

Simple Paint

2.1

Wygenerowano przez Doxygen 1.8.14

Spis treści

1	Lista błędów	1
2	Indeks hierarchiczny	3
2.1	Hierarchia klas	3
3	Indeks klas	5
3.1	Lista klas	5
4	Indeks plików	7
4.1	Lista plików	7
5	Dokumentacja klas	11
5.1	Dokumentacja klasy base_window	11
5.1.1	Opis szczegółowy	12
5.1.2	Dokumentacja konstruktora i destruktora	12
5.1.2.1	~base_window()	12
5.1.3	Dokumentacja atrybutów składowych	12
5.1.3.1	menu_height	12
5.1.3.2	window_area	12
5.1.3.3	window_height	12
5.1.3.4	window_width	12
5.2	Dokumentacja klasy basic_filter	13
5.2.1	Opis szczegółowy	13
5.2.2	Dokumentacja funkcji składowych	13
5.2.2.1	delete_event()	13

5.2.2.2	destroy()	14
5.2.2.3	do_filter()	14
5.2.2.4	open_fiter_properties()	14
5.2.2.5	update_filter_information()	15
5.2.3	Dokumentacja atrybutów składowych	16
5.2.3.1	type	16
5.3	Dokumentacja struktury image_container::basic_image_info	16
5.3.1	Opis szczegółowy	16
5.3.2	Dokumentacja atrybutów składowych	16
5.3.2.1	height	17
5.3.2.2	image_type	17
5.3.2.3	number_of_colors	17
5.3.2.4	width	17
5.4	Dokumentacja struktury bmp_image_container::bmp_file_info	17
5.4.1	Opis szczegółowy	18
5.4.2	Dokumentacja atrybutów składowych	18
5.4.2.1	bits_per_pixel	18
5.4.2.2	compression	18
5.4.2.3	header_size	18
5.4.2.4	height	18
5.4.2.5	horizontal_resolution	18
5.4.2.6	image_size	18
5.4.2.7	number_of_color_in_palette	19
5.4.2.8	number_of_important_color	19
5.4.2.9	planes	19
5.4.2.10	size	19
5.4.2.11	vertical_resolution	19
5.4.2.12	width	19
5.5	Dokumentacja klasy bmp_image_container	20
5.5.1	Opis szczegółowy	20

5.5.2	Dokumentacja konstruktora i destruktora	20
5.5.2.1	bmp_image_container() [1/2]	20
5.5.2.2	bmp_image_container() [2/2]	21
5.5.2.3	~bmp_image_container()	21
5.5.3	Dokumentacja funkcji składowych	21
5.5.3.1	save_to_file()	21
5.6	Dokumentacja klasy brush	22
5.6.1	Opis szczegółowy	22
5.6.2	Dokumentacja konstruktora i destruktora	22
5.6.2.1	brush()	23
5.6.3	Dokumentacja funkcji składowych	23
5.6.3.1	is_a_brush()	23
5.6.3.2	use_tool()	23
5.7	Dokumentacja struktury image_container::color	24
5.7.1	Opis szczegółowy	24
5.7.2	Dokumentacja atrybutów składowych	24
5.7.2.1	alpha	24
5.7.2.2	blue	24
5.7.2.3	green	24
5.7.2.4	red	24
5.8	Dokumentacja klasy color_inverse	25
5.8.1	Opis szczegółowy	25
5.8.2	Dokumentacja konstruktora i destruktora	25
5.8.2.1	color_inverse()	25
5.8.3	Dokumentacja funkcji składowych	25
5.8.3.1	do_filter()	26
5.8.3.2	open_fiter_properties()	26
5.9	Dokumentacja klasy color_picker	26
5.9.1	Opis szczegółowy	27
5.9.2	Dokumentacja konstruktora i destruktora	27

5.9.2.1	color_picker()	27
5.9.3	Dokumentacja funkcji składowych	27
5.9.3.1	isBrush()	27
5.9.3.2	use_tool()	27
5.10	Dokumentacja klasy ellipse	28
5.10.1	Opis szczegółowy	29
5.10.2	Dokumentacja konstruktora i destruktora	29
5.10.2.1	ellipse()	29
5.10.3	Dokumentacja funkcji składowych	29
5.10.3.1	draw_shape()	29
5.10.3.2	open_tool_properties()	29
5.10.3.3	PaintPixel()	30
5.10.3.4	set_cicle_flag()	30
5.10.4	Dokumentacja atrybutów składowych	30
5.10.4.1	is_circle	30
5.11	Dokumentacja klasy file_manager	31
5.11.1	Opis szczegółowy	32
5.11.2	Dokumentacja konstruktora i destruktora	32
5.11.2.1	file_manager()	32
5.11.2.2	~file_manager()	32
5.11.3	Dokumentacja funkcji składowych	32
5.11.3.1	get_file_manager_mode()	33
5.11.3.2	get_file_path()	33
5.11.3.3	get_image_dimentionations()	33
5.11.3.4	load_image() [1/2]	33
5.11.3.5	load_image() [2/2]	34
5.11.3.6	open_gtk_window()	34
5.11.3.7	open_open_file_widget()	34
5.11.3.8	open_save_file_widget()	34
5.11.3.9	print_image()	35

5.11.3.10 quick_save_button_clicked()	35
5.11.3.11 reset_state()	35
5.11.3.12 save_window_to_image()	35
5.11.3.13 update_file_manager_information()	36
5.11.4 Dokumentacja atrybutów składowych	36
5.11.4.1 current_file_type	36
5.11.4.2 current_mode	36
5.11.4.3 file_path	36
5.11.4.4 image	36
5.11.4.5 working_on_file	37
5.12 Dokumentacja klasy filter_menu	37
5.12.1 Opis szczegółowy	38
5.12.2 Dokumentacja konstruktora i destruktor	38
5.12.2.1 filter_menu()	38
5.12.3 Dokumentacja funkcji składowych	38
5.12.3.1 get_filter_response()	38
5.12.3.2 get_filter_type()	38
5.12.3.3 open_gtk_window()	39
5.12.4 Dokumentacja atrybutów składowych	39
5.12.4.1 filter_response	39
5.12.4.2 filter_types	39
5.12.4.3 number_of_filters	39
5.13 Dokumentacja struktury filter_type_data	39
5.13.1 Opis szczegółowy	40
5.13.2 Dokumentacja atrybutów składowych	40
5.13.2.1 response	40
5.13.2.2 type	40
5.14 Dokumentacja klasy flood_fill	40
5.14.1 Opis szczegółowy	41
5.14.2 Dokumentacja konstruktora i destruktor	41

5.14.2.1	flood_fill()	41
5.14.3	Dokumentacja funkcji składowych	42
5.14.3.1	calculate_tolerance()	42
5.14.3.2	isBrush()	42
5.14.3.3	open_tool_properties()	42
5.14.3.4	set_confirmed_flag()	42
5.14.3.5	use_tool()	43
5.14.3.6	value_changed()	43
5.14.4	Dokumentacja atrybutów składowych	43
5.14.4.1	tolerance_ratio	43
5.15	Dokumentacja klasy grayscale	44
5.15.1	Opis szczegółowy	44
5.15.2	Dokumentacja konstruktora i destruktora	44
5.15.2.1	grayscale()	44
5.15.3	Dokumentacja funkcji składowych	45
5.15.3.1	do_filter()	45
5.15.3.2	open_fiter_properties()	45
5.16	Dokumentacja klasy gtk_base_window	45
5.16.1	Opis szczegółowy	46
5.16.2	Dokumentacja konstruktora i destruktora	46
5.16.2.1	~gtk_base_window()	46
5.16.3	Dokumentacja funkcji składowych	46
5.16.3.1	delete_event()	46
5.16.3.2	destroy()	47
5.16.3.3	open_gtk_window()	47
5.16.4	Dokumentacja atrybutów składowych	47
5.16.4.1	failed_to_load_resources	47
5.16.4.2	resource_file_path	48
5.16.4.3	resource_strings	48
5.17	Dokumentacja klasy image_container	48

5.17.1	Opis szczegółowy	49
5.17.2	Dokumentacja składowych definicji typu	49
5.17.2.1	ARGB_8888_color	49
5.17.3	Dokumentacja składowych wyliczanych	49
5.17.3.1	file_type	49
5.17.4	Dokumentacja konstruktora i destruktor	50
5.17.4.1	image_container()	50
5.17.4.2	~image_container()	50
5.17.5	Dokumentacja funkcji składowych	50
5.17.5.1	get_file_type()	50
5.17.5.2	get_image_info()	51
5.17.5.3	image_path()	51
5.17.5.4	operator[]()	51
5.17.5.5	save_to_file()	51
5.17.5.6	set_pixels_form_array()	51
5.17.6	Dokumentacja atrybutów składowych	52
5.17.6.1	image_info	52
5.17.6.2	image_pixel_array	52
5.18	Dokumentacja klasy line	52
5.18.1	Opis szczegółowy	53
5.18.2	Dokumentacja konstruktora i destruktor	53
5.18.2.1	line()	53
5.18.3	Dokumentacja funkcji składowych	53
5.18.3.1	draw_shape()	53
5.19	Dokumentacja klasy menu	54
5.19.1	Opis szczegółowy	55
5.19.2	Dokumentacja konstruktora i destruktor	55
5.19.2.1	menu()	56
5.19.2.2	~menu()	56
5.19.3	Dokumentacja funkcji składowych	56

5.19.3.1	add_color_to_queue()	56
5.19.3.2	get_clicked_button()	56
5.19.3.3	is_in_buttons_area()	57
5.19.3.4	is_in_color_menu()	57
5.19.3.5	print_current_color()	57
5.19.3.6	print_menu()	58
5.19.3.7	update_brush_size_state()	58
5.19.3.8	update_colors()	58
5.19.3.9	update_tool_pointer()	58
5.19.3.10	update_tool_state()	58
5.19.3.11	update_window_information()	59
5.19.4	Dokumentacja atrybutów składowych	59
5.19.4.1	color_buffer	59
5.19.4.2	COLOR_BUFFER_MAX_SIZE	59
5.19.4.3	colors_to_print	59
5.19.4.4	menu_buttons_images_paths	60
5.19.4.5	menu_buttons_labels	60
5.19.4.6	NUMBER_OF_BUTTONS	60
5.19.4.7	NUMBER_OF_LEFT_ELEMENTS	60
5.19.4.8	NUMBER_OF_TOOLS	60
5.19.4.9	tool_labels	60
5.19.4.10	tool_prt	60
5.19.4.11	tools_images_paths	61
5.20	Dokumentacja klasy paint_exception	61
5.20.1	Opis szczegółowy	62
5.20.2	Dokumentacja składowych wyliczanych	62
5.20.2.1	exception_type	62
5.20.3	Dokumentacja konstruktora i destruktor	63
5.20.3.1	paint_exception() [1/3]	63
5.20.3.2	paint_exception() [2/3]	63

5.20.3.3	<code>paint_exception()</code> [3/3]	64
5.20.3.4	<code>~paint_exception()</code>	64
5.20.4	Dokumentacja funkcji składowych	64
5.20.4.1	<code>what()</code>	64
5.20.5	Dokumentacja atrybutów składowych	65
5.20.5.1	<code>exception_string</code>	65
5.20.5.2	<code>exception_strings</code>	65
5.20.5.3	<code>number_of_exceptions</code>	66
5.21	Dokumentacja klasy <code>pixelate</code>	66
5.21.1	Opis szczegółowy	67
5.21.2	Dokumentacja konstruktora i destruktora	67
5.21.2.1	<code>pixelate()</code>	67
5.21.3	Dokumentacja funkcji składowych	67
5.21.3.1	<code>do_filter()</code>	67
5.21.3.2	<code>open_fiter_properties()</code>	67
5.21.3.3	<code>set_confirmed_flag()</code>	67
5.21.3.4	<code>value_changed()</code>	68
5.21.4	Dokumentacja atrybutów składowych	68
5.21.4.1	<code>cell_size</code>	68
5.22	Dokumentacja klasy <code>png_image_container</code>	68
5.22.1	Opis szczegółowy	69
5.22.2	Dokumentacja konstruktora i destruktora	69
5.22.2.1	<code>png_image_container()</code> [1/2]	69
5.22.2.2	<code>png_image_container()</code> [2/2]	69
5.22.2.3	<code>~png_image_container()</code>	70
5.22.3	Dokumentacja funkcji składowych	70
5.22.3.1	<code>save_to_file()</code>	70
5.23	Dokumentacja struktury <code>ppm_image_container::ppm_file_info</code>	70
5.23.1	Opis szczegółowy	70
5.23.2	Dokumentacja atrybutów składowych	71

5.23.2.1	bits_per_color	71
5.23.2.2	height	71
5.23.2.3	width	71
5.24	Dokumentacja klasy ppm_image_container	71
5.24.1	Opis szczegółowy	72
5.24.2	Dokumentacja konstruktora i destruktora	72
5.24.2.1	ppm_image_container() [1/2]	72
5.24.2.2	ppm_image_container() [2/2]	72
5.24.2.3	~ppm_image_container()	73
5.24.3	Dokumentacja funkcji składowych	73
5.24.3.1	save_to_file()	73
5.25	Dokumentacja klasy rectangle	73
5.25.1	Opis szczegółowy	74
5.25.2	Dokumentacja konstruktora i destruktora	74
5.25.2.1	rectangle()	74
5.25.3	Dokumentacja funkcji składowych	74
5.25.3.1	draw_shape()	74
5.25.3.2	open_tool_properties()	75
5.25.3.3	set_square_flag()	75
5.25.4	Dokumentacja atrybutów składowych	75
5.25.4.1	is_square	75
5.26	Dokumentacja klasy sepia	76
5.26.1	Opis szczegółowy	76
5.26.2	Dokumentacja konstruktora i destruktora	76
5.26.2.1	sepia()	76
5.26.3	Dokumentacja funkcji składowych	76
5.26.3.1	do_filter()	77
5.26.3.2	open_fiter_properties()	77
5.27	Dokumentacja klasy shape	77
5.27.1	Opis szczegółowy	78

5.27.2	Dokumentacja funkcji składowych	78
5.27.2.1	draw_shape()	78
5.27.2.2	is_brush()	78
5.27.2.3	use_tool()	79
5.27.3	Dokumentacja atrybutów składowych	79
5.27.3.1	copied_pixels	79
5.27.3.2	x_begining	79
5.27.3.3	y_begining	79
5.28	Dokumentacja klasy spray	80
5.28.1	Opis szczegółowy	81
5.28.2	Dokumentacja konstruktora i destruktora	81
5.28.2.1	spray()	81
5.28.3	Dokumentacja funkcji składowych	81
5.28.3.1	open_tool_properties()	81
5.28.3.2	PaintPixel()	81
5.28.3.3	set_confirmed_flag()	82
5.28.3.4	value_changed()	82
5.28.4	Dokumentacja atrybutów składowych	82
5.28.4.1	dots_ratio	82
5.28.4.2	generator	83
5.29	Dokumentacja klasy star_brush	83
5.29.1	Opis szczegółowy	83
5.29.2	Dokumentacja konstruktora i destruktora	84
5.29.2.1	star_brush()	84
5.29.3	Dokumentacja funkcji składowych	84
5.29.3.1	PaintPixel()	84
5.30	Dokumentacja klasy tool	84
5.30.1	Opis szczegółowy	86
5.30.2	Dokumentacja konstruktora i destruktora	86
5.30.2.1	~tool()	86

5.30.3 Dokumentacja funkcji składowych	86
5.30.3.1 delete_event()	86
5.30.3.2 destroy()	87
5.30.3.3 get_brush_size()	87
5.30.3.4 get_current_color()	87
5.30.3.5 get_tool_type()	88
5.30.3.6 is_a_brush()	88
5.30.3.7 open_tool_properties()	88
5.30.3.8 operator=()	88
5.30.3.9 PaintPixel()	88
5.30.3.10 resize_tool_size()	89
5.30.3.11 update_current_color()	89
5.30.3.12 update_window_information()	89
5.30.3.13 use_tool()	90
5.30.4 Dokumentacja atrybutów składowych	90
5.30.4.1 brush_size	90
5.30.4.2 current_color	91
5.30.4.3 pressed_left_button	91
5.30.4.4 pressed_left_button_second_time	91
5.30.4.5 type	91
5.31 Dokumentacja klasy tool_menu	91
5.31.1 Opis szczegółowy	92
5.31.2 Dokumentacja konstruktora i destruktor	93
5.31.2.1 tool_menu()	93
5.31.2.2 ~tool_menu()	93
5.31.3 Dokumentacja funkcji składowych	93
5.31.3.1 convert_from_Uint8()	93
5.31.3.2 get_color_response()	93
5.31.3.3 get_tool_response()	94
5.31.3.4 get_tool_type()	94

5.31.3.5	<code>open_gtk_window()</code>	94
5.31.3.6	<code>set_current_color()</code>	94
5.31.3.7	<code>set_current_tool()</code>	95
5.31.3.8	<code>trigger_color_chooser()</code>	95
5.31.4	Dokumentacja atrybutów składowych	95
5.31.4.1	<code>new_color</code>	95
5.31.4.2	<code>number_of_tools</code>	95
5.31.4.3	<code>RGB</code>	96
5.31.4.4	<code>tool_response</code>	96
5.31.4.5	<code>tool_types_array</code>	96
5.32	Dokumentacja struktury <code>tool_type_data</code>	96
5.32.1	Opis szczegółowy	96
5.32.2	Dokumentacja atrybutów składowych	96
5.32.2.1	<code>response</code>	97
5.32.2.2	<code>type</code>	97
5.33	Dokumentacja klasy <code>triangle</code>	97
5.33.1	Opis szczegółowy	98
5.33.2	Dokumentacja konstruktora i destruktora	98
5.33.2.1	<code>triangle()</code>	98
5.33.3	Dokumentacja funkcji składowych	98
5.33.3.1	<code>draw_shape()</code>	98
5.34	Dokumentacja klasy <code>window</code>	99
5.34.1	Opis szczegółowy	101
5.34.2	Dokumentacja konstruktora i destruktora	101
5.34.2.1	<code>window()</code>	101
5.34.2.2	<code>~window()</code>	101
5.34.3	Dokumentacja funkcji składowych	101
5.34.3.1	<code>clicked_on_menu()</code>	101
5.34.3.2	<code>file_drop()</code>	102
5.34.3.3	<code>get_menu_height()</code>	102

5.34.3.4	init_sld()	102
5.34.3.5	is_the_tool_a_brush()	102
5.34.3.6	pressed_left_button_second_time_tool_flag()	103
5.34.3.7	pressed_left_button_tool_flag()	103
5.34.3.8	print_sdl_error()	103
5.34.3.9	redraw_window()	104
5.34.3.10	render_window()	104
5.34.3.11	reset_canva()	104
5.34.3.12	resize_brush()	104
5.34.3.13	resized_window() [1/2]	104
5.34.3.14	resized_window() [2/2]	105
5.34.3.15	set_copied_pixels()	105
5.34.3.16	swap_current_and_undo_pixels()	105
5.34.3.17	trigger_file_manager_menu()	106
5.34.3.18	trigger_filter_menu()	106
5.34.3.19	trigger_tool_menu()	106
5.34.3.20	use_tool()	106
5.34.4	Dokumentacja atrybutów składowych	106
5.34.4.1	copied_pixels	107
5.34.4.2	current_tool	107
5.34.4.3	file_manager_widget	107
5.34.4.4	filter_menu_widget	107
5.34.4.5	render	107
5.34.4.6	renderer	107
5.34.4.7	sdl_window	107
5.34.4.8	texture	107
5.34.4.9	tool_menu_widget	108
5.34.4.10	window_menu	108
5.34.4.11	window_name	108

6	Dokumentacja plików	109
6.1	Dokumentacja pliku <code>src/base_window.hpp</code>	109
6.1.1	Opis szczegółowy	109
6.2	Dokumentacja pliku <code>src/file_manager/file_manager.cpp</code>	109
6.2.1	Opis szczegółowy	109
6.3	Dokumentacja pliku <code>src/file_manager/file_manager.hpp</code>	110
6.3.1	Opis szczegółowy	110
6.3.2	Dokumentacja typów wyliczanych	110
6.3.2.1	<code>file_manager_mode</code>	110
6.4	Dokumentacja pliku <code>src/file_manager/image_container/bmp_image_container.cpp</code>	111
6.4.1	Opis szczegółowy	111
6.5	Dokumentacja pliku <code>src/file_manager/image_container/bmp_image_container.hpp</code>	111
6.5.1	Opis szczegółowy	111
6.6	Dokumentacja pliku <code>src/file_manager/image_container/image_container.cpp</code>	111
6.6.1	Opis szczegółowy	111
6.7	Dokumentacja pliku <code>src/file_manager/image_container/image_container.hpp</code>	112
6.7.1	Opis szczegółowy	112
6.8	Dokumentacja pliku <code>src/file_manager/image_container/png_image_container.cpp</code>	112
6.8.1	Opis szczegółowy	112
6.9	Dokumentacja pliku <code>src/file_manager/image_container/png_image_container.hpp</code>	112
6.9.1	Opis szczegółowy	113
6.10	Dokumentacja pliku <code>src/file_manager/image_container/ppm_image_container.cpp</code>	113
6.10.1	Opis szczegółowy	113
6.11	Dokumentacja pliku <code>src/file_manager/image_container/ppm_image_container.hpp</code>	113
6.11.1	Opis szczegółowy	113
6.12	Dokumentacja pliku <code>src/filters/basic_filter.cpp</code>	113
6.12.1	Opis szczegółowy	113
6.13	Dokumentacja pliku <code>src/filters/basic_filter.hpp</code>	114
6.13.1	Opis szczegółowy	114
6.13.2	Dokumentacja typów wyliczanych	114

6.13.2.1	filter_type	114
6.14	Dokumentacja pliku src/filters/color_inverse.cpp	115
6.14.1	Opis szczegółowy	115
6.15	Dokumentacja pliku src/filters/color_inverse.hpp	115
6.15.1	Opis szczegółowy	115
6.16	Dokumentacja pliku src/filters/filter_menu.cpp	115
6.16.1	Opis szczegółowy	115
6.17	Dokumentacja pliku src/filters/filter_menu.hpp	115
6.17.1	Opis szczegółowy	116
6.18	Dokumentacja pliku src/filters/grayscale.cpp	116
6.18.1	Opis szczegółowy	116
6.19	Dokumentacja pliku src/filters/grayscale.hpp	116
6.19.1	Opis szczegółowy	116
6.20	Dokumentacja pliku src/filters/pixelate.cpp	116
6.20.1	Opis szczegółowy	117
6.21	Dokumentacja pliku src/filters/pixelate.hpp	117
6.21.1	Opis szczegółowy	117
6.22	Dokumentacja pliku src/filters/sepia.cpp	117
6.22.1	Opis szczegółowy	117
6.23	Dokumentacja pliku src/filters/sepia.hpp	117
6.23.1	Opis szczegółowy	118
6.24	Dokumentacja pliku src/gtk_base_window.hpp	118
6.24.1	Opis szczegółowy	118
6.25	Dokumentacja pliku src/main.cpp	118
6.25.1	Opis szczegółowy	118
6.25.2	Dokumentacja funkcji	119
6.25.2.1	main()	119
6.26	Dokumentacja pliku src/paint_exception.cpp	119
6.26.1	Opis szczegółowy	119
6.27	Dokumentacja pliku src/paint_exception.hpp	119

6.27.1	Opis szczegółowy	119
6.28	Dokumentacja pliku src/tools/brushes/brush.cpp	119
6.28.1	Opis szczegółowy	120
6.29	Dokumentacja pliku src/tools/brushes/brush.hpp	120
6.29.1	Opis szczegółowy	120
6.30	Dokumentacja pliku src/tools/brushes/spray.cpp	120
6.30.1	Opis szczegółowy	120
6.31	Dokumentacja pliku src/tools/brushes/spray.hpp	120
6.31.1	Opis szczegółowy	121
6.32	Dokumentacja pliku src/tools/brushes/star_brush.cpp	121
6.32.1	Opis szczegółowy	121
6.33	Dokumentacja pliku src/tools/brushes/star_brush.hpp	121
6.33.1	Opis szczegółowy	121
6.34	Dokumentacja pliku src/tools/color_picker.cpp	121
6.34.1	Opis szczegółowy	122
6.35	Dokumentacja pliku src/tools/color_picker.hpp	122
6.35.1	Opis szczegółowy	122
6.36	Dokumentacja pliku src/tools/flood_fill.cpp	122
6.36.1	Opis szczegółowy	122
6.37	Dokumentacja pliku src/tools/flood_fill.hpp	122
6.37.1	Opis szczegółowy	123
6.38	Dokumentacja pliku src/tools/shapes/ellipse.cpp	123
6.38.1	Opis szczegółowy	123
6.39	Dokumentacja pliku src/tools/shapes/ellipse.hpp	123
6.39.1	Opis szczegółowy	123
6.40	Dokumentacja pliku src/tools/shapes/line.cpp	123
6.40.1	Opis szczegółowy	124
6.41	Dokumentacja pliku src/tools/shapes/line.hpp	124
6.41.1	Opis szczegółowy	124
6.42	Dokumentacja pliku src/tools/shapes/rectangle.cpp	124

6.42.1	Opis szczegółowy	124
6.43	Dokumentacja pliku <code>src/tools/shapes/rectangle.hpp</code>	124
6.43.1	Opis szczegółowy	125
6.44	Dokumentacja pliku <code>src/tools/shapes/shape.cpp</code>	125
6.44.1	Opis szczegółowy	125
6.45	Dokumentacja pliku <code>src/tools/shapes/shape.hpp</code>	125
6.45.1	Opis szczegółowy	125
6.46	Dokumentacja pliku <code>src/tools/shapes/triangle.cpp</code>	125
6.46.1	Opis szczegółowy	125
6.47	Dokumentacja pliku <code>src/tools/shapes/triangle.hpp</code>	126
6.47.1	Opis szczegółowy	126
6.48	Dokumentacja pliku <code>src/tools/tool.cpp</code>	126
6.48.1	Opis szczegółowy	126
6.49	Dokumentacja pliku <code>src/tools/tool.hpp</code>	126
6.49.1	Opis szczegółowy	127
6.49.2	Dokumentacja typów wyliczanych	127
6.49.2.1	<code>tool_type</code>	127
6.50	Dokumentacja pliku <code>src/tools/tool_menu.cpp</code>	127
6.50.1	Opis szczegółowy	127
6.51	Dokumentacja pliku <code>src/tools/tool_menu.hpp</code>	128
6.51.1	Opis szczegółowy	128
6.52	Dokumentacja pliku <code>src/window/menu.cpp</code>	128
6.52.1	Opis szczegółowy	128
6.53	Dokumentacja pliku <code>src/window/menu.hpp</code>	128
6.53.1	Opis szczegółowy	129
6.53.2	Dokumentacja typów wyliczanych	129
6.53.2.1	<code>menu_buttons</code>	129
6.54	Dokumentacja pliku <code>src/window/window.cpp</code>	129
6.54.1	Opis szczegółowy	130
6.55	Dokumentacja pliku <code>src/window/window.hpp</code>	130
6.55.1	Opis szczegółowy	130

Rozdział 1

Lista błędów

Plik `color_picker.cpp` (str. 121)

Użycie color pickera jest traktowane jako operacja "możliwa" do cofnięcia. Nie powoduje to większych problemów, poza tym, że nie da się wycofać poprzedzającej użycie narzędzia operacji.

