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)

RenderTarget.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\_RENDERTARGET\_HPP

26 #define SFML\_RENDERTARGET\_HPP

27

29 // Headers

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

32 #include <SFML/Graphics/Color.hpp>

33 #include <SFML/Graphics/Rect.hpp>

34 #include <SFML/Graphics/View.hpp>

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

36 #include <SFML/Graphics/BlendMode.hpp>

37 #include <SFML/Graphics/RenderStates.hpp>

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

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

40 #include <SFML/System/NonCopyable.hpp>

41

42

43 namespace sf

44 {

45 class Drawable;

46

[51](http://docs.google.com/classsf_1_1RenderTarget.htm) class SFML\_GRAPHICS\_API [RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm) : [NonCopyable](http://docs.google.com/classsf_1_1NonCopyable.htm)

52 {

53 public :

54

59  virtual ~[RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm)();

60

70  void clear(const [Color](http://docs.google.com/classsf_1_1Color.htm)& color = [Color](http://docs.google.com/classsf_1_1Color.htm)(0, 0, 0, 255));

71

91  void setView(const [View](http://docs.google.com/classsf_1_1View.htm)& view);

92

101  const [View](http://docs.google.com/classsf_1_1View.htm)& getView() const;

102

114  const [View](http://docs.google.com/classsf_1_1View.htm)& getDefaultView() const;

115

129  [IntRect](http://docs.google.com/classsf_1_1Rect.htm) getViewport(const [View](http://docs.google.com/classsf_1_1View.htm)& view) const;

130

149  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) mapPixelToCoords(const [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm)& point) const;

150

180  [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm) mapPixelToCoords(const [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm)& point, const [View](http://docs.google.com/classsf_1_1View.htm)& view) const;

181

200  [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm) mapCoordsToPixel(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& point) const;

201

227  [Vector2i](http://docs.google.com/classsf_1_1Vector2.htm) mapCoordsToPixel(const [Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)& point, const [View](http://docs.google.com/classsf_1_1View.htm)& view) const;

228

236  void draw(const [Drawable](http://docs.google.com/classsf_1_1Drawable.htm)& drawable, const [RenderStates](http://docs.google.com/classsf_1_1RenderStates.htm)& states = [RenderStates::Default](http://docs.google.com/classsf_1_1RenderStates.htm#ad29672df29f19ce50c3021d95f2bb062));

237

247  void draw(const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)\* vertices, unsigned int vertexCount,

248  [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) type, const [RenderStates](http://docs.google.com/classsf_1_1RenderStates.htm)& states = [RenderStates::Default](http://docs.google.com/classsf_1_1RenderStates.htm#ad29672df29f19ce50c3021d95f2bb062));

249

256  virtual [Vector2u](http://docs.google.com/classsf_1_1Vector2.htm) getSize() const = 0;

257

290  void pushGLStates();

291

301  void popGLStates();

302

324  void resetGLStates();

325

326 protected :

327

332  [RenderTarget](http://docs.google.com/classsf_1_1RenderTarget.htm)();

333

341  void initialize();

342

343 private:

344

349  void applyCurrentView();

350

357  void applyBlendMode([BlendMode](http://docs.google.com/group__graphics.htm#ga80c52fe2f7050d7f7573b7ed3c995388) mode);

358

365  void applyTransform(const [Transform](http://docs.google.com/classsf_1_1Transform.htm)& transform);

366

373  void applyTexture(const [Texture](http://docs.google.com/classsf_1_1Texture.htm)\* texture);

374

381  void applyShader(const [Shader](http://docs.google.com/classsf_1_1Shader.htm)\* shader);

382

395  virtual bool activate(bool active) = 0;

396

401  struct StatesCache

402  {

403  enum {VertexCacheSize = 4};

404

405  bool glStatesSet;

406  bool viewChanged;

407  [BlendMode](http://docs.google.com/group__graphics.htm#ga80c52fe2f7050d7f7573b7ed3c995388) lastBlendMode;

408  Uint64 lastTextureId;

409  bool useVertexCache;

410  [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) vertexCache[VertexCacheSize];

411  };

412

414  // Member data

416  [View](http://docs.google.com/classsf_1_1View.htm) m\_defaultView;

417  [View](http://docs.google.com/classsf_1_1View.htm) m\_view;

418  StatesCache m\_cache;

419 };

420

421 } // namespace sf

422

423

424 #endif // SFML\_RENDERTARGET\_HPP

425

426

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