

LISTADE VARIAVEIS DEADLY-WISH

knight.h

MÉTODOS PROTEGIDOS

Nome Antigo	Novo Nome	Tipo	Descrição
heavy_attack	do_heavy_attack	void	Realiza o ataque forte
light_attack	do_light_attack	void	Realiza o ataque leve
defense	do_defense	void	Realiza habilidade de defesa
special	do_special	void	Realiza golpe especial

ATRIBUTOS PROTEGIDOS

Nome Antigo	Novo Nome	Tipo	Descrição
m_sprite_paths	sprite_paths	vector<string>	Local do arquivo de sprite do personagem.
id	player_id	unsigned	Guarda qual player está controlando este personagem.
x	x_position	double	Guarda a posição x atual do personagem.
y	y_position	double	Guarda a posição y atual do personagem.
character_code	character_id	int	Identificador do personagem.

knight.cpp

MÉTODOS PROTEGIDOS

Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
heavy_attack	do_heavy_attack	void	Realiza o ataque forte
light_attack	do_light_attack	void	Realiza o ataque leve
defense	do_defense	void	Realiza habilidade de defesa
special	do_special	void	Realiza golpe especial
change_character_state	change_character_state	void	Muda estado do personagem. Exemplo: estado morto

ATRIBUTOS PROTEGIDOS

Nome Antigo	Novo Nome	Tipo	Descrição
m_special_cooldown	special_cooldown	int	Tempo de cooldown para ataques especiais do personagem.
m_heavy_attack_cooldown	heavy_attack_cooldown	int	Tempo de cooldown para ataques pesados do personagem.
m_light_attack_cooldown	light_attack_cooldown	int	Tempo de cooldown para ataques leves do personagem.
m_defense_cooldown	defense_cooldown	int	Tempo de cooldown para habilidade de defesa do personagem.
m_last_used_heavy_attack	last_used_heavy_attack	int	Ultimo vez ataque pesado foi usado.
m_last_used_light_attack	last_used_light_attack	int	Ultimo vez ataque leve foi usado.
m_last_used_defense	last_used_defense	int	Ultimo vez habilidade de defesa foi usada.
m_last_used_special	last_used_special	int	Ultimo vez ataque especial foi usado.
m_active	active	bool	Atributo para dizer se o personagem está ativo ou não.
light_attack_x_pos	light_attack_x_position	double	Posição x no momento do ataque leve
m_moving_state	moving_state	MovingState	Atributo para estado de movimento do personagem.
m_sprite_paths	sprite_paths	vector<string>	Local do arquivo de sprite do personagem.
id	player_id	unsigned	Guarda qual player está controlando este personagem.
x	x_position	double	Guarda a posição x atual do personagem.
y	y_position	double	Guarda a posição y atual do personagem.
character_code	character_id	int	Identificador do personagem.

light_attack.h

MÉTODOS PUBLICOS

Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
active	IsActive	bool	Retorna se o personagem está ativo.
bounding_box	GetBoundingBox	Rectangle&	Retorna o "espaço" que está o personagem
hit_boxes	GetHitBoxes	list<Rectangle>&	Retorna o "espaço" que o personagem colidiu.
direction	GetDirection	pair<double, double>	Retorna a direção que o personagem está se mexendo
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	UpdateSelf	void	Atualiza o evento de ataque
draw_self	DrawSelf	void	Desenha o personagem
choose_sprite_path	ChooseSpritePath	string	Escolhe qual o caminho para o sprite do personagem
update_sprite_state	UpdateSpriteState	void	Atualiza o estado do sprite
update_time	UpdateTime	void	Atualiza o cooldown do ataque
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	Identificador do personagem.
m_state	state	State	Atributo para estado atual do personagem.
m_sprite_state	sprite_state	SpriteState	Atributo para o estado do sprite.
m_damage	damage_attack	int	Dano do ataque leve.
m_speed	speed_attack	double	Velocidade do ataque leve.
m_frame	frame	int	
m_sprite_path	sprite_path	string	
m_start	start	unsigned	
m_current_time	current_time	unsigned	Tempo atual do jogo.
m_texture	texture	vector< shared_ptr<Texture> >	Textura usada para a imagem do personagem.
m_bounding_box	bounding_box	Rectangle	
m_dx	x_direction	double	Guarda a direção x em que o personagem está se movendo.
m_dy	y_direction	double	Guarda a direção y em que o personagem está se movendo.
mage_id	hero_id	unsigned	Identificador do personagem.
xp	x_position	double	Guarda a posição x atual do personagem.
yp	y_position	double	Guarda a posição y atual do personagem.
light_attack.cpp			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
active	IsActive	bool	Retorna se o personagem está ativo.
bounding_box	GetBoundingBox	Rectangle&	Retorna o "espaço" que está o personagem.
hit_boxes	HitBoxes	list<Rectangle>&	Retorna o "espaço" que o personagem colidiu.
direction	GetDirection	pair<double, double>	Retorna a direção para qual o personagem está se movendo.
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	UpdateSelf	void	Atualiza o evento de ataque.
draw_self	DrawSelf	void	Desenha o personagem.
update_time	UpdateTime	void	Atualiza o cooldown do ataque.
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	Identificador do personagem.
m_damage	damage_attack	int	Dano do ataque leve.