Składowa `triangle::draw_shape` (str. 98) (`std::size_t x_position`, `std::size_t y_position`)

Trójkąt "traci" równoboczność przy prawej krawędzi okna.

Błąd przy rysowaniu koła kiedy `x_position ≈ x_begining`.

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

base_window	11
basic_filter	13
color_inverse	25
grayscale	44
pixelate	66
sepia	76
file_manager	31
menu	54
tool	84
brush	22
spray	80
star_brush	83
color_picker	26
flood_fill	40
shape	77
ellipse	28
line	52
rectangle	73
triangle	97
window	99
image_container::basic_image_info	16
bmp_image_container::bmp_file_info	17
image_container::color	24
filter_type_data	39
gtk_base_window	45
file_manager	31
filter_menu	37
tool_menu	91
image_container	48
bmp_image_container	20
png_image_container	68
ppm_image_container	71
paint_exception	61
ppm_image_container::ppm_file_info	70
tool_type_data	96

Rozdział 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

base_window	Klasa bazowa dla elementów okna programu oraz narzędzi	11
basic_filter	Klasa stanowiąca bazę dla klas realizujących filtry	13
image_container::basic_image_info	Struktura przechowująca podstawowe informacje o przechowywanym obrazie	16
bmp_image_container::bmp_file_info	Struktura pomocnicza przechowująca informacje o obrazach bmp	17
bmp_image_container	Kontener dla plików bmp	20
brush	Klasa narzędzia pędzel	22
image_container::color	Struktura przechowująca informacje o składowych koloru	24
color_inverse	Klasa filtra odwracającego kolory obrazu	25
color_picker	Definicja klasy narzędzia pobierania koloru z obszaru rysowania	26
ellipse	Klasa służąca do rysowania elipsy	28
file_manager	Klasa mangera plików	31
filter_menu	Klasa menu do wyboru filtrów	37
filter_type_data	Struktura przechowująca dane o wybranym filtrze. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych	39
flood_fill	Klasa służącą do wypełniania obszaru kolorem	40
grayscale	Klasa filtra zmieniająca obraz na czarno-biały	44
gtk_base_window	Klasa bazowa dla okien używających biblioteki GTK	45
image_container	Klasa bazowa dla wszystkich typów kontenerów obrazu	48

line		
	Klasa służąca do rysowania prostej linii	52
menu		
	Klasa służąca do rysowania menu oraz obsługi kliknięć w jego obrębie	54
paint_exception		
	Klasa do obsługi wyjątków programu	61
pixelate		
	Klasa filtra pikselizującego obraz	66
png_image_container		
	Kontener dla plików png	68
ppm_image_container::ppm_file_info		
	Struktura pomocnicza przechowująca informacje o obrazach ppm	70
ppm_image_container		
	Kontener dla plików ppm	71
rectangle		
	Klasa rysująca prostokąt	73
sepia		
	Klasa filtra sepia	76
shape		
	Klasa bazowa dla klas rysujących figury geometryczne	77
spray		
	Klasa narzędzia sprej	80
star_brush		
	Klasa narzędzia pędzla w kształcie plusa	83
tool		
	Klasa bazowa dla wszystkich typów narzędzi	84
tool_menu		
	Klasa związana z wyświetlaniem menu wyboru narzędzi	91
tool_type_data		
	Struktura przechowująca dane o wybranym narzędziu. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych	96
triangle		
	Klasa rysująca trójkąt równoboczny	97
window		
	Klasa służąca do obsługi okna głównego programu	99

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

src/ base_window.hpp	
Plik z definicją klasy base_window (str. 11)	109
src/ gtk_base_window.hpp	
Plik definicję klasy gtk_base_window (str. 45)	118
src/ main.cpp	
Plik zawiera funkcję main	118
src/ paint_exception.cpp	
Plik z definicjami metod klasy paint_exception (str. 61)	119
src/ paint_exception.hpp	
Plik z definicją klasy paint_exception (str. 61)	119
src/file_manager/ file_manager.cpp	
Plik z definicjami metod klasy file_manager (str. 31)	109
src/file_manager/ file_manager.hpp	
Plik definicję klasy file_manager (str. 31) oraz typu wyliczeniowego file_manager_mode	110
src/file_manager/image_container/ bmp_image_container.cpp	
Plik z definicjami metod klasy bmp_image_container (str. 20)	111
src/file_manager/image_container/ bmp_image_container.hpp	
Plik z definicją klasy bmp_image_container (str. 20)	111
src/file_manager/image_container/ image_container.cpp	
Plik z definicjami metod klasy image_container (str. 48)	111
src/file_manager/image_container/ image_container.hpp	
Plik z definicją klasy image_container (str. 48) oraz typu wyliczeniowego file_type	112
src/file_manager/image_container/ png_image_container.cpp	
Plik definicję metod klasy png_image_container (str. 68)	112
src/file_manager/image_container/ png_image_container.hpp	
Plik definicję klasy ppm_image_container (str. 71)	112
src/file_manager/image_container/ ppm_image_container.cpp	
Plik z definicjami metod klasy ppm_image_container (str. 71)	113
src/file_manager/image_container/ ppm_image_container.hpp	
Plik definicję klasy ppm_image_container (str. 71)	113
src/filters/ basic_filter.cpp	
Plik z definicjami metod klasy basic_filter (str. 13)	113
src/filters/ basic_filter.hpp	
Plik z definicją klasy base_filter oraz typu wyliczeniowego filter_type	114
src/filters/ color_inverse.cpp	
Plik z definicjami metod klasy color_inverse (str. 25)	115

src/filters/ color_inverse.hpp	
Plik z definicją klasy color_inverse (str. 25)	115
src/filters/ filter_menu.cpp	
Plik z definicjami metod klasy filter_menu (str. 37)	115
src/filters/ filter_menu.hpp	
Plik definicję klasy filter_menu (str. 37)	115
src/filters/ grayscale.cpp	
Plik z definicjami metod klasy grayscale	116
src/filters/ grayscale.hpp	
Plik z definicją klasy grayscale	116
src/filters/ pixelate.cpp	
Plik z definicjami metod klasy pixelate	116
src/filters/ pixelate.hpp	
Plik z definicją klasy pixelate	117
src/filters/ sepia.cpp	
Plik z definicjami metod klasy sepia	117
src/filters/ sepia.hpp	
Plik z definicją klasy sepia	117
src/tools/ color_picker.cpp	
Plik z definicjami metod klasy color_picker (str. 26)	121
src/tools/ color_picker.hpp	
Plik z definicją klasy color_picker (str. 26)	122
src/tools/ flood_fill.cpp	
Plik z definicjami metod klasy flood_fill (str. 40)	122
src/tools/ flood_fill.hpp	
Plik z definicją klasy flood_fill (str. 40)	122
src/tools/ tool.cpp	
Plik z definicjami metod klasy tool	126
src/tools/ tool.hpp	
Plik z definicją klasy tool oraz typu wyliczeniowego tool_type	126
src/tools/ tool_menu.cpp	
Plik z definicjami metod klasy tool_menu (str. 91)	127
src/tools/ tool_menu.hpp	
Plik z definicją klasy tool_menu (str. 91)	128
src/tools/brushes/ brush.cpp	
Plik z definicjami metod klasy brush	119
src/tools/brushes/ brush.hpp	
Plik z definicją klasy brush	120
src/tools/brushes/ spray.cpp	
Plik z definicjami metod klasy spray	120
src/tools/brushes/ spray.hpp	
Plik z definicją klasy spray	120
src/tools/brushes/ star_brush.cpp	
Plik z definicjami metod klasy star_brush (str. 83)	121
src/tools/brushes/ star_brush.hpp	
Plik z definicją klasy star_brush (str. 83)	121
src/tools/shapes/ ellipse.cpp	
Plik z definicjami metod klasy ellipse	123
src/tools/shapes/ ellipse.hpp	
Plik z definicją klasy ellipse	123
src/tools/shapes/ line.cpp	
Plik z definicjami metod klasy line	123
src/tools/shapes/ line.hpp	
Plik z definicją klasy line	124
src/tools/shapes/ rectangle.cpp	
Plik z definicjami metod klasy rectangle	124
src/tools/shapes/ rectangle.hpp	
Plik z definicją klasy rectangle	124

src/tools/shapes/ shape.cpp	
Plik z definicjami metod klasy shape	125
src/tools/shapes/ shape.hpp	
Plik z definicją klasy shape	125
src/tools/shapes/ triangle.cpp	
Plik z definicją klasy triangle	125
src/tools/shapes/ triangle.hpp	
Plik z definicją metod klasy triangle	126
src/window/ menu.cpp	
Plik z definicjami metod klasy menu	128
src/window/ menu.hpp	
Plik z definicją klasy menu oraz typu wyliczeniowego menu_buttons	128
src/window/ window.cpp	
Plik z definicjami metod klasy window	129
src/window/ window.hpp	
Plik z definicją klasy window	130

Rozdział 5

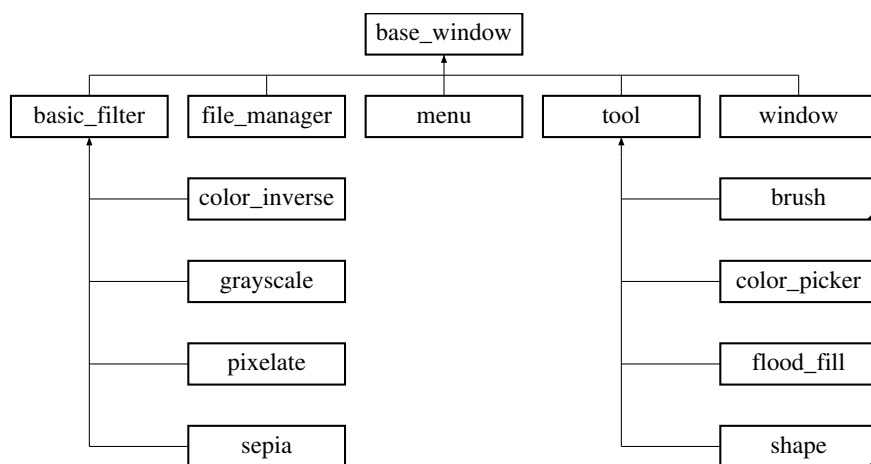
Dokumentacja klas

5.1 Dokumentacja klasy base_window

Klasa bazowa dla elementów okna programu oraz narzędzi.

```
#include <base_window.hpp>
```

Diagram dziedziczenia dla base_window



Metody publiczne

- virtual ~**base_window** ()
Destruktor klasy.

Atrybuty chronione

- std::size_t **window_width**
- std::size_t **window_height**
- std::size_t **menu_height**
- Uint32 * **window_area**

5.1.1 Opis szczegółowy

Klasa bazowa dla elementów okna programu oraz narzędzi.

5.1.2 Dokumentacja konstruktora i destruktora

5.1.2.1 ~base_window()

```
virtual base_window::~~base_window ( ) [inline], [virtual]
```

Destruktor klasy.

5.1.3 Dokumentacja atrybutów składowych

5.1.3.1 menu_height

```
std::size_t base_window::menu_height [protected]
```

Wysokość menu okna programu.

5.1.3.2 window_area

```
Uint32* base_window::window_area [protected]
```

Tablica z pikselami, na jej podstawie jest renderowane okno programu

5.1.3.3 window_height

```
std::size_t base_window::window_height [protected]
```

Wysokość okna programu.

5.1.3.4 window_width

```
std::size_t base_window::window_width [protected]
```

Szerokość okna programu.

Dokumentacja dla tej klasy została wygenerowana z pliku:

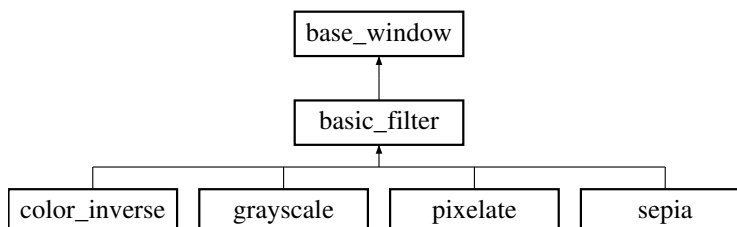
- src/ **base_window.hpp**

5.2 Dokumentacja klasy `basic_filter`

Klasa stanowiąca bazę dla klas realizujących filtry.

```
#include <basic_filter.hpp>
```

Diagram dziedziczenia dla `basic_filter`



Metody publiczne

- void **update_filter_information** (UInt32 * **window**, std::size_t **window_width**, std::size_t **window_height**, std::size_t **menu_height**)
Metoda przekazuje do obiektu filtra informacje na temat okna programu. Metoda nie zwraca żadnej wartości.
- virtual void **open_fiter_properties** ()=0
Metoda otwiera okno GTK, które pozwala ustawić parametry filtra. Metoda nie zwraca żadnej wartości.
- virtual void **do_filter** ()=0
Metoda wykonująca filter. Nie zwraca żadnej wartości.

Statyczne metody chronione

- static gboolean **delete_event** (GtkWidget *widget, GdkEvent *event, gpointer data)
Funkcja obsługująca sygnał usunięcia okna GTK.
- static void **destroy** (GtkWidget *widget, gpointer data)
Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Atrybuty chronione

- enum **filter_type** type

5.2.1 Opis szczegółowy

Klasa stanowiąca bazę dla klas realizujących filtry.

5.2.2 Dokumentacja funkcji składowych

5.2.2.1 `delete_event()`

```
static gboolean basic_filter::delete_event (
    GtkWidget * widget,
    GdkEvent * event,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał usunięcia okna GTK.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>event</i>	GdkEvent* Zdarzenie.
<i>data</i>	gpointer Dane dla funkcji.

Zwraca

gboolean Funkcja zwraca wartość TRUE.

5.2.2.2 destroy()

```
static void basic_filter::destroy (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>data</i>	gpointer Dane dla funkcji.

5.2.2.3 do_filter()

```
virtual void basic_filter::do_filter ( ) [pure virtual]
```

Metoda wykonująca filter. Nie zwraca żadnej wartości.

Implementowany w **grayscale** (str. 45), **pixelate** (str. 67), **sepia** (str. 76) i **color_inverse** (str. 25).

5.2.2.4 open_fiter_properties()

```
virtual void basic_filter::open_fiter_properties ( ) [pure virtual]
```

Metoda otwiera okno GTK, które pozwala ustawić parametry filtra. Metoda nie zwraca żadnej wartości.

Implementowany w **grayscale** (str. 45), **pixelate** (str. 67), **sepia** (str. 77) i **color_inverse** (str. 26).

5.2.2.5 `update_filter_information()`

```
void basic_filter::update_filter_information (
    Uint32 * window,
    std::size_t window_width,
    std::size_t window_height,
    std::size_t menu_height )
```

Metoda przekazuje do obiektu filtra informacje na temat okna programu. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>window</i>	Uint32* Wskaźnik na tablicę pikseli okna programu.
in	<i>window_width</i>	std::size_t Szerokość okna.
in	<i>window_height</i>	std::size_t Wysokość okna.
in	<i>menu_height</i>	std::size_t Wysokość menu okna.

5.2.3 Dokumentacja atrybutów składowych

5.2.3.1 type

```
enum filter_type basic_filter::type [protected]
```

Rodzaj filtra.

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/filters/ **basic_filter.hpp**
- src/filters/ **basic_filter.cpp**

5.3 Dokumentacja struktury image_container::basic_image_info

Struktura przechowująca podstawowe informacje o przechowywanym obrazie.

```
#include <image_container.hpp>
```

Atrybuty publiczne

- std::uint32_t **width**
- std::uint32_t **height**
- std::uint32_t **number_of_colors**
- enum **file_type** **image_type**

5.3.1 Opis szczegółowy

Struktura przechowująca podstawowe informacje o przechowywanym obrazie.

5.3.2 Dokumentacja atrybutów składowych

5.3.2.1 height

```
std::uint32_t image_container::basic_image_info::height
```

Wysokość obrazu.

5.3.2.2 image_type

```
enum file_type image_container::basic_image_info::image_type
```

5.3.2.3 number_of_colors

```
std::uint32_t image_container::basic_image_info::number_of_colors
```

Liczba części składowych koloru.

5.3.2.4 width

```
std::uint32_t image_container::basic_image_info::width
```

Szerokość obrazu

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `src/file_manager/image_container/ image_container.hpp`

5.4 Dokumentacja struktury `bmp_image_container::bmp_file_info`

Struktura pomocnicza przechowująca informacje o obrazach bmp.

Atrybuty publiczne

- `std::uint32_t header_size`
- `std::uint32_t width`
- `std::uint32_t height`
- `std::uint32_t size`
- `std::uint32_t compression`
- `std::uint32_t image_size`
- `std::uint32_t horizontal_resolution`
- `std::uint32_t vertical_resolution`
- `std::uint32_t number_of_color_in_palette`
- `std::uint32_t number_of_important_color`
- `std::uint16_t planes`
- `std::uint16_t bits_per_pixel`

5.4.1 Opis szczegółowy

Struktura pomocnicza przechowująca informacje o obrazach bmp.

5.4.2 Dokumentacja atrybutów składowych

5.4.2.1 bits_per_pixel

```
std::uint16_t bmp_image_container::bmp_file_info::bits_per_pixel
```

Liczba bitów na piksel.

5.4.2.2 compression

```
std::uint32_t bmp_image_container::bmp_file_info::compression
```

Znacznik rodzaju zastosowanej kompresji.

5.4.2.3 header_size

```
std::uint32_t bmp_image_container::bmp_file_info::header_size
```

Wielość nagłówka.

5.4.2.4 height

```
std::uint32_t bmp_image_container::bmp_file_info::height
```

Wysokość obrazu

5.4.2.5 horizontal_resolution

```
std::uint32_t bmp_image_container::bmp_file_info::horizontal_resolution
```

Rozdzielczość w pionie.

5.4.2.6 image_size

```
std::uint32_t bmp_image_container::bmp_file_info::image_size
```

Rozmiar obrazu.

5.4.2.7 `number_of_color_in_palette`

```
std::uint32_t bmp_image_container::bmp_file_info::number_of_color_in_palette
```

Liczba kolorów w palecie.

5.4.2.8 `number_of_important_color`

```
std::uint32_t bmp_image_container::bmp_file_info::number_of_important_color
```

Liczba znaczących kolorów w palecie.

5.4.2.9 `planes`

```
std::uint16_t bmp_image_container::bmp_file_info::planes
```

Płaszczyzny.

5.4.2.10 `size`

```
std::uint32_t bmp_image_container::bmp_file_info::size
```

Rozmiar pliku.

5.4.2.11 `vertical_resolution`

```
std::uint32_t bmp_image_container::bmp_file_info::vertical_resolution
```

Rozdzielczość w poziomie.

5.4.2.12 `width`

```
std::uint32_t bmp_image_container::bmp_file_info::width
```

Wysokość obrazu.

Dokumentacja dla tej struktury została wygenerowana z pliku:

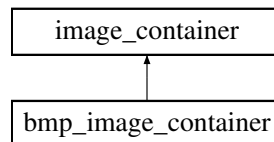
- `src/file_manager/image_container/ bmp_image_container.hpp`

5.5 Dokumentacja klasy `bmp_image_container`

Kontener dla plików bmp.

```
#include <bmp_image_container.hpp>
```

Diagram dziedziczenia dla `bmp_image_container`



Komponenty

- struct **`bmp_file_info`**
Struktura pomocnicza przechowująca informacje o obrazach bmp.

Metody publiczne

- **`bmp_image_container`** (`std::string file_path`)
Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.
- **`bmp_image_container`** (`std::size_t width`, `std::size_t height`, `std::size_t number_of_colors`)
Konstruktor. Inicjuje pusty kontener o podanych parametrach.
- void **`save_to_file`** (`std::string file_path`)
Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.
- **`~bmp_image_container`** ()
Destruktor.

Dodatkowe Dziedziczone Składowe

5.5.1 Opis szczegółowy

Kontener dla plików bmp.

5.5.2 Dokumentacja konstruktora i destruktora

5.5.2.1 `bmp_image_container()` [1/2]

```
bmp_image_container::bmp_image_container (  
    std::string file_path )
```

Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.

Parametry

in	<i>file_path</i>	std::string ścieżka do pliku zawierającego obraz.
----	------------------	---

5.5.2.2 `bmp_image_container()` [2/2]

```
bmp_image_container::bmp_image_container (
    std::size_t width,
    std::size_t height,
    std::size_t number_of_colors )
```

Konstruktor. Inicjuje pusty kontener o podanych parametrach.

Parametry

in	<i>width</i>	std::size_t szerokość obrazu w kontenerze.
in	<i>height</i>	std::size_t wysokość obrazu w kontenerze.
in	<i>number_of_colors</i>	std::size_t liczba składowych koloru w kontenerze (3 lub 4).

5.5.2.3 `~bmp_image_container()`

```
bmp_image_container::~~bmp_image_container ( )
```

Destruktor.

5.5.3 Dokumentacja funkcji składowych

5.5.3.1 `save_to_file()`

```
void bmp_image_container::save_to_file (
    std::string file_path ) [virtual]
```

Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.

Parametry

in	<i>file_path</i>	std::string Ścieżka do pliku, który ma być zapisany.
----	------------------	--

Implementuje **image_container** (str. 51).

Dokumentacja dla tej klasy została wygenerowana z plików:

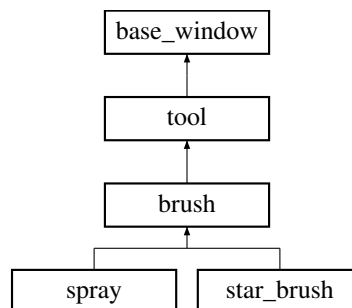
- src/file_manager/image_container/ **bmp_image_container.hpp**
- src/file_manager/image_container/ **bmp_image_container.cpp**

5.6 Dokumentacja klasy brush

Klasa narzędzia pędzel.

```
#include <brush.hpp>
```

Diagram dziedziczenia dla brush



Metody publiczne

- **brush ()**
Konstruktor klasy, służy do ustawienia pola type;.
- virtual void **use_tool** (std::size_t x_position, std::size_t y_position)
Metoda wykonuje akcję rysowania kwadratu (wywołanie metody paint_pixel) w podanym punkcie. Metoda nie zwraca żadnej wartości.
- bool **is_a_brush** ()
Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Dodatkowe Dziedziczone Składowe

5.6.1 Opis szczegółowy

Klasa narzędzia pędzel.

5.6.2 Dokumentacja konstruktora i destruktor

5.6.2.1 brush()

```
brush::brush ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.6.3 Dokumentacja funkcji składowych

5.6.3.1 is_a_brush()

```
bool brush::is_a_brush ( ) [virtual]
```

Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Zwraca

bool Metoda zwraca true w przypadku tej klasy.

Reimplementowana z **tool** (str. 88).

5.6.3.2 use_tool()

```
void brush::use_tool (
    std::size_t x_position,
    std::size_t y_position ) [virtual]
```

Metoda wykonuje akcję rysowania kwadratu (wywołanie metody paint_pixel) w podanym punkcie. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 90).

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/tools/brushes/ **brush.hpp**
- src/tools/brushes/ **brush.cpp**

5.7 Dokumentacja struktury `image_container::color`

Struktura przechowująca informacje o składowych koloru.

