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)

Rect.inl

1

2 //

3 // SFML - Simple and Fast Multimedia Library

4 // Copyright (C) 2007-2012 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

27 template <typename T>

28 Rect<T>::Rect() :

29 left (0),

30 top (0),

31 width (0),

32 height(0)

33 {

34

35 }

36

37

39 template <typename T>

40 Rect<T>::Rect(T rectLeft, T rectTop, T rectWidth, T rectHeight) :

41 left (rectLeft),

42 top (rectTop),

43 width (rectWidth),

44 height(rectHeight)

45 {

46

47 }

48

49

51 template <typename T>

52 Rect<T>::Rect(const Vector2<T>& position, const Vector2<T>& size) :

53 left (position.x),

54 top (position.y),

55 width (size.x),

56 height(size.y)

57 {

58

59 }

60

61

63 template <typename T>

64 template <typename U>

65 Rect<T>::Rect(const Rect<U>& rectangle) :

66 left (static\_cast<T>(rectangle.left)),

67 top (static\_cast<T>(rectangle.top)),

68 width (static\_cast<T>(rectangle.width)),

69 height(static\_cast<T>(rectangle.height))

70 {

71 }

72

73

75 template <typename T>

76 bool Rect<T>::contains(T x, T y) const

77 {

78  return (x >= left) && (x < left + width) && (y >= top) && (y < top + height);

79 }

80

81

83 template <typename T>

84 bool Rect<T>::contains(const Vector2<T>& point) const

85 {

86  return contains(point.x, point.y);

87 }

88

89

91 template <typename T>

92 bool Rect<T>::intersects(const Rect<T>& rectangle) const

93 {

94  Rect<T> intersection;

95  return intersects(rectangle, intersection);

96 }

97

98

100 template <typename T>

101 bool Rect<T>::intersects(const Rect<T>& rectangle, Rect<T>& intersection) const

102 {

103  // Compute the intersection boundaries

104  T interLeft = std::max(left, rectangle.left);

105  T interTop = std::max(top, rectangle.top);

106  T interRight = std::min(left + width, rectangle.left + rectangle.width);

107  T interBottom = std::min(top + height, rectangle.top + rectangle.height);

108

109  // If the intersection is valid (positive non zero area), then there is an intersection

110  if ((interLeft < interRight) && (interTop < interBottom))

111  {

112  intersection = Rect<T>(interLeft, interTop, interRight - interLeft, interBottom - interTop);

113  return true;

114  }

115  else

116  {

117  intersection = Rect<T>(0, 0, 0, 0);

118  return false;

119  }

120 }

121

122

124 template <typename T>

125 inline bool [operator ==](http://docs.google.com/classsf_1_1Color.htm#a2adc3f68860f7aa5e4d7c79dcbb31d30)(const Rect<T>& left, const Rect<T>& right)

126 {

127  return (left.left == right.left) && (left.width == right.width) &&

128  (left.top == right.top) && (left.height == right.height);

129 }

130

131

133 template <typename T>

134 inline bool [operator !=](http://docs.google.com/classsf_1_1Color.htm#a394c3495753c4b17f9cd45556ef00b8c)(const Rect<T>& left, const Rect<T>& right)

135 {

136  return !(left == right);

137 }

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