class)

class `vgl.geojsonReader()`

Create a new instance of geojson reader

This contains code that reads a geoJSON file and produces rendering primitives from it.

Returns

rtype vgl.geojsonReader

`vgl.geojsonReader.readScalars(`*coordinates*, *geom*, *size_estimate*, *idx*`)`

Arguments

- **coordinates** –
- **geom** –
- **size_estimate** –
- **idx** –

Read scalars

`vgl.geojsonReader.readPoint(`*coordinates*`)`

Arguments

- **coordinates** –

Read point data

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readMultiPoint`**`(`*coordinates*`)`

Arguments

- **coordinates** –

Read multipoint data

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readLineString`**`(`*coordinates*`)`

Arguments

- **coordinates** –

Read line string data

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readMultiLineString`**`(`*coordinates*`)`

Arguments

- **coordinates** –

Read multi line string

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readPolygon`**`(`*coordinates*`)`

Arguments

- **coordinates** –

Read polygon data

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readMultiPolygon`**`(`*coordinates*`)`

Arguments

- **coordinates** –

Read multi polygon data

Returns

Return type vgl.geometryData

`vgl.geojsonReader.`**`readGJObjectInt`**`(`*object*`)`

Arguments

- **object** –

Returns

Return type

-

`vgl.geojsonReader.`**`readGJObject`**`(`*object*`)`

Arguments

- **object** –

Returns

Return type

-

`vgl.geojsonReader.`**`linearizeGeoms`**`(`*geoms*, *geom*`)`

Arguments

- **geoms** –
- **geom** –

Linearize geometries

`vgl.geojsonReader.`**`readGeomObject`**`(`*object*`)`

Arguments

- **object** –

Read geometries from geojson object

Returns

Return type Array

`vgl.geojsonReader.`**`getPrimitives`**`(`*buffer*`)`

Arguments

- **buffer** –

Given a buffer get rendering primitives

Returns

Return type

-

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.12 vgl.geometryData (class)

class `vgl.`**`geometryData`**`()`

Create a new instance of class geometryData

Returns

rtype vgl.geometryData

`vgl.geometryData.`**`type`**`()`

Return type

`vgl.geometryData.`**`name`**`()`

Return ID of the geometry data

`vgl.geometryData.`**`setName`**`(`*name*`)`

Arguments

- **name** –

Set name of the geometry data

`vgl.geometryData.`**`addSource`**`(`*source*`)`

Arguments

- **source** –

Add new source

`vgl.geometryData.`**`source`**`(`*index*`)`

Arguments

- **index** –

Return source for a given index. Returns 0 if not found.

`vgl.geometryData.`**`numberOfSources`**`()`
Return number of sources

`vgl.geometryData.`**`sourceData`**`(`*key*`)`

Arguments

- **key** –

Return source data given a key

`vgl.geometryData.`**`addPrimitive`**`(`*primitive*`)`

Arguments

- **primitive** –

Add new primitive

`vgl.geometryData.`**`primitive`**`(`*index*`)`

Arguments

- **index** –

Return primitive for a given index. Returns null if not found.

`vgl.geometryData.`**`numberOfPrimitives`**`()`
Return number of primitives

`vgl.geometryData.`**`bounds`**`()`
Return bounds [minX, maxX, minY, maxY, minZ, maxZ]

`vgl.geometryData.`**`resetBounds`**`()`
Reset bounds

`vgl.geometryData.`**`setBounds`**`(`*minX*, *maxX*, *minY*, *maxY*, *minZ*, *maxZ*`)`