```
#include <image_container.hpp>
```

Atrybuty publiczne

- `std::uint8_t red`
- `std::uint8_t green`
- `std::uint8_t blue`
- `std::uint8_t alpha`

5.7.1 Opis szczegółowy

Struktura przechowująca informacje o składowych koloru.

5.7.2 Dokumentacja atrybutów składowych

5.7.2.1 alpha

```
std::uint8_t image_container::color::alpha
```

Kanał przeźroczystości.

5.7.2.2 blue

```
std::uint8_t image_container::color::blue
```

Składowa niebieska.

5.7.2.3 green

```
std::uint8_t image_container::color::green
```

Składowa zielona.

5.7.2.4 red

```
std::uint8_t image_container::color::red
```

Składowa czerwieni.

Dokumentacja dla tej struktury została wygenerowana z pliku:

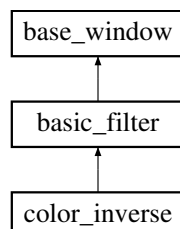
- `src/file_manager/image_container/ image_container.hpp`

5.8 Dokumentacja klasy color_inverse

Klasa filtra odwracającego kolory obrazu.

```
#include <color_inverse.hpp>
```

Diagram dziedziczenia dla color_inverse



Metody publiczne

- **color_inverse ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_fiter_properties ()**
Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.
- void **do_filter ()**
Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, odejmuje od nich liczbę 255 i zapisuje nowe wartości. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.8.1 Opis szczegółowy

Klasa filtra odwracającego kolory obrazu.

5.8.2 Dokumentacja konstruktora i destruktor

5.8.2.1 color_inverse()

```
color_inverse::color_inverse ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.8.3 Dokumentacja funkcji składowych

5.8.3.1 do_filter()

```
void color_inverse::do_filter ( ) [virtual]
```

Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, odejmuje od nich liczbę 255 i zapisuje nowe wartości. Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

5.8.3.2 open_fiter_properties()

```
void color_inverse::open_fiter_properties ( ) [virtual]
```

Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

Dokumentacja dla tej klasy została wygenerowana z plików:

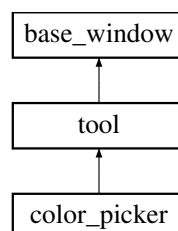
- src/filters/ **color_inverse.hpp**
- src/filters/ **color_inverse.cpp**

5.9 Dokumentacja klasy color_picker

Definicja klasy narzędzia pobierania koloru z obszaru rysowania.

```
#include <color_picker.hpp>
```

Diagram dziedziczenia dla color_picker



Metody publiczne

- **color_picker ()**
Konstruktor klasy, służy do ustawienia pola type;
- void **use_tool** (std::size_t x_position, std::size_t y_position)
Metoda zapisuje w polu current_color kolor wskazanego piksela. Metoda nie zwraca żadnej wartości.
- bool **isBrush** ()
Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Dodatkowe Dziedziczone Składowe

5.9.1 Opis szczegółowy

Definicja klasy narzędzia pobierania koloru z obszaru rysowania.

5.9.2 Dokumentacja konstruktora i destruktora

5.9.2.1 `color_picker()`

```
color_picker::color_picker ( )
```

Konstruktor klasy, służy do ustawienia pola `type`.

5.9.3 Dokumentacja funkcji składowych

5.9.3.1 `isBrush()`

```
bool color_picker::isBrush ( )
```

Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Zwraca

`bool` Metoda zwraca wartość `false` w przypadku tej klasy.

5.9.3.2 `use_tool()`

```
void color_picker::use_tool (
    std::size_t x_position,
    std::size_t y_position ) [virtual]
```

Metoda zapisuje w polu `current_color` kolor wskazanego piksela. Metoda nie zwraca żadnej wartości.

Parametry

in	<code>x_position</code>	<code>std::size_t</code> Składowa 'x' punktu.
in	<code>y_position</code>	<code>std::size_t</code> Składowa 'y' punktu.

Reimplementowana z **tool** (str. 90).

Dokumentacja dla tej klasy została wygenerowana z plików:

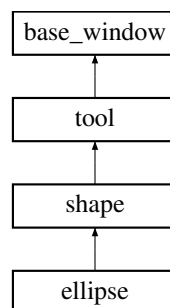
- src/tools/ **color_picker.hpp**
- src/tools/ **color_picker.cpp**

5.10 Dokumentacja klasy ellipse

Klasa służąca do rysowania elipsy.

```
#include <ellipse.hpp>
```

Diagram dziedziczenia dla ellipse



Metody publiczne

- **ellipse ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_tool_properties ()**
Metoda otwiera okno właściwości narzędzia. Pozwala ustawić flagę rysowania koła.

Metody prywatne

- void **PaintPixel** (std::size_t x_position, std::size_t y_position)
Przesłonięta metoda PaintPixel z klasy tool. Rysuje kształt plusa o rozmiarze pędzla. Metoda nie zwraca żadnej wartości.
- void **draw_shape** (std::size_t x_position, std::size_t y_position)
Metoda rysuje prostokąt od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Statyczne metody prywatne

- static void **set_cicle_flag** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał kliknięcia w przycisk oknie właściwości narzędzia. Ustawia flagę rysowania koła. Metoda nie zwraca żadnej wartości.

Atrybuty prywatne

- bool **is_circle** = false

Dodatkowe Dziedziczone Składowe

5.10.1 Opis szczegółowy

Klasa służąca do rysowania elipsy.

5.10.2 Dokumentacja konstruktora i destruktora

5.10.2.1 ellipse()

```
ellipse::ellipse ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.10.3 Dokumentacja funkcji składowych

5.10.3.1 draw_shape()

```
void ellipse::draw_shape (
    std::size_t x_position,
    std::size_t y_position ) [private], [virtual]
```

Metoda rysuje prostokąt od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **shape** (str. 78).

5.10.3.2 open_tool_properties()

```
void ellipse::open_tool_properties ( ) [virtual]
```

Metoda otwiera okno właściwości narzędzia. Pozwala ustawić flagę rysowania koła.

Reimplementowana z **tool** (str. 88).

5.10.3.3 PaintPixel()

```
void ellipse::PaintPixel (
    std::size_t x_position,
    std::size_t y_position ) [private], [virtual]
```

Przesłonięta metoda PaintPixel z klasy tool. Rysuje kształt plusa o rozmiarze pędzla. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 88).

5.10.3.4 set_cicle_flag()

```
static void ellipse::set_cicle_flag (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał kliknięcia w przycisk oknie właściwości narzędzia. Ustawia flagę rysowania koła. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno GTK, w którym generowany jest sygnał.
out	<i>data</i>	gpointer Dodatkowe dane, w tym przypadku wskaźnik na flagę typu bool.

5.10.4 Dokumentacja atrybutów składowych

5.10.4.1 is_circle

```
bool ellipse::is_circle = false [private]
```

Flaga rysowania koła.

Dokumentacja dla tej klasy została wygenerowana z plików:

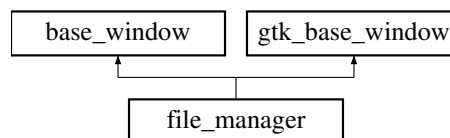
- src/tools/shapes/ **ellipse.hpp**
- src/tools/shapes/ **ellipse.cpp**

5.11 Dokumentacja klasy file_manager

Klasa mangera plików.

```
#include <file_manager.hpp>
```

Diagram dziedziczenia dla file_manager



Metody publiczne

- **file_manager ()**
*Konstruktor klasy. Odczytuje informacje z pliku zasobów. Może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu.*
- void **update_file_manager_information** (UInt32 * **window**, std::size_t **window_width**, std::size_t **window_height**, std::size_t **menu_height**)
Metoda przekazuje do klasy informacje na temat okna programu. Metoda nie zwraca żadnej wartości.
- enum **file_manager_mode** **get_file_manager_mode ()**
Getter aktualnego trybu pracy menadżera plików.
- std::string **get_file_path ()**
Getter ścieżki aktualnie obsługiwanego pliku.
- void **get_image_dimensions** (std::size_t &width, std::size_t &height)
Getter (poprzez wyjściowe argumenty wywołania) wymiarów aktualnie obsługiwanego obrazu. Metoda nie zwraca żadnej wartości.
- void **load_image ()**
*Metoda wczytuje obraz z pliku, przy pomocy klasy **image_container** (str. 48). Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.*
- void **load_image** (std::string **file_path**)
*Metoda wczytuje obraz z pliku (do którego ścieżka została podana jako argument wywołania), przy pomocy klasy **image_container** (str. 48). Wykorzystywana przy upuszczaniu plików na okno. Metoda nie zwraca żadnej wartości.*
- void **print_image ()**
Metoda rysuje na wczytany obraz w obszarze roboczym okna programu. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.
- void **save_window_to_image ()**
Metoda zapisuje obszar roboczy okna programu do pliku. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.
- void **reset_state ()**
Metoda resetuje stan menadżera plików.
- bool **open_gtk_window ()**
*Metoda zawiera inicjuje menu menadżera plików oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.*
- ~**file_manager ()**
Destruktor klasy.

Statyczne metody prywatne

- static void **open_save_file_widget** (GtkWidget *widget, gpointer data)
Metoda otwiera okno dialogowe GTK zapisu plików. Nie zwraca żadnej wartości.
- static void **open_open_file_widget** (GtkWidget *widget, gpointer data)
Metoda otwiera okno dialogowe GTK odczytu plików. Nie zwraca żadnej wartości.
- static void **quick_save_button_clicked** (GtkWidget *widget, gpointer data)
Metoda odpowiada za obsługę sygnału kliknięcia na przycisk szybkiego zapisu. Nie zwraca żadnej wartości.

Atrybuty prywatne

- **image_container** * **image** = nullptr
- std::string **file_path**
- bool **working_on_file** = false
- enum **file_manager_mode** **current_mode** = NONE
- **image_container::file_type** **current_file_type** = **image_container::UNSUPPORTED**

Dodatkowe Dziedziczone Składowe

5.11.1 Opis szczegółowy

Klasa mangera plików.

5.11.2 Dokumentacja konstruktora i destruktora

5.11.2.1 file_manager()

```
file_manager::file_manager ( )
```

Konstruktor klasy. Odczytuje informacje z pliku zasobów. Może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu.

5.11.2.2 ~file_manager()

```
file_manager::~file_manager ( )
```

Destruktor klasy.

5.11.3 Dokumentacja funkcji składowych

5.11.3.1 `get_file_manager_mode()`

```
enum file_manager_mode file_manager::get_file_manager_mode ( )
```

Getter aktualnego trybu pracy menadżera plików.

Zwraca

enum `file_manager_mode` Tryb pracy menadżera plików.

5.11.3.2 `get_file_path()`

```
std::string file_manager::get_file_path ( )
```

Getter ścieżki aktualnie obsługiwanego pliku.

Zwraca

std::string ścieżka do pliku.

5.11.3.3 `get_image_dimensions()`

```
void file_manager::get_image_dimensions (
    std::size_t & width,
    std::size_t & height )
```

Getter (poprzez wyjściowe argumenty wywołania) wymiarów aktualnie obsługiwanego obrazu. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>width</i>	std::size_t& Szerokość obrazu.
out	<i>height</i>	std::size_t& Wysokość obrazu.

5.11.3.4 `load_image()` [1/2]

```
void file_manager::load_image ( )
```

Metoda wczytuje obraz z pliku, przy pomocy klasy **image_container** (str. 48). Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.11.3.5 load_image() [2/2]

```
void file_manager::load_image (
    std::string file_path )
```

Metoda wczytuje obraz z pliku (do którego ścieżka została podana jako argument wywołania), przy pomocy klasy **image_container** (str. 48). Wykorzystywana przy upuszczaniu plików na okno. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>file_path</i>	std::string ścieżka do pliku
----	------------------	------------------------------

5.11.3.6 open_gtk_window()

```
bool file_manager::open_gtk_window ( ) [virtual]
```

Metoda zawiera inicjuje menu menadżera plików oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.

Zwraca

bool Funkcja zwraca wartość true jeśli nie wystąpił w niej krytyczny błąd.

Implementuje **gtk_base_window** (str. 47).

5.11.3.7 open_open_file_widget()

```
static void file_manager::open_open_file_widget (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda otwiera okno dialogowe GTK odczytu plików. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* okno GTK z którego zostało wywołane okno dialogowe.
out	<i>data</i>	gpointer dodatkowe dane. W tym przypadku wskaźnik na 2 elementową tablicę std::string.

5.11.3.8 open_save_file_widget()

```
static void file_manager::open_save_file_widget (
```

```
GtkWidget * widget,  
gpointer data ) [inline], [static], [private]
```

Metoda otwiera okno dialogowe GTK zapisu plików. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* okno GTK z którego zostało wywołane okno dialogowe.
out	<i>data</i>	gpointer dodatkowe dane. W tym przypadku wskaźnik na 2 elementową tablicę std::string.

5.11.3.9 `print_image()`

```
void file_manager::print_image ( )
```

Metoda rysuje na wczytany obraz w obszarze roboczym okna programu. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.11.3.10 `quick_save_button_clicked()`

```
static void file_manager::quick_save_button_clicked (  
    GtkWidget * widget,  
    gpointer data ) [inline], [static], [private]
```

Metoda odpowiada za obsługę sygnału kliknięcia na przycisk szybkiego zapisu. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* okno GTK z którego zostało wywołane okno dialogowe.
out	<i>data</i>	gpointer dodatkowe dane. W tym przypadku wskaźnik na 2 elementową tablicę std::string.

5.11.3.11 `reset_state()`

```
void file_manager::reset_state ( )
```

Metoda resetuje stan menedżera plików.

5.11.3.12 `save_window_to_image()`

```
void file_manager::save_window_to_image ( )
```

Metoda zapisuje obszar roboczy okna programu do pliku. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.11.3.13 update_file_manager_information()

```
void file_manager::update_file_manager_information (
    Uint32 * window,
    std::size_t window_width,
    std::size_t window_height,
    std::size_t menu_height )
```

Metoda przekazuje do klasy informacje na temat okna programu. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>window</i>	Uint32* Wskaźnik na tablicę pikseli okna programu.
in	<i>window_width</i>	std::size_t Szerokość okna.
in	<i>window_height</i>	std::size_t Wysokość okna.
in	<i>menu_height</i>	std::size_t Wysokość menu okna.

5.11.4 Dokumentacja atrybutów składowych

5.11.4.1 current_file_type

```
image_container::file_type file_manager::current_file_type = image_container::UNSUPPORTED
[private]
```

Typ obrazu, na którym są wykonywane operacje.

5.11.4.2 current_mode

```
enum file_manager_mode file_manager::current_mode = NONE [private]
```

Tryb pracy menadżera plików

5.11.4.3 file_path

```
std::string file_manager::file_path [private]
```

Ścieżka do pliku.

5.11.4.4 image

```
image_container* file_manager::image = nullptr [private]
```

Wskaźnik na kontener obrazu.

5.11.4.5 working_on_file

```
bool file_manager::working_on_file = false [private]
```

Flaga informująca o tym, że został otwarty/zapisany plik.

Dokumentacja dla tej klasy została wygenerowana z plików:

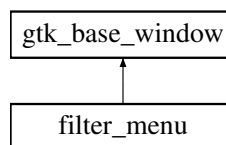
- src/file_manager/ **file_manager.hpp**
- src/file_manager/ **file_manager.cpp**

5.12 Dokumentacja klasy filter_menu

Klasa menu do wyboru filtrów.

```
#include <filter_menu.hpp>
```

Diagram dziedziczenia dla filter_menu



Metody publiczne

- **filter_menu ()**
*Konstruktor klasy. Odczytuje informacje z pliku zasobów. Może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu.*
- bool **open_gtk_window ()**
*Metoda zawiera inicjuje menu filtrów oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.*
- enum **filter_type** **get_filter_response ()**
Getter wybranego w menu rodzaju filtru.

Statyczne metody prywatne

- static void **get_filter_type** (GtkWidget *widget, gpointer data)
Funkcja obsługująca sygnał kliknięcia na przycisk z wyborem filtra. Funkcja nie zwraca żadnej wartości.

Atrybuty prywatne

- std::size_t **number_of_filters**
- std::deque< struct **filter_type_data** > **filter_types**
- enum **filter_type** **filter_response**

Dodatkowe Dziedziczone Składowe

5.12.1 Opis szczegółowy

Klasa menu do wyboru filtrów.

5.12.2 Dokumentacja konstruktora i destruktora

5.12.2.1 filter_menu()

```
filter_menu::filter_menu ( )
```

Konstruktor klasy. Odczytuje informacje z pliku zasobów. Może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu.

5.12.3 Dokumentacja funkcji składowych

5.12.3.1 get_filter_response()

```
enum filter_type filter_menu::get_filter_response ( )
```

Getter wybranego w menu rodzaju filtru.

Zwraca

enum filter_type wybrany rodzaj filtru.

5.12.3.2 get_filter_type()

```
static void filter_menu::get_filter_type (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Funkcja obsługująca sygnał kliknięcia na przycisk z wyborem filtra. Funkcja nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
out	<i>data</i>	gpointer Dane dla funkcji.

5.12.3.3 `open_gtk_window()`

```
bool filter_menu::open_gtk_window ( ) [virtual]
```

Metoda zawiera inicjuje menu filtrów oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.

Zwraca

bool Funkcja zwraca wartość true jeśli nie wystąpił w niej krytyczny błąd.

Implementuje **gtk_base_window** (str. 47).

5.12.4 Dokumentacja atrybutów składowych

5.12.4.1 `filter_response`

```
enum filter_type filter_menu::filter_response [private]
```

Filter wybrany przez użytkownika.

5.12.4.2 `filter_types`

```
std::deque<struct filter_type_data> filter_menu::filter_types [private]
```

Kolejka, która ułatwia działanie GTK.

5.12.4.3 `number_of_filters`

```
std::size_t filter_menu::number_of_filters [private]
```

Liczba filtrów programu.

Dokumentacja dla tej klasy została wygenerowana z plików:

- `src/filters/ filter_menu.hpp`
- `src/filters/ filter_menu.cpp`

5.13 Dokumentacja struktury `filter_type_data`

Struktura przechowująca dane o wybranym filtrze. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

```
#include <filter_menu.hpp>
```

Atrybuty publiczne

- enum **filter_type** type
- enum **filter_type** * response

5.13.1 Opis szczegółowy

Struktura przechowująca dane o wybranym filtrze. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

5.13.2 Dokumentacja atrybutów składowych

5.13.2.1 response

```
enum filter_type* filter_type_data::response
```

Wskaźnik na zmienną przechowującą wybór użytkownika.

5.13.2.2 type

```
enum filter_type filter_type_data::type
```

Typ narzędzia.

Dokumentacja dla tej struktury została wygenerowana z pliku:

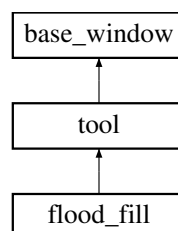
- src/filters/ **filter_menu.hpp**

5.14 Dokumentacja klasy flood_fill

Klasa służąca do wypełniania obszaru kolorem.

```
#include <flood_fill.hpp>
```

Diagram dziedziczenia dla flood_fill



Metody publiczne

- **flood_fill** ()
Konstruktor klasy, służy do ustawienia pola type;.
- void **use_tool** (std::size_t x_position, std::size_t y_position)
Przesłonięta metoda z klasy tool. Wypełnia obszar (z uwzględnieniem tolerancji) od podanego punktu. Metoda nie zwraca żadnej wartości.
- bool **isBrush** ()
Metoda zwraca czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.
- void **open_tool_properties** ()
Metoda otwiera okno GTK, które pozwala ustawić współczynnik tolerancji. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

Metody prywatne

- bool **calculate_tolerance** (Uint32 pointed_color, Uint32 replace_color)
Metoda oblicza na podstawie tolerancji czy kolor danego piksela powinien być zastąpiony nowym kolorem.

Statyczne metody prywatne

- static void **set_confirmed_flag** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości narzędzia. Metoda nie zwraca żadnej wartości.
- static void **value_changed** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Atrybuty prywatne

- int **tolerance_ratio** = 0

Dodatkowe Dziedziczone Składowe

5.14.1 Opis szczegółowy

Klasa służącą do wypełniania obszaru kolorem.

5.14.2 Dokumentacja konstruktora i destruktora

5.14.2.1 flood_fill()

```
flood_fill::flood_fill ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.14.3 Dokumentacja funkcji składowych

5.14.3.1 calculate_tolerance()

```
bool flood_fill::calculate_tolerance (
    Uint32 pointed_color,
    Uint32 replace_color ) [private]
```

Metoda oblicza na podstawie tolerancji czy kolor danego piksela powinien być zastąpiony nowym kolorem.

Parametry

in	<i>pointed_color</i>	Uint32 Kolor piksela, który został wskazany podczas rozpoczynania wypełnienia.
in	<i>replace_color</i>	Uint32 Kolor piksela, dla którego ma być obliczone czy jego kolor ma zostać zastąpiony nowym.

Zwraca

bool Jeśli różnica kolorów między pikselami będzie mniejsza niż wynikająca ze współczynnika tolerancji, metoda zwróci wartość false, w przeciwnym wypadku zwróci wartość true.

5.14.3.2 isBrush()

```
bool flood_fill::isBrush ( )
```

Metoda zwraca czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Zwraca

bool W przypadku tego narzędzia metoda zwraca wartość false.

5.14.3.3 open_tool_properties()

```
void flood_fill::open_tool_properties ( ) [virtual]
```

Metoda otwiera okno GTK, które pozwala ustawić współczynnik tolerancji. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

Reimplementowana z **tool** (str. 88).

5.14.3.4 set_confirmed_flag()

```
static void flood_fill::set_confirmed_flag (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości narzędzia. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno GTK, w którym generowany jest sygnał.
out	<i>data</i>	gpointer Dodatkowe dane, w tym przypadku wskaźnik na flagę typu bool.

5.14.3.5 use_tool()

```
void flood_fill::use_tool (
    std::size_t x_position,
    std::size_t y_position ) [virtual]
```

Przesłonięta metoda z klasy tool. Wypełnia obszar (z uwzględnieniem tolerancji) od podanego punktu. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 90).

5.14.3.6 value_changed()

```
static void flood_fill::value_changed (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno, w którym wygenerowany został sygnał.
out	<i>data</i>	gpointer Dodatkowe dane. W tym przypadku liczba naturalna.

5.14.4 Dokumentacja atrybutów składowych

5.14.4.1 tolerance_ratio

```
int flood_fill::tolerance_ratio = 0 [private]
```

Współczynnik tolerancji (w %)

Dokumentacja dla tej klasy została wygenerowana z plików:

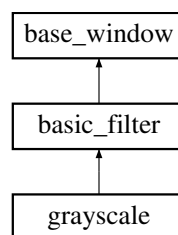
- src/tools/ **flood_fill.hpp**
- src/tools/ **flood_fill.cpp**

5.15 Dokumentacja klasy grayscale

Klasa filtra zmieniająca obraz na czarno-biały.

```
#include <grayscale.hpp>
```

Diagram dziedziczenia dla grayscale



Metody publiczne

- **grayscale ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_fiter_properties ()**
Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.
- void **do_filter ()**
Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, oblicza średnią tych wartości i zapisuje tę wartość w każdej składowej. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.15.1 Opis szczegółowy

Klasa filtra zmieniająca obraz na czarno-biały.

5.15.2 Dokumentacja konstruktora i destruktor

5.15.2.1 grayscale()

```
grayscale::grayscale ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.15.3 Dokumentacja funkcji składowych

5.15.3.1 do_filter()

```
void grayscale::do_filter ( ) [virtual]
```

Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, oblicza średnią tych wartości i zapisuje tę wartość w każdej składowej. Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

5.15.3.2 open_fiter_properties()

```
void grayscale::open_fiter_properties ( ) [virtual]
```

Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

Dokumentacja dla tej klasy została wygenerowana z plików:

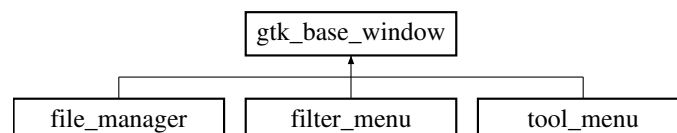
- src/filters/ **grayscale.hpp**
- src/filters/ **grayscale.cpp**

5.16 Dokumentacja klasy gtk_base_window

Klasa bazowa dla okien używających biblioteki GTK.

```
#include <gtk_base_window.hpp>
```

Diagram dziedziczenia dla gtk_base_window



Metody publiczne

- virtual bool **open_gtk_window** ()=0
Metoda zawiera inicjuje menu oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania.
- virtual **~gtk_base_window** ()
Destruktor klasy.

Statyczne metody chronione

- static gboolean **delete_event** (GtkWidget *widget, GdkEvent *event, gpointer data)

Funkcja obsługująca sygnał usunięcia okna GTK.

- static void **destroy** (GtkWidget *widget, gpointer data)

Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Atrybuty chronione

- std::string **resource_file_path** [3]
- std::deque< std::string > **resource_strings**
- bool **failed_to_load_resources**

5.16.1 Opis szczegółowy

Klasa bazowa dla okien używających biblioteki GTK.

5.16.2 Dokumentacja konstruktora i destruktora

5.16.2.1 ~gtk_base_window()

```
virtual gtk_base_window::~~gtk_base_window ( ) [inline], [virtual]
```

Destruktor klasy.

5.16.3 Dokumentacja funkcji składowych

5.16.3.1 delete_event()

```
static gboolean gtk_base_window::delete_event (
    GtkWidget * widget,
    GdkEvent * event,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał usunięcia okna GTK.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>event</i>	GdkEvent* Zdarzenie.
<i>data</i>	gpointer Dane dla funkcji.

Zwraca

gboolean Funkcja zwraca wartość TRUE.

5.16.3.2 destroy()

```
static void gtk_base_window::destroy (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>data</i>	gpointer Dane dla funkcji.

5.16.3.3 open_gtk_window()

```
virtual bool gtk_base_window::open_gtk_window ( ) [pure virtual]
```

Metoda zawiera inicjuje menu oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania.

