SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [File List](http://docs.google.com/files.htm)
* [include](http://docs.google.com/dir_f3190241575fd2bd132a392ae6942f4a.htm)
* [SFML](http://docs.google.com/dir_692f376662c82a26cfe4cfa3aceebe24.htm)
* [Graphics](http://docs.google.com/dir_aaa96c3797a59111c2945d0d638ce5cf.htm)

VertexArray.hpp

1

2 //

3 // SFML - Simple and Fast Multimedia Library

4 // Copyright (C) 2007-2013 Laurent Gomila (laurent.gom@gmail.com)

5 //

6 // This software is provided 'as-is', without any express or implied warranty.

7 // In no event will the authors be held liable for any damages arising from the use of this software.

8 //

9 // Permission is granted to anyone to use this software for any purpose,

10 // including commercial applications, and to alter it and redistribute it freely,

11 // subject to the following restrictions:

12 //

13 // 1. The origin of this software must not be misrepresented;

14 // you must not claim that you wrote the original software.

15 // If you use this software in a product, an acknowledgment

16 // in the product documentation would be appreciated but is not required.

17 //

18 // 2. Altered source versions must be plainly marked as such,

19 // and must not be misrepresented as being the original software.

20 //

21 // 3. This notice may not be removed or altered from any source distribution.

22 //

24

25 #ifndef SFML\_VERTEXARRAY\_HPP

26 #define SFML\_VERTEXARRAY\_HPP

27

29 // Headers

31 #include <SFML/Graphics/Export.hpp>

32 #include <SFML/Graphics/Vertex.hpp>

33 #include <SFML/Graphics/PrimitiveType.hpp>

34 #include <SFML/Graphics/Rect.hpp>

35 #include <SFML/Graphics/Drawable.hpp>

36 #include <vector>

37

38

39 namespace sf

40 {

[45](http://docs.google.com/classsf_1_1VertexArray.htm) class SFML\_GRAPHICS\_API [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm) : public [Drawable](http://docs.google.com/classsf_1_1Drawable.htm)

46 {

47 public :

48

55  [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm)();

56

64  explicit [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm)([PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) type, unsigned int vertexCount = 0);

65

72  unsigned int getVertexCount() const;

73

88  [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)& operator [](unsigned int index);

89

104  const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)& operator [](unsigned int index) const;

105

115  void clear();

116

129  void resize(unsigned int vertexCount);

130

137  void append(const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)& vertex);

138

153  void setPrimitiveType([PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) type);

154

161  [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) getPrimitiveType() const;

162

172  [FloatRect](http://docs.google.com/classsf_1_1Rect.htm) getBounds() const;

173

174 private :

175

183  virtual void draw([RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm)& target, [RenderStates](http://docs.google.com/classsf_1_1RenderStates.htm) states) const;

184

185 private:

186

188  // Member data

190  std::vector<Vertex> m\_vertices;

191  [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) m\_primitiveType;

192 };

193

194 } // namespace sf

195

196

197 #endif // SFML\_VERTEXARRAY\_HPP

198

199

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::