Yuki

Generated by Doxygen 1.9.6

1 Namespace Index 1
1.1 Namespace List . . . 1

2 Hierarchical Index 3
2.1 Class Hierarchy . . . 3

3 Class Index 5
3.1 Class List . . . 5

4 File Index 7
4.1 File List . . . 7

5 Namespace Documentation 9
5.1 yuki Namespace Reference . . . 9
5.1.1 Enumeration Type Documentation . . . 10
5.1.1.1 Camp . . . 10
5.1.1.2 Direction . . . 11
5.1.1.3 ElementumReactionType . . . 11
5.1.1.4 ElementumType . . . 11
5.1.1.5 TileCategory . . . 11
5.1.2 Function Documentation . . . 12
5.1.2.1 getElementumReaction() . . . 12
5.1.2.2 getElementumReactionType() . . . 13
5.1.2.3 getHealthBar() . . . 13
5.1.2.4 getNormalSoldier() . . . 14
5.1.2.5 getReactedElementum() . . . 14
5.1.2.6 getSoldier() . . . 14
5.1.2.7 getTileCategoryName() . . . 15
5.1.2.8 getTileScaleFactor() . . . 15
5.1.2.9 getTileSrcName() . . . 16
5.1.2.10 getVectorSpeed() . . . 16
5.1.2.11 isOppositeDirection() . . . 17
5.1.3 Variable Documentation . . . 17
5.1.3.1 kPi . . . 17
5.1.3.2 kSkillIconBorderDistance . . . 17
5.1.3.3 kWindowHeight . . . 17
5.1.3.4 kWindowWidth . . . 17
5.2 yuki::GameFont Namespace Reference . . . 18
5.2.1 Function Documentation . . . 18
5.2.1.1 initFont() . . . 18
5.2.2 Variable Documentation . . . 18
5.2.2.1 DefaultFont . . . 18
5.3 yuki::YukiColor Namespace Reference . . . 18
5.3.1 Function Documentation . . . 18

5.3.1.1 Gray() . . . 19
5.3.1.2 Normal() . . . 19
5.3.1.3 Transparent() . . . 19
5.3.1.4 Transparent_25() . . . 20
5.3.1.5 Transparent_50() . . . 20
5.3.1.6 Transparent_75() . . . 20

6 Class Documentation **21**
6.1 yuki::AttackInfo Struct Reference . . . 21
6.1.1 Constructor & Destructor Documentation . . . 21
6.1.1.1 AttackInfo() . . . 21
6.1.2 Member Data Documentation . . . 21
6.1.2.1 damage . . . 22
6.1.2.2 elementum_level . . . 22
6.1.2.3 elementum_type . . . 22
6.2 yuki::BasicAttack Class Reference . . . 22
6.2.1 Constructor & Destructor Documentation . . . 23
6.2.1.1 BasicAttack() [1/3] . . . 23
6.2.1.2 BasicAttack() [2/3] . . . 23
6.2.1.3 BasicAttack() [3/3] . . . 23
6.2.1.4 ~BasicAttack() . . . 23
6.2.2 Member Function Documentation . . . 23
6.2.2.1 getAttackInfo() . . . 24
6.2.2.2 getDamage() . . . 24
6.2.2.3 getElementum() . . . 24
6.2.3 Member Data Documentation . . . 24
6.2.3.1 info_ . . . 25
6.3 yuki::BubbleItem Class Reference . . . 25
6.3.1 Constructor & Destructor Documentation . . . 26
6.3.1.1 BubbleItem() [1/2] . . . 26
6.3.1.2 BubbleItem() [2/2] . . . 27
6.3.1.3 ~BubbleItem() . . . 27
6.3.2 Member Function Documentation . . . 27
6.3.2.1 loadTexture() . . . 27
6.3.3 Friends And Related Function Documentation . . . 27
6.3.3.1 FloatingBubble . . . 27
6.3.4 Member Data Documentation . . . 27
6.3.4.1 texture_ . . . 27
6.4 yuki::Button Class Reference . . . 28
6.4.1 Member Enumeration Documentation . . . 30
6.4.1.1 Alignment . . . 30
6.4.2 Constructor & Destructor Documentation . . . 30

6.4.2.1 Button() [1/4] . . . 30
6.4.2.2 Button() [2/4] . . . 30
6.4.2.3 Button() [3/4] . . . 31
6.4.2.4 Button() [4/4] . . . 31
6.4.2.5 ~Button() . . . 31
6.4.3 Member Function Documentation . . . 31
6.4.3.1 adjustTextPosition() . . . 31
6.4.3.2 autoAdjustTextSize() . . . 31
6.4.3.3 draw() . . . 32
6.4.3.4 getOrigin() . . . 32
6.4.3.5 getPosition() . . . 32
6.4.3.6 getSize() . . . 32
6.4.3.7 getString() . . . 32
6.4.3.8 getTexture() . . . 32
6.4.3.9 setFillColor() . . . 32
6.4.3.10 setOrigin() . . . 33
6.4.3.11 setOutlineColor() . . . 33
6.4.3.12 setOutlineThickness() . . . 33
6.4.3.13 setPosition() [1/2] . . . 34
6.4.3.14 setPosition() [2/2] . . . 34
6.4.3.15 setSize() . . . 35
6.4.3.16 setString() . . . 36
6.4.3.17 setTextAlignment() . . . 36
6.4.3.18 setTextFillColor() . . . 36
6.4.3.19 setTextFont() . . . 37
6.4.3.20 setTextOutlineColor() . . . 37
6.4.3.21 setTextOutlineThickness() . . . 37
6.4.3.22 setTextSize() . . . 37
6.4.3.23 setTexture() . . . 37
6.4.3.24 setTextureRect() . . . 37
6.4.4 Member Data Documentation . . . 37
6.4.4.1 alignment_ . . . 38
6.4.4.2 shape_ . . . 38
6.4.4.3 text_ . . . 38
6.5 yuki::CircularSector Class Reference . . . 38
6.5.1 Constructor & Destructor Documentation . . . 39
6.5.1.1 CircularSector() . . . 39
6.5.2 Member Function Documentation . . . 39
6.5.2.1 getCenter() . . . 39
6.5.2.2 getPoint() . . . 39
6.5.2.3 getPointCount() . . . 40
6.5.2.4 setCenter() . . . 40

6.5.2.5 setEndAngle() . . . 40
6.5.2.6 setRadius() . . . 40
6.5.2.7 setStartAngle() . . . 40
6.5.3 Member Data Documentation . . . 40
6.5.3.1 center_ . . . 41
6.5.3.2 end_angle_ . . . 41
6.5.3.3 radius_ . . . 41
6.5.3.4 start_angle_ . . . 41
6.6 yuki::Elementum Struct Reference . . . 41
6.6.1 Constructor & Destructor Documentation . . . 41
6.6.1.1 Elementum() . . . 42
6.6.2 Member Data Documentation . . . 42
6.6.2.1 level . . . 42
6.6.2.2 type . . . 42
6.7 yuki::ElementumPanel Class Reference . . . 42
6.7.1 Constructor & Destructor Documentation . . . 44
6.7.1.1 ElementumPanel() . . . 44
6.7.2 Member Function Documentation . . . 44
6.7.2.1 activate() . . . 44
6.7.2.2 deactivate() . . . 44
6.7.2.3 getElementumType() . . . 44
6.7.2.4 setElementumType() . . . 45
6.7.3 Member Data Documentation . . . 45
6.7.3.1 elementum_type_ . . . 45
6.8 yuki::ElementumReaction Struct Reference . . . 46
6.8.1 Member Data Documentation . . . 46
6.8.1.1 elementum . . . 46
6.8.1.2 type . . . 46
6.9 yuki::FloatingBubble Class Reference . . . 47
6.9.1 Constructor & Destructor Documentation . . . 48
6.9.1.1 FloatingBubble() [1/2] . . . 48
6.9.1.2 FloatingBubble() [2/2] . . . 48
6.9.2 Member Function Documentation . . . 48
6.9.2.1 draw() . . . 49
6.9.2.2 getBubbleItems() [1/2] . . . 49
6.9.2.3 getBubbleItems() [2/2] . . . 49
6.9.2.4 getBubblePosition() . . . 49
6.9.2.5 getBubbleRadius() . . . 50
6.9.2.6 getIndexByPosition() . . . 50
6.9.2.7 getPosition() . . . 50
6.9.2.8 setBubbleRadius() . . . 50
6.9.2.9 setPosition() . . . 51

6.9.2.10 updateBubblesPosition() . . . 51
6.9.3 Member Data Documentation . . . 52
6.9.3.1 bubble_num_ . . . 52
6.9.3.2 bubble_radius_ . . . 52
6.9.3.3 bubbles_ . . . 52
6.9.3.4 direction_ . . . 52
6.9.3.5 distribute_radius_ . . . 52
6.9.3.6 position_ . . . 53
6.10 yuki::Focusable Class Reference . . . 53
6.10.1 Member Typedef Documentation . . . 55
6.10.1.1 Callback . . . 55
6.10.2 Constructor & Destructor Documentation . . . 55
6.10.2.1 Focusable() . . . 55
6.10.2.2 ~Focusable() . . . 55
6.10.3 Member Function Documentation . . . 55
6.10.3.1 bindFocus() . . . 55
6.10.3.2 bindUnfocus() . . . 55
6.10.3.3 isFocused() . . . 55
6.10.3.4 onFocus() . . . 56
6.10.3.5 onUnfocus() . . . 56
6.10.3.6 updateStatus() . . . 56
6.10.4 Member Data Documentation . . . 57
6.10.4.1 is_focused_ . . . 57
6.10.4.2 on_focusCallback_ . . . 57
6.10.4.3 on_unfocusCallback_ . . . 57
6.11 yuki::GranuleAttack Class Reference . . . 58
6.11.1 Constructor & Destructor Documentation . . . 59
6.11.1.1 GranuleAttack() [1/2] . . . 59
6.11.1.2 GranuleAttack() [2/2] . . . 60
6.11.1.3 ~GranuleAttack() . . . 60
6.11.2 Member Function Documentation . . . 60
6.11.2.1 draw() . . . 60
6.11.2.2 getPosition() . . . 60
6.11.2.3 update() . . . 60
6.11.2.4 updateGranule() . . . 61
6.11.3 Member Data Documentation . . . 61
6.11.3.1 direction_ . . . 61
6.11.3.2 granule_ . . . 61
6.11.3.3 speed_ . . . 61
6.12 yuki::InfoBar Class Reference . . . 62
6.12.1 Constructor & Destructor Documentation . . . 63
6.12.1.1 InfoBar() [1/2] . . . 63

6.12.1.2 InfoBar() [2/2] . . . 63
6.12.1.3 ~InfoBar() . . . 64
6.12.2 Member Function Documentation . . . 64
6.12.2.1 draw() . . . 64
6.12.2.2 getPosition() . . . 64
6.12.2.3 getSize() . . . 64
6.12.2.4 initTextures() . . . 64
6.12.2.5 initUi() . . . 65
6.12.2.6 setMoney() . . . 65
6.12.2.7 setPosition() . . . 65
6.12.2.8 setSize() . . . 65
6.12.3 Member Data Documentation . . . 66
6.12.3.1 background_ . . . 66
6.12.3.2 background_texture_ . . . 66
6.12.3.3 icon_money_ . . . 66
6.12.3.4 icon_money_texture_ . . . 66
6.12.3.5 kIconSize . . . 66
6.12.3.6 kMoneyIconOffset . . . 66
6.12.3.7 kMoneyTextOffset . . . 66
6.12.3.8 text_money_ . . . 67
6.13 yuki::MainScene Class Reference . . . 67
6.13.1 Member Typedef Documentation . . . 69
6.13.1.1 GranuleAttackWithSenderInfo . . . 70
6.13.2 Member Enumeration Documentation . . . 70
6.13.2.1 Message . . . 70
6.13.2.2 ObjectType . . . 70
6.13.3 Constructor & Destructor Documentation . . . 70
6.13.3.1 MainScene() . . . 70
6.13.3.2 ~MainScene() . . . 71
6.13.4 Member Function Documentation . . . 71
6.13.4.1 coordinateToPixel() . . . 71
6.13.4.2 decreaseEnemyMoney() . . . 72
6.13.4.3 decreaseMoney() . . . 72
6.13.4.4 decreaseOwnMoney() . . . 73
6.13.4.5 enemyBaseMakeDecision() . . . 73
6.13.4.6 eraseDeadSoldier() . . . 74
6.13.4.7 generateSoldier() . . . 74
6.13.4.8 implMsgGenerateEnemySoldier() . . . 75
6.13.4.9 implMsgGenerateOwnSoldier() . . . 75
6.13.4.10 increaseEnemyMoney() . . . 75
6.13.4.11 increaseMoney() . . . 76
6.13.4.12 increaseOwnMoney() . . . 76

6.13.4.13 initBuildings() . . . 77
6.13.4.14 initButton() . . . 78
6.13.4.15 initElementumPanel() . . . 78
6.13.4.16 initInfoHint() . . . 79
6.13.4.17 initMap() . . . 79
6.13.4.18 initUi() . . . 80
6.13.4.19 normalSoldierAttack() . . . 81
6.13.4.20 pixelToCoordinate() . . . 81
6.13.4.21 registerGranuleAttack() . . . 82
6.13.4.22 sendMessage() . . . 82
6.13.4.23 setEnemyCurrentElementumType() . . . 83
6.13.4.24 setMoney() . . . 83
6.13.4.25 setOwnCurrentElementumType() . . . 84
6.13.4.26 soldierUpdatePosition() . . . 84
6.13.4.27 updateAttacks() . . . 85
6.13.5 Member Data Documentation . . . 85
6.13.5.1 attack_button_ . . . 85
6.13.5.2 attack_button_texture_ . . . 85
6.13.5.3 debug_text_ . . . 85
6.13.5.4 elementum_panels_ . . . 85
6.13.5.5 enemies_ . . . 86
6.13.5.6 enemy_base_ . . . 86
6.13.5.7 enemy_current_elementum_type_ . . . 86
6.13.5.8 enemy_money_ . . . 86
6.13.5.9 focused_object_ . . . 86
6.13.5.10 focused_object_type_ . . . 86
6.13.5.11 granule_attacks_ . . . 86
6.13.5.12 info_bar_ . . . 86
6.13.5.13 kEnemyBaseCoordinate . . . 87
6.13.5.14 kEnemySoldierBirthCoordinate . . . 87
6.13.5.15 kOwnBaseCoordinate . . . 87
6.13.5.16 kOwnSoldierBirthCoordinate . . . 87
6.13.5.17 map_ . . . 87
6.13.5.18 message_queue_ . . . 87
6.13.5.19 own_base_ . . . 87
6.13.5.20 own_current_elementum_type_ . . . 87
6.13.5.21 own_money_ . . . 88
6.13.5.22 skill_bar_ . . . 88
6.13.5.23 soldiers_ . . . 88
6.14 yuki::Map Class Reference . . . 88
6.14.1 Constructor & Destructor Documentation . . . 89
6.14.1.1 Map() . . . 89

6.14.1.2 ~Map() . . . 89
6.14.2 Member Function Documentation . . . 89
6.14.2.1 draw() . . . 90
6.14.2.2 getCoordinate() . . . 90
6.14.2.3 getSize() . . . 90
6.14.2.4 getTile() [1/4] . . . 90
6.14.2.5 getTile() [2/4] . . . 91
6.14.2.6 getTile() [3/4] . . . 91
6.14.2.7 getTile() [4/4] . . . 91
6.14.2.8 getTileSize() . . . 92
6.14.2.9 setSize() . . . 92
6.14.2.10 setTile() . . . 92
6.14.2.11 setTileSize() . . . 93
6.14.3 Member Data Documentation . . . 93
6.14.3.1 bg_texture_ . . . 93
6.14.3.2 bg_tiles_ . . . 93
6.14.3.3 size_ . . . 93
6.14.3.4 tile_size_ . . . 94
6.14.3.5 tiles_ . . . 94
6.15 yuki::Menu Class Reference . . . 94
6.15.1 Member Enumeration Documentation . . . 95
6.15.1.1 Layout . . . 95
6.15.2 Constructor & Destructor Documentation . . . 95
6.15.2.1 Menu() . . . 96
6.15.2.2 ~Menu() . . . 96
6.15.3 Member Function Documentation . . . 96
6.15.3.1 addItem() . . . 96
6.15.3.2 adjustItemsPosition() . . . 96
6.15.3.3 draw() . . . 96
6.15.3.4 getByName() . . . 96
6.15.3.5 getLayout() . . . 96
6.15.3.6 operator[]() . . . 97
6.15.3.7 setLayout() . . . 97
6.15.4 Member Data Documentation . . . 97
6.15.4.1 layout_ . . . 97
6.15.4.2 menu_items_ . . . 97
6.15.4.3 position_ . . . 97
6.15.4.4 spacing_ . . . 97
6.16 yuki::MilitaryBase Class Reference . . . 98
6.16.1 Constructor & Destructor Documentation . . . 100
6.16.1.1 MilitaryBase() . . . 100
6.16.1.2 ~MilitaryBase() . . . 101

6.16.2 Member Function Documentation . . . 101
6.16.2.1 canGenerateSoldier() . . . 101
6.16.2.2 draw() . . . 101
6.16.2.3 getAttacked() . . . 101
6.16.2.4 getCurrentHealth() . . . 102
6.16.2.5 getFloatingBubble() [1/2] . . . 102
6.16.2.6 getFloatingBubble() [2/2] . . . 102
6.16.2.7 getFloatingBubbleIndexByPosition() . . . 102
6.16.2.8 getFloatingBubbleVisible() . . . 103
6.16.2.9 getLevel() . . . 103
6.16.2.10 getMaxHealth() . . . 103
6.16.2.11 initTexture() . . . 103
6.16.2.12 levelDown() . . . 104
6.16.2.13 levelUp() . . . 104
6.16.2.14 resetGenerateSoldierClock() . . . 104
6.16.2.15 setCurrentHealth() . . . 104
6.16.2.16 setFloatingBubbleVisible() . . . 105
6.16.2.17 setMaxHealth() . . . 105
6.16.2.18 setPosition() . . . 105
6.16.2.19 updateTexture() . . . 106
6.16.3 Member Data Documentation . . . 106
6.16.3.1 camp_ . . . 106
6.16.3.2 floating_bubble_ . . . 106
6.16.3.3 floating_bubble_visible_ . . . 107
6.16.3.4 generateSoldierClock_ . . . 107
6.16.3.5 health_bar_ . . . 107
6.16.3.6 health_bar_offset_y . . . 107
6.16.3.7 level_ . . . 107
6.16.3.8 texture_ . . . 107
6.17 yuki::MouseEventProcessEngine Class Reference . . . 107
6.17.1 Constructor & Destructor Documentation . . . 108
6.17.1.1 MouseEventProcessEngine() . . . 108
6.17.1.2 ~MouseEventProcessEngine() . . . 108
6.17.2 Member Function Documentation . . . 108
6.17.2.1 processEvent() . . . 108
6.17.2.2 registerTouchableObject() . . . 109
6.17.2.3 unregisterTouchableObject() . . . 109
6.17.3 Member Data Documentation . . . 109
6.17.3.1 touchable_objects_ . . . 109
6.18 yuki::NormalSoldier Class Reference . . . 110
6.18.1 Constructor & Destructor Documentation . . . 113
6.18.1.1 NormalSoldier() [1/2] . . . 113

6.18.1.2 NormalSoldier() [2/2] . . . 113
6.18.1.3 ~NormalSoldier() . . . 114
6.18.2 Member Function Documentation . . . 114
6.18.2.1 canGranuleAttack() . . . 114
6.18.2.2 draw() . . . 114
6.18.2.3 getDefaultAttack() . . . 115
6.18.2.4 getDefaultGranuleAttack() . . . 115
6.18.2.5 getGranuleAttackCD() . . . 116
6.18.2.6 resetGranuleAttackClock() . . . 116
6.18.2.7 setGranuleAttackCD() . . . 116
6.18.2.8 update() . . . 116
6.18.3 Member Data Documentation . . . 117
6.18.3.1 all_textures_ . . . 117
6.18.3.2 current_frame_ . . . 117
6.18.3.3 granule_attack_cd_ . . . 117
6.18.3.4 granule_attack_clock_ . . . 117
6.18.3.5 max_frame_ . . . 117
6.18.3.6 place_offset_ . . . 117
6.18.3.7 previous_frame_ . . . 117
6.19 yuki::ResultScene Class Reference . . . 118
6.19.1 Constructor & Destructor Documentation . . . 119
6.19.1.1 ResultScene() . . . 119
6.19.1.2 ~ResultScene() . . . 120
6.19.2 Member Function Documentation . . . 120
6.19.2.1 initUi() . . . 120
6.19.3 Member Data Documentation . . . 120
6.19.3.1 background_ . . . 120
6.19.3.2 background_texture_ . . . 120
6.19.3.3 own_win_ . . . 120
6.19.3.4 result_text_ . . . 121
6.20 yuki::SeniorSoldier Class Reference . . . 121
6.20.1 Constructor & Destructor Documentation . . . 124
6.20.1.1 SeniorSoldier() [1/2] . . . 124
6.20.1.2 SeniorSoldier() [2/2] . . . 124
6.20.1.3 ~SeniorSoldier() . . . 124
6.20.2 Member Function Documentation . . . 125
6.20.2.1 draw() . . . 125
6.20.2.2 update() . . . 125
6.20.3 Member Data Documentation . . . 125
6.20.3.1 all_textures_ . . . 125
6.20.3.2 current_frame_ . . . 125
6.20.3.3 floating_bubble_ . . . 125

6.20.3.4 max_frame_ . . . 125
6.20.3.5 place_offset_ . . . 126
6.20.3.6 previous_frame_ . . . 126
6.21 yuki::SkillBar< MaxSkillNum > Class Template Reference . . . 126
6.21.1 Constructor & Destructor Documentation . . . 128
6.21.1.1 SkillBar() . . . 128
6.21.2 Member Function Documentation . . . 128
6.21.2.1 draw() . . . 128
6.21.2.2 getSkillNum() . . . 129
6.21.2.3 updatePosition() . . . 129
6.21.3 Member Data Documentation . . . 129
6.21.3.1 border_thick_ . . . 129
6.21.3.2 boxes_ . . . 129
6.21.3.3 icon_radius_ . . . 129
6.21.3.4 position_ . . . 130
6.21.3.5 skill_num_ . . . 130
6.21.3.6 skills_ . . . 130
6.22 yuki::SkillIcon Class Reference . . . 130
6.22.1 Constructor & Destructor Documentation . . . 132
6.22.1.1 SkillIcon() . . . 132
6.22.2 Member Function Documentation . . . 132
6.22.2.1 draw() . . . 132
6.22.2.2 getCurrentValue() . . . 133
6.22.2.3 getMaxValue() . . . 133
6.22.2.4 setCurrentValue() . . . 133
6.22.2.5 setMaxValue() . . . 133
6.22.2.6 setPosition() . . . 133
6.22.2.7 updateCircularSector() . . . 133
6.22.3 Member Data Documentation . . . 134
6.22.3.1 background_ . . . 134
6.22.3.2 circular_sector_ . . . 134
6.22.3.3 current_value_ . . . 134
6.22.3.4 icon_ . . . 134
6.22.3.5 max_value_ . . . 134
6.23 yuki::Soldier Class Reference . . . 135
6.23.1 Constructor & Destructor Documentation . . . 138
6.23.1.1 Soldier() [1/2] . . . 138
6.23.1.2 Soldier() [2/2] . . . 138
6.23.1.3 ~Soldier() . . . 138
6.23.2 Member Function Documentation . . . 139
6.23.2.1 decreaseHealth() . . . 139
6.23.2.2 draw() . . . 139

6.23.2.3 getAdhesionElementum() . . . 140
6.23.2.4 getAttacked() . . . 140
6.23.2.5 getCamp() . . . 140
6.23.2.6 getColor() . . . 140
6.23.2.7 getDefaultAttack() . . . 140
6.23.2.8 getDirection() . . . 141
6.23.2.9 getElementum() . . . 141
6.23.2.10 getElementumType() . . . 141
6.23.2.11 getFloatingBubble() [1/2] . . . 141
6.23.2.12 getFloatingBubble() [2/2] . . . 142
6.23.2.13 getHealth() . . . 142
6.23.2.14 getMaxHealth() . . . 142
6.23.2.15 getOrigin() . . . 142
6.23.2.16 getPlaceOffset() . . . 143
6.23.2.17 getPosition() . . . 143
6.23.2.18 getSize() . . . 143
6.23.2.19 getSpeed() . . . 143
6.23.2.20 increaseHealth() . . . 144
6.23.2.21 isFloatingBubbleVisible() . . . 144
6.23.2.22 isFreezed() . . . 144
6.23.2.23 isMoving() . . . 144
6.23.2.24 setAdhesionElementum() . . . 144
6.23.2.25 setColor() . . . 144
6.23.2.26 setDirection() . . . 145
6.23.2.27 setElementum() . . . 145
6.23.2.28 setElementumType() . . . 145
6.23.2.29 setFloatingBubbleVisible() . . . 145
6.23.2.30 setHealth() . . . 145
6.23.2.31 setMaxHealth() . . . 146
6.23.2.32 setMoving() . . . 146
6.23.2.33 setOrigin() . . . 146
6.23.2.34 setPlaceOffset() . . . 147
6.23.2.35 setPosition() . . . 147
6.23.2.36 setSpeed() . . . 147
6.23.2.37 update() . . . 148
6.23.2.38 walk() . . . 148
6.23.3 Member Data Documentation . . . 149
6.23.3.1 adhesion_elementum_ . . . 149
6.23.3.2 camp_ . . . 149
6.23.3.3 clock_ . . . 149
6.23.3.4 direction_ . . . 149
6.23.3.5 elementum_ . . . 149

6.23.3.6 floating_bubble_ . . . 150
6.23.3.7 floating_bubble_visible_ . . . 150
6.23.3.8 freeze_clock_ . . . 150
6.23.3.9 health_bar_ . . . 150
6.23.3.10 health_bar_offset_ . . . 150
6.23.3.11 is_freeze_ . . . 150
6.23.3.12 is_moving_ . . . 150
6.23.3.13 place_offset_ . . . 150
6.23.3.14 speed_ . . . 151
6.23.3.15 sprite_ . . . 151
6.24 yuki::StartupScene Class Reference . . . 151
6.24.1 Constructor & Destructor Documentation . . . 152
6.24.1.1 StartupScene() . . . 153
6.24.1.2 ~StartupScene() . . . 153
6.24.2 Member Function Documentation . . . 153
6.24.2.1 initUi() . . . 154
6.24.3 Member Data Documentation . . . 154
6.24.3.1 background_ . . . 155
6.24.3.2 background_texture_ . . . 155
6.24.3.3 btn_connect_ . . . 155
6.24.3.4 btn_start_ . . . 155
6.25 yuki::StatusBar Class Reference . . . 155
6.25.1 Constructor & Destructor Documentation . . . 157
6.25.1.1 StatusBar() [1/2] . . . 157
6.25.1.2 StatusBar() [2/2] . . . 157
6.25.1.3 ~StatusBar() . . . 157
6.25.2 Member Function Documentation . . . 157
6.25.2.1 adjustIfOverflow() . . . 158
6.25.2.2 decrease() . . . 158
6.25.2.3 draw() . . . 158
6.25.2.4 getBackgroundColor() . . . 159
6.25.2.5 getFilledColor() . . . 159
6.25.2.6 getMaxValue() . . . 159
6.25.2.7 getPosition() . . . 159
6.25.2.8 getValue() . . . 160
6.25.2.9 hide() . . . 160
6.25.2.10 increase() . . . 160
6.25.2.11 isVisible() . . . 161
6.25.2.12 setBackgroudColor() . . . 161
6.25.2.13 setFilledColor() . . . 161
6.25.2.14 setFilledPercentage() . . . 162
6.25.2.15 setMaxValue() . . . 162

6.25.2.16 setOutlineColor() . . . 162
6.25.2.17 setOutlineStyle() . . . 163
6.25.2.18 setOutlineThickness() . . . 163
6.25.2.19 setPosition() . . . 164
6.25.2.20 setSize() . . . 164
6.25.2.21 setValue() . . . 164
6.25.2.22 show() . . . 165
6.25.2.23 updateFilledPercentage() . . . 165
6.25.3 Member Data Documentation . . . 166
6.25.3.1 background_ . . . 166
6.25.3.2 current_value_ . . . 166
6.25.3.3 filled_ . . . 166
6.25.3.4 max_value_ . . . 166
6.25.3.5 position_ . . . 166
6.25.3.6 size_ . . . 166
6.25.3.7 visible_ . . . 166
6.26 yuki::Tile Class Reference . . . 167
6.26.1 Member Function Documentation . . . 167
6.26.1.1 getTileInfo() . . . 168
6.26.1.2 setTextureSrc() . . . 168
6.26.1.3 setTileInfo() [1/2] . . . 168
6.26.1.4 setTileInfo() [2/2] . . . 168
6.26.2 Member Data Documentation . . . 168
6.26.2.1 texture_ . . . 168
6.26.2.2 tile_info_ . . . 169
6.27 yuki::TileInfo Struct Reference . . . 169
6.27.1 Constructor & Destructor Documentation . . . 169
6.27.1.1 TileInfo() [1/2] . . . 169
6.27.1.2 TileInfo() [2/2] . . . 169
6.27.2 Member Data Documentation . . . 169
6.27.2.1 category . . . 170
6.27.2.2 index . . . 170
6.28 yuki::Touchable Class Reference . . . 170
6.28.1 Member Typedef Documentation . . . 171
6.28.1.1 Callback . . . 171
6.28.2 Member Enumeration Documentation . . . 171
6.28.2.1 Action . . . 171
6.28.3 Constructor & Destructor Documentation . . . 172
6.28.3.1 Touchable() . . . 172
6.28.3.2 ~Touchable() . . . 172
6.28.4 Member Function Documentation . . . 172
6.28.4.1 bindAction() . . . 172

6.28.4.2 bindClick() . . . 172
6.28.4.3 bindHover() . . . 173
6.28.4.4 bindLeave() . . . 173
6.28.4.5 bindRelease() . . . 173
6.28.4.6 inRange() . . . 174
6.28.4.7 onClick() . . . 174
6.28.4.8 onHover() . . . 175
6.28.4.9 onLeave() . . . 175
6.28.4.10 onRelease() . . . 175
6.28.4.11 updateStatus() . . . 176
6.28.5 Member Data Documentation . . . 176
6.28.5.1 is_clicked_ . . . 176
6.28.5.2 is_hovered_ . . . 177
6.28.5.3 on_clickCallback_ . . . 177
6.28.5.4 on_hoverCallback_ . . . 177
6.28.5.5 on_leaveCallback_ . . . 177
6.28.5.6 on_releaseCallback_ . . . 177
6.29 yuki::TouchableObject< Decorated > Class Template Reference . . . 178
6.29.1 Constructor & Destructor Documentation . . . 179
6.29.1.1 TouchableObject() . . . 179
6.29.1.2 ~TouchableObject() . . . 179
6.29.2 Member Function Documentation . . . 179
6.29.2.1 inRange() . . . 180
6.29.2.2 updateStatus() . . . 180
6.30 yuki::TouchableSprite Class Reference . . . 180
6.30.1 Constructor & Destructor Documentation . . . 182
6.30.1.1 TouchableSprite() [1/3] . . . 182
6.30.1.2 TouchableSprite() [2/3] . . . 182
6.30.1.3 TouchableSprite() [3/3] . . . 183
6.30.1.4 ~TouchableSprite() . . . 183
6.30.2 Member Function Documentation . . . 183
6.30.2.1 inRange() . . . 183
6.31 yuki::YukiScene Class Reference . . . 183
6.31.1 Member Enumeration Documentation . . . 184
6.31.1.1 Notify . . . 184
6.31.2 Constructor & Destructor Documentation . . . 185
6.31.2.1 YukiScene() . . . 185
6.31.2.2 ~YukiScene() . . . 185
6.31.3 Member Function Documentation . . . 185
6.31.3.1 draw() . . . 185
6.31.3.2 processEvent() . . . 186
6.31.3.3 registerTouchableObject() . . . 186

6.31.3.4 setReturnCode() . . . 187
6.31.3.5 show() . . . 187
6.31.3.6 unregisterTouchableObject() . . . 188
6.31.3.7 updateInfo() . . . 188
6.31.4 Member Data Documentation . . . 188
6.31.4.1 mouse_event_engine_ . . . 189
6.31.4.2 notify_ . . . 189
6.31.4.3 ret_code_ . . . 189
6.31.4.4 window_ . . . 189

7 File Documentation **191**
7.1 E:/Project/Yuki/include/Attack.hpp File Reference . . . 191
7.2 Attack.hpp . . . 192
7.3 E:/Project/Yuki/include/Basic.hpp File Reference . . . 193
7.4 Basic.hpp . . . 195
7.5 E:/Project/Yuki/include/Button.hpp File Reference . . . 195
7.6 Button.hpp . . . 197
7.7 E:/Project/Yuki/include/CircularSector.hpp File Reference . . . 198
7.8 CircularSector.hpp . . . 199
7.9 E:/Project/Yuki/include/Config.hpp File Reference . . . 200
7.9.1 Macro Definition Documentation . . . 200
7.9.1.1 YUKI_DEBUG . . . 201
7.10 Config.hpp . . . 201
7.11 E:/Project/Yuki/include/Elementum.hpp File Reference . . . 201
7.12 Elementum.hpp . . . 202
7.13 E:/Project/Yuki/include/ElementumPanel.hpp File Reference . . . 202
7.14 ElementumPanel.hpp . . . 204
7.15 E:/Project/Yuki/include/FloatingBubble.hpp File Reference . . . 204
7.16 FloatingBubble.hpp . . . 205
7.17 E:/Project/Yuki/include/InfoBar.hpp File Reference . . . 207
7.18 InfoBar.hpp . . . 208
7.19 E:/Project/Yuki/include/InputProcessEngine.hpp File Reference . . . 208
7.20 InputProcessEngine.hpp . . . 210
7.21 E:/Project/Yuki/include/InteractiveObject.hpp File Reference . . . 210
7.21.1 Macro Definition Documentation . . . 211
7.21.1.1 FOCUSABLE_OBJECT . . . 211
7.21.1.2 TOUCHABLE_OBJECT . . . 211
7.22 InteractiveObject.hpp . . . 212
7.23 E:/Project/Yuki/include/MainScene.hpp File Reference . . . 213
7.24 MainScene.hpp . . . 214
7.25 E:/Project/Yuki/include/Map.hpp File Reference . . . 215
7.26 Map.hpp . . . 217

7.27 E:/Project/Yuki/include/Menu.hpp File Reference . . . 217
7.28 Menu.hpp . . . 219
7.29 E:/Project/Yuki/include/MilitaryBase.hpp File Reference . . . 219
7.30 MilitaryBase.hpp . . . 220
7.31 E:/Project/Yuki/include/Network.hpp File Reference . . . 222
7.32 Network.hpp . . . 222
7.33 E:/Project/Yuki/include/ResultScene.hpp File Reference . . . 222
7.34 ResultScene.hpp . . . 223
7.35 E:/Project/Yuki/include/Skill.hpp File Reference . . . 223
7.36 Skill.hpp . . . 225
7.37 E:/Project/Yuki/include/Soldier.hpp File Reference . . . 226
7.38 Soldier.hpp . . . 228
7.39 E:/Project/Yuki/include/StartupScene.hpp File Reference . . . 230
7.40 StartupScene.hpp . . . 231
7.41 E:/Project/Yuki/include/StatusBar.hpp File Reference . . . 232
7.42 StatusBar.hpp . . . 233
7.43 E:/Project/Yuki/include/Tile.hpp File Reference . . . 234
7.44 Tile.hpp . . . 236
7.45 E:/Project/Yuki/include/TouchableSprite.hpp File Reference . . . 237
7.46 TouchableSprite.hpp . . . 238
7.47 E:/Project/Yuki/include/YukiScene.hpp File Reference . . . 238
7.47.1 Macro Definition Documentation . . . 239
7.47.1.1 YUKI_SCENE . . . 239
7.48 YukiScene.hpp . . . 240
7.49 E:/Project/Yuki/src/Attack.cpp File Reference . . . 241
7.50 E:/Project/Yuki/src/Button.cpp File Reference . . . 241
7.51 E:/Project/Yuki/src/CircularSector.cpp File Reference . . . 242
7.51.1 Variable Documentation . . . 242
7.51.1.1 kPi . . . 242
7.52 E:/Project/Yuki/src/Elementum.cpp File Reference . . . 243
7.53 E:/Project/Yuki/src/ElementumPanel.cpp File Reference . . . 243
7.54 E:/Project/Yuki/src/FloatingBubble.cpp File Reference . . . 243
7.54.1 Variable Documentation . . . 244
7.54.1.1 kPi . . . 244
7.55 E:/Project/Yuki/src/InfoBar.cpp File Reference . . . 244
7.56 E:/Project/Yuki/src/InputProcessEngine.cpp File Reference . . . 245
7.57 E:/Project/Yuki/src/InteractiveObject.cpp File Reference . . . 245
7.58 E:/Project/Yuki/src/Main.cpp File Reference . . . 246
7.58.1 Function Documentation . . . 246
7.58.1.1 main() . . . 246
7.59 E:/Project/Yuki/src/MainScene.cpp File Reference . . . 246
7.60 E:/Project/Yuki/src/Map.cpp File Reference . . . 247

7.61 E:/Project/Yuki/src/Menu.cpp File Reference 248
7.62 E:/Project/Yuki/src/MilitaryBase.cpp File Reference 248
7.63 E:/Project/Yuki/src/Network.cpp File Reference 249
7.64 E:/Project/Yuki/src/ResultScene.cpp File Reference 249
7.65 E:/Project/Yuki/src/Skill.cpp File Reference 250
7.66 E:/Project/Yuki/src/Soldier.cpp File Reference 250
7.67 E:/Project/Yuki/src/StartupScene.cpp File Reference 251
7.68 E:/Project/Yuki/src/StatusBar.cpp File Reference 251
7.69 E:/Project/Yuki/src/Tile.cpp File Reference 252
7.70 E:/Project/Yuki/src/YukiScene.cpp File Reference 253

Index **255**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

yuki 9
yuki::GameFont 18
yuki::YukiColor 18

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

yuki::AttackInfo . . . 21
yuki::BasicAttack . . . 22
  yuki::GranuleAttack . . . 58
sf::CircleShape
  yuki::TouchableObject< sf::CircleShape > . . . 178
  yuki::BubbleItem . . . 25
  yuki::ElementumPanel . . . 42
Decorated
  yuki::TouchableObject< Decorated > . . . 178
sf::Drawable
  yuki::Button . . . 28
  yuki::FloatingBubble . . . 47
  yuki::GranuleAttack . . . 58
  yuki::InfoBar . . . 62
  yuki::Map . . . 88
  yuki::Menu . . . 94
  yuki::SkillBar< MaxSkillNum > . . . 126
  yuki::SkillIcon . . . 130
  yuki::Soldier . . . 135
    yuki::NormalSoldier . . . 110
    yuki::SeniorSoldier . . . 121
  yuki::StatusBar . . . 155
yuki::Elementum . . . 41
yuki::ElementumReaction . . . 46
yuki::MouseEventProcessEngine . . . 107
sf::Shape
  yuki::CircularSector . . . 38
sf::Sprite
  yuki::Tile . . . 167
  yuki::TouchableSprite . . . 180
    yuki::MilitaryBase . . . 98
yuki::TileInfo . . . 169
yuki::Touchable . . . 170
  yuki::TouchableObject< sf::CircleShape > . . . 178
  yuki::BubbleItem . . . 25

yuki::Button 28
yuki::ElementumPanel 42
yuki::Focusable 53
yuki::Soldier 135
yuki::SkillBar< MaxSkillNum > 126
yuki::SkillIcon 130
yuki::TouchableObject< Decorated > 178
yuki::TouchableSprite 180
yuki::YukiScene 183
yuki::MainScene 67
yuki::ResultScene 118
yuki::StartupScene 151

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

yuki::AttackInfo 21
yuki::BasicAttack 22
yuki::BubbleItem 25
yuki::Button 28
yuki::CircularSector 38
yuki::Elementum 41
yuki::ElementumPanel 42
yuki::ElementumReaction 46
yuki::FloatingBubble 47
yuki::Focusable 53
yuki::GranuleAttack 58
yuki::InfoBar 62
yuki::MainScene 67
yuki::Map 88
yuki::Menu 94
yuki::MilitaryBase 98
yuki::MouseEventProcessEngine 107
yuki::NormalSoldier 110
yuki::ResultScene 118
yuki::SeniorSoldier 121
yuki::SkillBar< MaxSkillNum > 126
yuki::SkillIcon 130
yuki::Soldier 135
yuki::StartupScene 151
yuki::StatusBar 155
yuki::Tile 167
yuki::TileInfo 169
yuki::Touchable 170
yuki::TouchableObject< Decorated > 178
yuki::TouchableSprite 180
yuki::YukiScene 183

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

E:/Project/Yuki/include/Attack.hpp . . . 191
E:/Project/Yuki/include/Basic.hpp . . . 193
E:/Project/Yuki/include/Button.hpp . . . 195
E:/Project/Yuki/include/CircularSector.hpp . . . 198
E:/Project/Yuki/include/Config.hpp . . . 200
E:/Project/Yuki/include/Elementum.hpp . . . 201
E:/Project/Yuki/include/ElementumPanel.hpp . . . 202
E:/Project/Yuki/include/FloatingBubble.hpp . . . 204
E:/Project/Yuki/include/InfoBar.hpp . . . 207
E:/Project/Yuki/include/InputProcessEngine.hpp . . . 208
E:/Project/Yuki/include/InteractiveObject.hpp . . . 210
E:/Project/Yuki/include/MainScene.hpp . . . 213
E:/Project/Yuki/include/Map.hpp . . . 215
E:/Project/Yuki/include/Menu.hpp . . . 217
E:/Project/Yuki/include/MilitaryBase.hpp . . . 219
E:/Project/Yuki/include/Network.hpp . . . 222
E:/Project/Yuki/include/ResultScene.hpp . . . 222
E:/Project/Yuki/include/Skill.hpp . . . 223
E:/Project/Yuki/include/Soldier.hpp . . . 226
E:/Project/Yuki/include/StartupScene.hpp . . . 230
E:/Project/Yuki/include/StatusBar.hpp . . . 232
E:/Project/Yuki/include/Tile.hpp . . . 234
E:/Project/Yuki/include/TouchableSprite.hpp . . . 237
E:/Project/Yuki/include/YukiScene.hpp . . . 238
E:/Project/Yuki/src/Attack.cpp . . . 241
E:/Project/Yuki/src/Button.cpp . . . 241
E:/Project/Yuki/src/CircularSector.cpp . . . 242
E:/Project/Yuki/src/Elementum.cpp . . . 243
E:/Project/Yuki/src/ElementumPanel.cpp . . . 243
E:/Project/Yuki/src/FloatingBubble.cpp . . . 243
E:/Project/Yuki/src/InfoBar.cpp . . . 244
E:/Project/Yuki/src/InputProcessEngine.cpp . . . 245
E:/Project/Yuki/src/InteractiveObject.cpp . . . 245
E:/Project/Yuki/src/Main.cpp . . . 246
E:/Project/Yuki/src/MainScene.cpp . . . 246

E:/Project/Yuki/src/Map.cpp 247
E:/Project/Yuki/src/Menu.cpp 248
E:/Project/Yuki/src/MilitaryBase.cpp 248
E:/Project/Yuki/src/Network.cpp 249
E:/Project/Yuki/src/ResultScene.cpp 249
E:/Project/Yuki/src/Skill.cpp 250
E:/Project/Yuki/src/Soldier.cpp 250
E:/Project/Yuki/src/StartupScene.cpp 251
E:/Project/Yuki/src/StatusBar.cpp 251
E:/Project/Yuki/src/Tile.cpp 252
E:/Project/Yuki/src/YukiScene.cpp 253

Chapter 5

Namespace Documentation

5.1 yuki Namespace Reference

Namespaces

- namespace GameFont
- namespace YukiColor

Classes

- struct AttackInfo
- class BasicAttack
- class BubbleItem
- class Button
- class CircularSector
- struct Elementum
- class ElementumPanel
- struct ElementumReaction
- class FloatingBubble
- class Focusable
- class GranuleAttack
- class InfoBar
- class MainScene
- class Map
- class Menu
- class MilitaryBase
- class MouseEventProcessEngine
- class NormalSoldier
- class ResultScene
- class SeniorSoldier
- class SkillBar
- class SkillIcon
- class Soldier
- class StartupScene
- class StatusBar
- class Tile
- struct TileInfo
- class Touchable
- class TouchableObject
- class TouchableSprite
- class YukiScene

Enumerations

- enum class Direction {
  Down = 0 , Left , Right , Up ,
  Other }
- enum class Camp { Own , Enemy , Other }
- enum class ElementumType { None , Pyro , Hydro , Cyro }
- enum class ElementumReactionType { None , Evaporation , Freeze , Melt }
- enum class TileCategory {
  Building , Decor , Greenery , Land ,
  River , Road , Stones , Tree }

Functions

- auto isOppositeDirection (const Direction direction1, const Direction direction2)
- sf::Vector2f getVectorSpeed (const Direction direction, const float speed)
- ElementumReactionType getElementumReactionType (const Elementum &lhs, const Elementum &rhs)
- Elementum getReactedElementum (const Elementum &lhs, const Elementum &rhs)
- ElementumReaction getElementumReaction (const Elementum &lhs, const Elementum &rhs)
- std::unique_ptr< NormalSoldier > getNormalSoldier (Camp camp=Camp::Own)
- std::unique_ptr< Soldier > getSoldier (const std::string &name, Camp camp=Camp::Own)
- StatusBar getHealthBar (Camp camp=Camp::Own, const sf::Vector2f &size={30.f, 5.f})
- std::string getTileCategoryName (const TileCategory category)
- std::string getTileSrcName (const TileInfo &tile_info)
- sf::Vector2f getTileScaleFactor (const TileInfo &tile_info)

Variables

- constexpr auto kWindowWidth = 960
- constexpr auto kWindowHeight = 640
- constexpr auto kPi = 3.14159265358979323846f
- constexpr auto kSkillIconBorderDistance = 3.f

5.1.1 Enumeration Type Documentation

5.1.1.1 Camp

```
enum class yuki::Camp  [strong]
```

Enumerator

Own	
Enemy	
Other	

5.1.1.2 Direction

`enum class yuki::Direction [strong]`

Enumerator

Down	
Left	
Right	
Up	
Other	

5.1.1.3 ElementumReactionType

`enum class yuki::ElementumReactionType [strong]`

Enumerator

None	
Evaporation	
Freeze	
Melt	

5.1.1.4 ElementumType

`enum class yuki::ElementumType [strong]`

Enumerator

None	
Pyro	
Hydro	
Cyro	

5.1.1.5 TileCategory

`enum class yuki::TileCategory [strong]`

Enumerator

Building	

Enumerator

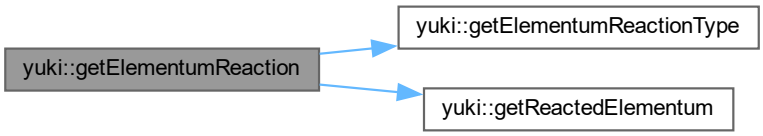
Decor	
Greenery	
Land	
River	
Road	
Stones	
Tree	

5.1.2 Function Documentation

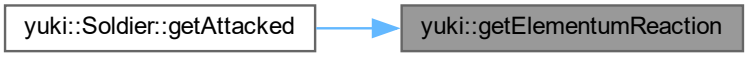
5.1.2.1 getElementumReaction()

```
ElementumReaction yuki::getElementumReaction (
            const Elementum & lhs,
            const Elementum & rhs )
```

Here is the call graph for this function:



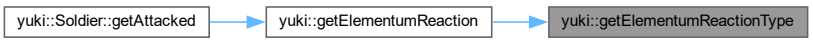
Here is the caller graph for this function:



5.1.2.2 getElementumReactionType()

```
ElementumReactionType yuki::getElementumReactionType (
            const Elementum & lhs,
            const Elementum & rhs )
```

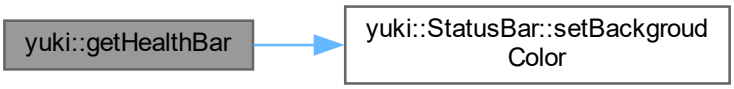
Here is the caller graph for this function:



5.1.2.3 getHealthBar()

```
StatusBar yuki::getHealthBar (
            Camp camp = Camp::Own,
            const sf::Vector2f & size = {30.f, 5.f} )
```

Here is the call graph for this function:

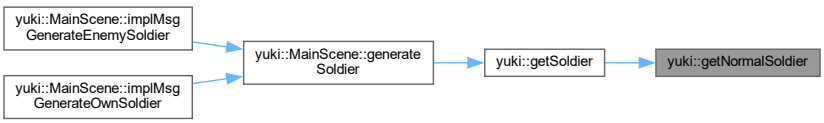


Here is the caller graph for this function:

5.1.2.4 getNormalSoldier()

```
std::unique_ptr< NormalSoldier > yuki::getNormalSoldier (
            Camp camp = Camp::Own )
```

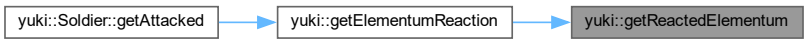
Here is the caller graph for this function:



5.1.2.5 getReactedElementum()

```
Elementum yuki::getReactedElementum (
            const Elementum & lhs,
            const Elementum & rhs )
```

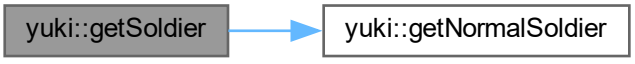
Here is the caller graph for this function:



5.1.2.6 getSoldier()

```
std::unique_ptr< Soldier > yuki::getSoldier (
            const std::string & name,
            Camp camp = Camp::Own )
```

Here is the call graph for this function:

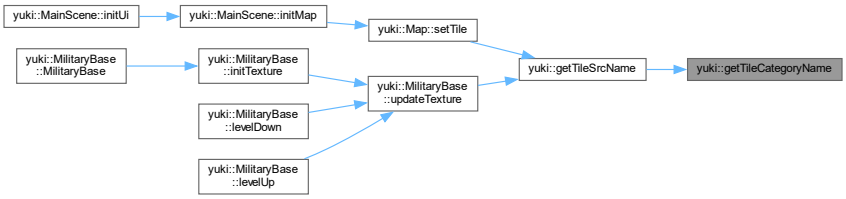


Here is the caller graph for this function:

5.1.2.7 getTileCategoryName()

```
std::string yuki::getTileCategoryName (
            const TileCategory category )
```

Here is the caller graph for this function:



5.1.2.8 getTileScaleFactor()

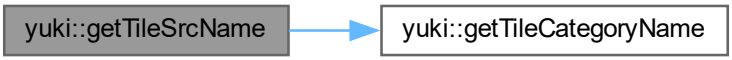
```
sf::Vector2f yuki::getTileScaleFactor (
            const TileInfo & tile_info )
```

Here is the caller graph for this function:

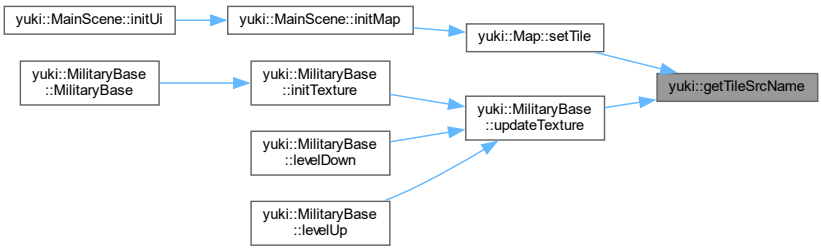
5.1.2.9 getTileSrcName()

```
std::string yuki::getTileSrcName (
            const TileInfo & tile_info )
```

Here is the call graph for this function:



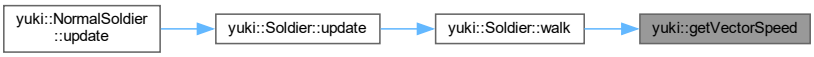
Here is the caller graph for this function:



5.1.2.10 getVectorSpeed()

```
sf::Vector2f yuki::getVectorSpeed (
            const Direction direction,
            const float speed )  [inline]
```

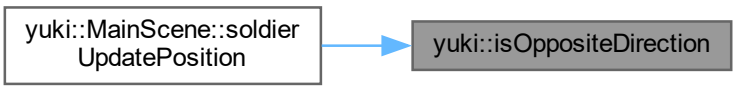
Here is the caller graph for this function:



5.1.2.11 isOppositeDirection()

```
auto yuki::isOppositeDirection (
            const Direction direction1,
            const Direction direction2 )  [inline]
```

Here is the caller graph for this function:



5.1.3 Variable Documentation

5.1.3.1 kPi

```
constexpr auto yuki::kPi = 3.14159265358979323846f  [constexpr]
```

5.1.3.2 kSkillIconBorderDistance

```
constexpr auto yuki::kSkillIconBorderDistance = 3.f  [constexpr]
```

5.1.3.3 kWindowHeight

```
constexpr auto yuki::kWindowHeight = 640  [constexpr]
```

5.1.3.4 kWindowWidth

```
constexpr auto yuki::kWindowWidth = 960  [constexpr]
```

5.2 yuki::GameFont Namespace Reference

Functions

- auto initFont (const sf::String &font_path)

Variables

- static const auto DefaultFont = initFont("assets/font/ARLRDBD.ttf")

5.2.1 Function Documentation

5.2.1.1 initFont()

```
auto yuki::GameFont::initFont (
            const sf::String & font_path )  [inline]
```

5.2.2 Variable Documentation

5.2.2.1 DefaultFont

```
const auto yuki::GameFont::DefaultFont = initFont("assets/font/ARLRDBD.ttf")  [static]
```

5.3 yuki::YukiColor Namespace Reference

Functions

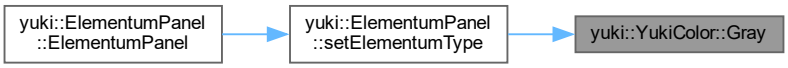
- static const sf::Color Transparent_50 (255, 255, 255, 128)
- static const sf::Color Transparent (255, 255, 255, 0)
- static const sf::Color Transparent_25 (255, 255, 255, 192)
- static const sf::Color Transparent_75 (255, 255, 255, 64)
- static const sf::Color Normal (255, 255, 255, 255)
- static const sf::Color Gray (128, 128, 128, 255)

5.3.1 Function Documentation

5.3.1.1 Gray()

```
static const sf::Color yuki::YukiColor::Gray (
            128 ,
            128 ,
            128 ,
            255  )  [static]
```

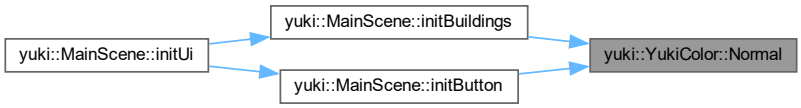
Here is the caller graph for this function:



5.3.1.2 Normal()

```
static const sf::Color yuki::YukiColor::Normal (
            255 ,
            255 ,
            255 ,
            255  )  [static]
```

Here is the caller graph for this function:



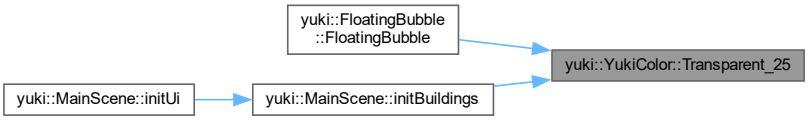
5.3.1.3 Transparent()

```
static const sf::Color yuki::YukiColor::Transparent (
            255 ,
            255 ,
            255 ,
            0  )  [static]
```

5.3.1.4 Transparent_25()

```
static const sf::Color yuki::YukiColor::Transparent_25 (
            255 ,
            255 ,
            255 ,
            192  )  [static]
```

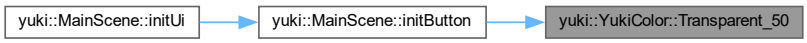
Here is the caller graph for this function:



5.3.1.5 Transparent_50()

```
static const sf::Color yuki::YukiColor::Transparent_50 (
            255 ,
            255 ,
            255 ,
            128  )  [static]
```

Here is the caller graph for this function:



5.3.1.6 Transparent_75()

```
static const sf::Color yuki::YukiColor::Transparent_75 (
            255 ,
            255 ,
            255 ,
            64  )  [static]
```

Chapter 6

Class Documentation

6.1 yuki::AttackInfo Struct Reference

```
#include <Attack.hpp>
```

Public Member Functions

- AttackInfo (ElementumType elementum_type=ElementumType::None, float elementum_level=0.0f, float damage=0.0f)

Public Attributes

- ElementumType elementum_type
- float elementum_level
- float damage

6.1.1 Constructor & Destructor Documentation

6.1.1.1 AttackInfo()

```
yuki::AttackInfo::AttackInfo (
            ElementumType elementum_type = ElementumType::None,
            float elementum_level = 0.0f,
            float damage = 0.0f )  [inline]
```

6.1.2 Member Data Documentation

6.1.2.1 damage

```
float yuki::AttackInfo::damage
```

6.1.2.2 elementum_level

```
float yuki::AttackInfo::elementum_level
```

6.1.2.3 elementum_type

```
ElementumType yuki::AttackInfo::elementum_type
```

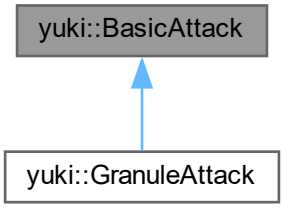
The documentation for this struct was generated from the following file:

- E:/Project/Yuki/include/Attack.hpp

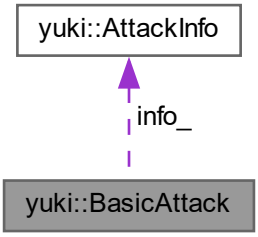
6.2 yuki::BasicAttack Class Reference

```
#include <Attack.hpp>
```

Inheritance diagram for yuki::BasicAttack:



Collaboration diagram for yuki::BasicAttack:



Public Member Functions

- BasicAttack ()=default
- BasicAttack (ElementumType elementum_type, float elementum_level, float damage)
- BasicAttack (const AttackInfo &info)
- const auto & getAttackInfo () const
- const auto getElementum () const
- const auto getDamage () const
- virtual ~BasicAttack ()

Protected Attributes

- AttackInfo info_

6.2.1 Constructor & Destructor Documentation

6.2.1.1 BasicAttack() [1/3]

```
yuki::BasicAttack::BasicAttack ( ) [default]
```

6.2.1.2 BasicAttack() [2/3]

```
yuki::BasicAttack::BasicAttack (
            ElementumType elementum_type,
            float elementum_level,
            float damage ) [inline]
```

6.2.1.3 BasicAttack() [3/3]

```
yuki::BasicAttack::BasicAttack (
            const AttackInfo & info ) [inline]
```

6.2.1.4 ~BasicAttack()

```
virtual yuki::BasicAttack::~BasicAttack ( ) [inline], [virtual]
```

6.2.2 Member Function Documentation

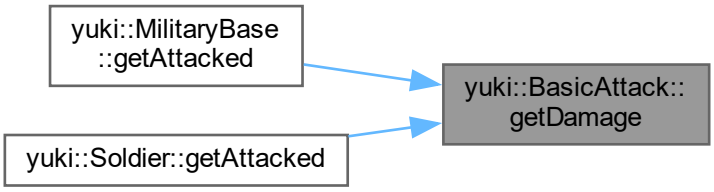
6.2.2.1 getAttackInfo()

```
const auto & yuki::BasicAttack::getAttackInfo ( ) const  [inline]
```

6.2.2.2 getDamage()

```
const auto yuki::BasicAttack::getDamage ( ) const  [inline]
```

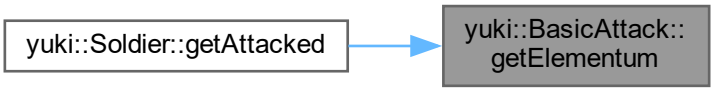
Here is the caller graph for this function:



6.2.2.3 getElementum()

```
const auto yuki::BasicAttack::getElementum ( ) const  [inline]
```

Here is the caller graph for this function:



6.2.3 Member Data Documentation

6.2.3.1 info_

```
AttackInfo yuki::BasicAttack::info_ [protected]
```

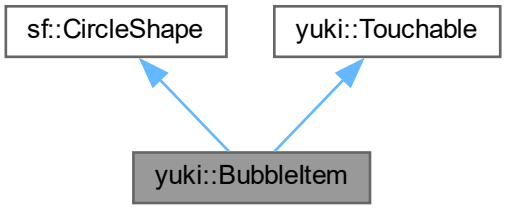
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/Attack.hpp

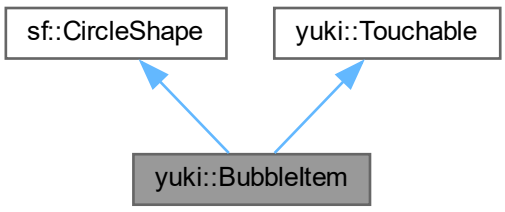
6.3 yuki::BubbleItem Class Reference

```
#include <FloatingBubble.hpp>
```

Inheritance diagram for yuki::BubbleItem:



Collaboration diagram for yuki::BubbleItem:



Public Member Functions

- BubbleItem ()=default
- BubbleItem (float radius)
- virtual ~BubbleItem ()=default
- void loadTexture (const std::string &path)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Attributes

- sf::Texture texture_

Friends

- class FloatingBubble

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.3.1 Constructor & Destructor Documentation

6.3.1.1 BubbleItem() [1/2]

```
yuki::BubbleItem::BubbleItem ( ) [default]
```

6.3.1.2 BubbleItem() [2/2]

```
yuki::BubbleItem::BubbleItem (
            float radius )  [inline]
```

6.3.1.3 ~BubbleItem()

```
virtual yuki::BubbleItem::~BubbleItem ( )  [virtual], [default]
```

6.3.2 Member Function Documentation

6.3.2.1 loadTexture()

```
void yuki::BubbleItem::loadTexture (
            const std::string & path )  [inline]
```

6.3.3 Friends And Related Function Documentation

6.3.3.1 FloatingBubble

```
friend class FloatingBubble  [friend]
```

6.3.4 Member Data Documentation

6.3.4.1 texture_

```
sf::Texture yuki::BubbleItem::texture_  [private]
```

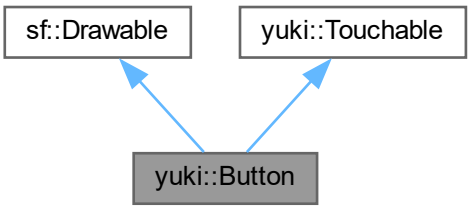
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/FloatingBubble.hpp

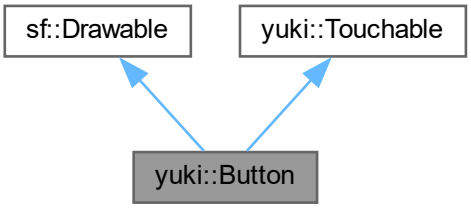
6.4 yuki::Button Class Reference

```
#include <Button.hpp>
```

Inheritance diagram for yuki::Button:



Collaboration diagram for yuki::Button:



Public Types

- enum class Alignment { Left , Center , Right }

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Public Member Functions

- Button ()
- Button (const sf::Vector2f &size)
- Button (const sf::String &text, const Alignment alignment=Alignment::Center)
- Button (const sf::Vector2f &size, const sf::String &text, const Alignment Alignment=Alignment::Center)
- virtual ~Button ()
- void setSize (const sf::Vector2f &size)
- const auto getSize () const
- void setPosition (const sf::Vector2f &position)
- void setPosition (float x, float y)
- const auto getPosition () const
- void setFillColor (const sf::Color &color)
- void setOutlineColor (const sf::Color &color)
- void setOutlineThickness (float thickness)
- void setOrigin (const sf::Vector2f &origin)
- const auto getOrigin () const
- void setTexture (const sf::Texture *texture)
- const auto getTexture () const
- void setTextureRect (const sf::IntRect &rect)
- void setString (const sf::String &text)
- const auto getString () const
- void setTextSize (unsigned int size)
- void setTextFont (const sf::Font &font)
- void setTextFillColor (const sf::Color &color)
- void setTextOutlineColor (const sf::Color &color)
- void setTextOutlineThickness (float thickness)
- void setTextAlignment (Alignment alignment)
- void autoAdjustTextSize ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Protected Member Functions

- void adjustTextPosition ()

Protected Attributes

- sf::RectangleShape shape_
- sf::Text text_
- Alignment alignment_ = Alignment::Center

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.4.1 Member Enumeration Documentation

6.4.1.1 Alignment

```
enum class yuki::Button::Alignment  [strong]
```

Enumerator

Left	
Center	
Right	

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Button() [1/4]

```
Button::Button ( )
```

6.4.2.2 Button() [2/4]

```
yuki::Button::Button (
            const sf::Vector2f & size )
```

6.4.2.3 Button() [3/4]

```
yuki::Button::Button (
            const sf::String & text,
            const Alignment alignment = Alignment::Center )
```

6.4.2.4 Button() [4/4]

```
yuki::Button::Button (
            const sf::Vector2f & size,
            const sf::String & text,
            const Alignment Alignment = Alignment::Center )
```

6.4.2.5 ~Button()

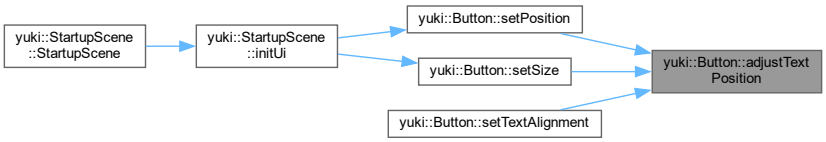
```
Button::~Button ( )  [virtual]
```

6.4.3 Member Function Documentation

6.4.3.1 adjustTextPosition()

```
void Button::adjustTextPosition ( )  [protected]
```

Here is the caller graph for this function:



6.4.3.2 autoAdjustTextSize()

```
void Button::autoAdjustTextSize ( )
```

6.4.3.3 draw()

```
void Button::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

6.4.3.4 getOrigin()

```
const auto yuki::Button::getOrigin ( ) const  [inline]
```

6.4.3.5 getPosition()

```
const auto yuki::Button::getPosition ( ) const  [inline]
```

6.4.3.6 getSize()

```
const auto yuki::Button::getSize ( ) const  [inline]
```

6.4.3.7 getString()

```
const auto yuki::Button::getString ( ) const  [inline]
```

6.4.3.8 getTexture()

```
const auto yuki::Button::getTexture ( ) const  [inline]
```

6.4.3.9 setFillColor()

```
void yuki::Button::setFillColor (
            const sf::Color & color )  [inline]
```

Here is the caller graph for this function:

6.4.3.10 setOrigin()

```
void yuki::Button::setOrigin (
            const sf::Vector2f & origin )  [inline]
```

Here is the caller graph for this function:

6.4.3.11 setOutlineColor()

```
void yuki::Button::setOutlineColor (
            const sf::Color & color )  [inline]
```

Here is the caller graph for this function:

6.4.3.12 setOutlineThickness()

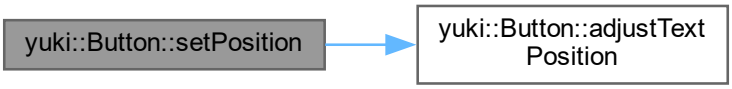
```
void yuki::Button::setOutlineThickness (
            float thickness )  [inline]
```

Here is the caller graph for this function:

6.4.3.13 setPosition() [1/2]

```
void yuki::Button::setPosition (
            const sf::Vector2f & position )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.4.3.14 setPosition() [2/2]

```
void yuki::Button::setPosition (
            float x,
            float y )  [inline]
```

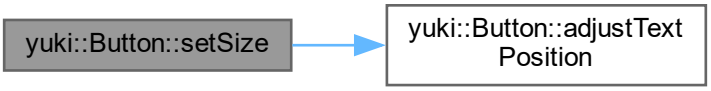
Here is the call graph for this function:

Here is the caller graph for this function:

6.4.3.15 setSize()

```
void yuki::Button::setSize (
            const sf::Vector2f & size )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.4.3.16 setString()

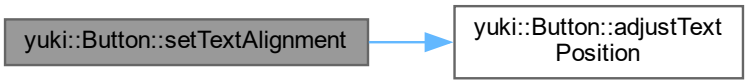
```
void yuki::Button::setString (
            const sf::String & text )  [inline]
```

Here is the caller graph for this function:

6.4.3.17 setTextAlignment()

```
void yuki::Button::setTextAlignment (
            Alignment alignment )  [inline]
```

Here is the call graph for this function:



6.4.3.18 setTextFillColor()

```
void yuki::Button::setTextFillColor (
            const sf::Color & color )  [inline]
```

Here is the caller graph for this function:

6.4.3.19 setTextFont()

```
void yuki::Button::setTextFont (
            const sf::Font & font )  [inline]
```

6.4.3.20 setTextOutlineColor()

```
void yuki::Button::setTextOutlineColor (
            const sf::Color & color )  [inline]
```

6.4.3.21 setTextOutlineThickness()

```
void yuki::Button::setTextOutlineThickness (
            float thickness )  [inline]
```

6.4.3.22 setTextSize()

```
void yuki::Button::setTextSize (
            unsigned int size )  [inline]
```

6.4.3.23 setTexture()

```
void yuki::Button::setTexture (
            const sf::Texture * texture )  [inline]
```

6.4.3.24 setTextureRect()

```
void yuki::Button::setTextureRect (
            const sf::IntRect & rect )  [inline]
```

6.4.4 Member Data Documentation

6.4.4.1 alignment_

```
Alignment yuki::Button::alignment_ = Alignment::Center  [protected]
```

6.4.4.2 shape_

```
sf::RectangleShape yuki::Button::shape_  [protected]
```

6.4.4.3 text_

```
sf::Text yuki::Button::text_  [protected]
```

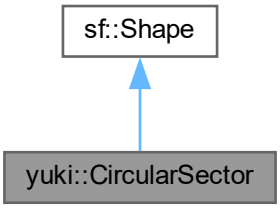
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Button.hpp
- E:/Project/Yuki/src/Button.cpp

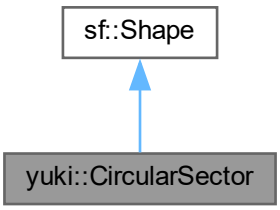
6.5 yuki::CircularSector Class Reference

```
#include <CircularSector.hpp>
```

Inheritance diagram for yuki::CircularSector:



Collaboration diagram for yuki::CircularSector:



Public Member Functions

- CircularSector (const sf::Vector2f &center, float radius, int start_angle, int end_angle)
- void setCenter (const sf::Vector2f &center)
- void setRadius (float radius)
- void setStartAngle (float start_angle)
- void setEndAngle (float end_angle)
- const sf::Vector2f & getCenter () const
- sf::Vector2f getPoint (std::size_t index) const override
- std::size_t getPointCount () const override

Private Attributes

- sf::Vector2f center_
- float radius_
- int start_angle_
- int end_angle_

6.5.1 Constructor & Destructor Documentation

6.5.1.1 CircularSector()

```
yuki::CircularSector::CircularSector (
            const sf::Vector2f & center,
            float radius,
            int start_angle,
            int end_angle )  [inline]
```

6.5.2 Member Function Documentation

6.5.2.1 getCenter()

```
const sf::Vector2f & yuki::CircularSector::getCenter ( ) const  [inline]
```

6.5.2.2 getPoint()

```
Vector2f CircularSector::getPoint (
            std::size_t index ) const  [override]
```

6.5.2.3 getPointCount()

```
std::size_t yuki::CircularSector::getPointCount ( ) const  [inline], [override]
```

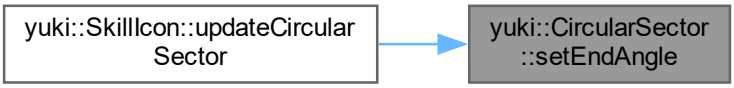
6.5.2.4 setCenter()

```
void yuki::CircularSector::setCenter (
            const sf::Vector2f & center )  [inline]
```

6.5.2.5 setEndAngle()

```
void yuki::CircularSector::setEndAngle (
            float end_angle )  [inline]
```

Here is the caller graph for this function:



6.5.2.6 setRadius()

```
void yuki::CircularSector::setRadius (
            float radius )  [inline]
```

6.5.2.7 setStartAngle()

```
void yuki::CircularSector::setStartAngle (
            float start_angle )  [inline]
```

6.5.3 Member Data Documentation

6.5.3.1 center_

```
sf::Vector2f yuki::CircularSector::center_ [private]
```

6.5.3.2 end_angle_

```
int yuki::CircularSector::end_angle_ [private]
```

6.5.3.3 radius_

```
float yuki::CircularSector::radius_ [private]
```

6.5.3.4 start_angle_

```
int yuki::CircularSector::start_angle_ [private]
```

The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/CircularSector.hpp
- E:/Project/Yuki/src/CircularSector.cpp

6.6 yuki::Elementum Struct Reference

```
#include <Elementum.hpp>
```

Public Member Functions

- Elementum (ElementumType type=ElementumType::None, float level=0.0f)

Public Attributes

- ElementumType type
- float level

6.6.1 Constructor & Destructor Documentation

6.6.1.1 Elementum()

```
yuki::Elementum::Elementum (
            ElementumType type = ElementumType::None,
            float level = 0.0f )  [inline]
```

6.6.2 Member Data Documentation

6.6.2.1 level

```
float yuki::Elementum::level
```

6.6.2.2 type

```
ElementumType yuki::Elementum::type
```

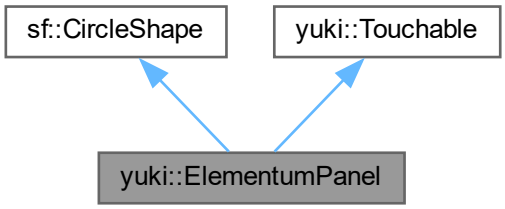
The documentation for this struct was generated from the following file:

- E:/Project/Yuki/include/Elementum.hpp

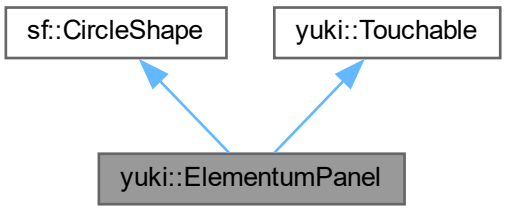
6.7 yuki::ElementumPanel Class Reference

```
#include <ElementumPanel.hpp>
```

Inheritance diagram for yuki::ElementumPanel:



Collaboration diagram for yuki::ElementumPanel:



Public Member Functions

- ElementumPanel (ElementumType type=ElementumType::Pyro)
- void activate ()
- void deactivate ()
- void setElementumType (ElementumType type)
- constexpr auto getElementumType () const

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Attributes

- ElementumType elementum_type_

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

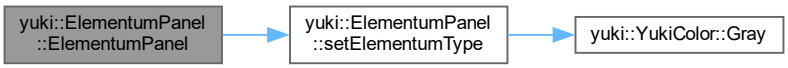
- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.7.1 Constructor & Destructor Documentation

6.7.1.1 ElementumPanel()

```
yuki::ElementumPanel::ElementumPanel (
            ElementumType type = ElementumType::Pyro )  [inline]
```

Here is the call graph for this function:



6.7.2 Member Function Documentation

6.7.2.1 activate()

```
void ElementumPanel::activate ( )
```

6.7.2.2 deactivate()

```
void ElementumPanel::deactivate ( )
```

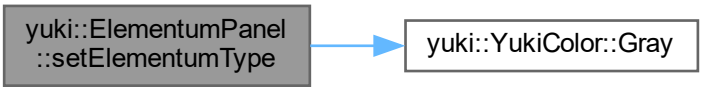
6.7.2.3 getElementumType()

```
constexpr auto yuki::ElementumPanel::getElementumType ( ) const  [inline], [constexpr]
```

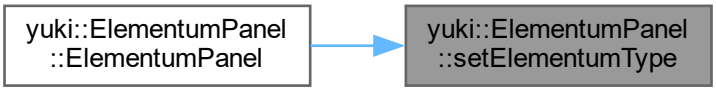
6.7.2.4 setElementumType()

```
void ElementumPanel::setElementumType (
            ElementumType type )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.7.3 Member Data Documentation

6.7.3.1 elementum_type_

```
ElementumType yuki::ElementumPanel::elementum_type_  [private]
```

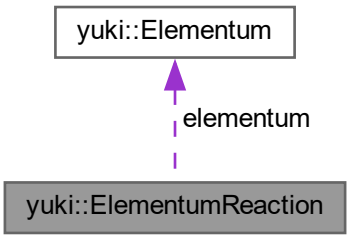
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/ElementumPanel.hpp
- E:/Project/Yuki/src/ElementumPanel.cpp

6.8 yuki::ElementumReaction Struct Reference

```
#include <Elementum.hpp>
```

Collaboration diagram for yuki::ElementumReaction:



Public Attributes

- ElementumReactionType type
- Elementum elementum

6.8.1 Member Data Documentation

6.8.1.1 elementum

```
Elementum yuki::ElementumReaction::elementum
```

6.8.1.2 type

```
ElementumReactionType yuki::ElementumReaction::type
```

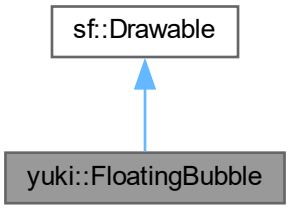
The documentation for this struct was generated from the following file:

- E:/Project/Yuki/include/Elementum.hpp

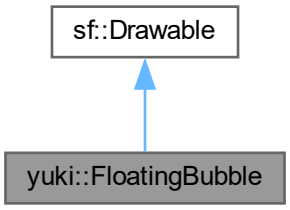
6.9 yuki::FloatingBubble Class Reference

```
#include <FloatingBubble.hpp>
```

Inheritance diagram for yuki::FloatingBubble:



Collaboration diagram for yuki::FloatingBubble:



Public Member Functions

- FloatingBubble (const sf::Vector2f &position, const float distribute_radius, const int bubble_num, const float bubble_radius=10.f, const Direction direction=Direction::Down)
- FloatingBubble ()=default
- void setPosition (const sf::Vector2f &position)
- const auto getPosition () const
- void setBubbleRadius (const float radius)
- const auto getBubbleRadius () const
- int getIndexByPosition (const sf::Vector2f &position) const
- auto & getBubbleItems ()
- const auto & getBubbleItems () const
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

Private Member Functions

- sf::Vector2f getBubblePosition (const int index)
- void updateBubblesPosition ()

Private Attributes

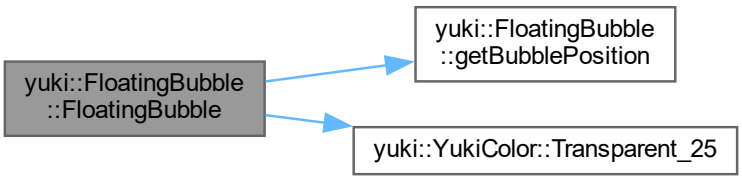
- sf::Vector2f position_
- float distribute_radius_
- int bubble_num_
- float bubble_radius_
- Direction direction_
- std::vector< std::unique_ptr< BubbleItem > > bubbles_

6.9.1 Constructor & Destructor Documentation

6.9.1.1 FloatingBubble() [1/2]

```
yuki::FloatingBubble::FloatingBubble (
            const sf::Vector2f & position,
            const float distribute_radius,
            const int bubble_num,
            const float bubble_radius = 10.f,
            const Direction direction = Direction::Down )  [inline]
```

Here is the call graph for this function:



6.9.1.2 FloatingBubble() [2/2]

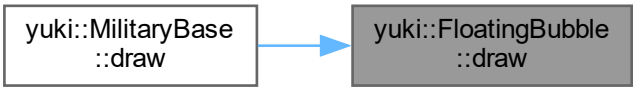
```
yuki::FloatingBubble::FloatingBubble ( )  [default]
```

6.9.2 Member Function Documentation

6.9.2.1 draw()

```
void FloatingBubble::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override]
```

Here is the caller graph for this function:



6.9.2.2 getBubbleItems() [1/2]

```
auto & yuki::FloatingBubble::getBubbleItems ( )  [inline]
```

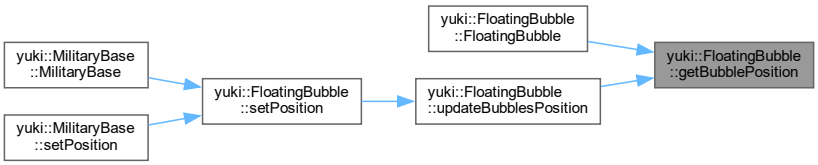
6.9.2.3 getBubbleItems() [2/2]

```
const auto & yuki::FloatingBubble::getBubbleItems ( ) const  [inline]
```

6.9.2.4 getBubblePosition()

```
Vector2f FloatingBubble::getBubblePosition (
            const int index )  [private]
```

Here is the caller graph for this function:



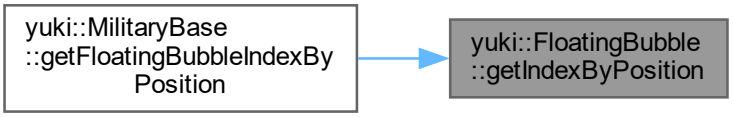
6.9.2.5 getBubbleRadius()

```
const auto yuki::FloatingBubble::getBubbleRadius ( ) const  [inline]
```

6.9.2.6 getIndexByPosition()

```
int FloatingBubble::getIndexByPosition (
            const sf::Vector2f & position ) const
```

Here is the caller graph for this function:



6.9.2.7 getPosition()

```
const auto yuki::FloatingBubble::getPosition ( ) const  [inline]
```

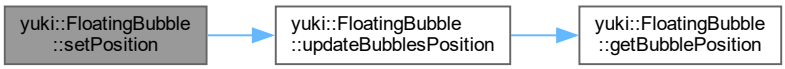
6.9.2.8 setBubbleRadius()

```
void yuki::FloatingBubble::setBubbleRadius (
            const float radius )  [inline]
```

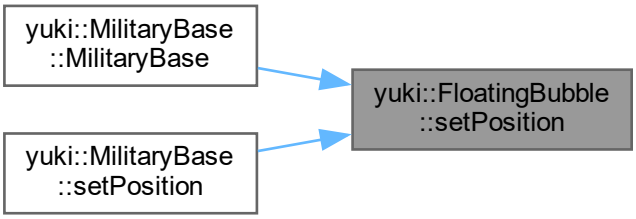
6.9.2.9 setPosition()

```
void yuki::FloatingBubble::setPosition (
            const sf::Vector2f & position )  [inline]
```

Here is the call graph for this function:



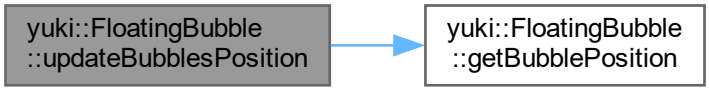
Here is the caller graph for this function:



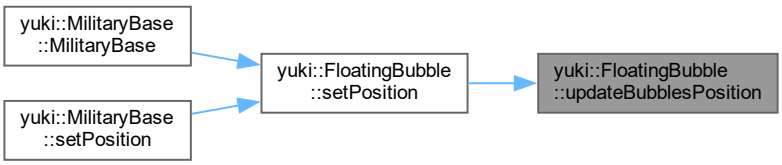
6.9.2.10 updateBubblesPosition()

```
void FloatingBubble::updateBubblesPosition ( )  [private]
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.3 Member Data Documentation

6.9.3.1 bubble_num_

```
int yuki::FloatingBubble::bubble_num_ [private]
```

6.9.3.2 bubble_radius_

```
float yuki::FloatingBubble::bubble_radius_ [private]
```

6.9.3.3 bubbles_

```
std::vector<std::unique_ptr<BubbleItem> > yuki::FloatingBubble::bubbles_ [private]
```

6.9.3.4 direction_

```
Direction yuki::FloatingBubble::direction_ [private]
```

6.9.3.5 distribute_radius_

```
float yuki::FloatingBubble::distribute_radius_ [private]
```

6.9.3.6 position_

```
sf::Vector2f yuki::FloatingBubble::position_ [private]
```

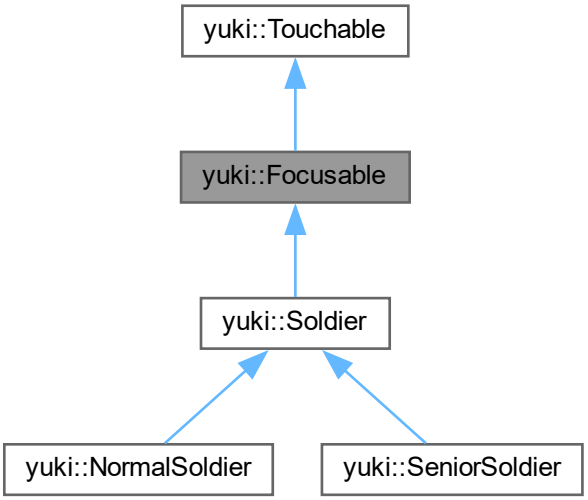
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/FloatingBubble.hpp
- E:/Project/Yuki/src/FloatingBubble.cpp

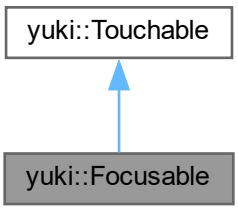
6.10 yuki::Focusable Class Reference

```
#include <InteractiveObject.hpp>
```

Inheritance diagram for yuki::Focusable:



Collaboration diagram for yuki::Focusable:



Public Types

- using Callback = std::function< void(sf::Event e)>

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Public Member Functions

- Focusable ()=default
- virtual ~Focusable ()
- virtual void onFocus (sf::Event e)
- virtual void onUnfocus (sf::Event e)
- constexpr auto isFocused () const
- void bindFocus (Callback callback)
- void bindUnfocus (Callback callback)
- void updateStatus (sf::Event event)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Attributes

- bool is_focused_ = false
- Callback on_focusCallback_ = [ ](sf::Event) {}
- Callback on_unfocusCallback_ = [ ](sf::Event) {}

Additional Inherited Members

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.10.1 Member Typedef Documentation

6.10.1.1 Callback

```
using yuki::Focusable::Callback = std::function<void(sf::Event e)>
```

6.10.2 Constructor & Destructor Documentation

6.10.2.1 Focusable()

```
yuki::Focusable::Focusable ( )  [default]
```

6.10.2.2 ~Focusable()

```
virtual yuki::Focusable::~Focusable ( )  [inline], [virtual]
```

6.10.3 Member Function Documentation

6.10.3.1 bindFocus()

```
void yuki::Focusable::bindFocus (
            Callback callback )  [inline]
```

6.10.3.2 bindUnfocus()

```
void yuki::Focusable::bindUnfocus (
            Callback callback )  [inline]
```

6.10.3.3 isFocused()

```
constexpr auto yuki::Focusable::isFocused ( ) const  [inline], [constexpr]
```

6.10.3.4 onFocus()

```
virtual void yuki::Focusable::onFocus (
            sf::Event e )  [inline], [virtual]
```

Here is the caller graph for this function:

6.10.3.5 onUnfocus()

```
virtual void yuki::Focusable::onUnfocus (
            sf::Event e )  [inline], [virtual]
```

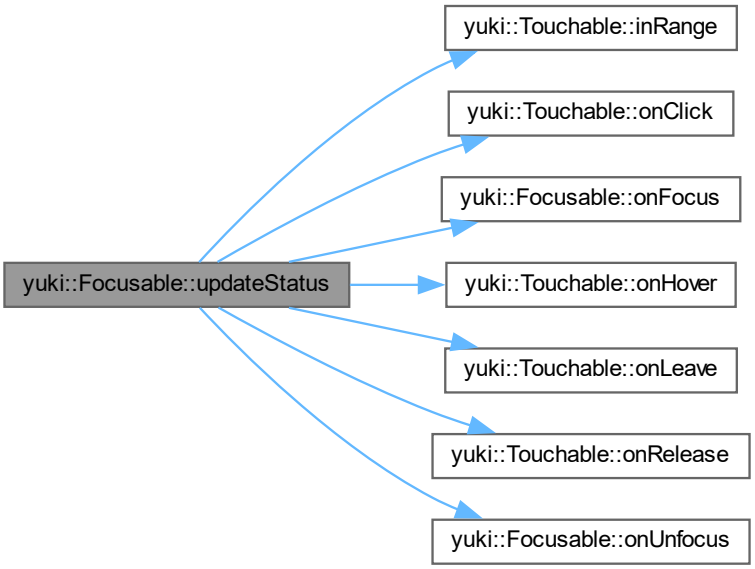
Here is the caller graph for this function:

6.10.3.6 updateStatus()

```
void Focusable::updateStatus (
            sf::Event event )  [virtual]
```

Reimplemented from yuki::Touchable.

Here is the call graph for this function:



6.10.4 Member Data Documentation

6.10.4.1 is_focused_

```
bool yuki::Focusable::is_focused_ = false  [private]
```

6.10.4.2 on_focusCallback_

```
Callback yuki::Focusable::on_focusCallback_ = [](sf::Event) {}  [private]
```

6.10.4.3 on_unfocusCallback_

```
Callback yuki::Focusable::on_unfocusCallback_ = [](sf::Event) {}  [private]
```

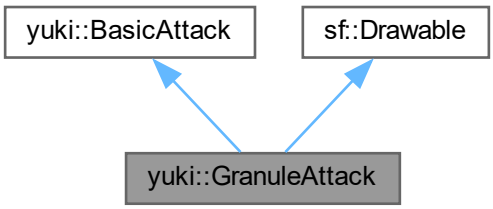
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/InteractiveObject.hpp
- E:/Project/Yuki/src/InteractiveObject.cpp

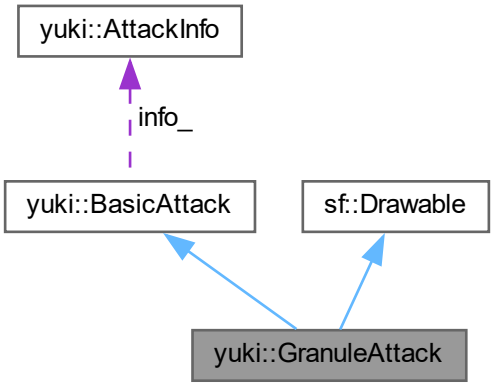
6.11 yuki::GranuleAttack Class Reference

```
#include <Attack.hpp>
```

Inheritance diagram for yuki::GranuleAttack:



Collaboration diagram for yuki::GranuleAttack:



Public Member Functions

- GranuleAttack ()
- GranuleAttack (const sf::Vector2f &position, float speed, Direction direction, const AttackInfo &attck_info)
- virtual ~GranuleAttack ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update ()
- auto getPosition () const

Public Member Functions inherited from yuki::BasicAttack

- BasicAttack ()=default
- BasicAttack (ElementumType elementum_type, float elementum_level, float damage)
- BasicAttack (const AttackInfo &info)
- const auto & getAttackInfo () const
- const auto getElementum () const
- const auto getDamage () const
- virtual ~BasicAttack ()

Protected Member Functions

- void updateGranule ()

Protected Attributes

- sf::CircleShape granule_
- float speed_
- Direction direction_

Protected Attributes inherited from yuki::BasicAttack

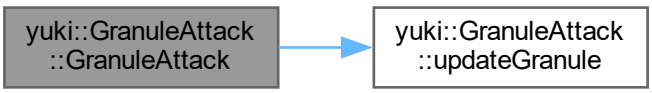
- AttackInfo info_

6.11.1 Constructor & Destructor Documentation

6.11.1.1 GranuleAttack() [1/2]

```
yuki::GranuleAttack::GranuleAttack ( ) [inline]
```

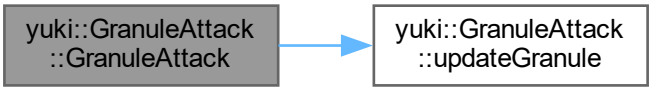
Here is the call graph for this function:



6.11.1.2 GranuleAttack() [2/2]

```
yuki::GranuleAttack::GranuleAttack (
            const sf::Vector2f & position,
            float speed,
            Direction direction,
            const AttackInfo & attck_info )  [inline]
```

Here is the call graph for this function:



6.11.1.3 ~GranuleAttack()

```
virtual yuki::GranuleAttack::~GranuleAttack ( )  [inline], [virtual]
```

6.11.2 Member Function Documentation

6.11.2.1 draw()

```
void GranuleAttack::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

6.11.2.2 getPosition()

```
auto yuki::GranuleAttack::getPosition ( ) const  [inline]
```

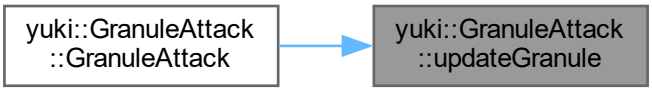
6.11.2.3 update()

```
void GranuleAttack::update ( )  [virtual]
```

6.11.2.4 updateGranule()

```
void GranuleAttack::updateGranule ( ) [protected]
```

Here is the caller graph for this function:



6.11.3 Member Data Documentation

6.11.3.1 direction_

```
Direction yuki::GranuleAttack::direction_ [protected]
```

6.11.3.2 granule_

```
sf::CircleShape yuki::GranuleAttack::granule_ [protected]
```

6.11.3.3 speed_

```
float yuki::GranuleAttack::speed_ [protected]
```

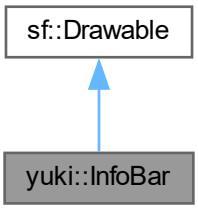
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Attack.hpp
- E:/Project/Yuki/src/Attack.cpp

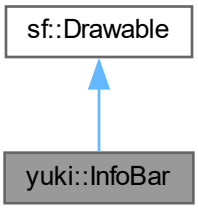
6.12 yuki::InfoBar Class Reference

```
#include <InfoBar.hpp>
```

Inheritance diagram for yuki::InfoBar:



Collaboration diagram for yuki::InfoBar:



Public Member Functions

- InfoBar ()
- InfoBar (const sf::Vector2f &position, const sf::Vector2f &size=sf::Vector2f(256.f, 32.f))
- virtual ~InfoBar ()=default
- void setMoney (const int money)
- void setPosition (const sf::Vector2f &position)
- const auto & getPosition () const
- void setSize (const sf::Vector2f &size)
- const auto & getSize () const

Private Member Functions

- void initTextures ()
- void initUi ()
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override

Private Attributes

- sf::RectangleShape background_
- sf::Texture background_texture_
- sf::Sprite icon_money_
- sf::Texture icon_money_texture_
- sf::Text text_money_

Static Private Attributes

- static constexpr float kIconSize = 32.f
- static constexpr float kMoneyIconOffset = 10.f
- static constexpr float kMoneyTextOffset = 60.f

6.12.1 Constructor & Destructor Documentation

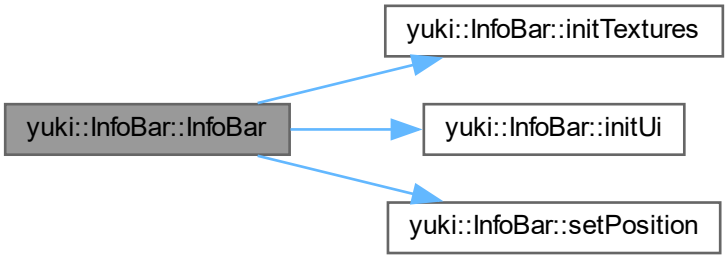
6.12.1.1 InfoBar() [1/2]

```
yuki::InfoBar::InfoBar ( ) [inline]
```

6.12.1.2 InfoBar() [2/2]

```
InfoBar::InfoBar (
            const sf::Vector2f & position,
            const sf::Vector2f & size = sf::Vector2f(256.f, 32.f) )
```

Here is the call graph for this function:



6.12.1.3 ~InfoBar()

```
virtual yuki::InfoBar::~InfoBar ( )  [virtual], [default]
```

6.12.2 Member Function Documentation

6.12.2.1 draw()

```
void InfoBar::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [private]
```

6.12.2.2 getPosition()

```
const auto & yuki::InfoBar::getPosition ( ) const  [inline]
```

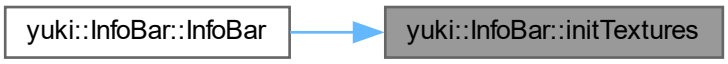
6.12.2.3 getSize()

```
const auto & yuki::InfoBar::getSize ( ) const  [inline]
```

6.12.2.4 initTextures()

```
void InfoBar::initTextures ( )  [private]
```

Here is the caller graph for this function:



6.12.2.5 initUi()

```
void InfoBar::initUi ( ) [private]
```

Here is the caller graph for this function:

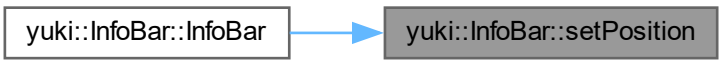
6.12.2.6 setMoney()

```
void InfoBar::setMoney (
            const int money )
```

6.12.2.7 setPosition()

```
void InfoBar::setPosition (
            const sf::Vector2f & position )
```

Here is the caller graph for this function:



6.12.2.8 setSize()

```
void yuki::InfoBar::setSize (
            const sf::Vector2f & size ) [inline]
```

6.12.3 Member Data Documentation

6.12.3.1 background_

```
sf::RectangleShape yuki::InfoBar::background_  [private]
```

6.12.3.2 background_texture_

```
sf::Texture yuki::InfoBar::background_texture_  [private]
```

6.12.3.3 icon_money_

```
sf::Sprite yuki::InfoBar::icon_money_  [private]
```

6.12.3.4 icon_money_texture_

```
sf::Texture yuki::InfoBar::icon_money_texture_  [private]
```

6.12.3.5 kIconSize

```
constexpr float yuki::InfoBar::kIconSize = 32.f  [static], [constexpr], [private]
```

6.12.3.6 kMoneyIconOffset

```
constexpr float yuki::InfoBar::kMoneyIconOffset = 10.f  [static], [constexpr], [private]
```

6.12.3.7 kMoneyTextOffset

```
constexpr float yuki::InfoBar::kMoneyTextOffset = 60.f  [static], [constexpr], [private]
```

6.12.3.8 text_money_

```
sf::Text yuki::InfoBar::text_money_ [private]
```

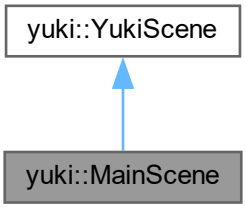
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/InfoBar.hpp
- E:/Project/Yuki/src/InfoBar.cpp

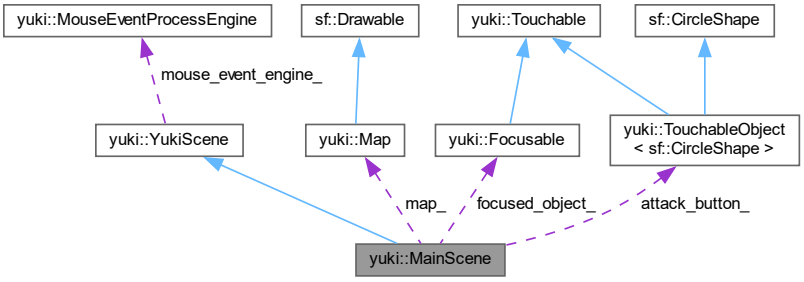
6.13 yuki::MainScene Class Reference

```
#include <MainScene.hpp>
```

Inheritance diagram for yuki::MainScene:



Collaboration diagram for yuki::MainScene:



Public Types

- enum class Message {
  GenerateOwnSoldier , GenerateEnemySoldier , OwnBaseLevelUp , FocusedObjectChanged ,
  OwnWin , EnemyWin }
- enum class ObjectType {
  OwnSoldier , EnemySoldier , OwnBase , EnemyBase ,
  None }

Public Types inherited from yuki::YukiScene

- enum class Notify { None , End }

Public Member Functions

- MainScene (sf::RenderWindow &window)
- virtual ~MainScene ()
- void sendMessage (const Message &message)

Public Member Functions inherited from yuki::YukiScene

- YukiScene (sf::RenderWindow &window)
- virtual ~YukiScene ()
- virtual int show ()

Private Types

- using GranuleAttackWithSenderInfo = std::pair< Soldier *, GranuleAttack >

Private Member Functions

- void increaseOwnMoney (int money)
- void decreaseOwnMoney (int money)
- void increaseEnemyMoney (int money)
- void decreaseEnemyMoney (int money)
- void initUi ()
- void initMap ()
- void initBuildings ()
- void initInfoHint ()
- void setMoney (int money, Camp camp=Camp::Own)
- void increaseMoney (int money, Camp camp=Camp::Own)
- void decreaseMoney (int money, Camp camp=Camp::Own)
- std::unique_ptr< yuki::Soldier > generateSoldier (const std::string &name, Camp camp=Camp::Own)
- void eraseDeadSoldier ()
- void soldierUpdatePosition ()
- void updateAttacks ()
- void enemyBaseMakeDecision ()
- sf::Vector2f coordinateToPixel (const sf::Vector2i &coordinate)
- sf::Vector2i pixelToCoordinate (const sf::Vector2f &pixel_position)
- void registerGranuleAttack (Soldier *soldier, const GranuleAttack &attack)
- void initElementumPanel ()
- void setOwnCurrentElementumType (ElementumType type)
- void setEnemyCurrentElementumType (ElementumType type)
- void initButton ()
- void implMsgGenerateOwnSoldier ()
- void implMsgGenerateEnemySoldier ()
- void normalSoldierAttack (NormalSoldier *attacker)