Zwraca

bool Funkcja zwraca wartość true jeśli nie wystąpił w niej krytyczny błąd.

Implementowany w **file_manager** (str. 34), **tool_menu** (str. 94) i **filter_menu** (str. 39).

5.16.4 Dokumentacja atrybutów składowych**5.16.4.1 failed_to_load_resources**

```
bool gtk_base_window::failed_to_load_resources [protected]
```

Flaga informująca o błędzie załadowania zasobów okna menu.

5.16.4.2 resource_file_path

```
std::string gtk_base_window::resource_file_path[3] [protected]
```

Wartość początkowa:

```
= {
    "assets/resources/tool_menu.rc",
    "assets/resources/file_manager.rc",
    "assets/resources/filter_menu.rc"
}
```

Tablica zawiera ścieżki do plików z zasobami różnych menu.

5.16.4.3 resource_strings

```
std::deque<std::string> gtk_base_window::resource_strings [protected]
```

Kolejka z ciągami znaków wykorzystywanych w menu.

Dokumentacja dla tej klasy została wygenerowana z pliku:

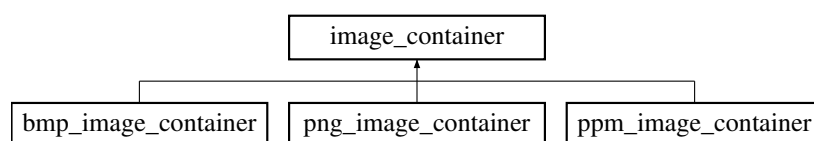
- src/ **gtk_base_window.hpp**

5.17 Dokumentacja klasy image_container

Klasa bazowa dla wszystkich typów kontenerów obrazu.

```
#include <image_container.hpp>
```

Diagram dziedziczenia dla image_container



Komponenty

- struct **basic_image_info**
Struktura przechowująca podstawowe informacje o przechowywanym obrazie.
- struct **color**
Struktura przechowująca informacje o składowych koloru.

Typy publiczne

- enum **file_type** { **PPM_FILE**, **BMP_FILE**, **PNG_FILE**, **UNSUPPORTED** }
Typ wyliczeniowy określający typ pliku.
- typedef std::uint32_t **ARGB_8888_color**

Metody publiczne

- `image_container ()`
Konstruktor klasy.
- `struct basic_image_info get_image_info ()`
Getter podstawowych informacji na temat przechowywanego obrazu.
- `void set_pixels_form_array (const ARGB_8888_color * image_pixel_array)`
Metoda wczytuje obraz z tablicy pikseli, która zawiera 4 składowe po 8 bitów każdy. Nie zwraca żadnej wartości.
- `virtual void save_to_file (std::string file_path)=0`
Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.
- `ARGB_8888_color operator[] (std::size_t n)`
Przeciążenie operatora [], zwraca kolor piksela w formacie `ARGB_8888`.
- `virtual ~image_container ()`
Destruktor. Usuwa zmienne zaalokowane dynamicznie.

Statyczne metody publiczne

- `static enum file_type get_file_type (std::string file_path)`
Metoda statyczna, która zwraca typ obrazu, do którego ścieżka została podana jako argument wywołania. Metoda "rzuca" wyjątek typu `paint_exception` (str. 61) gdy otwarcie lub czytanie pliku się nie powiedzie.

Metody chronione

- `std::string image_path ()`

Atrybuty chronione

- `struct basic_image_info image_info = {0, 0, 0, UNSUPPORTED}`
- `struct color * image_pixel_array = nullptr`

5.17.1 Opis szczegółowy

Klasa bazowa dla wszystkich typów kontenerów obrazu.

5.17.2 Dokumentacja składowych definicji typu

5.17.2.1 `ARGB_8888_color`

```
typedef std::uint32_t image_container::ARGB_8888_color
```

Definicja typu przechowującego 4 składowe koloru po 8 bitów każdy.

5.17.3 Dokumentacja składowych wyliczanych

5.17.3.1 `file_type`

```
enum image_container::file_type
```

Typ wyliczeniowy określający typ pliku.

Wartości wyliczeń

PPM_FILE	Plik ppm.
BMP_FILE	Plik bmp.
PNG_FILE	Plik jpg.
UNSUPPORTED	Typ nieobsługiwany przez kontener.

5.17.4 Dokumentacja konstruktora i destruktora

5.17.4.1 image_container()

```
image_container::image_container ( )
```

Konstruktor klasy.

5.17.4.2 ~image_container()

```
image_container::~~image_container ( ) [virtual]
```

Destruktor. Usuwa zmienne zaalokowane dynamicznie.

5.17.5 Dokumentacja funkcji składowych

5.17.5.1 get_file_type()

```
static enum file_type image_container::get_file_type (
    std::string file_path ) [inline], [static]
```

Metoda statyczna, która zwraca typ obrazu, do którego ścieżka została podana jako argument wywołania. Metoda "rzuca" wyjątek typu **paint_exception** (str. 61) gdy otwarcie lub czytanie pliku się nie powiedzie.

Parametry

in	<i>file_path</i>	std::string ścieżka do pliku.
----	------------------	-------------------------------

Zwraca

enum file_type Typ pliku lub wartość UNSUPPORTED.

5.17.5.2 `get_image_info()`

```
struct image_container::basic_image_info image_container::get_image_info ( )
```

Getter podstawowych informacji na temat przechowywanego obrazu.

Zwraca

struct **basic_image_info** (str. 16) struktura przechowująca informacje o obrazie.

5.17.5.3 `image_path()`

```
std::string image_container::image_path ( ) [protected]
```

Ścieżka do pliku, nad którym wykonywana jest operacja.

5.17.5.4 `operator[]()`

```
image_container::ARGB_8888_color image_container::operator[] (
    std::size_t n )
```

Przeciążenie operatora `[]`, zwraca kolor piksela w formacie ARGB_8888.

5.17.5.5 `save_to_file()`

```
virtual void image_container::save_to_file (
    std::string file_path ) [pure virtual]
```

Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.

Parametry

in	<i>file_path</i>	std::string Ścieżka do pliku, który ma być zapisany.
----	------------------	--

Implementowany w **png_image_container** (str. 70), **bmp_image_container** (str. 21) i **ppm_image_container** (str. 73).

5.17.5.6 `set_pixels_form_array()`

```
void image_container::set_pixels_form_array (
    const ARGB_8888_color * image_pixel_array )
```

Metoda wczytuje obraz z tablicy pikseli, która zawiera 4 składowe po 8 bitów każdy. Nie zwraca żadnej wartości.

Parametry

in	<i>image_pixel_array</i>	const ARGB_8888_color* tablica pikseli.
----	--------------------------	---

5.17.6 Dokumentacja atrybutów składowych

5.17.6.1 image_info

```
struct basic_image_info image_container::image_info = {0, 0, 0, UNSUPPORTED} [protected]
```

Informacje o otwartym obrazie.

5.17.6.2 image_pixel_array

```
struct color* image_container::image_pixel_array = nullptr [protected]
```

Wskaźnik na tablicę pikseli obrazu.

Dokumentacja dla tej klasy została wygenerowana z plików:

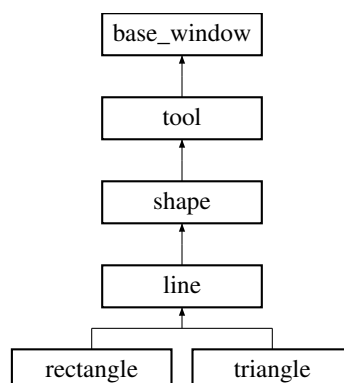
- src/file_manager/image_container/ **image_container.hpp**
- src/file_manager/image_container/ **image_container.cpp**

5.18 Dokumentacja klasy line

Klasa służąca do rysowania prostej linii.

```
#include <line.hpp>
```

Diagram dziedziczenia dla line



Metody publiczne

- **line ()**

Konstruktor klasy, służy do ustawienia pola type;.

Metody chronione

- void **draw_shape** (std::size_t x_position, std::size_t y_position)

Metoda rysuje linię od punktu początkowego do wskazanego przez argumenty wywołania. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.18.1 Opis szczegółowy

Klasa służąca do rysowania prostej linii.

5.18.2 Dokumentacja konstruktora i destruktora

5.18.2.1 line()

```
line::line ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.18.3 Dokumentacja funkcji składowych

5.18.3.1 draw_shape()

```
void line::draw_shape (
    std::size_t x_position,
    std::size_t y_position ) [protected], [virtual]
```

Metoda rysuje linię od punktu początkowego do wskazanego przez argumenty wywołania. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **shape** (str. 78).

Reimplementowana w **rectangle** (str. 74) i **triangle** (str. 98).

Dokumentacja dla tej klasy została wygenerowana z plików:

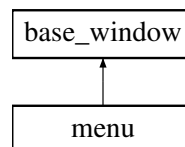
- src/tools/shapes/ **line.hpp**
- src/tools/shapes/ **line.cpp**

5.19 Dokumentacja klasy menu

Klasa służąca do rysowania menu oraz obsługi kliknięć w jego obrębie.

```
#include <menu.hpp>
```

Diagram dziedziczenia dla menu



Metody publiczne

- **menu ()**
*Konstruktor klasy. Otwiera pliki zawierające etykiety przycisków i narzędzi oraz wypełnia tablice pikseli ich zawartością. Jeśli nie uda się otworzyć bądź odczytać jednego z pliku zostanie ustawiona flaga informująca o tym fakcie. Konstruktor tej klasy "rzuca" wyjątki typu **paint_exception** (str. 61), w przypadku ich wystąpienia program nie może działać prawidłowo.*
- void **update_window_information** (Uint32 * **window**, std::size_t **menu_height**, std::size_t **window_width**)
*Metoda zapewnia spójność danych z klasą window dotyczących rozmiaru okna i jego tablicy pikseli. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.*
- void **update_tool_pointer** (tool * **tool_ptr**)
*Metoda zapewnia spójność danych z klasą window dotyczących bieżącego narzędzia. Klasa potrzebuje tych informacji, ponieważ odpowiada za wyświetlanie informacji o stanie bieżącego narzędzia. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.*
- void **print_menu** ()
Metoda odpowiada za rysowanie menu programu. Funkcja nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.
- void **update_colors** ()
*Metoda rysuje kolory w menu, które są zawarte w kolejce. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku braku miejsca do rysowania.*
- void **update_brush_size_state** ()
*Metoda rysuje w menu czarny kwadrat, który reprezentuje rozmiar narzędzia. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku rozmiaru pędzla równego 0.*
- void **update_tool_state** ()

Metoda rysuje etykietę bieżącego narzędzia. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

- void **add_color_to_queue** (Uint32 color)

Metoda dodaje kolor podany jako argument do kolejki kolorów. Metoda nie zwraca żadnej wartości.

- void **print_current_color** ()

Metoda rysuje w menu kwadrat w kolorze pobieranym od bieżącego narzędzia. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątek typu **paint_exception** (str. 61), gdy dodanie koloru do kolejki się nie powiedzie.

- bool **is_in_color_menu** (std::size_t x_position)

Metoda informuje czy punkt (wystarczy tylko położenie X) znajduje się o obszarze kolorów w menu. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

- bool **is_in_buttons_area** (std::size_t x_position)

Metoda informuje czy punkt (wystarczy tylko położenie X) znajduje się o obszarze przycisków w menu. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

- enum **menu_buttons get_clicked_button** (std::size_t x_position)

Metoda zwraca na podstawie punktu kliknięcia (wystarczy tylko położenie X) jaki przycisk menu został wciśnięty przez użytkownika. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

- ~**menu** ()

Destruktor klasy, zwalnia pamięć po wszystkich zmiennych zaalokowanych dynamicznie.

Atrybuty prywatne

- std::size_t **NUMBER_OF_BUTTONS**
- std::size_t **NUMBER_OF_TOOLS**
- std::deque< std::string > **menu_buttons_images_paths**
- std::deque< Uint32 * > **menu_buttons_labels**
- tool * **tool_prt** = nullptr
- std::deque< Uint32 * > **tool_labels**
- std::deque< std::string > **tools_images_paths**
- std::size_t **colors_to_print**
- std::deque< Uint32 > **color_buffer**

Statyczne atrybuty prywatne

- static const std::size_t **NUMBER_OF_LEFT_ELEMENTS** = 3
- static const std::size_t **COLOR_BUFFER_MAX_SIZE** = 25

Dodatkowe Dziedziczone Składowe

5.19.1 Opis szczegółowy

Klasa służąca do rysowania menu oraz obsługi kliknięć w jego obrębie.

5.19.2 Dokumentacja konstruktora i destruktor

5.19.2.1 menu()

```
menu::menu ( )
```

Konstruktor klasy. Otwiera pliki zawierające etykiety przycisków i narzędzi oraz wypełnia tablice pikseli ich zawartością. Jeśli nie uda się otworzyć bądź odczytać jednego z pliku zostanie ustawiona flaga informująca o tym fakcie. Konstruktor tej klasy "rzuca" wyjątki typu **paint_exception** (str. 61), w przypadku ich wystąpienia program nie może działać prawidłowo.

5.19.2.2 ~menu()

```
menu::~~menu ( )
```

Destruktor klasy, zwalnia pamięć po wszystkich zmiennych zaalokowanych dynamicznie.

5.19.3 Dokumentacja funkcji składowych

5.19.3.1 add_color_to_queue()

```
void menu::add_color_to_queue (
    Uint32 color )
```

Metoda dodaje kolor podany jako argument do kolejki kolorów. Metoda nie zwraca żadnej wartości.

Parametry

in	color	Uint32 Kolor (format RGBA) jaki ma zostać dodany do kolejki.
----	-------	--

5.19.3.2 get_clicked_button()

```
enum menu_buttons menu::get_clicked_button (
    std::size_t x_position )
```

Metoda zwraca na podstawie punktu kliknięcia (wystarczy tylko położenie X) jaki przycisk menu został wciśnięty przez użytkownika. Metoda może "rzucać" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

Parametry

in	x_position	std::size_t Składowa 'x' punktu kliknięcia.
----	------------	---

Zwraca

enum menu_buttons Wartość typu wyliczeniowego odpowiadająca klikniętemu przez użytkownika przyciskowi.

5.19.3.3 is_in_buttons_area()

```
bool menu::is_in_buttons_area (
    std::size_t x_position )
```

Metoda informuje czy punkt (wystarczy tylko położenie X) znajduje się o obszarze przycisków w menu. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

Parametry

in	x_position	std::size_t Składowa 'x' punktu kliknięcia.
----	------------	---

Zwraca

bool Metoda zwraca true jeśli punkt znajduje się w obszarze przycisków w menu, w przeciwnym wypadku zwraca false.

5.19.3.4 is_in_color_menu()

```
bool menu::is_in_color_menu (
    std::size_t x_position )
```

Metoda informuje czy punkt (wystarczy tylko położenie X) znajduje się o obszarze kolorów w menu. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

Parametry

in	x_position	std::size_t Składowa 'x' punktu kliknięcia.
----	------------	---

Zwraca

bool Metoda zwraca true jeśli punkt znajduje się w obszarze kolorów w menu, w przeciwnym wypadku zwraca false.

5.19.3.5 print_current_color()

```
void menu::print_current_color ( )
```

Metoda rysuje w menu kwadrat w kolorze pobieranym od bieżącego narzędzia. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątek typu **paint_exception** (str. 61), gdy dodanie koloru do kolejki się nie powiedzie.

5.19.3.6 print_menu()

```
void menu::print_menu ( )
```

Metoda odpowiada za rysowanie menu programu. Funkcja nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

5.19.3.7 update_brush_size_state()

```
void menu::update_brush_size_state ( )
```

Metoda rysuje w menu czarny kwadrat, który reprezentuje rozmiar narzędzia. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku rozmiaru pędzla równego 0.

5.19.3.8 update_colors()

```
void menu::update_colors ( )
```

Metoda rysuje kolory w menu, które są zawarte w kolejce. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku braku miejsca do rysowania.

5.19.3.9 update_tool_pointer()

```
void menu::update_tool_pointer (
    tool * tool_ptr )
```

Metoda zapewnia spójność danych z klasą window dotyczących bieżącego narzędzia. Klasa potrzebuje tych informacji, ponieważ odpowiada za wyświetlanie informacji o stanie bieżącego narzędzia. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

Parametry

in	tool_ptr	tool* Wskaźnik na bieżące narzędzie.
----	----------	--------------------------------------

5.19.3.10 update_tool_state()

```
void menu::update_tool_state ( )
```

Metoda rysuje etykietę bieżącego narzędzia. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.19.3.11 update_window_information()

```
void menu::update_window_information (
    Uint32 * window,
    std::size_t menu_height,
    std::size_t window_width )
```

Metoda zapewnia spójność danych z klasą window dotyczących rozmiaru okna i jego tablicy pikseli. Metoda nie zwraca żadnej wartości, może "rzucić" wyjątki typu **paint_exception** (str. 61) w przypadku nieprawidłowych danych wejściowych.

Parametry

in	<i>window</i>	Uint32* Wskaźnik na tablicę pikseli okna.
in	<i>menu_height</i>	std::size_t Wysokość menu okna.
in	<i>window_width</i>	std::size_t Szerokość okna.

5.19.4 Dokumentacja atrybutów składowych

5.19.4.1 color_buffer

```
std::deque<Uint32> menu::color_buffer [private]
```

Wartość początkowa:

```
= {0x000000, 0x808080, 0xFFFFFF, 0x800000, 0xFF0000,
    0xFFA500, 0x800080, 0xFF69B4, 0x008000, 0x08E100,
    0x93FF2D, 0xFFFF00, 0xFFFF83, 0x000080, 0x0000FF}
```

Kolejka zawierająca kolory wyświetlane w menu.

5.19.4.2 COLOR_BUFFER_MAX_SIZE

```
const std::size_t menu::COLOR_BUFFER_MAX_SIZE = 25 [static], [private]
```

Maksymalna wielkość bufora kolorów w menu.

5.19.4.3 colors_to_print

```
std::size_t menu::colors_to_print [private]
```

Liczba kolorów jaką można narysować przy danym rozmiarze okna programu.

5.19.4.4 menu_buttons_images_paths

```
std::deque<std::string> menu::menu_buttons_images_paths [private]
```

Kolejka ze ścieżkami do plików zawierających etykiety przycisków.

5.19.4.5 menu_buttons_labels

```
std::deque<Uint32*> menu::menu_buttons_labels [private]
```

Kolejka wskaźników na tablice pikseli z etykietami przycisków.

5.19.4.6 NUMBER_OF_BUTTONS

```
std::size_t menu::NUMBER_OF_BUTTONS [private]
```

Liczba przycisków w menu.

5.19.4.7 NUMBER_OF_LEFT_ELEMENTS

```
const std::size_t menu::NUMBER_OF_LEFT_ELEMENTS = 3 [static], [private]
```

Liczba elementów sygnalizujących stan bieżącego narzędzia (kolor, rozmiar, typ).

5.19.4.8 NUMBER_OF_TOOLS

```
std::size_t menu::NUMBER_OF_TOOLS [private]
```

Liczba rodzajów narzędzi.

5.19.4.9 tool_labels

```
std::deque<Uint32*> menu::tool_labels [private]
```

Kolejka wskaźników na tablice pikseli zawierające etykiety narzędzi.

5.19.4.10 tool_ptr

```
tool* menu::tool_ptr = nullptr [private]
```

Wskaźnik na bieżące narzędzie.

5.19.4.11 `tools_images_paths`

```
std::deque<std::string> menu::tools_images_paths [private]
```

Kolejka ze ścieżkami do plików zawierających etykiety narzędzi.

Dokumentacja dla tej klasy została wygenerowana z plików:

- `src/window/ menu.hpp`
- `src/window/ menu.cpp`

5.20 Dokumentacja klasy `paint_exception`

Klasa do obsługi wyjątków programu.

```
#include <paint_exception.hpp>
```

Typy publiczne

- enum `exception_type` {
`WINDOW_BAD_ALLOC`, `WINDOW_UPDATING_DATA`, `WINDOW_PIXELS_SWAP`, `WINDOW_SDL_INIT`,
`WINDOW_SDL_RENDER`, `WINDOW_SDL_RESIZED`, `WINDOW_SDL_DIALOG`, `BAD_WINDOW_DIM`,
`BAD_WINDOW_PTR`, `BAD_TOOL_PTR`, `MENU_BAD_ALLOC`, `MENU_RESOURCES`,
`MENU_NUMBER_OF_BUTTONS`, `MENU_BUTTON_LABELS_PATH`, `MENU_TOOL_LABELS_PATH`,
`MENU_BUTTON_LABEL`,
`MENU_TOOL_LABEL`, `MENU_COLOR_SPACE`, `MENU_BRUSH_SIZE`, `MENU_PRINT_TOOL_LABEL`,
`MENU_BAD_X_POSITION`, `GTK_WINDOW_RESOURCES`, `GTK_WINDOW_BAD_ALLOC`, `GTK_WINDOW_INIT`,
`UNSUPPORTED_IMAGE_FILE`, `IMAGE_FILE_ERROR`, `IMAGE_CONTAINER_PIXEL_ARRAY`, `IMAGE_CONTAINER_BAD_RGBA8888_PIXBUF`,
`IMAGE_CONTAINER_OPERATOR`, `IMAGE_CONTAINER_BAD_ALLOC`, `IMAGE_CONTAINER_BAD_PPM`, `IMAGE_CONTAINER_BAD_BMP`,
`IMAGE_CONTAINER_BAD_PNG`, `SHAPE_BAD_ALLOC` }

Typ wyliczeniowy służący do obsługi wyjątków w programie.

Metody publiczne

- `paint_exception` (enum `exception_type`)
Konstruktor klasy. Tworzy komunikat wyjątku na podstawie tylko jego kodu.
- `paint_exception` (enum `exception_type`, `std::string additional_data`)
Konstruktor klasy. Tworzy komunikat wyjątku na podstawie jego kodu oraz dodatkowych danych typu `std::string`.
- `paint_exception` (enum `exception_type`, `const char *additional_data`)
Konstruktor klasy. Tworzy komunikat wyjątku na podstawie jego kodu oraz dodatkowych danych typu `const char`.*
- `std::string what ()`
Metoda zwraca komunikat wyjątku w zmiennej typu `std::string`. Nie przyjmuje argumentów wywołania.
- `virtual ~paint_exception ()`
Destruktor klasy.

Atrybuty prywatne

- `std::string exception_string = ""`
- `std::string exception_strings [number_of_exceptions]`

Statyczne atrybuty prywatne

- `static const std::size_t number_of_exceptions = 34`

5.20.1 Opis szczegółowy

Klasa do obsługi wyjątków programu.

5.20.2 Dokumentacja składowych wyliczanych

5.20.2.1 exception_type

`enum paint_exception::exception_type`

Typ wyliczeniowy służący do obsługi wyjątków w programie.

Wartości wyliczeń

WINDOW_BAD_ALLOC	Błąd alokowania pamięci w obiekcie klasy window.
WINDOW_UPDATING_DATA	Błąd aktualizowania danych między obiektami programu.
WINDOW_PIXELS_SWAP	Błąd podmieniania tablicy pikseli.
WINDOW_SDL_INIT	Błąd inicjowania okna SDL2.
WINDOW_SDL_RENDER	Błąd podczas renderowania okna.
WINDOW_SDL_RESIZED	Błąd podczas zmiany rozmiaru okna.
WINDOW_SDL_DIALOG	Błąd otwarcia okna dialogowego SDL2.
BAD_WINDOW_DIM	Podano błędne rozmiary okna.
BAD_WINDOW_PTR	Podano błędny wskaźnik na okno.
BAD_TOOL_PTR	Podano błędny wskaźnik na narzędzie.
MENU_BAD_ALLOC	Błąd alokowania pamięci w obiekcie klasy menu.
MENU_RESOURCES	Błąd otwarcia/odczytu pliku z zasobami klasy menu.
MENU_NUMBER_OF_BUTTONS	Liczba przycisków równa zero.
MENU_BUTTON_LABELS_PATH	Błąd otwarcia/odczytu pliku ze ścieżkami do etykiet przycisków.
MENU_TOOL_LABELS_PATH	Błąd otwarcia/odczytu pliku ze ścieżkami do etykiet narzędzi.
MENU_BUTTON_LABEL	Błąd otwarcia/odczytu pliku z etykietą przycisku.
MENU_TOOL_LABEL	Błąd otwarcia/odczytu pliku z etykietą narzędzia.
MENU_COLOR_SPACE	Brak miejsca na rysowanie kolorów z kolejki.

Wartości wyliczeń

<code>MENU_BRUSH_SIZE</code>	Próba narysowania stanu narzędzia o rozmiarze równym zero.
<code>MENU_PRINT_TOOL_LABEL</code>	Błąd podczas rysowania etykiety narzędzia.
<code>MENU_BAD_X_POSITION</code>	Podano współrzędną X większą niż rozmiar okna.
<code>GTK_WINDOW_RESOURCES</code>	Błąd podczas otwierania/odczytu pliku zasobów klasy.
<code>GTK_WINDOW_BAD_ALLOC</code>	Błąd alokacji pamięci w obiekcie obsługującym okno GTK.
<code>GTK_WINDOW_INIT</code>	Błąd podczas inicjowania okna GTK.
<code>UNSUPPORTED_IMAGE_FILE</code>	Niewspierany typ obrazu.
<code>IMAGE_FILE_ERROR</code>	Błąd odczytu obrazu.
<code>IMAGE_CONTAINER_PIXEL_ARRAY</code>	Docelowa tablica pikseli nie została zainicjowana.
<code>IMAGE_CONTAINER_BAD_RGB888_PIXBUF</code>	Podano wskaźnik o wartości NULL na tablicę źródłową pikseli.
<code>IMAGE_CONTAINER_OPERATOR</code>	Przekroczono zakres tablicy pikseli przy używaniu operatora[]
<code>IMAGE_CONTAINER_BAD_ALLOC</code>	Błąd alokowania tablicy w klasie <code>image_container</code> (str. 48).
<code>IMAGE_CONTAINER_BAD_PPM</code>	Próba otwarcia pliku PPM o niespodziewanej strukturze.
<code>IMAGE_CONTAINER_BAD_BMP</code>	Próba otwarcia pliku BMP o niespodziewanej strukturze.
<code>IMAGE_CONTAINER_BAD_PNG</code>	Próba otwarcia pliku PNG o niespodziewanej strukturze.
<code>SHAPE_BAD_ALLOC</code>	Błąd alokowania pamięci podczas rysowania figury geometrycznej.

5.20.3 Dokumentacja konstruktora i destruktor

5.20.3.1 `paint_exception()` [1/3]