Arguments

- **minX** –
- **maxX** –
- **minY** –
- **maxY** –

- **minZ** –
- **maxZ** –

Set bounds

`vgl.geometryData.`**`computeBounds`**`()`
Compute bounds

`vgl.geometryData.`**`findClosestVertex`**`(`*point*`)`

Arguments

- **point** –

Returns the vertex closest to a given position

`vgl.geometryData.`**`getPosition`**`(`*index*`)`

Arguments

- **index** –

Returns the requested vertex position

`vgl.geometryData.`**`getScalar`**`(`*index*`)`

Arguments

- **index** –

Returns the scalar corresponding to a given vertex index

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.13 vgl.groupNode (class)

class `vgl.`**`groupNode`**`()`
Create a new instance of class groupNode

Returns

rtype vgl.groupNode

`vgl.groupNode.`**`setVisible`**`(`*flag*`)`

Arguments

- **flag** –

Turn on / off visibility

Returns

Return type boolean

`vgl.groupNode.`**`addChild`**`(`*childNode*`)`

Arguments

- **childNode** –

Make the incoming node as child of the group node

Returns

Return type boolean

`vgl.groupNode.`**`removeChild`**`(`*childNode*`)`

Arguments

- **childNode –**

Remove parent-child relationship between the group and incoming node

Returns

Return type boolean

`vgl.groupNode.`**`removeChildren`**`()`

Remove parent-child relationship between child nodes and the group node

`vgl.groupNode.`**`children`**`()`

Return children of this group node

Returns

Return type Array

`vgl.groupNode.`**`accept`**`(`*visitor*`)`

Arguments

- **visitor –**

Accept a visitor and traverse the scene tree

`vgl.groupNode.`**`traverse`**`(`*visitor*`)`

Arguments

- **visitor –**

Traverse the scene

`vgl.groupNode.`**`traverseChildrenAndUpdateBounds`**`(`*visitor*`)`

Arguments

- **visitor –**

Traverse all of the children and update the bounds for each

`vgl.groupNode.`**`traverseChildren`**`(`*visitor*`)`

Arguments

- **visitor –**

Traverse children of the group node

`vgl.groupNode.`**`computeBounds`**`()`

Compute bounds for the group node

`vgl.groupNode.`**`updateBounds`**`(`*child*`)`

Arguments

- **child –**

Update bounds for the group node

This method is used internally to update bounds of the group node by traversing each of its child.

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.14 vgl.interactorStyle (class)

vgl.interactorStyle interactorStyle is a base class for all interactor styles

class `vgl.`**`interactorStyle`**`()`

Create a new instance of class interactorStyle

Returns

rtype vgl.interactorStyle

`vgl.interactorStyle.`**`viewer`**`()`

Return viewer referenced by the interactor style

Returns

Return type null

`vgl.interactorStyle.`**`setViewer`**`(`*viewer*`)`

Arguments

- **viewer** –

Set viewer for the interactor style

`vgl.interactorStyle.`**`handleMouseDown`**`(`*event*`)`

Arguments

- **event** –

Handle mouse down event

Returns

Return type boolean

`vgl.interactorStyle.`**`handleMouseUp`**`(`*event*`)`

Arguments

- **event** –

Handle mouse up event

Returns

Return type boolean

`vgl.interactorStyle.`**`handleMouseMove`**`(`*event*`)`

Arguments

- **event** –

Handle mouse move event

Returns

Return type boolean

`vgl.interactorStyle.`**`handleKeyPress`**`(`*event*`)`

Arguments

- **event** –

Handle key press event

Returns

Return type boolean

`vgl.interactorStyle.`**`handleContextMenu`**`(`*event*`)`

Arguments

- **event** –

Handle context menu event

Returns

Return type boolean

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.15 vgl.legend (class)

vgl.legend

class `vgl.`**`legend`**`()`

Create a new instance of class legend legend class is intended to create legend for 2D/3D scene.

