

# Chapter 55

## Mesh class

Inherits: Sprite

### Supported platforms

- All platforms

Available since version: Gideros 2012.09

### Description

Mesh class is used to create and display custom constructed set of triangles (triangle meshes). It basically consists of 4 arrays: vertex, index, color (optional), textureCoordinate (optional) and a texture (optional) and it provides more than one way to set/modify these arrays. **Note 1:** Mesh class doesn't do bounds check. If an element at index array points to a non-existent vertex, the application may crash.

### Common uses and examples

Mesh usage

```
local mesh = Mesh.new()
stage:addChild(mesh)
-- 1. vertex (0, 0)
-- 2. vertex (100, 0)
-- 3. vertex (100, 150)
-- 4. vertex (0, 150)
mesh:setVertexArray(0, 0, 100, 0, 100, 150, 0, 150)

-- 1. triangle from 1, 2 and 3 vertex
-- 2. triangle from 1, 3 and 4 vertex
mesh:setIndexArray(1, 2, 3, 1, 3, 4)

-- 1. vertex 0xff0000 color with 0.5 alpha
-- 2. vertex 0x00ff00 color with 0.7 alpha
-- 3. vertex 0x0000ff color with 1 alpha
-- 4. vertex 0xffff00 color with 0 alpha
mesh:setColorArray(0xff0000, 0.5, 0x00ff00, 0.7, 0x0000ff, 1.0, 0xffff00, 0)
```

## 55.1 Mesh.new

**Available since version:** Gideros 2012.09

### Description

Creates a new Mesh object.

### Syntax:

```
Mesh.new()
```

## 55.2 Mesh:clearColorArray

**Available since version:** Gideros 2012.09

### Description

Clears the color array.

### Syntax:

```
Mesh:clearColorArray()
```

## 55.3 Mesh:clearIndexArray

**Available since version:** Gideros 2012.09

### Description

Clears the index array.

### Syntax:

```
Mesh:clearIndexArray()
```

## 55.4 Mesh:clearTexture

**Available since version:** Gideros 2012.09

### Description

Clears the texture.

### Syntax:

```
Mesh:clearTexture()
```

## 55.5 Mesh:clearTextureCoordinateArray

Available since version: Gideros 2012.09

### Description

Clears the texture coordinate array.

### Syntax:

```
Mesh:clearTextureCoordinateArray()
```

## 55.6 Mesh:clearVertexArray

Available since version: Gideros 2012.09

### Description

Clears the vertex array.

### Syntax:

```
Mesh:clearVertexArray()
```

## 55.7 Mesh:getColor

Available since version: Gideros 2013.06

### Description

Returns color and alpha of the i-th element from color array

### Syntax:

```
Mesh:getColor(i)
```

### Parameters:

- i: (number) index

## 55.8 Mesh:getColorArraySize

Available since version: Gideros 2013.06

### Description

Get size of the Color array

**Syntax:**

```
Mesh:getColorArraySize()
```

## 55.9 Mesh:getIndex

**Available since version:** Gideros 2013.06

**Description**

Returns the i-th element from index array

**Syntax:**

```
Mesh:getIndex(i)
```

**Parameters:**

- i: (number) index

## 55.10 Mesh:getIndexArraySize

**Available since version:** Gideros 2013.06

**Description**

Get size of the Index array

**Syntax:**

```
Mesh:getIndexArraySize()
```

## 55.11 Mesh:getTextureCoordinate

**Available since version:** Gideros 2013.06

**Description**

Returns u and v coordinate of the i-th element from texture coordinate array

**Syntax:**

```
Mesh:getTextureCoordinate(i)
```

**Parameters:**

- i: (number) index

## 55.12 Mesh:getTextureCoordinateArraySize

Available since version: Gideros 2013.06

### Description

Get size of the Texture Coordinate array

### Syntax:

```
Mesh:getTextureCoordinateArraySize()
```

## 55.13 Mesh:getVertex

Available since version: Gideros 2013.06

### Description

Returns x and y coordinate of the i-th element from vertex array

### Syntax:

```
Mesh:getVertex(i)
```

### Parameters:

- i: (number) index

## 55.14 Mesh:getVertexArraySize

Available since version: Gideros 2013.06

### Description

Get size of the Vertices array

### Syntax:

```
Mesh:getVertexArraySize()
```

## 55.15 Mesh:resizeColorArray

Available since version: Gideros 2012.09

## Description

Resizes the color array to contain **size** elements. If **size** is smaller than the current color array size, the content is reduced to its first **size** elements, the rest being dropped. If **size** is greater than the current color array size, the content is expanded by inserting at the end as many copies of 0s as needed to reach a size of **size** elements.

## Syntax:

```
Mesh:resizeColorArray(size)
```

## Parameters:

- **size**: (number) new color array size

## 55.16 Mesh:resizeIndexArray

**Available since version:** Gideros 2012.09

## Description

Resizes the index array to contain **size** elements. If **size** is smaller than the current index array size, the content is reduced to its first **size** elements, the rest being dropped. If **size** is greater than the current index array size, the content is expanded by inserting at the end as many copies of 0s as needed to reach a size of **size** elements.