```
paint_exception::paint_exception (
    enum exception_type type )
```

Konstruktor klasy. Tworzy komunikat wyjątku na podstawie tylko jego kodu.

Parametry

<code>in</code>	<code>exception_type</code>	enum Kod wyjątku.
-----------------	-----------------------------	-------------------

5.20.3.2 `paint_exception()` [2/3]

```
paint_exception::paint_exception (
```

```
enum exception_type type,
std::string additional_data )
```

Konstruktor klasy. Tworzy komunikat wyjątku na podstawie jego kodu oraz dodatkowych danych typu std::string.

Parametry

in	<i>exception_type</i>	enum Kod wyjątku.
in	<i>additional_data</i>	std::string Dodatkowe informacje.

5.20.3.3 paint_exception() [3/3]

```
paint_exception::paint_exception (
    enum exception_type type,
    const char * additional_data )
```

Konstruktor klasy. Tworzy komunikat wyjątku na podstawie jego kodu oraz dodatkowych danych typu const char*.

Parametry

<i>exception_type</i>	enum Kod wyjątku.
<i>additional_data</i>	const char* Dodatkowe informacje

5.20.3.4 ~paint_exception()

```
paint_exception::~paint_exception ( ) [virtual]
```

Destruktor klasy.

5.20.4 Dokumentacja funkcji składowych

5.20.4.1 what()

```
std::string paint_exception::what ( )
```

Metoda zwraca komunikat wyjątku w zmiennej typu std::string. Nie przyjmuje argumentów wywołania.

Zwraca

std::string Komunikat wyjątku.

5.20.5 Dokumentacja atrybutów składowych

5.20.5.1 exception_string

```
std::string paint_exception::exception_string = " " [private]
```

Komunikat wyjątku.

5.20.5.2 exception_strings

```
std::string paint_exception::exception_strings[ number_of_exceptions] [private]
```

Wartość początkowa:

```
= { "Failed to allocate memory in window object.",
    error: " ",
    SDL_Error: " ",
    to: " ",
    ,
    assets/resources/menu.rc",
    form resources file.",
    resources file.",
    file: " ",
    color buffer to print",
    unexpected type.",
    method: " ",
    GTK menu object.",
    ,
    pixbuf array.",
    container object: " ",
    "Failed to update information in: ",
    "Undo facility will not work due to an
    "SDL Window could not initialize!
    "Render failed! SDL_Error: ",
    "Resize failed due to: ",
    "Opening SDL2 dialog failed. ",
    "Trying to point invalid window dimensions
    "Trying to point null window pointer to: ",
    "Trying to point null tool pointer to: ",
    "Failed to allocate memory in menu object."
    "Failed to open/read resource file:
    "Number of menu buttons couldn't be 0.",
    "Failed to load menu buttons labels paths
    "Failed to load tool labels paths form
    "Failed to load menu button label form
    "Failed to load tool label form file: ",
    "There's no space in menu for colors form
    "Trying to print brush size which is 0.",
    "Trying to print tool label for tool with
    "Pointed to position outside window in
    "Failed to open/read resource file: ",
    "Failed to allocate memory in one of the
    "Failed to initialize GTK menu: ",
    "Unsupported file type. ",
    "Failed to open/read image file: ",
    "Trying to fill uninitialized pixel array."
    "Trying to fill pixel with null pointer
    "Image container operator[] out of range!",
    "Failed to allocate memory in image
    "Unsupported pixmap. ",
    "Unsupported bmp file. ",
    " ",
    "Failed to use this tool because of: "
}
```

Tablica komunikatów wyjątków.

5.20.5.3 number_of_exceptions

```
const std::size_t paint_exception::number_of_exceptions = 34 [static], [private]
```

Liczba wyjątków w programie

Dokumentacja dla tej klasy została wygenerowana z plików:

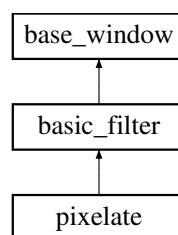
- src/ **paint_exception.hpp**
- src/ **paint_exception.cpp**

5.21 Dokumentacja klasy pixelate

Klasa filtra pikselizującego obraz.

```
#include <pixelate.hpp>
```

Diagram dziedziczenia dla pixelate



Metody publiczne

- **pixelate ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_fiter_properties ()**
Metoda otwiera okno GTK, w którym można ustawić rozmiar komórki(wielkość "piksela" wyjściowego). Metoda nie zwraca żadnej wartości.
- void **do_filter ()**
Metoda dzieli obszar na kwadraty o boku równemu rozmiarowi komórki. Dla każdego obszaru wylicza średnie wartości składowych koloru. Następnie ustawia ten kolor dla wszystkich pikseli w obszarze. Metoda nie zwraca żadnej wartości.

Statyczne metody prywatne

- static void **set_confirmed_flag** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości filtra. Metoda nie zwraca żadnej wartości.
- static void **value_changed** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Atrybuty prywatne

- `std::size_t cell_size = 1`

Dodatkowe Dziedziczone Składowe

5.21.1 Opis szczegółowy

Klasa filtra pikselizującego obraz.

5.21.2 Dokumentacja konstruktora i destruktora

5.21.2.1 pixelate()

```
pixelate::pixelate ( )
```

Konstruktor klasy, służy do ustawienia pola `type`;

5.21.3 Dokumentacja funkcji składowych

5.21.3.1 do_filter()

```
void pixelate::do_filter ( ) [virtual]
```

Metoda dzieli obszar na kwadraty o boku równemu rozmiarowi komórki. Dla każdego obszaru wylicza średnie wartości składowych koloru. Następnie ustawia ten kolor dla wszystkich pikseli w obszarze. Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

5.21.3.2 open_fiter_properties()

```
void pixelate::open_fiter_properties ( ) [virtual]
```

Metoda otwiera okno GTK, w którym można ustawić rozmiar komórki (wielkość "piksela" wyjściowego). Metoda nie zwraca żadnej wartości.

Zwraca

`void`

Implementuje **basic_filter** (str. 14).

5.21.3.3 set_confirmed_flag()

```
static void pixelate::set_confirmed_flag (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości filtra. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno GTK, w którym generowany jest sygnał.
out	<i>data</i>	gpointer Dodatkowe dane, w tym przypadku wskaźnik na flagę typu bool.

5.21.3.4 value_changed()

```
static void pixelate::value_changed (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno, w którym wygenerowany został sygnał.
out	<i>data</i>	gpointer Dodatkowe dane. W tym przypadku liczba naturalna.

5.21.4 Dokumentacja atrybutów składowych

5.21.4.1 cell_size

```
std::size_t pixelate::cell_size = 1 [private]
```

Rozmiar wyjściowego ""piksela"/

Dokumentacja dla tej klasy została wygenerowana z plików:

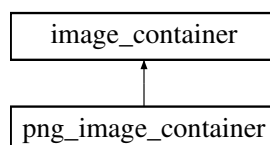
- src/filters/ **pixelate.hpp**
- src/filters/ **pixelate.cpp**

5.22 Dokumentacja klasy png_image_container

Kontener dla plików png.

```
#include <png_image_container.hpp>
```

Diagram dziedziczenia dla png_image_container



Metody publiczne

- **`png_image_container`** (`std::string file_path`)
Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.
- **`png_image_container`** (`std::size_t width`, `std::size_t height`, `std::size_t number_of_colors`)
Konstruktor. Inicjuje pusty kontener o podanych parametrach.
- **`void save_to_file`** (`std::string file_path`)
Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.
- **`~png_image_container`** ()
Destruktor.

Dodatkowe Dziedziczone Składowe

5.22.1 Opis szczegółowy

Kontener dla plików png.

5.22.2 Dokumentacja konstruktora i destruktora

5.22.2.1 `png_image_container()` [1/2]

```
png_image_container::png_image_container (
    std::string file_path )
```

Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.

Parametry

in	<i>file_path</i>	<code>std::string</code> ścieżka do pliku zawierającego obraz.
----	------------------	--

5.22.2.2 `png_image_container()` [2/2]

```
png_image_container::png_image_container (
    std::size_t width,
    std::size_t height,
    std::size_t number_of_colors )
```

Konstruktor. Inicjuje pusty kontener o podanych parametrach.

Parametry

in	<i>width</i>	<code>std::size_t</code> szerokość obrazu w kontenerze.
in	<i>height</i>	<code>std::size_t</code> wysokość obrazu w kontenerze.
in	<i>number_of_colors</i>	<code>std::size_t</code> liczba składowych koloru w kontenerze (3 lub 4).

5.22.2.3 ~png_image_container()

```
png_image_container::~~png_image_container ( )
```

Destruktor.

5.22.3 Dokumentacja funkcji składowych

5.22.3.1 save_to_file()

```
void png_image_container::save_to_file (
    std::string file_path ) [virtual]
```

Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.

Parametry

in	<i>file_path</i>	std::string Ścieżka do pliku, który ma być zapisany.
----	------------------	--

Implementuje **image_container** (str. 51).

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/file_manager/image_container/ **png_image_container.hpp**
- src/file_manager/image_container/ **png_image_container.cpp**

5.23 Dokumentacja struktury ppm_image_container::ppm_file_info

Struktura pomocnicza przechowująca informacje o obrazach ppm.

Atrybuty publiczne

- std::uint32_t **width**
- std::uint32_t **height**
- std::uint32_t **bits_per_color**

5.23.1 Opis szczegółowy

Struktura pomocnicza przechowująca informacje o obrazach ppm.

5.23.2 Dokumentacja atrybutów składowych

5.23.2.1 bits_per_color

```
std::uint32_t ppm_image_container::ppm_file_info::bits_per_color
```

Liczba bitów na składową koloru.

5.23.2.2 height

```
std::uint32_t ppm_image_container::ppm_file_info::height
```

Wysokość obrazu.

5.23.2.3 width

```
std::uint32_t ppm_image_container::ppm_file_info::width
```

Szerokość obrazu.

Dokumentacja dla tej struktury została wygenerowana z pliku:

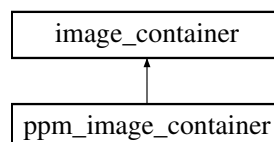
- src/file_manager/image_container/ **ppm_image_container.hpp**

5.24 Dokumentacja klasy ppm_image_container

Kontener dla plików ppm.

```
#include <ppm_image_container.hpp>
```

Diagram dziedziczenia dla ppm_image_container



Komponenty

- struct **ppm_file_info**

Struktura pomocnicza przechowująca informacje o obrazach ppm.

Metody publiczne

- **ppm_image_container** (std::string file_path)
Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.
- **ppm_image_container** (std::size_t width, std::size_t height, std::size_t number_of_colors)
Konstruktor. Inicjuje pusty kontener o podanych parametrach.
- void **save_to_file** (std::string file_path)
Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.
- **~ppm_image_container** ()
Destruktor.

Dodatkowe Dziedziczone Składowe

5.24.1 Opis szczegółowy

Kontener dla plików ppm.

5.24.2 Dokumentacja konstruktora i destruktora

5.24.2.1 ppm_image_container() [1/2]

```
ppm_image_container::ppm_image_container (
    std::string file_path )
```

Konstruktor. Inicjuje kontener na podstawie pliku o podanej ścieżce.

Parametry

in	<i>file_path</i>	std::string ścieżka do pliku zawierającego obraz.
----	------------------	---

5.24.2.2 ppm_image_container() [2/2]

```
ppm_image_container::ppm_image_container (
    std::size_t width,
    std::size_t height,
    std::size_t number_of_colors )
```

Konstruktor. Inicjuje pusty kontener o podanych parametrach.

Parametry

in	<i>width</i>	std::size_t szerokość obrazu w kontenerze.
in	<i>height</i>	std::size_t wysokość obrazu w kontenerze.
in	<i>number_of_colors</i>	std::size_t liczba składowych koloru w kontenerze (3 lub 4).

5.24.2.3 ~ppm_image_container()

```
ppm_image_container::~~ppm_image_container ( )
```

Destruktor.

5.24.3 Dokumentacja funkcji składowych

5.24.3.1 save_to_file()

```
void ppm_image_container::save_to_file (
    std::string file_path ) [virtual]
```

Metoda zapisuje zawartość kontenera do pliku, do którego ścieżka została podana jako argument wywołania.

Parametry

in	file_path	std::string Ścieżka do pliku, który ma być zapisany.
----	-----------	--

Implementuje **image_container** (str. 51).

Dokumentacja dla tej klasy została wygenerowana z plików:

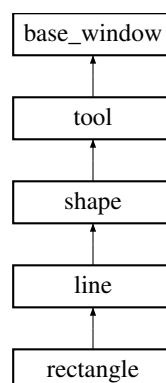
- src/file_manager/image_container/ **ppm_image_container.hpp**
- src/file_manager/image_container/ **ppm_image_container.cpp**

5.25 Dokumentacja klasy rectangle

Klasa rysująca prostokąt.

```
#include <rectangle.hpp>
```

Diagram dziedziczenia dla rectangle



Metody publiczne

- **rectangle ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_tool_properties ()**
Metoda otwiera okno właściwości narzędzia. Pozwala ustawić flagę rysowania kwadratu.

Metody prywatne

- void **draw_shape** (std::size_t x_position, std::size_t y_position)
Metoda rysuje prostokąt od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Statyczne metody prywatne

- static void **set_square_flag** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał kliknięcia w przycisk oknie właściwości narzędzia. Ustawia flagę rysowania kwadratu. Metoda nie zwraca żadnej wartości.

Atrybuty prywatne

- bool **is_square** = false

Dodatkowe Dziedziczone Składowe

5.25.1 Opis szczegółowy

Klasa rysująca prostokąt.

5.25.2 Dokumentacja konstruktora i destruktora

5.25.2.1 rectangle()

```
rectangle::rectangle ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.25.3 Dokumentacja funkcji składowych

5.25.3.1 draw_shape()

```
void rectangle::draw_shape (
    std::size_t x_position,
    std::size_t y_position ) [private], [virtual]
```

Metoda rysuje prostokąt od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **line** (str. 53).

5.25.3.2 open_tool_properties()

```
void rectangle::open_tool_properties ( ) [virtual]
```

Metoda otwiera okno właściwości narzędzia. Pozwala ustawić flagę rysowania kwadratu.

Reimplementowana z **tool** (str. 88).

5.25.3.3 set_square_flag()

```
static void rectangle::set_square_flag (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał kliknięcia w przycisk oknie właściwości narzędzia. Ustawia flagę rysowania kwadratu. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno GTK, w którym generowany jest sygnał.
out	<i>data</i>	gpointer Dodatkowe dane, w tym przypadku wskaźnik na flagę typu bool.

5.25.4 Dokumentacja atrybutów składowych

5.25.4.1 is_square

```
bool rectangle::is_square = false [private]
```

Flaga rysowania kwadratu.

Dokumentacja dla tej klasy została wygenerowana z plików:

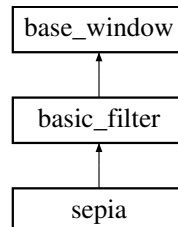
- src/tools/shapes/ **rectangle.hpp**
- src/tools/shapes/ **rectangle.cpp**

5.26 Dokumentacja klasy sepia

Klasa filtra sepia.

```
#include <sepia.hpp>
```

Diagram dziedziczenia dla sepia



Metody publiczne

- **sepia ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_fiter_properties ()**
Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.
- void **do_filter ()**
Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, mnoży wartości przez odpowiedni licznik i zapisuje nową wartość. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.26.1 Opis szczegółowy

Klasa filtra sepia.

5.26.2 Dokumentacja konstruktora i destruktor

5.26.2.1 sepia()

```
sepia::sepia ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.26.3 Dokumentacja funkcji składowych

5.26.3.1 do_filter()

```
void sepia::do_filter ( ) [virtual]
```

Metoda wyluskuje składowe kolorów wszystkich pikseli obszaru roboczego, mnoży wartości przez odpowiedni licznik i zapisuje nową wartość. Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

5.26.3.2 open_fiter_properties()

```
void sepia::open_fiter_properties ( ) [virtual]
```

Przesłonięta metoda otwierająca menu właściwości filtra. W przypadku tego filtra Metoda nie zwraca żadnej wartości.

Implementuje **basic_filter** (str. 14).

Dokumentacja dla tej klasy została wygenerowana z plików:

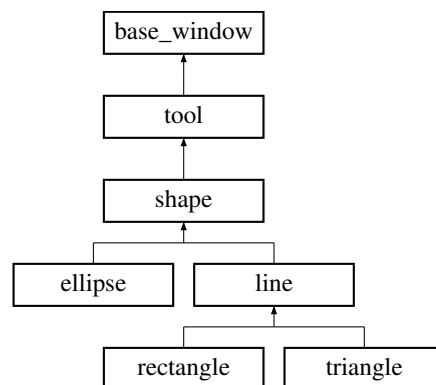
- src/filters/ **sepia.hpp**
- src/filters/ **sepia.cpp**

5.27 Dokumentacja klasy shape

Klasa bazowa dla klas rysujących figury geometryczne.

```
#include <shape.hpp>
```

Diagram dziedziczenia dla shape



Metody publiczne

- bool **is_brush** ()
Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.
- void **use_tool** (std::size_t x_position, std::size_t y_position)
*Metoda wykonuje akcję "ciągłego" rysowania kształtu. Może rzucić wyjątek typu **paint_exception** (str. 61) gdy dynamiczny przydział pamięci się nie powiedzie. Metoda nie zwraca żadnej wartości.*

Metody chronione

- virtual void **draw_shape** (std::size_t x_position, std::size_t y_position)

Metoda rysuje figurę od punktu początkowego do wskazanego przez argumenty wywołania. Metoda nie zwraca żadnej wartości.

Atrybuty chronione

- std::size_t **x_begining**
- std::size_t **y_begining**
- Uint32 * **copied_pixels**

Dodatkowe Dziedziczone Składowe

5.27.1 Opis szczegółowy

Klasa bazowa dla klas rysujących figury geometryczne.

5.27.2 Dokumentacja funkcji składowych

5.27.2.1 draw_shape()

```
void shape::draw_shape (
    std::size_t x_position,
    std::size_t y_position ) [protected], [virtual]
```

Metoda rysuje figurę od punktu początkowego do wskazanego przez argumenty wywołania. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana w **ellipse** (str. 29), **rectangle** (str. 74), **triangle** (str. 98) i **line** (str. 53).

5.27.2.2 is_brush()

```
bool shape::is_brush ( )
```

Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Zwraca

bool Metoda zwraca wartość false w przypadku tej klasy.

5.27.2.3 use_tool()

```
void shape::use_tool (
    std::size_t x_position,
    std::size_t y_position ) [virtual]
```

Metoda wykonuje akcję "ciąglego" rysowania kształtu. Może rzucić wyjątek typu **paint_exception** (str. 61) gdy dynamiczny przydział pamięci się nie powiedzie. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 90).

5.27.3 Dokumentacja atrybutów składowych**5.27.3.1 copied_pixels**

```
Uint32* shape::copied_pixels [protected]
```

Piksele skopiowane przed rozpoczęciem rysowania kształtu.

5.27.3.2 x_begining

```
std::size_t shape::x_begining [protected]
```

Składowa 'x' punktu rozpoczęcia rysowania kształtu.

5.27.3.3 y_begining

```
std::size_t shape::y_begining [protected]
```

Składowa 'y' punktu rozpoczęcia rysowania kształtu.

Dokumentacja dla tej klasy została wygenerowana z plików:

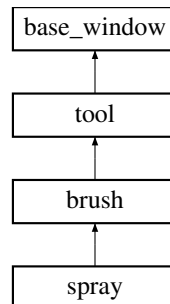
- src/tools/shapes/ **shape.hpp**
- src/tools/shapes/ **shape.cpp**

5.28 Dokumentacja klasy spray

Klasa narzędzia sprej.

```
#include <spray.hpp>
```

Diagram dziedziczenia dla spray



Metody publiczne

- **spray ()**
Konstruktor klasy, służy do ustawienia pola type;.
- void **open_tool_properties ()**
Metoda otwiera okno GTK, które pozwala ustawić współczynnik ilości generowanych pikseli. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

Metody chronione

- void **PaintPixel** (std::size_t x_position, std::size_t y_position)
Przesłonięta metoda PaintPixel z klasy tool. Na podstawie współczynnika (w %) generowana jest ilość pikseli, których kolor zostanie zmieniony. Do generowania użyty jest generator rozkładu równomiernego. Metoda nie zwraca żadnej wartości.

Statyczne metody prywatne

- static void **set_confirmed_flag** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości narzędzia. Metoda nie zwraca żadnej wartości.
- static void **value_changed** (GtkWidget *widget, gpointer data)
Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Atrybuty prywatne

- int **dots_ratio** = 15
- std::default_random_engine **generator**

Dodatkowe Dziedziczone Składowe

5.28.1 Opis szczegółowy

Klasa narzędzia sprej.

5.28.2 Dokumentacja konstruktora i destruktora

5.28.2.1 spray()

```
spray::spray ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.28.3 Dokumentacja funkcji składowych

5.28.3.1 open_tool_properties()

```
void spray::open_tool_properties ( ) [virtual]
```

Metoda otwiera okno GTK, które pozwala ustawić współczynnik ilości generowanych pikseli. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

Reimplementowana z **tool** (str. 88).

5.28.3.2 PaintPixel()

```
void spray::PaintPixel (
    std::size_t x_position,
    std::size_t y_position ) [protected], [virtual]
```

Przesłonięta metoda PaintPixel z klasy tool. Na podstawie współczynnika (w %) generowana jest ilość pikseli, których kolor zostanie zmieniony. Do generowania użyty jest generator rozkładu równomiernego. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 88).

5.28.3.3 set_confirmed_flag()

```
static void spray::set_confirmed_flag (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał kliknięcia w przycisk zatwierdzający zmiany wprowadzone w oknie właściwości narzędzia. Metoda nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno GTK, w którym generowany jest sygnał.
out	<i>data</i>	gpointer Dodatkowe dane, w tym przypadku wskaźnik na flagę typu bool.

5.28.3.4 value_changed()

```
static void spray::value_changed (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Metoda obsługująca sygnał zmiany wartości w przycisku ustawiania wartości. Nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno, w którym wygenerowany został sygnał.
out	<i>data</i>	gpointer Dodatkowe dane. W tym przypadku liczba naturalna.

5.28.4 Dokumentacja atrybutów składowych

5.28.4.1 dots_ratio

```
int spray::dots_ratio = 15 [private]
```

Współczynnik (w %) ilości pikseli, których kolor ma być zmieniony. Mnożony jest przez pole obszaru rysowania pędzla.

5.28.4.2 generator

```
std::default_random_engine spray::generator [private]
```

Generator rozkładu równomiernego.

Dokumentacja dla tej klasy została wygenerowana z plików:

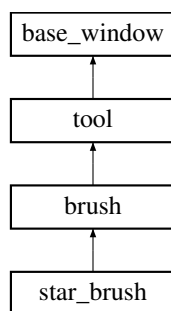
- src/tools/brushes/ **spray.hpp**
- src/tools/brushes/ **spray.cpp**

5.29 Dokumentacja klasy star_brush

Klasa narzędzia pędzla w kształcie plusa.

```
#include <star_brush.hpp>
```

Diagram dziedziczenia dla star_brush



Metody publiczne

- **star_brush ()**
Konstruktor klasy, służy do ustawienia pola type;.

Metody chronione

- void **PaintPixel** (std::size_t x_position, std::size_t y_position)
Przesłonięta metoda PaintPixel z klasy tool. Rysuje kształt plusa o rozmiarze pędzla. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.29.1 Opis szczegółowy

Klasa narzędzia pędzla w kształcie plusa.

5.29.2 Dokumentacja konstruktora i destruktora

5.29.2.1 star_brush()

```
star_brush::star_brush ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.29.3 Dokumentacja funkcji składowych

5.29.3.1 PaintPixel()

```
void star_brush::PaintPixel (
    std::size_t x_position,
    std::size_t y_position ) [protected], [virtual]
```

Przesłonięta metoda PaintPixel z klasy tool. Rysuje kształt plusa o rozmiarze pędzla. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana z **tool** (str. 88).

