

GCC Code Coverage Report

Directory: ./

File: Draw.cpp

Date: 2021-11-17 01:36:45

Exec Total Coverage

Lines: 35 36 97.2%
Branches: 29 56 51.8%

Line	Branch	Exec	Source
1			/**
2			* @file Draw.cpp
3			* @brief Draw implementation, all drawing actions are commands.
4			* @author Mike and Dennis Ping
5			* @date 2021-11-13
6			*****/
7			
8			// Include our Third-Party SFML header
9			#include <SFML/Graphics/Color.hpp>
10			// Include standard library C++ libraries.
11			#include <string>
12			#include <map>
13			// Project header files
14			#include "Draw.hpp"
15			#include "Command.hpp"
16			
17			/*! \brief Constructor for a Draw command.
18			* Stores the current (x,y) mouse coordinates upon creation.
19			*
20			*/
21	▶ 3/6	41636	Draw::Draw(int x, int y, sf::Image& image, sf::Color paintbrushColor) : Command(image),
22		10409	xCoord(x),
23		10409	yCoord(y),
24		31227	currColor(paintbrushColor) {
25		20818	}
26			
27		31226	Draw::~Draw(){}
28			
29			/*! \brief Store the previous pixel color and set the new pixel color at (x,y).
30			*
31			*/
32		5010	bool Draw::execute() {
33	▶ 4/8	5010	if (xCoord < 0 xCoord >= (int)m_image.getSize().x yCoord < 0 yCoord >= (int)m_image.getSize().y) {
34		x	return false;
35			}
36		5010	prevColor = m_image.getPixel(xCoord, yCoord);
37	▶ 2/2	5010	if (m_image.getPixel(xCoord, yCoord) == currColor) {
38		3377	return false;
39			}
40		1633	m_image.setPixel(xCoord, yCoord, currColor);
41		1633	return true;
42		5010	}
43			
44			/*! \brief Set the pixel color at (x,y) to the previous pixel color.
45			*
46			*/
47		291	bool Draw::undo(){
48		291	m_image.setPixel(xCoord, yCoord, prevColor);
49		291	return true;
50			}
51			
52			/*! \brief Set the pixel color at (x,y) to the original pixel color.
53			*
54			*/
55		97	bool Draw::redo() {
56		97	m_image.setPixel(xCoord, yCoord, currColor);
57		97	return true;
58			}
59			
60			/*! \brief Return the (x,y) coordinates of this Draw command.
61			*
62			*/
63		776	std::pair<int, int> Draw::getCoords() {
64		776	return std::make_pair(xCoord, yCoord);
65			}
66			
67			/*! \brief Get a string representation of this Draw command in the form (x, y, color).
68			*
69			*/
70		1	std::string Draw::getDescription() {
71			// C++ does not know how to hash an sf::Color object, so we must use the literal integer value.
72	▶ 1/2	1	const std::map<int, std::string> colorMap {
73	▶ 1/2	1	{255, "Black"},
74	▶ 1/2	1	{4294967295, "White"},
75	▶ 1/2	1	{4278190335, "Red"},
76	▶ 1/2	1	{16711935, "Green"},

77	▶ 1/2	1	{65535, "Blue"},
78	▶ 1/2	1	{4294902015, "Yellow"},
79	▶ 1/2	1	{4278255615, "Magenta"},
80	▶ 1/2	1	{16777215, "Cyan"},
81	▶ 1/2	1	{0, "Transparent"}
82			};
83	▶ 10/20	1	return "Draw (" + std::to_string(xCoord) + ", " + std::to_string(yCoord) + ", " + colorMap.at(currColor.toInteger())
84		1	};
85			