Jan Of Empires

Generated by Doxygen 1.8.14

Contents

Chapter 1

Jan of Empires Deluxe Edition

Simple Strategy game

Ideia do jogo

É um jogo de estratégia em turnos onde você controla 3 unidades para explorar, adquirir recursos, aumentar o poder da sua unidade usando os pilares para vencer o oponente. Cada tipo de unidade tem vantagem sobre outra unidade. O jogo é single-player com 1 CPU de oponente.

Elementos do Jogo

Recursos

Os recursos servem para criar pilares novos, criar necromancers (a partir de pilares) ou aumentar o poder de necromancers

Dois tipos de recursos:

- Metal: para criação de pilares. Menos abundante no mapa (terá o suficiente para criar mais 2 pilares para cada jogador).
- Ossos: para aumento de poder das unidades. Mais abundante no mapa.

Prédios (Pilares)

Os prédios são pilares. Cada pilar representa uma fábrica de um tipo de unidade que será criada a partir da quantidade de ossos. Cada pilar cria uma única unidade, que é o necromancer. Depois de criado o necromancer, o pilar pode aumentar o poder do necromancer utilizando ossos.

Cada pilar tem:

- HP: representa a vida do pilar. Se chega a zero, ele é destruído.

É possível reconstruir pilares, com o máximo de 3 por jogador.

Três tipos de pilares:

- Pilar da Espada: é o pilar que cria o necromancer guerreiro.
- Pilar da Lança: é o pilar que cria o necromancer cavaleiro.
- Pilar do Arco : é o pilar que cria o necromancer arqueiro.

Unidades (Necromancers)

As unidades são representadas por um único elemento, o Necromancer. Cada necromancer tem um número associado que define quanto de poder ele tem. Por exemplo: no mapa há uma unidade A com o número 20. Isso significa que a unidade A tem 20 de poder.

Cada unidade tem:

- MP: representa quanto de poder a unidade tem e representa a vida da unidade. Se ela chegar a zero, a unidade morre.

São 3 tipos de unidades:

- Necromancer Guerreiro (A): é o necromancer que invoca somente undeads guerreiros. A quantidade de undeads é o que define o poder do necromancer (é o número que mostra a força da unidade). Ele tem vantagem sobre o Necromancer Cavaleiro e desvantagem sobre o Necromancer Arqueiro.
- Necromancer Cavaleiro (B): é o necromancer que invoca somente undeads cavaleiros. A quantidade de undeads é o que define o poder do necromancer (é o número que mostra a força da unidade). Ele tem vantagem sobre o Necromancer Arqueiro e desvantagem sobre o Necromancer Guerreiro.
- Necromancer Arqueiro (C): é o necromancer que invoca somente undeads arqueiros. EA quantidade de undeads é o que define o poder do necromancer (é o número que mostra a força da unidade). Ele tem vantagem sobre o Necromancer Guerreiro e desvantagem sobre o Necromancer Cavaleiro.

O máximo de unidades no jogo são 6.

Regras definidas para o jogo

Inicialização

Inicialmente cada jogardor possui:

- 1 Pilar da Espada
- 1 Necromancer guerreiro

Turno

Em UM turno o jogador pode criar pilares, fortalecer unidades e mover apenas uma delas. Ele acaba quando o jogador move uma peça ou decide terminar o turno.

Captação de recursos

Os recursos são espalhados randômicamente no mapa no início do jogo. São capturados quando o jogador move uma unidade para o mesmo bloco em que o recurso está inserido.

Combate

O combate acontece automaticamente quando duas unidades estão vizinhas O combate acontece sempre entre apenas duas unidades No combate, verificam-se os poderes de cada unidade, e as capacidades e fraquezas de cada uma em relação a outra, de acordo com os seus tipos. Então, é decrescido poder de cada uma de acordo com o estabelecido.

Condições de término do jogo

As condições para o término do jogo são os seguintes:

- Se todos os pilares do oponente forem destruídos;
- Se todos os recursos do mapa acabarem: nesse caso, a quantidade da vida dos pilares + unidades é somado. Quem tiver mais, vence.

Useful links

- `SDL Tutorial link`
- `Bibliotecas de SDL`

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

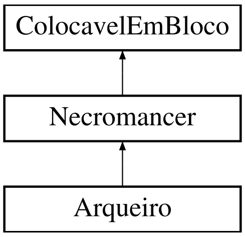
Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 Arqueiro Class Reference

Inheritance diagram for Arqueiro:



Public Member Functions

- Arqueiro ()

  Construct a new Arqueiro:: Arqueiro object.