Dokumentacja dla tej klasy została wygenerowana z plików:

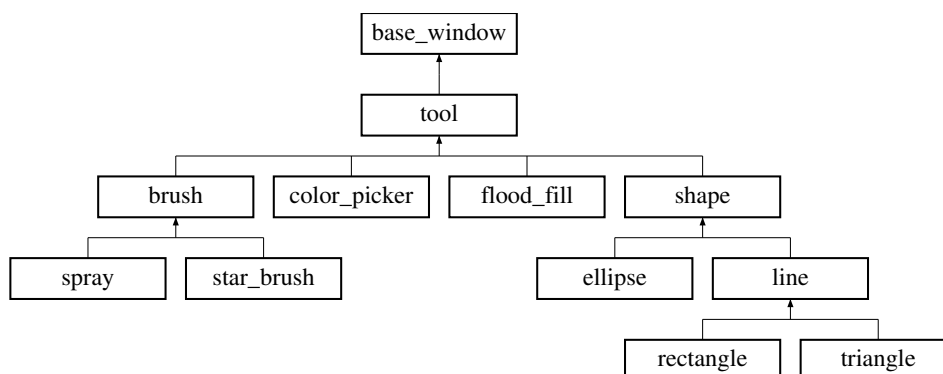
- src/tools/brushes/ **star_brush.hpp**
- src/tools/brushes/ **star_brush.cpp**

5.30 Dokumentacja klasy tool

Klasa bazowa dla wszystkich typów narzędzi.

```
#include <tool.hpp>
```

Diagram dziedziczenia dla tool



Metody publiczne

- virtual void **use_tool** (std::size_t x_position, std::size_t y_position)
Metoda wykonuje akcję zależną od typu narzędzia w podanym punkcie. Metoda nie zwraca żadnej wartości.
- virtual void **open_tool_properties** ()
- virtual bool **is_a_brush** ()
Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.
- virtual enum **tool_type get_tool_type** ()
Metoda zwraca informację o typie narzędzia. Nie przyjmuje argumentów wywołania.
- void **update_window_information** (UInt32 * window, std::size_t window_width, std::size_t window_height, std::size_t menu_height)
Metoda zapewnia spójność danych z klasą window dotyczących rozmiaru okna i jego tablicy pikseli. Metoda nie zwraca żadnej wartości.
- bool **resize_tool_size** (int difference)
Metoda zmienia rozmiar bieżącego narzędzia.
- void **update_current_color** (UInt32 new_color)
Metoda pozwala zmienić kolor narzędzia. Nie zwraca żadnej wartości.
- UInt32 **get_current_color** ()
Getter koloru narzędzia.
- std::size_t **get_brush_size** ()
Getter rozmiaru narzędzia.
- tool & **operator=** (const tool &old_tool)
Przeciążony operator przypisania. Kopiuje kolor i rozmiar narzędzia.
- virtual ~**tool** ()
Destruktor klasy.

Atrybuty publiczne

- bool **pressed_left_button** = false
- bool **pressed_left_button_second_time** = false

Metody chronione

- virtual void **PaintPixel** (std::size_t x_position, std::size_t y_position)
Metoda rysuje na ekranie kwadrat o środku w danym punkcie (na podstawie argumentów wywołania) oraz rozmiarze narzędzia. Metoda nie zwraca żadnej wartości.

Statyczne metody chronione

- static gboolean **delete_event** (GtkWidget *widget, GdkEvent *event, gpointer data)
Funkcja obsługująca sygnał usunięcia okna GTK.
- static void **destroy** (GtkWidget *widget, gpointer data)
Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Atrybuty chronione

- enum **tool_type** type
- Uint32 **current_color** = 0
- std::size_t **brush_size** = 15

5.30.1 Opis szczegółowy

Klasa bazowa dla wszystkich typów narzędzi.

5.30.2 Dokumentacja konstruktora i destruktora

5.30.2.1 ~tool()

```
tool::~~tool ( ) [virtual]
```

Destruktor klasy.

5.30.3 Dokumentacja funkcji składowych

5.30.3.1 delete_event()

```
static gboolean tool::delete_event (
    GtkWidget * widget,
    GdkEvent * event,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał usunięcia okna GTK.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>event</i>	GdkEvent* Zdarzenie.
<i>data</i>	gpointer Dane dla funkcji.

Zwraca

gboolean Funkcja zwraca wartość TRUE.

5.30.3.2 destroy()

```
static void tool::destroy (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [protected]
```

Funkcja obsługująca sygnał zamknięcia okna, nie zwraca żadnej wartości.

Parametry

<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
<i>data</i>	gpointer Dane dla funkcji.

5.30.3.3 get_brush_size()

```
std::size_t tool::get_brush_size ( )
```

Getter rozmiaru narzędzia.

Zwraca

std::size_t Metoda zwraca rozmiar narzędzia.

5.30.3.4 get_current_color()

```
UInt32 tool::get_current_color ( )
```

Getter koloru narzędzia.

Zwraca

UInt32 Metoda zwraca bieżący kolor narzędzia (format RGBA).

5.30.3.5 get_tool_type()

```
enum tool_type tool::get_tool_type ( ) [virtual]
```

Metoda zwraca informację o typie narzędzia. Nie przyjmuje argumentów wywołania.

Zwraca

enum tool_type Metoda zwraca wartość typu wyliczeniowego tool_type w zależności od typu narzędzia.

5.30.3.6 is_a_brush()

```
bool tool::is_a_brush ( ) [virtual]
```

Metoda zwraca informację czy narzędzie jest pędzlem. Nie przyjmuje argumentów wywołania.

Zwraca

bool Jeżeli narzędzie jest pędzlem metoda zwraca wartość true, w przeciwnym wypadku wartość false.

Reimplementowana w **brush** (str. 23).

5.30.3.7 open_tool_properties()

```
void tool::open_tool_properties ( ) [virtual]
```

Reimplementowana w **flood_fill** (str. 42), **spray** (str. 81), **rectangle** (str. 75) i **ellipse** (str. 29).

5.30.3.8 operator=()

```
tool& tool::operator= (
    const tool & old_tool ) [inline]
```

Przeciążony operator przypisania. Kopiuje kolor i rozmiar narzędzia.

5.30.3.9 PaintPixel()

```
void tool::PaintPixel (
    std::size_t x_position,
    std::size_t y_position ) [protected], [virtual]
```

Metoda rysuje na ekranie kwadrat o środku w danym punkcie (na podstawie argumentów wywołania) oraz rozmiarze narzędzia. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu środka kwadratu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu środka kwadratu.

Zwraca

void

Reimplementowana w **spray** (str. 81), **ellipse** (str. 30) i **star_brush** (str. 84).

5.30.3.10 `resize_tool_size()`

```
bool tool::resize_tool_size (
    int difference )
```

Metoda zmienia rozmiar bieżącego narzędzia.

Parametry

in	<i>difference</i>	int Różnica względem obecnego rozmiaru.
----	-------------------	---

Zwraca

Zwraca true jeśli operacja się powiedzie, w przeciwnym wypadku zwraca false.

5.30.3.11 `update_current_color()`

```
void tool::update_current_color (
    Uint32 new_color )
```

Metoda pozwala zmienić kolor narzędzia. Nie zwraca żadnej wartości.

Parametry

in	<i>new_color</i>	Uint32 Kolor (format RGBA) jaki ma zostać ustawiony.
----	------------------	--

5.30.3.12 `update_window_information()`

```
void tool::update_window_information (
    Uint32 * window,
```

```
std::size_t window_width,
std::size_t window_height,
std::size_t menu_height )
```

Metoda zapewnia spójność danych z klasą `window` dotyczących rozmiaru okna i jego tablicy pikseli. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>window</i>	UInt32* Wskaźnik na tablicę pikseli okna.
in	<i>window_width</i>	std::size_t Szerokość okna.
in	<i>window_height</i>	std::size_t Wysokość okna.
in	<i>menu_height</i>	std::size_t Wysokość menu okna.

Zwraca

void

5.30.3.13 use_tool()

```
void tool::use_tool (
    std::size_t x_position,
    std::size_t y_position ) [virtual]
```

Metoda wykonuje akcję zależną od typu narzędzia w podanym punkcie. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Reimplementowana w **brush** (str. 23), **shape** (str. 79), **flood_fill** (str. 43) i **color_picker** (str. 27).

5.30.4 Dokumentacja atrybutów składowych

5.30.4.1 brush_size

```
std::size_t tool::brush_size = 15 [protected]
```

Rozmiar narzędzia

5.30.4.2 current_color

```
UInt32 tool::current_color = 0 [protected]
```

Kolor, którym narzędzie rysuje po ekranie.

5.30.4.3 pressed_left_button

```
bool tool::pressed_left_button = false
```

Flaga informująca o stanie wciśnięcia lewego przycisku myszy.

5.30.4.4 pressed_left_button_second_time

```
bool tool::pressed_left_button_second_time = false
```

Flaga informująca o ponownym wciśnięciu lewego przycisku myszy.

5.30.4.5 type

```
enum tool_type tool::type [protected]
```

Typ narzędzia.

Dokumentacja dla tej klasy została wygenerowana z plików:

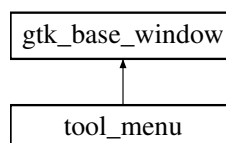
- src/tools/ **tool.hpp**
- src/tools/ **tool.cpp**

5.31 Dokumentacja klasy tool_menu

Klasa związana z wyświetlaniem menu wyboru narzędzi.

```
#include <tool_menu.hpp>
```

Diagram dziedziczenia dla tool_menu



Metody publiczne

- **tool_menu ()**
*Konstruktor klasy. Wczytuje z pliku zasoby dla tego menu z plików. Może "rzucić" wyjątek typu **paint_exception** (str. 61) kiedy nie powiedzie się odczytanie tych zasobów.*
- **bool open_gtk_window ()**
*Metoda zawiera inicjuje menu narzędzi oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.*
- **void set_current_color (UInt32 curent_color)**
Metoda konwertuje aktualny kolor używany przez narzędzie na format przyjazny dla GTK. Metoda nie zwraca żadnej wartości.
- **void set_current_tool (enum tool_type curent_tool)**
Metoda ustawia aktualne typ aktualnie używanego narzędzia. Nie zwraca żadnej wartości.
- **UInt32 get_color_response ()**
Metoda zwraca kolor wybrany przez użytkownika w oknie color selector w postaci RGBA. Metoda nie przyjmuje żadnych argumentów wywołania.
- **enum tool_type get_tool_response ()**
Metoda zwraca wybór użytkownika odnośnie typu narzędzia, nie przyjmuje argumentów wywołania.
- **~tool_menu ()**

Metody prywatne

- **GdkColor convert_from_UInt8 (UInt8 *color)**
Konweter koloru z formatu RGB 888 do formatu przyjaznego dla GTK.

Statyczne metody prywatne

- **static void get_tool_type (GtkWidget *widget, gpointer data)**
Funkcja obsługująca sygnał kliknięcia na przycisk z wyborem nowego narzędzia. Funkcja nie zwraca żadnej wartości.
- **static void trigger_color_chooser (GtkWidget *widget, gpointer data)**
Funkcja obsługująca sygnał kliknięcia na przycisk otwierający wybór koloru. Funkcja nie zwraca żadnej wartości.

Atrybuty prywatne

- **std::size_t number_of_tools**
- **enum tool_type tool_response**
- **std::deque< struct tool_type_data > tool_types_array**
- **UInt8 RGB [3] = {0, 0, 0}**
- **UInt32 new_color**

Dodatkowe Dziedziczone Składowe

5.31.1 Opis szczegółowy

Klasa związana z wyświetlaniem menu wyboru narzędzi.

5.31.2 Dokumentacja konstruktora i destruktor

5.31.2.1 tool_menu()

```
tool_menu::tool_menu ( )
```

Konstruktor klasy. Wczytuje z pliku zasoby dla tego menu z plików. Może "rzucić" wyjątek typu **paint_exception** (str. 61) kiedy nie powiedzie się odczytanie tych zasobów.

5.31.2.2 ~tool_menu()

```
tool_menu::~~tool_menu ( )
```

5.31.3 Dokumentacja funkcji składowych

5.31.3.1 convert_from_Uint8()

```
GdkColor tool_menu::convert_from_Uint8 (
    Uint8 * color ) [private]
```

Konwerter koloru z formatu RGB 888 do formatu przyjaznego dla GTK.

Parametry

<i>color</i>	Uint8* Tablica zawierająca składowe koloru RGB 888.
--------------	---

Zwraca

GdkColor Kolor w formacie przyjaznym dla GTK (RGBA 16 16 16 16).

5.31.3.2 get_color_response()

```
Uint32 tool_menu::get_color_response ( )
```

Metoda zwraca kolor wybrany przez użytkownika w oknie color selector w postaci RGBA. Metoda nie przyjmuje żadnych argumentów wywołania.

Zwraca

Uint32 Kolor wybrany przez użytkownika (format RGBA 8888).

5.31.3.3 `get_tool_response()`

```
enum tool_type tool_menu::get_tool_response ( )
```

Metoda zwraca wybór użytkownika odnośnie typu narzędzia, nie przyjmuje argumentów wywołania.

Zwraca

Narzędzie wybrane przez użytkownika.

5.31.3.4 `get_tool_type()`

```
static void tool_menu::get_tool_type (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Funkcja obsługująca sygnał kliknięcia na przycisk z wyborem nowego narzędzia. Funkcja nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
out	<i>data</i>	gpointer Dane dla funkcji.

5.31.3.5 `open_gtk_window()`

```
bool tool_menu::open_gtk_window ( ) [virtual]
```

Metoda zawiera inicjuje menu narzędzi oraz zawiera pętlę main GTK. Metoda nie przyjmuje żadnych argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) gdy operacja inicjacji menu się nie powiedzie.

Zwraca

bool Funkcja zwraca wartość true jeśli nie wystąpił w niej krytyczny błąd.

Implementuje **gtk_base_window** (str. 47).

5.31.3.6 `set_current_color()`

```
void tool_menu::set_current_color (
    Uint32 curent_color )
```

Metoda konwertuje aktualny kolor używany przez narzędzie na format przyjazny dla GTK. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>curent_color</i>	UInt32 Aktualny kolor narzędzia.
----	---------------------	----------------------------------

5.31.3.7 set_current_tool()

```
void tool_menu::set_current_tool (
    enum tool_type curent_tool )
```

Metoda ustawia aktualne typ aktualnie używanego narzędzia. Nie zwraca żadnej wartości.

Parametry

<i>curent_tool</i>	enum tool_type Typ aktualnie używanego narzędzia.
--------------------	---

5.31.3.8 trigger_color_chooser()

```
static void tool_menu::trigger_color_chooser (
    GtkWidget * widget,
    gpointer data ) [inline], [static], [private]
```

Funkcja obsługująca sygnał kliknięcia na przycisk otwierający wybór koloru. Funkcja nie zwraca żadnej wartości.

Parametry

out	<i>widget</i>	GtkWidget* Wskaźnik na okno z którego pochodził sygnał.
out	<i>data</i>	gpointer Dane dla funkcji.

5.31.4 Dokumentacja atrybutów składowych

5.31.4.1 new_color

```
UInt32 tool_menu::new_color [private]
```

Kolor wyjściowy z okna color selector

5.31.4.2 number_of_tools

```
std::size_t tool_menu::number_of_tools [private]
```

Ilość dostępnych narzędzi w programie.

5.31.4.3 RGB

```
UInt8 tool_menu::RGB[3] = {0, 0, 0} [private]
```

Kolor do wyświetlenia jako aktualny w oknie color selector.

5.31.4.4 tool_response

```
enum tool_type tool_menu::tool_response [private]
```

Wybrane przez użytkownika narzędzie.

5.31.4.5 tool_types_array

```
std::deque<struct tool_type_data> tool_menu::tool_types_array [private]
```

Kolejka, która ułatwia działanie GTK.

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/tools/ **tool_menu.hpp**
- src/tools/ **tool_menu.cpp**

5.32 Dokumentacja struktury tool_type_data

Struktura przechowująca dane o wybranym narzędziu. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

```
#include <tool_menu.hpp>
```

Atrybuty publiczne

- enum **tool_type** **type**
- enum **tool_type** * **response**

5.32.1 Opis szczegółowy

Struktura przechowująca dane o wybranym narzędziu. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

5.32.2 Dokumentacja atrybutów składowych

5.32.2.1 response

```
enum tool_type* tool_type_data::response
```

Wskaźnik zmienną przechowującą wybór użytkownika.

5.32.2.2 type

```
enum tool_type tool_type_data::type
```

Typ narzędzia.

Dokumentacja dla tej struktury została wygenerowana z pliku:

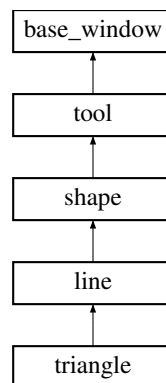
- src/tools/ **tool_menu.hpp**

5.33 Dokumentacja klasy triangle

Klasa rysująca trójkąt równoboczny.

```
#include <triangle.hpp>
```

Diagram dziedziczenia dla triangle



Metody publiczne

- **triangle ()**
Konstruktor klasy, służy do ustawienia pola type;.

Metody prywatne

- void **draw_shape** (std::size_t x_position, std::size_t y_position)
Metoda rysuje trójkąt równoboczny od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Dodatkowe Dziedziczone Składowe

5.33.1 Opis szczegółowy

Klasa rysująca trójkąt równoboczny.

5.33.2 Dokumentacja konstruktora i destruktora

5.33.2.1 triangle()

```
triangle::triangle ( )
```

Konstruktor klasy, służy do ustawienia pola type;.

5.33.3 Dokumentacja funkcji składowych

5.33.3.1 draw_shape()

```
void triangle::draw_shape (
    std::size_t x_position,
    std::size_t y_position ) [private], [virtual]
```

Metoda rysuje trójkąt równoboczny od punktu początkowego do wskazanego przez argumenty wywołania, wykorzystując metodę draw_shape klasy line. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>x_position</i>	std::size_t Składowa 'x' punktu.
in	<i>y_position</i>	std::size_t Składowa 'y' punktu.

Błąd Trójkąt "traci" równoboczną przy prawej krawędzi okna.

Błąd przy rysowaniu koła kiedy `x_position == x_begining`.

Reimplementowana z **line** (str. 53).

Dokumentacja dla tej klasy została wygenerowana z plików:

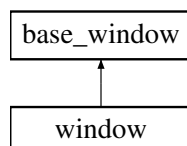
- src/tools/shapes/ **triangle.hpp**
- src/tools/shapes/ **triangle.cpp**

5.34 Dokumentacja klasy window

Klasa służąca do obsługi okna głównego programu.

```
#include <window.hpp>
```

Diagram dziedziczenia dla window



Metody publiczne

- **window** (std::size_t **window_width**=1080, std::size_t **window_height**=720)
*Konstruktor klasy. Ustawia jej pola wartościami domyślnymi, lub podanymi jako argumenty (dotyczy wysokości i szerokości okna), Alokuje również pamięć na tablicę pikseli, rysuje na niej menu, tworzy komponenty programu m.in. domyślnie narzędzie (pedzel). Konstruktor tej klasy "rzuci" wyjątki typu **paint_exception** (str. 61), w przypadku ich wystąpienia program nie może działać prawidłowo.*
- std::size_t **get_menu_height** ()
Getter wysokości menu.
- void **clicked_on_menu** (std::size_t x_position, std::size_t y_position)
*Metoda zajmująca się obsługą menu. Nie zwraca żadnej wartości. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu alokacji pamięci (podczas otwierania pliku).*
- void **set_copied_pixels** ()
Metoda kopiuje obszar roboczy tablicy pikseli (bez menu) do bufora. Pozwala na wycofanie ostatniej operacji użytkownika. Nie zwraca żadnej wartości, nie przyjmuje argumentów.
- void **use_tool** (std::size_t mouse_x, std::size_t mouse_y)
Metoda używa bieżącego narzędzia do rysowania na tablicy pikseli w miejscu o pozycji podanym jej przez argumenty wywołania. Nie zwraca żadnej wartości.
- void **resize_brush** (int difference)
Metoda zmienia rozmiar bieżącego narzędzia. Nie zwraca żadnej wartości.
- bool **is_the_tool_a_brush** ()
Dzięki tej metodzie można dowiedzieć się czy bieżące narzędzie jest pędzlem. Jest to ważne dla mechaniki rysowania. Metoda zwraca wartość typu bool, nie przyjmuje argumentów wywołania.
- bool **pressed_left_button_tool_flag** (bool change_flag=false)
Metoda informuje o stanie flagi wciśnięcia prawego przycisku myszy (1 raz) bieżącego narzędzia, pozwala zmienić jej stan. Znajomość stanu flagi jest kluczowa dla mechanizmu rysowania.
- bool **pressed_left_button_second_time_tool_flag** (bool change_flag=false)
Metoda informuje o stanie flagi wciśnięcia prawego przycisku myszy (2 raz) bieżącego narzędzia, pozwala zmienić jej stan. Znajomość stanu flagi jest kluczowa dla mechanizmu rysowania.
- void **init_sld** ()
*Metoda inicjuje bibliotekę SDL2 oraz okno programu. Metoda nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) kiedy wystąpi błąd podczas inicjowania biblioteki bądź okna.*
- void **render_window** ()
*Metoda wyświetla zawartość tablicy pikseli w oknie programu. Wykonanie operacji renderowania jest zależne od stanu flagi. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu **paint_exception** jeśli wystąpi błąd jednej z funkcji SDL2 służącej do renderowania obrazu.*
- void **file_drop** (char *file_path)
Metoda odpowiada za obsługę upuszczenia pliku w oknie programu. Metoda nie zwraca żadnej wartości.
- bool **resized_window** ()

Metoda zmienia rozmiar okna programu, zmienia parametry okna oraz kopiuje zawartość tablicy i bufora pikseli. Używana, gdy użytkownik zmieni rozmiar okna programu. Nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu `paint_exception`. Metoda może "rzucać" wyjątki w przypadku błędu alokacji pamięci lub aktualizowania okna.

- `bool resized_window (std::size_t new_width, std::size_t new_height)`

Metoda zmienia rozmiar okna programu na podstawie argumentów, zmienia parametry okna oraz kopiuje zawartość tablicy i bufora pikseli. Używana podczas wczytywania obrazów z pliku. Okno nie może być mniejsze niż 400x200 pikseli. Metoda dba o to żeby okno programu nie było mniejsze niż 400x200 pikseli oraz o spójność danych wszystkich obiektów. Metoda może "rzucać" wyjątki w przypadku błędu alokacji pamięci lub aktualizowania okna.

- `void redraw_window ()`

Metoda odpowiada za odrysowanie okna programu. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

- `~window ()`

Destruktor, zajmuje się usuwaniem zmiennych zaalokowanych dynamicznie.

Metody prywatne

- `void reset_canvas ()`

Metoda czyści obszar roboczy okna programu. Sprawdza czy użytkownik go edytował, w takim wypadku prosi go o potwierdzenie operacji. Metoda zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

- `void swap_current_and_undo_pixels ()`

Metoda podmienia zawartość bufora pikseli z tablicą pikseli okna programu. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu `paint_exception` (str. 61) w przypadku błędu alokacji pamięci.

- `void trigger_tool_menu ()`

Metoda otwiera menu do wyboru narzędzi zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

- `void trigger_file_manager_menu ()`

Metoda otwiera menu do wyboru pliku zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

- `void trigger_filter_menu ()`

Metoda otwiera menu do wyboru filtrów zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

- `void print_sdl_error (const char *text)`

Metoda służąca do wypisywania komunikatu o błędzie zrealizowana za pomocą biblioteki SDL2. Metoda nie zwraca żadnej wartości.

Atrybuty prywatne

- `std::string window_name = "Simple Paint"`
- `Uint32 * copied_pixels`
- `menu window_menu`
- `tool_menu tool_menu_widget`
- `file_manager file_manager_widget`
- `filter_menu filter_menu_widget`
- `tool * current_tool`
- `bool render = true`
- `SDL_Window * sdl_window`
- `SDL_Renderer * renderer`
- `SDL_Texture * texture`

Dodatkowe Dziedziczone Składowe

5.34.1 Opis szczegółowy

Klasa służąca do obsługi okna głównego programu.

5.34.2 Dokumentacja konstruktora i destruktora

5.34.2.1 window()

```
window::window (
    std::size_t window_width = 1080,
    std::size_t window_height = 720 )
```

Konstruktor klasy. Ustawia jej pola wartościami domyślnymi, lub podanymi jako argumenty (dotyczy wysokości i szerokości okna), Alokuje również pamięć na tablicę pikseli, rysuje na niej menu, tworzy komponenty programu m.in. domyślnie narzędzie (pedzel). Konstruktor tej klasy "rzuca" wyjątki typu **paint_exception** (str. 61), w przypadku ich wystąpienia program nie może działać prawidłowo.

Parametry

in	1080	std::size_t window_width= Szerokość okna (domyślnie 1080 pikseli).
in	720	std::size_t window_height= Wysokość okna (domyślnie 720 pikseli).

5.34.2.2 ~window()

```
window::~~window ( )
```

Destruktor, zajmuje się usuwaniem zmiennych zaalokowanych dynamicznie.

5.34.3 Dokumentacja funkcji składowych

5.34.3.1 clicked_on_menu()

```
void window::clicked_on_menu (
    std::size_t x_position,
    std::size_t y_position )
```

Metoda zajmująca się obsługą menu. Nie zwraca żadnej wartości. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu alokacji pamięci (podczas otwierania pliku).

Parametry

in	<i>x_position</i>	std::size_t Pozycja x kliknięcia myszy.
in	<i>y_position</i>	std::size_t Pozycja y kliknięcia myszy.

5.34.3.2 file_drop()

```
void window::file_drop (
    char * file_path )
```

Metoda odpowiada za obsługę upuszczenia pliku w oknie programu. Metoda nie zwraca żadnej wartości.

Parametry

in	<i>file_path</i>	char* Ścieżka do upuszczonego pliku.
----	------------------	--------------------------------------

5.34.3.3 get_menu_height()

```
std::size_t window::get_menu_height ( )
```

Getter wysokości menu.