## Syntax:

```
Mesh:resizeIndexArray(size)
```

## Parameters:

- **size**: (number) new index array size

## 55.17 Mesh:resizeTextureCoordinateArray

**Available since version:** Gideros 2012.09

## Description

Resizes the texture coordinate array to contain **size** elements. If **size** is smaller than the current texture coordinate array size, the content is reduced to its first **size** elements, the rest being dropped. If **size** is greater than the current texture coordinate array size, the content is expanded by inserting at the end as many copies of 0s as needed to reach a size of **size** elements.

## Syntax:

```
Mesh:resizeTextureCoordinateArray(size)
```

**Parameters:**

- size: (number) new texture coordinate array size

## 55.18 Mesh:resizeVertexArray

**Available since version:** Gideros 2012.09

**Description**

Resizes the vertex array to contain **size** elements. If **size** is smaller than the current vertex array size, the content is reduced to its first **size** elements, the rest being dropped. If **size** is greater than the current vertex array size, the content is expanded by inserting at the end as many copies of 0s as needed to reach a size of **size** elements.

**Syntax:**

```
Mesh:resizeVertexArray(size)
```

**Parameters:**

- size: (number) new vertex array size

## 55.19 Mesh:setColor

**Available since version:** Gideros 2012.09

**Description**

Sets a color at color array. Indices are start from 1. If the color array is not large enough, it's expanded automatically.

**Syntax:**

```
Mesh:setColor(i, color, alpha)
```

**Parameters:**

- i: (number) index
- color: (number) color in hexedecial value
- alpha: (number, default=1.0) alpha value

**Examples:**

```

-- set the first 3 colors as (0xff0000, 0.5), (0x00ff00, 0.7)
-- and (0x0000ff, 1.0).
mesh:setColor(1, 0xff0000, 0.5) -- red with 0.5 alpha
mesh:setColor(2, 0x00ff00, 0.7) -- green with 0.7 alpha
mesh:setColor(3, 0x0000ff)      -- blue with 1.0 alpha

```

**55.20 Mesh:setColorArray**

**Available since version:** Gideros 2012.09

**Description**

Assigns new content to the color array, dropping all the elements contained in the color array before the call and replacing them by those specified by the parameters. It accepts multiple values or a Lua array.

**Syntax:**

```
Mesh:setColorArray(colors)
```

**Parameters:**

- colors:

**Examples:**

```

-- set the color array as (0xff0000, 0.5), (0x00ff00, 0.7)
-- and (0x0000ff, 1.0).
mesh:setColorArray(0xff0000, 0.5, 0x00ff00, 0.7, 0x0000ff, 1.0)

-- same as above
mesh:setColorArray{0xff0000, 0.5, 0x00ff00, 0.7, 0x0000ff, 1.0}

```

**55.21 Mesh:setColors**

**Available since version:** Gideros 2012.09

**Description**

Sets zero or more colors at color array with a single function call. It accepts multiple values or a Lua array.



**Syntax:**

```
Mesh:setColors(colors)
```

**Parameters:**

- colors:

**Examples:**

```
-- set 3 colors with separate function calls
mesh:setColor(1, 0xff0000, 0.5)
mesh:setColor(2, 0x00ff00, 0.7)
mesh:setColor(3, 0x0000ff)

-- set 3 colors with one function call
mesh:setColors(1, 0xff0000, 0.5, 2, 0x00ff00, 0.7, 3, 0x0000ff, 1.0)

-- same as above
mesh:setColors{1, 0xff0000, 0.5, 2, 0x00ff00, 0.7, 3, 0x0000ff, 1.0}

-- these two functions do nothing
mesh:setColors()
mesh:setColors{}
```

## 55.22 Mesh:setIndex

**Available since version:** Gideros 2012.09

**Description**

Sets a index at index array. Indices are start from 1. If the index array is not large enough, it's expanded automatically.

**Syntax:**

```
Mesh:setIndex(i, index)
```

**Parameters:**

- i: (number) index
- index: (number) index

**Examples:**

```
-- set the first 3 indices as 10, 11 and 12.
mesh:setIndex(1, 10)
mesh:setIndex(2, 11)
mesh:setIndex(3, 12)
```

## 55.23 Mesh:setIndexArray

Available since version: Gideros 2012.09

**Description**

Assigns new content to the index array, dropping all the elements contained in the index array before the call and replacing them by those specified by the parameters. It accepts multiple values or a Lua array.

**Syntax:**

```
Mesh:setIndexArray(indices)
```

**Parameters:**

- indices:

**Examples:**

```
-- set the index array as 10, 11 and 12.
mesh:setIndexArray(10, 11, 12)

-- same as above
mesh:setIndexArray{10, 11, 12}
```

## 55.24 Mesh:setIndices

Available since version: Gideros 2012.09

**Description**

Sets zero or more indices at index array with a single function call. It accepts multiple values or a Lua array.