m_speed	speed_attack	double	Velocidade do ataque leve.
m_frame	frame	int	
m_start	start	unsigned	
m_current_time	current_time	unsigned	Tempo atual do jogo.
m_bounding_box	bounding_box	Rectangle	
boxes	boxes	static list<Rectangle> {bounding_box}	Guarda as posições em que o personagem pode colidir.
m_dx	x_direction	double	Guarda a direção x em que o personagem está se movendo.
m_dy	y_direction	double	Guarda a direção y em que o personagem está se movendo.
now	time_now	unsigned	Tempo atual do jogo.
mage_id	hero_id	unsigned	Identificador do personagem.
xp	x_position	double	Guarda a posição x atual do personagem.
yp	y_position	double	Guarda a posição y atual do personagem.
m_x	position_axis_x	double	Guarda a posição x atual do personagem.
m_y	position_axis_y	double	Guarda a posição y atual do personagem.
sem nome		unsigned	
sem nome		unsigned	
frost_nova.h			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
active	IsActive	bool	Retorna se o personagem está ativo.
bounding_box	GetBoundingBox	Rectangle&	Retorna o "espaço" que está o personagem.
hit_boxes	HitBoxes	list<Rectangle>&	Retorna o "espaço" que o personagem colidiu.
on_collision	OnCollision	void	Trata a colisão no momento em que ela ocorreu.
direction	GetDirection	pair<double, double>	Retorna a direção para qual o personagem está se movendo.
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	UpdateSelf	void	Atualiza o evento de ataque.
draw_self	DrawSelf	void	Desenha o personagem.
choose_sprite_path	ChooseSpritePath	string	Escolhe qual o caminho para o sprite do personagem
update_sprite_state	UpdateSpriteState	void	Atualiza o estado do sprite
update_time	UpdateTime	void	Atualiza o cooldown do ataque.
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	Identificador do personagem.
m_state	state	State	Atributo para estado atual do personagem.
m_sprite_state	sprite_state	SpriteState	Atributo para o estado do sprite.
m_damage	new_skill_damage	int	Dano do ataque leve.
m_speed	speed_attack	double	Velocidade do ataque leve.
m_frame	frame	int	
m_sprite_path	sprite_path	string	
m_start	start	unsigned	
m_current_time	current_time	unsigned	Tempo atual do jogo.
m_texture	texture	vector< shared_ptr<Texture> >	Textura usada para a imagem do personagem.
m_bounding_box	bounding_box	Rectangle	
dx	x_direction	double	Guarda a direção x em que o personagem está se movendo.
dy	y_direction	double	Guarda a direção y em que o personagem está se movendo.
now	time_now	unsigned	Tempo atual do jogo.
mage_id	hero_id	unsigned	Identificador do personagem.

xp	position_axis_x	double	Guarda a posição x atual do personagem.
yp	position_axis_y	double	Guarda a posição y atual do personagem.
m_dx	skill_x_direction	double	Guarda a direção x em que a skill está se movendo.
m_dy	skill_y_direction	double	Guarda a direção y em que a skill está se movendo.
frost_nova.cpp			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
active	IsActive	bool	Retorna se o personagem está ativo.
bounding_box	GetBoundingBox	Rectangle&	Retorna o "espaço" que está o personagem.
hit_boxes	HitBoxes	list<Rectangle>&	Retorna o "espaço" que o personagem colidiu.
on_collision	OnCollision	void	Trata a colisão no momento em que ela ocorreu.
direction	GetDirection	pair<double, double>	Retorna a direção para qual o personagem está se movendo.
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	UpdateSelf	void	Atualiza o evento de ataque.
draw_self	DrawSelf	void	Desenha o personagem.
choose_sprite_path	ChooseSpritePath	string	Escolhe qual o caminho para o sprite do personagem
update_time	UpdateTime	void	Atualiza o cooldown do ataque.
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	Identificador do personagem.
m_dx	skill_x_direction	double	Guarda a direção x em que a frost nova está se expandindo.
m_dy	skill_y_direction	double	Guarda a direção y em que a frost nova está se expandindo.
m_damage	damage_attack	int	Dano do ataque leve.
m_speed	speed_attack	double	Velocidade do ataque leve.
m_frame	frame	int	
m_sprite_path	sprite_path	string	
m_start	start	unsigned	
m_current_time	current_time	unsigned	Tempo atual do jogo.
m_texture	texture	vector < shared_ptr<Texture> >	Textura usada para a imagem do personagem.
m_bounding_box	bounding_box	Rectangle	
dx	x_direction	double	Guarda a direção x em que o personagem está se movendo.
dy	y_direction	double	Guarda a direção y em que o personagem está se movendo.
now	now_moment	unsigned	Tempo atual do jogo.
last	last_moment	unsigned	
mage_id	hero_id	unsigned	Identificador do personagem.
xp	x_position	double	Guarda a posição x atual do personagem.
yp	y_position	double	Guarda a posição y atual do personagem.
hip	hypotenuse	double	Guarda o valor da hipotenusa.
m_x	x_position_frost_nova	double	Guarda a posição x atual da frost nova.
m_y	y_position_frost_nova	double	Guarda a posição y atual da frost nova.
rect	rectangle	Rectangle	
canvas	canvas	Canvas*	
boxes	boxes	static list<Rectangle> {bounding_box}	Guarda as posições em que o personagem pode colidir.
directory	sprite	string	Guarda o nome do sprite.
sprite_path	sprite_path	string	Guarda o endereço completo do arquivo em que está o sprite.
player_id	player_id	unsigned	Identificador do player.
c	character	Character *	Guarda o personagem.