Private Attributes

- Map map_
- std::unique_ptr< MilitaryBase > own_base_
- std::unique_ptr< MilitaryBase > enemy_base_
- std::unique_ptr< InfoBar > info_bar_
- std::unique_ptr< SkillBar< 3 > > skill_bar_
- int own_money_
- int enemy_money_
- std::vector< std::unique_ptr< Soldier > > soldiers_
- std::vector< std::unique_ptr< Soldier > > enemies_
- Focusable * focused_object_
- ObjectType focused_object_type_
- std::vector< GranuleAttackWithSenderInfo > granule_attacks_
- std::array< ElementumPanel, 3 > elementum_panels_
- ElementumType own_current_elementum_type_
- ElementumType enemy_current_elementum_type_
- TouchableObject< sf::CircleShape > attack_button_
- sf::Texture attack_button_texture_
- std::queue< Message > message_queue_
- sf::Text debug_text_

Static Private Attributes

- static const sf::Vector2i kOwnSoldierBirthCoordinate = {3, 17}
- static const sf::Vector2i kEnemySoldierBirthCoordinate = {25, 17}
- static const sf::Vector2i kOwnBaseCoordinate = {3, 14}
- static const sf::Vector2i kEnemyBaseCoordinate = {26, 14}

Additional Inherited Members

Protected Member Functions inherited from yuki::YukiScene

- void registerTouchableObject (Touchable *object)
- void unregisterTouchableObject (Touchable *object)
- void setReturnCode (int ret_code)
- virtual void processEvent (sf::Event event)
- virtual void draw () const =0
- virtual void updateInfo ()=0

Protected Attributes inherited from yuki::YukiScene

- yuki::MouseEventProcessEngine mouse_event_engine_
- Notify notify_
- int ret_code_
- sf::RenderWindow & window_

6.13.1 Member Typedef Documentation

6.13.1.1 GranuleAttackWithSenderInfo

```
using yuki::MainScene::GranuleAttackWithSenderInfo = std::pair<Soldier*, GranuleAttack>  [private]
```

6.13.2 Member Enumeration Documentation

6.13.2.1 Message

```
enum class yuki::MainScene::Message  [strong]
```

Enumerator

GenerateOwnSoldier	
GenerateEnemySoldier	
OwnBaseLevelUp	
FocusedObjectChanged	
OwnWin	
EnemyWin	

6.13.2.2 ObjectType

```
enum class yuki::MainScene::ObjectType  [strong]
```

Enumerator

OwnSoldier	
EnemySoldier	
OwnBase	
EnemyBase	
None	

6.13.3 Constructor & Destructor Documentation

6.13.3.1 MainScene()

```
MainScene::MainScene (
            sf::RenderWindow & window )
```

6.13.3.2 ~MainScene()

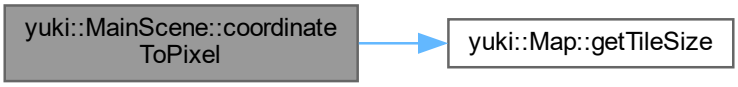
```
virtual yuki::MainScene::~MainScene ( )  [inline], [virtual]
```

6.13.4 Member Function Documentation

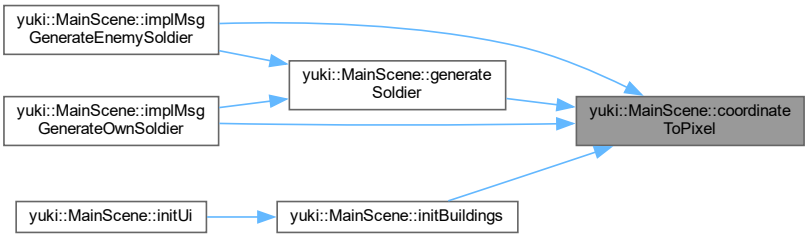
6.13.4.1 coordinateToPixel()

```
sf::Vector2f MainScene::coordinateToPixel (
            const sf::Vector2i & coordinate )  [private]
```

Here is the call graph for this function:



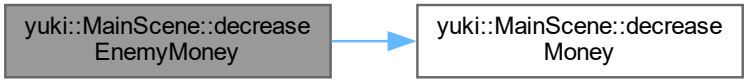
Here is the caller graph for this function:



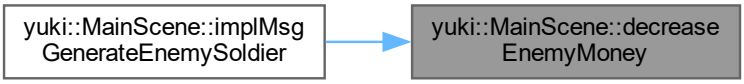
6.13.4.2 decreaseEnemyMoney()

```
void yuki::MainScene::decreaseEnemyMoney (
            int money )  [inline], [private]
```

Here is the call graph for this function:



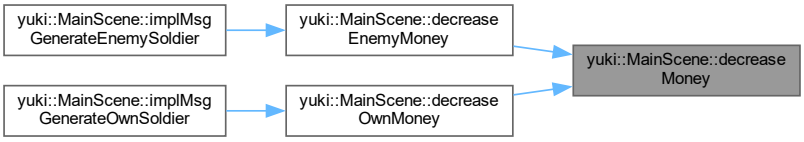
Here is the caller graph for this function:



6.13.4.3 decreaseMoney()

```
void MainScene::decreaseMoney (
            int money,
            Camp camp = Camp::Own )  [private]
```

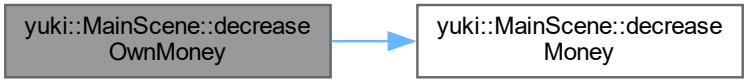
Here is the caller graph for this function:



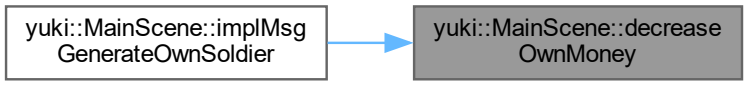
6.13.4.4 decreaseOwnMoney()

```
void yuki::MainScene::decreaseOwnMoney (
            int money )  [inline], [private]
```

Here is the call graph for this function:



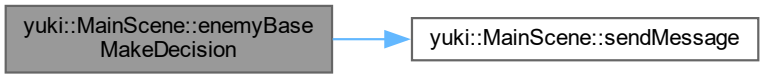
Here is the caller graph for this function:



6.13.4.5 enemyBaseMakeDecision()

```
void MainScene::enemyBaseMakeDecision ( )  [private]
```

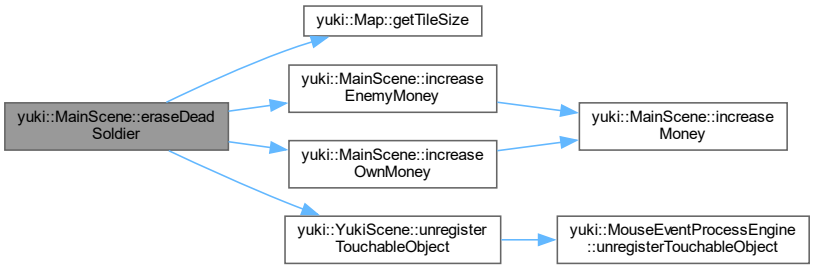
Here is the call graph for this function:



6.13.4.6 eraseDeadSoldier()

```
void MainScene::eraseDeadSoldier ( ) [private]
```

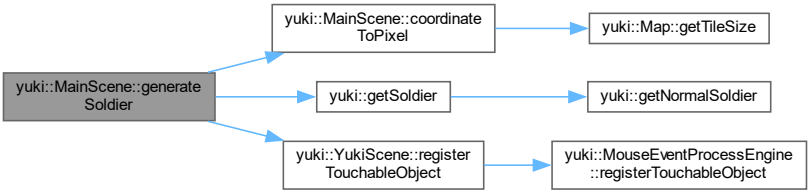
Here is the call graph for this function:



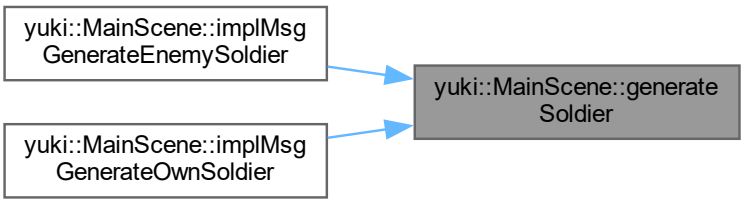
6.13.4.7 generateSoldier()

```
std::unique_ptr< Soldier > MainScene::generateSoldier (
            const std::string & name,
            Camp camp = Camp::Own ) [private]
```

Here is the call graph for this function:



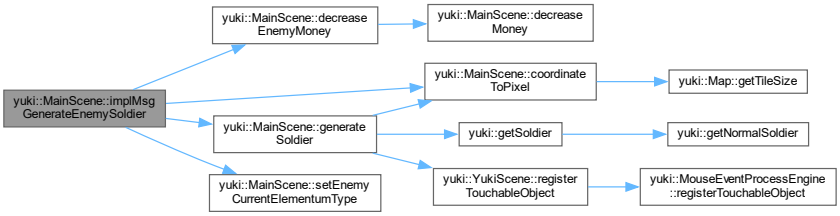
Here is the caller graph for this function:



6.13.4.8 implMsgGenerateEnemySoldier()

```
void MainScene::implMsgGenerateEnemySoldier ( ) [private]
```

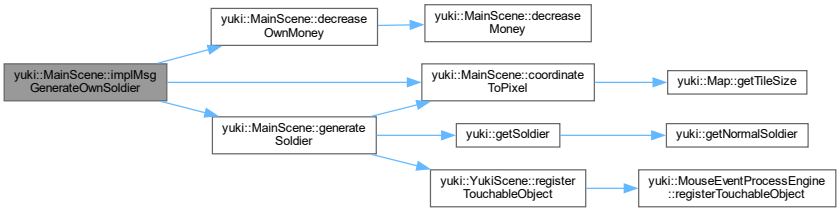
Here is the call graph for this function:



6.13.4.9 implMsgGenerateOwnSoldier()

```
void MainScene::implMsgGenerateOwnSoldier ( ) [private]
```

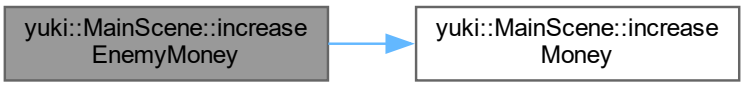
Here is the call graph for this function:



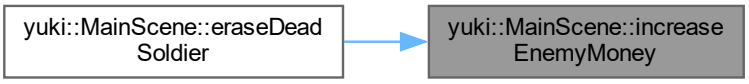
6.13.4.10 increaseEnemyMoney()

```
void yuki::MainScene::increaseEnemyMoney (
            int money ) [inline], [private]
```

Here is the call graph for this function:



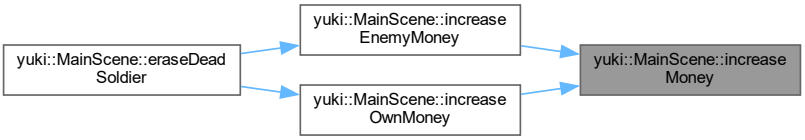
Here is the caller graph for this function:



6.13.4.11 increaseMoney()

```
void MainScene::increaseMoney (
            int money,
            Camp camp = Camp::Own )  [private]
```

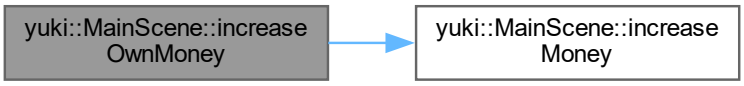
Here is the caller graph for this function:



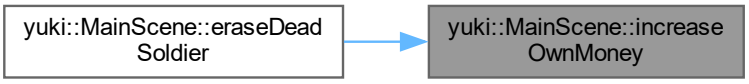
6.13.4.12 increaseOwnMoney()

```
void yuki::MainScene::increaseOwnMoney (
            int money )  [inline], [private]
```

Here is the call graph for this function:



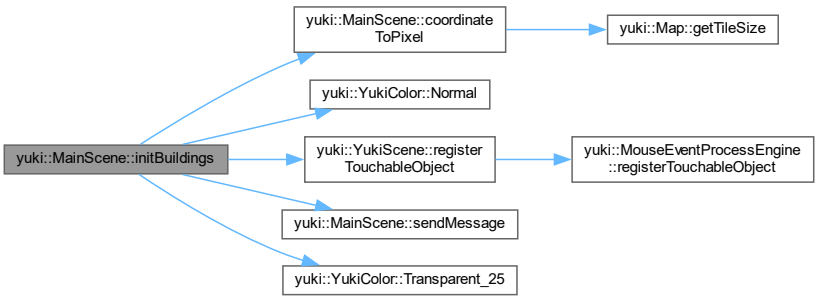
Here is the caller graph for this function:



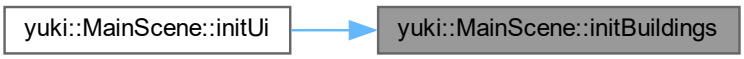
6.13.4.13 initBuildings()

```
void MainScene::initBuildings ( ) [private]
```

Here is the call graph for this function:



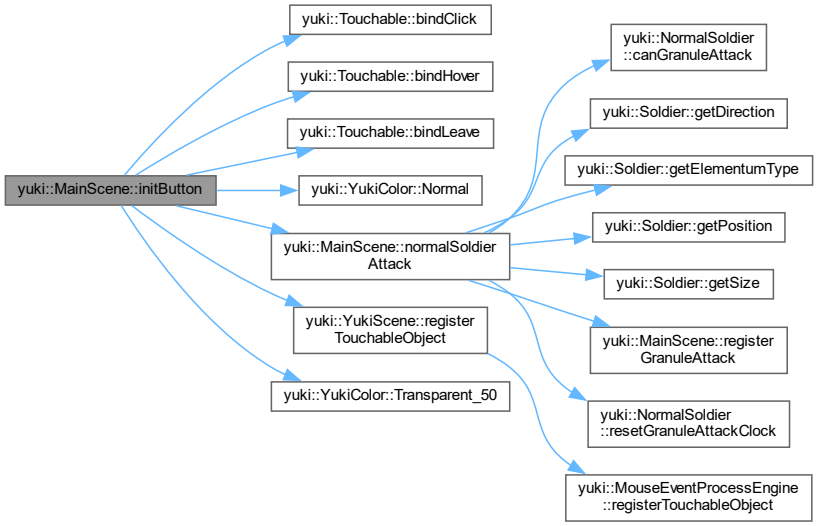
Here is the caller graph for this function:



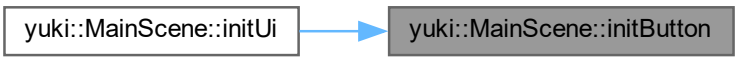
6.13.4.14 initButton()

```
void MainScene::initButton ( ) [private]
```

Here is the call graph for this function:



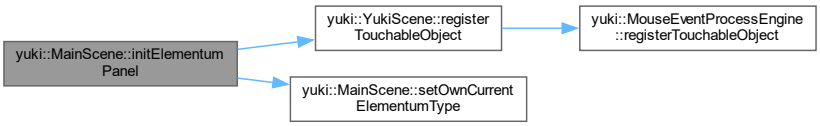
Here is the caller graph for this function:



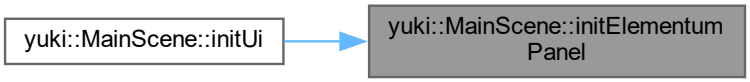
6.13.4.15 initElementumPanel()

```
void MainScene::initElementumPanel ( ) [private]
```

Here is the call graph for this function:



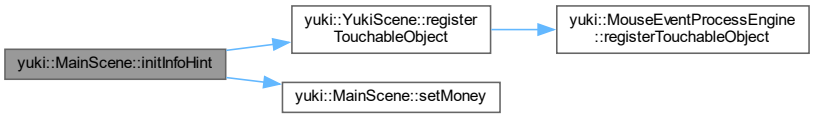
Here is the caller graph for this function:



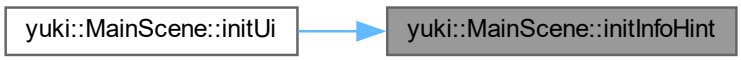
6.13.4.16 initInfoHint()

```
void MainScene::initInfoHint ( ) [private]
```

Here is the call graph for this function:



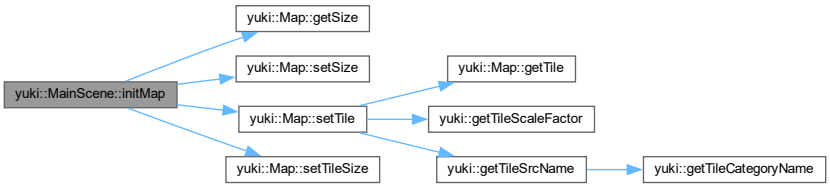
Here is the caller graph for this function:



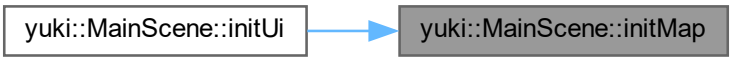
6.13.4.17 initMap()

```
void MainScene::initMap ( ) [private]
```

Here is the call graph for this function:



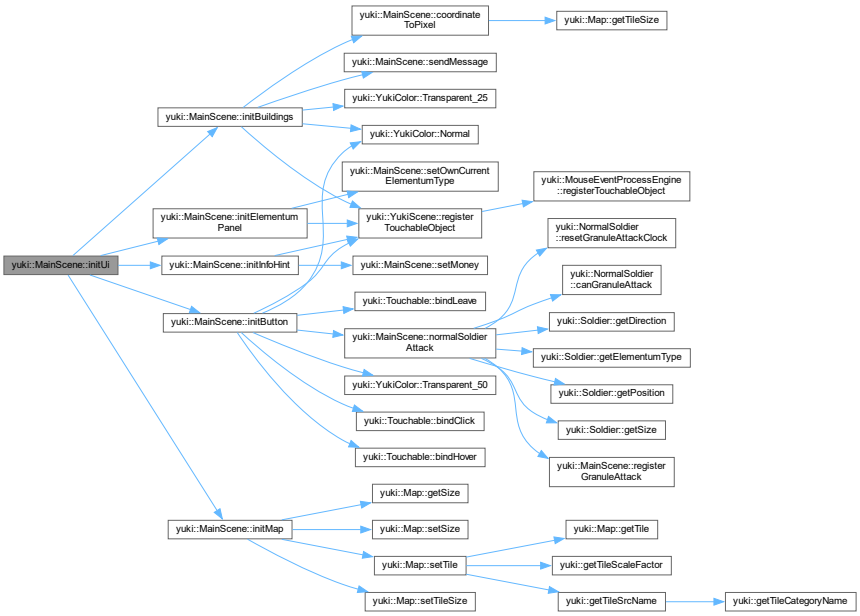
Here is the caller graph for this function:



6.13.4.18 initUi()

```
void MainScene::initUi ( ) [private]
```

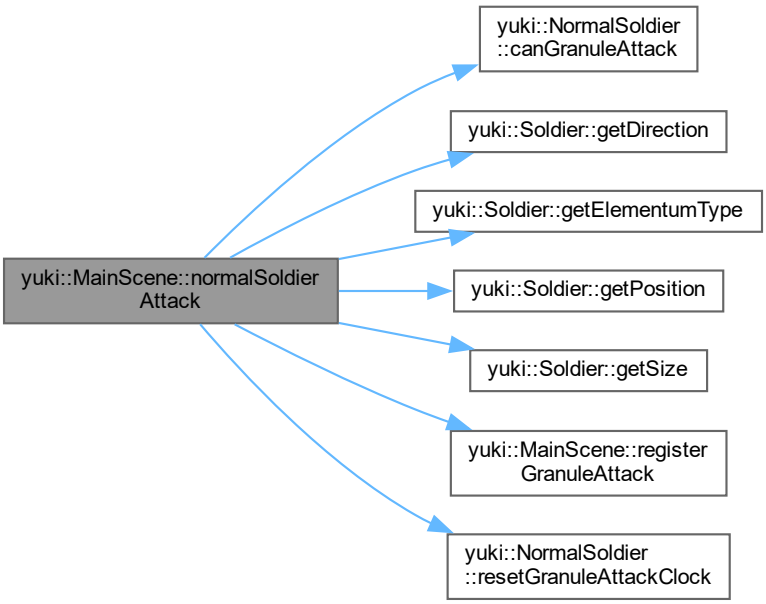
Here is the call graph for this function:



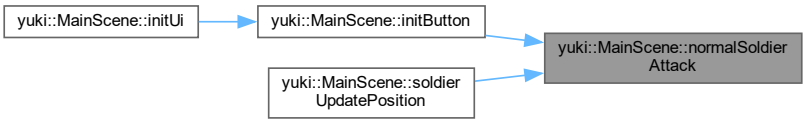
6.13.4.19 normalSoldierAttack()

```
void MainScene::normalSoldierAttack (
            NormalSoldier * attacker )  [private]
```

Here is the call graph for this function:



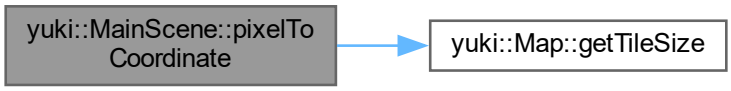
Here is the caller graph for this function:



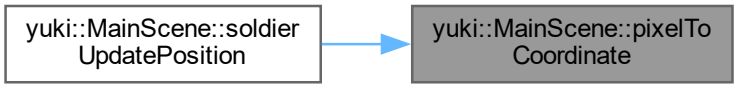
6.13.4.20 pixelToCoordinate()

```
sf::Vector2i MainScene::pixelToCoordinate (
            const sf::Vector2f & pixel_position )  [private]
```

Here is the call graph for this function:



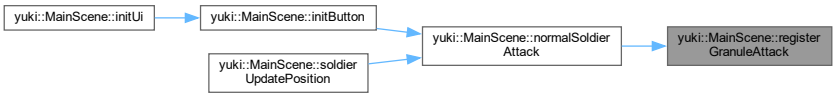
Here is the caller graph for this function:



6.13.4.21 registerGranuleAttack()

```
void yuki::MainScene::registerGranuleAttack (
            Soldier * soldier,
            const GranuleAttack & attack )  [inline], [private]
```

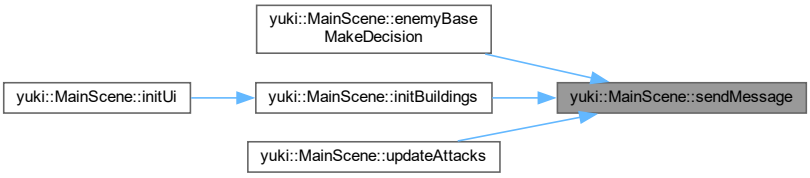
Here is the caller graph for this function:



6.13.4.22 sendMessage()

```
void yuki::MainScene::sendMessage (
            const Message & message )  [inline]
```

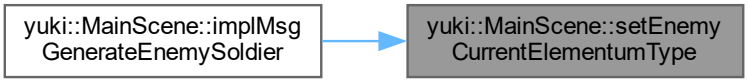
Here is the caller graph for this function:



6.13.4.23 setEnemyCurrentElementumType()

```
void MainScene::setEnemyCurrentElementumType (
            ElementumType type )  [private]
```

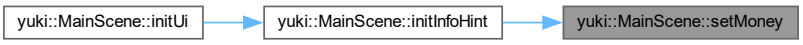
Here is the caller graph for this function:



6.13.4.24 setMoney()

```
void MainScene::setMoney (
            int money,
            Camp camp = Camp::Own )  [private]
```

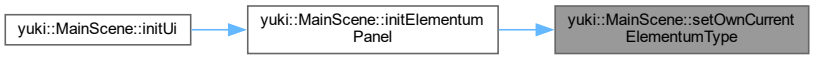
Here is the caller graph for this function:



6.13.4.25 setOwnCurrentElementumType()

```
void MainScene::setOwnCurrentElementumType (
            ElementumType type ) [private]
```

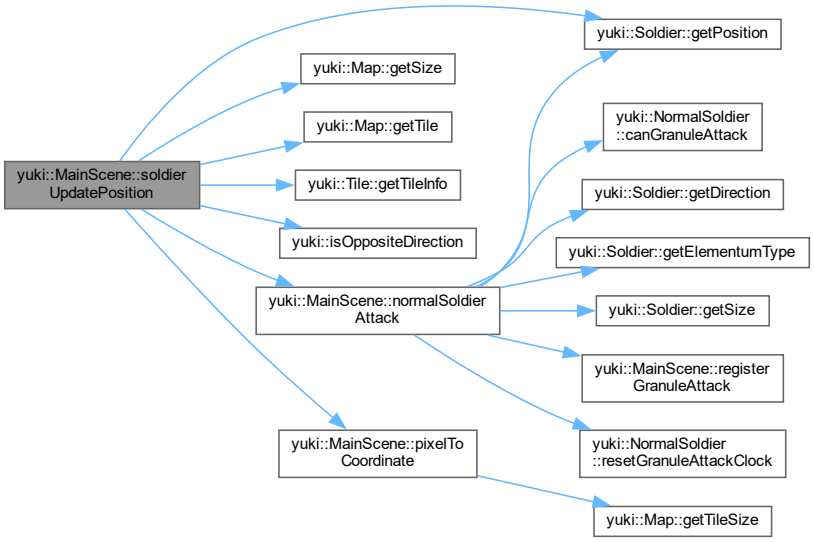
Here is the caller graph for this function:



6.13.4.26 soldierUpdatePosition()

```
void MainScene::soldierUpdatePosition ( ) [private]
```

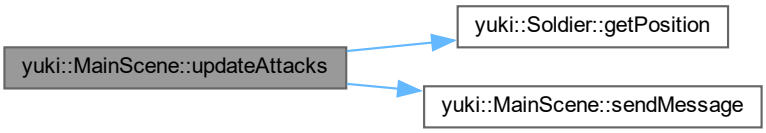
Here is the call graph for this function:



6.13.4.27 updateAttacks()

```
void MainScene::updateAttacks ( ) [private]
```

Here is the call graph for this function:



6.13.5 Member Data Documentation

6.13.5.1 attack_button_

```
TouchableObject<sf::CircleShape> yuki::MainScene::attack_button_ [private]
```

6.13.5.2 attack_button_texture_

```
sf::Texture yuki::MainScene::attack_button_texture_ [private]
```

6.13.5.3 debug_text_

```
sf::Text yuki::MainScene::debug_text_ [private]
```

6.13.5.4 elementum_panels_

```
std::array<ElementumPanel, 3> yuki::MainScene::elementum_panels_ [private]
```

6.13.5.5 enemies_

```
std::vector<std::unique_ptr<Soldier> > yuki::MainScene::enemies_ [private]
```

6.13.5.6 enemy_base_

```
std::unique_ptr<MilitaryBase> yuki::MainScene::enemy_base_ [private]
```

6.13.5.7 enemy_current_elementum_type_

```
ElementumType yuki::MainScene::enemy_current_elementum_type_ [private]
```

6.13.5.8 enemy_money_

```
int yuki::MainScene::enemy_money_ [private]
```

6.13.5.9 focused_object_

```
Focusable* yuki::MainScene::focused_object_ [private]
```

6.13.5.10 focused_object_type_

```
ObjectType yuki::MainScene::focused_object_type_ [private]
```

6.13.5.11 granule_attacks_

```
std::vector<GranuleAttackWithSenderInfo> yuki::MainScene::granule_attacks_ [private]
```

6.13.5.12 info_bar_

```
std::unique_ptr<InfoBar> yuki::MainScene::info_bar_ [private]
```

6.13.5.13 kEnemyBaseCoordinate

```
const sf::Vector2i MainScene::kEnemyBaseCoordinate = {26, 14}  [static], [private]
```

6.13.5.14 kEnemySoldierBirthCoordinate

```
const sf::Vector2i MainScene::kEnemySoldierBirthCoordinate = {25, 17}  [static], [private]
```

6.13.5.15 kOwnBaseCoordinate

```
const sf::Vector2i MainScene::kOwnBaseCoordinate = {3, 14}  [static], [private]
```

6.13.5.16 kOwnSoldierBirthCoordinate

```
const sf::Vector2i MainScene::kOwnSoldierBirthCoordinate = {3, 17}  [static], [private]
```

6.13.5.17 map_

```
Map yuki::MainScene::map_  [private]
```

6.13.5.18 message_queue_

```
std::queue<Message> yuki::MainScene::message_queue_  [private]
```

6.13.5.19 own_base_

```
std::unique_ptr<MilitaryBase> yuki::MainScene::own_base_  [private]
```

6.13.5.20 own_current_elementum_type_

```
ElementumType yuki::MainScene::own_current_elementum_type_  [private]
```

6.13.5.21 own_money_

```
int yuki::MainScene::own_money_ [private]
```

6.13.5.22 skill_bar_

```
std::unique_ptr<SkillBar<3> > yuki::MainScene::skill_bar_ [private]
```

6.13.5.23 soldiers_

```
std::vector<std::unique_ptr<Soldier> > yuki::MainScene::soldiers_ [private]
```

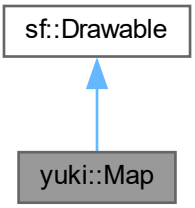
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/MainScene.hpp
- E:/Project/Yuki/src/MainScene.cpp

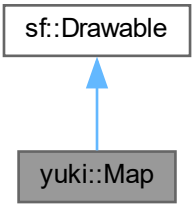
6.14 yuki::Map Class Reference

```
#include <Map.hpp>
```

Inheritance diagram for yuki::Map:



Collaboration diagram for yuki::Map:



Public Member Functions

- Map (const sf::Vector2i &size, const sf::Vector2f &tile_size)
- virtual ~Map ()
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- void setSize (const sf::Vector2i &size)
- const auto & getSize () const
- void setTileSize (const sf::Vector2f &tile_size)
- const auto & getTileSize () const
- auto getCoordinate (const sf::Vector2f &position) const
- Tile & getTile (int row, int col)
- const Tile & getTile (int row, int col) const
- Tile & getTile (const sf::Vector2i &coordinate)
- const Tile & getTile (const sf::Vector2i &coordinate) const
- void setTile (int row, int col, const TileInfo &tile_info)

Private Attributes

- sf::Vector2i size_
- sf::Vector2f tile_size_
- std::vector< Tile > tiles_
- sf::Texture bg_texture_
- std::vector< sf::Sprite > bg_tiles_

6.14.1 Constructor & Destructor Documentation

6.14.1.1 Map()

```
Map::Map (
            const sf::Vector2i & size,
            const sf::Vector2f & tile_size )
```

6.14.1.2 ~Map()

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

6.14.2 Member Function Documentation

6.14.2.1 draw()

```
void Map::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override]
```

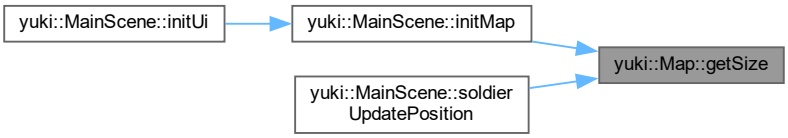
6.14.2.2 getCoordinate()

```
auto yuki::Map::getCoordinate (
            const sf::Vector2f & position ) const  [inline]
```

6.14.2.3 getSize()

```
const auto & yuki::Map::getSize ( ) const  [inline]
```

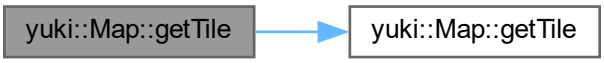
Here is the caller graph for this function:



6.14.2.4 getTile() [1/4]

```
Tile & Map::getTile (
            const sf::Vector2i & coordinate )
```

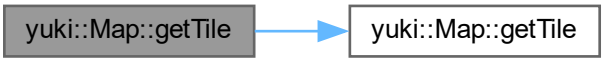
Here is the call graph for this function:



6.14.2.5 getTile() [2/4]

```
const Tile & Map::getTile (
            const sf::Vector2i & coordinate ) const
```

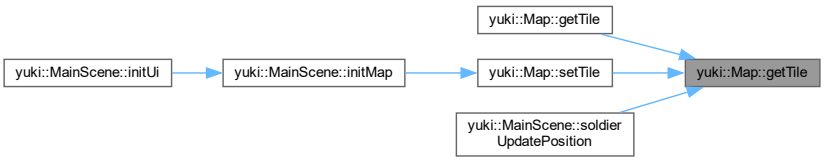
Here is the call graph for this function:



6.14.2.6 getTile() [3/4]

```
Tile & Map::getTile (
            int row,
            int col )
```

Here is the caller graph for this function:



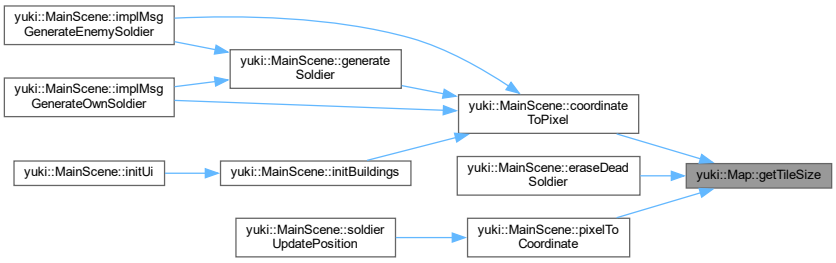
6.14.2.7 getTile() [4/4]

```
const Tile & Map::getTile (
            int row,
            int col ) const
```

6.14.2.8 getTileSize()

```
const auto & yuki::Map::getTileSize ( ) const  [inline]
```

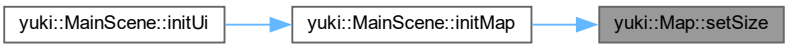
Here is the caller graph for this function:



6.14.2.9 setSize()

```
void yuki::Map::setSize (
            const sf::Vector2i & size )  [inline]
```

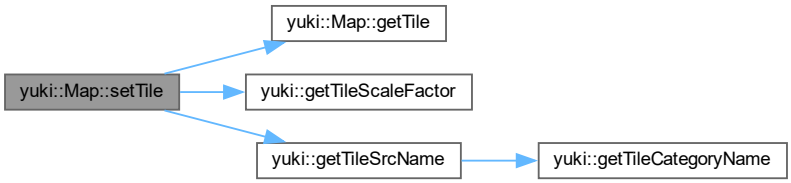
Here is the caller graph for this function:



6.14.2.10 setTile()

```
void Map::setTile (
            int row,
            int col,
            const TileInfo & tile_info )
```

Here is the call graph for this function:

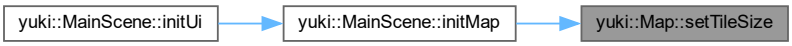


Here is the caller graph for this function:

6.14.2.11 setTileSize()

```
void yuki::Map::setTileSize (
            const sf::Vector2f & tile_size )  [inline]
```

Here is the caller graph for this function:



6.14.3 Member Data Documentation

6.14.3.1 bg_texture_

```
sf::Texture yuki::Map::bg_texture_  [private]
```

6.14.3.2 bg_tiles_

```
std::vector<sf::Sprite> yuki::Map::bg_tiles_  [private]
```

6.14.3.3 size_

```
sf::Vector2i yuki::Map::size_  [private]
```

6.14.3.4 tile_size_

```
sf::Vector2f yuki::Map::tile_size_ [private]
```

6.14.3.5 tiles_

```
std::vector<Tile> yuki::Map::tiles_ [private]
```

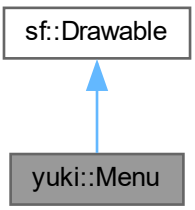
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Map.hpp
- E:/Project/Yuki/src/Map.cpp

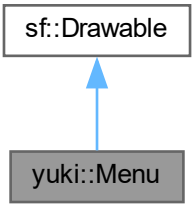
6.15 yuki::Menu Class Reference

```
#include <Menu.hpp>
```

Inheritance diagram for yuki::Menu:



Collaboration diagram for yuki::Menu:



Public Types

- enum class Layout { Vertical , Horizontal }

Public Member Functions

- Menu ()
- virtual ~Menu ()
- void setLayout (Layout layout)
- const Layout & getLayout () const
- void addItem (std::unique_ptr< yuki::Button > item)
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const
- Button & operator[ ] (unsigned int index)
- Button & getByName (const sf::String &name)

Private Member Functions

- void adjustItemsPosition ()

Private Attributes

- std::vector< std::unique_ptr< yuki::Button > > menu_items_
- Layout layout_
- unsigned int spacing_
- sf::Vector2f position_

6.15.1 Member Enumeration Documentation

6.15.1.1 Layout

```
enum class yuki::Menu::Layout  [strong]
```

Enumerator

Vertical	
Horizontal	

6.15.2 Constructor & Destructor Documentation

6.15.2.1 Menu()

```
Menu::Menu ( )
```

6.15.2.2 ~Menu()

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

6.15.3 Member Function Documentation

6.15.3.1 addItem()

```
void Menu::addItem (
            std::unique_ptr< yuki::Button > item )
```

6.15.3.2 adjustItemsPosition()

```
void Menu::adjustItemsPosition ( )  [private]
```

6.15.3.3 draw()

```
void Menu::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [virtual]
```

6.15.3.4 getByName()

```
Button & yuki::Menu::getByName (
            const sf::String & name )  [inline]
```

6.15.3.5 getLayout()

```
const Layout & yuki::Menu::getLayout ( ) const  [inline]
```

6.15.3.6 operator[]()

```
Button & yuki::Menu::operator[] (
            unsigned int index )  [inline]
```

6.15.3.7 setLayout()

```
void yuki::Menu::setLayout (
            Layout layout )  [inline]
```

6.15.4 Member Data Documentation

6.15.4.1 layout_

```
Layout yuki::Menu::layout_  [private]
```

6.15.4.2 menu_items_

```
std::vector<std::unique_ptr<yuki::Button> > yuki::Menu::menu_items_  [private]
```

6.15.4.3 position_

```
sf::Vector2f yuki::Menu::position_  [private]
```

6.15.4.4 spacing_

```
unsigned int yuki::Menu::spacing_  [private]
```

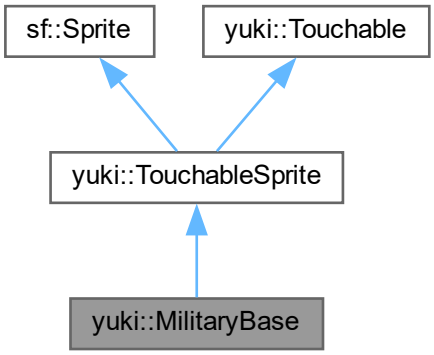
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Menu.hpp
- E:/Project/Yuki/src/Menu.cpp

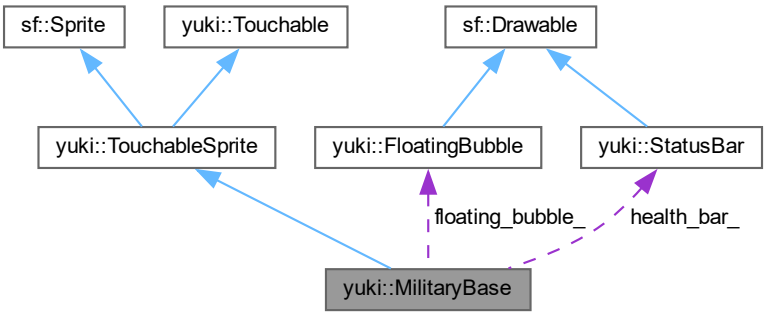
6.16 yuki::MilitaryBase Class Reference

```
#include <MilitaryBase.hpp>
```

Inheritance diagram for yuki::MilitaryBase:



Collaboration diagram for yuki::MilitaryBase:



Public Member Functions

- MilitaryBase (Camp camp=Camp::Own)
- virtual ~MilitaryBase ()
- void levelUp ()
- void levelDown ()
- auto getLevel () const
- void setCurrentHealth (int health)
- void setMaxHealth (int health)
- constexpr auto getCurrentHealth () const

- constexpr auto getMaxHealth () const
- void setPosition (const sf::Vector2f &position)
- void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- void setFloatingBubbleVisible (bool visible)
- constexpr auto getFloatingBubbleVisible () const
- int getFloatingBubbleIndexByPosition (const sf::Vector2f &position)
- auto & getFloatingBubble ()
- const auto & getFloatingBubble () const
- bool canGenerateSoldier ()
- void resetGenerateSoldierClock ()
- void getAttacked (const BasicAttack &attack)

Public Member Functions inherited from yuki::TouchableSprite

- TouchableSprite ()=default
- TouchableSprite (const sf::Texture &texture)
- TouchableSprite (const sf::Texture &texture, const sf::IntRect &rect)
- virtual ~TouchableSprite ()

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Member Functions

- void initTexture ()
- void updateTexture ()

Private Attributes

- Camp camp_
- int level_ = 1
- sf::Texture texture_
- FloatingBubble floating_bubble_
- bool floating_bubble_visible_ = false
- StatusBar health_bar_
- sf::Clock generateSoldierClock_

Static Private Attributes

- static constexpr float health_bar_offset_y = -10.f

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

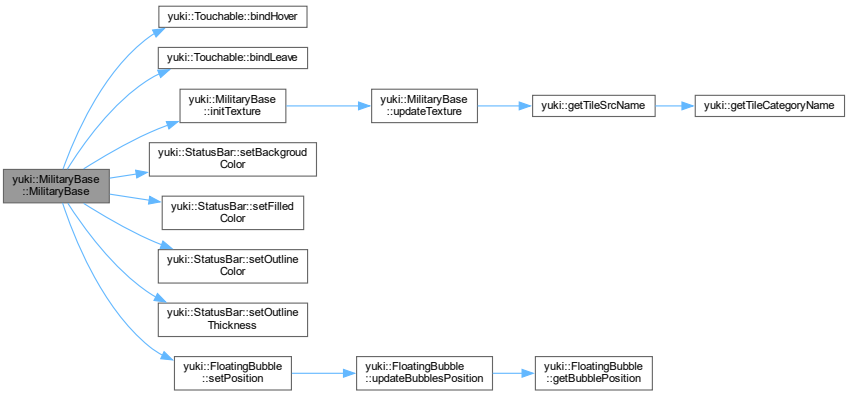
- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.16.1 Constructor & Destructor Documentation

6.16.1.1 MilitaryBase()

```
MilitaryBase::MilitaryBase (
            Camp camp = Camp::Own )
```

Here is the call graph for this function:



6.16.1.2 ~MilitaryBase()

```
virtual yuki::MilitaryBase::~MilitaryBase ( )  [inline], [virtual]
```

6.16.2 Member Function Documentation

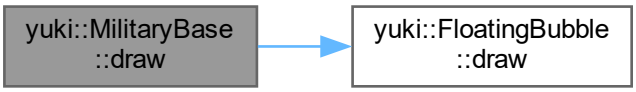
6.16.2.1 canGenerateSoldier()

```
bool MilitaryBase::canGenerateSoldier ( )
```

6.16.2.2 draw()

```
void MilitaryBase::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override]
```

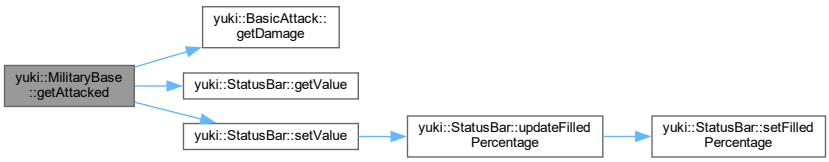
Here is the call graph for this function:



6.16.2.3 getAttacked()

```
void yuki::MilitaryBase::getAttacked (
            const BasicAttack & attack )  [inline]
```

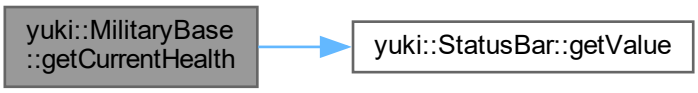
Here is the call graph for this function:



6.16.2.4 getCurrentHealth()

```
constexpr auto yuki::MilitaryBase::getCurrentHealth ( ) const  [inline], [constexpr]
```

Here is the call graph for this function:



6.16.2.5 getFloatingBubble() [1/2]

```
auto & yuki::MilitaryBase::getFloatingBubble ( )  [inline]
```

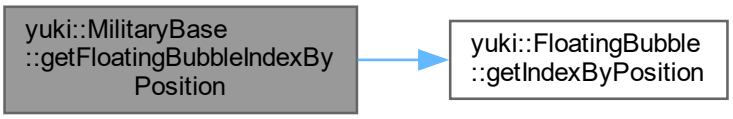
6.16.2.6 getFloatingBubble() [2/2]

```
const auto & yuki::MilitaryBase::getFloatingBubble ( ) const  [inline]
```

6.16.2.7 getFloatingBubbleIndexByPosition()

```
int yuki::MilitaryBase::getFloatingBubbleIndexByPosition (
            const sf::Vector2f & position )  [inline]
```

Here is the call graph for this function:



6.16.2.8 getFloatingBubbleVisible()

```
constexpr auto yuki::MilitaryBase::getFloatingBubbleVisible ( ) const  [inline], [constexpr]
```

6.16.2.9 getLevel()

```
auto yuki::MilitaryBase::getLevel ( ) const  [inline]
```

6.16.2.10 getMaxHealth()

```
constexpr auto yuki::MilitaryBase::getMaxHealth ( ) const  [inline], [constexpr]
```

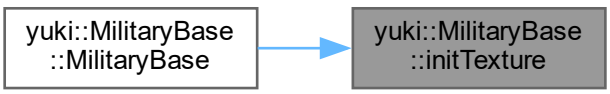
Here is the call graph for this function:

6.16.2.11 initTexture()

```
void MilitaryBase::initTexture ( )  [private]
```

Here is the call graph for this function:

Here is the caller graph for this function:



6.16.2.12 levelDown()

```
void yuki::MilitaryBase::levelDown ( ) [inline]
```

Here is the call graph for this function:

6.16.2.13 levelUp()

```
void yuki::MilitaryBase::levelUp ( ) [inline]
```

Here is the call graph for this function:

6.16.2.14 resetGenerateSoldierClock()

```
void yuki::MilitaryBase::resetGenerateSoldierClock ( ) [inline]
```

6.16.2.15 setCurrentHealth()

```
void yuki::MilitaryBase::setCurrentHealth (
            int health ) [inline]
```

Here is the call graph for this function:

6.16.2.16 setFloatingBubbleVisible()

```
void yuki::MilitaryBase::setFloatingBubbleVisible (
            bool visible )  [inline]
```

6.16.2.17 setMaxHealth()

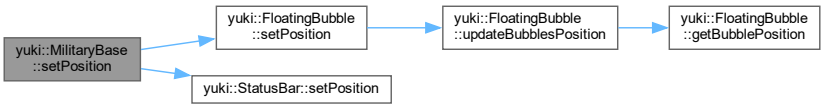
```
void yuki::MilitaryBase::setMaxHealth (
            int health )  [inline]
```

Here is the call graph for this function:

6.16.2.18 setPosition()

```
void yuki::MilitaryBase::setPosition (
            const sf::Vector2f & position )  [inline]
```

Here is the call graph for this function:

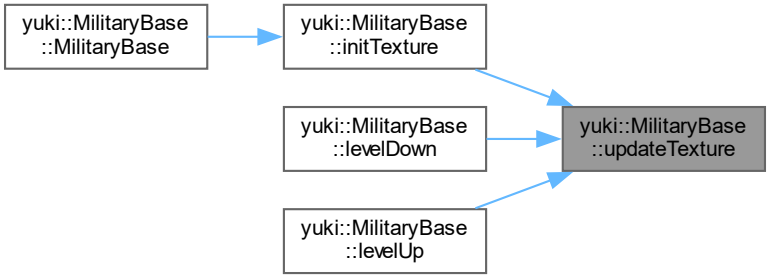


6.16.2.19 updateTexture()

```
void MilitaryBase::updateTexture ( )  [private]
```

Here is the call graph for this function:

Here is the caller graph for this function:



6.16.3 Member Data Documentation

6.16.3.1 camp_

```
Camp yuki::MilitaryBase::camp_  [private]
```

6.16.3.2 floating_bubble_

```
FloatingBubble yuki::MilitaryBase::floating_bubble_  [private]
```

6.16.3.3 floating_bubble_visible_

```
bool yuki::MilitaryBase::floating_bubble_visible_ = false  [private]
```

6.16.3.4 generateSoldierClock_

```
sf::Clock yuki::MilitaryBase::generateSoldierClock_  [private]
```

6.16.3.5 health_bar_

```
StatusBar yuki::MilitaryBase::health_bar_  [private]
```

6.16.3.6 health_bar_offset_y

```
constexpr float yuki::MilitaryBase::health_bar_offset_y = -10.f  [static], [constexpr], [private]
```

6.16.3.7 level_

```
int yuki::MilitaryBase::level_ = 1  [private]
```

6.16.3.8 texture_

```
sf::Texture yuki::MilitaryBase::texture_  [private]
```

The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/MilitaryBase.hpp
- E:/Project/Yuki/src/MilitaryBase.cpp

6.17 yuki::MouseEventProcessEngine Class Reference

```
#include <InputProcessEngine.hpp>
```

Public Member Functions

- MouseEventProcessEngine ()
- virtual ~MouseEventProcessEngine ()
- void registerTouchableObject (Touchable *object)
- void unregisterTouchableObject (Touchable *object)
- void processEvent (sf::Event event)

Private Attributes

- std::vector< Touchable * > touchable_objects_

6.17.1 Constructor & Destructor Documentation

6.17.1.1 MouseEventProcessEngine()

```
MouseEventProcessEngine::MouseEventProcessEngine ( )
```

6.17.1.2 ~MouseEventProcessEngine()

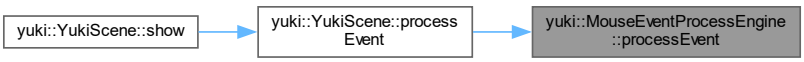
```
MouseEventProcessEngine::~MouseEventProcessEngine ( )  [virtual]
```

6.17.2 Member Function Documentation

6.17.2.1 processEvent()

```
void MouseEventProcessEngine::processEvent (
            sf::Event event )
```

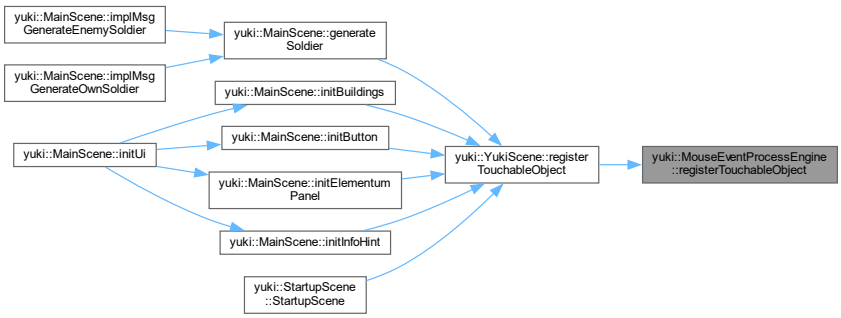
Here is the caller graph for this function:



6.17.2.2 registerTouchableObject()

```
void MouseEventProcessEngine::registerTouchableObject (
            Touchable * object )
```

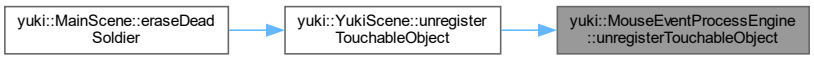
Here is the caller graph for this function:



6.17.2.3 unregisterTouchableObject()

```
void MouseEventProcessEngine::unregisterTouchableObject (
            Touchable * object )
```

Here is the caller graph for this function:



6.17.3 Member Data Documentation

6.17.3.1 touchable_objects_

```
std::vector<Touchable*> yuki::MouseEventProcessEngine::touchable_objects_  [private]
```

