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)

View.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\_VIEW\_HPP

26 #define SFML\_VIEW\_HPP

27

29 // Headers

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

32 #include <SFML/Graphics/Rect.hpp>

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

34 #include <SFML/System/Vector2.hpp>

35

36

37 namespace sf

38 {

[43](http://docs.google.com/classsf_1_1View.htm) class SFML\_GRAPHICS\_API [View](http://docs.google.com/classsf_1_1View.htm)

44 {

45 public :

46

53  [View](http://docs.google.com/classsf_1_1View.htm)();

54

61  explicit [View](http://docs.google.com/classsf_1_1View.htm)(const [FloatRect](http://docs.google.com/classsf_1_1Rect.htm)& rectangle);

62

70  [View](http://docs.google.com/classsf_1_1View.htm)(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& center, const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& size);

71

81  void setCenter(float x, float y);

82

91  void setCenter(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& center);

92

102  void setSize(float width, float height);

103

112  void setSize(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& size);

113

124  void setRotation(float angle);

125

141  void setViewport(const [FloatRect](http://docs.google.com/classsf_1_1Rect.htm)& viewport);

142

153  void reset(const [FloatRect](http://docs.google.com/classsf_1_1Rect.htm)& rectangle);

154

163  const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& getCenter() const;

164

173  const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& getSize() const;

174

183  float getRotation() const;

184

193  const [FloatRect](http://docs.google.com/classsf_1_1Rect.htm)& getViewport() const;

194

204  void move(float offsetX, float offsetY);

205

214  void move(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& offset);

215

224  void rotate(float angle);

225

241  void zoom(float factor);

242

253  const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& getTransform() const;

254

265  const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& getInverseTransform() const;

266

267 private :

268

270  // Member data

272  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) m\_center;

273  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) m\_size;

274  float m\_rotation;

275  [FloatRect](http://docs.google.com/classsf_1_1Rect.htm) m\_viewport;

276  mutable [Transform](http://docs.google.com/classsf_1_1Transform.htm) m\_transform;

277  mutable [Transform](http://docs.google.com/classsf_1_1Transform.htm) m\_inverseTransform;

278  mutable bool m\_transformUpdated;

279  mutable bool m\_invTransformUpdated;

280 };

281

282 } // namespace sf

283

284

285 #endif // SFML\_VIEW\_HPP

286

287

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