Returns

rtype vgl.legend

`vgl.legend.`**`lookupTable`**`()`

Returns

Return type

-

`vgl.legend.`**`setLookupTable`**`(`*lookupTable*`)`

Arguments

- **lookupTable** –

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.16 vgl.lineSource (class)

class `vgl.`**`lineSource`**`(`*positions*, *colors*`)`

Create a new instance of class lineSource

Arguments

- **positions** –
- **colors** –

Returns

rtype vgl.lineSource

`vgl.lineSource.`**`setPositions`**`(`*positions*`)`

Arguments

- **positions** –

Set start positions for the lines

`vgl.lineSource.`**`setColors`**`(`*colors*`)`

Arguments

- **colors** –

Set colors for the lines

`vgl.lineSource.`**`create`**`()`

Create a point geometry given input parameters

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.17 vgl.lookupTable (class)

class `vgl.`**`lookupTable`**`()`

Create a new instance of class lookupTable

Returns

rtype vgl.lookupTable

`vgl.lookupTable.`**`setup`**`(`*renderState*`)`

Arguments

- **renderState** (*vgl.renderState*) –

Create lookup table, initialize parameters, and bind data to it

`vgl.lookupTable.`**`colorTable`**`()`

Get color table used by the lookup table

Returns

Return type

-

`vgl.lookupTable.`**`setColorTable`**`(`*colors*`)`

Arguments

- **colors** –

Set color table used by the lookup table

Returns

Return type boolean

`vgl.lookupTable.`**`range`**`()`

Get scalar range

Returns

Return type Array

`vgl.lookupTable.`**`setRange`**`(`*range*`)`

Arguments

- **range** –

Set scalar range for the lookup table

Returns

Return type boolean

`vgl.lookupTable.`**`updateRange`**`(`*range*`)`

Arguments

- **range** –

Given a [min,max] range update the lookup table range

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.18 vgl.mapper (class)

class `vgl.`**`mapper`**`()`

Create a new instance of class mapper

Returns

rtype vgl.mapper

`vgl.mapper.`**`computeBounds`**`()`

Compute bounds of the data

`vgl.mapper.`**`color`**`()`

Get solid color of the geometry

`vgl.mapper.`**`setColor`**`(`*r*, *g*, *b*`)`

Arguments

- **r** – Red component of the color [0.0 - 1.0]
- **g** – Green component of the color [0.0 - 1.0]
- **b** – Blue component of the color [0.0 - 1.0]

Set solid color of the geometry. Default is teal [1.0, 1.0, 1.0]

`vgl.mapper.`**`geometryData`**`()`

Return stored geometry data if any

`vgl.mapper.`**`setGeometryData`**`(`*geom*`)`

Arguments

- **geom** –

Connect mapper to its geometry data

`vgl.mapper.`**`render`**`(`*renderState*`)`

Arguments

- **renderState** –

Render the mapper

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.19 vgl.material (class)

class `vgl.`**`material`**`()`

Create a new instance of class material

Returns

rtype vgl.material

`vgl.material.`**`binNumber`**`()`

Return bin number for the material

Returns

Return type number

`vgl.material.`**`setBinNumber`**`(`*binNo*`)`

Arguments

- **binNo** –

Set bin number for the material

`vgl.material.`**`exists`**`(`*attr*`)`

Arguments

- **attr** –

Check if incoming attribute already exists in the material

Returns

Return type boolean

`vgl.material.`**`setAttribute`**`(`*attr*`)`

Arguments

- **attr** –

Set a new attribute for the material

This method replace any existing attribute except for textures as materials can have multiple textures.

Returns

Return type boolean

`vgl.material.`**`addAttribute`**`(`*attr*`)`

Arguments

- **attr** –

Add a new attribute to the material.

Returns

Return type boolean

`vgl.material.`**`shaderProgram`**`()`

Return shader program used by the material

Returns

Return type vgl.shaderProgram

`vgl.material.`**`render`**`(`*renderState*`)`

Arguments

- **renderState** –

Activate the material

`vgl.material.`**`remove`**`(`*renderState*`)`

Arguments

- **renderState** –

Deactivate the material

`vgl.material.`**`bind`**`(`*renderState*`)`

Arguments

- **renderState** –

Bind and activate material states

`vgl.material.`**`undoBind`**`(`*renderState*`)`

Arguments

- **renderState** –

Undo-bind and de-activate material states

`vgl.material.`**`bindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState** –
- **key** –