b	base	Base *	Guarda a base do personagem.
c_bool	character_verify	bool	Atributo para verificar o personagem.
b_bool	base_verify	bool	Atributo para verificar a base.
m_collided	skill_collided	int	Atributo que guarda a colisão da skill.
credits_level.h			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
done	ExitLevel	bool	Retorna se o usuário quer sair da tela de créditos.
next	goToNextLevel	string	Retorna o próximo level.
audio	Audio	string	Retorna o nome do arquivo de áudio a ser executado.
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	UpdateSelf	void	Dispara o evento de montar a tela de créditos, caso ela ainda não tenha sido desenhada.
draw_self	DrawSelf	void	Desenha a tela de créditos.
set_credits_strings	SetCreditsText	void	Monta o texto dos créditos.
on_event	ExitLevelEvent	bool	Seta o exit como true quando o player aperta o botão correspondente.
ATRIBUTOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_done	exit	bool	Atributo que guarda a intenção do player de sair da tela de créditos.
m_next	next_level	string	Atributo que guarda o próximo level após a tela de créditos.
m_credits_strings	credits_text	vector <string>	Guarda o texto dos créditos.
m_start	start	int	Atributo que guarda se a tela de créditos deve ser montada.
m_winner_player	winner_player	int	Atributo não utilizado.
character_factory.cpp/character_factory.h			
Arquivos.h			
Nome Antigo	Novo Nome	Tipo	Descrição
character_factory	character_factory	.h	cria personagem
character	character	.h	define personagem
mage	mage	.h	define personagem mago
infiltrator	infiltrator	.h	define personagem infiltrador
soldier	soldier	.h	define personagem soldado
knight	knight	.h	define personagem cavaleiro
util	util	.h	
Constantes			
Nome Antigo	Novo Nome	Valor	Descrição
-	IDLE_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + chosen_character + "Idle.png"	
-	WALK_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Walk.png"	
-	DEATH_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Death.png"	
-	ATTACK_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Attack.png"	
-	SPECIAL_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Special.png"	
-	ULTIMATE_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Ultimate.png"	
-	DEFENSE_PNG	"Spritesheets/" + color_identifier + "/Spritesheet" + color_identifier + chosen_character + "Defense.png"	

CHARACTER_FACTORY_H	-	-	-
Classes			
Nome Antigo	Novo Nome	Tipo	Descrição
CharacterFactory	CharacterFactory	class	-
Métodos Protegidos			
Nome Antigo	Novo Nome	Tipo	Descrição
Players	Players	enum	enumera jogadores de 1-4
choose_sprite_path()	choose_sprite_path()	vector<string>	escolhe o caminho que a animação irá fazer
Métodos Públicos			
Nome Antigo	Novo Nome	Tipo	Descrição
CharacterFactory()	CharacterFactory()		
~CharacterFactory()	~CharacterFactory()		
make_character()	make_character()	Character	atribui um personagem a uma cor
Atributos Protected			
Nome Antigo	Novo Nome	Tipo	Descrição
m_sprites_paths	sprites_paths	vector<string>	escolhe o caminho de um personagem
Atributos Public			
Nome Antigo	Novo Nome	Tipo	Descrição
character_code	character_code	int	identificador de personagem
player_id	player_id	unsigned	identidade do jogador
x_pos	axis_x_position	double	posição do x no eixo
y_pos	axis_y_position	double	posição do y no eixo
sprite_vector	sprite_action_vector	vector<string>	vetor de movimento da animação
directory	color_identifier	String	define a "cor" indicadora do jogador
choosen_class	choosen_character	String	escolhe o personagem do personagem
fireball.cpp/fireball.h			
Arquivos.h			
Nome Antigo	Novo Nome	Tipo	Descrição
skill	skill	.h	-
fireball	fireball	.h	-
character	character	.h	-
memory	memory	.h	-
ijengine/rectangle	ijengine/rectangle	.h	-
ijengine/game_object	ijengine/game_object	.h	-
ijengine/engine	ijengine/engine	.h	-
ijengine/color	ijengine/color	.h	-
ijengine/canvas	ijengine/canvas	.h	-
ijengine/collidable	ijengine/collidable	.h	-
Constantes			
Nome Antigo	Novo Nome	Valor	Descrição
FIREBALL_BASE_DAMAGE	FIREBALL_BASE_DAMAGE	20	-

-	CHARACTER_WIDTH	32	-
-	SPEED	100	velocidade ms
-	HEIGHT_RECTANGLE	20	altura do retangulo
-	WIDTH_RECTANGLE	20	largura do retangulo
-	UPDATE_SPRITE_TIME	300	segundos
-	UPDATE_SELF_TIME	1000	1 segundo
-	-	150	-
-	-	2000	2 segundos
FIREBALL_H	FIREBALL_H	-	-
	SPRITE_PATH	"Spritesheets/" + color_identifier + "/ObjectHadouken" + color_identifier + ".png";	