**Syntax:**

```
Mesh:setIndices(indices)
```

**Parameters:**

- indices:

**Examples:**

```
-- set 3 indices with separate function calls
mesh:setIndex(1, 10)
mesh:setIndex(2, 11)
mesh:setIndex(3, 12)

-- set 3 indices with one function call
mesh:setIndices(1, 10, 2, 11, 3, 12)

-- same as above
mesh:setIndices{1, 10, 2, 11, 3, 12}

-- these two functions do nothing
mesh:setIndices()
mesh:setIndices{}
```

## 55.25 Mesh:setTexture

Available since version: Gideros 2012.09

**Description**

Sets the texture.

**Syntax:**

```
Mesh:setTexture(texture)
```

**Parameters:**

- texture: (TextureBase)

## 55.26 Mesh:setTextureCoordinate

Available since version: Gideros 2012.09

**Description**

Sets a texture coordinate at texture coordinate array. Indices are start from 1. If the texture coordinate array is not large enough, it's expanded automatically.

**Syntax:**

```
Mesh:setTextureCoordinate(i, u, v)
```

**Parameters:**

- i: (number) index
- u: (number) u coordinate
- v: (number) v coordinate

**Examples:**

```
-- set the first 3 texture coordinates as (0, 0), (100, 0)
-- and (0, 100).
mesh:setTextureCoordinate(1, 0, 0)
mesh:setTextureCoordinate(2, 100, 0)
mesh:setTextureCoordinate(3, 0, 100)
```

**55.27 Mesh:setTextureCoordinateArray**

Available since version: Gideros 2012.09

**Description**

Assigns new content to the texture coordinate array, dropping all the elements contained in the texture coordinate array before the call and replacing them by those specified by the parameters. It accepts multiple values or a Lua array.

**Syntax:**

```
Mesh:setTextureCoordinateArray(textureCoordinates)
```

**Parameters:**

- textureCoordinates:

**Examples:**

```
-- set the color array as (0, 0), (100, 0) and (0, 100)
mesh:setTextureCoordinateArray(0, 0, 100, 0, 0, 100)

-- same as above
mesh:setTextureCoordinateArray{0, 0, 100, 0, 0, 100}
```

## 55.28 Mesh:setTextureCoordinates

Available since version: Gideros 2012.09

### Description

### Syntax:

```
Mesh:setTextureCoordinates(textureCoordinates)
```

### Parameters:

- textureCoordinates: Sets zero or more texture coordinates at texture coordinate array with a single function call. It accepts multiple values or a Lua array.

### Examples:

```
-- set 3 texture coordinates with separate function calls
mesh:setTextureCoordinate(1, 0, 0)
mesh:setTextureCoordinate(2, 100, 0)
mesh:setTextureCoordinate(3, 0, 100)

-- set 3 texture coordinates with one function call
mesh:setTextureCoordinates(1, 0, 0, 2, 100, 0, 3, 0, 100)

-- same as above
mesh:setTextureCoordinates{1, 0, 0, 2, 100, 0, 3, 0, 100}

-- these two functions do nothing
mesh:setTextureCoordinates()
mesh:setTextureCoordinates{}
```

## 55.29 Mesh:setVertex

Available since version: Gideros 2012.09

### Description

Sets a vertex at vertex array. Indices are start from 1. If the vertex array is not large enough, it's expanded automatically.

### Syntax:

```
Mesh:setVertex(i, x, y)
```

**Parameters:**

- i: (number) index
- x: (number) x coordinate
- y: (number) y coordinate

**Examples:**

```
-- set the first 3 vertex positions as (100, 100), (150, 100) and (100, 150).
mesh:setVertex(1, 100, 100)
mesh:setVertex(2, 150, 100)
mesh:setVertex(3, 100, 150)
```

**55.30 Mesh:setVertexArray**

Available since version: Gideros 2012.09

**Description**

Assigns new content to the vertex array, dropping all the elements contained in the vertex array before the call and replacing them by those specified by the parameters. It accepts multiple values or a Lua array.

**Syntax:**

```
Mesh:setVertexArray(vertices)
```

**Parameters:**

- vertices:

**Examples:**

```
-- set the vertex array as (100, 100), (150, 100) and (100, 150)
mesh:setVertexArray(100, 100, 150, 100, 100, 150)

-- same as above
mesh:setVertexArray{100, 100, 150, 100, 100, 150}
```

**55.31 Mesh:setVertices**

Available since version: Gideros 2012.09

## Description

Sets zero or more vertices at vertex array with a single function call. It accepts multiple values or a Lua array.

## Syntax:

```
Mesh:setVertices(vertices)
```

## Parameters:

- vertices:

## Examples:

```
-- set 3 vertices with separate function calls
mesh:setVertex(1, 100, 100)
mesh:setVertex(2, 150, 100)
mesh:setVertex(3, 100, 150)

-- set 3 vertices with one function call
mesh:setVertices(1, 100, 100, 2, 150, 100, 3, 100, 150)

-- same as above
mesh:setVertices{1, 100, 100, 2, 150, 100, 3, 100, 150}

-- these two functions do nothing
mesh:setVertices()
mesh:setVertices{}
```