Bind vertex data

`vgl.material.`**`undoBindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState** –
- **key** –

Undo bind vertex data

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.20 vgl.materialAttribute (class)

class `vgl.`**`materialAttribute`**`(`*type*`)`

Create a new instance of class materialAttribute

Arguments

- **type** –

Returns

rtype vgl.materialAttribute

`vgl.materialAttribute.`**`type`**`()`

Return tyep of the material attribute

Returns

Return type

-

`vgl.materialAttribute.`**`enabled`**`()`

Return if material attribute is enabled or not

Returns

Return type boolean

`vgl.materialAttribute.`**`setup`**`(`*renderState*`)`

Arguments

- **renderState –**

Setup (initialize) the material attribute

Returns

Return type boolean

`vgl.materialAttribute.`**`bind`**`(`*renderState*`)`

Arguments

- **renderState –**

Bind and activate the material attribute

Returns

Return type boolean

`vgl.materialAttribute.`**`undoBind`**`(`*renderState*`)`

Arguments

- **renderState –**

Undo bind and deactivate the material

Returns

Return type boolean

`vgl.materialAttribute.`**`setupVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState –**
- **key –**

Initialize vertex data for the material attribute

Returns

Return type boolean

`vgl.materialAttribute.`**`bindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState –**
- **key –**

Bind and activate vertex specific data

Returns

Return type boolean

`vgl.materialAttribute.`**`undoBindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState –**
- **key –**

Undo bind and deactivate vertex specific data

Returns

Return type boolean

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.21 vgl.node (class)

class `vgl.`**`node`**`()`

Create a new instance of class node

Returns

rtype vgl.node

`vgl.node.`**`accept`**`(`*visitor*`)`

Arguments

- **visitor –**

Accept visitor for scene traversal

`vgl.node.`**`material`**`()`

Return active material used by the node

`vgl.node.`**`setMaterial`**`(`*material*`)`

Arguments

- **material –**

Set material to be used the node

Returns

Return type boolean

`vgl.node.`**`visible`**`()`

Check if the node is visible or node

Returns

Return type boolean

`vgl.node.`**`setVisible`**`(`*flag*`)`

Arguments

- **flag –**

Turn ON/OFF visibility of the node

Returns

Return type boolean

`vgl.node.parent()`
Return current parent of the node

Returns

Return type null

`vgl.node.setParent(parent)`

Arguments

- **parent** –

Set parent of the node

Returns

Return type boolean

`vgl.node.overlay()`
Check if the node is an overlay node

Returns

Return type boolean

`vgl.node.setOverlay(flag)`

Arguments

- **flag** –

Set if the node is an overlay node or not

Returns

Return type boolean

`vgl.node.traverse(visitor)`

Arguments

- **visitor** –

Traverse children

`vgl.node.boundsModified()`
Mark that the bounds are modified

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.22 vgl.object (class)

class `vgl.object()`
Create a new instance of class object

Returns

rtype vgl.object

`vgl.object.modified()`
Mark the object modified

`vgl.object.getMTime()`
Return modified time of the object

Returns

Return type

-

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.23 vgl.picker (class)

vgl.picker

class `vgl.picker()`

Create a new instance of class picker

Returns

rtype vgl.picker

`vgl.picker.getActors()`

Get actors intersected

`vgl.picker.pick(selectionX, selectionY, renderer)`

Arguments

- **selectionX** –
- **selectionY** –
- **renderer** –

Perform pick operation

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.24 vgl.planeSource (class)

class `vgl.planeSource()`

Create a new instance of class planeSource

Returns

rtype vgl.planeSource

`vgl.planeSource.setOrigin(x, y, z)`

Arguments

- **x** –
- **y** –
- **z** –

Set origin of the plane

`vgl.planeSource.setPoint1(x, y, z)`

Arguments

- **x** –
- **y** –
- **z** –

Set point that defines the first axis of the plane

`vgl.planeSource.`**`setPoint2`**`(`*x*, *y*, *z*`)`

Arguments

- **x** –
- **y** –
- **z** –

Set point that defines the first axis of the plane

`vgl.planeSource.`**`create`**`()`
Create a plane geometry given input parameters