Classes

Nome Antigo	Novo Nome	Tipo	Descrição
Fireball	Fireball	Class	declara ponto de colisão/direção/personagem a ser atingido/velocidade da bola de fogo -
Skill	Skill	Class	instancia a habilidade de um jogador

Métodos Protegidos

Nome Antigo	Novo Nome	Tipo	Descrição
update_self()	update_self()	void	atualiza posição
update_time()	update_time()	void	atualiza tempo
draw_self()	draw_self()	void	-
choose_sprite_path()	choose_sprite_path()	String	escolhe o caminho que a bola de fogo vai seguir
update_sprite_state()	update_sprite_state()	void	atualiza o estado da animação
State	State	enum	listagem dos sentidos de movimentação esquerda/direita
SpriteState	SpriteState	enum	listagem das ações das animações movendo/batendo/colocado
players	Players	enum	listagem de jogadores

Métodos Públicos

Nome Antigo	Novo Nome	Tipo	Descrição
Fireball()	Fireball()	Fireball	construtor
active()	get_active()	bool	verifica se o fireball esta ativo
~Fireball()	~Fireball()	Fireball	-
bounding_box()	get_bounding_box()	const Rectangle&	retorna ponto de colisão do personagem
direction()	get_direction()	pair<double,double>	retorna direção que o fireball deve seguir
hit_box()	get_hit_box()	list<Rectangle>&	deteção de colisão

Atributos Protected

Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	identificação do personagem que esta em movimento
m_sprite_state	sprite_state	SpriteState	estados de ataque de uma animação
m_dx	direction_axis_x	double	

m_dy	direction_axis_y	double	
m_damage	damage	int	dano de movimento
m_frame	frame	int	quadro de movimento
m_texture	texture	shared_ptr<Texture>	textura do movimento na tela
m_speed	skill_speed	double	
m_sprite_path	sprites_path	String	escolhe o caminho de um personagem
m_start	start	unsigned	início do movimento de ataque
m_current_time	current_time	unsigned	tempo atual do movimento de ataque
m_state	state	State	sentido do movimento esquerda ou direita
m_bounding_box	bounding_box	Rectangle	detecção de colisão do jogador

Atributos Public

Nome Antigo	Novo Nome	Tipo	Descrição
mage_id	-	unsigned	identificador do mago
xp	axis_position_x	double	posição do eixo x
yp	axis_position_y	double	posição do eixo y
dx	direction_x	double	direção x
dy	direction_y	double	direção y
parent	-	-	-
m_state	state	-	sentido do movimento esquerda ou direita
m_sprite_path	sprites_path	String	escolhe o caminho de um personagem
player_id	-	-	identificador do jogador
directory	color_identifier	String	define a "cor" indicadora do jogador
now	now_moment	unsigned	tempo atual
m_start	start	unsigned	tempo inicial do jogo
new_y	new_x_coordinate	double	nova coordenada y do centro do retângulo
new_x	new_y_coordinate	double	nova coordenada x do centro do retângulo
last	last_moment	unsigned	última vez de skill usada
sprite_path	-	String	caminho da animação no jogo

soldier.cpp/soldier.h

Arquivos.h

Nome Antigo	Novo Nome	Tipo	Descrição
soldier	-	.h	-
spear	-	.h	-
light_attack	-	.h	-
ije02_game	-	.h	-
character	-	.h	-

Constantes

Nome Antigo	Novo Nome	Valor	Descrição
MAX_LIFE	-	150	máximo de vidas do personagem
EFFECTS_VOLUME	-	100	volume do efeitos sonoros
-	RECHARGE_DEFENSE	300	tempo de recarga para defesa
-	RECHARGE_LIGHT_ATTACK	300	tempo de recarga ataque leve
-	-	20	-

Nome Antigo	Novo Nome	Descrição	Tipo
WinnerLevel	WinnerLevel	classe da tela do vencedor	classe
MAX_W	MAX_WEIGHT	constante que define a largura máxima	constante
MAX_H	MAX_HEIGHT	constante que define a altura máxima	constante
done	get_level_done	Indica que o nível foi finalizado	método(bool)
next	get_next_level	Indica o próximo nível	método(string)
audio	get_level_audio	Indica o áudio do nível do jogo	método(string)
update_self	update_self	Atualiza a fase	método(void)
draw_self	draw_self	Desenha a fase	método(void)
m_done	level_done	Indica que alguém venceu	bool
m_congratulations	congratulation_level	string que indica o jogador que venceu	string
m_start	start_level	indica que começou a nova partida	int
m_winner_player	winner_player	Indica qual o jogador que venceu	int
set_congratulations_string	congratulations_string	Indica qual jogador venceu com uma string	método(void)
m_next	next_level	Indica o próximo nível	string

winner_level.cpp

Nome Antigo	Novo Nome	Descrição	Tipo
stop_audio_channel	stop_audio_channel	pausa o áudio	método
play_sound_effect	play_sound_effect	começa o áudio	método
font	font	font que mostra para o usuário	auto
m_start	set_start_level	indica que começou a nova partida	int
now			
m_winner_player	set_winner_player	Indica qual o jogador que venceu	int
m_congratulations	set_congratulation_level	string que indica o jogador que venceu	string
set_congratulations_string	set_congratulations_string	Indica qual jogador venceu com uma string	método(void)
m_congratulations	set_congratulation_level	string que indica o jogador que venceu	string
m_done	set_level_done	Indica que alguém venceu	bool
m_next	set_next_level	Indica o próximo nível	função(void)