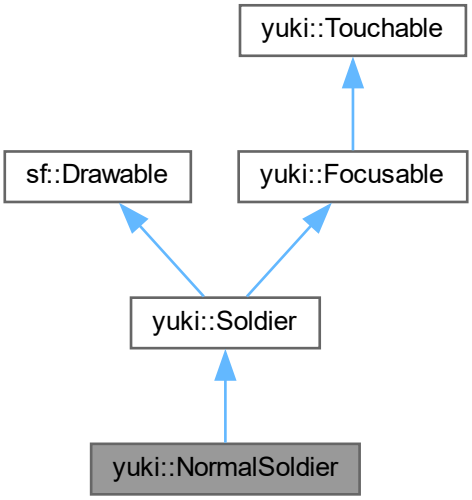
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/InputProcessEngine.hpp
- E:/Project/Yuki/src/InputProcessEngine.cpp

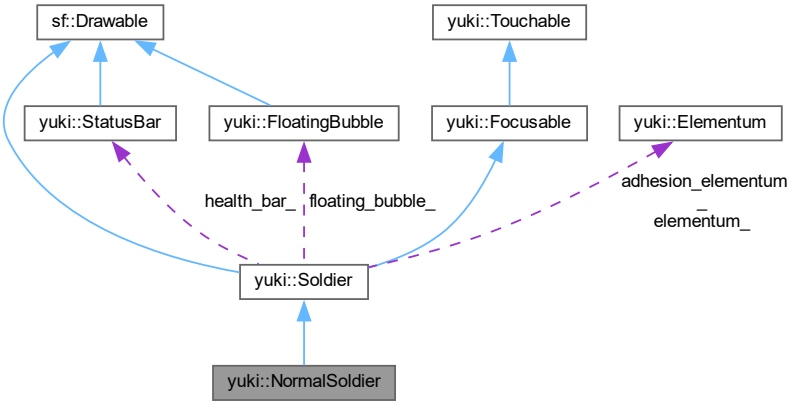
6.18 yuki::NormalSoldier Class Reference

```
#include <Soldier.hpp>
```

Inheritance diagram for yuki::NormalSoldier:



Collaboration diagram for yuki::NormalSoldier:



Public Member Functions

- NormalSoldier (Camp camp=Camp::Own)
- NormalSoldier (const sf::Vector2f &position, Camp camp=Camp::Own)

- virtual ~NormalSoldier ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update () override
- std::unique_ptr< BasicAttack > getDefaultAttack () override
- std::unique_ptr< GranuleAttack > getDefaultGranuleAttack ()
- void setGranuleAttackCD (const sf::Time &cd)
- const auto getGranuleAttackCD () const
- bool canGranuleAttack () const
- void resetGranuleAttackClock ()

Public Member Functions inherited from yuki::Soldier

- Soldier (Camp camp=Camp::Own)
- Soldier (const sf::Vector2f &position, Camp camp=Camp::Own)
- virtual ~Soldier ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update ()
- void setDirection (const Direction direction)
- constexpr auto getDirection () const
- void setSpeed (const float speed)
- constexpr auto getSpeed () const
- void setMoving (const bool moving)
- constexpr auto isMoving () const
- void setPosition (const sf::Vector2f &position)
- const auto getPosition () const
- void setOrigin (const sf::Vector2f &origin)
- const auto getOrigin () const
- auto getSize () const
- void setMaxHealth (const float max_health)
- constexpr auto getMaxHealth () const
- void setHealth (const float current_health)
- const auto getHealth () const
- void increaseHealth (const float health)
- void decreaseHealth (const float health)
- void setColor (const sf::Color &color)
- auto getColor () const
- void setPlaceOffset (const sf::Vector2f &offset)
- virtual const sf::Vector2f & getPlaceOffset () const
- void setFloatingBubbleVisible (const bool visible)
- constexpr auto isFloatingBubbleVisible () const
- void walk ()
- auto & getFloatingBubble ()
- const auto & getFloatingBubble () const
- void setElementum (const Elementum &elementum)
- const auto & getElementum () const
- void setElementumType (const ElementumType &elementum_type)
- constexpr auto getElementumType () const
- constexpr auto getCamp () const
- void setAdhesionElementum (const Elementum &elementum)
- const auto & getAdhesionElementum () const
- void getAttacked (const BasicAttack &attack)
- constexpr auto isFreezed () const
- virtual std::unique_ptr< BasicAttack > getDefaultAttack ()=0

Public Member Functions inherited from yuki::Focusable

- Focusable ()=default
- virtual ~Focusable ()
- virtual void onFocus (sf::Event e)
- virtual void onUnfocus (sf::Event e)
- constexpr auto isFocused () const
- void bindFocus (Callback callback)
- void bindUnfocus (Callback callback)
- void updateStatus (sf::Event event)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Attributes

- sf::Texture all_textures_
- sf::Vector2f place_offset_ = {0, 0}
- int current_frame_
- int previous_frame_
- int max_frame_
- sf::Time granule_attack_cd_ = sf::seconds(0.5f)
- sf::Clock granule_attack_clock_

Additional Inherited Members

Public Types inherited from yuki::Focusable

- using Callback = std::function< void(sf::Event e)>

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Soldier

- sf::Sprite sprite_
- sf::Vector2f health_bar_offset_ = {0, -15}
- Camp camp_
- sf::Vector2f place_offset_ = {0, 0}
- bool is_moving_ = false
- Direction direction_ = Direction::Down
- float speed_ = 0.0f
- Elementum elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- Elementum adhesion_elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- StatusBar health_bar_
- sf::Clock freeze_clock_
- bool is_freeze_ = false
- FloatingBubble floating_bubble_
- bool floating_bubble_visible_
- sf::Clock clock_

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.18.1 Constructor & Destructor Documentation

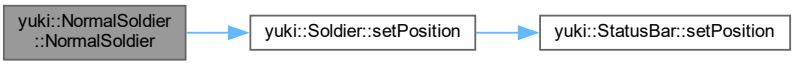
6.18.1.1 NormalSoldier() [1/2]

```
NormalSoldier::NormalSoldier (
            Camp camp = Camp::Own )
```

6.18.1.2 NormalSoldier() [2/2]

```
NormalSoldier::NormalSoldier (
            const sf::Vector2f & position,
            Camp camp = Camp::Own )
```

Here is the call graph for this function:



6.18.1.3 ~NormalSoldier()

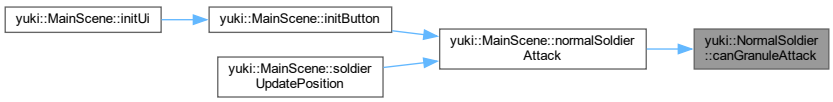
```
virtual yuki::NormalSoldier::~NormalSoldier ( )  [inline], [virtual]
```

6.18.2 Member Function Documentation

6.18.2.1 canGranuleAttack()

```
bool yuki::NormalSoldier::canGranuleAttack ( ) const  [inline]
```

Here is the caller graph for this function:

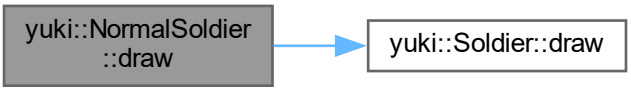


6.18.2.2 draw()

```
void NormalSoldier::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

Reimplemented from yuki::Soldier.

Here is the call graph for this function:

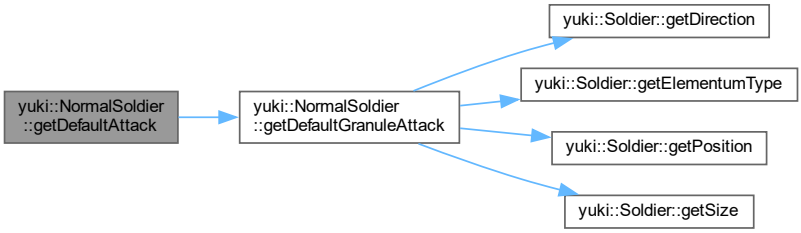


6.18.2.3 getDefaultAttack()

```
std::unique_ptr< BasicAttack > yuki::NormalSoldier::getDefaultAttack ( ) [inline], [override], [virtual]
```

Implements yuki::Soldier.

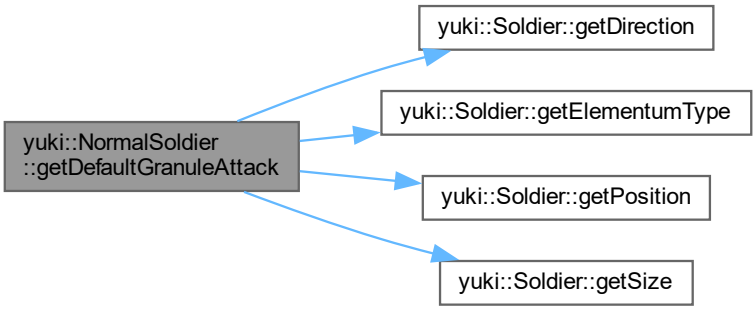
Here is the call graph for this function:



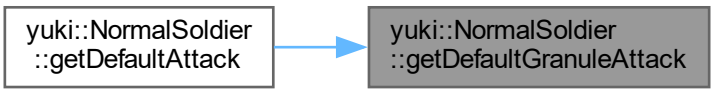
6.18.2.4 getDefaultGranuleAttack()

```
std::unique_ptr< GranuleAttack > NormalSoldier::getDefaultGranuleAttack ( )
```

Here is the call graph for this function:



Here is the caller graph for this function:



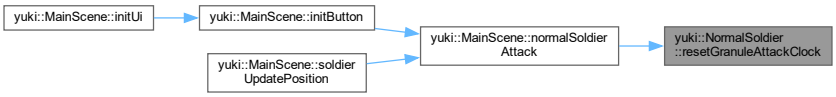
6.18.2.5 getGranuleAttackCD()

```
const auto yuki::NormalSoldier::getGranuleAttackCD ( ) const  [inline]
```

6.18.2.6 resetGranuleAttackClock()

```
void yuki::NormalSoldier::resetGranuleAttackClock ( )  [inline]
```

Here is the caller graph for this function:



6.18.2.7 setGranuleAttackCD()

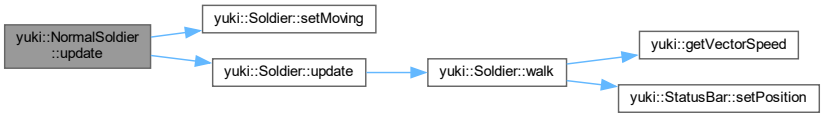
```
void yuki::NormalSoldier::setGranuleAttackCD (
            const sf::Time & cd )  [inline]
```

6.18.2.8 update()

```
void NormalSoldier::update ( )  [override], [virtual]
```

Reimplemented from yuki::Soldier.

Here is the call graph for this function:



6.18.3 Member Data Documentation

6.18.3.1 all_textures_

```
sf::Texture yuki::NormalSoldier::all_textures_ [private]
```

6.18.3.2 current_frame_

```
int yuki::NormalSoldier::current_frame_ [private]
```

6.18.3.3 granule_attack_cd_

```
sf::Time yuki::NormalSoldier::granule_attack_cd_ = sf::seconds(0.5f) [private]
```

6.18.3.4 granule_attack_clock_

```
sf::Clock yuki::NormalSoldier::granule_attack_clock_ [private]
```

6.18.3.5 max_frame_

```
int yuki::NormalSoldier::max_frame_ [private]
```

6.18.3.6 place_offset_

```
sf::Vector2f yuki::NormalSoldier::place_offset_ = {0, 0} [private]
```

6.18.3.7 previous_frame_

```
int yuki::NormalSoldier::previous_frame_ [private]
```

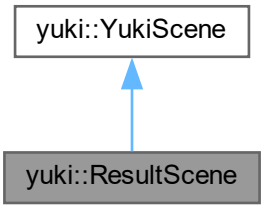
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Soldier.hpp
- E:/Project/Yuki/src/Soldier.cpp

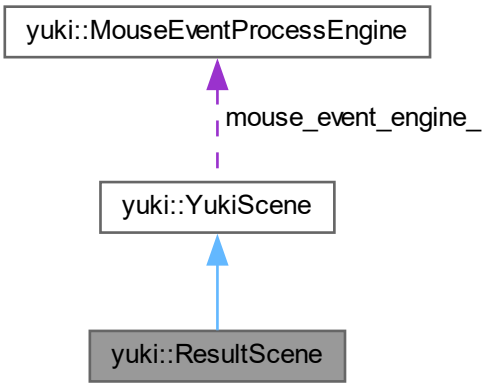
6.19 yuki::ResultScene Class Reference

```
#include <ResultScene.hpp>
```

Inheritance diagram for yuki::ResultScene:



Collaboration diagram for yuki::ResultScene:



Public Member Functions

- ResultScene (sf::RenderWindow &window, bool own_win=true)
- virtual ~ResultScene ()

Public Member Functions inherited from yuki::YukiScene

- YukiScene (sf::RenderWindow &window)
- virtual ~YukiScene ()
- virtual int show ()

Private Member Functions

- void initUi ()

Private Attributes

- sf::Text result_text_
- bool own_win_
- sf::Sprite background_
- sf::Texture background_texture_

Additional Inherited Members

Public Types inherited from yuki::YukiScene

- enum class Notify { None , End }

Protected Member Functions inherited from yuki::YukiScene

- void registerTouchableObject (Touchable *object)
- void unregisterTouchableObject (Touchable *object)
- void setReturnCode (int ret_code)
- virtual void processEvent (sf::Event event)
- virtual void draw () const =0
- virtual void updateInfo ()=0

Protected Attributes inherited from yuki::YukiScene

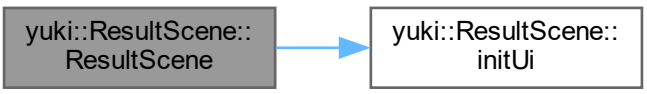
- yuki::MouseEventProcessEngine mouse_event_engine_
- Notify notify_
- int ret_code_
- sf::RenderWindow & window_

6.19.1 Constructor & Destructor Documentation

6.19.1.1 ResultScene()

```
ResultScene::ResultScene (
            sf::RenderWindow & window,
            bool own_win = true )
```

Here is the call graph for this function:



6.19.1.2 ~ResultScene()

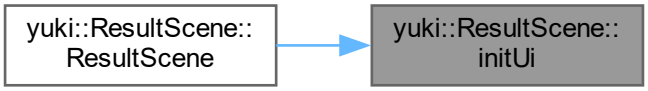
```
ResultScene::~ResultScene ( )  [virtual]
```

6.19.2 Member Function Documentation

6.19.2.1 initUi()

```
void ResultScene::initUi ( )  [private]
```

Here is the caller graph for this function:



6.19.3 Member Data Documentation

6.19.3.1 background_

```
sf::Sprite yuki::ResultScene::background_  [private]
```

6.19.3.2 background_texture_

```
sf::Texture yuki::ResultScene::background_texture_  [private]
```

6.19.3.3 own_win_

```
bool yuki::ResultScene::own_win_  [private]
```

6.19.3.4 result_text_

```
sf::Text yuki::ResultScene::result_text_ [private]
```

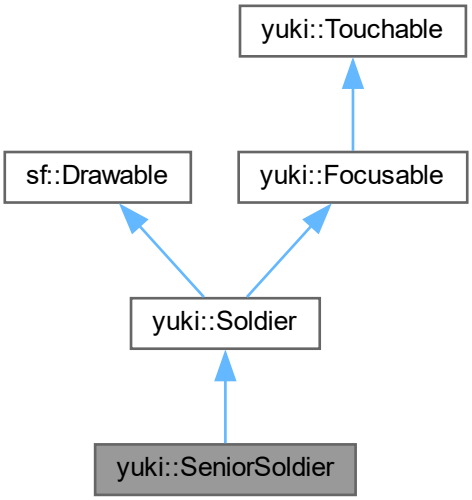
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/ResultScene.hpp
- E:/Project/Yuki/src/ResultScene.cpp

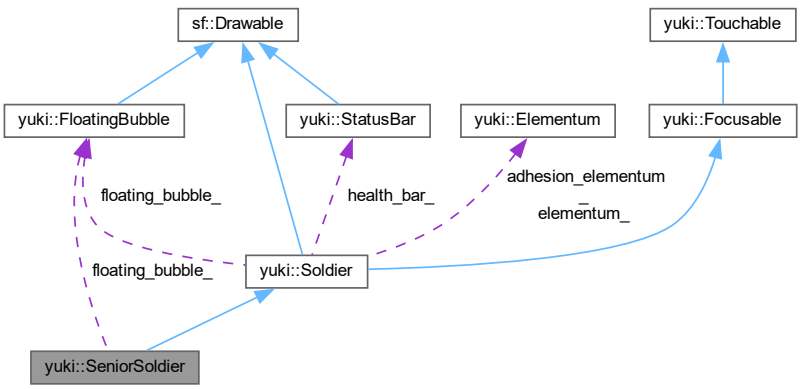
6.20 yuki::SeniorSoldier Class Reference

```
#include <Soldier.hpp>
```

Inheritance diagram for yuki::SeniorSoldier:



Collaboration diagram for yuki::SeniorSoldier:



Public Member Functions

- SeniorSoldier (Camp camp=Camp::Own)
- SeniorSoldier (const sf::Vector2f &position, Camp camp=Camp::Own)
- virtual ~SeniorSoldier ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update () override

Public Member Functions inherited from yuki::Soldier

- Soldier (Camp camp=Camp::Own)
- Soldier (const sf::Vector2f &position, Camp camp=Camp::Own)
- virtual ~Soldier ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update ()
- void setDirection (const Direction direction)
- constexpr auto getDirection () const
- void setSpeed (const float speed)
- constexpr auto getSpeed () const
- void setMoving (const bool moving)
- constexpr auto isMoving () const
- void setPosition (const sf::Vector2f &position)
- const auto getPosition () const
- void setOrigin (const sf::Vector2f &origin)
- const auto getOrigin () const
- auto getSize () const
- void setMaxHealth (const float max_health)
- constexpr auto getMaxHealth () const
- void setHealth (const float current_health)
- const auto getHealth () const
- void increaseHealth (const float health)
- void decreaseHealth (const float health)
- void setColor (const sf::Color &color)
- auto getColor () const
- void setPlaceOffset (const sf::Vector2f &offset)
- virtual const sf::Vector2f & getPlaceOffset () const
- void setFloatingBubbleVisible (const bool visible)
- constexpr auto isFloatingBubbleVisible () const
- void walk ()
- auto & getFloatingBubble ()
- const auto & getFloatingBubble () const
- void setElementum (const Elementum &elementum)
- const auto & getElementum () const
- void setElementumType (const ElementumType &elementum_type)
- constexpr auto getElementumType () const
- constexpr auto getCamp () const
- void setAdhesionElementum (const Elementum &elementum)
- const auto & getAdhesionElementum () const
- void getAttacked (const BasicAttack &attack)
- constexpr auto isFreezed () const
- virtual std::unique_ptr< BasicAttack > getDefaultAttack ()=0

Public Member Functions inherited from yuki::Focusable

- Focusable ()=default
- virtual ~Focusable ()
- virtual void onFocus (sf::Event e)
- virtual void onUnfocus (sf::Event e)
- constexpr auto isFocused () const
- void bindFocus (Callback callback)
- void bindUnfocus (Callback callback)
- void updateStatus (sf::Event event)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Attributes

- sf::Texture all_textures_
- sf::Vector2f place_offset_ = {0, 0}
- int current_frame_
- int previous_frame_
- int max_frame_
- FloatingBubble floating_bubble_

Additional Inherited Members

Public Types inherited from yuki::Focusable

- using Callback = std::function< void(sf::Event e)>

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Soldier

- sf::Sprite sprite_
- sf::Vector2f health_bar_offset_ = {0, -15}
- Camp camp_
- sf::Vector2f place_offset_ = {0, 0}
- bool is_moving_ = false
- Direction direction_ = Direction::Down
- float speed_ = 0.0f
- Elementum elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- Elementum adhesion_elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- StatusBar health_bar_
- sf::Clock freeze_clock_
- bool is_freeze_ = false
- FloatingBubble floating_bubble_
- bool floating_bubble_visible_
- sf::Clock clock_

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.20.1 Constructor & Destructor Documentation

6.20.1.1 SeniorSoldier() [1/2]

```
yuki::SeniorSoldier::SeniorSoldier (
            Camp camp = Camp::Own )
```

6.20.1.2 SeniorSoldier() [2/2]

```
yuki::SeniorSoldier::SeniorSoldier (
            const sf::Vector2f & position,
            Camp camp = Camp::Own )
```

6.20.1.3 ~SeniorSoldier()

```
virtual yuki::SeniorSoldier::~SeniorSoldier ( )  [inline], [virtual]
```

6.20.2 Member Function Documentation

6.20.2.1 draw()

```
virtual void yuki::SeniorSoldier::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

Reimplemented from yuki::Soldier.

6.20.2.2 update()

```
virtual void yuki::SeniorSoldier::update ( )  [override], [virtual]
```

Reimplemented from yuki::Soldier.

6.20.3 Member Data Documentation

6.20.3.1 all_textures_

```
sf::Texture yuki::SeniorSoldier::all_textures_  [private]
```

6.20.3.2 current_frame_

```
int yuki::SeniorSoldier::current_frame_  [private]
```

6.20.3.3 floating_bubble_

```
FloatingBubble yuki::SeniorSoldier::floating_bubble_  [private]
```

6.20.3.4 max_frame_

```
int yuki::SeniorSoldier::max_frame_  [private]
```

6.20.3.5 place_offset_

```
sf::Vector2f yuki::SeniorSoldier::place_offset_ = {0, 0}  [private]
```

6.20.3.6 previous_frame_

```
int yuki::SeniorSoldier::previous_frame_  [private]
```

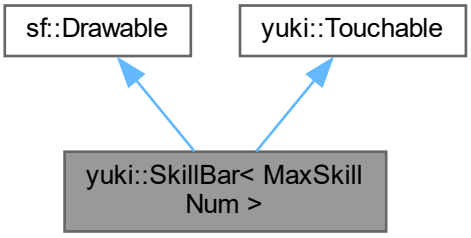
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/Soldier.hpp

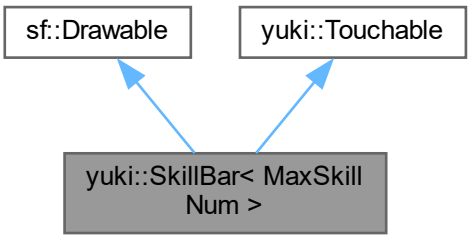
6.21 yuki::SkillBar< MaxSkillNum > Class Template Reference

```
#include <Skill.hpp>
```

Inheritance diagram for yuki::SkillBar< MaxSkillNum >:



Collaboration diagram for yuki::SkillBar< MaxSkillNum >:



Public Member Functions

- SkillBar (const sf::Vector2f &position={0, 0}, float radius=10.f, float border_thick=2.f, int skill_num=0)
- constexpr auto getSkillNum () const

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Protected Member Functions

- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override

Private Member Functions

- void updatePosition ()

Private Attributes

- unsigned skill_num_
- std::array< SkillIcon, MaxSkillNum > skills_
- std::array< sf::RectangleShape, MaxSkillNum > boxes_
- sf::Vector2f position_
- float icon_radius_ = 10.f
- float border_thick_ = 2.f

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

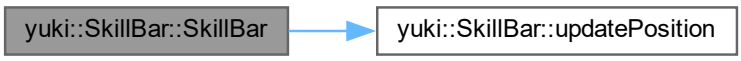
- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.21.1 Constructor & Destructor Documentation

6.21.1.1 SkillBar()

```
template<std::size_t N>
yuki::SkillBar< N >::SkillBar (
            const sf::Vector2f & position = {0, 0},
            float radius = 10.f,
            float border_thick = 2.f,
            int skill_num = 0 )
```

Here is the call graph for this function:



6.21.2 Member Function Documentation

6.21.2.1 draw()

```
template<std::size_t N>
void yuki::SkillBar< N >::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [protected], [virtual]
```

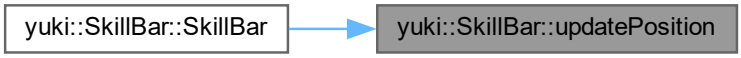
6.21.2.2 getSkillNum()

```
template<std::size_t MaxSkillNum = 3>
constexpr auto yuki::SkillBar< MaxSkillNum >::getSkillNum ( ) const  [inline], [constexpr]
```

6.21.2.3 updatePosition()

```
template<std::size_t N>
void yuki::SkillBar< N >::updatePosition  [private]
```

Here is the caller graph for this function:



6.21.3 Member Data Documentation

6.21.3.1 border_thick_

```
template<std::size_t MaxSkillNum = 3>
float yuki::SkillBar< MaxSkillNum >::border_thick_ = 2.f  [private]
```

6.21.3.2 boxes_

```
template<std::size_t MaxSkillNum = 3>
std::array<sf::RectangleShape, MaxSkillNum> yuki::SkillBar< MaxSkillNum >::boxes_  [private]
```

6.21.3.3 icon_radius_

```
template<std::size_t MaxSkillNum = 3>
float yuki::SkillBar< MaxSkillNum >::icon_radius_ = 10.f  [private]
```

6.21.3.4 position_

```
template<std::size_t MaxSkillNum = 3>
sf::Vector2f yuki::SkillBar< MaxSkillNum >::position_ [private]
```

6.21.3.5 skill_num_

```
template<std::size_t MaxSkillNum = 3>
unsigned yuki::SkillBar< MaxSkillNum >::skill_num_ [private]
```

6.21.3.6 skills_

```
template<std::size_t MaxSkillNum = 3>
std::array<SkillIcon, MaxSkillNum> yuki::SkillBar< MaxSkillNum >::skills_ [private]
```

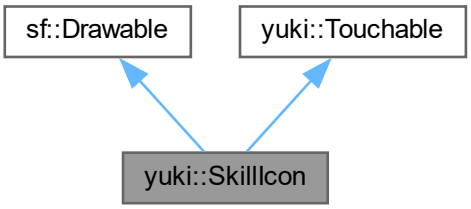
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/Skill.hpp

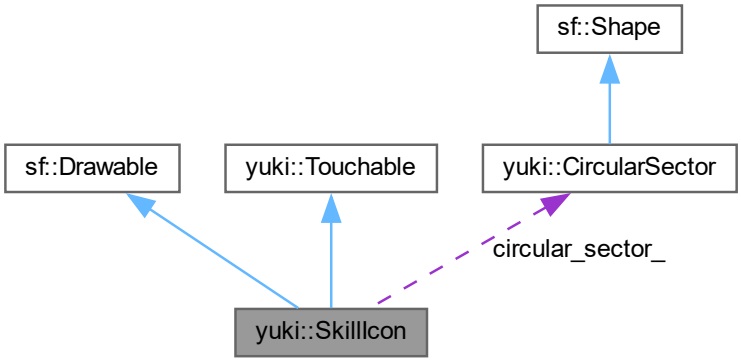
6.22 yuki::SkillIcon Class Reference

```
#include <Skill.hpp>
```

Inheritance diagram for yuki::SkillIcon:



Collaboration diagram for yuki::SkillIcon:



Public Member Functions

- SkillIcon (const sf::Vector2f &position={0, 0}, float radius=10.f, float border_thick=2.f)
- void setMaxValue (int max_value)
- constexpr auto getMaxValue () const
- void setCurrentValue (int current_value)
- constexpr auto getCurrentValue () const
- void setPosition (const sf::Vector2f &position)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Protected Member Functions

- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override

Private Member Functions

- void updateCircularSector ()

Private Attributes

- CircularSector circular_sector_
- sf::CircleShape background_
- sf::CircleShape icon_
- int max_value_
- int current_value_

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.22.1 Constructor & Destructor Documentation

6.22.1.1 SkillIcon()

```
SkillIcon::SkillIcon (
            const sf::Vector2f & position = {0, 0},
            float radius = 10.f,
            float border_thick = 2.f )
```

6.22.2 Member Function Documentation

6.22.2.1 draw()

```
void SkillIcon::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [protected], [virtual]
```

6.22.2.2 getCurrentValue()

```
constexpr auto yuki::SkillIcon::getCurrentValue ( ) const  [inline], [constexpr]
```

6.22.2.3 getMaxValue()

```
constexpr auto yuki::SkillIcon::getMaxValue ( ) const  [inline], [constexpr]
```

6.22.2.4 setCurrentValue()

```
void yuki::SkillIcon::setCurrentValue (
            int current_value )  [inline]
```

6.22.2.5 setMaxValue()

```
void yuki::SkillIcon::setMaxValue (
            int max_value )  [inline]
```

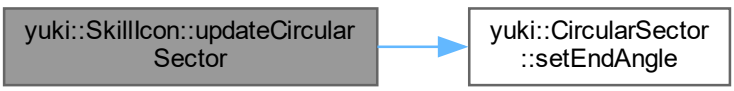
6.22.2.6 setPosition()

```
void SkillIcon::setPosition (
            const sf::Vector2f & position )
```

6.22.2.7 updateCircularSector()

```
void SkillIcon::updateCircularSector ( )  [private]
```

Here is the call graph for this function:



6.22.3 Member Data Documentation

6.22.3.1 background_

```
sf::CircleShape yuki::SkillIcon::background_ [private]
```

6.22.3.2 circular_sector_

```
CircularSector yuki::SkillIcon::circular_sector_ [private]
```

6.22.3.3 current_value_

```
int yuki::SkillIcon::current_value_ [private]
```

6.22.3.4 icon_

```
sf::CircleShape yuki::SkillIcon::icon_ [private]
```

6.22.3.5 max_value_

```
int yuki::SkillIcon::max_value_ [private]
```

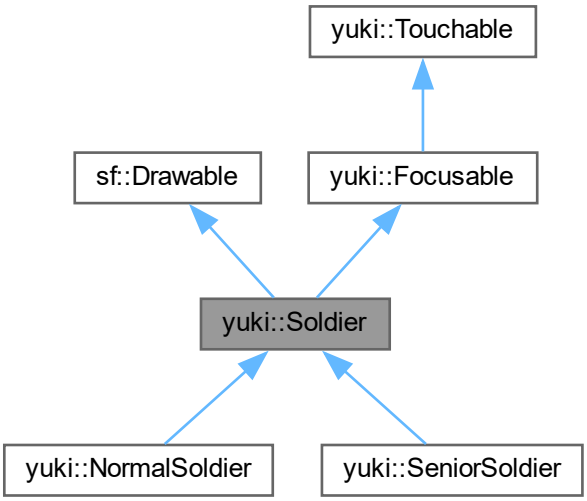
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Skill.hpp
- E:/Project/Yuki/src/Skill.cpp

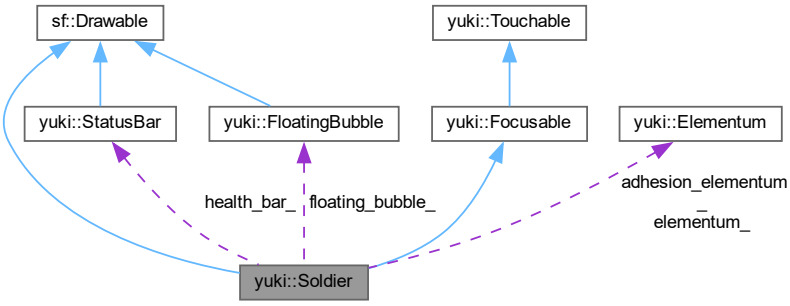
6.23 yuki::Soldier Class Reference

```
#include <Soldier.hpp>
```

Inheritance diagram for yuki::Soldier:



Collaboration diagram for yuki::Soldier:



Public Member Functions

- Soldier (Camp camp=Camp::Own)
- Soldier (const sf::Vector2f &position, Camp camp=Camp::Own)
- virtual ~Soldier ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- virtual void update ()

- void setDirection (const Direction direction)
- constexpr auto getDirection () const
- void setSpeed (const float speed)
- constexpr auto getSpeed () const
- void setMoving (const bool moving)
- constexpr auto isMoving () const
- void setPosition (const sf::Vector2f &position)
- const auto getPosition () const
- void setOrigin (const sf::Vector2f &origin)
- const auto getOrigin () const
- auto getSize () const
- void setMaxHealth (const float max_health)
- constexpr auto getMaxHealth () const
- void setHealth (const float current_health)
- const auto getHealth () const
- void increaseHealth (const float health)
- void decreaseHealth (const float health)
- void setColor (const sf::Color &color)
- auto getColor () const
- void setPlaceOffset (const sf::Vector2f &offset)
- virtual const sf::Vector2f & getPlaceOffset () const
- void setFloatingBubbleVisible (const bool visible)
- constexpr auto isFloatingBubbleVisible () const
- void walk ()
- auto & getFloatingBubble ()
- const auto & getFloatingBubble () const
- void setElementum (const Elementum &elementum)
- const auto & getElementum () const
- void setElementumType (const ElementumType &elementum_type)
- constexpr auto getElementumType () const
- constexpr auto getCamp () const
- void setAdhesionElementum (const Elementum &elementum)
- const auto & getAdhesionElementum () const
- void getAttacked (const BasicAttack &attack)
- constexpr auto isFreezed () const
- virtual std::unique_ptr< BasicAttack > getDefaultAttack ()=0

Public Member Functions inherited from yuki::Focusable

- Focusable ()=default
- virtual ~Focusable ()
- virtual void onFocus (sf::Event e)
- virtual void onUnfocus (sf::Event e)
- constexpr auto isFocused () const
- void bindFocus (Callback callback)
- void bindUnfocus (Callback callback)
- void updateStatus (sf::Event event)

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Protected Attributes

- sf::Sprite sprite_
- sf::Vector2f health_bar_offset_ = {0, -15}
- Camp camp_
- sf::Vector2f place_offset_ = {0, 0}
- bool is_moving_ = false
- Direction direction_ = Direction::Down
- float speed_ = 0.0f
- Elementum elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- Elementum adhesion_elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
- StatusBar health_bar_
- sf::Clock freeze_clock_
- bool is_freeze_ = false
- FloatingBubble floating_bubble_
- bool floating_bubble_visible_
- sf::Clock clock_

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

Additional Inherited Members

Public Types inherited from yuki::Focusable

- using Callback = std::function< void(sf::Event e)>

Public Types inherited from yuki::Touchable

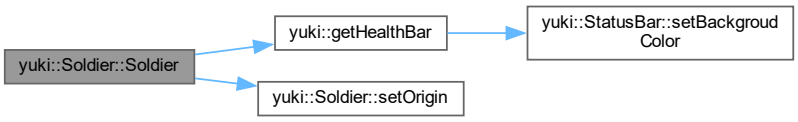
- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

6.23.1 Constructor & Destructor Documentation

6.23.1.1 Soldier() [1/2]

```
yuki::Soldier::Soldier (
            Camp camp = Camp::Own )  [inline]
```

Here is the call graph for this function:



6.23.1.2 Soldier() [2/2]

```
yuki::Soldier::Soldier (
            const sf::Vector2f & position,
            Camp camp = Camp::Own )  [inline]
```

Here is the call graph for this function:

6.23.1.3 ~Soldier()

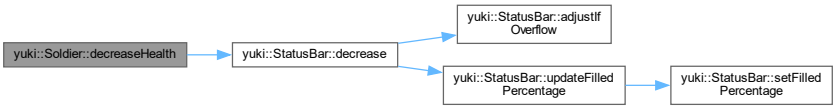
```
virtual yuki::Soldier::~Soldier ( )  [inline], [virtual]
```

6.23.2 Member Function Documentation

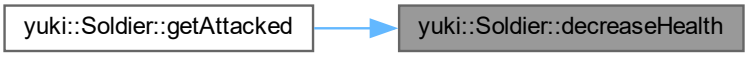
6.23.2.1 decreaseHealth()

```
void yuki::Soldier::decreaseHealth (
            const float health )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:

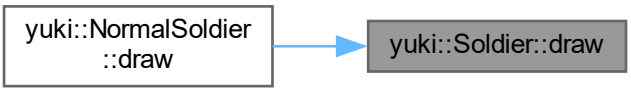


6.23.2.2 draw()

```
void Soldier::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

Reimplemented in yuki::NormalSoldier, and yuki::SeniorSoldier.

Here is the caller graph for this function:



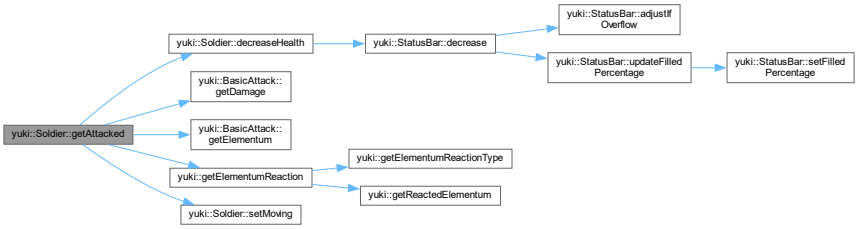
6.23.2.3 getAdhesionElementum()

```
const auto & yuki::Soldier::getAdhesionElementum ( ) const  [inline]
```

6.23.2.4 getAttacked()

```
void Soldier::getAttacked (
            const BasicAttack & attack )
```

Here is the call graph for this function:



6.23.2.5 getCamp()

```
constexpr auto yuki::Soldier::getCamp ( ) const  [inline], [constexpr]
```

6.23.2.6 getColor()

```
auto yuki::Soldier::getColor ( ) const  [inline]
```

6.23.2.7 getDefaultAttack()

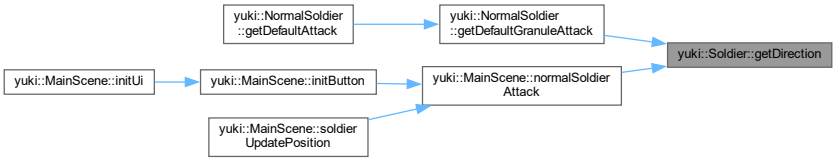
```
virtual std::unique_ptr< BasicAttack > yuki::Soldier::getDefaultAttack ( )  [pure virtual]
```

Implemented in yuki::NormalSoldier.

6.23.2.8 getDirection()

```
constexpr auto yuki::Soldier::getDirection ( ) const  [inline], [constexpr]
```

Here is the caller graph for this function:



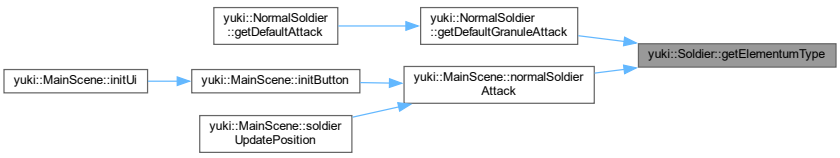
6.23.2.9 getElementum()

```
const auto & yuki::Soldier::getElementum ( ) const  [inline]
```

6.23.2.10 getElementumType()

```
constexpr auto yuki::Soldier::getElementumType ( ) const  [inline], [constexpr]
```

Here is the caller graph for this function:



6.23.2.11 getFloatingBubble() [1/2]

```
auto & yuki::Soldier::getFloatingBubble ( )  [inline]
```

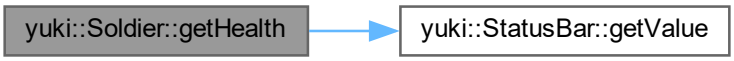
6.23.2.12 getFloatingBubble() [2/2]

```
const auto & yuki::Soldier::getFloatingBubble ( ) const  [inline]
```

6.23.2.13 getHealth()

```
const auto yuki::Soldier::getHealth ( ) const  [inline]
```

Here is the call graph for this function:



6.23.2.14 getMaxHealth()

```
constexpr auto yuki::Soldier::getMaxHealth ( ) const  [inline], [constexpr]
```

Here is the call graph for this function:

6.23.2.15 getOrigin()

```
const auto yuki::Soldier::getOrigin ( ) const  [inline]
```

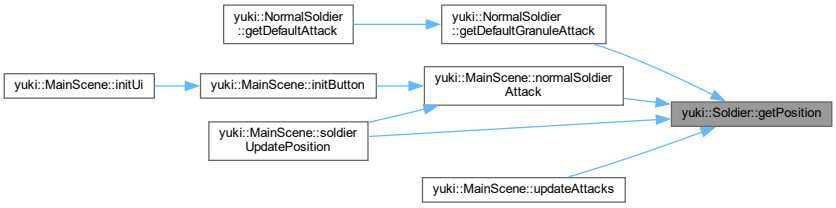
6.23.2.16 getPlaceOffset()

```
virtual const sf::Vector2f & yuki::Soldier::getPlaceOffset ( ) const  [inline], [virtual]
```

6.23.2.17 getPosition()

```
const auto yuki::Soldier::getPosition ( ) const  [inline]
```

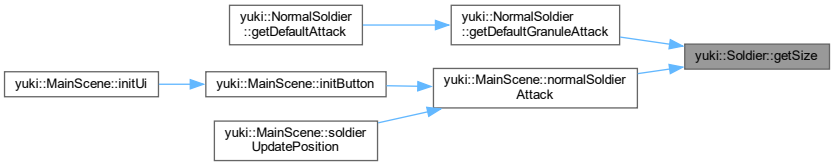
Here is the caller graph for this function:



6.23.2.18 getSize()

```
auto yuki::Soldier::getSize ( ) const  [inline]
```

Here is the caller graph for this function:



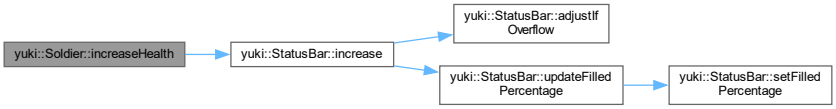
6.23.2.19 getSpeed()

```
constexpr auto yuki::Soldier::getSpeed ( ) const  [inline], [constexpr]
```

6.23.2.20 increaseHealth()

```
void yuki::Soldier::increaseHealth (
            const float health )  [inline]
```

Here is the call graph for this function:



6.23.2.21 isFloatingBubbleVisible()

```
constexpr auto yuki::Soldier::isFloatingBubbleVisible ( ) const  [inline], [constexpr]
```

6.23.2.22 isFreezed()

```
constexpr auto yuki::Soldier::isFreezed ( ) const  [inline], [constexpr]
```

6.23.2.23 isMoving()

```
constexpr auto yuki::Soldier::isMoving ( ) const  [inline], [constexpr]
```

6.23.2.24 setAdhesionElementum()

```
void yuki::Soldier::setAdhesionElementum (
            const Elementum & elementum )  [inline]
```

6.23.2.25 setColor()

```
void yuki::Soldier::setColor (
            const sf::Color & color )  [inline]
```

6.23.2.26 setDirection()

```
void yuki::Soldier::setDirection (
            const Direction direction )  [inline]
```

6.23.2.27 setElementum()

```
void yuki::Soldier::setElementum (
            const Elementum & elementum )  [inline]
```

6.23.2.28 setElementumType()

```
void yuki::Soldier::setElementumType (
            const ElementumType & elementum_type )  [inline]
```

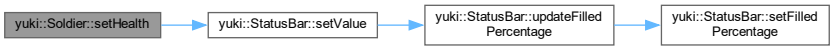
6.23.2.29 setFloatingBubbleVisible()

```
void yuki::Soldier::setFloatingBubbleVisible (
            const bool visible )  [inline]
```

6.23.2.30 setHealth()

```
void yuki::Soldier::setHealth (
            const float current_health )  [inline]
```

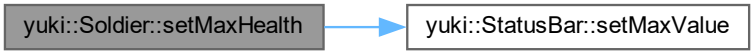
Here is the call graph for this function:



6.23.2.31 setMaxHealth()

```
void yuki::Soldier::setMaxHealth (
            const float max_health )  [inline]
```

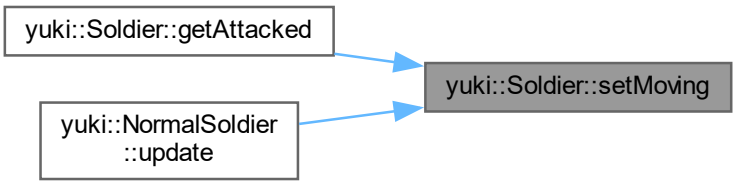
Here is the call graph for this function:



6.23.2.32 setMoving()

```
void yuki::Soldier::setMoving (
            const bool moving )  [inline]
```

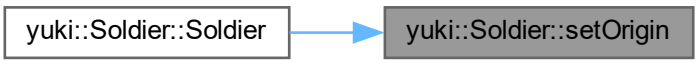
Here is the caller graph for this function:



6.23.2.33 setOrigin()

```
void yuki::Soldier::setOrigin (
            const sf::Vector2f & origin )  [inline]
```

Here is the caller graph for this function:



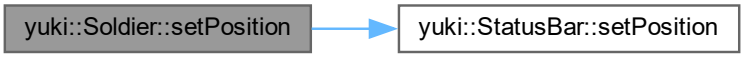
6.23.2.34 setPlaceOffset()

```
void yuki::Soldier::setPlaceOffset (
            const sf::Vector2f & offset )  [inline]
```

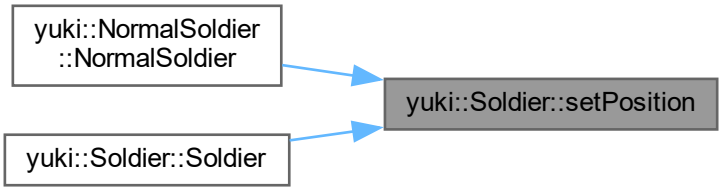
6.23.2.35 setPosition()

```
void yuki::Soldier::setPosition (
            const sf::Vector2f & position )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.23.2.36 setSpeed()

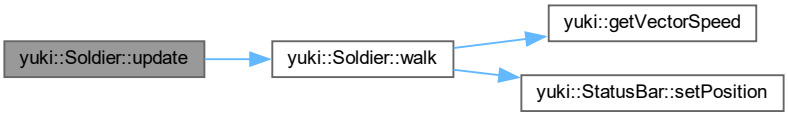
```
void yuki::Soldier::setSpeed (
            const float speed )  [inline]
```

6.23.2.37 update()

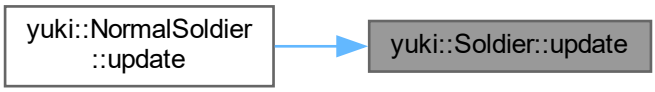
```
void Soldier::update ( ) [virtual]
```

Reimplemented in yuki::NormalSoldier, and yuki::SeniorSoldier.

Here is the call graph for this function:



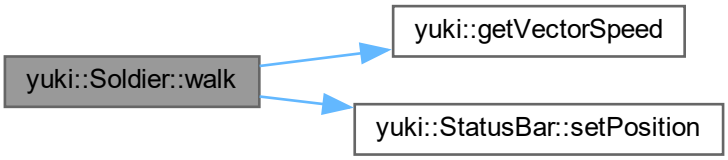
Here is the caller graph for this function:



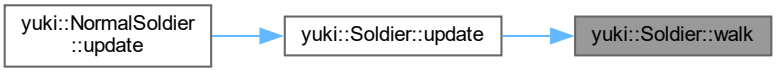
6.23.2.38 walk()

```
void Soldier::walk ( )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.23.3 Member Data Documentation

6.23.3.1 adhesion_elementum_

```
Elementum yuki::Soldier::adhesion_elementum_ = Elementum(yuki::ElementumType::None, 0.0f)
[protected]
```

6.23.3.2 camp_

```
Camp yuki::Soldier::camp_  [protected]
```

6.23.3.3 clock_

```
sf::Clock yuki::Soldier::clock_  [protected]
```

6.23.3.4 direction_

```
Direction yuki::Soldier::direction_ = Direction::Down  [protected]
```

6.23.3.5 elementum_

```
Elementum yuki::Soldier::elementum_ = Elementum(yuki::ElementumType::None, 0.0f)  [protected]
```

6.23.3.6 floating_bubble_

```
FloatingBubble yuki::Soldier::floating_bubble_ [protected]
```

6.23.3.7 floating_bubble_visible_

```
bool yuki::Soldier::floating_bubble_visible_ [protected]
```

6.23.3.8 freeze_clock_

```
sf::Clock yuki::Soldier::freeze_clock_ [protected]
```

6.23.3.9 health_bar_

```
StatusBar yuki::Soldier::health_bar_ [protected]
```

6.23.3.10 health_bar_offset_

```
sf::Vector2f yuki::Soldier::health_bar_offset_ = {0, -15} [protected]
```

6.23.3.11 is_freeze_

```
bool yuki::Soldier::is_freeze_ = false [protected]
```

6.23.3.12 is_moving_

```
bool yuki::Soldier::is_moving_ = false [protected]
```

6.23.3.13 place_offset_

```
sf::Vector2f yuki::Soldier::place_offset_ = {0, 0} [protected]
```

6.23.3.14 speed_

```
float yuki::Soldier::speed_ = 0.0f  [protected]
```

6.23.3.15 sprite_

```
sf::Sprite yuki::Soldier::sprite_  [protected]
```

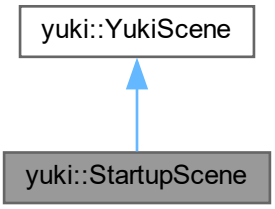
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/Soldier.hpp
- E:/Project/Yuki/src/Soldier.cpp

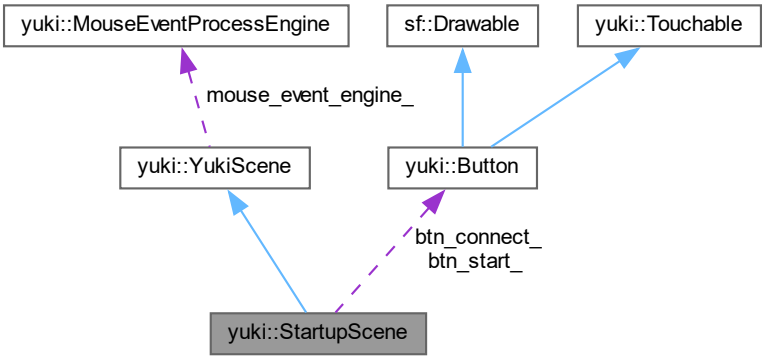
6.24 yuki::StartupScene Class Reference

```
#include <StartupScene.hpp>
```

Inheritance diagram for yuki::StartupScene:



Collaboration diagram for yuki::StartupScene:



Public Member Functions

- StartupScene (sf::RenderWindow &window)
- virtual ~StartupScene ()

Public Member Functions inherited from yuki::YukiScene

- YukiScene (sf::RenderWindow &window)
- virtual ~YukiScene ()
- virtual int show ()

Private Member Functions

- void initUi ()

Private Attributes

- Button btn_start_
- Button btn_connect_
- sf::Sprite background_
- sf::Texture background_texture_

Additional Inherited Members

Public Types inherited from yuki::YukiScene

- enum class Notify { None , End }

Protected Member Functions inherited from yuki::YukiScene

- void registerTouchableObject (Touchable *object)
- void unregisterTouchableObject (Touchable *object)
- void setReturnCode (int ret_code)
- virtual void processEvent (sf::Event event)
- virtual void draw () const =0
- virtual void updateInfo ()=0