Returns

Return type null

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.25 vgl.pointSource (class)

class `vgl.`**`pointSource`**`()`
Create a new instance of class pointSource

Returns

rtype vgl.pointSource

`vgl.pointSource.`**`setPositions`**`(`*positions*`)`

Arguments

- **positions** –

Set positions for the source

`vgl.pointSource.`**`setColors`**`(`*colors*`)`

Arguments

- **colors** –

Set colors for the points

`vgl.pointSource.`**`setTextureCoordinates`**`(`*texcoords*`)`

Arguments

- **texcoords** –

Set texture coordinates for the points

`vgl.pointSource.`**`create`**`()`
Create a point geometry given input parameters

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.26 vgl.primitive (class)

class `vgl.`**`primitive`**`()`

Create a new instance of class primitive

Returns

rtype vgl.primitive

`vgl.primitive.`**`indices`**`()`

Get indices of the primitive

Returns

Return type null

`vgl.primitive.`**`createIndices`**`(`*type*`)`

Arguments

- **type** –

Create indices array for the primitive

`vgl.primitive.`**`numberOfIndices`**`()`

Return the number of indices

`vgl.primitive.`**`sizeInBytes`**`()`

Return size of indices in bytes

`vgl.primitive.`**`setPrimitiveType`**`(`*type*`)`

Arguments

- **type** –

Set primitive type

`vgl.primitive.`**`indicesPerPrimitive`**`()`

Return count of indices that form a primitives

`vgl.primitive.`**`setIndicesPerPrimitive`**`(`*count*`)`

Arguments

- **count** –

Set count of indices that form a primitive

`vgl.primitive.`**`indicesValueType`**`()`

Return indices value type

`vgl.primitive.`**`setIndicesValueType`**`(`*type*`)`

Arguments

- **type** –

Set indices value type

`vgl.primitive.`**`setIndices`**`(`*indicesArray*`)`

Arguments

- **indicesArray** –

Set indices from a array

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.27 vgl.pvwInteractorStyle (class)

vgl.pvwInteractorStyle

class `vgl.`**`pvwInteractorStyle`**`()`

Create a new instance of pvwInteractorStyle (for ParaViewWeb)

Returns

rtype vgl.pvwInteractorStyle

`vgl.pvwInteractorStyle.`**`handleMouseMove`**`(`*event*`)`

Arguments

- **event** –

Handle mouse move event

Returns

Return type boolean

`vgl.pvwInteractorStyle.`**`handleMouseDown`**`(`*event*`)`

Arguments

- **event** –

Handle mouse down event

Returns

Return type boolean

`vgl.pvwInteractorStyle.`**`handleMouseUp`**`(`*event*`)`

Arguments

- **event** –

Handle mouse up event

Returns

Return type boolean

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.28 vgl.renderWindow (class)

class `vgl.`**`renderWindow`**`(`*canvas*`)`

Create a new instance of class renderWindow

Arguments

- **canvas** –

Returns

rtype vgl.renderWindow

`vgl.renderWindow.`**`windowSize`**`()`

Get size of the render window

Returns

Return type Array

`vgl.renderWindow.`**`setWindowSize`**(*width*, *height*)

Arguments

- **width** –
- **height** –

Set size of the render window

Returns

Return type boolean

`vgl.renderWindow.`**`windowPosition`**()
Get window position (top left coordinates)

Returns

Return type Array

`vgl.renderWindow.`**`setWindowPosition`**(*x*, *y*)

Arguments

- **x** –
- **y** –

Set window position (top left coordinates)

Returns

Return type boolean

`vgl.renderWindow.`**`renderers`**()
Return all renderers contained in the render window