special_state.cpp

Nome Antigo	Novo Nome	Descrição	Tipo
SpecialState	SpecialStateCharacter	estado especial do personagem	classe
m_current_state	set_current_state	Indica o estado atual do personagem	MÉTODO
SPECIAL_STATE	SPECIAL_STATE	estado atual recebe estado especial	constante
m_refresh_rate	set_refresh_rate	taxa de atualização do estado	unsigned

special_state.h

Nome Antigo	Novo Nome	Descrição	Tipo
SpecialState	SpecialStateCharacter	estado especial do personagem	classe
CharacterState	CharacterState	estado do personagem	classe

test_level_factory.h			
Nome Antigo	Novo Nome	Descrição	Tipo
character_selection	character_selection	escolher o personagem	namespace
players_characters	player_options	opções de personagens	vector int
winner	winner_player	jogador vencedor	namespace
chosen_mode	chosen_mode	escolher modulo do jogo	string
TestLevelFactory	TestLevelFactory	testar a fabricação do level	classe
make_level	make_level	fazer o nivel	
release	release_level	fabricar o nivel	método(void)
game_mode	game_mode	modulo do jogo	namespace
test_level_factory.cpp			
Nome Antigo	Novo Nome	Descrição	Tipo
character_selection	character_selection	escolher o personagem	namespace
players_characters	player_options	opções de personagens	vector int
winner_players	winner_player	jogador vencedor	namespace
game_mode	game_mode	modulo do jogo	namespace
chosen_mode	chosen_mode	escolher modulo do jogo	string
TestLevelFactory	TestLevelFactory	testar a fabricação do level	classe
make_level	make_level	chama o construtor para a fase	método
level_id	level_id	identificador da tela	string
MenuLevel	MenuLevel	tela do menu	Classe
CharacterChooseLevel	CharacterChooseLevel	tela para escolher o personagem	Classe
MainLevel	MainLevel	tela principal do jogo	Classe
DeathMatchLevel	DeathMatchLevel	tela principal partida da morte	Classe
character_selection	character_selection	escolha do personagem	
players_characters	player_options	opções de personagens	vector int
CreditsLevel	CreditsLevel	tela dos créditos	Classe
TestLevelFactory	TestLevelFactory	testar a fabricação do level	classe
translator.h			
Nome Antigo	Novo Nome	Descrição	Tipo
Translator	Translator	traduzir todos eventos do jogo	Classe
translate	translate	traduzir determinado evento	Método(bool)
MouseEvent	MouseEvent	evento do mouse	
SystemEvent	SystemEvent	evento do sistema	
KeyboardEvent	KeyboardEvent	evento do teclado	
JoystickEvent	JoystickEvent	evento do joystick	
translator.h			
Nome Antigo	Novo Nome	Descrição	Tipo
set_timestamp	set_timestamp	periodo de tempo do movimento do jogador	Método
set_property	set_axis	definir o eixo que o personagem vai andar	Método
set_id	set_id_event	id do evento ocorrido	Método
done	get_level_done	Indica que o nivel foi finalizado	método(bool)
MouseEvent	MouseEvent	evento do mouse	
SystemEvent	SystemEvent	evento do sistema	
KeyboardEvent	KeyboardEvent	evento do teclado	
JoystickEvent	JoystickEvent	evento do joystick	

character_choose_level.h			
Classe CharacterChooseLevel			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
done	is_done	bool const	retorna o atributo m_done
next	get_next	string const	retorna o atributo m_next
audio	get_audio	string const	retorna a musica do menu de selecao de personagem
players_characters	get_players_caracters	vector<int> const	retorna o atributo m_players_caracter
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	-	void	fica atualizando a selecao de personagem enquanto nao forem escolhidos
draw_self	-	void	responsavel por desenhar a tela de selecao de personagem
on_event	-	bool	retorna falso quando e chamada
ATRIBUTOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_players_characters	players_characters	vector<int>	vetor que enumera os personagens
ATRIBUTOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_done	done	bool	Verifica se a selecao de personagem acabou/ foi confirmada
m_next	next	string	recebe a proxima tela
m_start	start_level	int	inicia o level
m_frame	frame	int	
m_selected_characters	selected_character	vector<int>	vetor de personagens selecionados
m_current_player	current_player	int	numero que representa o jogador atual
m_number_of_players	number_of_players	int	numero que representa o numero de jogadores
m_character_selections	character_selections	vector<CharacterSelection *>	responsavel pela escolha do personagem
m_current_player_character	current_player_character	int	inteiro que representa o personagem atual do jogador
m_textures	textures	vector<shared_ptr<Texture> >	vetor que armazena as texturas/imagens
character_choose_level.cpp			
Classe CharacterChooseLevel			
Construtor			
Nome Antigo	Novo Nome	Tipo	Descrição
base_x	virou uma constante	int	usada em apenas uma iteracao
i	-	int	variavel auxiliar para iteracao
update_self			
Nome Antigo	Novo Nome	Tipo	Descrição
dones	-	int	sentinela para ver se terminou a selecao
draw_self			
Nome Antigo	Novo Nome	Tipo	Descrição
portrait_rects	-	vector<Rectangle>	tamanho das imagens dos personagens
rects_positions	-	vector< pair<int, int> >	posicao onde os retangulos vao ser posicionados
CONSTANTES			
Nome Antigo	Novo Nome	Tipo	Descrição
(base_x)era uma variavel	BASE_X	int	usada em uma iteracao
mage.h			
Classe Mage			
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
heavy_attack	do_heavy_attack	void	metodo responsavel pelo ataque forte
light_attack	do_light_attack	void	metodo responsavel pelo ataque fraco
defense	do_defense	void	metodo responsavel pela defesa