Protected Attributes inherited from yuki::YukiScene

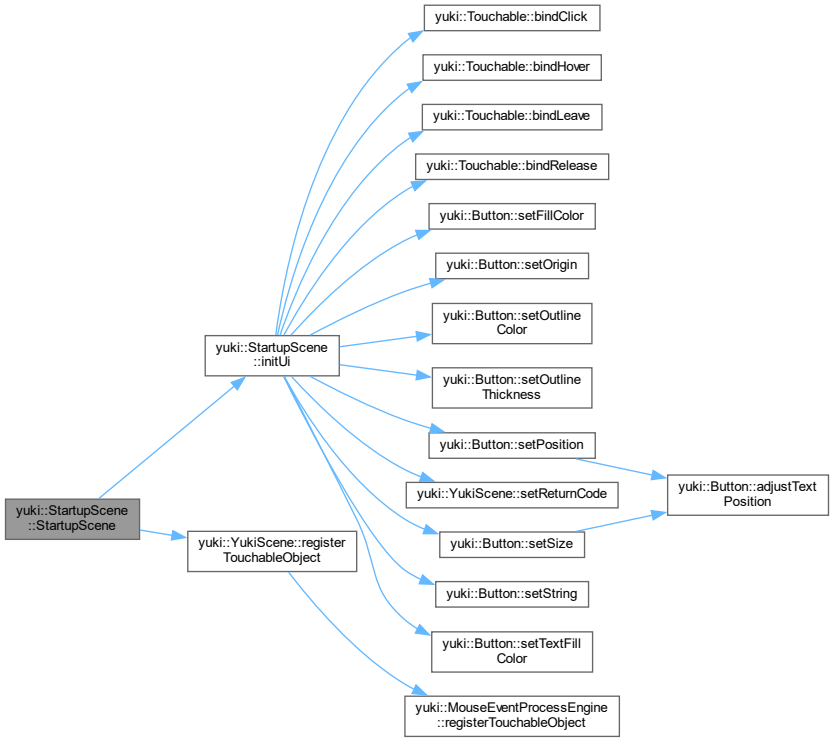
- yuki::MouseEventProcessEngine mouse_event_engine_
- Notify notify_
- int ret_code_
- sf::RenderWindow & window_

6.24.1 Constructor & Destructor Documentation

6.24.1.1 StartupScene()

```
StartupScene::StartupScene (
            sf::RenderWindow & window )
```

Here is the call graph for this function:



6.24.1.2 ~StartupScene()

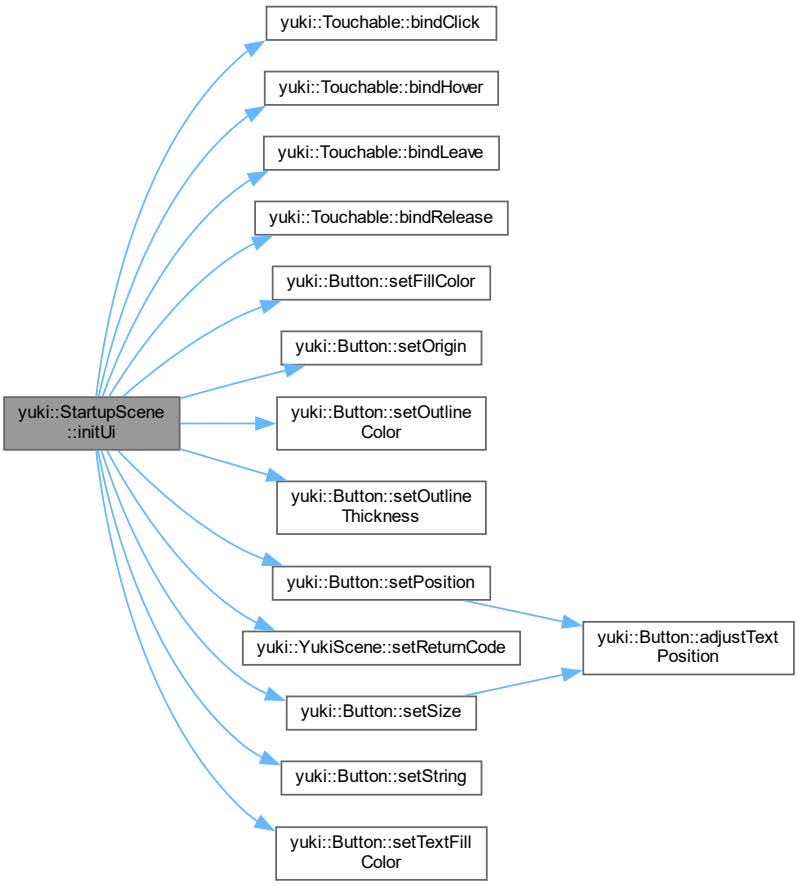
```
virtual yuki::StartupScene::~StartupScene ( )  [inline], [virtual]
```

6.24.2 Member Function Documentation

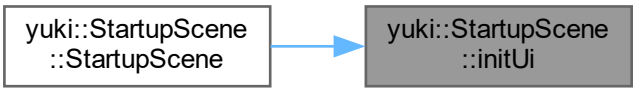
6.24.2.1 initUi()

```
void StartupScene::initUi ( ) [private]
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.24.3 Member Data Documentation

6.24.3.1 background_

```
sf::Sprite yuki::StartupScene::background_ [private]
```

6.24.3.2 background_texture_

```
sf::Texture yuki::StartupScene::background_texture_ [private]
```

6.24.3.3 btn_connect_

```
Button yuki::StartupScene::btn_connect_ [private]
```

6.24.3.4 btn_start_

```
Button yuki::StartupScene::btn_start_ [private]
```

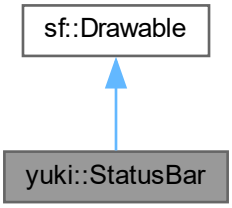
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/StartupScene.hpp
- E:/Project/Yuki/src/StartupScene.cpp

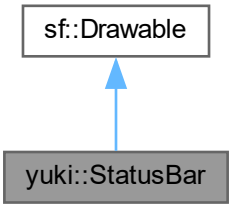
6.25 yuki::StatusBar Class Reference

```
#include <StatusBar.hpp>
```

Inheritance diagram for yuki::StatusBar:



Collaboration diagram for yuki::StatusBar:



Public Member Functions

- StatusBar ()=default
- StatusBar (sf::Vector2f size, sf::Vector2f position, sf::Color background_color=sf::Color::Black, sf::Color filled_color=sf::Color::Blue, float max_value=100.f, float current_value=100.f)
- void show ()
- void hide ()
- auto isVisible () const
- void setBackgroudColor (const sf::Color color)
- void setFilledColor (const sf::Color color)
- void setMaxValue (const float max_value)
- void setValue (const float current_value)
- void setPosition (const sf::Vector2f &position)
- void setSize (const sf::Vector2f &size)
- void setOutlineThickness (const float thickness)
- void setOutlineColor (const sf::Color &color)
- void setOutlineStyle (const sf::Color &color, const float thickness)
- const auto getBackgroundColor () const
- const auto getFilledColor () const
- constexpr auto getMaxValue () const
- constexpr auto getValue () const
- const auto getPosition () const
- void increase (const float value)
- void decrease (const float value)
- virtual ~StatusBar ()
- virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override
- void updateFilledPercentage ()

Protected Member Functions

- void setFilledPercentage (float percentage)
- void adjustIfOverflow ()

Protected Attributes

- sf::RectangleShape background_
- sf::RectangleShape filled_
- sf::Vector2f size_
- sf::Vector2f position_
- float max_value_
- float current_value_
- bool visible_

6.25.1 Constructor & Destructor Documentation

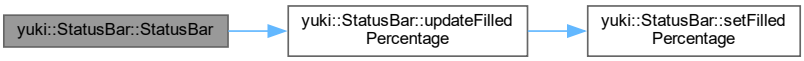
6.25.1.1 StatusBar() [1/2]

```
yuki::StatusBar::StatusBar ( ) [default]
```

6.25.1.2 StatusBar() [2/2]

```
yuki::StatusBar::StatusBar (
            sf::Vector2f size,
            sf::Vector2f position,
            sf::Color background_color = sf::Color::Black,
            sf::Color filled_color = sf::Color::Blue,
            float max_value = 100.f,
            float current_value = 100.f ) [inline]
```

Here is the call graph for this function:



6.25.1.3 ~StatusBar()

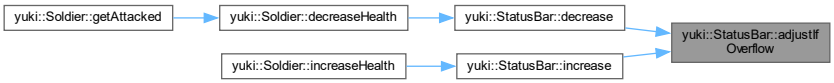
```
virtual yuki::StatusBar::~StatusBar ( ) [inline], [virtual]
```

6.25.2 Member Function Documentation

6.25.2.1 adjustIfOverflow()

```
void StatusBar::adjustIfOverflow ( )  [protected]
```

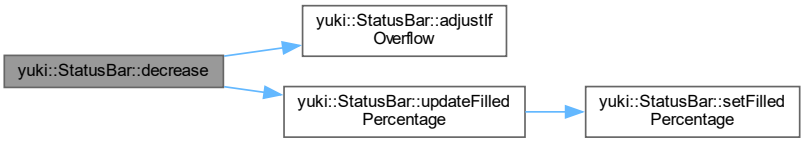
Here is the caller graph for this function:



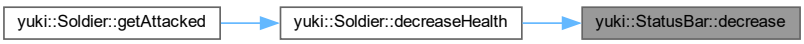
6.25.2.2 decrease()

```
void StatusBar::decrease (
            const float value )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.25.2.3 draw()

```
void StatusBar::draw (
            sf::RenderTarget & target,
            sf::RenderStates states ) const  [override], [virtual]
```

6.25.2.4 getBackgroundColor()

```
const auto yuki::StatusBar::getBackgroundColor ( ) const  [inline]
```

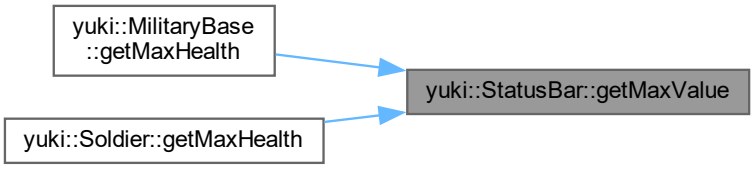
6.25.2.5 getFilledColor()

```
const auto yuki::StatusBar::getFilledColor ( ) const  [inline]
```

6.25.2.6 getMaxValue()

```
constexpr auto yuki::StatusBar::getMaxValue ( ) const  [inline], [constexpr]
```

Here is the caller graph for this function:



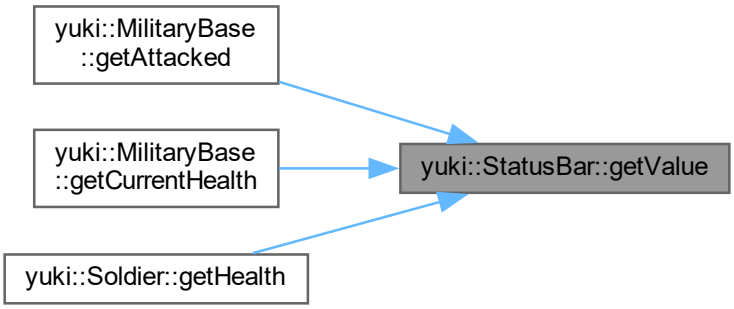
6.25.2.7 getPosition()

```
const auto yuki::StatusBar::getPosition ( ) const  [inline]
```

6.25.2.8 getValue()

```
constexpr auto yuki::StatusBar::getValue ( ) const  [inline], [constexpr]
```

Here is the caller graph for this function:



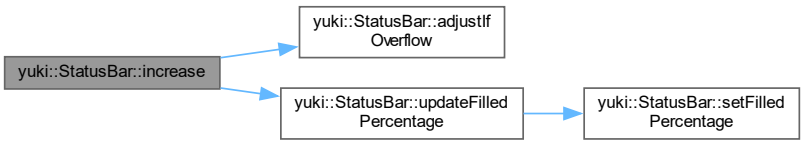
6.25.2.9 hide()

```
void yuki::StatusBar::hide ( )  [inline]
```

6.25.2.10 increase()

```
void StatusBar::increase (
            const float value )
```

Here is the call graph for this function:



Here is the caller graph for this function:

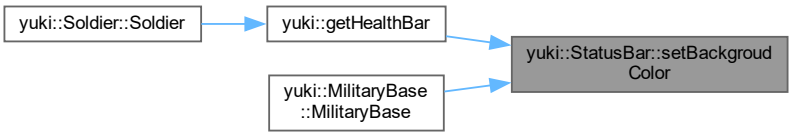
6.25.2.11 isVisible()

```
auto yuki::StatusBar::isVisible ( ) const  [inline]
```

6.25.2.12 setBackgroudColor()

```
void yuki::StatusBar::setBackgroudColor (
            const sf::Color color )  [inline]
```

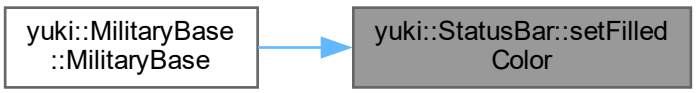
Here is the caller graph for this function:



6.25.2.13 setFilledColor()

```
void yuki::StatusBar::setFilledColor (
            const sf::Color color )  [inline]
```

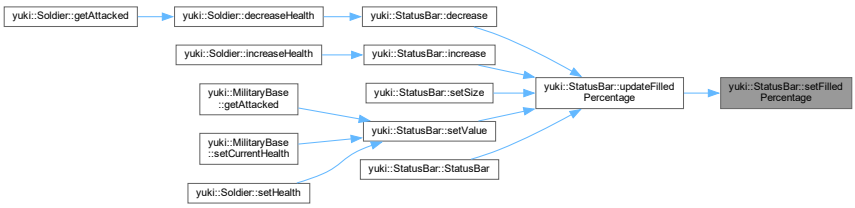
Here is the caller graph for this function:



6.25.2.14 setFilledPercentage()

```
void StatusBar::setFilledPercentage (
            float percentage )  [protected]
```

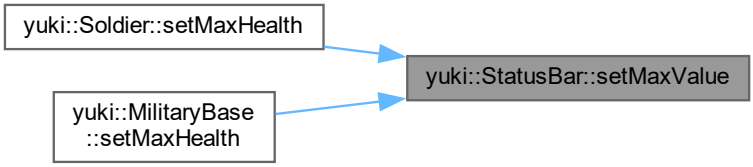
Here is the caller graph for this function:



6.25.2.15 setMaxValue()

```
void yuki::StatusBar::setMaxValue (
            const float max_value )  [inline]
```

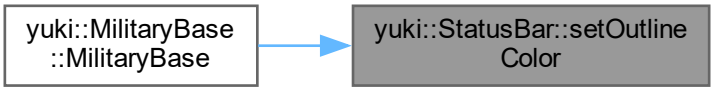
Here is the caller graph for this function:



6.25.2.16 setOutlineColor()

```
void yuki::StatusBar::setOutlineColor (
            const sf::Color & color )  [inline]
```

Here is the caller graph for this function:



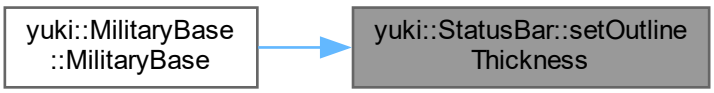
6.25.2.17 setOutlineStyle()

```
void yuki::StatusBar::setOutlineStyle (
            const sf::Color & color,
            const float thickness )  [inline]
```

6.25.2.18 setOutlineThickness()

```
void yuki::StatusBar::setOutlineThickness (
            const float thickness )  [inline]
```

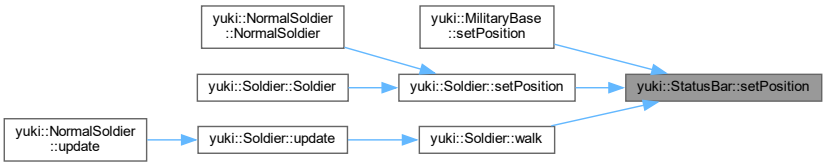
Here is the caller graph for this function:



6.25.2.19 setPosition()

```
void StatusBar::setPosition (
            const sf::Vector2f & position )
```

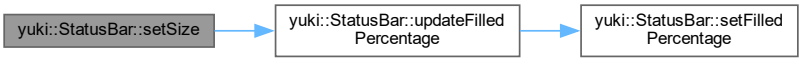
Here is the caller graph for this function:



6.25.2.20 setSize()

```
void yuki::StatusBar::setSize (
            const sf::Vector2f & size )  [inline]
```

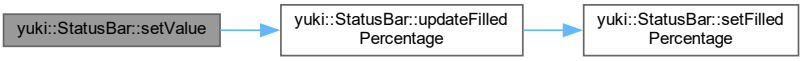
Here is the call graph for this function:



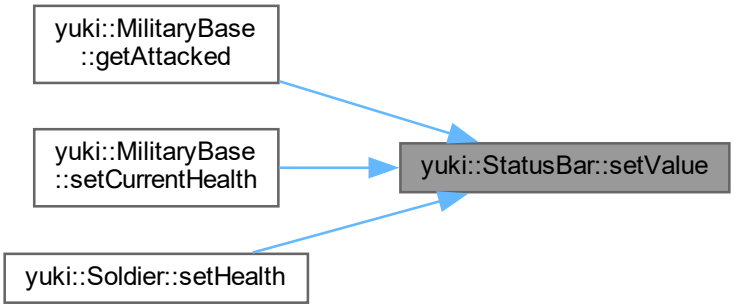
6.25.2.21 setValue()

```
void yuki::StatusBar::setValue (
            const float current_value )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



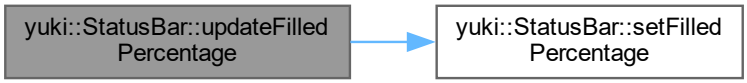
6.25.2.22 show()

```
void yuki::StatusBar::show ( ) [inline]
```

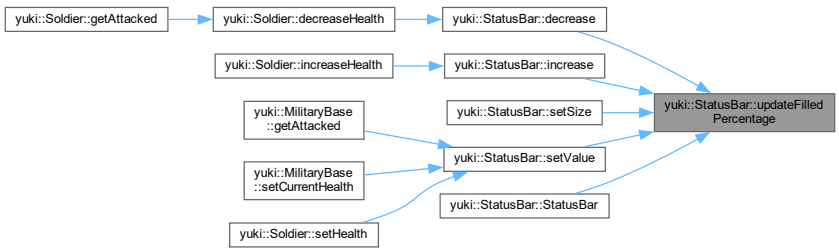
6.25.2.23 updateFilledPercentage()

```
void yuki::StatusBar::updateFilledPercentage ( ) [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.25.3 Member Data Documentation

6.25.3.1 background_

```
sf::RectangleShape yuki::StatusBar::background_ [protected]
```

6.25.3.2 current_value_

```
float yuki::StatusBar::current_value_ [protected]
```

6.25.3.3 filled_

```
sf::RectangleShape yuki::StatusBar::filled_ [protected]
```

6.25.3.4 max_value_

```
float yuki::StatusBar::max_value_ [protected]
```

6.25.3.5 position_

```
sf::Vector2f yuki::StatusBar::position_ [protected]
```

6.25.3.6 size_

```
sf::Vector2f yuki::StatusBar::size_ [protected]
```

6.25.3.7 visible_

```
bool yuki::StatusBar::visible_ [protected]
```

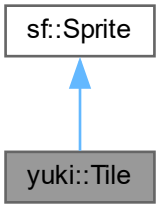
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/StatusBar.hpp
- E:/Project/Yuki/src/StatusBar.cpp

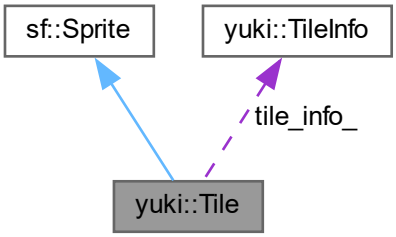
6.26 yuki::Tile Class Reference

```
#include <Tile.hpp>
```

Inheritance diagram for yuki::Tile:



Collaboration diagram for yuki::Tile:



Public Member Functions

- auto getTileInfo () const
- void setTileInfo (TileInfo &&tile_info)
- void setTileInfo (const TileInfo &tile_info)
- void setTextureSrc (const std::string &src_location)

Private Attributes

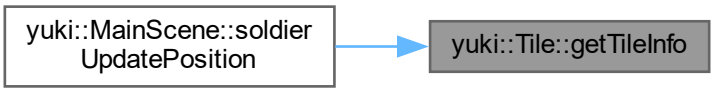
- TileInfo tile_info_
- sf::Texture texture_

6.26.1 Member Function Documentation

6.26.1.1 getTileInfo()

```
auto yuki::Tile::getTileInfo ( ) const  [inline]
```

Here is the caller graph for this function:



6.26.1.2 setTextureSrc()

```
void yuki::Tile::setTextureSrc (
            const std::string & src_location )  [inline]
```

6.26.1.3 setTileInfo() [1/2]

```
void yuki::Tile::setTileInfo (
            const TileInfo & tile_info )  [inline]
```

6.26.1.4 setTileInfo() [2/2]

```
void yuki::Tile::setTileInfo (
            TileInfo && tile_info )  [inline]
```

6.26.2 Member Data Documentation

6.26.2.1 texture_

```
sf::Texture yuki::Tile::texture_  [private]
```

6.26.2.2 tile_info_

```
TileInfo yuki::Tile::tile_info_ [private]
```

The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/Tile.hpp

6.27 yuki::TileInfo Struct Reference

```
#include <Tile.hpp>
```

Public Member Functions

- TileInfo ()=default
- TileInfo (TileCategory category, int index)

Public Attributes

- TileCategory category
- int index

6.27.1 Constructor & Destructor Documentation

6.27.1.1 TileInfo() [1/2]

```
yuki::TileInfo::TileInfo ( ) [default]
```

6.27.1.2 TileInfo() [2/2]

```
yuki::TileInfo::TileInfo (
            TileCategory category,
            int index ) [inline]
```

6.27.2 Member Data Documentation

6.27.2.1 category

```
TileCategory yuki::TileInfo::category
```

6.27.2.2 index

```
int yuki::TileInfo::index
```

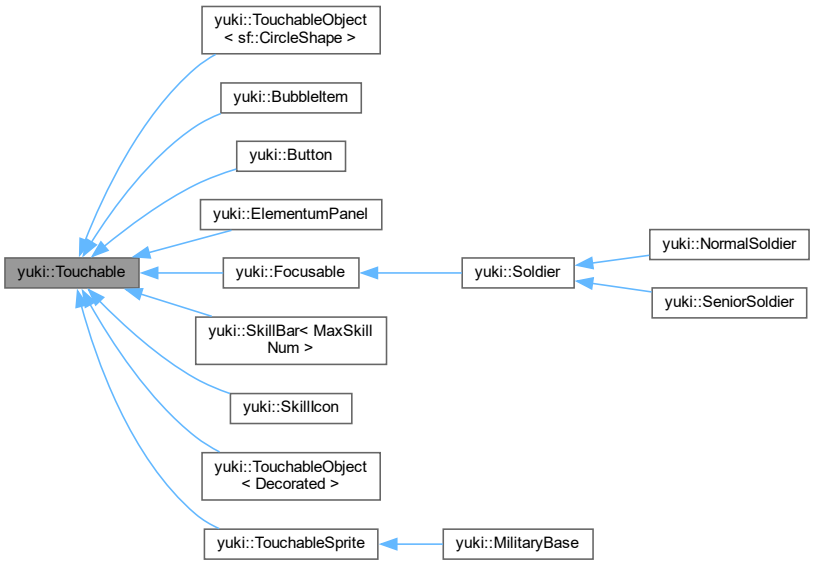
The documentation for this struct was generated from the following file:

- E:/Project/Yuki/include/Tile.hpp

6.28 yuki::Touchable Class Reference

```
#include <InteractiveObject.hpp>
```

Inheritance diagram for yuki::Touchable:



Public Types

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Public Member Functions

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Protected Attributes

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.28.1 Member Typedef Documentation

6.28.1.1 Callback

```
using yuki::Touchable::Callback = std::function<void(sf::Event e)>
```

6.28.2 Member Enumeration Documentation

6.28.2.1 Action

```
enum class yuki::Touchable::Action [strong]
```

Enumerator

Click	
Release	
Hover	
Leave	

6.28.3 Constructor & Destructor Documentation

6.28.3.1 Touchable()

```
yuki::Touchable::Touchable ( ) [default]
```

6.28.3.2 ~Touchable()

```
virtual yuki::Touchable::~Touchable ( ) [inline], [virtual]
```

6.28.4 Member Function Documentation

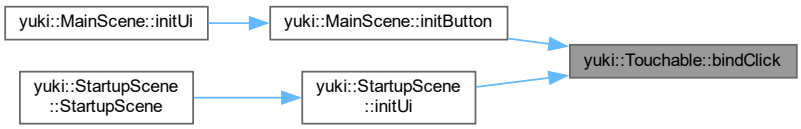
6.28.4.1 bindAction()

```
void Touchable::bindAction (
            const Action action,
            Callback callback )
```

6.28.4.2 bindClick()

```
void yuki::Touchable::bindClick (
            Callback callback ) [inline]
```

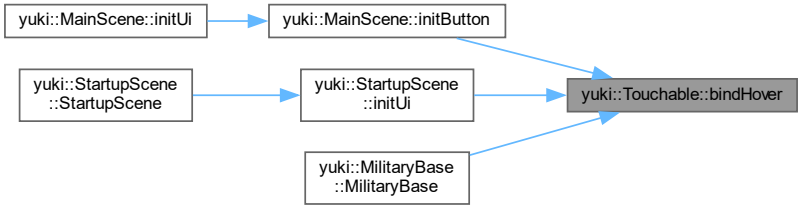
Here is the caller graph for this function:



6.28.4.3 bindHover()

```
void yuki::Touchable::bindHover (
            Callback callback )  [inline]
```

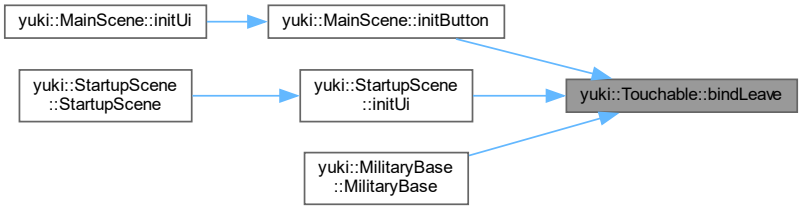
Here is the caller graph for this function:



6.28.4.4 bindLeave()

```
void yuki::Touchable::bindLeave (
            Callback callback )  [inline]
```

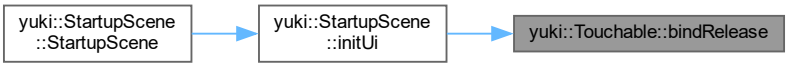
Here is the caller graph for this function:



6.28.4.5 bindRelease()

```
void yuki::Touchable::bindRelease (
            Callback callback )  [inline]
```

Here is the caller graph for this function:

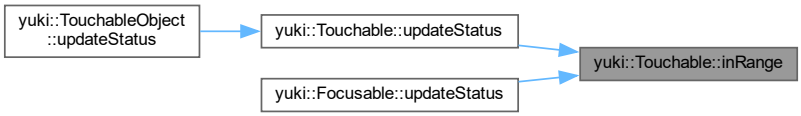


6.28.4.6 inRange()

```
virtual bool yuki::Touchable::inRange (
            const sf::Vector2f & position ) const  [pure virtual]
```

Implemented in yuki::TouchableObject< Decorated >, yuki::TouchableObject< sf::CircleShape >, and yuki::TouchableSprite.

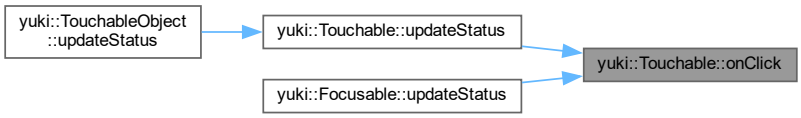
Here is the caller graph for this function:



6.28.4.7 onClick()

```
virtual void yuki::Touchable::onClick (
            sf::Event e )  [inline], [virtual]
```

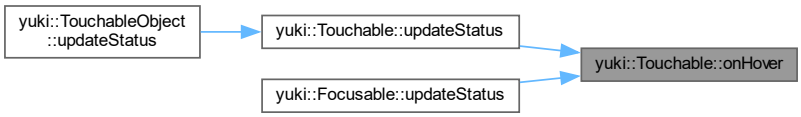
Here is the caller graph for this function:



6.28.4.8 onHover()

```
virtual void yuki::Touchable::onHover (
            sf::Event e )  [inline], [virtual]
```

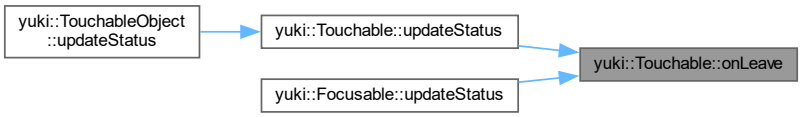
Here is the caller graph for this function:



6.28.4.9 onLeave()

```
virtual void yuki::Touchable::onLeave (
            sf::Event e )  [inline], [virtual]
```

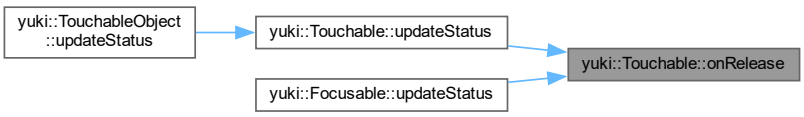
Here is the caller graph for this function:



6.28.4.10 onRelease()

```
virtual void yuki::Touchable::onRelease (
            sf::Event e )  [inline], [virtual]
```

Here is the caller graph for this function:

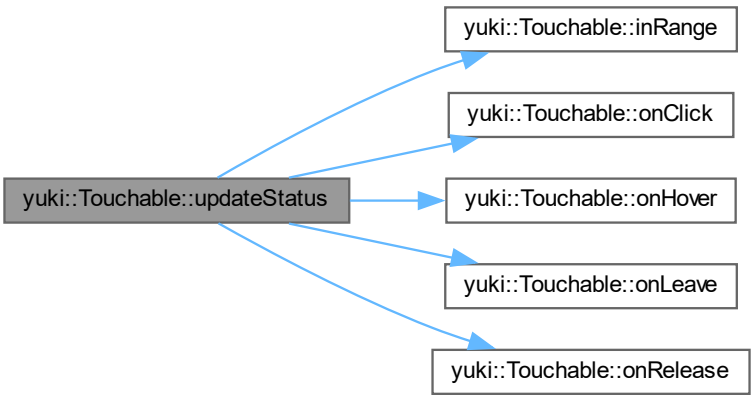


6.28.4.11 updateStatus()

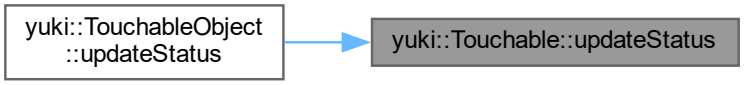
```
void Touchable::updateStatus (
            sf::Event event )  [virtual]
```

Reimplemented in yuki::Focusable, yuki::TouchableObject< Decorated >, and yuki::TouchableObject< sf::CircleShape >.

Here is the call graph for this function:



Here is the caller graph for this function:



6.28.5 Member Data Documentation

6.28.5.1 is_clicked_

```
bool yuki::Touchable::is_clicked_ = false  [protected]
```

6.28.5.2 is_hovered_

```
bool yuki::Touchable::is_hovered_ = false  [protected]
```

6.28.5.3 on_clickCallback_

```
Callback yuki::Touchable::on_clickCallback_ = [](sf::Event) {}  [protected]
```

6.28.5.4 on_hoverCallback_

```
Callback yuki::Touchable::on_hoverCallback_ = [](sf::Event) {}  [protected]
```

6.28.5.5 on_leaveCallback_

```
Callback yuki::Touchable::on_leaveCallback_ = [](sf::Event) {}  [protected]
```

6.28.5.6 on_releaseCallback_

```
Callback yuki::Touchable::on_releaseCallback_ = [](sf::Event) {}  [protected]
```

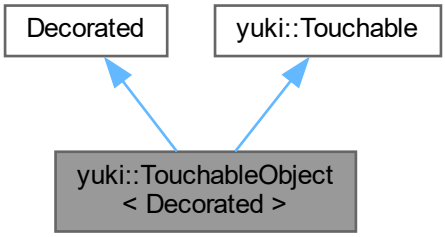
The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/InteractiveObject.hpp
- E:/Project/Yuki/src/InteractiveObject.cpp

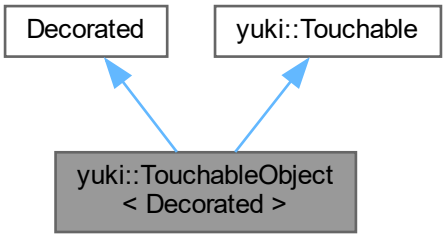
6.29 yuki::TouchableObject< Decorated > Class Template Reference

```
#include <InteractiveObject.hpp>
```

Inheritance diagram for yuki::TouchableObject< Decorated >:



Collaboration diagram for yuki::TouchableObject< Decorated >:



Public Member Functions

- TouchableObject ()=default
- virtual ~TouchableObject ()
- virtual bool inRange (const sf::Vector2f &position) const override
- virtual void updateStatus (sf::Event event) override

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)

- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)
- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.29.1 Constructor & Destructor Documentation

6.29.1.1 TouchableObject()

```
template<class Decorated >
yuki::TouchableObject< Decorated >::TouchableObject ( )  [default]
```

6.29.1.2 ~TouchableObject()

```
template<class Decorated >
virtual yuki::TouchableObject< Decorated >::~TouchableObject ( )  [inline], [virtual]
```

6.29.2 Member Function Documentation

6.29.2.1 inRange()

```
template<class Decorated >
virtual bool yuki::TouchableObject< Decorated >::inRange (
            const sf::Vector2f & position ) const  [inline], [override], [virtual]
```

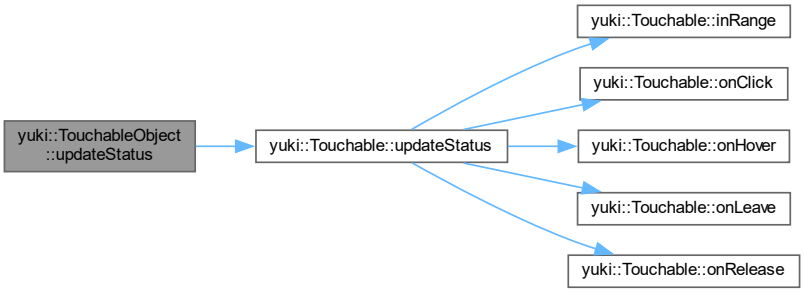
Implements yuki::Touchable.

6.29.2.2 updateStatus()

```
template<class Decorated >
virtual void yuki::TouchableObject< Decorated >::updateStatus (
            sf::Event event )  [inline], [override], [virtual]
```

Reimplemented from yuki::Touchable.

Here is the call graph for this function:



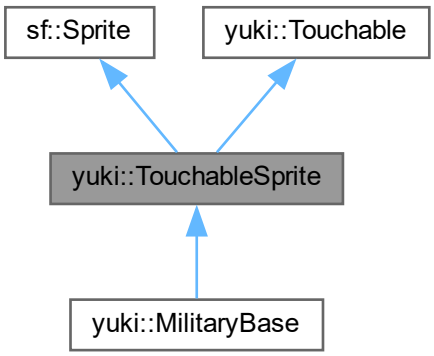
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/InteractiveObject.hpp

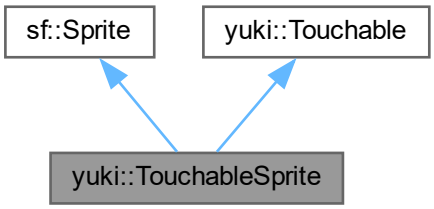
6.30 yuki::TouchableSprite Class Reference

```
#include <TouchableSprite.hpp>
```

Inheritance diagram for yuki::TouchableSprite:



Collaboration diagram for yuki::TouchableSprite:



Public Member Functions

- TouchableSprite ()=default
- TouchableSprite (const sf::Texture &texture)
- TouchableSprite (const sf::Texture &texture, const sf::IntRect &rect)
- virtual ~TouchableSprite ()

Public Member Functions inherited from yuki::Touchable

- Touchable ()=default
- virtual ~Touchable ()
- virtual void onRelease (sf::Event e)
- virtual void onClick (sf::Event e)
- virtual void onHover (sf::Event e)
- virtual void onLeave (sf::Event e)
- void bindAction (const Action action, Callback callback)

- void bindClick (Callback callback)
- void bindRelease (Callback callback)
- void bindHover (Callback callback)
- void bindLeave (Callback callback)
- virtual bool inRange (const sf::Vector2f &position) const =0
- virtual void updateStatus (sf::Event event)

Private Member Functions

- virtual bool inRange (const sf::Vector2f &position) const override

Additional Inherited Members

Public Types inherited from yuki::Touchable

- enum class Action { Click , Release , Hover , Leave }
- using Callback = std::function< void(sf::Event e)>

Protected Attributes inherited from yuki::Touchable

- bool is_clicked_ = false
- bool is_hovered_ = false
- Callback on_clickCallback_ = [ ](sf::Event) {}
- Callback on_releaseCallback_ = [ ](sf::Event) {}
- Callback on_hoverCallback_ = [ ](sf::Event) {}
- Callback on_leaveCallback_ = [ ](sf::Event) {}

6.30.1 Constructor & Destructor Documentation

6.30.1.1 TouchableSprite() [1/3]

```
yuki::TouchableSprite::TouchableSprite ( ) [default]
```

6.30.1.2 TouchableSprite() [2/3]

```
yuki::TouchableSprite::TouchableSprite (
            const sf::Texture & texture ) [inline]
```

6.30.1.3 TouchableSprite() [3/3]

```
yuki::TouchableSprite::TouchableSprite (
            const sf::Texture & texture,
            const sf::IntRect & rect )  [inline]
```

6.30.1.4 ~TouchableSprite()

```
virtual yuki::TouchableSprite::~TouchableSprite ( )  [inline], [virtual]
```

6.30.2 Member Function Documentation

6.30.2.1 inRange()

```
virtual bool yuki::TouchableSprite::inRange (
            const sf::Vector2f & position ) const  [inline], [override], [private], [virtual]
```

Implements yuki::Touchable.

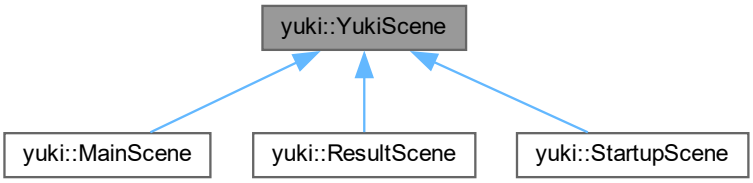
The documentation for this class was generated from the following file:

- E:/Project/Yuki/include/TouchableSprite.hpp

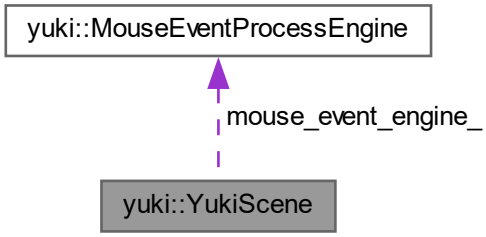
6.31 yuki::YukiScene Class Reference

```
#include <YukiScene.hpp>
```

Inheritance diagram for yuki::YukiScene:



Collaboration diagram for yuki::YukiScene:



Public Types

- enum class Notify { None , End }

Public Member Functions

- YukiScene (sf::RenderWindow &window)
- virtual ~YukiScene ()
- virtual int show ()

Protected Member Functions

- void registerTouchableObject (Touchable *object)
- void unregisterTouchableObject (Touchable *object)
- void setReturnCode (int ret_code)
- virtual void processEvent (sf::Event event)
- virtual void draw () const =0
- virtual void updateInfo ()=0

Protected Attributes

- yuki::MouseEventProcessEngine mouse_event_engine_
- Notify notify_
- int ret_code_
- sf::RenderWindow & window_

6.31.1 Member Enumeration Documentation

6.31.1.1 Notify

```
enum class yuki::YukiScene::Notify  [strong]
```

Enumerator

None	
End	

6.31.2 Constructor & Destructor Documentation

6.31.2.1 YukiScene()

```
yuki::YukiScene::YukiScene (
            sf::RenderWindow & window )  [inline]
```

6.31.2.2 ~YukiScene()

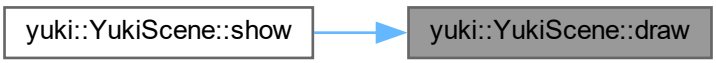
```
YukiScene::~YukiScene ( )  [virtual]
```

6.31.3 Member Function Documentation

6.31.3.1 draw()

```
virtual void yuki::YukiScene::draw ( ) const  [protected], [pure virtual]
```

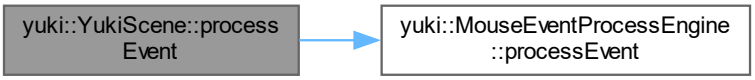
Here is the caller graph for this function:



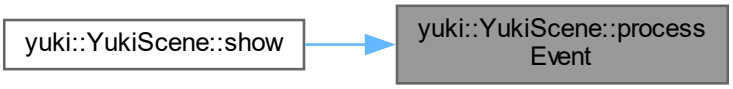
6.31.3.2 processEvent()

```
void YukiScene::processEvent (
            sf::Event event )  [protected], [virtual]
```

Here is the call graph for this function:



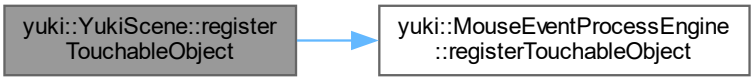
Here is the caller graph for this function:



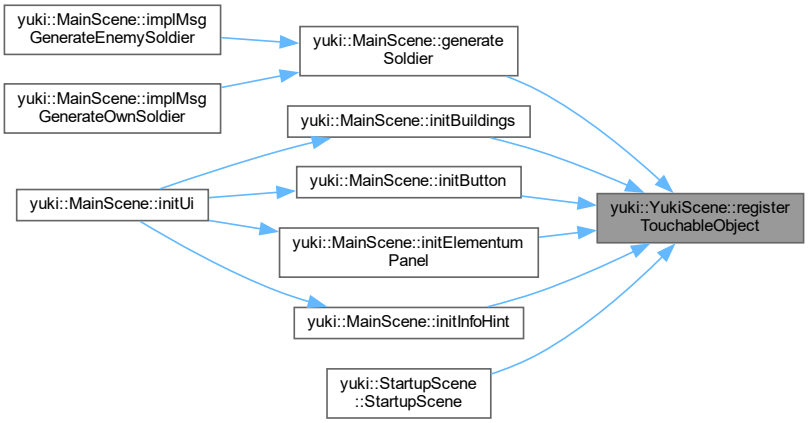
6.31.3.3 registerTouchableObject()

```
void yuki::YukiScene::registerTouchableObject (
            Touchable * object )  [inline], [protected]
```

Here is the call graph for this function:



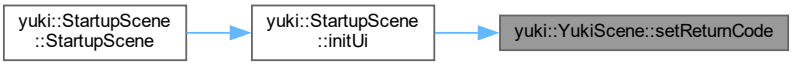
Here is the caller graph for this function:



6.31.3.4 setReturnCode()

```
void yuki::YukiScene::setReturnCode (
            int ret_code )  [inline], [protected]
```

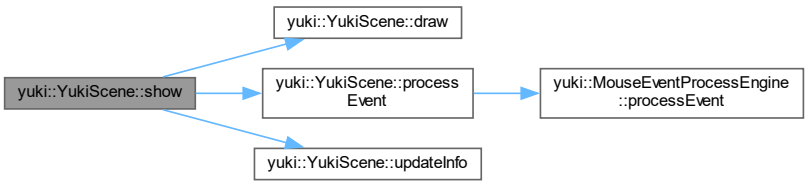
Here is the caller graph for this function:



6.31.3.5 show()

```
int YukiScene::show ( )  [virtual]
```

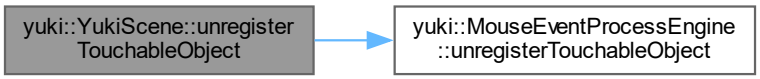
Here is the call graph for this function:



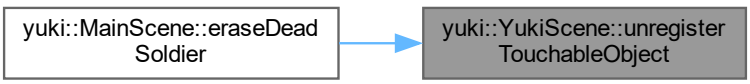
6.31.3.6 unregisterTouchableObject()

```
void yuki::YukiScene::unregisterTouchableObject (
            Touchable * object )  [inline], [protected]
```

Here is the call graph for this function:



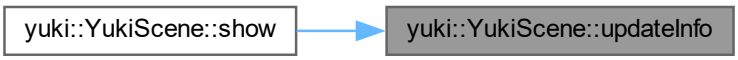
Here is the caller graph for this function:



6.31.3.7 updateInfo()

```
virtual void yuki::YukiScene::updateInfo ( )  [protected], [pure virtual]
```

Here is the caller graph for this function:



6.31.4 Member Data Documentation

6.31.4.1 mouse_event_engine_

```
yuki::MouseEventProcessEngine yuki::YukiScene::mouse_event_engine_ [protected]
```

6.31.4.2 notify_

```
Notify yuki::YukiScene::notify_ [protected]
```

6.31.4.3 ret_code_

```
int yuki::YukiScene::ret_code_ [protected]
```

6.31.4.4 window_

```
sf::RenderWindow& yuki::YukiScene::window_ [protected]
```

The documentation for this class was generated from the following files:

- E:/Project/Yuki/include/YukiScene.hpp
- E:/Project/Yuki/src/YukiScene.cpp

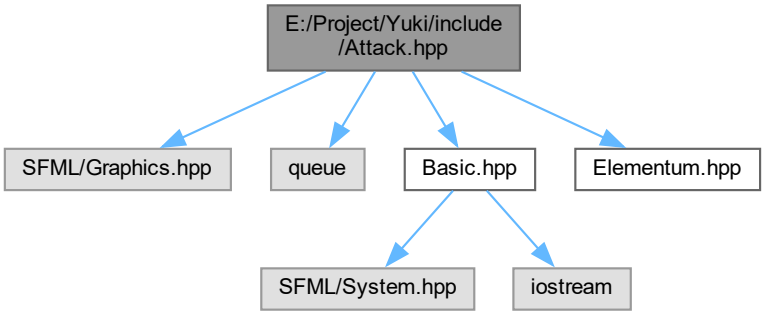
Chapter 7

File Documentation

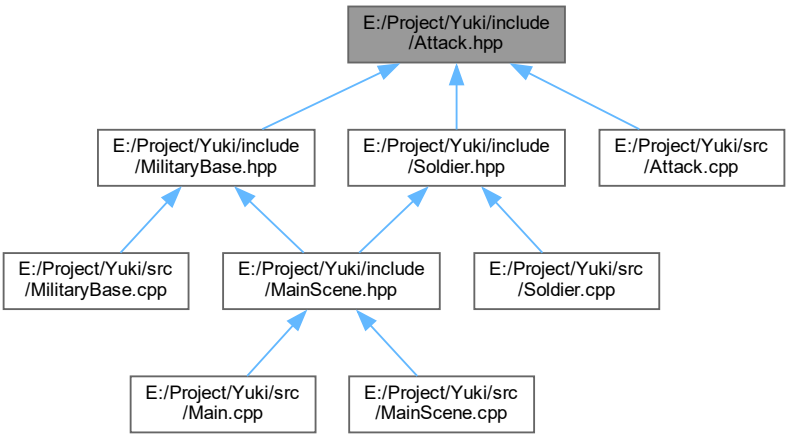
7.1 E:/Project/Yuki/include/Attack.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <queue>
#include "Basic.hpp"
#include "Elementum.hpp"
```

Include dependency graph for Attack.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct yuki::AttackInfo
- class yuki::BasicAttack
- class yuki::GranuleAttack

Namespaces

- namespace yuki

7.2 Attack.hpp

Go to the documentation of this file.

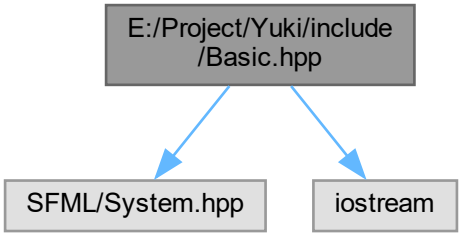
```
#ifndef ATTACK_HPP
#define ATTACK_HPP

#include <SFML/Graphics.hpp>
#include <queue>

#include "Basic.hpp"
#include "Elementum.hpp"

namespace yuki {

struct AttackInfo {
  ElementumType elementum_type;
  float elementum_level;
  float damage;

  AttackInfo(ElementumType elementum_type = ElementumType::None,
             float elementum_level = 0.0f, float damage = 0.0f)
      : elementum_type(elementum_type),
        elementum_level(elementum_level),
        damage(damage) {}
};

class BasicAttack {
 public:
  BasicAttack() = default;
  BasicAttack(ElementumType elementum_type, float elementum_level,
```

```
                float damage) {
    info_.elementum_type = elementum_type;
    info_.elementum_level = elementum_level;
    info_.damage = damage;
  }
  BasicAttack(const AttackInfo& info) : info_(info) {}

  const auto& getAttackInfo() const { return info_; }

  const auto getElementum() const {
    return Elementum(info_.elementum_type, info_.elementum_level);
  }
  const auto getDamage() const { return info_.damage; }

  virtual ~BasicAttack() {}

 protected:
  AttackInfo info_;
};

class GranuleAttack : public BasicAttack, public sf::Drawable {
 public:
  GranuleAttack() { updateGranule(); }
  GranuleAttack(const sf::Vector2f& position, float speed, Direction direction,
                const AttackInfo& attck_info)
      : BasicAttack(attck_info) {
    speed_ = speed;
    direction_ = direction;
    granule_.setPosition(position);
    updateGranule();
  }

  virtual ~GranuleAttack() {}

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

  virtual void update();

  auto getPosition() const { return granule_.getPosition(); }

 protected:
  sf::CircleShape granule_;
  float speed_;
  Direction direction_;

  void updateGranule();
};

}  // namespace yuki

#endif  // ATTACK_HPP
```

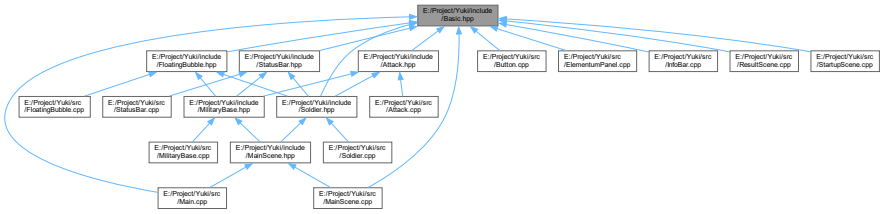
7.3 E:/Project/Yuki/include/Basic.hpp File Reference