Returns

Return type Array

`vgl.renderWindow.`**`activeRenderer`**()
Get active renderer of the the render window

Returns vgl.renderer

`vgl.renderWindow.`**`addRenderer`**(*ren*)

Arguments

- **ren** –

Add renderer to the render window

Returns

Return type boolean

`vgl.renderWindow.`**`removeRenderer`**(*ren*)

Arguments

- **ren** –

Remove renderer from the render window

Returns

Return type boolean

`vgl.renderWindow.`**`getRenderer`**`(`*index*`)`

Arguments

- **index** –

Return a renderer at a given index

Returns

Return type vgl.renderer

`vgl.renderWindow.`**`hasRenderer`**`(`*ren*`)`

Arguments

- **ren** –

Check if the renderer exists

Returns

Return type boolean

`vgl.renderWindow.`**`resize`**`(`*width*, *height*`)`

Arguments

- **width** –
- **height** –

Resize window

`vgl.renderWindow.`**`positionAndResize`**`(`*x*, *y*, *width*, *height*`)`

Arguments

- **x** –
- **y** –
- **width** –
- **height** –

Resize and reposition the window

`vgl.renderWindow.`**`createWindow`**`()`
Create the window

Returns

Return type boolean

`vgl.renderWindow.`**`deleteWindow`**`()`
Delete this window and release any graphics resources

`vgl.renderWindow.`**`render`**`()`
Render the scene

`vgl.renderWindow.`**`focusDisplayPoint`**`()`
Get the focusDisplayPoint from the activeRenderer

Returns

Return type vec4

`vgl.renderWindow.`**`displayToWorld`**(*x*, *y*, *focusDisplayPoint*)

Arguments

- **x** (*Number*) –
- **y** (*Number*) –
- **focusDisplayPoint** (*vec4*) –

Transform a point in display space to world space

Returns

Return type vec4

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.29 vgl.shaderProgram (class)

class `vgl.`**`shaderProgram`**()

Create a new instace of class shaderProgram

Returns

rtype vgl.shaderProgram

`vgl.shaderProgram.`**`queryUniformLocation`**(*name*)

Arguments

- **name** –

Query uniform location in the program

Returns

Return type

-

`vgl.shaderProgram.`**`queryAttributeLocation`**(*name*)

Arguments

- **name** –

Query attribute location in the program

Returns

Return type

-

`vgl.shaderProgram.`**`addShader`**(*shader*)

Arguments

- **shader** –

Add a new shader to the program

Returns

Return type boolean

`vgl.shaderProgram.`**`addUniform`**(*uniform*)

Arguments

- **uniform** –

Add a new uniform to the program

Returns

Return type boolean

`vgl.shaderProgram.`**`addVertexAttribute`**`(`*attr*, *key*`)`

Arguments

- **attr** –
- **key** –

Add a new vertex attribute to the program

`vgl.shaderProgram.`**`uniformLocation`**`(`*name*`)`

Arguments

- **name** –

Get uniform location

This method does not perform any query into the program but relies on the fact that it depends on a call to queryUniformLocation earlier.

Returns

Return type number

`vgl.shaderProgram.`**`attributeLocation`**`(`*name*`)`

Arguments

- **name** –

Get attribute location

This method does not perform any query into the program but relies on the fact that it depends on a call to queryUniformLocation earlier.

Returns

Return type number

`vgl.shaderProgram.`**`uniform`**`(`*name*`)`

Arguments

- **name** –

Get uniform object using name as the key

Returns

Return type

-

`vgl.shaderProgram.`**`updateUniforms`**`()`

Update all uniforms

This method should be used directly unless required `vgl.shaderProgram.`**`link`**`()`