Zwraca

std::size_t Wysokość menu okna.

5.34.3.4 init_sld()

```
void window::init_sld ( )
```

Metoda inicjuje bibliotekę SDL2 oraz okno programu. Metoda nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu **paint_exception** (str. 61) kiedy wystąpi błąd podczas inicjowania biblioteki bądź okna.

5.34.3.5 is_the_tool_a_brush()

```
bool window::is_the_tool_a_brush ( )
```

Dzięki tej metodzie można dowiedzieć się czy bieżące narzędzie jest pędzlem. Jest to ważne dla mechaniki rysowania. Metoda zwraca wartość typu bool, nie przyjmuje argumentów wywołania.

Zwraca

bool Zwraca true jeśli bieżące narzędzie jest pędzlem, w przeciwnym przypadku zwraca false.

5.34.3.6 pressed_left_button_second_time_tool_flag()

```
bool window::pressed_left_button_second_time_tool_flag (
    bool change_flag = false )
```

Metoda informuje o stanie flagi wciśnięcia prawego przycisku myszy (2 raz) bieżącego narzędzia, pozwala zmienić jej stan. Znajomość stanu flagi jest kluczowa dla mechanizmu rysowania.

Parametry

in	bool	change_flag = false Jeżeli wartość tego argumentu to true, to flaga zostanie zmieniona. Domyślna wartość to false.
----	------	--

Zwraca

bool Metoda zwraca stan flagi wciśnięcia prawego przycisku myszy po raz drugi bieżącego narzędzia.

5.34.3.7 pressed_left_button_tool_flag()

```
bool window::pressed_left_button_tool_flag (
    bool change_flag = false )
```

Metoda informuje o stanie flagi wciśnięcia prawego przycisku myszy (1 raz) bieżącego narzędzia, pozwala zmienić jej stan. Znajomość stanu flagi jest kluczowa dla mechanizmu rysowania.

Parametry

in	bool	change_flag = false Jeżeli wartość tego argumentu to true, to flaga zostanie zmieniona. Domyślna wartość to false.
----	------	--

Zwraca

bool Metoda zwraca stan flagi wciśnięcia prawego przycisku myszy bieżącego narzędzia.

5.34.3.8 print_sdl_error()

```
void window::print_sdl_error (
    const char * text ) [private]
```

Metoda służąca do wypisywania komunikatu o błędzie zrealizowana za pomocą biblioteki SDL2. Metoda nie zwraca żadnej wartości.

Parametry

text	const char*	Komunikat o błędzie do wyświetlenia w oknie.
------	-------------	--

5.34.3.9 redraw_window()

```
void window::redraw_window ( )
```

Metoda odpowiada za odrysowanie okna programu. Nie przyjmuje argumentów wywołania i nie zwraca żadnej wartości.

5.34.3.10 render_window()

```
void window::render_window ( )
```

Metoda wyświetla zawartość tablicy pikseli w oknie programu. Wykonanie operacji renderowania jest zależne od stanu flagi. Metoda nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu `paint_exception` jeśli wystąpi błąd jednej z funkcji SDL2 służącej do renderowania obrazu.

5.34.3.11 reset_canva()

```
void window::reset_canva ( ) [private]
```

Metoda czyści obszar roboczy okna programu. Sprawdza czy użytkownik go edytował, w takim wypadku prosi go o potwierdzenie operacji. Metoda zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.34.3.12 resize_brush()

```
void window::resize_brush (
    int difference )
```

Metoda zmienia rozmiar bieżącego narzędzia. Nie zwraca żadnej wartości.

Parametry

in	<i>difference</i>	int Różnica względem obecnego rozmiaru.
----	-------------------	---

5.34.3.13 resized_window() [1/2]

```
bool window::resized_window ( )
```

Metoda zmienia rozmiar okna programu, zmienia parametry okna oraz kopiuje zawartość tablicy i bufora pikseli. Używana, gdy użytkownik zmieni rozmiar okna programu. Nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątki typu `paint_exception`. Metoda może "rzucać" wyjątki w przypadku błędu alokacji pamięci lub aktualizowania okna.

Zwraca

bool Metoda zwraca true jeśli operacja się powiedzie, w przeciwnym wypadku zwraca false.

5.34.3.14 resized_window() [2/2]

```
bool window::resized_window (
    std::size_t new_width,
    std::size_t new_height )
```

Metoda zmienia rozmiar okna programu na podstawie argumentów, zmienia parametry okna oraz kopiuje zawartość tablicy i bufora pikseli. Używana podczas wczytywania obrazów z pliku. Okno nie może być mniejsze niż 400x200 pikseli. Metoda dba o to żeby okno programu nie było mniejsze niż 400x200 pikseli oraz o spójność danych wszystkich obiektów. Metoda może "rzucać" wyjątki w przypadku błędu alokacji pamięci lub aktualizowania okna.

Parametry

in	<i>new_width</i>	std::size_t Nowa szerokość okna.
in	<i>new_height</i>	std::size_t Nowa wysokość okna.

Zwraca

bool

5.34.3.15 set_coppied_pixels()

```
void window::set_coppied_pixels ( )
```

Metoda kopiuje obszar roboczy tablicy pikseli (bez menu) do bufora. Pozwala na wycofanie ostatniej operacji użytkownika. Nie zwraca żadnej wartości, nie przyjmuje argumentów.

5.34.3.16 swap_current_and_undo_pixels()

```
void window::swap_current_and_undo_pixels ( ) [private]
```

Metoda podmienia zawartość bufora pikseli z tablicą pikseli okna programu. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania. Metoda może "rzucić" wyjątek typu **paint_exception** (str. 61) w przypadku błędu alokacji pamięci.

5.34.3.17 trigger_file_manager_menu()

```
void window::trigger_file_manager_menu ( ) [private]
```

Metoda otwiera menu do wyboru pliku zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.34.3.18 trigger_filter_menu()

```
void window::trigger_filter_menu ( ) [private]
```

Metoda otwiera menu do wyboru filtrów zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.34.3.19 trigger_tool_menu()

```
void window::trigger_tool_menu ( ) [private]
```

Metoda otwiera menu do wyboru narzędzi zrealizowanym za pomocą biblioteki GTK+-2.0. Nie zwraca żadnej wartości i nie przyjmuje argumentów wywołania.

5.34.3.20 use_tool()

```
void window::use_tool (
    std::size_t mouse_x,
    std::size_t mouse_y )
```

Metoda używa bieżącego narzędzia do rysowania na tablicy pikseli w miejscu o pozycji podanym jej przez argumenty wywołania. Nie zwraca żadnej wartości.

Parametry

in	<i>mouse</i> ↔ _x	std::size_t Pozycja x kliknięcia myszy.
in	<i>mouse</i> ↔ _y	std::size_t Pozycja y kliknięcia myszy.

5.34.4 Dokumentacja atrybutów składowych

5.34.4.1 copied_pixels

```
Uint32* window::copied_pixels [private]
```

Buffer z plkselami, umożliwia realizacje operacji undo/redo.

5.34.4.2 current_tool

```
tool* window::current_tool [private]
```

Wskaźnik na bieżące narzędzie.

5.34.4.3 file_manager_widget

```
file_manager window::file_manager_widget [private]
```

Obiekt- okno menagera plików.

5.34.4.4 filter_menu_widget

```
filter_menu window::filter_menu_widget [private]
```

Obiekt- okno menu filtrów.

5.34.4.5 render

```
bool window::render = true [private]
```

Flaga renderowania okna, domyślna wartość do true.

5.34.4.6 renderer

```
SDL_Renderer* window::renderer [private]
```

Wskaźnik na obiekt biblioteki SDL2 służący do renderownia obrazu.

5.34.4.7 sdl_window

```
SDL_Window* window::sdl_window [private]
```

Wskaźnik na okno główne programu.

5.34.4.8 texture

```
SDL_Texture* window::texture [private]
```

Wskaźnik na obiekt biblioteki SDL2 zawierający teksturę okna.

5.34.4.9 tool_menu_widget

```
tool_menu window::tool_menu_widget [private]
```

Obiekt- okno menu z narzędziami.

5.34.4.10 window_menu

```
menu window::window_menu [private]
```

Obiekt- menu programu.

5.34.4.11 window_name

```
std::string window::window_name = "Simple Paint" [private]
```

Pole typu std::string zawierające nazwę okna programu.

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/window/ **window.hpp**
- src/window/ **window.cpp**

Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku src/base_window.hpp

Plik z definicją klasy **base_window** (str. 11).

```
#include <SDL2/SDL.h>
#include <iostream>
#include "paint_exception.hpp"
```

Komponenty

- class **base_window**

Klasa bazowa dla elementów okna programu oraz narzędzi.

6.1.1 Opis szczegółowy

Plik z definicją klasy **base_window** (str. 11).

6.2 Dokumentacja pliku src/file_manager/file_manager.cpp

Plik z definicjami metod klasy **file_manager** (str. 31).

```
#include "file_manager.hpp"
```

6.2.1 Opis szczegółowy

Plik z definicjami metod klasy **file_manager** (str. 31).

6.3 Dokumentacja pliku src/file_manager/file_manager.hpp

Plik definicję klasy **file_manager** (str. 31) oraz typu wyliczeniowego `file_manager_mode`.

```
#include "../base_window.hpp"
#include "../gtk_base_window.hpp"
#include "image_container/image_container.hpp"
#include "image_container/bmp_image_container.hpp"
#include "image_container/ppm_image_container.hpp"
#include "image_container/png_image_container.hpp"
```

Komponenty

- class **file_manager**
Klasa mangera plików.

Wyliczenia

- enum **file_manager_mode** { **OPEN_FILE**, **SAVE_FILE**, **QUICK_SAVE**, **NONE** }
Typ wyliczeniowy pozwalający określić tryb pracy menadżera plików.

6.3.1 Opis szczegółowy

Plik definicję klasy **file_manager** (str. 31) oraz typu wyliczeniowego `file_manager_mode`.

6.3.2 Dokumentacja typów wyliczanych

6.3.2.1 file_manager_mode

```
enum file_manager_mode
```

Typ wyliczeniowy pozwalający określić tryb pracy menadżera plików.

Wartości wyliczeń

OPEN_FILE	Otwieranie plików.
SAVE_FILE	Zapis plików.
QUICK_SAVE	Szybki zapis.
NONE	Nie podejmuj akcji.

6.4 Dokumentacja pliku src/file_manager/image_container/bmp_image_container.cpp

Plik z definicjami metod klasy **bmp_image_container** (str. 20).

```
#include "bmp_image_container.hpp"  
#include <iostream>
```

6.4.1 Opis szczegółowy

Plik z definicjami metod klasy **bmp_image_container** (str. 20).

6.5 Dokumentacja pliku src/file_manager/image_container/bmp_image_container.hpp

Plik z definicją klasy **bmp_image_container** (str. 20).

```
#include "image_container.hpp"
```

Komponenty

- class **bmp_image_container**
Kontener dla plików bmp.
- struct **bmp_image_container::bmp_file_info**
Struktura pomocnicza przechowująca informacje o obrazach bmp.

6.5.1 Opis szczegółowy

Plik z definicją klasy **bmp_image_container** (str. 20).

6.6 Dokumentacja pliku src/file_manager/image_container/image_container.cpp

Plik z definicjami metod klasy **image_container** (str. 48).

```
#include "image_container.hpp"
```

6.6.1 Opis szczegółowy

Plik z definicjami metod klasy **image_container** (str. 48).

6.7 Dokumentacja pliku src/file_manager/image_container/image_container.hpp

Plik z definicją klasy **image_container** (str. 48) oraz typu wyliczeniowego **file_type**.

```
#include <string>
#include <fstream>
#include <algorithm>
#include "../paint_exception.hpp"
```

Komponenty

- class **image_container**
Klasa bazowa dla wszystkich typów kontenerów obrazu.
- struct **image_container::basic_image_info**
Struktura przechowująca podstawowe informacje o przechowywanym obrazie.
- struct **image_container::color**
Struktura przechowująca informacje o składowych koloru.

6.7.1 Opis szczegółowy

Plik z definicją klasy **image_container** (str. 48) oraz typu wyliczeniowego **file_type**.

6.8 Dokumentacja pliku src/file_manager/image_container/png_image_container.cpp

Plik definicję metod klasy **png_image_container** (str. 68).

```
#include "png_image_container.hpp"
```

6.8.1 Opis szczegółowy

Plik definicję metod klasy **png_image_container** (str. 68).

6.9 Dokumentacja pliku src/file_manager/image_container/png_image_container.hpp

Plik definicję klasy **ppm_image_container** (str. 71).

```
#include "image_container.hpp"
#include <png++/png.hpp>
```

Komponenty

- class **png_image_container**
Kontener dla plików png.

6.9.1 Opis szczegółowy

Plik definicję klasy **ppm_image_container** (str. 71).

6.10 Dokumentacja pliku src/file_manager/image_container/ppm_image_container.cpp

Plik z definicjami metod klasy **ppm_image_container** (str. 71).

```
#include "ppm_image_container.hpp"
```

6.10.1 Opis szczegółowy

Plik z definicjami metod klasy **ppm_image_container** (str. 71).

6.11 Dokumentacja pliku src/file_manager/image_container/ppm_image_container.hpp

Plik definicję klasy **ppm_image_container** (str. 71).

```
#include "image_container.hpp"
```

Komponenty

- class **ppm_image_container**
Kontener dla plików ppm.
- struct **ppm_image_container::ppm_file_info**
Struktura pomocnicza przechowująca informacje o obrazach ppm.

6.11.1 Opis szczegółowy

Plik definicję klasy **ppm_image_container** (str. 71).

6.12 Dokumentacja pliku src/filters/basic_filter.cpp

Plik z definicjami metod klasy **basic_filter** (str. 13).

```
#include "basic_filter.hpp"
```

6.12.1 Opis szczegółowy

Plik z definicjami metod klasy **basic_filter** (str. 13).

6.13 Dokumentacja pliku src/filters/basic_filter.hpp

Plik z definicją klasy `base_filter` oraz typu wyliczeniowego `filter_type`.

```
#include "../base_window.hpp"
#include <gtk/gtk.h>
```

Komponenty

- class **basic_filter**

Klasa stanowiąca bazę dla klas realizujących filtry.

Wyliczenia

- enum **filter_type** {
 GRAYSCALE, **SEPIA**, **COLOR_INVERSE**, **PIXELATE**,
 NONE_FILTER = -1 }

Typ wyliczeniowy określający jaki filter został zabrany.

6.13.1 Opis szczegółowy

Plik z definicją klasy `base_filter` oraz typu wyliczeniowego `filter_type`.

6.13.2 Dokumentacja typów wyliczanych

6.13.2.1 filter_type

```
enum filter_type
```

Typ wyliczeniowy określający jaki filter został zabrany.

Wartości wyliczeń

GRAYSCALE	Skala szarości.
SEPIA	Sepia.
COLOR_INVERSE	Odwracanie koloru.
PIXELATE	Pikselizacja obrazu.
NONE_FILTER	Nie wybrano żadnego filtra

6.14 Dokumentacja pliku src/filters/color_inverse.cpp

Plik z definicjami metod klasy **color_inverse** (str. 25).

```
#include "color_inverse.hpp"
```

6.14.1 Opis szczegółowy

Plik z definicjami metod klasy **color_inverse** (str. 25).

6.15 Dokumentacja pliku src/filters/color_inverse.hpp

Plik z definicją klasy **color_inverse** (str. 25).

```
#include "basic_filter.hpp"
```

Komponenty

- class **color_inverse**
Klasa filtra odwracającego kolory obrazu.

6.15.1 Opis szczegółowy

Plik z definicją klasy **color_inverse** (str. 25).

6.16 Dokumentacja pliku src/filters/filter_menu.cpp

Plik z definicjami metod klasy **filter_menu** (str. 37).

```
#include "filter_menu.hpp"
```

6.16.1 Opis szczegółowy

Plik z definicjami metod klasy **filter_menu** (str. 37).

6.17 Dokumentacja pliku src/filters/filter_menu.hpp

Plik definicję klasy **filter_menu** (str. 37).

```
#include "../gtk_base_window.hpp"  
#include "basic_filter.hpp"  
#include <deque>
```

Komponenty

- struct **filter_type_data**

Struktura przechowująca dane o wybranym filtrze. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

- class **filter_menu**

Klasa menu do wyboru filtrów.

6.17.1 Opis szczegółowy

Plik definicję klasy **filter_menu** (str. 37).

6.18 Dokumentacja pliku src/filters/grayscale.cpp

Plik z definicjami metod klasy grayscale.

```
#include "grayscale.hpp"
```

6.18.1 Opis szczegółowy

Plik z definicjami metod klasy grayscale.

6.19 Dokumentacja pliku src/filters/grayscale.hpp

Plik z definicją klasy grayscale.

```
#include "basic_filter.hpp"
```

Komponenty

- class **grayscale**

Klasa filtra zmieniająca obraz na czarno-biały.

6.19.1 Opis szczegółowy

Plik z definicją klasy grayscale.

6.20 Dokumentacja pliku src/filters/pixelate.cpp

Plik z definicjami metod klasy pixelate.

```
#include "pixelate.hpp"
```


6.20.1 Opis szczegółowy

Plik z definicjami metod klasy pixelate.

6.21 Dokumentacja pliku src/filters/pixelate.hpp

Plik z definicją klasy pixelate.

```
#include <cmath>
#include "basic_filter.hpp"
```

Komponenty

- class **pixelate**
Klasa filtra pikselizującego obraz.

6.21.1 Opis szczegółowy

Plik z definicją klasy pixelate.

6.22 Dokumentacja pliku src/filters/sepia.cpp

Plik z definicjami metod klasy sepia.

```
#include "sepia.hpp"
```

6.22.1 Opis szczegółowy

Plik z definicjami metod klasy sepia.

6.23 Dokumentacja pliku src/filters/sepia.hpp

Plik z definicją klasy sepia.

```
#include "basic_filter.hpp"
```

Komponenty

- class **sepia**
Klasa filtra sepia.

6.23.1 Opis szczegółowy

Plik z definicją klasy `sepia`.

6.24 Dokumentacja pliku `src/gtk_base_window.hpp`

Plik definicję klasy `gtk_base_window` (str. 45).

```
#include <gtk/gtk.h>
#include <fstream>
#include <string>
#include <deque>
#include "paint_exception.hpp"
```

Komponenty

- class `gtk_base_window`
Klasa bazowa dla okien używających biblioteki GTK.

6.24.1 Opis szczegółowy

Plik definicję klasy `gtk_base_window` (str. 45).

6.25 Dokumentacja pliku `src/main.cpp`

Plik zawiera funkcję `main`.

```
#include <deque>
#include <cstring>
#include <iostream>
#include <gtk/gtk.h>
#include <SDL2/SDL.h>
#include "window/menu.hpp"
#include "window/window.hpp"
#include "paint_exception.hpp"
```

Funkcje

- int `main` (int argc, char **argv)

6.25.1 Opis szczegółowy

Plik zawiera funkcję `main`.

6.25.2 Dokumentacja funkcji

6.25.2.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.26 Dokumentacja pliku src/paint_exception.cpp

Plik z definicjami metod klasy **paint_exception** (str. 61).

```
#include "paint_exception.hpp"
```

6.26.1 Opis szczegółowy

Plik z definicjami metod klasy **paint_exception** (str. 61).

6.27 Dokumentacja pliku src/paint_exception.hpp

Plik z definicją klasy **paint_exception** (str. 61).

```
#include <string>
```

Komponenty

- class **paint_exception**
Klasa do obsługi wyjątków programu.

6.27.1 Opis szczegółowy

Plik z definicją klasy **paint_exception** (str. 61).

6.28 Dokumentacja pliku src/tools/brushes/brush.cpp

Plik z definicjami metod klasy brush.

```
#include "brush.hpp"
```

6.28.1 Opis szczegółowy

Plik z definicjami metod klasy brush.

6.29 Dokumentacja pliku src/tools/brushes/brush.hpp

Plik z definicją klasy brush.

```
#include "../tool.hpp"
#include <SDL2/SDL.h>
#include <iostream>
```

Komponenty

- class **brush**

Klasa narzędzia pędzel.

6.29.1 Opis szczegółowy

Plik z definicją klasy brush.

6.30 Dokumentacja pliku src/tools/brushes/spray.cpp

Plik z definicjami metod klasy spray.

```
#include "spray.hpp"
```

6.30.1 Opis szczegółowy

Plik z definicjami metod klasy spray.

6.31 Dokumentacja pliku src/tools/brushes/spray.hpp

Plik z definicją klasy spray.

```
#include "brush.hpp"
#include <random>
#include <chrono>
```

Komponenty

- class **spray**

Klasa narzędzia sprej.

6.31.1 Opis szczegółowy

Plik z definicją klasy **spray**.

6.32 Dokumentacja pliku src/tools/brushes/star_brush.cpp

Plik z definicjami metod klasy **star_brush** (str. 83).

```
#include "star_brush.hpp"
```

6.32.1 Opis szczegółowy

Plik z definicjami metod klasy **star_brush** (str. 83).

6.33 Dokumentacja pliku src/tools/brushes/star_brush.hpp

Plik z definicją klasy **star_brush** (str. 83).

```
#include "brush.hpp"
```

Komponenty

- class **star_brush**

Klasa narzędzia pędzla w kształcie plusa.

6.33.1 Opis szczegółowy

Plik z definicją klasy **star_brush** (str. 83).

6.34 Dokumentacja pliku src/tools/color_picker.cpp

Plik z definicjami metod klasy **color_picker** (str. 26).

```
#include "color_picker.hpp"
```

6.34.1 Opis szczegółowy

Plik z definicjami metod klasy **color_picker** (str. 26).

Błąd Użycie color pickera jest traktowane jako operacja "możliwa" do cofnięcia. Nie powoduje to większych problemów, poza tym, że nie da się wycofać poprzedzającej użycie narzędzia operacji.

6.35 Dokumentacja pliku src/tools/color_picker.hpp

Plik z definicją klasy **color_picker** (str. 26).

```
#include "tool.hpp"
```

Komponenty

- class **color_picker**

Definicja klasy narzędzia pobierania koloru z obszaru rysowania.

6.35.1 Opis szczegółowy

Plik z definicją klasy **color_picker** (str. 26).

6.36 Dokumentacja pliku src/tools/flood_fill.cpp

Plik z definicjami metod klasy **flood_fill** (str. 40).

```
#include "flood_fill.hpp"
```

6.36.1 Opis szczegółowy

Plik z definicjami metod klasy **flood_fill** (str. 40).

6.37 Dokumentacja pliku src/tools/flood_fill.hpp

Plik z definicją klasy **flood_fill** (str. 40).

```
#include "tool.hpp"
```

Komponenty

- class **flood_fill**

Klasa służąca do wypełniania obszaru kolorem.

6.37.1 Opis szczegółowy

Plik z definicją klasy **flood_fill** (str. 40).

6.38 Dokumentacja pliku src/tools/shapes/ellipse.cpp

Plik z definicjami metod klasy ellipse.

```
#include "ellipse.hpp"
```

6.38.1 Opis szczegółowy

Plik z definicjami metod klasy ellipse.

6.39 Dokumentacja pliku src/tools/shapes/ellipse.hpp

Plik z definicją klasy ellipse.

```
#include "shape.hpp"
```

Komponenty

- class **ellipse**

Klasa służąca do rysowania elipsy.

6.39.1 Opis szczegółowy

Plik z definicją klasy ellipse.

6.40 Dokumentacja pliku src/tools/shapes/line.cpp

Plik z definicjami metod klasy line.

```
#include "line.hpp"
```

6.40.1 Opis szczegółowy

Plik z definicjami metod klasy line.

6.41 Dokumentacja pliku src/tools/shapes/line.hpp

Plik z definicją klasy line.

```
#include "shape.hpp"
```

Komponenty

- class **line**
Klasa służąca do rysowania prostej linii.

6.41.1 Opis szczegółowy

Plik z definicją klasy line.

6.42 Dokumentacja pliku src/tools/shapes/rectangle.cpp

Plik z definicjami metod klasy rectangle.

```
#include "rectangle.hpp"
```

6.42.1 Opis szczegółowy

Plik z definicjami metod klasy rectangle.

6.43 Dokumentacja pliku src/tools/shapes/rectangle.hpp

Plik z definicją klasy rectangle.

```
#include <cmath>  
#include "line.hpp"
```

Komponenty

- class **rectangle**
Klasa rysująca prostokąt.

6.43.1 Opis szczegółowy

Plik z definicją klasy rectangle.

6.44 Dokumentacja pliku src/tools/shapes/shape.cpp

Plik z definicjami metod klasy shape.

```
#include "shape.hpp"
```

6.44.1 Opis szczegółowy

Plik z definicjami metod klasy shape.

6.45 Dokumentacja pliku src/tools/shapes/shape.hpp

Plik z definicją klasy shape.

```
#include "../tool.hpp"
```

Komponenty

- class **shape**

Klasa bazowa dla klas rysujących figury geometryczne.

6.45.1 Opis szczegółowy

Plik z definicją klasy shape.

6.46 Dokumentacja pliku src/tools/shapes/triangle.cpp

Plik z definicją klasy triangle.

```
#include "triangle.hpp"
```

6.46.1 Opis szczegółowy

Plik z definicją klasy triangle.

6.47 Dokumentacja pliku src/tools/shapes/triangle.hpp

Plik z definicją metod klasy triangle.

```
#include <cmath>
#include "line.hpp"
```

Komponenty

- class **triangle**

Klasa rysująca trójkąt równoboczny.