```
#include <SFML/System.hpp>
#include <iostream>
```

Include dependency graph for Basic.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace yuki
- namespace yuki::GameFont
- namespace yuki::YukiColor

Enumerations

- enum class yuki::Direction {
  yuki::Down = 0 , yuki::Left , yuki::Right , yuki::Up ,
  yuki::Other }
- enum class yuki::Camp { yuki::Own , yuki::Enemy , yuki::Other }

Functions

- auto yuki::isOppositeDirection (const Direction direction1, const Direction direction2)
- auto yuki::GameFont::initFont (const sf::String &font_path)
- static const sf::Color yuki::YukiColor::Transparent_50 (255, 255, 255, 128)
- static const sf::Color yuki::YukiColor::Transparent (255, 255, 255, 0)
- static const sf::Color yuki::YukiColor::Transparent_25 (255, 255, 255, 192)
- static const sf::Color yuki::YukiColor::Transparent_75 (255, 255, 255, 64)
- static const sf::Color yuki::YukiColor::Normal (255, 255, 255, 255)
- static const sf::Color yuki::YukiColor::Gray (128, 128, 128, 255)
- sf::Vector2f yuki::getVectorSpeed (const Direction direction, const float speed)

Variables

- constexpr auto yuki::kWindowWidth = 960
- constexpr auto yuki::kWindowHeight = 640
- static const auto yuki::GameFont::DefaultFont = initFont("assets/font/ARLRDBD.ttf")

7.4 Basic.hpp

Go to the documentation of this file.

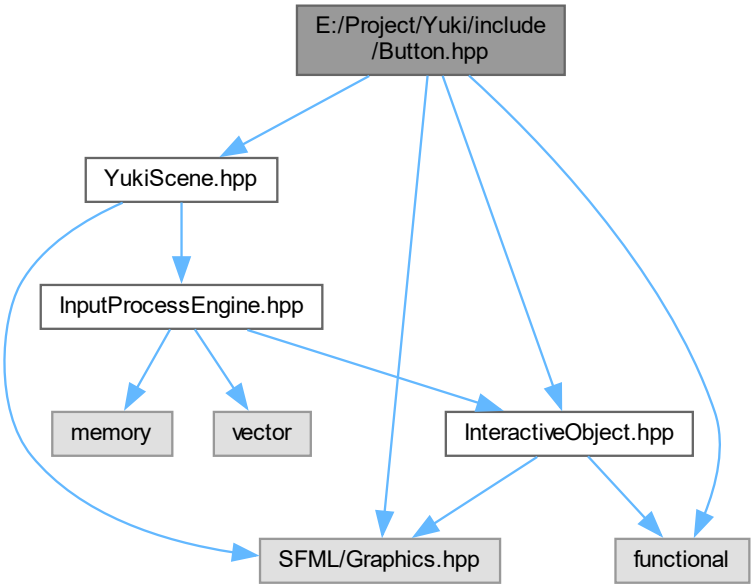
```
#ifndef BASIC_HPP
#define BASIC_HPP

#include <SFML/System.hpp>
#include <iostream>

namespace yuki {

constexpr auto kWindowWidth = 960;
constexpr auto kWindowHeight = 640;

enum class Direction { Down = 0, Left, Right, Up, Other };

inline auto isOppositeDirection(const Direction direction1,
                                const Direction direction2) {
  return (direction1 == Direction::Down && direction2 == Direction::Up) ||
         (direction1 == Direction::Up && direction2 == Direction::Down) ||
         (direction1 == Direction::Left && direction2 == Direction::Right) ||
         (direction1 == Direction::Right && direction2 == Direction::Left);
}

enum class Camp { Own, Enemy, Other };

namespace GameFont {
inline auto initFont(const sf::String& font_path) {
  sf::Font font;
  font.loadFromFile(font_path);
  if (font.getInfo().family == "") {
    std::cerr « "Font not found" « std::endl;
    throw std::runtime_error("Font not found");
  } else {
    std::cout « "Font loaded: " « font.getInfo().family « std::endl;
  }
  return font;
}

const static auto DefaultFont = initFont("assets/font/ARLRDBD.ttf");
}  // namespace GameFont

namespace YukiColor {
const static sf::Color Transparent_50(255, 255, 255, 128);
const static sf::Color Transparent(255, 255, 255, 0);
const static sf::Color Transparent_25(255, 255, 255, 192);
const static sf::Color Transparent_75(255, 255, 255, 64);
const static sf::Color Normal(255, 255, 255, 255);
const static sf::Color Gray(128, 128, 128, 255);
}  // namespace YukiColor

inline sf::Vector2f getVectorSpeed(const Direction direction,
                                   const float speed) {
  switch (direction) {
    case Direction::Down:
      return sf::Vector2f(0.f, speed);
    case Direction::Left:
      return sf::Vector2f(-speed, 0.f);
    case Direction::Right:
      return sf::Vector2f(speed, 0.f);
    case Direction::Up:
      return sf::Vector2f(0.f, -speed);
    default:
      return sf::Vector2f(0.f, 0.f);
  }
}

}  // namespace yuki

#endif
```

7.5 E:/Project/Yuki/include/Button.hpp File Reference

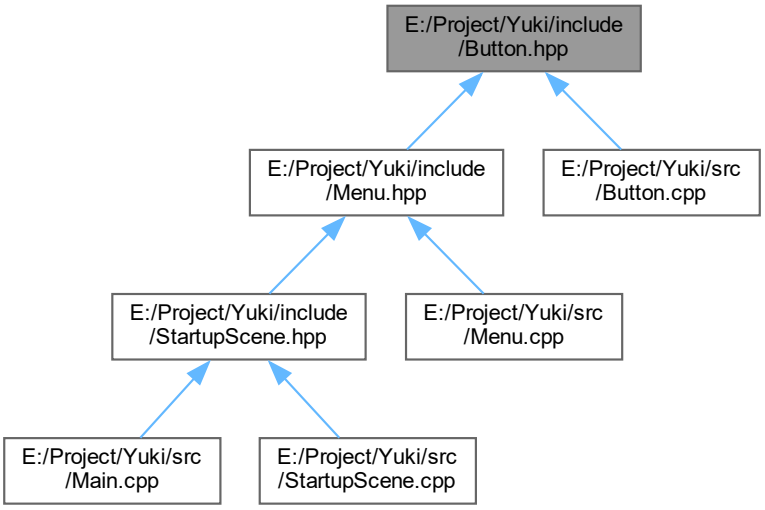
```
#include <SFML/Graphics.hpp>
#include <functional>
#include "InteractiveObject.hpp"
```

```
#include "YukiScene.hpp"
```

Include dependency graph for Button.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::Button

Namespaces

- namespace yuki

7.6 Button.hpp

Go to the documentation of this file.

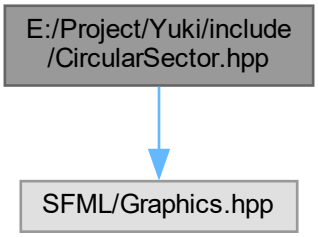
```
#ifndef BUTTON_HPP
#define BUTTON_HPP

#include <SFML/Graphics.hpp>
#include <functional>

#include "InteractiveObject.hpp"
#include "YukiScene.hpp"


namespace yuki {

class Button : public sf::Drawable, public Touchable {
 public:
  enum class Alignment { Left, Center, Right };

  Button();
  Button(const sf::Vector2f& size);
  Button(const sf::String& text, const Alignment alignment = Alignment::Center);
  Button(const sf::Vector2f& size, const sf::String& text,
         const Alignment Alignment = Alignment::Center);
  // template <typename... Args>
  // Button(YukiScene& scene, Args&&... args)
  //     : Button(std::forward<Args>(args)...) {
  //   scene.registerTouchableObject(this);
  // }

  virtual ~Button();

  // Interact
  TOUCHABLE_OBJECT(Button)

  // Background
  void setSize(const sf::Vector2f& size) {
    shape_.setSize(size);
    adjustTextPosition();
  }
  const auto getSize() const { return shape_.getSize(); }

  void setPosition(const sf::Vector2f& position) {
    shape_.setPosition(position);
    adjustTextPosition();
  }
  void setPosition(float x, float y) { setPosition(sf::Vector2f(x, y)); }

  const auto getPosition() const { return shape_.getPosition(); }

  void setFillColor(const sf::Color& color) { shape_.setFillColor(color); }

  void setOutlineColor(const sf::Color& color) {
    shape_.setOutlineColor(color);
  }

  void setOutlineThickness(float thickness) {
    shape_.setOutlineThickness(thickness);
  }

  void setOrigin(const sf::Vector2f& origin) {
    shape_.setOrigin(origin);
    text_.setOrigin(origin);
  }
  const auto getOrigin() const { return shape_.getOrigin(); }

  void setTexture(const sf::Texture* texture) { shape_.setTexture(texture); }
  const auto getTexture() const { return shape_.getTexture(); }

  void setTextureRect(const sf::IntRect& rect) { shape_.setTextureRect(rect); }

  // Text
  void setString(const sf::String& text) { text_.setString(text); }
  const auto getString() const { return text_.getString(); }
  void setTextSize(unsigned int size) { text_.setCharacterSize(size); }
  void setTextFont(const sf::Font& font) { text_.setFont(font); }
```

```
  void setTextFillColor(const sf::Color& color) { text_.setFillColor(color); }
  void setTextOutlineColor(const sf::Color& color) {
    text_.setOutlineColor(color);
  }
  void setTextOutlineThickness(float thickness) {
    text_.setOutlineThickness(thickness);
  }
  void setTextAlignment(Alignment alignment) {
    alignment_ = alignment;
    adjustTextPosition();
  }

  void autoAdjustTextSize();

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

 protected:
  sf::RectangleShape shape_;
  sf::Text text_;

  Alignment alignment_ = Alignment::Center;
  void adjustTextPosition();
};

}  // namespace yuki

#endif
```

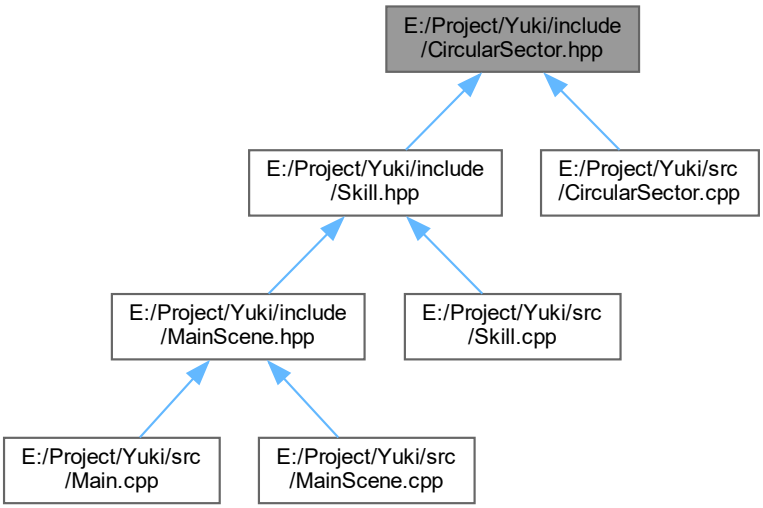
7.7 E:/Project/Yuki/include/CircularSector.hpp File Reference

`#include <SFML/Graphics.hpp>`

Include dependency graph for CircularSector.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::CircularSector

Namespaces

- namespace yuki

7.8 CircularSector.hpp

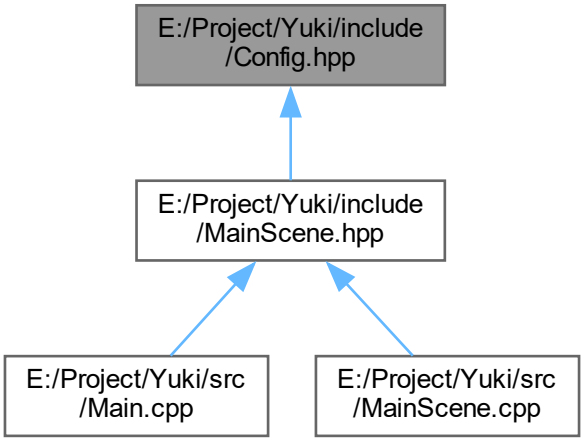
Go to the documentation of this file.

```
#ifndef CIRCULAR_SECTOR_HPP
#define CIRCULAR_SECTOR_HPP

#include <SFML/Graphics.hpp>

namespace yuki {

class CircularSector : public sf::Shape {
 public:
  CircularSector(const sf::Vector2f& center, float radius, int start_angle,
                 int end_angle)
      : center_(center),
        radius_(radius),
        start_angle_(start_angle),
        end_angle_(end_angle) {
    update();
  }

  void setCenter(const sf::Vector2f& center) {
    center_ = center;
    update();
  }

  void setRadius(float radius) {
    radius_ = radius;
```

```
    update();
  }

  void setStartAngle(float start_angle) {
    start_angle_ = start_angle;
    update();
  }

  void setEndAngle(float end_angle) {
    end_angle_ = end_angle;
    update();
  }

  const sf::Vector2f& getCenter() const { return center_; }

  sf::Vector2f getPoint(std::size_t index) const override;
  std::size_t getPointCount() const override {
    return end_angle_ - start_angle_ + 2;
  }

 private:
  sf::Vector2f center_;
  float radius_;
  int start_angle_;
  int end_angle_;
};

}  // namespace yuki

#endif  // CIRCULAR_SECTOR_HPP
```

7.9 E:/Project/Yuki/include/Config.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define YUKI_DEBUG

7.9.1 Macro Definition Documentation

7.9.1.1 YUKI_DEBUG

```
#define YUKI_DEBUG
```

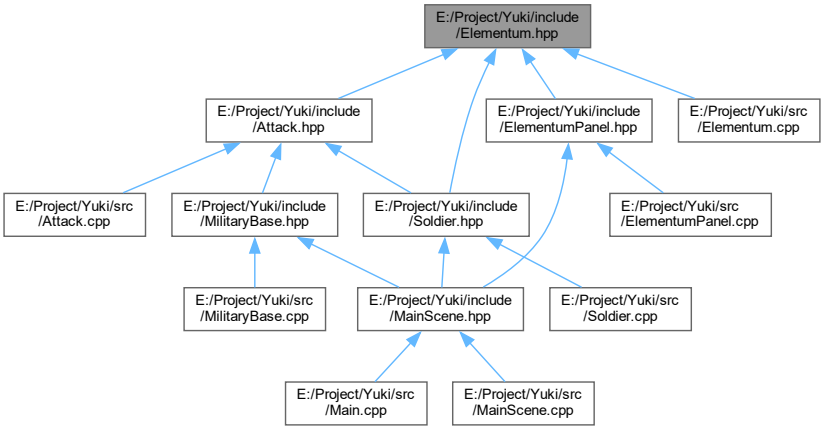
7.10 Config.hpp

Go to the documentation of this file.

```
00001 #ifndef CONFIG_HPP
00002 #define CONFIG_HPP
00003
00004 #define YUKI_DEBUG
00005
00006 #endif // CONFIG_HPP
```

7.11 E:/Project/Yuki/include/Elementum.hpp File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct yuki::Elementum
- struct yuki::ElementumReaction

Namespaces

- namespace yuki

Enumerations

- enum class yuki::ElementumType { yuki::None , yuki::Pyro , yuki::Hydro , yuki::Cyro }
- enum class yuki::ElementumReactionType { yuki::None , yuki::Evaporation , yuki::Freeze , yuki::Melt }

Functions

- ElementumReactionType yuki::getElementumReactionType (const Elementum &lhs, const Elementum &rhs)
- Elementum yuki::getReactedElementum (const Elementum &lhs, const Elementum &rhs)
- ElementumReaction yuki::getElementumReaction (const Elementum &lhs, const Elementum &rhs)

7.12 Elementum.hpp

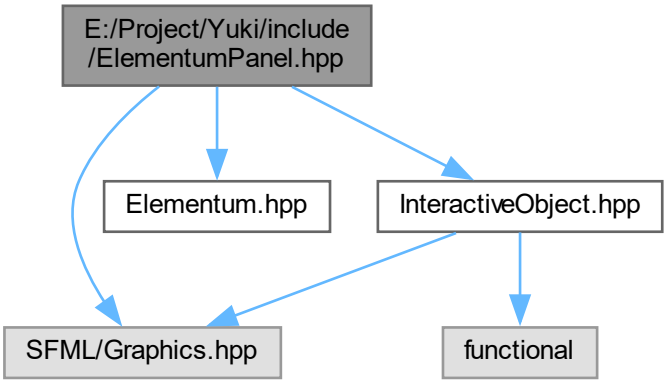
Go to the documentation of this file.

```
#ifndef ELEMENTUM_HPP
#define ELEMENTUM_HPP

namespace yuki {

enum class ElementumType { None, Pyro, Hydro, Cyro };
enum class ElementumReactionType { None, Evaporation, Freeze, Melt };

struct Elementum {
  Elementum(ElementumType type = ElementumType::None, float level = 0.0f)
      : type(type), level(level) {}

  ElementumType type;
  float level;
};

struct ElementumReaction {
  ElementumReactionType type;
  Elementum elementum;
};

ElementumReactionType getElementumReactionType(const Elementum& lhs,
                                               const Elementum& rhs);

Elementum getReactedElementum(const Elementum& lhs, const Elementum& rhs);

ElementumReaction getElementumReaction(const Elementum& lhs,
                                       const Elementum& rhs);

}  // namespace yuki

#endif  // ELEMENTUM_HPP
```

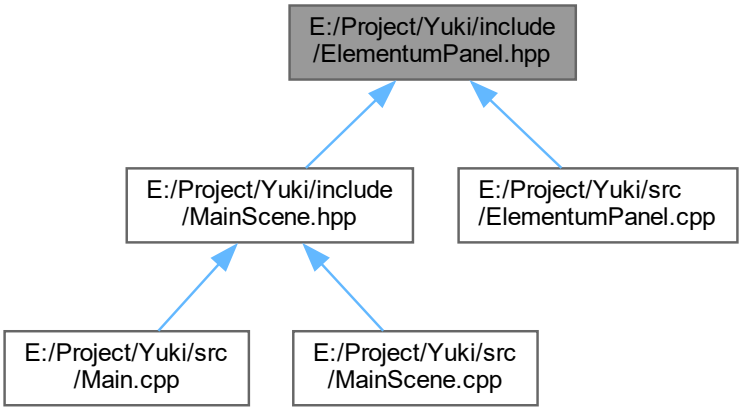
7.13 E:/Project/Yuki/include/ElementumPanel.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Elementum.hpp"
#include "InteractiveObject.hpp"
```

Include dependency graph for ElementumPanel.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::ElementumPanel

Namespaces

- namespace yuki

7.14 ElementumPanel.hpp

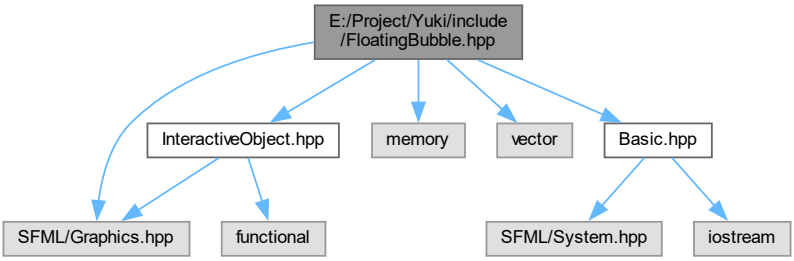
Go to the documentation of this file.

```
#ifndef ELEMENTUM_PANEL_HPP
#define ELEMENTUM_PANEL_HPP

#include <SFML/Graphics.hpp>

#include "Elementum.hpp"
#include "InteractiveObject.hpp"

namespace yuki {
class ElementumPanel : public sf::CircleShape, public Touchable {
 public:
  ElementumPanel(ElementumType type = ElementumType::Pyro) {
    setElementumType(type);
  }

  TOUCHABLE_OBJECT(ElementumPanel)

  void activate();
  void deactivate();

  void setElementumType(ElementumType type);
  constexpr auto getElementumType() const { return elementum_type_; }

 private:
  ElementumType elementum_type_;
};

}  // namespace yuki

#endif
```

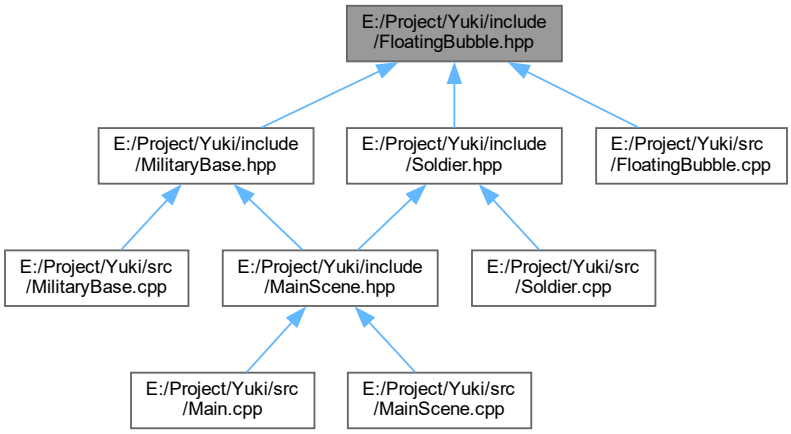
7.15 E:/Project/Yuki/include/FloatingBubble.hpp File Reference

```
#include <InteractiveObject.hpp>
#include <SFML/Graphics.hpp>
#include <memory>
#include <vector>
#include "Basic.hpp"
```

Include dependency graph for FloatingBubble.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::BubbleItem
- class yuki::FloatingBubble

Namespaces

- namespace yuki

7.16 FloatingBubble.hpp

Go to the documentation of this file.

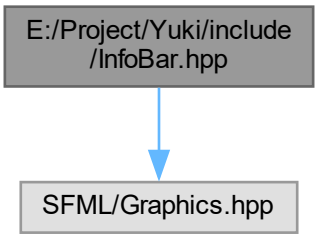
```
#ifndef FLOATING_BUBBLE_HPP
#define FLOATING_BUBBLE_HPP

#include <InteractiveObject.hpp>
#include <SFML/Graphics.hpp>
#include <memory>
#include <vector>

#include "Basic.hpp"

namespace yuki {

class BubbleItem : public sf::CircleShape, public Touchable {
  friend class FloatingBubble;

 public:
  BubbleItem() = default;
  BubbleItem(float radius) : sf::CircleShape(radius) {}
  virtual ~BubbleItem() = default;
  TOUCHABLE_OBJECT(BubbleItem)

  void loadTexture(const std::string& path) {
    texture_.loadFromFile(path);
    setTexture(&texture_);
  }
 private:
  sf::Texture texture_;
};

```

```
class FloatingBubble : public sf::Drawable {
 public:
  FloatingBubble(const sf::Vector2f& position, const float distribute_radius,
                 const int bubble_num, const float bubble_radius = 10.f,
                 const Direction direction = Direction::Down)
      : position_(position),
        distribute_radius_(distribute_radius),
        bubble_num_(bubble_num),
        bubble_radius_(bubble_radius),
        direction_(direction) {
    for (int i = 0; i < bubble_num_; ++i) {
      auto bubble = std::make_unique<BubbleItem>(bubble_radius);
      bubble->setFillColor(YukiColor::Transparent_25);
      bubble->setOutlineColor(sf::Color::Black);
      bubble->setOutlineThickness(2.f);
      bubble->setPosition(getBubblePosition(i));
      bubbles_.push_back(std::move(bubble));
    }
  }
  FloatingBubble() = default;

  //   TOUCHABLE_OBJECT(FloatingBubble)

  void setPosition(const sf::Vector2f& position) {
    position_ = position;
    updateBubblesPosition();
  }
  const auto getPosition() const { return position_; }

  void setBubbleRadius(const float radius) {
    bubble_radius_ = radius;
    for (auto& bubble : bubbles_) {
      bubble->setRadius(radius);
    }
  }
  const auto getBubbleRadius() const { return bubble_radius_; }
  int getIndexByPosition(const sf::Vector2f& position) const;

  auto& getBubbleItems() { return bubbles_; }
  const auto& getBubbleItems() const { return bubbles_; }

  void draw(sf::RenderTarget& target, sf::RenderStates states) const override;

 private:
  sf::Vector2f position_;
  float distribute_radius_;
  int bubble_num_;
  float bubble_radius_;
  Direction direction_;
  std::vector<std::unique_ptr<BubbleItem>> bubbles_;

  sf::Vector2f getBubblePosition(const int index);
  void updateBubblesPosition();
};

}  // namespace yuki

#endif  // FLOATING_BUBBLE_HPP
```

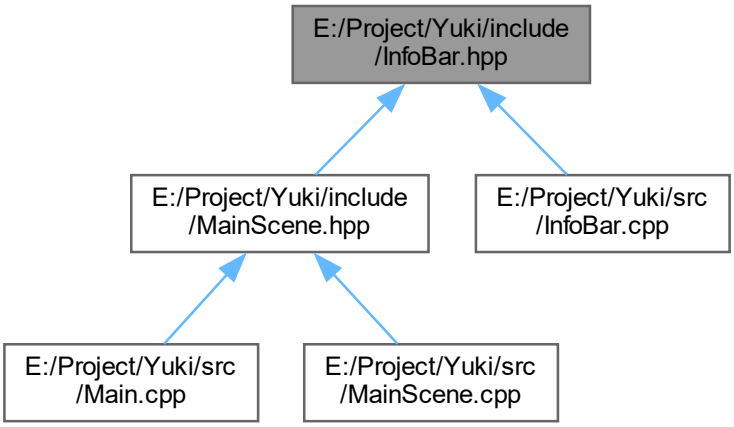
7.17 E:/Project/Yuki/include/InfoBar.hpp File Reference

```
#include <SFML/Graphics.hpp>
```

Include dependency graph for InfoBar.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::InfoBar

Namespaces

- namespace yuki

7.18 InfoBar.hpp

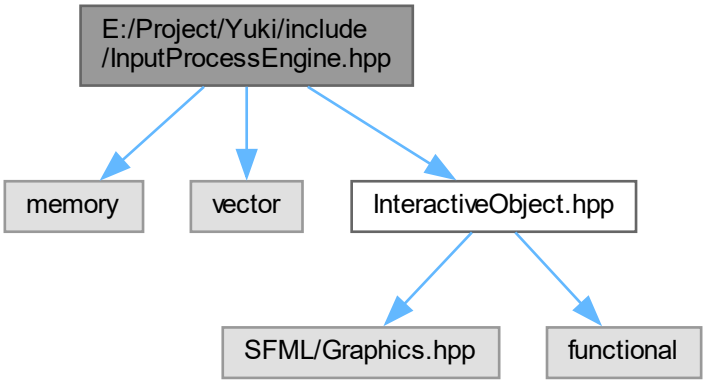
Go to the documentation of this file.

```
#ifndef INFO_BAR_HPP
#define INFO_BAR_HPP

#include <SFML/Graphics.hpp>

namespace yuki {

class InfoBar : public sf::Drawable {
 private:
  constexpr static float kIconSize = 32.f;
  constexpr static float kMoneyIconOffset = 10.f;
  constexpr static float kMoneyTextOffset = 60.f;

 public:
  InfoBar() : InfoBar(sf::Vector2f(0.f, 0.f)) {}
  InfoBar(const sf::Vector2f& position,
          const sf::Vector2f& size = sf::Vector2f(256.f, 32.f));
  virtual ~InfoBar() = default;

  void setMoney(const int money);
  void setPosition(const sf::Vector2f& position);
  const auto& getPosition() const { return background_.getPosition(); }

  void setSize(const sf::Vector2f& size) {
    background_.setSize(size);
  }
  const auto& getSize() const { return background_.getSize(); }


 private:
  sf::RectangleShape background_;
  sf::Texture background_texture_;
  sf::Sprite icon_money_;
  sf::Texture icon_money_texture_;
  sf::Text text_money_;

  void initTextures();
  void initUi();
  void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
};

}  // namespace yuki

#endif  // INFO_BAR_HPP
```

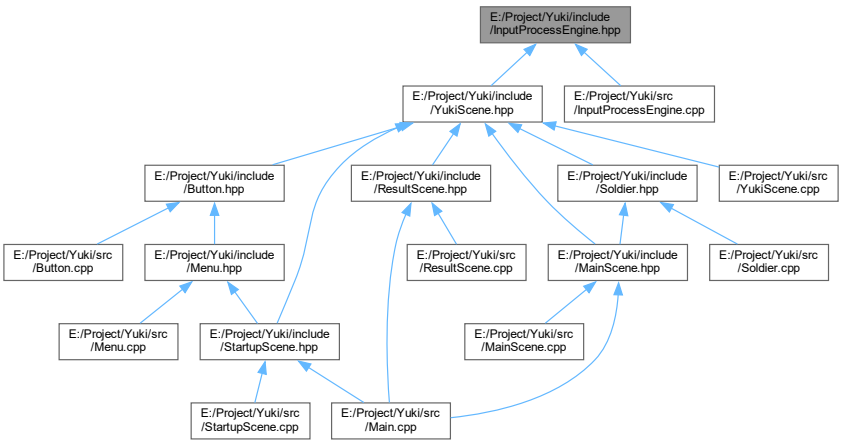
7.19 E:/Project/Yuki/include/InputProcessEngine.hpp File Reference

```
#include <memory>
#include <vector>
#include "InteractiveObject.hpp"
```

Include dependency graph for InputProcessEngine.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::MouseEventProcessEngine

Namespaces

- namespace yuki

7.20 InputProcessEngine.hpp

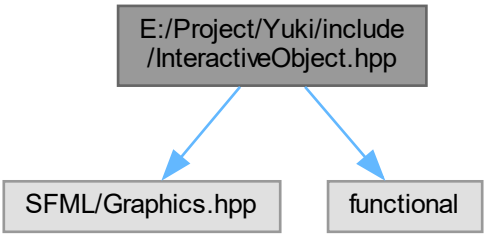
Go to the documentation of this file.

```
#ifndef INPUT_PROCESS_ENGINE_HPP
#define INPUT_PROCESS_ENGINE_HPP

#include <memory>
#include <vector>

#include "InteractiveObject.hpp"


namespace yuki {

class MouseEventProcessEngine {
 public:
  MouseEventProcessEngine();
  virtual ~MouseEventProcessEngine();

  void registerTouchableObject(Touchable* object);
  void unregisterTouchableObject(Touchable* object);

  void processEvent(sf::Event event);

 private:
  std::vector<Touchable*> touchable_objects_;
};

}  // namespace yuki

#endif
```

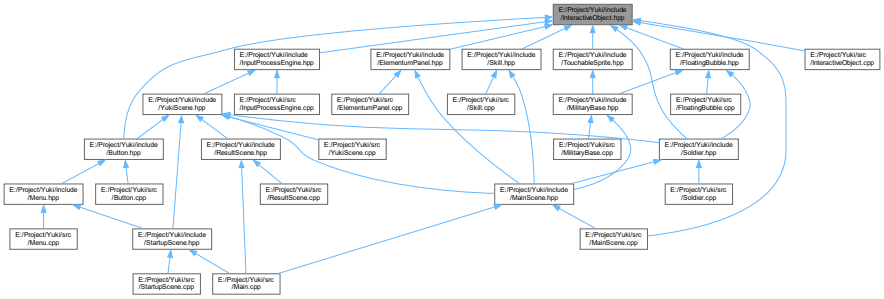
7.21 E:/Project/Yuki/include/InteractiveObject.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <functional>
```

Include dependency graph for InteractiveObject.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::Touchable
- class yuki::Focusable
- class yuki::TouchableObject< Decorated >

Namespaces

- namespace yuki

Macros

- #define TOUCHABLE_OBJECT(OBJECT_NAME)
- #define FOCUSABLE_OBJECT(OBJECT_NAME) TOUCHABLE_OBJECT(OBJECT_NAME)

7.21.1 Macro Definition Documentation

7.21.1.1 FOCUSABLE_OBJECT

```
#define FOCUSABLE_OBJECT(
            OBJECT_NAME ) TOUCHABLE_OBJECT(OBJECT_NAME)
```

7.21.1.2 TOUCHABLE_OBJECT

```
#define TOUCHABLE_OBJECT(
            OBJECT_NAME )
```

Value:

```
 public:                                \
  virtual bool inRange(const sf::Vector2f& position) const override;
```

7.22 InteractiveObject.hpp

Go to the documentation of this file.

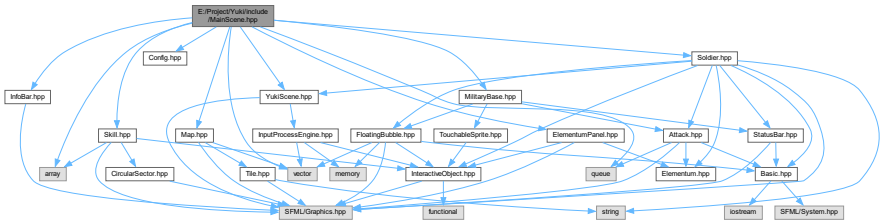
```
#ifndef INTERACTIVE_OBJECT_HPP
#define INTERACTIVE_OBJECT_HPP

#include <SFML/Graphics.hpp>
#include <functional>

#define TOUCHABLE_OBJECT(OBJECT_NAME) \
 public:                              \
  virtual bool inRange(const sf::Vector2f& position) const override;

#define FOCUSABLE_OBJECT(OBJECT_NAME) TOUCHABLE_OBJECT(OBJECT_NAME)

namespace yuki {

class Touchable {
 public:
  using Callback = std::function<void(sf::Event e)>;
  // using Callback = void (*)(sf::Event e);

  Touchable() = default;
  virtual ~Touchable() {}

  virtual void onRelease(sf::Event e) { on_releaseCallback_(e); }
  virtual void onClick(sf::Event e) { on_clickCallback_(e); }
  virtual void onHover(sf::Event e) { on_hoverCallback_(e); }
  virtual void onLeave(sf::Event e) { on_leaveCallback_(e); }

  enum class Action { Click, Release, Hover, Leave };

  void bindAction(const Action action, Callback callback);

  void bindClick(Callback callback) { on_clickCallback_ = callback; }
  void bindRelease(Callback callback) { on_releaseCallback_ = callback; }
  void bindHover(Callback callback) { on_hoverCallback_ = callback; }
  void bindLeave(Callback callback) { on_leaveCallback_ = callback; }

  virtual bool inRange(const sf::Vector2f& position) const = 0;
  virtual void updateStatus(sf::Event event);

 protected:
  bool is_clicked_ = false;
  bool is_hovered_ = false;

  Callback on_clickCallback_ = [](sf::Event) {};
  Callback on_releaseCallback_ = [](sf::Event) {};
  Callback on_hoverCallback_ = [](sf::Event) {};
  Callback on_leaveCallback_ = [](sf::Event) {};
};

class Focusable : public Touchable {
 public:
  using Callback = std::function<void(sf::Event e)>;

  Focusable() = default;
  virtual ~Focusable() {}

  virtual void onFocus(sf::Event e) { on_focusCallback_(e); }
  virtual void onUnfocus(sf::Event e) { on_unfocusCallback_(e); }

  constexpr auto isFocused() const { return is_focused_; }

  void bindFocus(Callback callback) { on_focusCallback_ = callback; }
  void bindUnfocus(Callback callback) { on_unfocusCallback_ = callback; }

  void updateStatus(sf::Event event);

 private:
  bool is_focused_ = false;

  Callback on_focusCallback_ = [](sf::Event) {};
  Callback on_unfocusCallback_ = [](sf::Event) {};
};

template <class Decorated>
class TouchableObject : public Decorated, public Touchable {
 public:
  TouchableObject() = default;
  virtual ~TouchableObject() {}

  virtual bool inRange(const sf::Vector2f& position) const override {
    return Decorated::getGlobalBounds().contains(position);
  }
```

```
00083
00084   virtual void updateStatus(sf::Event event) override {
00085     Touchable::updateStatus(event);
00086   }
00087 };
00088
00089 }  // namespace yuki
00090
00091 #endif  // INTERACTIVE_OBJECT_HPP
```

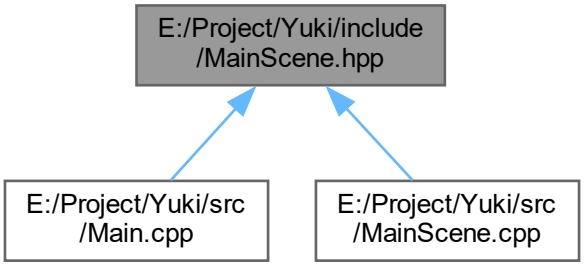
7.23 E:/Project/Yuki/include/MainScene.hpp File Reference

```
#include <array>
#include <queue>
#include <vector>
#include "Config.hpp"
#include "ElementumPanel.hpp"
#include "InfoBar.hpp"
#include "Map.hpp"
#include "MilitaryBase.hpp"
#include "Skill.hpp"
#include "Soldier.hpp"
#include "YukiScene.hpp"
```

Include dependency graph for MainScene.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::MainScene

Namespaces

- namespace yuki

7.24 MainScene.hpp

Go to the documentation of this file.

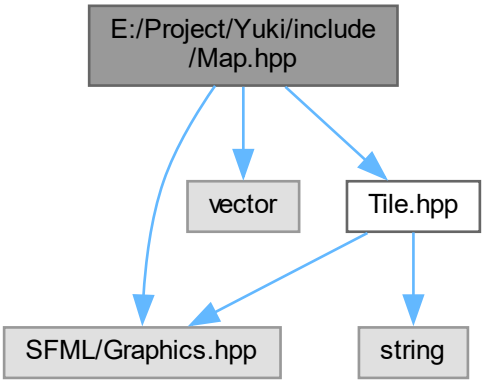
```
#ifndef MAIN_SCENE_HPP
#define MAIN_SCENE_HPP

#include <array>
#include <queue>
#include <vector>

#include "Config.hpp"
#include "ElementumPanel.hpp"
#include "InfoBar.hpp"
#include "Map.hpp"
#include "MilitaryBase.hpp"
#include "Skill.hpp"
#include "Soldier.hpp"
#include "YukiScene.hpp"


namespace yuki {

class MainScene : public YukiScene {
  YUKI_SCENE

 public:
  enum class Message {
    GenerateOwnSoldier,
    GenerateEnemySoldier,
    OwnBaseLevelUp,
    FocusedObjectChanged,
    OwnWin,
    EnemyWin
  };

  enum class ObjectType {
    OwnSoldier,
    EnemySoldier,
    OwnBase,
    EnemyBase,
    None
  };

 public:
  MainScene(sf::RenderWindow& window);
  virtual ~MainScene() {}

  // void generateSoldier();
  void sendMessage(const Message& message) { message_queue_.push(message); }

 private:
  const static sf::Vector2i kOwnSoldierBirthCoordinate;
  const static sf::Vector2i kEnemySoldierBirthCoordinate;
  const static sf::Vector2i kOwnBaseCoordinate;
  const static sf::Vector2i kEnemyBaseCoordinate;

  using GranuleAttackWithSenderInfo = std::pair<Soldier*, GranuleAttack>;

  Map map_;
  std::unique_ptr<MilitaryBase> own_base_;
  std::unique_ptr<MilitaryBase> enemy_base_;
  std::unique_ptr<InfoBar> info_bar_;
  std::unique_ptr<SkillBar<3» skill_bar_;

  int own_money_;
  int enemy_money_;

  void increaseOwnMoney(int money) { increaseMoney(money, Camp::Own); }
  void decreaseOwnMoney(int money) { decreaseMoney(money, Camp::Own); }
  void increaseEnemyMoney(int money) { increaseMoney(money, Camp::Enemy); }
  void decreaseEnemyMoney(int money) { decreaseMoney(money, Camp::Enemy); }

  std::vector<std::unique_ptr<Soldier» soldiers_;
  std::vector<std::unique_ptr<Soldier» enemies_;

  Focusable* focused_object_;
```

```
  ObjectType focused_object_type_;

  void initUi();
  void initMap();
  void initBuildings();
  void initInfoHint();

  void setMoney(int money, Camp camp = Camp::Own);
  void increaseMoney(int money, Camp camp = Camp::Own);
  void decreaseMoney(int money, Camp camp = Camp::Own);

  // std::shared_ptr<yuki::Soldier> getDefaultSoldier(Camp camp = Camp::Own);
  std::unique_ptr<yuki::Soldier> generateSoldier(const std::string& name,
                                                 Camp camp = Camp::Own);

  void eraseDeadSoldier();
  // void sodierAdjustDirection();
  void soldierUpdatePosition();
  void updateAttacks();
  void enemyBaseMakeDecision();

  sf::Vector2f coordinateToPixel(const sf::Vector2i& coordinate);
  sf::Vector2i pixelToCoordinate(const sf::Vector2f& pixel_position);

  std::vector<GranuleAttackWithSenderInfo> granule_attacks_;
  void registerGranuleAttack(Soldier* soldier, const GranuleAttack& attack) {
    granule_attacks_.push_back(std::make_pair(soldier, attack));
  }

  // Elementum Panel
  std::array<ElementumPanel, 3> elementum_panels_;
  ElementumType own_current_elementum_type_;
  ElementumType enemy_current_elementum_type_;
  void initElementumPanel();
  void setOwnCurrentElementumType(ElementumType type);
  void setEnemyCurrentElementumType(ElementumType type);

  // Attack Button
  void initButton();
  TouchableObject<sf::CircleShape> attack_button_;
  sf::Texture attack_button_texture_;

  std::queue<Message> message_queue_;

  void implMsgGenerateOwnSoldier();
  void implMsgGenerateEnemySoldier();

  void normalSoldierAttack(NormalSoldier* attacker);

#ifdef YUKI_DEBUG
  sf::Text debug_text_;
#endif
};

}  // namespace yuki

#endif  // MAIN_SCENE_HPP
```

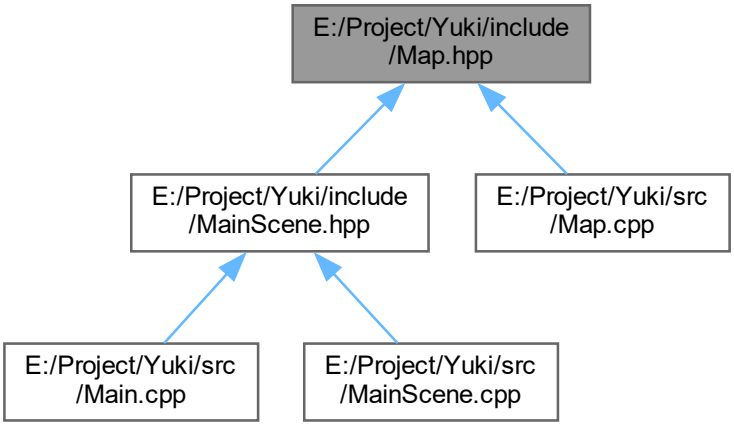
7.25 E:/Project/Yuki/include/Map.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <vector>
#include "Tile.hpp"
```

Include dependency graph for Map.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::Map

Namespaces

- namespace yuki

7.26 Map.hpp

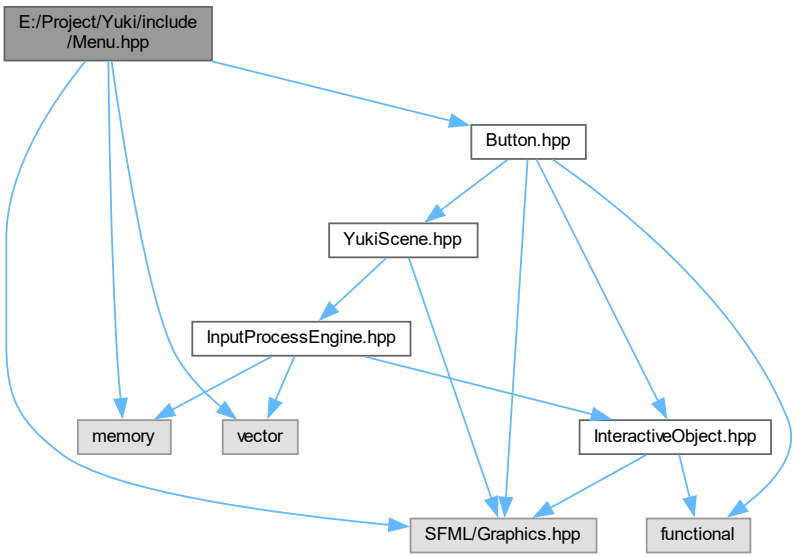
Go to the documentation of this file.

```
#ifndef MAP_HPP
#define MAP_HPP

#include <SFML/Graphics.hpp>
#include <vector>

#include "Tile.hpp"

namespace yuki {

class Map : public sf::Drawable {
 public:
  Map(const sf::Vector2i& size, const sf::Vector2f& tile_size);
  virtual ~Map();

  void draw(sf::RenderTarget& target, sf::RenderStates states) const override;

  void setSize(const sf::Vector2i& size) { size_ = size; }
  const auto& getSize() const { return size_; }
  void setTileSize(const sf::Vector2f& tile_size) { tile_size_ = tile_size; }
  const auto& getTileSize() const { return tile_size_; }

  auto getCoordinate(const sf::Vector2f& position) const {
    return sf::Vector2i(position.x / tile_size_.x, position.y / tile_size_.y);
  }

  Tile& getTile(int row, int col);
  const Tile& getTile(int row, int col) const;

  Tile& getTile(const sf::Vector2i& coordinate);
  const Tile& getTile(const sf::Vector2i& coordinate) const;

  void setTile(int row, int col, const TileInfo& tile_info);

 private:
  sf::Vector2i size_;
  sf::Vector2f tile_size_;

  std::vector<Tile> tiles_;

  sf::Texture bg_texture_;
  std::vector<sf::Sprite> bg_tiles_;
};

}  // namespace yuki

#endif
```

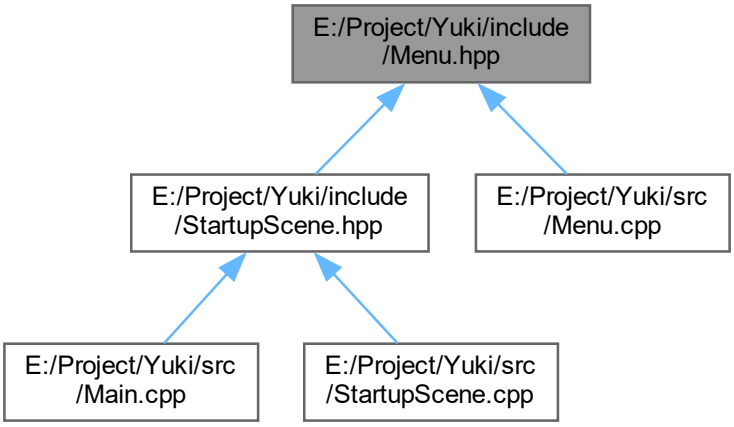
7.27 E:/Project/Yuki/include/Menu.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <memory>
#include <vector>
#include "Button.hpp"
```

Include dependency graph for Menu.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::Menu

Namespaces

- namespace yuki

7.28 Menu.hpp

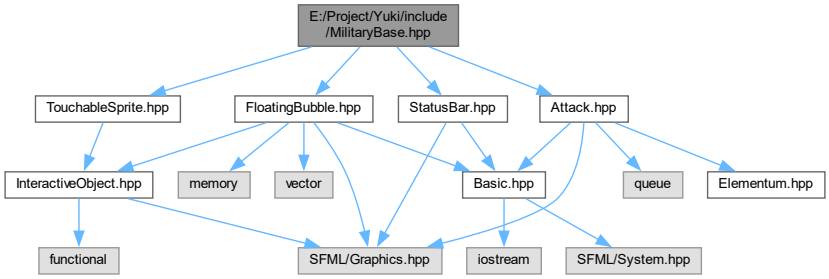
Go to the documentation of this file.

```
#ifndef MENU_HPP
#define MENU_HPP

#include <SFML/Graphics.hpp>
#include <memory>
#include <vector>
#include "Button.hpp"

namespace yuki {

class Menu : public sf::Drawable {
 public:
  enum class Layout {
    Vertical,
    Horizontal,
  };

  Menu();

  virtual ~Menu();

  void setLayout(Layout layout) { layout_ = layout; }
  const Layout& getLayout() const { return layout_; }

  void addItem(std::unique_ptr<yuki::Button> item);

  virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;

  Button& operator[](unsigned int index) { return *menu_items_[index]; }
  Button& getByName(const sf::String& name) {
    for (auto& item : menu_items_) {
      if (item->getString() == name) {
        return *item;
      }
    }
    throw std::runtime_error("Menu item not found");
    return *menu_items_[0];
  }

 private:
  std::vector<std::unique_ptr<yuki::Button» menu_items_;
  Layout layout_;
  unsigned int spacing_;

  sf::Vector2f position_;

  void adjustItemsPosition();
};

}  // namespace yuki

#endif
```

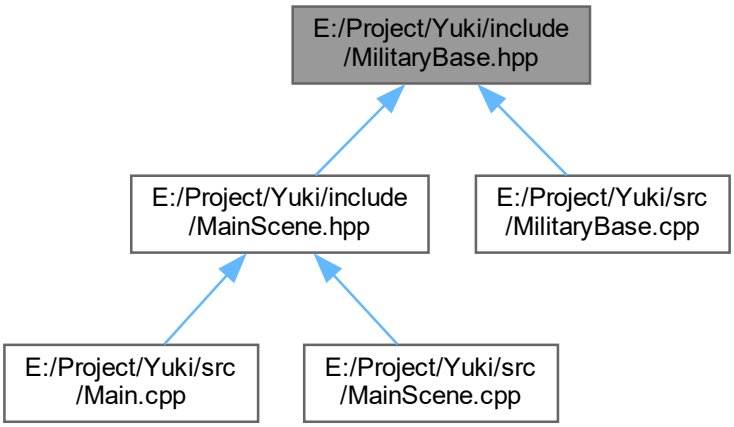
7.29 E:/Project/Yuki/include/MilitaryBase.hpp File Reference

```
#include "Attack.hpp"
#include "FloatingBubble.hpp"
#include "StatusBar.hpp"
#include "TouchableSprite.hpp"
```

Include dependency graph for MilitaryBase.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::MilitaryBase

Namespaces

- namespace yuki

7.30 MilitaryBase.hpp

Go to the documentation of this file.