Link shader program

Returns

Return type boolean

`vgl.shaderProgram.`**`use`**`()`

Use the shader program

`vgl.shaderProgram.`**`cleanUp`**`()`

Peform any clean up required when the program gets deleted

`vgl.shaderProgram.`**`deleteProgram`**`()`

Delete the shader program

`vgl.shaderProgram.`**`deleteVertexAndFragment`**`()`

Delete vertex and fragment shaders

`vgl.shaderProgram.`**`bind`**`(`*renderState*`)`

Arguments

- **renderState** –

Bind the program with its shaders

Returns

Return type boolean

`vgl.shaderProgram.`**`undoBind`**`(`*renderState*`)`

Arguments

- **renderState** –

Undo binding of the shader program

`vgl.shaderProgram.`**`bindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState** –
- **key** –

Bind vertex data

`vgl.shaderProgram.`**`undoBindVertexData`**`(`*renderState*, *key*`)`

Arguments

- **renderState** –
- **key** –

Undo bind vetex data

`vgl.shaderProgram.`**`bindUniforms`**`()`

Bind uniforms

`vgl.shaderProgram.`**`bindAttributes`**`()`

Bind vertex attributes

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.30 vgl.shapefileReader (class)

class `vgl.`**`shapefileReader`**`()`

Create a new instance of shapefile reader

This contains code that reads a shapefile and produces vgl geometries

Returns

rtype vgl.shapefileReader

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.31 vgl.sourceData (class)

class `vgl.`**`sourceData`**`()`

Create a new instance of class sourceData

Returns

rtype vgl.sourceData

`vgl.sourceData.`**`data`**`()`

Return raw data for this source

Returns

Return type Array

`vgl.sourceData.`**`addAttribute`**`(`*key*, *dataType*, *sizeOfDataType*, *offset*, *stride*, *noOfComponents*, *normalized*`)`

Arguments

- **key** –
- **dataType** –
- **sizeOfDataType** –
- **offset** –
- **stride** –
- **noOfComponents** –
- **normalized** –

Add new attribute data to the source

`vgl.sourceData.`**`sizeOfArray`**`()`

Return size of the source data

`vgl.sourceData.`**`lengthOfArray`**`()`

Return length of array

`vgl.sourceData.`**`sizeInBytes`**`()`

Return size of the source data in bytes

`vgl.sourceData.`**`hasKey`**`(`*key*`)`

Arguments

- **key** –

Check if there is attribute exists of a given key type `vgl.sourceData.`**`keys`**`()`

Return keys of all attributes

`vgl.sourceData.`**`numberOfAttributes`**`()`

Return number of attributes of source data

`vgl.sourceData.`**`attributeNumberOfComponents`**`(`*key*`)`

Arguments

- **key** –

Return number of components of the attribute data

`vgl.sourceData.`**`normalized`**`(`*key*`)`

Arguments

- **key** –

Return if the attribute data is normalized

`vgl.sourceData.`**`sizeOfAttributeDataType`**`(`*key*`)`

Arguments

- **key** –

Return size of the attribute data type

`vgl.sourceData.`**`attributeDataType`**`(`*key*`)`

Arguments

- **key** –

Return attribute data type

`vgl.sourceData.`**`attributeOffset`**`(`*key*`)`

Arguments

- **key** –

Return attribute offset

`vgl.sourceData.`**`attributeStride`**`(`*key*`)`

Arguments

- **key** –

Return attribute stride

`vgl.sourceData.`**`pushBack`**`(`*vertexData*`)`

Arguments

- **vertexData** –

Virtual function to insert new vertex data at the end

`vgl.sourceData.`**`insert`**`(`*data*`)`

Arguments

- **data** –

Insert new data block to the raw data

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.32 vgl.sourceDataSf (class)

class `vgl.`**`sourceDataSf`**`()`

Create a new instance of class sourceDataSf meant to hold scalar float values

Returns

rtype vgl.sourceDataSf

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.33 vgl.texture (class)