special	do_special	void	metodo responsavel pelo ataque especial
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_sprite_paths	sprite_paths	vector<string>	Local do arquivo de sprite do personagem.
id	character_id	unsigned	Guarda qual player está controlando este personagem.
x	x_position	double	Guarda a posição x atual do personagem.
y	y_position	double	Guarda a posição y atual do personagem.
character_code	-	int	Identificador do personagem.
mage.cpp			
Classe Mage			
Construtor			
Nome Antigo	Novo Nome	Tipo	Descrição
m_special_cooldown	special_cooldown	int	tempo de recarga do ataque especial
m_heavy_attack_cooldown	heavy_attack_cooldown	int	tempo de recarga do ataque forte
m_light_attack_cooldown	light_attack_cooldown	int	tempo de recarga do ataque fraco
m_defense_cooldown	defense_cooldown	int	tempo de recarga da defesa
m_last_used_special	last_used_special	int	marca a ultima vez que o ataque especial foi usado e é reduzido do tempo de recarga ate poder ser usado de novo
m_last_used_heavy_attack	last_used_heavy_attack	int	marca a ultima vez que o ataque forte foi usado e é reduzido do tempo de recarga ate poder ser usado de novo
m_last_used_light_attack	last_used_light_attack	int	marca a ultima vez que o ataque fraco foi usado e é reduzido do tempo de recarga ate poder ser usado de novo
m_last_used_defense	last_used_defense	int	marca a ultima vez que a defesa foi usada e é reduzido do tempo de recarga ate poder ser usado de novo
m_active	active	bool	Verifica se o personagem esta ativo ou nao
heavy_attack			
Nome Antigo	Novo Nome	Tipo	Descrição
p	-	auto	recebe o retorno da funcao parent para fazer um print
fireball_dx	fireball_x_direction	double	indica se o ataque vai pra esquerda ou direita
fireball_x_pos	fireball_x_position	double	indica a posicao do ataque no eixo x
m_moving_state	moving_state	MovingState	Atributo para estado de movimento do personagem.
light_attack			
Nome Antigo	Novo Nome	Tipo	Descrição
p	-	auto	recebe o retorno da funcao parent para fazer um print
light_attack_x_pos	light_attack_x_position	double	posicao onde o ataque vai ocorrer no eixo x
special			
Nome Antigo	Novo Nome	Tipo	Descrição
p	-	auto	recebe o retorno da funcao parent para fazer um print
CONSTANTES			
Nome Antigo	Novo Nome	Tipo	Descrição
MAX_LIFE	-	int	Vida do personagem
spear.h			
Classe Spear			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
active	is_active	bool	verifica se o personagem esta ativo
bounding_box	get_bounding_box	const Rectangle&	Retorna a caixa delimitadora
hit_boxes	get_hit_boxes	const list<Rectangle>&	Retorna a lista de hit boxes
direction	get_direction	pair<double, double>	Retorna a direcao da lanca
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	-	bool	atualiza a posicao da lanca

draw_self	-	void	desenha a lanca
update_sprite_state	-	void	atualiza os frames do sprite da lanca
update_time	-	void	atualiza o tempo para atualizar a tela
choose_sprite_path	-	string	Seleciona a cor da lanca
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_character_id	character_id	unsigned	ver qual o personagem usa a lanca
m_state	state	State	estado do personagem(parado ou para que lado se move)
m_sprite_state	sprite_state	SpriteState	estado do desenho do personagem
m_dx	axis_x_direction	double	coordenada x
m_dy	axis_y_direction	double	coordenada y
m_damage	damage	int	representa o dano
m_speed	speed	double	representa a velocidade
m_frame	frame	int	representa o quadro (imagem de animacao atual)
m_sprite_path	sprite_path	string	define a cor do personagem
m_start	start	unsigned	marca o tempo de inicio
m_current_time	current_time	unsigned	guarda o tempo atual
m_texture	texture	shared_ptr<Texture>	responsavel pelas texturas
m_bounding_box	bounding_box	Rectangle	responsavel pela caixa delimitadora
spear.cpp			
Classe Spear			
Construtor			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self			
Nome Antigo	Novo Nome	Tipo	Descrição
new_y	-	double	nova posicao no eixo y
new_x	-	double	nova posicao no eixo x
choose_sprite_path			
Nome Antigo	Novo Nome	Tipo	Descrição
directory	color	string	escolhe a cor de acordo com o numero do jogador
sprite_path	-	string	string q armazena o caminho do diretorio onde esta localizada a imagem
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo do Retorno	Descrição
active	is_active	bool	Retorna se o personagem está ativo.
bounding_box	get_bounding_box	Rectangle&	Retorna o "espaço" que esta o personagem
hit_boxes	hit_boxes	list<Rectangle>&	Retorna o "espaço" que o personagem colidiu.
on_collision	on_collision	void	Faz a interação da colisão
number_of_lives	get_number_of_lives	int	Retorna o numero de vidas do personagem
set_base	set_base	void	Seta qual será a base do personagem
id	get_id	int	retorna o id do personagem
direction	get_direction	pair<double, double>	Retorna a direção que o personagem está se mexendo
MÉTODOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	update_self	void	Realizar update no personagem
draw_self	draw_self	void	Desenha o personagem
change_character_state	change_character_state	void	Muda estado do personagem. Exemplo: estado morto