```
00001 #ifndef MILITARY_BASE_HPP
00002 #define MILITARY_BASE_HPP
00003
```

```
#include "Attack.hpp"
#include "FloatingBubble.hpp"
#include "StatusBar.hpp"
#include "TouchableSprite.hpp"

namespace yuki {

class MilitaryBase : public TouchableSprite {
  TOUCHABLE_OBJECT(MilitaryBase)

  static constexpr float health_bar_offset_y = -10.f;

 public:
  MilitaryBase(Camp camp = Camp::Own);
  virtual ~MilitaryBase() {}

  void levelUp() {
    level_++;
    updateTexture();
  }
  void levelDown() {
    level_--;
    updateTexture();
  }

  auto getLevel() const { return level_; }

  void setCurrentHealth(int health) { health_bar_.setValue(health); }
  void setMaxHealth(int health) { health_bar_.setMaxValue(health); }
  constexpr auto getCurrentHealth() const { return health_bar_.getValue(); }
  constexpr auto getMaxHealth() const { return health_bar_.getMaxValue(); }

  void setPosition(const sf::Vector2f& position) {
    TouchableSprite::setPosition(position);
    floating_bubble_.setPosition({position.x + getGlobalBounds().width / 2.f,
                                  position.y + getGlobalBounds().height / 2.f});
    health_bar_.setPosition({position.x, position.y + health_bar_offset_y});
  }

  void draw(sf::RenderTarget& target, sf::RenderStates states) const override;

  void setFloatingBubbleVisible(bool visible) {
    floating_bubble_visible_ = visible;
  }
  constexpr auto getFloatingBubbleVisible() const {
    return floating_bubble_visible_;
  }
  // void setHealthBarVisible(bool visible) { health_bar_visible_ = visible; }
  // constexpr auto getHealthBarVisible() const { return health_bar_visible_; }

  int getFloatingBubbleIndexByPosition(const sf::Vector2f& position) {
    if (!floating_bubble_visible_) {
      return -1;
    }
    return floating_bubble_.getIndexByPosition(position);
  }

  auto& getFloatingBubble() { return floating_bubble_; }
  const auto& getFloatingBubble() const { return floating_bubble_; }

  bool canGenerateSoldier();
  void resetGenerateSoldierClock() { generateSoldierClock_.restart(); }

  void getAttacked(const BasicAttack& attack) {
    health_bar_.setValue(health_bar_.getValue() - attack.getDamage());
  }

 private:
  Camp camp_;
  int level_ = 1;
  sf::Texture texture_;
  FloatingBubble floating_bubble_;
  bool floating_bubble_visible_ = false;
  StatusBar health_bar_;
  sf::Clock generateSoldierClock_;

  void initTexture();
  void updateTexture();
};

}  // namespace yuki

#endif
```

7.31 E:/Project/Yuki/include/Network.hpp File Reference

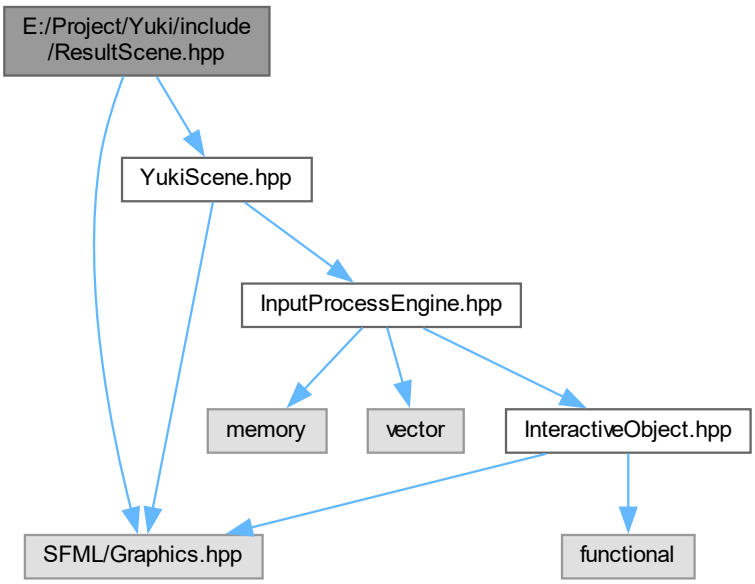
7.32 Network.hpp

Go to the documentation of this file.

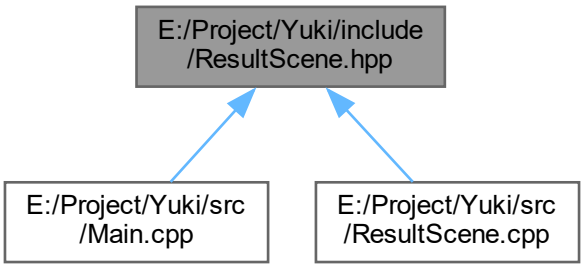
7.33 E:/Project/Yuki/include/ResultScene.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "YukiScene.hpp"
```

Include dependency graph for ResultScene.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::ResultScene

Namespaces

- namespace yuki

7.34 ResultScene.hpp

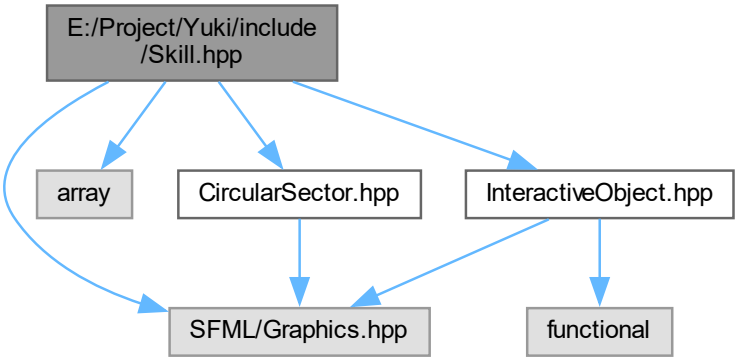
Go to the documentation of this file.

```
#ifndef RESULT_SCENE_HPP
#define RESULT_SCENE_HPP

#include <SFML/Graphics.hpp>

#include "YukiScene.hpp"

namespace yuki {

class ResultScene : public YukiScene {
  YUKI_SCENE

 public:
  ResultScene(sf::RenderWindow& window, bool own_win = true);
  virtual ~ResultScene();

 private:
  sf::Text result_text_;
  bool own_win_;

  sf::Sprite background_;
  sf::Texture background_texture_;

  void initUi();
};

}  // namespace yuki

#endif  // RESULT_SCENE_HPP
```

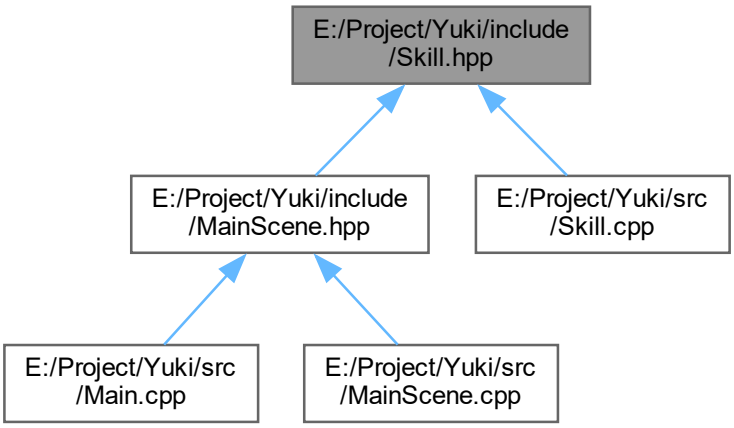
7.35 E:/Project/Yuki/include/Skill.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <array>
#include "CircularSector.hpp"
#include "InteractiveObject.hpp"
```

Include dependency graph for Skill.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::SkillIcon
- class yuki::SkillBar< MaxSkillNum >

Namespaces

- namespace yuki

Variables

- constexpr auto yuki::kPi = 3.14159265358979323846f
- constexpr auto yuki::kSkillIconBorderDistance = 3.f

7.36 Skill.hpp

Go to the documentation of this file.

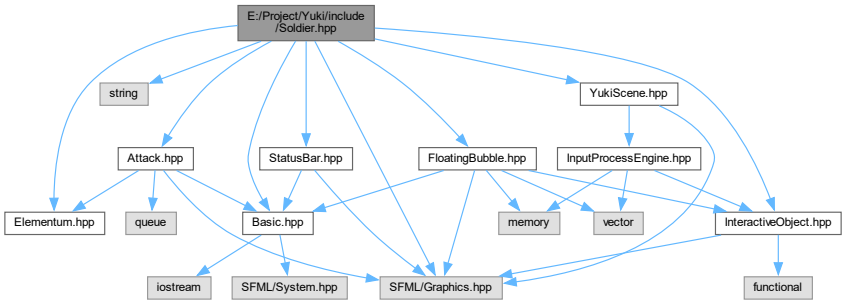
```
#ifndef SKILL_HPP
#define SKILL_HPP

#include <SFML/Graphics.hpp>
#include <array>

#include "CircularSector.hpp"
#include "InteractiveObject.hpp"

namespace yuki {

constexpr auto kPi = 3.14159265358979323846f;
constexpr auto kSkillIconBorderDistance = 3.f;

class SkillIcon : public sf::Drawable, public Touchable {
 public:
  SkillIcon(const sf::Vector2f& position = {0, 0}, float radius = 10.f,
            float border_thick = 2.f);

  void setMaxValue(int max_value) { max_value_ = max_value; }
  constexpr auto getMaxValue() const { return max_value_; }
  void setCurrentValue(int current_value) { current_value_ = current_value; }
  constexpr auto getCurrentValue() const { return current_value_; }

  void setPosition(const sf::Vector2f& position);

  TOUCHABLE_OBJECT(SkillIcon)
 protected:
  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

 private:
  CircularSector circular_sector_;
  sf::CircleShape background_;
  sf::CircleShape icon_;

  int max_value_;
  int current_value_;

  void updateCircularSector();
};

template <std::size_t MaxSkillNum = 3>
class SkillBar : public sf::Drawable, public Touchable {
 public:
  SkillBar(const sf::Vector2f& position = {0, 0}, float radius = 10.f,
           float border_thick = 2.f, int skill_num = 0);

  constexpr auto getSkillNum() const { return skills_.size(); }

 protected:
  TOUCHABLE_OBJECT(SkillBar)
  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

 private:
  unsigned skill_num_;

  std::array<SkillIcon, MaxSkillNum> skills_;
  std::array<sf::RectangleShape, MaxSkillNum> boxes_;

  sf::Vector2f position_;

  float icon_radius_ = 10.f;
  float border_thick_ = 2.f;

  void updatePosition();
};

template <std::size_t N>
SkillBar<N>::SkillBar(const sf::Vector2f& position, float radius,
                      float border_thick, int skill_num)
```

```
    : skill_num_(skill_num),
      position_(position),
      icon_radius_(radius),
      border_thick_(border_thick) {
  for (auto& skill : skills_) {
    skill.setMaxValue(100);
    skill.setCurrentValue(33);
  }

  for (int i = 0; i != skill_num; ++i) {
    boxes_[i].setFillColor(sf::Color::Black);
    boxes_[i].setOutlineColor(sf::Color::White);
    boxes_[i].setOutlineThickness(2.f);
  }

  updatePosition();
}

template <std::size_t N>
void SkillBar<N>::draw(sf::RenderTarget& target,
                       sf::RenderStates states) const {
  for (const auto& box : boxes_) {
    target.draw(box, states);
  }

  for (const auto& skill : skills_) {
    target.draw(skill, states);
  }
}

template <std::size_t N>
void SkillBar<N>::updatePosition() {
  const float box_length = 2 * icon_radius_ + 2 * kSkillIconBorderDistance;

  for (std::size_t i = 0; i != skills_.size(); ++i) {
    boxes_[i].setPosition(position_.x + i * box_length, position_.y);

    skills_[i].setPosition(
        sf::Vector2f{static_cast<float>(position_.x + (i + 0.5) * box_length),
                     static_cast<float>(position_.y + 0.5 * box_length)});
  }
}

template <std::size_t N>
bool SkillBar<N>::inRange(const sf::Vector2f& point) const {
  auto box_length = 2 * icon_radius_ + 2 * kSkillIconBorderDistance;

  return point.x >= position_.x &&
         point.x <= position_.x + box_length * skill_num_ &&
         point.y >= position_.y &&
         point.y <= position_.y + box_length * skill_num_;
}

}  // namespace yuki

#endif  // SKILL_BAR_HPP
```

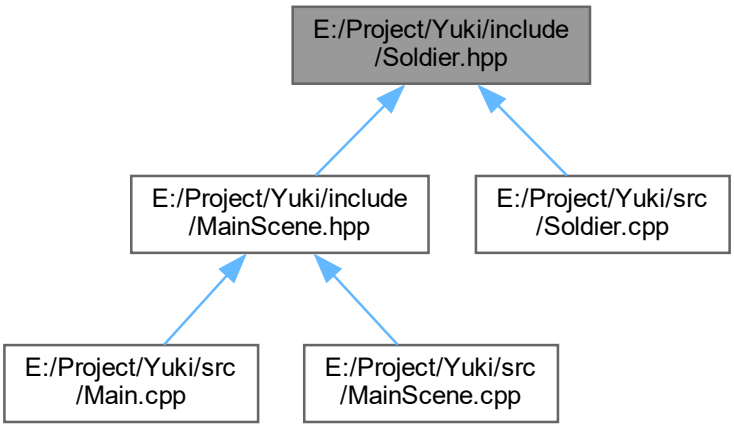
7.37 E:/Project/Yuki/include/Soldier.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <string>
#include "Attack.hpp"
#include "Basic.hpp"
#include "Elementum.hpp"
#include "FloatingBubble.hpp"
#include "InteractiveObject.hpp"
#include "StatusBar.hpp"
#include "YukiScene.hpp"
```

Include dependency graph for Soldier.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::Soldier
- class yuki::NormalSoldier
- class yuki::SeniorSoldier

Namespaces

- namespace yuki

Functions

- std::unique_ptr< NormalSoldier > yuki::getNormalSoldier (Camp camp=Camp::Own)
- std::unique_ptr< Soldier > yuki::getSoldier (const std::string &name, Camp camp=Camp::Own)

7.38 Soldier.hpp

Go to the documentation of this file.

```
#ifndef SOLDIER_HPP
#define SOLDIER_HPP

#include <SFML/Graphics.hpp>
#include <string>

#include "Attack.hpp"
#include "Basic.hpp"
#include "Elementum.hpp"
#include "FloatingBubble.hpp"
#include "InteractiveObject.hpp"
#include "StatusBar.hpp"
#include "YukiScene.hpp"

namespace yuki {
class Soldier : public sf::Drawable, public Focusable {
 public:
  Soldier(Camp camp = Camp::Own)
      : camp_(camp), adhesion_elementum_(Elementum(ElementumType::None, 0.0f)) {
    health_bar_ = getHealthBar(camp);
    setOrigin({sprite_.getLocalBounds().width / 2.f,
               sprite_.getLocalBounds().height});
  }
  Soldier(const sf::Vector2f& position, Camp camp = Camp::Own) : Soldier(camp) {
    setPosition(position);
  }

  virtual ~Soldier() {}

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

  // Focusable object
  FOCUSABLE_OBJECT(Soldier)

  virtual void update();

  // Action
  void setDirection(const Direction direction) { direction_ = direction; }
  constexpr auto getDirection() const { return direction_; }
  void setSpeed(const float speed) { speed_ = speed; }
  constexpr auto getSpeed() const { return speed_; }
  void setMoving(const bool moving) { is_moving_ = moving; }
  constexpr auto isMoving() const { return is_moving_; }
  void setPosition(const sf::Vector2f& position) {
    sprite_.setPosition(position);
    health_bar_.setPosition(
        {position.x + health_bar_offset_.x, position.y + health_bar_offset_.y});
  }
  const auto getPosition() const { return sprite_.getPosition(); }
  void setOrigin(const sf::Vector2f& origin) { sprite_.setOrigin(origin); }
  const auto getOrigin() const { return sprite_.getOrigin(); }

  auto getSize() const {
    return sf::Vector2f(sprite_.getGlobalBounds().width,
                        sprite_.getGlobalBounds().height);
  }

  void setMaxHealth(const float max_health) {
    health_bar_.setMaxValue(max_health);
  }
  constexpr auto getMaxHealth() const { return health_bar_.getMaxValue(); }
  void setHealth(const float current_health) {
    health_bar_.setValue(current_health);
  }
  const auto getHealth() const { return health_bar_.getValue(); }
  void increaseHealth(const float health) { health_bar_.increase(health); }
  void decreaseHealth(const float health) { health_bar_.decrease(health); }

  void setColor(const sf::Color& color) { sprite_.setColor(color); }
  auto getColor() const { return sprite_.getColor(); }

  void setPlaceOffset(const sf::Vector2f& offset) { place_offset_ = offset; }
  virtual const sf::Vector2f& getPlaceOffset() const { return place_offset_; }

  void setFloatingBubbleVisible(const bool visible) {
    floating_bubble_visible_ = visible;
  }
  constexpr auto isFloatingBubbleVisible() const {
    return floating_bubble_visible_;
  }
  void walk();
```

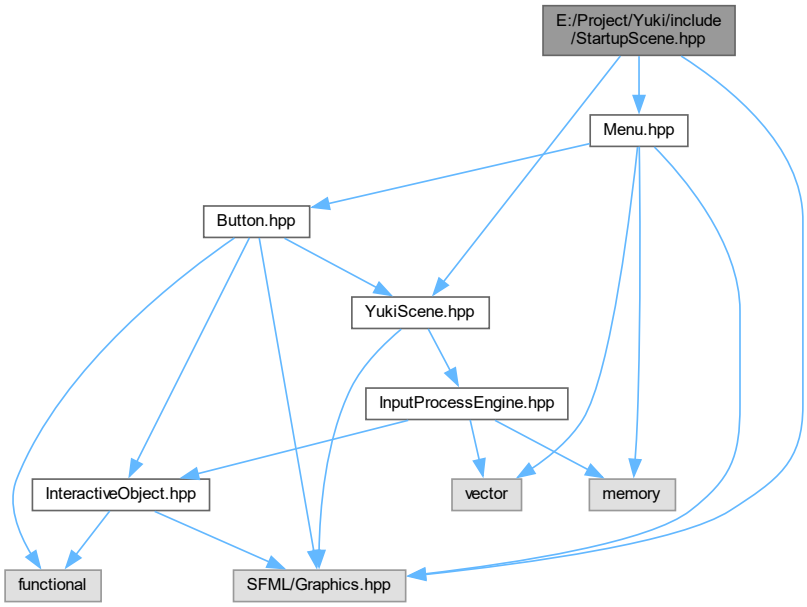
```

  auto& getFloatingBubble() { return floating_bubble_; }
  const auto& getFloatingBubble() const { return floating_bubble_; }

  void setElementum(const Elementum& elementum) { elementum_ = elementum; }
  const auto& getElementum() const { return elementum_; }
  void setElementumType(const ElementumType& elementum_type) {
    elementum_.type = elementum_type;
  }
  constexpr auto getElementumType() const { return elementum_.type; }

  constexpr auto getCamp() const { return camp_; }

  void setAdhesionElementum(const Elementum& elementum) {
    adhesion_elementum_ = elementum;
  }
  const auto& getAdhesionElementum() const { return adhesion_elementum_; }

  void getAttacked(const BasicAttack& attack);

  constexpr auto isFreezed() const { return is_freeze_; }

  // constexpr auto getCost() const { return cost_; }

  virtual std::unique_ptr<BasicAttack> getDefaultAttack() = 0;

 protected:
  // sf::Sprite sprite_;
  sf::Sprite sprite_;
  sf::Vector2f health_bar_offset_ = {0, -15};
  Camp camp_;

  sf::Vector2f place_offset_ = {0, 0};

  bool is_moving_ = false;
  Direction direction_ = Direction::Down;
  float speed_ = 0.0f;
  Elementum elementum_ = Elementum(yuki::ElementumType::None, 0.0f);

  Elementum adhesion_elementum_ = Elementum(yuki::ElementumType::None, 0.0f);
  StatusBar health_bar_;
  sf::Clock freeze_clock_;
  bool is_freeze_ = false;

  FloatingBubble floating_bubble_;
  bool floating_bubble_visible_;

  sf::Clock clock_;
};

class NormalSoldier : public Soldier {
 public:
  NormalSoldier(Camp camp = Camp::Own);
  NormalSoldier(const sf::Vector2f& position, Camp camp = Camp::Own);

  virtual ~NormalSoldier() {}

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

  virtual void update() override;

  std::unique_ptr<BasicAttack> getDefaultAttack() override { return getDefaultGranuleAttack(); }
  std::unique_ptr<GranuleAttack> getDefaultGranuleAttack();

  void setGranuleAttackCD(const sf::Time& cd) { granule_attack_cd_ = cd; }
  const auto getGranuleAttackCD() const { return granule_attack_cd_; }
  bool canGranuleAttack() const {
    return granule_attack_clock_.getElapsedTime() >= granule_attack_cd_;
  }
  void resetGranuleAttackClock() { granule_attack_clock_.restart(); }

 private:
  sf::Texture all_textures_;

  sf::Vector2f place_offset_ = {0, 0};

  int current_frame_;
  int previous_frame_;
  int max_frame_;

  sf::Time granule_attack_cd_ = sf::seconds(0.5f);
  sf::Clock granule_attack_clock_;

  // FloatingBubble floating_bubble_;
};

```

```
class SeniorSoldier : public Soldier {
 public:
  SeniorSoldier(Camp camp = Camp::Own);
  SeniorSoldier(const sf::Vector2f& position, Camp camp = Camp::Own);

  virtual ~SeniorSoldier() {}

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

  virtual void update() override;

 private:
  sf::Texture all_textures_;
  sf::Vector2f place_offset_ = {0, 0};

  int current_frame_;
  int previous_frame_;
  int max_frame_;

  FloatingBubble floating_bubble_;
};
std::unique_ptr<NormalSoldier> getNormalSoldier(Camp camp = Camp::Own);
std::unique_ptr<Soldier> getSoldier(const std::string& name,
                                    Camp camp = Camp::Own);

}  // namespace yuki

#endif
```

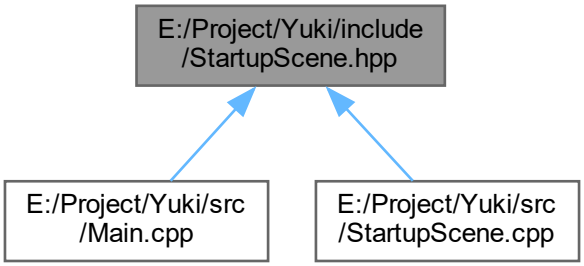
7.39 E:/Project/Yuki/include/StartupScene.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Menu.hpp"
#include "YukiScene.hpp"
```

Include dependency graph for StartupScene.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::StartupScene

Namespaces

- namespace yuki

7.40 StartupScene.hpp

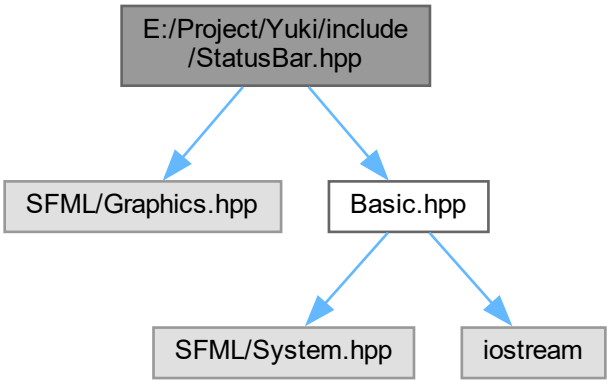
Go to the documentation of this file.

```
#ifndef STARTUP_SCENE_HPP
#define STARTUP_SCENE_HPP

#include <SFML/Graphics.hpp>

#include "Menu.hpp"
#include "YukiScene.hpp"

namespace yuki {

class StartupScene : public YukiScene {
 YUKI_SCENE
 public:
  StartupScene(sf::RenderWindow& window);
  virtual ~StartupScene() {}

 private:
  Button btn_start_;
  Button btn_connect_;
  sf::Sprite background_;
  sf::Texture background_texture_;

  void initUi();
};

}  // namespace yuki

#endif  // STARTUP_SCENE_HPP
```

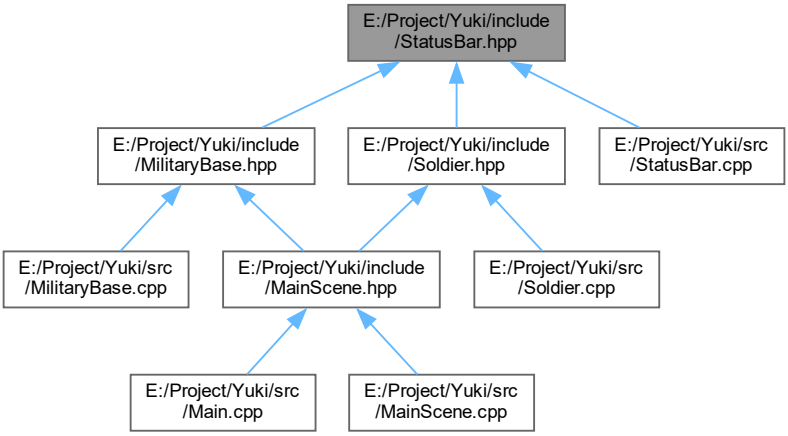
7.41 E:/Project/Yuki/include/StatusBar.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Basic.hpp"
```

Include dependency graph for StatusBar.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::StatusBar

Namespaces

- namespace yuki

Functions

- StatusBar yuki::getHealthBar (Camp camp=Camp::Own, const sf::Vector2f &size={30.f, 5.f})

7.42 StatusBar.hpp

Go to the documentation of this file.

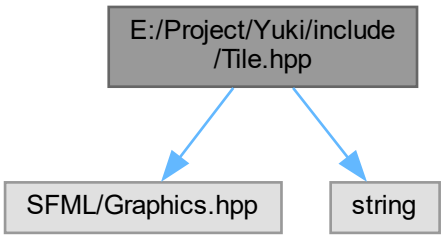
```
#ifndef STATUS_BAR_HPP
#define STATUS_BAR_HPP

#include <SFML/Graphics.hpp>

#include "Basic.hpp"

namespace yuki {

class StatusBar : public sf::Drawable {
 public:
  StatusBar() = default;
  StatusBar(sf::Vector2f size, sf::Vector2f position,
            sf::Color background_color = sf::Color::Black,
            sf::Color filled_color = sf::Color::Blue, float max_value = 100.f,
            float current_value = 100.f)
      : size_(size),
        position_(position),
        max_value_(max_value),
        current_value_(current_value) {
    background_.setPosition(position_);
    filled_.setPosition(position_);
    background_.setSize(size_);
    background_.setFillColor(background_color);
    filled_.setFillColor(filled_color);

    visible_ = true;

    updateFilledPercentage();
  }

  void show() { visible_ = true; }
  void hide() { visible_ = false; }

  auto isVisible() const { return visible_; }

  void setBackgroudColor(const sf::Color color) {
    background_.setFillColor(color);
  }
  void setFilledColor(const sf::Color color) { filled_.setFillColor(color); }
  void setMaxValue(const float max_value) { max_value_ = max_value; }
  void setValue(const float current_value) {
    current_value_ = current_value;
    updateFilledPercentage();
  }
  void setPosition(const sf::Vector2f& position);

  void setSize(const sf::Vector2f& size) {
    size_ = size;
    background_.setSize(size);
    updateFilledPercentage();
  }

  void setOutlineThickness(const float thickness) {
    background_.setOutlineThickness(thickness);
  }

  void setOutlineColor(const sf::Color& color) {
    background_.setOutlineColor(color);
  }

  void setOutlineStyle(const sf::Color& color, const float thickness) {
    background_.setOutlineColor(color);
    background_.setOutlineThickness(thickness);
  }

  const auto getBackgroundColor() const { return background_.getFillColor(); }
  const auto getFilledColor() const { return filled_.getFillColor(); }
  constexpr auto getMaxValue() const { return max_value_; }
  constexpr auto getValue() const { return current_value_; }
  const auto getPosition() const { return position_; }

  void increase(const float value);
```

```
  void decrease(const float value);

  virtual ~StatusBar() {}

  virtual void draw(sf::RenderTarget& target,
                    sf::RenderStates states) const override;

  void updateFilledPercentage() {
    setFilledPercentage(current_value_ / max_value_);
  }

 protected:
  sf::RectangleShape background_;
  sf::RectangleShape filled_;

  sf::Vector2f size_;
  sf::Vector2f position_;

  float max_value_;
  float current_value_;

  bool visible_;

  void setFilledPercentage(float percentage);
  void adjustIfOverflow();
};

StatusBar getHealthBar(Camp camp = Camp::Own,
                       const sf::Vector2f& size = {30.f, 5.f});

}  // namespace yuki

#endif  // STATUS_BAR_HPP
```

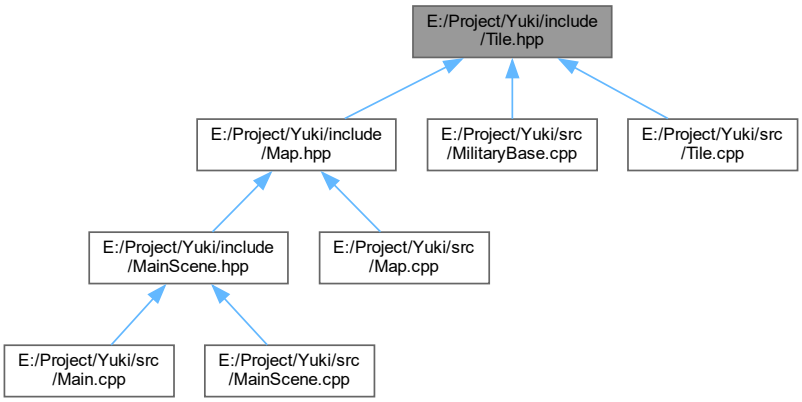
7.43 E:/Project/Yuki/include/Tile.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <string>
```

Include dependency graph for Tile.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct yuki::TileInfo
- class yuki::Tile

Namespaces

- namespace yuki

Enumerations

- enum class yuki::TileCategory {
  yuki::Building , yuki::Decor , yuki::Greenery , yuki::Land ,
  yuki::River , yuki::Road , yuki::Stones , yuki::Tree }

Functions

- std::string yuki::getTileCategoryName (const TileCategory category)
- std::string yuki::getTileSrcName (const TileInfo &tile_info)
- sf::Vector2f yuki::getTileScaleFactor (const TileInfo &tile_info)

7.44 Tile.hpp

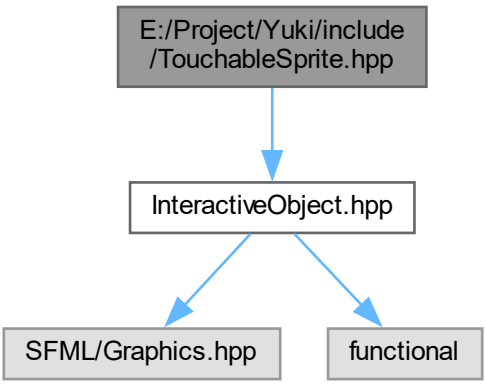
Go to the documentation of this file.

```
#ifndef TILE_HPP
#define TILE_HPP

#include <SFML/Graphics.hpp>
#include <string>

namespace yuki {

enum class TileCategory {
  Building,
  Decor,
  Greenery,
  Land,
  River,
  Road,
  Stones,
  Tree
};

struct TileInfo {
  TileCategory category;
  int index;

  TileInfo() = default;
  TileInfo(TileCategory category, int index)
      : category(category), index(index) {}
};

std::string getTileCategoryName(const TileCategory category);
std::string getTileSrcName(const TileInfo& tile_info);
sf::Vector2f getTileScaleFactor(const TileInfo& tile_info);

class Tile : public sf::Sprite {
 public:
  auto getTileInfo() const { return tile_info_; }
  void setTileInfo(TileInfo&& tile_info) { tile_info_ = std::move(tile_info); }
  void setTileInfo(const TileInfo& tile_info) { tile_info_ = tile_info; }

  void setTextureSrc(const std::string& src_location) {
    texture_.loadFromFile(src_location);
    this->setTexture(texture_);
  }

 private:
  TileInfo tile_info_;
  sf::Texture texture_;
};

} // namespace yuki

#endif // TILE_HPP
```

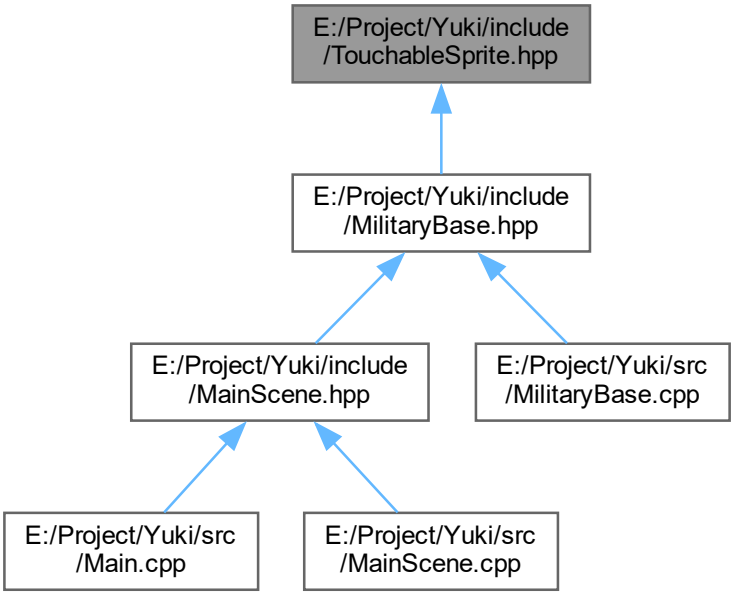
7.45 E:/Project/Yuki/include/TouchableSprite.hpp File Reference

```
#include "InteractiveObject.hpp"
```

Include dependency graph for TouchableSprite.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::TouchableSprite

Namespaces

- namespace yuki

7.46 TouchableSprite.hpp

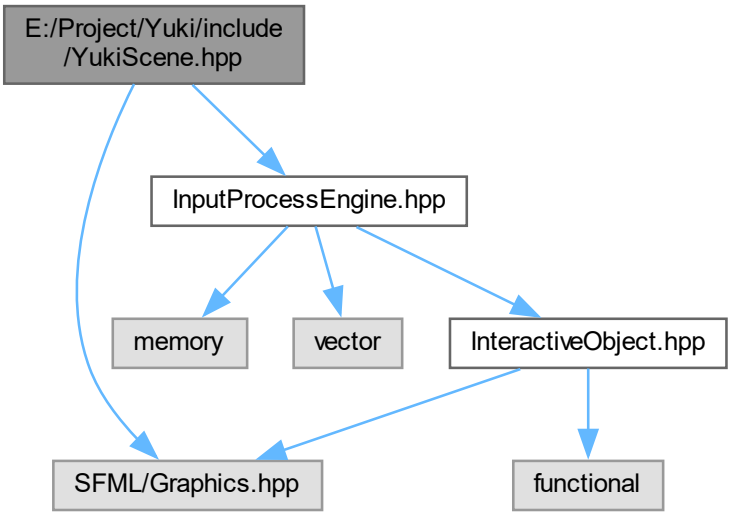
Go to the documentation of this file.

```
#ifndef TOUCHABLE_SPRITE_HPP
#define TOUCHABLE_SPRITE_HPP

#include "InteractiveObject.hpp"

namespace yuki {

class TouchableSprite : public sf::Sprite, public yuki::Touchable {
  // TOUCHABLE_OBJECT(TouchableSprite)

 public:
  TouchableSprite() = default;
  TouchableSprite(const sf::Texture& texture) : sf::Sprite(texture) {}
  TouchableSprite(const sf::Texture& texture, const sf::IntRect& rect)
      : sf::Sprite(texture, rect) {}

  virtual ~TouchableSprite() {}

private:
  virtual bool inRange(const sf::Vector2f& position) const override {
    return getGlobalBounds().contains(position);
  }
};

} // namespace yuki

#endif
```

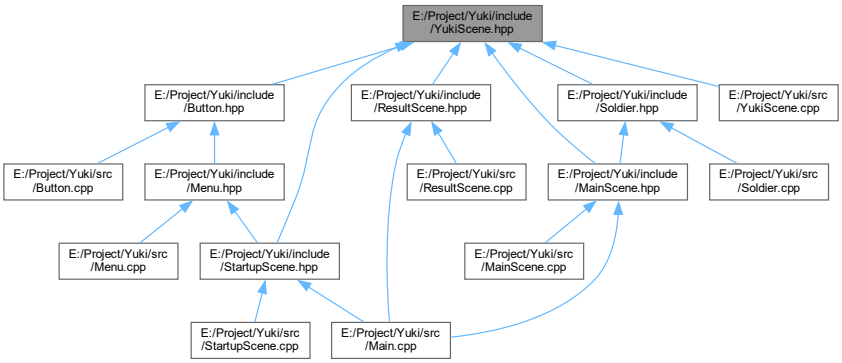
7.47 E:/Project/Yuki/include/YukiScene.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "InputProcessEngine.hpp"
```

Include dependency graph for YukiScene.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class yuki::YukiScene

Namespaces

- namespace yuki

Macros

- #define YUKI_SCENE

7.47.1 Macro Definition Documentation

7.47.1.1 YUKI_SCENE

```
#define YUKI_SCENE
```

Value:

```
public:                                          \
 int show() override;                            \
                                                 \
protected:                                       \
 void processEvent(sf::Event event) override;    \
 void draw() const override;                     \
 void updateInfo() override;                     \
                                                 \
public:
```

7.48 YukiScene.hpp

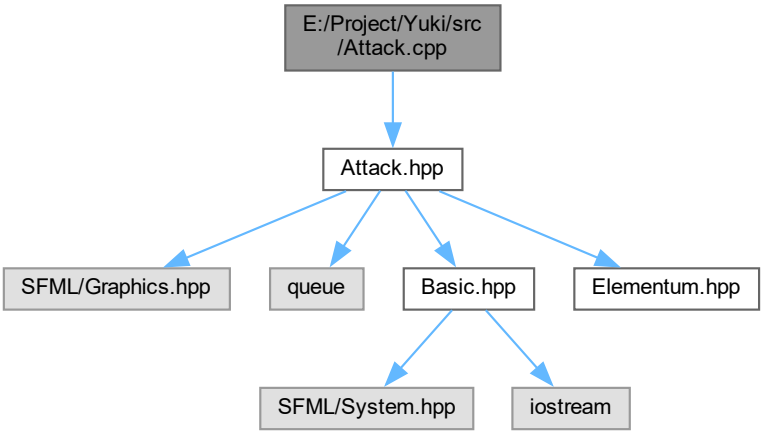
Go to the documentation of this file.

```
#ifndef YUKI_SCENE_HPP
#define YUKI_SCENE_HPP

#include <SFML/Graphics.hpp>

#include "InputProcessEngine.hpp"

namespace yuki {

#define YUKI_SCENE                                  \
 public:                                            \
  int show() override;                              \
                                                    \
 protected:                                         \
  void processEvent(sf::Event event) override;      \
  void draw() const override;                       \
  void updateInfo() override;                       \
                                                    \
 public:

// Abstract class for all scenes
class YukiScene {
 public:
  enum class Notify { None, End };

  YukiScene(sf::RenderWindow& window) : window_(window) {}
  virtual ~YukiScene();

  virtual int show();

 protected:
  yuki::MouseEventProcessEngine mouse_event_engine_;
  Notify notify_;
  int ret_code_;

  void registerTouchableObject(Touchable* object) {
    mouse_event_engine_.registerTouchableObject(object);
  }
  void unregisterTouchableObject(Touchable* object) {
    mouse_event_engine_.unregisterTouchableObject(object);
  }

  void setReturnCode(int ret_code) { ret_code_ = ret_code; }

  sf::RenderWindow& window_;
  virtual void processEvent(sf::Event event);
  virtual void draw() const = 0;
  virtual void updateInfo() = 0;
};

}  // namespace yuki

#endif  // YUKI_SCENE_HPP
```

7.49 E:/Project/Yuki/src/Attack.cpp File Reference

```
#include "Attack.hpp"
```

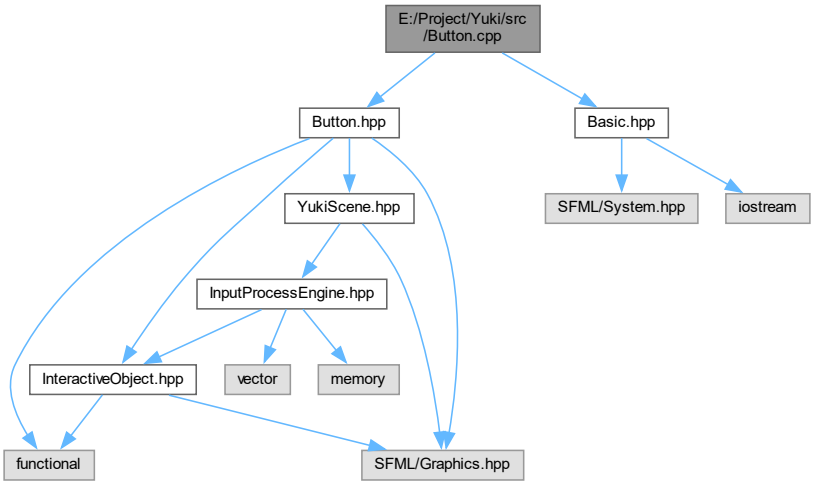
Include dependency graph for Attack.cpp:



7.50 E:/Project/Yuki/src/Button.cpp File Reference

```
#include "Button.hpp"
#include "Basic.hpp"
```

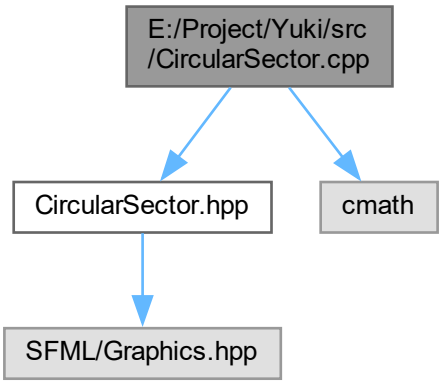
Include dependency graph for Button.cpp:



7.51 E:/Project/Yuki/src/CircularSector.cpp File Reference

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

Include dependency graph for CircularSector.cpp:



Variables

- constexpr auto kPi = 3.14159265358979323846f

7.51.1 Variable Documentation

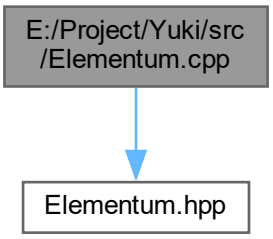
7.51.1.1 kPi

```
constexpr auto kPi = 3.14159265358979323846f  [constexpr]
```

7.52 E:/Project/Yuki/src/Elementum.cpp File Reference

```
#include "Elementum.hpp"
```

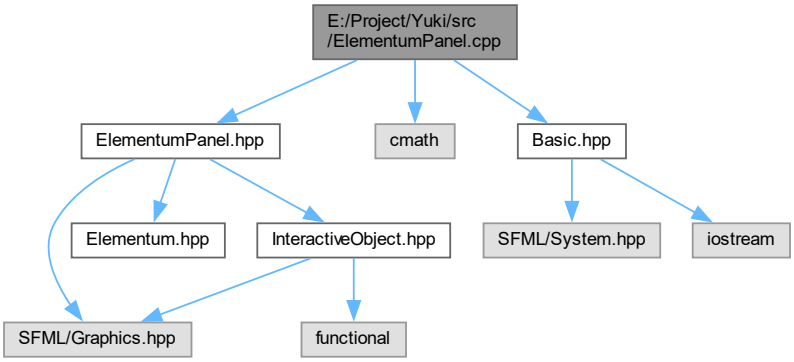
Include dependency graph for Elementum.cpp:



7.53 E:/Project/Yuki/src/ElementumPanel.cpp File Reference

```
#include "ElementumPanel.hpp"
#include <cmath>
#include "Basic.hpp"
```

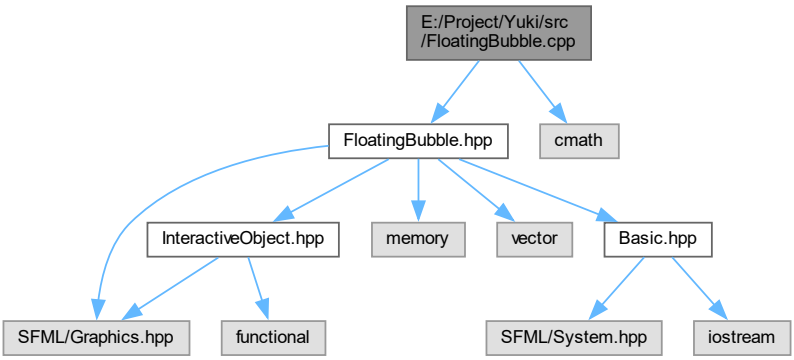
Include dependency graph for ElementumPanel.cpp:



7.54 E:/Project/Yuki/src/FloatingBubble.cpp File Reference

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

Include dependency graph for FloatingBubble.cpp:



Variables

- constexpr float kPi = 3.14159265358979323846f

7.54.1 Variable Documentation

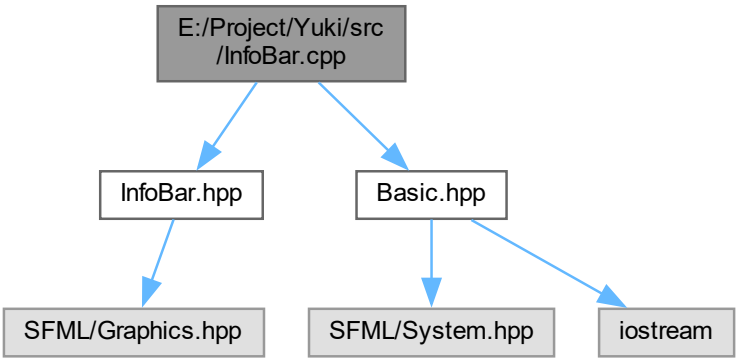
7.54.1.1 kPi

```
constexpr float kPi = 3.14159265358979323846f  [constexpr]
```

7.55 E:/Project/Yuki/src/InfoBar.cpp File Reference

```
#include "InfoBar.hpp"
#include "Basic.hpp"
```

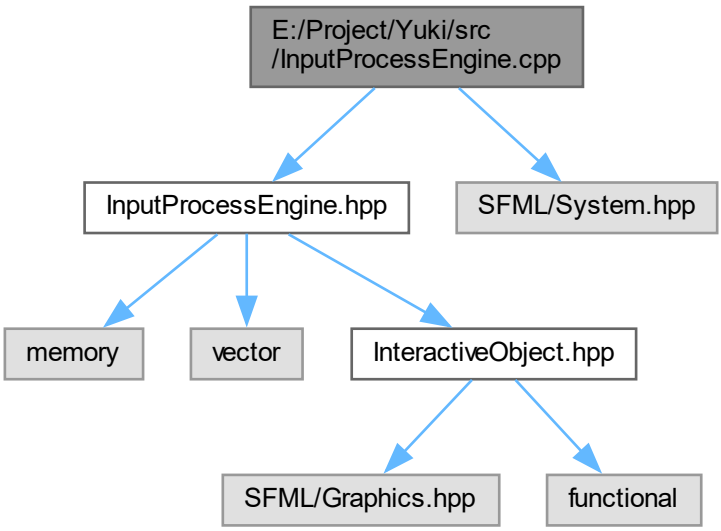
Include dependency graph for InfoBar.cpp:



7.56 E:/Project/Yuki/src/InputProcessEngine.cpp File Reference

```
#include "InputProcessEngine.hpp"
#include <SFML/System.hpp>
```

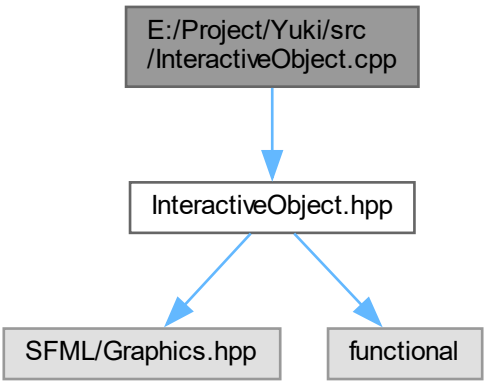
Include dependency graph for InputProcessEngine.cpp:



7.57 E:/Project/Yuki/src/InteractiveObject.cpp File Reference

```
#include "InteractiveObject.hpp"
```

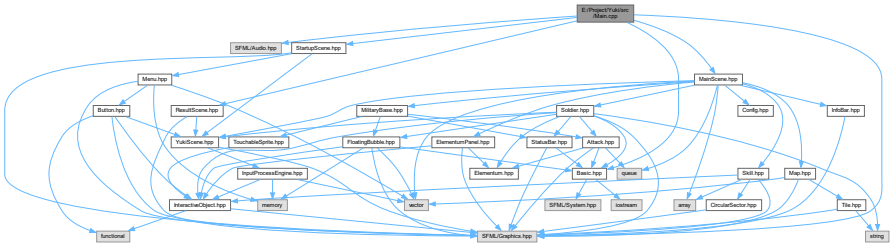
Include dependency graph for InteractiveObject.cpp:



7.58 E:/Project/Yuki/src/Main.cpp File Reference

```
#include <SFML/Audio.hpp>
#include <SFML/Graphics.hpp>
#include "Basic.hpp"
#include "MainScene.hpp"
#include "StartupScene.hpp"
#include "ResultScene.hpp"
```

Include dependency graph for Main.cpp:



Functions

- int main ()

7.58.1 Function Documentation

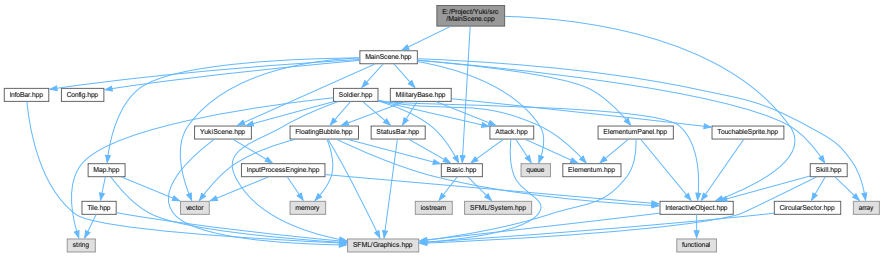
7.58.1.1 main()

```
int main ( )
```

7.59 E:/Project/Yuki/src/MainScene.cpp File Reference

```
#include "MainScene.hpp"
#include "Basic.hpp"
#include "InteractiveObject.hpp"
```

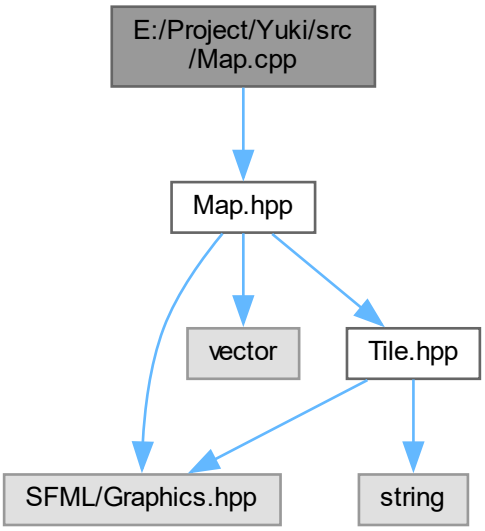
Include dependency graph for MainScene.cpp:



7.60 E:/Project/Yuki/src/Map.cpp File Reference

```
#include "Map.hpp"
```

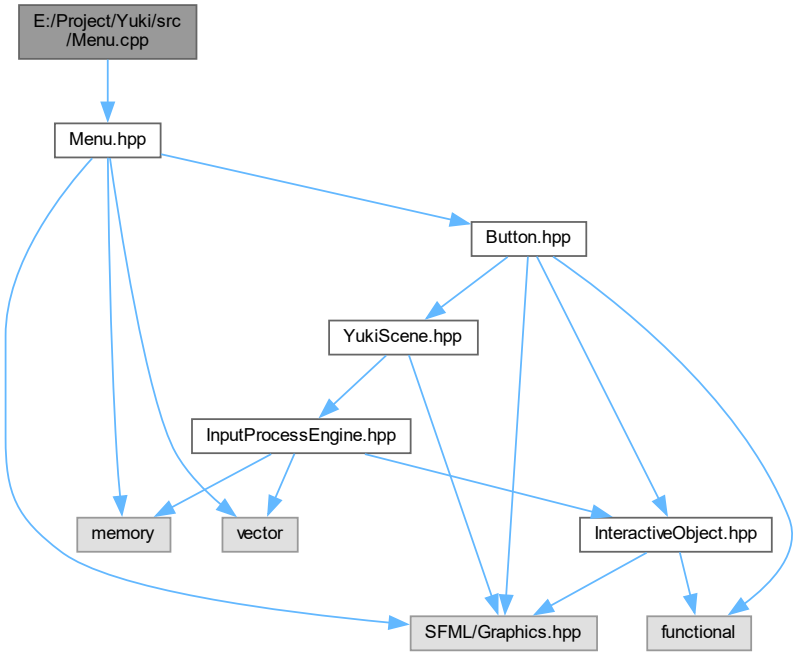
Include dependency graph for Map.cpp:



7.61 E:/Project/Yuki/src/Menu.cpp File Reference

```
#include "Menu.hpp"
```

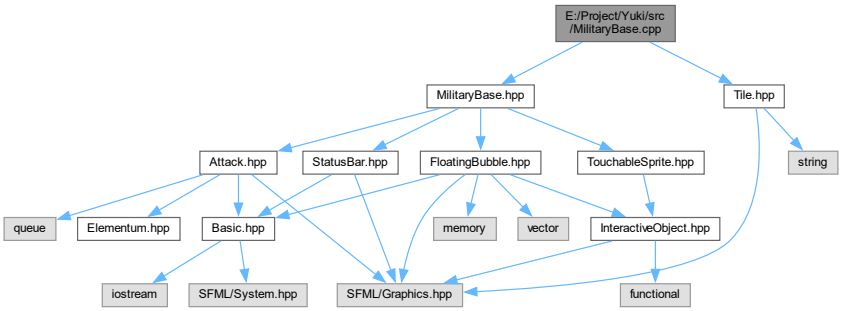
Include dependency graph for Menu.cpp:



7.62 E:/Project/Yuki/src/MilitaryBase.cpp File Reference

```
#include "MilitaryBase.hpp"
#include "Tile.hpp"
```

Include dependency graph for MilitaryBase.cpp:

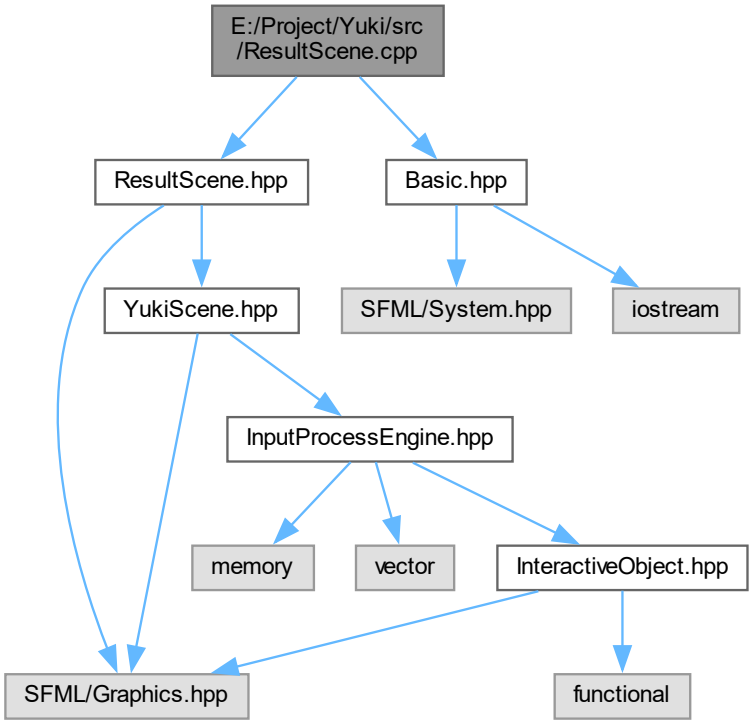


7.63 E:/Project/Yuki/src/Network.cpp File Reference

7.64 E:/Project/Yuki/src/ResultScene.cpp File Reference

```
#include "ResultScene.hpp"
#include "Basic.hpp"
```

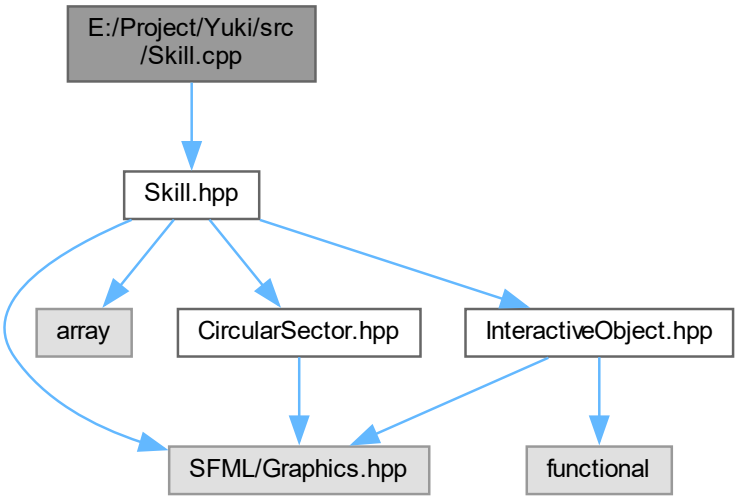
Include dependency graph for ResultScene.cpp:



7.65 E:/Project/Yuki/src/Skill.cpp File Reference

```
#include "Skill.hpp"
```

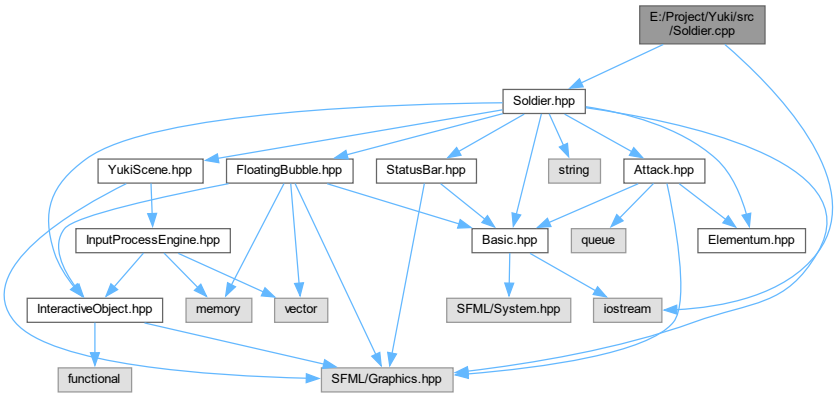
Include dependency graph for Skill.cpp:



7.66 E:/Project/Yuki/src/Soldier.cpp File Reference

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

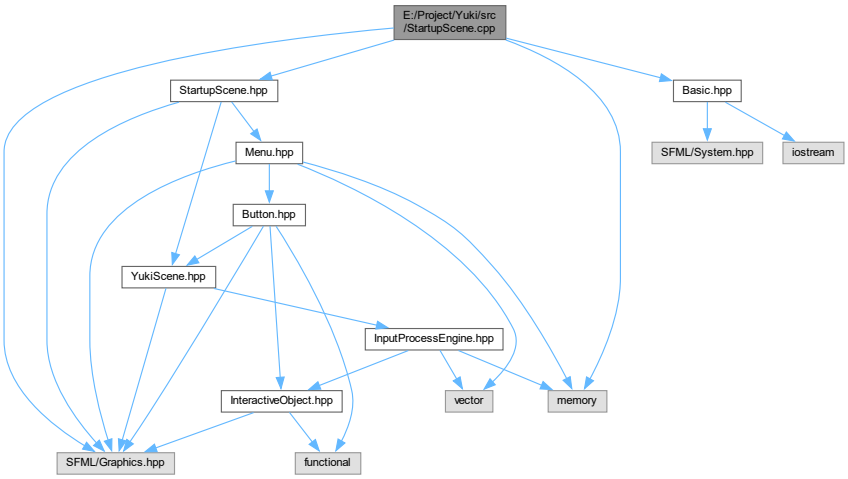
Include dependency graph for Soldier.cpp:



7.67 E:/Project/Yuki/src/StartupScene.cpp File Reference

```
#include "StartupScene.hpp"
#include <SFML/Graphics.hpp>
#include <memory>
#include "Basic.hpp"
```

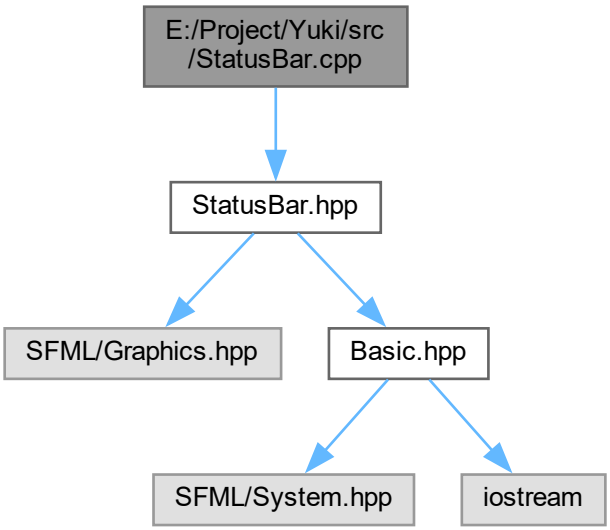
Include dependency graph for StartupScene.cpp:



7.68 E:/Project/Yuki/src/StatusBar.cpp File Reference

```
#include "StatusBar.hpp"
```

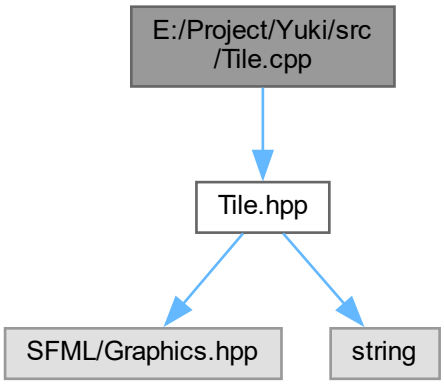
Include dependency graph for StatusBar.cpp:



7.69 E:/Project/Yuki/src/Tile.cpp File Reference

```
#include "Tile.hpp"
```

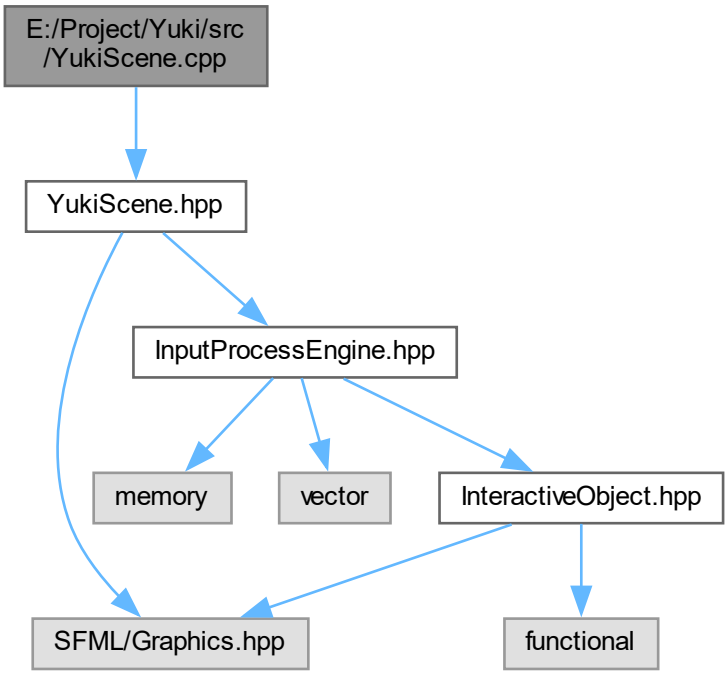
Include dependency graph for Tile.cpp:



7.70 E:/Project/Yuki/src/YukiScene.cpp File Reference

```
#include "YukiScene.hpp"
```

Include dependency graph for YukiScene.cpp:



Index

~BasicAttack
yuki::BasicAttack, 23
~BubbleItem
yuki::BubbleItem, 27
~Button
yuki::Button, 31
~Focusable
yuki::Focusable, 55
~GranuleAttack
yuki::GranuleAttack, 60
~InfoBar
yuki::InfoBar, 63
~MainScene
yuki::MainScene, 70
~Map
yuki::Map, 89
~Menu
yuki::Menu, 96
~MilitaryBase
yuki::MilitaryBase, 100
~MouseEventProcessEngine
yuki::MouseEventProcessEngine, 108
~NormalSoldier
yuki::NormalSoldier, 113
~ResultScene
yuki::ResultScene, 119
~SeniorSoldier
yuki::SeniorSoldier, 124
~Soldier
yuki::Soldier, 138
~StartupScene
yuki::StartupScene, 153
~StatusBar
yuki::StatusBar, 157
~Touchable
yuki::Touchable, 172
~TouchableObject
yuki::TouchableObject< Decorated >, 179
~TouchableSprite
yuki::TouchableSprite, 183
~YukiScene
yuki::YukiScene, 185

Action
yuki::Touchable, 171
activate
yuki::ElementumPanel, 44
addItem
yuki::Menu, 96
adhesion_elementum_
yuki::Soldier, 149
adjustIfOverflow
yuki::StatusBar, 157
adjustItemsPosition
yuki::Menu, 96
adjustTextPosition
yuki::Button, 31
Alignment
yuki::Button, 30
alignment_
yuki::Button, 37
all_textures_
yuki::NormalSoldier, 117
yuki::SeniorSoldier, 125
attack_button_
yuki::MainScene, 85
attack_button_texture_
yuki::MainScene, 85
AttackInfo
yuki::AttackInfo, 21
autoAdjustTextSize
yuki::Button, 31

background_
yuki::InfoBar, 66
yuki::ResultScene, 120
yuki::SkillIcon, 134
yuki::StartupScene, 154
yuki::StatusBar, 166
background_texture_
yuki::InfoBar, 66
yuki::ResultScene, 120
yuki::StartupScene, 155
BasicAttack
yuki::BasicAttack, 23
bg_texture_
yuki::Map, 93
bg_tiles_
yuki::Map, 93
bindAction
yuki::Touchable, 172
bindClick
yuki::Touchable, 172
bindFocus
yuki::Focusable, 55
bindHover
yuki::Touchable, 172
bindLeave
yuki::Touchable, 173
bindRelease

yuki::Touchable, 173
bindUnfocus
yuki::Focusable, 55
border_thick_
yuki::SkillBar< MaxSkillNum >, 129
boxes_
yuki::SkillBar< MaxSkillNum >, 129
btn_connect_
yuki::StartupScene, 155
btn_start_
yuki::StartupScene, 155
bubble_num_
yuki::FloatingBubble, 52
bubble_radius_
yuki::FloatingBubble, 52
BubbleItem
yuki::BubbleItem, 26
bubbles_
yuki::FloatingBubble, 52
Building
yuki, 11
Button
yuki::Button, 30, 31

Callback
yuki::Focusable, 55
yuki::Touchable, 171
Camp
yuki, 10
camp_
yuki::MilitaryBase, 106
yuki::Soldier, 149
canGenerateSoldier
yuki::MilitaryBase, 101
canGranuleAttack
yuki::NormalSoldier, 114
category
yuki::TileInfo, 169
Center
yuki::Button, 30
center_
yuki::CircularSector, 40
circular_sector_
yuki::SkillIcon, 134
CircularSector
yuki::CircularSector, 39
CircularSector.cpp
kPi, 242
Click
yuki::Touchable, 171
clock_
yuki::Soldier, 149
Config.hpp
YUKI_DEBUG, 200
coordinateToPixel
yuki::MainScene, 71
current_frame_
yuki::NormalSoldier, 117
yuki::SeniorSoldier, 125
current_value_
yuki::SkillIcon, 134
yuki::StatusBar, 166
Cyro
yuki, 11

damage
yuki::AttackInfo, 21
deactivate
yuki::ElementumPanel, 44
debug_text_
yuki::MainScene, 85
Decor
yuki, 12
decrease
yuki::StatusBar, 158
decreaseEnemyMoney
yuki::MainScene, 71
decreaseHealth
yuki::Soldier, 139
decreaseMoney
yuki::MainScene, 72
decreaseOwnMoney
yuki::MainScene, 72
DefaultFont
yuki::GameFont, 18
Direction
yuki, 10
direction_
yuki::FloatingBubble, 52
yuki::GranuleAttack, 61
yuki::Soldier, 149
distribute_radius_
yuki::FloatingBubble, 52
Down
yuki, 11
draw
yuki::Button, 31
yuki::FloatingBubble, 48
yuki::GranuleAttack, 60
yuki::InfoBar, 64
yuki::Map, 89
yuki::Menu, 96
yuki::MilitaryBase, 101
yuki::NormalSoldier, 114
yuki::SeniorSoldier, 125
yuki::SkillBar< MaxSkillNum >, 128
yuki::SkillIcon, 132
yuki::Soldier, 139
yuki::StatusBar, 158
yuki::YukiScene, 185

E:/Project/Yuki/include/Attack.hpp, 191, 192
E:/Project/Yuki/include/Basic.hpp, 193, 195
E:/Project/Yuki/include/Button.hpp, 195, 197
E:/Project/Yuki/include/CircularSector.hpp, 198, 199
E:/Project/Yuki/include/Config.hpp, 200, 201
E:/Project/Yuki/include/Elementum.hpp, 201, 202
E:/Project/Yuki/include/ElementumPanel.hpp, 202, 204

E:/Project/Yuki/include/FloatingBubble.hpp, 204, 205
E:/Project/Yuki/include/InfoBar.hpp, 207, 208
E:/Project/Yuki/include/InputProcessEngine.hpp, 208, 210
E:/Project/Yuki/include/InteractiveObject.hpp, 210, 212
E:/Project/Yuki/include/MainScene.hpp, 213, 214
E:/Project/Yuki/include/Map.hpp, 215, 217
E:/Project/Yuki/include/Menu.hpp, 217, 219
E:/Project/Yuki/include/MilitaryBase.hpp, 219, 220
E:/Project/Yuki/include/Network.hpp, 222
E:/Project/Yuki/include/ResultScene.hpp, 222, 223
E:/Project/Yuki/include/Skill.hpp, 223, 225
E:/Project/Yuki/include/Soldier.hpp, 226, 228
E:/Project/Yuki/include/StartupScene.hpp, 230, 231
E:/Project/Yuki/include/StatusBar.hpp, 232, 233
E:/Project/Yuki/include/Tile.hpp, 234, 236
E:/Project/Yuki/include/TouchableSprite.hpp, 237, 238
E:/Project/Yuki/include/YukiScene.hpp, 238, 240
E:/Project/Yuki/src/Attack.cpp, 241
E:/Project/Yuki/src/Button.cpp, 241
E:/Project/Yuki/src/CircularSector.cpp, 242
E:/Project/Yuki/src/Elementum.cpp, 243
E:/Project/Yuki/src/ElementumPanel.cpp, 243
E:/Project/Yuki/src/FloatingBubble.cpp, 243
E:/Project/Yuki/src/InfoBar.cpp, 244
E:/Project/Yuki/src/InputProcessEngine.cpp, 245
E:/Project/Yuki/src/InteractiveObject.cpp, 245
E:/Project/Yuki/src/Main.cpp, 246
E:/Project/Yuki/src/MainScene.cpp, 246
E:/Project/Yuki/src/Map.cpp, 247
E:/Project/Yuki/src/Menu.cpp, 248
E:/Project/Yuki/src/MilitaryBase.cpp, 248
E:/Project/Yuki/src/Network.cpp, 249
E:/Project/Yuki/src/ResultScene.cpp, 249
E:/Project/Yuki/src/Skill.cpp, 250
E:/Project/Yuki/src/Soldier.cpp, 250
E:/Project/Yuki/src/StartupScene.cpp, 251
E:/Project/Yuki/src/StatusBar.cpp, 251
E:/Project/Yuki/src/Tile.cpp, 252
E:/Project/Yuki/src/YukiScene.cpp, 253
Elementum
    yuki::Elementum, 41
elementum
    yuki::ElementumReaction, 46
elementum_
    yuki::Soldier, 149
elementum_level
    yuki::AttackInfo, 22
elementum_panels_
    yuki::MainScene, 85
elementum_type
    yuki::AttackInfo, 22
elementum_type_
    yuki::ElementumPanel, 45
ElementumPanel
    yuki::ElementumPanel, 44
ElementumReactionType
    yuki, 11
ElementumType
    yuki, 11
End
    yuki::YukiScene, 185
end_angle_
    yuki::CircularSector, 41
enemies_
    yuki::MainScene, 85
Enemy
    yuki, 10
enemy_base_
    yuki::MainScene, 86
enemy_current_elementum_type_
    yuki::MainScene, 86
enemy_money_
    yuki::MainScene, 86
EnemyBase
    yuki::MainScene, 70
enemyBaseMakeDecision
    yuki::MainScene, 73
EnemySoldier
    yuki::MainScene, 70
EnemyWin
    yuki::MainScene, 70
eraseDeadSoldier
    yuki::MainScene, 73
Evaporation
    yuki, 11

filled_
    yuki::StatusBar, 166
floating_bubble_
    yuki::MilitaryBase, 106
    yuki::SeniorSoldier, 125
    yuki::Soldier, 149
floating_bubble_visible_
    yuki::MilitaryBase, 106
    yuki::Soldier, 150
FloatingBubble
    yuki::BubbleItem, 27
    yuki::FloatingBubble, 48
FloatingBubble.cpp
    kPi, 244
Focusable
    yuki::Focusable, 55
FOCUSABLE_OBJECT
    InteractiveObject.hpp, 211
focused_object_
    yuki::MainScene, 86
focused_object_type_
    yuki::MainScene, 86
FocusedObjectChanged
    yuki::MainScene, 70
Freeze
    yuki, 11
freeze_clock_
    yuki::Soldier, 150

GenerateEnemySoldier

yuki::MainScene, 70
GenerateOwnSoldier
yuki::MainScene, 70
generateSoldier
yuki::MainScene, 74
generateSoldierClock_
yuki::MilitaryBase, 107
getAdhesionElementum
yuki::Soldier, 139
getAttacked
yuki::MilitaryBase, 101
yuki::Soldier, 140
getAttackInfo
yuki::BasicAttack, 23
getBackgroundColor
yuki::StatusBar, 158
getBubbleItems
yuki::FloatingBubble, 49
getBubblePosition
yuki::FloatingBubble, 49
getBubbleRadius
yuki::FloatingBubble, 49
getByName
yuki::Menu, 96
getCamp
yuki::Soldier, 140
getCenter
yuki::CircularSector, 39
getColor
yuki::Soldier, 140
getCoordinate
yuki::Map, 90
getCurrentHealth
yuki::MilitaryBase, 101
getCurrentValue
yuki::SkillIcon, 132
getDamage
yuki::BasicAttack, 24
getDefaultAttack
yuki::NormalSoldier, 114
yuki::Soldier, 140
getDefaultGranuleAttack
yuki::NormalSoldier, 115
getDirection
yuki::Soldier, 140
getElementum
yuki::BasicAttack, 24
yuki::Soldier, 141
getElementumReaction
yuki, 12
getElementumReactionType
yuki, 12
getElementumType
yuki::ElementumPanel, 44
yuki::Soldier, 141
getFilledColor
yuki::StatusBar, 159
getFloatingBubble
yuki::MilitaryBase, 102
yuki::Soldier, 141
getFloatingBubbleIndexByPosition
yuki::MilitaryBase, 102
getFloatingBubbleVisible
yuki::MilitaryBase, 102
getGranuleAttackCD
yuki::NormalSoldier, 116
getHealth
yuki::Soldier, 142
getHealthBar
yuki, 13
getIndexByPosition
yuki::FloatingBubble, 50
getLayout
yuki::Menu, 96
getLevel
yuki::MilitaryBase, 103
getMaxHealth
yuki::MilitaryBase, 103
yuki::Soldier, 142
getMaxValue
yuki::SkillIcon, 133
yuki::StatusBar, 159
getNormalSoldier
yuki, 13
getOrigin
yuki::Button, 32
yuki::Soldier, 142
getPlaceOffset
yuki::Soldier, 142
getPoint
yuki::CircularSector, 39
getPointCount
yuki::CircularSector, 39
getPosition
yuki::Button, 32
yuki::FloatingBubble, 50
yuki::GranuleAttack, 60
yuki::InfoBar, 64
yuki::Soldier, 143
yuki::StatusBar, 159
getReactedElementum
yuki, 14
getSize
yuki::Button, 32
yuki::InfoBar, 64
yuki::Map, 90
yuki::Soldier, 143
getSkillNum
yuki::SkillBar< MaxSkillNum >, 128
getSoldier
yuki, 14
getSpeed
yuki::Soldier, 143
getString
yuki::Button, 32
getTexture

yuki::Button, 32
getTile
yuki::Map, 90, 91
getTileCategoryName
yuki, 15
getTileInfo
yuki::Tile, 167
getTileScaleFactor
yuki, 15
getTileSize
yuki::Map, 91
getTileSrcName
yuki, 15
getValue
yuki::StatusBar, 159
getVectorSpeed
yuki, 16
granule_
yuki::GranuleAttack, 61
granule_attack_cd_
yuki::NormalSoldier, 117
granule_attack_clock_
yuki::NormalSoldier, 117
granule_attacks_
yuki::MainScene, 86
GranuleAttack
yuki::GranuleAttack, 59
GranuleAttackWithSenderInfo
yuki::MainScene, 69
Gray
yuki::YukiColor, 18
Greenery
yuki, 12

health_bar_
yuki::MilitaryBase, 107
yuki::Soldier, 150
health_bar_offset_
yuki::Soldier, 150
health_bar_offset_y
yuki::MilitaryBase, 107
hide
yuki::StatusBar, 160
Horizontal
yuki::Menu, 95
Hover
yuki::Touchable, 171
Hydro
yuki, 11

icon_
yuki::SkillIcon, 134
icon_money_
yuki::InfoBar, 66
icon_money_texture_
yuki::InfoBar, 66
icon_radius_
yuki::SkillBar< MaxSkillNum >, 129
implMsgGenerateEnemySoldier
yuki::MainScene, 74
implMsgGenerateOwnSoldier
yuki::MainScene, 75
increase
yuki::StatusBar, 160
increaseEnemyMoney
yuki::MainScene, 75
increaseHealth
yuki::Soldier, 143
increaseMoney
yuki::MainScene, 76
increaseOwnMoney
yuki::MainScene, 76
index
yuki::TileInfo, 170
info_
yuki::BasicAttack, 24
info_bar_
yuki::MainScene, 86
InfoBar
yuki::InfoBar, 63
initBuildings
yuki::MainScene, 77
initButton
yuki::MainScene, 77
initElementumPanel
yuki::MainScene, 78
initFont
yuki::GameFont, 18
initInfoHint
yuki::MainScene, 79
initMap
yuki::MainScene, 79
initTexture
yuki::MilitaryBase, 103
initTextures
yuki::InfoBar, 64
initUi
yuki::InfoBar, 64
yuki::MainScene, 80
yuki::ResultScene, 120
yuki::StartupScene, 153
inRange
yuki::Touchable, 174
yuki::TouchableObject< Decorated >, 179
yuki::TouchableSprite, 183
InteractiveObject.hpp
FOCUSABLE_OBJECT, 211
TOUCHABLE_OBJECT, 211
is_clicked_
yuki::Touchable, 176
is_focused_
yuki::Focusable, 57
is_freeze_
yuki::Soldier, 150
is_hovered_
yuki::Touchable, 176
is_moving_

yuki::Soldier, 150
isFloatingBubbleVisible
yuki::Soldier, 144
isFocused
yuki::Focusable, 55
isFreezed
yuki::Soldier, 144
isMoving
yuki::Soldier, 144
isOppositeDirection
yuki, 16
isVisible
yuki::StatusBar, 161

kEnemyBaseCoordinate
yuki::MainScene, 86
kEnemySoldierBirthCoordinate
yuki::MainScene, 87
kIconSize
yuki::InfoBar, 66
kMoneyIconOffset
yuki::InfoBar, 66
kMoneyTextOffset
yuki::InfoBar, 66
kOwnBaseCoordinate
yuki::MainScene, 87
kOwnSoldierBirthCoordinate
yuki::MainScene, 87
kPi
CircularSector.cpp, 242
FloatingBubble.cpp, 244
yuki, 17
kSkillIconBorderDistance
yuki, 17
kWindowHeight
yuki, 17
kWindowWidth
yuki, 17

Land
yuki, 12
Layout
yuki::Menu, 95
layout_
yuki::Menu, 97
Leave
yuki::Touchable, 171
Left
yuki, 11
yuki::Button, 30
level
yuki::Elementum, 42
level_
yuki::MilitaryBase, 107
levelDown
yuki::MilitaryBase, 103
levelUp
yuki::MilitaryBase, 104
loadTexture
yuki::BubbleItem, 27

main
Main.cpp, 246
Main.cpp
main, 246
MainScene
yuki::MainScene, 70
Map
yuki::Map, 89
map_
yuki::MainScene, 87
max_frame_
yuki::NormalSoldier, 117
yuki::SeniorSoldier, 125
max_value_
yuki::SkillIcon, 134
yuki::StatusBar, 166
Melt
yuki, 11
Menu
yuki::Menu, 95
menu_items_
yuki::Menu, 97
Message
yuki::MainScene, 70
message_queue_
yuki::MainScene, 87
MilitaryBase
yuki::MilitaryBase, 100
mouse_event_engine_
yuki::YukiScene, 188
MouseEventProcessEngine
yuki::MouseEventProcessEngine, 108

None
yuki, 11
yuki::MainScene, 70
yuki::YukiScene, 185
Normal
yuki::YukiColor, 19
NormalSoldier
yuki::NormalSoldier, 113
normalSoldierAttack
yuki::MainScene, 80
Notify
yuki::YukiScene, 184
notify_
yuki::YukiScene, 189

ObjectType
yuki::MainScene, 70
on_clickCallback_
yuki::Touchable, 177
on_focusCallback_
yuki::Focusable, 57
on_hoverCallback_
yuki::Touchable, 177
on_leaveCallback_

yuki::Touchable, 177
on_releaseCallback_
yuki::Touchable, 177
on_unfocusCallback_
yuki::Focusable, 57
onClick
yuki::Touchable, 174
onFocus
yuki::Focusable, 55
onHover
yuki::Touchable, 174
onLeave
yuki::Touchable, 175
onRelease
yuki::Touchable, 175
onUnfocus
yuki::Focusable, 56
operator[]
yuki::Menu, 96
Other
yuki, 10, 11
Own
yuki, 10
own_base_
yuki::MainScene, 87
own_current_elementum_type_
yuki::MainScene, 87
own_money_
yuki::MainScene, 87
own_win_
yuki::ResultScene, 120
OwnBase
yuki::MainScene, 70
OwnBaseLevelUp
yuki::MainScene, 70
OwnSoldier
yuki::MainScene, 70
OwnWin
yuki::MainScene, 70

pixelToCoordinate
yuki::MainScene, 81
place_offset_
yuki::NormalSoldier, 117
yuki::SeniorSoldier, 125
yuki::Soldier, 150
position_
yuki::FloatingBubble, 52
yuki::Menu, 97
yuki::SkillBar< MaxSkillNum >, 129
yuki::StatusBar, 166
previous_frame_
yuki::NormalSoldier, 117
yuki::SeniorSoldier, 126
processEvent
yuki::MouseEventProcessEngine, 108
yuki::YukiScene, 185
Pyro
yuki, 11
radius_
yuki::CircularSector, 41
registerGranuleAttack
yuki::MainScene, 82
registerTouchableObject
yuki::MouseEventProcessEngine, 108
yuki::YukiScene, 186
Release
yuki::Touchable, 171
resetGenerateSoldierClock
yuki::MilitaryBase, 104
resetGranuleAttackClock
yuki::NormalSoldier, 116
result_text_
yuki::ResultScene, 120
ResultScene
yuki::ResultScene, 119
ret_code_
yuki::YukiScene, 189
Right
yuki, 11
yuki::Button, 30
River
yuki, 12
Road
yuki, 12

sendMessage
yuki::MainScene, 82
SeniorSoldier
yuki::SeniorSoldier, 124
setAdhesionElementum
yuki::Soldier, 144
setBackgroudColor
yuki::StatusBar, 161
setBubbleRadius
yuki::FloatingBubble, 50
setCenter
yuki::CircularSector, 40
setColor
yuki::Soldier, 144
setCurrentHealth
yuki::MilitaryBase, 104
setCurrentValue
yuki::SkillIcon, 133
setDirection
yuki::Soldier, 144
setElementum
yuki::Soldier, 145
setElementumType
yuki::ElementumPanel, 44
yuki::Soldier, 145
setEndAngle
yuki::CircularSector, 40
setEnemyCurrentElementumType
yuki::MainScene, 83
setFillColor
yuki::Button, 32
setFilledColor

yuki::StatusBar, 161
setFilledPercentage
yuki::StatusBar, 161
setFloatingBubbleVisible
yuki::MilitaryBase, 104
yuki::Soldier, 145
setGranuleAttackCD
yuki::NormalSoldier, 116
setHealth
yuki::Soldier, 145
setLayout
yuki::Menu, 97
setMaxHealth
yuki::MilitaryBase, 105
yuki::Soldier, 145
setMaxValue
yuki::SkillIcon, 133
yuki::StatusBar, 162
setMoney
yuki::InfoBar, 65
yuki::MainScene, 83
setMoving
yuki::Soldier, 146
setOrigin
yuki::Button, 32
yuki::Soldier, 146
setOutlineColor
yuki::Button, 33
yuki::StatusBar, 162
setOutlineStyle
yuki::StatusBar, 163
setOutlineThickness
yuki::Button, 33
yuki::StatusBar, 163
setOwnCurrentElementumType
yuki::MainScene, 83
setPlaceOffset
yuki::Soldier, 146
setPosition
yuki::Button, 33, 34
yuki::FloatingBubble, 50
yuki::InfoBar, 65
yuki::MilitaryBase, 105
yuki::SkillIcon, 133
yuki::Soldier, 147
yuki::StatusBar, 163
setRadius
yuki::CircularSector, 40
setReturnCode
yuki::YukiScene, 187
setSize
yuki::Button, 35
yuki::InfoBar, 65
yuki::Map, 92
yuki::StatusBar, 164
setSpeed
yuki::Soldier, 147
setStartAngle
yuki::CircularSector, 40
setString
yuki::Button, 35
setTextAlignment
yuki::Button, 36
setTextFillColor
yuki::Button, 36
setTextFont
yuki::Button, 36
setTextOutlineColor
yuki::Button, 37
setTextOutlineThickness
yuki::Button, 37
setTextSize
yuki::Button, 37
setTexture
yuki::Button, 37
setTextureRect
yuki::Button, 37
setTextureSrc
yuki::Tile, 168
setTile
yuki::Map, 92
setTileInfo
yuki::Tile, 168
setTileSize
yuki::Map, 93
setValue
yuki::StatusBar, 164
shape_
yuki::Button, 38
show
yuki::StatusBar, 165
yuki::YukiScene, 187
size_
yuki::Map, 93
yuki::StatusBar, 166
skill_bar_
yuki::MainScene, 88
skill_num_
yuki::SkillBar< MaxSkillNum >, 130
SkillBar
yuki::SkillBar< MaxSkillNum >, 128
SkillIcon
yuki::SkillIcon, 132
skills_
yuki::SkillBar< MaxSkillNum >, 130
Soldier
yuki::Soldier, 138
soldiers_
yuki::MainScene, 88
soldierUpdatePosition
yuki::MainScene, 84
spacing_
yuki::Menu, 97
speed_
yuki::GranuleAttack, 61
yuki::Soldier, 150

sprite_
yuki::Soldier, 151
start_angle_
yuki::CircularSector, 41
StartupScene
yuki::StartupScene, 152
StatusBar
yuki::StatusBar, 157
Stones
yuki, 12

text_
yuki::Button, 38
text_money_
yuki::InfoBar, 66
texture_
yuki::BubbleItem, 27
yuki::MilitaryBase, 107
yuki::Tile, 168
tile_info_
yuki::Tile, 168
tile_size_
yuki::Map, 93
TileCategory
yuki, 11
TileInfo
yuki::TileInfo, 169
tiles_
yuki::Map, 94
Touchable
yuki::Touchable, 172
TOUCHABLE_OBJECT
InteractiveObject.hpp, 211
touchable_objects_
yuki::MouseEventProcessEngine, 109
TouchableObject
yuki::TouchableObject< Decorated >, 179
TouchableSprite
yuki::TouchableSprite, 182
Transparent
yuki::YukiColor, 19
Transparent_25
yuki::YukiColor, 19
Transparent_50
yuki::YukiColor, 20
Transparent_75
yuki::YukiColor, 20
Tree
yuki, 12
type
yuki::Elementum, 42
yuki::ElementumReaction, 46

unregisterTouchableObject
yuki::MouseEventProcessEngine, 109
yuki::YukiScene, 188
Up
yuki, 11
update
yuki::GranuleAttack, 60
yuki::NormalSoldier, 116
yuki::SeniorSoldier, 125
yuki::Soldier, 147
updateAttacks
yuki::MainScene, 84
updateBubblesPosition
yuki::FloatingBubble, 51
updateCircularSector
yuki::SkillIcon, 133
updateFilledPercentage
yuki::StatusBar, 165
updateGranule
yuki::GranuleAttack, 60
updateInfo
yuki::YukiScene, 188
updatePosition
yuki::SkillBar< MaxSkillNum >, 129
updateStatus
yuki::Focusable, 56
yuki::Touchable, 175
yuki::TouchableObject< Decorated >, 180
updateTexture
yuki::MilitaryBase, 105

Vertical
yuki::Menu, 95
visible_
yuki::StatusBar, 166

walk
yuki::Soldier, 148
window_
yuki::YukiScene, 189

yuki, 9
Building, 11
Camp, 10
Cyro, 11
Decor, 12
Direction, 10
Down, 11
ElementumReactionType, 11
ElementumType, 11
Enemy, 10
Evaporation, 11
Freeze, 11
getElementumReaction, 12
getElementumReactionType, 12
getHealthBar, 13
getNormalSoldier, 13
getReactedElementum, 14
getSoldier, 14
getTileCategoryName, 15
getTileScaleFactor, 15
getTileSrcName, 15
getVectorSpeed, 16
Greenery, 12
Hydro, 11

isOppositeDirection, 16
kPi, 17
kSkillIconBorderDistance, 17
kWindowHeight, 17
kWindowWidth, 17
Land, 12
Left, 11
Melt, 11
None, 11
Other, 10, 11
Own, 10
Pyro, 11
Right, 11
River, 12
Road, 12
Stones, 12
TileCategory, 11
Tree, 12
Up, 11
yuki::AttackInfo, 21
AttackInfo, 21
damage, 21
elementum_level, 22
elementum_type, 22
yuki::BasicAttack, 22
~BasicAttack, 23
BasicAttack, 23
getAttackInfo, 23
getDamage, 24
getElementum, 24
info_, 24
yuki::BubbleItem, 25
~BubbleItem, 27
BubbleItem, 26
FloatingBubble, 27
loadTexture, 27
texture_, 27
yuki::Button, 28
~Button, 31
adjustTextPosition, 31
Alignment, 30
alignment_, 37
autoAdjustTextSize, 31
Button, 30, 31
Center, 30
draw, 31
getOrigin, 32
getPosition, 32
getSize, 32
getString, 32
getTexture, 32
Left, 30
Right, 30
setFillColor, 32
setOrigin, 32
setOutlineColor, 33
setOutlineThickness, 33
setPosition, 33, 34
setSize, 35
setString, 35
setTextAlignment, 36
setTextFillColor, 36
setTextFont, 36
setTextOutlineColor, 37
setTextOutlineThickness, 37
setTextSize, 37
setTexture, 37
setTextureRect, 37
shape_, 38
text_, 38
yuki::CircularSector, 38
center_, 40
CircularSector, 39
end_angle_, 41
getCenter, 39
getPoint, 39
getPointCount, 39
radius_, 41
setCenter, 40
setEndAngle, 40
setRadius, 40
setStartAngle, 40
start_angle_, 41
yuki::Elementum, 41
Elementum, 41
level, 42
type, 42
yuki::ElementumPanel, 42
activate, 44
deactivate, 44
elementum_type_, 45
ElementumPanel, 44
getElementumType, 44
setElementumType, 44
yuki::ElementumReaction, 46
elementum, 46
type, 46
yuki::FloatingBubble, 47
bubble_num_, 52
bubble_radius_, 52
bubbles_, 52
direction_, 52
distribute_radius_, 52
draw, 48
FloatingBubble, 48
getBubbleItems, 49
getBubblePosition, 49
getBubbleRadius, 49
getIndexByPosition, 50
getPosition, 50
position_, 52
setBubbleRadius, 50
setPosition, 50
updateBubblesPosition, 51
yuki::Focusable, 53
~Focusable, 55

bindFocus, 55
bindUnfocus, 55
Callback, 55
Focusable, 55
is_focused_, 57
isFocused, 55
on_focusCallback_, 57
on_unfocusCallback_, 57
onFocus, 55
onUnfocus, 56
updateStatus, 56
yuki::GameFont, 18
DefaultFont, 18
initFont, 18
yuki::GranuleAttack, 58
~GranuleAttack, 60
direction_, 61
draw, 60
getPosition, 60
granule_, 61
GranuleAttack, 59
speed_, 61
update, 60
updateGranule, 60
yuki::InfoBar, 62
~InfoBar, 63
background_, 66
background_texture_, 66
draw, 64
getPosition, 64
getSize, 64
icon_money_, 66
icon_money_texture_, 66
InfoBar, 63
initTextures, 64
initUi, 64
kIconSize, 66
kMoneyIconOffset, 66
kMoneyTextOffset, 66
setMoney, 65
setPosition, 65
setSize, 65
text_money_, 66
yuki::MainScene, 67
~MainScene, 70
attack_button_, 85
attack_button_texture_, 85
coordinateToPixel, 71
debug_text_, 85
decreaseEnemyMoney, 71
decreaseMoney, 72
decreaseOwnMoney, 72
elementum_panels_, 85
enemies_, 85
enemy_base_, 86
enemy_current_elementum_type_, 86
enemy_money_, 86
EnemyBase, 70
enemyBaseMakeDecision, 73
EnemySoldier, 70
EnemyWin, 70
eraseDeadSoldier, 73
focused_object_, 86
focused_object_type_, 86
FocusedObjectChanged, 70
GenerateEnemySoldier, 70
GenerateOwnSoldier, 70
generateSoldier, 74
granule_attacks_, 86
GranuleAttackWithSenderInfo, 69
implMsgGenerateEnemySoldier, 74
implMsgGenerateOwnSoldier, 75
increaseEnemyMoney, 75
increaseMoney, 76
increaseOwnMoney, 76
info_bar_, 86
initBuildings, 77
initButton, 77
initElementumPanel, 78
initInfoHint, 79
initMap, 79
initUi, 80
kEnemyBaseCoordinate, 86
kEnemySoldierBirthCoordinate, 87
kOwnBaseCoordinate, 87
kOwnSoldierBirthCoordinate, 87
MainScene, 70
map_, 87
Message, 70
message_queue_, 87
None, 70
normalSoldierAttack, 80
ObjectType, 70
own_base_, 87
own_current_elementum_type_, 87
own_money_, 87
OwnBase, 70
OwnBaseLevelUp, 70
OwnSoldier, 70
OwnWin, 70
pixelToCoordinate, 81
registerGranuleAttack, 82
sendMessage, 82
setEnemyCurrentElementumType, 83
setMoney, 83
setOwnCurrentElementumType, 83
skill_bar_, 88
soldiers_, 88
soldierUpdatePosition, 84
updateAttacks, 84
yuki::Map, 88
~Map, 89
bg_texture_, 93
bg_tiles_, 93
draw, 89
getCoordinate, 90

getSize, 90
getTile, 90, 91
getTileSize, 91
Map, 89
setSize, 92
setTile, 92
setTileSize, 93
size_, 93
tile_size_, 93
tiles_, 94
yuki::Menu, 94
~Menu, 96
addItem, 96
adjustItemsPosition, 96
draw, 96
getByName, 96
getLayout, 96
Horizontal, 95
Layout, 95
layout_, 97
Menu, 95
menu_items_, 97
operator[], 96
position_, 97
setLayout, 97
spacing_, 97
Vertical, 95
yuki::MilitaryBase, 98
~MilitaryBase, 100
camp_, 106
canGenerateSoldier, 101
draw, 101
floating_bubble_, 106
floating_bubble_visible_, 106
generateSoldierClock_, 107
getAttacked, 101
getCurrentHealth, 101
getFloatingBubble, 102
getFloatingBubbleIndexByPosition, 102
getFloatingBubbleVisible, 102
getLevel, 103
getMaxHealth, 103
health_bar_, 107
health_bar_offset_y, 107
initTexture, 103
level_, 107
levelDown, 103
levelUp, 104
MilitaryBase, 100
resetGenerateSoldierClock, 104
setCurrentHealth, 104
setFloatingBubbleVisible, 104
setMaxHealth, 105
setPosition, 105
texture_, 107
updateTexture, 105
yuki::MouseEventProcessEngine, 107
~MouseEventProcessEngine, 108
MouseEventProcessEngine, 108
processEvent, 108
registerTouchableObject, 108
touchable_objects_, 109
unregisterTouchableObject, 109
yuki::NormalSoldier, 110
~NormalSoldier, 113
all_textures_, 117
canGranuleAttack, 114
current_frame_, 117
draw, 114
getDefaultAttack, 114
getDefaultGranuleAttack, 115
getGranuleAttackCD, 116
granule_attack_cd_, 117
granule_attack_clock_, 117
max_frame_, 117
NormalSoldier, 113
place_offset_, 117
previous_frame_, 117
resetGranuleAttackClock, 116
setGranuleAttackCD, 116
update, 116
yuki::ResultScene, 118
~ResultScene, 119
background_, 120
background_texture_, 120
initUi, 120
own_win_, 120
result_text_, 120
ResultScene, 119
yuki::SeniorSoldier, 121
~SeniorSoldier, 124
all_textures_, 125
current_frame_, 125
draw, 125
floating_bubble_, 125
max_frame_, 125
place_offset_, 125
previous_frame_, 126
SeniorSoldier, 124
update, 125
yuki::SkillBar< MaxSkillNum >, 126
border_thick_, 129
boxes_, 129
draw, 128
getSkillNum, 128
icon_radius_, 129
position_, 129
skill_num_, 130
SkillBar, 128
skills_, 130
updatePosition, 129
yuki::SkillIcon, 130
background_, 134
circular_sector_, 134
current_value_, 134
draw, 132

getCurrentValue, 132
getMaxValue, 133
icon_, 134
max_value_, 134
setCurrentValue, 133
setMaxValue, 133
setPosition, 133
SkillIcon, 132
updateCircularSector, 133
yuki::Soldier, 135
~Soldier, 138
adhesion_elementum_, 149
camp_, 149
clock_, 149
decreaseHealth, 139
direction_, 149
draw, 139
elementum_, 149
floating_bubble_, 149
floating_bubble_visible_, 150
freeze_clock_, 150
getAdhesionElementum, 139
getAttacked, 140
getCamp, 140
getColor, 140
getDefaultAttack, 140
getDirection, 140
getElementum, 141
getElementumType, 141
getFloatingBubble, 141
getHealth, 142
getMaxHealth, 142
getOrigin, 142
getPlaceOffset, 142
getPosition, 143
getSize, 143
getSpeed, 143
health_bar_, 150
health_bar_offset_, 150
increaseHealth, 143
is_freeze_, 150
is_moving_, 150
isFloatingBubbleVisible, 144
isFreezed, 144
isMoving, 144
place_offset_, 150
setAdhesionElementum, 144
setColor, 144
setDirection, 144
setElementum, 145
setElementumType, 145
setFloatingBubbleVisible, 145
setHealth, 145
setMaxHealth, 145
setMoving, 146
setOrigin, 146
setPlaceOffset, 146
setPosition, 147
setSpeed, 147
Soldier, 138
speed_, 150
sprite_, 151
update, 147
walk, 148
yuki::StartupScene, 151
~StartupScene, 153
background_, 154
background_texture_, 155
btn_connect_, 155
btn_start_, 155
initUi, 153
StartupScene, 152
yuki::StatusBar, 155
~StatusBar, 157
adjustIfOverflow, 157
background_, 166
current_value_, 166
decrease, 158
draw, 158
filled_, 166
getBackgroundColor, 158
getFilledColor, 159
getMaxValue, 159
getPosition, 159
getValue, 159
hide, 160
increase, 160
isVisible, 161
max_value_, 166
position_, 166
setBackgroudColor, 161
setFilledColor, 161
setFilledPercentage, 161
setMaxValue, 162
setOutlineColor, 162
setOutlineStyle, 163
setOutlineThickness, 163
setPosition, 163
setSize, 164
setValue, 164
show, 165
size_, 166
StatusBar, 157
updateFilledPercentage, 165
visible_, 166
yuki::Tile, 167
getTileInfo, 167
setTextureSrc, 168
setTileInfo, 168
texture_, 168
tile_info_, 168
yuki::TileInfo, 169
category, 169
index, 170
TileInfo, 169
yuki::Touchable, 170

~Touchable, 172
Action, 171
bindAction, 172
bindClick, 172
bindHover, 172
bindLeave, 173
bindRelease, 173
Callback, 171
Click, 171
Hover, 171
inRange, 174
is_clicked_, 176
is_hovered_, 176
Leave, 171
on_clickCallback_, 177
on_hoverCallback_, 177
on_leaveCallback_, 177
on_releaseCallback_, 177
onClick, 174
onHover, 174
onLeave, 175
onRelease, 175
Release, 171
Touchable, 172
updateStatus, 175
yuki::TouchableObject< Decorated >, 178
~TouchableObject, 179
inRange, 179
TouchableObject, 179
updateStatus, 180
yuki::TouchableSprite, 180
~TouchableSprite, 183
inRange, 183
TouchableSprite, 182
yuki::YukiColor, 18
Gray, 18
Normal, 19
Transparent, 19
Transparent_25, 19
Transparent_50, 20
Transparent_75, 20
yuki::YukiScene, 183
~YukiScene, 185
draw, 185
End, 185
mouse_event_engine_, 188
None, 185
Notify, 184
notify_, 189
processEvent, 185
registerTouchableObject, 186
ret_code_, 189
setReturnCode, 187
show, 187
unregisterTouchableObject, 188
updateInfo, 188
window_, 189
YukiScene, 185
YUKI_DEBUG
Config.hpp, 200
YUKI_SCENE
YukiScene.hpp, 239
YukiScene
yuki::YukiScene, 185
YukiScene.hpp
YUKI_SCENE, 239