6.47.1 Opis szczegółowy

Plik z definicją metod klasy triangle.

6.48 Dokumentacja pliku src/tools/tool.cpp

Plik z definicjami metod klasy tool.

```
#include "tool.hpp"
```

6.48.1 Opis szczegółowy

Plik z definicjami metod klasy tool.

6.49 Dokumentacja pliku src/tools/tool.hpp

Plik z definicją klasy tool oraz typu wyliczeniowego tool_type.

```
#include <SDL2/SDL.h>
#include <gtk/gtk.h>
#include <iostream>
#include <string>
#include <deque>
#include <cmath>
#include "../base_window.hpp"
#include "../paint_exception.hpp"
```

Komponenty

- class **tool**

Klasa bazowa dla wszystkich typów narzędzi.

Wyliczenia

```
enum tool_type {
    BRUSH, STAR_BRUSH, SPRAY, FLOOD_FILL,
    COLOR_PICKER, ELLIPSE, LINE, RECTANGLE,
    TRIAGNLE }

```

Typ wyliczeniowy służący do określenia narzędzia.

6.49.1 Opis szczegółowy

Plik z definicją klasy tool oraz typu wyliczeniowego tool_type.

6.49.2 Dokumentacja typów wyliczanych

6.49.2.1 tool_type

```
enum tool_type
```

Typ wyliczeniowy służący do określenia narzędzia.

Wartości wyliczeń

BRUSH	Pędzel
STAR_BRUSH	Pędzel w kształcie plusa.
SPRAY	Pędzel- sprej.
FLOOD_FILL	Wypełnianie obszaru kolorem
COLOR_PICKER	Próbnik koloru
ELLIPSE	Elipsa z możliwością rysowania koła.
LINE	Linia.
RECTANGLE	Prostokąt z możliwością rysowania kwadratu.
TRIAGNLE	Trójkąt równoboczny

6.50 Dokumentacja pliku src/tools/tool_menu.cpp

Plik z definicjami metod klasy **tool_menu** (str. 91).

```
#include "tool_menu.hpp"
```

6.50.1 Opis szczegółowy

Plik z definicjami metod klasy **tool_menu** (str. 91).

6.51 Dokumentacja pliku src/tools/tool_menu.hpp

Plik z definicją klasy **tool_menu** (str. 91).

```
#include <SDL2/SDL.h>
#include <deque>
#include "tool.hpp"
#include "../gtk_base_window.hpp"
```

Komponenty

- struct **tool_type_data**

Struktura przechowująca dane o wybranym narzędziu. Umożliwia korzystanie z prostych sygnałów GTK bez użycia zmiennych globalnych.

- class **tool_menu**

Klasa związana z wyświetlaniem menu wyboru narzędzi.

6.51.1 Opis szczegółowy

Plik z definicją klasy **tool_menu** (str. 91).

6.52 Dokumentacja pliku src/window/menu.cpp

Plik z definicjami metod klasy menu.

```
#include "menu.hpp"
```

6.52.1 Opis szczegółowy

Plik z definicjami metod klasy menu.

6.53 Dokumentacja pliku src/window/menu.hpp

Plik z definicją klasy menu oraz typu wyliczeniowego menu_buttons.

```
#include <SDL2/SDL.h>
#include <iostream>
#include <fstream>
#include <string>
#include <deque>
#include "../base_window.hpp"
#include "../tools/tool.hpp"
```

Komponenty

- class **menu**

Klasa służąca do rysowania menu oraz obsługi kliknięć w jego obrębie.

Wyliczenia

- enum **menu_buttons** {
 NEW_FILE, UNDO, OPEN_SAVE_FILE, TOOL_CHOOSER,
 FILTERS, TOOL_PROPERITES }

Typ wyliczeniowy służący do określenia przycisku z menu. Używany do określenia, który przycisk z menu został wybrany oraz do wczytywania ich etykiet.

6.53.1 Opis szczegółowy

Plik z definicją klasy menu oraz typu wyliczeniowego menu_buttons.

6.53.2 Dokumentacja typów wyliczanych

6.53.2.1 menu_buttons

enum **menu_buttons**

Typ wyliczeniowy służący do określenia przycisku z menu. Używany do określenia, który przycisk z menu został wybrany oraz do wczytywania ich etykiet.

Wartości wyliczeń

NEW_FILE	Resetowanie obszaru roboczego.
UNDO	Operacja undo/redo.
OPEN_SAVE_FILE	Menu dotyczące operacji na plikach.
TOOL_CHOOSER	Menu wyboru narzędzi.
FILTERS	Menu z filtrami.
TOOL_PROPERITES	Menu z opcjami aktualnie używanego narzędzia.

6.54 Dokumentacja pliku src/window/window.cpp

Plik z definicjami metod klasy window.

```
#include "window.hpp"
```

6.54.1 Opis szczegółowy

Plik z definicjami metod klasy window.

6.55 Dokumentacja pliku src/window/window.hpp

Plik z definicją klasy window.

```
#include <SDL2/SDL.h>
#include <cstring>
#include <string>
#include <iostream>
#include <fstream>
#include <deque>
#include "../base_window.hpp"
#include "menu.hpp"
#include "../file_manager/file_manager.hpp"
#include "../tools/tool.hpp"
#include "../tools/brushes/brush.hpp"
#include "../tools/brushes/star_brush.hpp"
#include "../tools/brushes/spray.hpp"
#include "../tools/flood_fill.hpp"
#include "../tools/shapes/line.hpp"
#include "../tools/color_picker.hpp"
#include "../tools/shapes/ellipse.hpp"
#include "../tools/shapes/rectangle.hpp"
#include "../tools/shapes/triangle.hpp"
#include "../tools/tool_menu.hpp"
#include "../filters/filter_menu.hpp"
#include "../filters/grayscale.hpp"
#include "../filters/sepia.hpp"
#include "../filters/color_inverse.hpp"
#include "../filters/pixelate.hpp"
```

Komponenty

- class **window**

Klasa służąca do obsługi okna głównego programu.

6.55.1 Opis szczegółowy

Plik z definicją klasy window.

Skorowidz

- ~base_window
 - base_window, 12
- ~bmp_image_container
 - bmp_image_container, 21
- ~file_manager
 - file_manager, 32
- ~gtk_base_window
 - gtk_base_window, 46
- ~image_container
 - image_container, 50
- ~menu
 - menu, 56
- ~paint_exception
 - paint_exception, 64
- ~png_image_container
 - png_image_container, 70
- ~ppm_image_container
 - ppm_image_container, 73
- ~tool
 - tool, 86
- ~tool_menu
 - tool_menu, 93
- ~window
 - window, 101
- ARGB_8888_color
 - image_container, 49
- add_color_to_queue
 - menu, 56
- alpha
 - image_container::color, 24
- base_window, 11
 - ~base_window, 12
 - menu_height, 12
 - window_area, 12
 - window_height, 12
 - window_width, 12
- basic_filter, 13
 - delete_event, 13
 - destroy, 14
 - do_filter, 14
 - open_fiter_properties, 14
 - type, 16
 - update_filter_information, 14
- basic_filter.hpp
 - filter_type, 114
- bits_per_color
 - ppm_image_container::ppm_file_info, 71
- bits_per_pixel
 - bmp_image_container::bmp_file_info, 18
- blue
 - image_container::color, 24
- bmp_image_container, 20
 - ~bmp_image_container, 21
 - bmp_image_container, 20, 21
 - save_to_file, 21
- bmp_image_container::bmp_file_info, 17
 - bits_per_pixel, 18
 - compression, 18
 - header_size, 18
 - height, 18
 - horizontal_resolution, 18
 - image_size, 18
 - number_of_color_in_palette, 18
 - number_of_important_color, 19
 - planes, 19
 - size, 19
 - vertical_resolution, 19
 - width, 19
- brush, 22
 - brush, 22
 - is_a_brush, 23
 - use_tool, 23
- brush_size
 - tool, 90
- COLOR_BUFFER_MAX_SIZE
 - menu, 59
- calculate_tolerance
 - flood_fill, 42
- cell_size
 - pixelate, 68
- clicked_on_menu
 - window, 101
- color_buffer
 - menu, 59
- color_inverse, 25
 - color_inverse, 25
 - do_filter, 25
 - open_fiter_properties, 26
- color_picker, 26
 - color_picker, 27
 - isBrush, 27
 - use_tool, 27
- colors_to_print
 - menu, 59
- compression
 - bmp_image_container::bmp_file_info, 18
- convert_from_Uint8

- tool_menu, 93
- copied_pixels
 - shape, 79
 - window, 106
- current_color
 - tool, 90
- current_file_type
 - file_manager, 36
- current_mode
 - file_manager, 36
- current_tool
 - window, 107
- delete_event
 - basic_filter, 13
 - gtk_base_window, 46
 - tool, 86
- destroy
 - basic_filter, 14
 - gtk_base_window, 47
 - tool, 87
- do_filter
 - basic_filter, 14
 - color_inverse, 25
 - grayscale, 45
 - pixelate, 67
 - sepia, 76
- dots_ratio
 - spray, 82
- draw_shape
 - ellipse, 29
 - line, 53
 - rectangle, 74
 - shape, 78
 - triangle, 98
- ellipse, 28
 - draw_shape, 29
 - ellipse, 29
 - is_circle, 30
 - open_tool_properties, 29
 - PaintPixel, 30
 - set_cicle_flag, 30
- exception_string
 - paint_exception, 65
- exception_strings
 - paint_exception, 65
- exception_type
 - paint_exception, 62
- failed_to_load_resources
 - gtk_base_window, 27
- file_drop
 - window, 102
- file_manager, 31
 - ~file_manager, 32
 - current_file_type, 36
 - current_mode, 36
 - file_manager, 32
 - file_path, 36
 - get_file_manager_mode, 32
 - get_file_path, 33
 - get_image_dimentions, 33
 - image, 36
 - load_image, 33
 - open_gtk_window, 34
 - open_open_file_widget, 34
 - open_save_file_widget, 34
 - print_image, 35
 - quick_save_button_clicked, 35
 - reset_state, 35
 - save_window_to_image, 35
 - update_file_manager_information, 35
 - working_on_file, 36
- file_manager.hpp
 - file_manager_mode, 110
- file_manager_mode
 - file_manager.hpp, 110
- file_manager_widget
 - window, 107
- file_path
 - file_manager, 36
- file_type
 - image_container, 49
- filter_menu, 37
 - filter_menu, 38
 - filter_response, 39
 - filter_types, 39
 - get_filter_response, 38
 - get_filter_type, 38
 - number_of_filters, 39
 - open_gtk_window, 39
- filter_menu_widget
 - window, 107
- filter_response
 - filter_menu, 39
- filter_type
 - basic_filter.hpp, 114
- filter_type_data, 39
 - response, 40
 - type, 40
- filter_types
 - filter_menu, 39
- flood_fill, 40
 - calculate_tolerance, 42
 - flood_fill, 41
 - isBrush, 42
 - open_tool_properties, 42
 - set_confirmed_flag, 42
 - tolerance_ratio, 43
 - use_tool, 43
 - value_changed, 43
- generator
 - spray, 82
- get_brush_size
 - tool, 87
- get_clicked_button

- menu, 56
- get_color_response
 - tool_menu, 93
- get_current_color
 - tool, 87
- get_file_manager_mode
 - file_manager, 32
- get_file_path
 - file_manager, 33
- get_file_type
 - image_container, 50
- get_filter_response
 - filter_menu, 38
- get_filter_type
 - filter_menu, 38
- get_image_dimentions
 - file_manager, 33
- get_image_info
 - image_container, 50
- get_menu_height
 - window, 102
- get_tool_response
 - tool_menu, 93
- get_tool_type
 - tool, 87
 - tool_menu, 94
- grayscale, 44
 - do_filter, 45
 - grayscale, 44
 - open_fiter_properties, 45
- green
 - image_container::color, 24
- gtk_base_window, 45
 - ~gtk_base_window, 46
 - delete_event, 46
 - destroy, 47
 - failed_to_load_resources, 47
 - open_gtk_window, 47
 - resource_file_path, 47
 - resource_strings, 48
- header_size
 - bmp_image_container::bmp_file_info, 18
- height
 - bmp_image_container::bmp_file_info, 18
 - image_container::basic_image_info, 16
 - ppm_image_container::ppm_file_info, 71
- horizontal_resolution
 - bmp_image_container::bmp_file_info, 18
- image
 - file_manager, 36
- image_container, 48
 - ~image_container, 50
 - ARGB_8888_color, 49
 - file_type, 49
 - get_file_type, 50
 - get_image_info, 50
 - image_container, 50
 - image_info, 52
 - image_path, 51
 - image_pixel_array, 52
 - operator[], 51
 - save_to_file, 51
 - set_pixels_form_array, 51
- image_container::basic_image_info, 16
 - height, 16
 - image_type, 17
 - number_of_colors, 17
 - width, 17
- image_container::color, 24
 - alpha, 24
 - blue, 24
 - green, 24
 - red, 24
- image_info
 - image_container, 52
- image_path
 - image_container, 51
- image_pixel_array
 - image_container, 52
- image_size
 - bmp_image_container::bmp_file_info, 18
- image_type
 - image_container::basic_image_info, 17
- init_sld
 - window, 102
- is_a_brush
 - brush, 23
 - tool, 88
- is_brush
 - shape, 78
- is_circle
 - ellipse, 30
- is_in_buttons_area
 - menu, 57
- is_in_color_menu
 - menu, 57
- is_square
 - rectangle, 75
- is_the_tool_a_brush
 - window, 102
- isBrush
 - color_picker, 27
 - flood_fill, 42
- line, 52
 - draw_shape, 53
 - line, 53
- load_image
 - file_manager, 33
- main
 - main.cpp, 119
- main.cpp
 - main, 119
- menu, 54
 - ~menu, 56

- add_color_to_queue, 56
- COLOR_BUFFER_MAX_SIZE, 59
- color_buffer, 59
- colors_to_print, 59
- get_clicked_button, 56
- is_in_buttons_area, 57
- is_in_color_menu, 57
- menu, 55
- menu_buttons_images_paths, 59
- menu_buttons_labels, 60
- NUMBER_OF_BUTTONS, 60
- NUMBER_OF_LEFT_ELEMENTS, 60
- NUMBER_OF_TOOLS, 60
- print_current_color, 57
- print_menu, 57
- tool_labels, 60
- tool_prt, 60
- tools_images_paths, 60
- update_brush_size_state, 58
- update_colors, 58
- update_tool_pointer, 58
- update_tool_state, 58
- update_window_information, 59
- menu.hpp
 - menu_buttons, 129
- menu_buttons
 - menu.hpp, 129
- menu_buttons_images_paths
 - menu, 59
- menu_buttons_labels
 - menu, 60
- menu_height
 - base_window, 12
- NUMBER_OF_BUTTONS
 - menu, 60
- NUMBER_OF_LEFT_ELEMENTS
 - menu, 60
- NUMBER_OF_TOOLS
 - menu, 60
- new_color
 - tool_menu, 95
- number_of_color_in_palette
 - bmp_image_container::bmp_file_info, 18
- number_of_colors
 - image_container::basic_image_info, 17
- number_of_exceptions
 - paint_exception, 65
- number_of_filters
 - filter_menu, 39
- number_of_important_color
 - bmp_image_container::bmp_file_info, 19
- number_of_tools
 - tool_menu, 95
- open_fiter_properties
 - basic_filter, 14
 - color_inverse, 26
 - grayscale, 45
 - pixelate, 67
 - sepia, 77
- open_gtk_window
 - file_manager, 34
 - filter_menu, 39
 - gtk_base_window, 47
 - tool_menu, 94
- open_open_file_widget
 - file_manager, 34
- open_save_file_widget
 - file_manager, 34
- open_tool_properties
 - ellipse, 29
 - flood_fill, 42
 - rectangle, 75
 - spray, 81
 - tool, 88
- operator=
 - tool, 88
- operator[]
 - image_container, 51
- paint_exception, 61
 - ~paint_exception, 64
 - exception_string, 65
 - exception_strings, 65
 - exception_type, 62
 - number_of_exceptions, 65
 - paint_exception, 63, 64
 - what, 64
- PaintPixel
 - ellipse, 30
 - spray, 81
 - star_brush, 84
 - tool, 88
- pixelate, 66
 - cell_size, 68
 - do_filter, 67
 - open_fiter_properties, 67
 - pixelate, 67
 - set_confirmed_flag, 67
 - value_changed, 68
- planes
 - bmp_image_container::bmp_file_info, 19
- png_image_container, 68
 - ~png_image_container, 70
 - png_image_container, 69
 - save_to_file, 70
- ppm_image_container, 71
 - ~ppm_image_container, 73
 - ppm_image_container, 72
 - save_to_file, 73
- ppm_image_container::ppm_file_info, 70
 - bits_per_color, 71
 - height, 71
 - width, 71
- pressed_left_button
 - tool, 91
- pressed_left_button_second_time

- tool, 91
- pressed_left_button_second_time_tool_flag
 - window, 102
- pressed_left_button_tool_flag
 - window, 103
- print_current_color
 - menu, 57
- print_image
 - file_manager, 35
- print_menu
 - menu, 57
- print_sdl_error
 - window, 103
- quick_save_button_clicked
 - file_manager, 35
- RGB
 - tool_menu, 95
- rectangle, 73
 - draw_shape, 74
 - is_square, 75
 - open_tool_properties, 75
 - rectangle, 74
 - set_square_flag, 75
- red
 - image_container::color, 24
- redraw_window
 - window, 104
- render
 - window, 107
- render_window
 - window, 104
- renderer
 - window, 107
- reset_canva
 - window, 104
- reset_state
 - file_manager, 35
- resize_brush
 - window, 104
- resize_tool_size
 - tool, 89
- resized_window
 - window, 104, 105
- resource_file_path
 - gtk_base_window, 47
- resource_strings
 - gtk_base_window, 48
- response
 - filter_type_data, 40
 - tool_type_data, 96
- save_to_file
 - bmp_image_container, 21
 - image_container, 51
 - png_image_container, 70
 - ppm_image_container, 73
- save_window_to_image
 - file_manager, 35
- sdl_window
 - window, 107
- sepia, 76
 - do_filter, 76
 - open_fiter_properties, 77
 - sepia, 76
- set_cicle_flag
 - ellipse, 30
- set_confirmed_flag
 - flood_fill, 42
 - pixelate, 67
 - spray, 82
- set_coppied_pixels
 - window, 105
- set_current_color
 - tool_menu, 94
- set_current_tool
 - tool_menu, 95
- set_pixels_form_array
 - image_container, 51
- set_square_flag
 - rectangle, 75
- shape, 77
 - copied_pixels, 79
 - draw_shape, 78
 - is_brush, 78
 - use_tool, 79
 - x_begining, 79
 - y_begining, 79
- size
 - bmp_image_container::bmp_file_info, 19
- spray, 80
 - dots_ratio, 82
 - generator, 82
 - open_tool_properties, 81
 - PaintPixel, 81
 - set_confirmed_flag, 82
 - spray, 81
 - value_changed, 82
- src/base_window.hpp, 109
- src/file_manager/file_manager.cpp, 109
- src/file_manager/file_manager.hpp, 110
- src/file_manager/image_container/bmp_image_↵
container.cpp, 111
- src/file_manager/image_container/bmp_image_↵
container.hpp, 111
- src/file_manager/image_container/image_container.↵
cpp, 111
- src/file_manager/image_container/image_container.↵
hpp, 112
- src/file_manager/image_container/png_image_↵
container.cpp, 112
- src/file_manager/image_container/png_image_↵
container.hpp, 112
- src/file_manager/image_container/ppm_image_↵
container.cpp, 113

- src/file_manager/image_container/ppm_image_↵
container.hpp, 113
- src/filters/basic_filter.cpp, 113
- src/filters/basic_filter.hpp, 114
- src/filters/color_inverse.cpp, 115
- src/filters/color_inverse.hpp, 115
- src/filters/filter_menu.cpp, 115
- src/filters/filter_menu.hpp, 115
- src/filters/grayscale.cpp, 116
- src/filters/grayscale.hpp, 116
- src/filters/pixelate.cpp, 116
- src/filters/pixelate.hpp, 117
- src/filters/sepia.cpp, 117
- src/filters/sepia.hpp, 117
- src/gtk_base_window.hpp, 118
- src/main.cpp, 118
- src/paint_exception.cpp, 119
- src/paint_exception.hpp, 119
- src/tools/brushes/brush.cpp, 119
- src/tools/brushes/brush.hpp, 120
- src/tools/brushes/spray.cpp, 120
- src/tools/brushes/spray.hpp, 120
- src/tools/brushes/star_brush.cpp, 121
- src/tools/brushes/star_brush.hpp, 121
- src/tools/color_picker.cpp, 121
- src/tools/color_picker.hpp, 122
- src/tools/flood_fill.cpp, 122
- src/tools/flood_fill.hpp, 122
- src/tools/shapes/ellipse.cpp, 123
- src/tools/shapes/ellipse.hpp, 123
- src/tools/shapes/line.cpp, 123
- src/tools/shapes/line.hpp, 124
- src/tools/shapes/rectangle.cpp, 124
- src/tools/shapes/rectangle.hpp, 124
- src/tools/shapes/shape.cpp, 125
- src/tools/shapes/shape.hpp, 125
- src/tools/shapes/triangle.cpp, 125
- src/tools/shapes/triangle.hpp, 126
- src/tools/tool.cpp, 126
- src/tools/tool.hpp, 126
- src/tools/tool_menu.cpp, 127
- src/tools/tool_menu.hpp, 128
- src/window/menu.cpp, 128
- src/window/menu.hpp, 128
- src/window/window.cpp, 129
- src/window/window.hpp, 130
- star_brush, 83
 - PaintPixel, 84
 - star_brush, 84
- swap_current_and_undo_pixels
window, 105
- texture
window, 107
- tolerance_ratio
flood_fill, 43
- tool, 84
 - ~tool, 86
 - brush_size, 90
 - current_color, 90
 - delete_event, 86
 - destroy, 87
 - get_brush_size, 87
 - get_current_color, 87
 - get_tool_type, 87
 - is_a_brush, 88
 - open_tool_properties, 88
 - operator=, 88
 - PaintPixel, 88
 - pressed_left_button, 91
 - pressed_left_button_second_time, 91
 - resize_tool_size, 89
 - type, 91
 - update_current_color, 89
 - update_window_information, 89
 - use_tool, 90
- tool.hpp
tool_type, 127
- tool_labels
menu, 60
- tool_menu, 91
 - ~tool_menu, 93
 - convert_from_Uint8, 93
 - get_color_response, 93
 - get_tool_response, 93
 - get_tool_type, 94
 - new_color, 95
 - number_of_tools, 95
 - open_gtk_window, 94
 - RGB, 95
 - set_current_color, 94
 - set_current_tool, 95
 - tool_menu, 93
 - tool_response, 96
 - tool_types_array, 96
 - trigger_color_chooser, 95
- tool_menu_widget
window, 107
- tool_prt
menu, 60
- tool_response
tool_menu, 96
- tool_type
tool.hpp, 127
- tool_type_data, 96
response, 96
type, 97
- tool_types_array
tool_menu, 96
- tools_images_paths
menu, 60
- triangle, 97
draw_shape, 98
triangle, 98
- trigger_color_chooser
tool_menu, 95
- trigger_file_manager_menu

- window, 105
- trigger_filter_menu
 - window, 106
- trigger_tool_menu
 - window, 106
- type
 - basic_filter, 16
 - filter_type_data, 40
 - tool, 91
 - tool_type_data, 97
- update_brush_size_state
 - menu, 58
- update_colors
 - menu, 58
- update_current_color
 - tool, 89
- update_file_manager_information
 - file_manager, 35
- update_filter_information
 - basic_filter, 14
- update_tool_pointer
 - menu, 58
- update_tool_state
 - menu, 58
- update_window_information
 - menu, 59
 - tool, 89
- use_tool
 - brush, 23
 - color_picker, 27
 - flood_fill, 43
 - shape, 79
 - tool, 90
 - window, 106
- value_changed
 - flood_fill, 43
 - pixelate, 68
 - spray, 82
- vertical_resolution
 - bmp_image_container::bmp_file_info, 19
- what
 - paint_exception, 64
- width
 - bmp_image_container::bmp_file_info, 19
 - image_container::basic_image_info, 17
 - ppm_image_container::ppm_file_info, 71
- window, 99
 - ~window, 101
 - clicked_on_menu, 101
 - copied_pixels, 106
 - current_tool, 107
 - file_drop, 102
 - file_manager_widget, 107
 - filter_menu_widget, 107
 - get_menu_height, 102
 - init_sld, 102
 - is_the_tool_a_brush, 102
 - pressed_left_button_second_time_tool_flag, 102
 - pressed_left_button_tool_flag, 103
 - print_sdl_error, 103
 - redraw_window, 104
 - render, 107
 - render_window, 104
 - renderer, 107
 - reset_canva, 104
 - resize_brush, 104
 - resized_window, 104, 105
 - sdl_window, 107
 - set_copied_pixels, 105
 - swap_current_and_undo_pixels, 105
 - texture, 107
 - tool_menu_widget, 107
 - trigger_file_manager_menu, 105
 - trigger_filter_menu, 106
 - trigger_tool_menu, 106
 - use_tool, 106
 - window, 101
 - window_menu, 108
 - window_name, 108
- window_area
 - base_window, 12
- window_height
 - base_window, 12
- window_menu
 - window, 108
- window_name
 - window, 108
- window_width
 - base_window, 12
- working_on_file
 - file_manager, 36
- x_begining
 - shape, 79
- y_begining
 - shape, 79