handle_state	handle_state	void	Realiza ação respectiva a seu estado
set_spawn_position	set_spawn_position	void	Faz o set da posição inicial, oque varia de acordo com o player
respawn_character	respawn_character	void	Faz o personagem renascer no jogo
kill_character	kill_character	void	Mata o personagem
choose_sprite_path	choose_sprite_path	string	Escolhe qual o caminho para o sprite do personagem
on_event	on_event	bool	Retorna quando ocorre um evento
heavy_attack	do_heavy_attack	void	Realiza o ataque forte
light_attack	do_light_attack	void	Realiza o ataque leve
defense	do_defense	void	Realiza habilidade de defesa
special	do_special	void	Realiza golpe especial

ATRIBUTOS PROTEGIDOS

Nome Antigo	Novo Nome	Tipo	Descrição
m_moving_state	moving_state	MovingState	Atributo para estado de movimento do personagem.
m_state	state	CharacterState*	Atributo para estado atual do personagem.
m_character_state_factory	character_state_factory	CharacterStateFactory	
m_active	active	bool	Atributo para dizer se o personagem está ativo ou não.
m_id	id	unsigned	ID do personagem.
m_max_life	max_life	int	Vida maxima do personagem.
m_current_life	current_life	int	Vida atual do personagem.
m_number_of_lives	number_of_lives	int	Numero de vidas do personagem.
m_frame	frame	int	
m_start	start	int	
m_w	width	int	Largura da imagem do personagem. Valor é dado apartir da divisão dos sprites respectivo.
m_h	height	int	Altura da imagem do personagem. Valor é dado apartir da divisão dos sprites respectivo.
m_heavy_attack_cooldown	heavy_attack_cooldown	int	Tempo de cooldown para ataques pesados do personagem.
m_light_attack_cooldown	light_attack_cooldown	int	Tempo de cooldown para ataques leves do personagem.
m_defense_cooldown	defense_cooldown	int	Tempo de cooldown para habilidade de defesa do personagem.
m_special_cooldown	special_cooldown	int	Tempo de cooldown para ataques especiais do personagem.
m_last_used_heavy_attack	last_used_heavy_attack	int	Ultimo vez ataque pesado foi usado.
m_last_used_light_attack	last_used_light_attack	int	Ultimo vez ataque leve foi usado.
m_last_used_defense	last_used_defense	int	Ultimo vez habilidade de defesa foi usada.
m_last_used_special	last_used_special	int	Ultimo vez ataque especial foi usado.
m_respawn_time	respawn_time	int	Tempo para o personagem renascer.
m_character_code	character_code	int	Código de personagem.
m_last_sound_played	last_sound_played	int	Id da ultima musica tocada.
m_freeze	freeze	bool	Boleano para se o personagem está "congelado".
m_dead	dead	bool	Boleano para se o personagem está morto.
m_x_speed	axis_x_speed	double	Velocidade de movimento no eixo X do personagem.
m_y_speed	axis_y_speed	double	Velocidade de movimento no eixo Y do personagem.
m_speed	speed	double	Velocidade de movimento do personagem.
m_base	base	Base*	Apontador para qual a base do personagem.
m_textures	textures	vector< shared_ptr<Texture> >	Textura usada para a imagem do personagem.
m_speed_vector	speed_vector	unordered_map<string, pair<double, double> >	
m_bounding_box	bounding_box	Rectangle	
m_sprite_paths	sprite_paths	vector<string>	Local do arquivo de sprite do personagem.

character.cpp

Constantes

Nome da Constante	Tipo	Valor	Descrição
BOUNDING_BOX_SIZE	int	24	Tamanho do "BOUNDING_BOX_SIZE" do personagem.