class `vgl.`**`texture`**`()`

Create a new instance of class texture

Returns

rtype vgl.texture

`vgl.texture.`**`setup`**`(`*renderState*`)`

Arguments

- **renderState** –

Create texture, update parameters, and bind data

`vgl.texture.`**`bind`**`(`*renderState*`)`

Arguments

- **renderState** –

Create texture and if already created use it

`vgl.texture.`**`undoBind`**`(`*renderState*`)`

Arguments

- **renderState** –

Turn off the use of this texture

`vgl.texture.`**`image`**`()`

Get image used by the texture

Returns

Return type vgl.image

`vgl.texture.`**`setImage`**`(`*image*`)`

Arguments

- **image** (*vgl.image*) –

Set image for the texture

Returns

Return type boolean

`vgl.texture.`**`textureUnit`**`()`

Get texture unit of the texture

Returns

Return type number

`vgl.texture.`**`setTextureUnit`**`(`*unit*`)`

Arguments

- **unit** (*number*) –

Set texture unit of the texture. Default is 0.

Returns

Return type boolean

`vgl.texture.`**`width`**`()`
Get width of the texture

Returns

Return type

-

`vgl.texture.`**`setWidth`**`(`*width*`)`

Arguments

- **width** (*number*) –

Set width of the texture

Returns

Return type boolean

`vgl.texture.`**`depth`**`()`
Get depth of the texture

Returns

Return type number

`vgl.texture.`**`setDepth`**`(`*depth*`)`

Arguments

- **depth** (*number*) –

Set depth of the texture

Returns

Return type boolean

`vgl.texture.`**`textureHandle`**`()`
Get the texture handle (id) of the texture

Returns

Return type

-

`vgl.texture.`**`internalFormat`**`()`
Get internal format of the texture

Returns

Return type

-

`vgl.texture.`**`setInternalFormat`**`(`*internalFormat*`)`

Arguments

- **internalFormat** –

Set internal format of the texture

Returns

Return type boolean

`vgl.texture.`**`pixelFormat`**`()`

Get pixel format of the texture

Returns

Return type

-

`vgl.texture.`**`setPixelFormat`**`(`*pixelFormat*`)`

Arguments

- **pixelFormat** –

Set pixel format of the texture

Returns

Return type boolean

`vgl.texture.`**`pixelDataType`**`()`

Get pixel data type

Returns

Return type

-

`vgl.texture.`**`setPixelDataType`**`(`*pixelDataType*`)`

Arguments

- **pixelDataType** –

Set pixel data type

Returns

Return type boolean

`vgl.texture.`**`computeInternalFormatUsingImage`**`()`

Compute internal format of the texture

`vgl.texture.`**`updateDimensions`**`()`

Update texture dimensions

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.34 vgl.trackballInteractorStyle (class)

vgl.trackballInteractorStyle

class `vgl.`**`trackballInteractorStyle`**`()`

Create a new instance of trackballInteractorStyle

Returns

rtype vgl.trackballInteractorStyle

`vgl.trackballInteractorStyle.`**`handleMouseMove`**`(`*event*`)`

Arguments

- **event** –

Handle mouse move event

Returns

Return type boolean

`vgl.trackballInteractorStyle.`**`handleMouseDown`**`(`*event*`)`

Arguments

- **event** –

Handle mouse down event

Returns

Return type boolean

`vgl.trackballInteractorStyle.`**`handleMouseUp`**`(`*event*`)`

Arguments

- **event** –

Handle mouse up event

Returns

Return type boolean

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.35 vgl.vertexDataP3N3f (class)

class `vgl.`**`vertexDataP3N3f`**`()`

Create a new instance of class vertexDataP3N3f

Returns

rtype vgl.vertexDataP3N3f

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

2.1.36 vgl.vertexDataP3T3f (class)

class `vgl.`**`vertexDataP3T3f`**`()`

Create a new instance of class vertexDataP3T3f

Returns

rtype vgl.vertexDataP3T3f

Documentation generated by jsdoc-toolkit 2.4.0 using jsdoc-toolkit-rst-template

CHAPTER 3

Designer Documentation

CHAPTER 4

About VGL

CHAPTER 5

Custom install Documentation

CHAPTER 6

Operations Documentation