SPEED	double	80	Velocidade do personagem
CHARACTER_WIDTH	int	32	Width and Height of the character, considering the sprite size
RESPAWN_TIME	int	10000	Time to player respawn
Character			
Nome Antigo	Novo Nome	Tipo	Descrição
sprite_paths	sprite_paths	vector<string>	<i>Paths</i> para os arquivos de sprite do personagem
id	new_character_id	unsigned	id para novo personagem
x	position_axis_x	double	Posição inicial do personagem a ser criado, no eixo X
y	position_axis_y	double	Posição inicial do personagem a ser criado, no eixo Y
max_life	max_life	int	Maximo de vida do personagem
character_code	character_code	int	Código do personagem
~Character			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self			
Nome Antigo	Novo Nome	Tipo	Descrição
now	now	unsigned	estado atual do personagem
last	last	unsigned	estado atual do personagem
update_position			
Nome Antigo	Novo Nome	Tipo	Descrição
now	now	unsigned &	endereço do estado atual do personagem
last	last	unsigned &	endereço do estado atual do personagem
backwards	backwards	bool	booleano para voltar ou não o <i>update</i>
draw_self			
Nome Antigo	Novo Nome	Tipo	Descrição
canvas	canvas	Canvas *	endereço do estado atual do personagem
"sem nome"	now	unsigned	estado do personagem atual
"sem nome"	last	unsigned	proximo estado do personagem
on_event			
Nome Antigo	Novo Nome	Tipo	Descrição
event	event	GameEvent&	endereço do estado atual do personagem
change_character_state			
Nome Antigo	Novo Nome	Tipo	Descrição

next_state	next_state	State	proximo estado do personagem
respawning	respawning	bool	booleano para se deve ou não ir para o proximo estado
Skill			
atributos protegidos			
nome antigo	nome novo	tipo	descrição
character_id	character_id	unsigned	identidade do personagem
damage	damage	int	dano da skill
speed	skill_speed	double	velocidade da skil
m_collided	skill_collided	int	colisão da skill
metodos publicos			
nome antigo	nome novo	tipo	descrição
Skill	Skill	Skill	metodo construtor
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
parent	parent	GameObject *	ponteiro pra o GameObject
xp	axis_position_x	double	posição no eixo x
yp	axis_position_y	double	posição no eixo y
damage	new_skill_damage	int	dano da skill
character_id	character_id	int	identidade do personagem
nome antigo	nome novo	tipo	descrição
~Skill	~Skill	Skill	
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
active	is_active	bool	retorna se a skill está ativa
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
bounding_box()	get_bounding_box()	Rectangle&	quadrante de contato do personagem
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
hit_boxes()	get_hit_boxes()	list<Rectangle>&	pontos de contato do personagem
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
direction	get_direction	pair<double, double>	pega direção x,y
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
on_collision	on_collision	void	
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
who	who	Collidable *	quem esta realizando a skill
where	where	Rectangle&	a onde a skill será usada
now	now_moment	unsigned	momento atual
last	last_moment	unsigned	último momento que a skill foi usada
nome antigo	nome novo	tipo	descrição
character_id	get_character_id	unsigned	retorna id do personagem que realizou a skill
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição

nome antigo	nome novo	tipo	descrição
damage	get_damage	int	retorna o dano feito de skill
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
nome antigo	nome novo	tipo	descrição
collided	get_collided	int	retorna o ponto de colisão do personagem
nome parametro entrada antigo	nome parametro entrada novo	tipo	descrição
Class DeathMatchLevel			
CONSTANTES			
Nome Antigo	Novo Nome	Tipo	Descrição
MAX_W	MAX_WEIGHT		constante que define a largura máxima
MAX_H	MAX_HEIGHT		constante que define a altura máxima
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
done	get_level_done	bool	Indica que o nível foi finalizado
next	get_next_level	string	Indica o próximo nível
audio	get_level_audio	string	Indica o áudio do nível do jogo
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self	update_self	void	Indica atualização do level
draw_self	draw_self	void	Desenha o level
set_players_characters_position	set_players_characters_position	void	Seta a posição dos jogadores
verify_characters	verify_characters	void	Verifica personagens para definir o vencedor
MÉTODOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_done	level_done	bool	Indica que terminou
m_next	next_level	string	Indica o próximo nível
m_start	start_level	int	Indica que começou o level
m_has_winner	has_winner_level	bool	Indica se tem um jogador vencedor
m_characters	characters_level	vector <Character* >	Armazena os jogadores em um vetor
m_players_characters	players_characters_number	vector < int >	Armazena o número de jogadores
m_texture	texture_level	shared_ptr<Texture>	Define a imagem do level
m_map[MAX_W][MAX_H]	map_level	int	
m_character_factory	character_factory_level	CharacterFactory	
Class MainLevel			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
done		bool	

next		string	
audio		string	
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self		void	
draw_self		void	
set_players_characters_position		void	
verify_characters		void	
MÉTODOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_done		bool	
m_next		string	
m_start		int	
m_has_winner		bool	
m_bases		vector <Base* >	
m_players_characters		vector < int >	
m_texture		shared_ptr<Texture>	
m_map[MAX_W][MAX_H]		int	
m_character_factory		CharacterFactory	
Class MenuLevel			
MÉTODOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
done		bool	
next		string	
audio		string	
MÉTODOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição
update_self		void	
draw_self		void	
update_position		void	
on_event		bool	
MÉTODOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PUBLICOS			
Nome Antigo	Novo Nome	Tipo	Descrição
ATRIBUTOS PROTEGIDOS			
Nome Antigo	Novo Nome	Tipo	Descrição

ATRIBUTOS PRIVADOS			
Nome Antigo	Novo Nome	Tipo	Descrição
m_done		bool	
m_next		string	
m_start		int	
m_option_y_position		int	
m_current_option		int	
m_textures		vector< shared_ptr<Texture> >	
m_options		vector < string >	