Additional Inherited Members

5.1.1 Detailed Description

Definition at line 70 of file Necromancer.hpp.

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

- include/Necromancer.hpp
- src/Necromancer.cpp

5.2 Bloco Class Reference

Public Member Functions

- Bloco ()

  Construct a new Bloco:: Bloco object.
- Bloco (ColocavelEmBloco ∗)

  Construct a new Bloco:: Bloco object.
- bool preenche (ColocavelEmBloco ∗)
- bool limpa ()

Public Attributes

- bool **vazio**
- ColocavelEmBloco ∗ **conteudo**

5.2.1 Detailed Description

Definition at line 38 of file Bloco.hpp.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Bloco()

```
Bloco::Bloco (
            ColocavelEmBloco * conteudo )
```

Construct a new Bloco:: Bloco object.

Parameters

conteudo	

Definition at line 32 of file Bloco.cpp.

```
32                                    {
33     this->conteudo = conteudo;
34     this->vazio = false;
35 }
```

5.2.3 Member Function Documentation

5.2.3.1 limpa()

```
bool Bloco::limpa ( )
```

Returns

true
false

Definition at line 57 of file Bloco.cpp.

```
57                    {
58     if (this->vazio) return false;
59     this->conteudo = nullptr;
60     this->vazio = true;
61     return true;
62 }
```

5.2.3.2 preenche()

```
bool Bloco::preenche (
               ColocavelEmBloco * c )
```

Parameters

c	

Returns

true
false

Definition at line 44 of file Bloco.cpp.

```
44                                          {
45     if (!this->vazio) return false;
46     this->conteudo = c;
47     this->vazio = false;
48     return true;
49 }
```

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

- include/Bloco.hpp
- src/Bloco.cpp

5.3 Button Class Reference

Public Member Functions

- Button ()

  Construct a new Button:: Button object.
- void setPositionSizeType (int x, int y, int width, int height, ButtonType type)
- void handleEvent (SDL_Event ∗e, Game ∗game)
- void setPositionX (int x)
- void setPositionY (int y)
- void setGeneralButtonWidth (int width)
- void setGeneralButtonHeight (int height)
- int getPositionX ()
- int getPositionY ()
- int getGeneralButtonWidth ()
- int getGeneralButtonHeight ()
- void setButtonType (ButtonType type)
- ButtonType getButtonType ()

5.3.1 Detailed Description

Definition at line 48 of file Button.hpp.

5.3.2 Member Function Documentation

5.3.2.1 getButtonType()

```
ButtonType Button::getButtonType ( )
```

Returns

ButtonType

Definition at line 175 of file Button.cpp.

```
175                                  {
176     return buttonType;
177 }
```

5.3.2.2 getGeneralButtonHeight()

```
int Button::getGeneralButtonHeight ( )
```

Returns

int

Definition at line 157 of file Button.cpp.

```
157                                     {
158     return button_height;
159 }
```

5.3.2.3 getGeneralButtonWidth()

```
int Button::getGeneralButtonWidth ( )
```

Returns

int

Definition at line 148 of file Button.cpp.

```
148                                    {
149     return button_width;
150 }
```

5.3.2.4 getPositionX()

```
int Button::getPositionX ( )
```

Returns

int

Definition at line 130 of file Button.cpp.

```
130                           {
131     return position_x;
132 }
```

5.3.2.5 getPositionY()

```
int Button::getPositionY ( )
```

Returns

int

Definition at line 139 of file Button.cpp.

```
139                          {
140     return position_y;
141 }
```

5.3.2.6 handleEvent()

```
void Button::handleEvent (
              SDL_Event * e,
              Game * game )
```

Parameters

e	
game	

Definition at line 51 of file Button.cpp.

```
51                                                    {
52     if (e->type == SDL_MOUSEBUTTONDOWN) {
53         // Get mouse position
54         int x, y;
55         SDL_GetMouseState(&x, &y);
56         // Check if mouse is in Button
57         bool inside = false;
58         if (y > position_y && y < (position_y + button_height)) {
59             if (x > position_x && x < (position_x + button_width)) {
60                 inside = true;
61             }
62         }
63 
64         if (inside) {
65             switch (e->type) {
66                 case SDL_MOUSEBUTTONDOWN:
67                     switch (buttonType) {
68                     case BUTTON_PLAY:
69                         game->setGameRunning(GAME_PLAY);
70                         break;
71                     case BUTTON_CREDITS:
72                         game->setGameRunning(GAME_CREDITS);
73                         break;
74                     case BUTTON_QUIT:
75                         game->setGameRunning(GAME_QUIT);
76                         break;
77                     case BUTTON_BACK_CREDITS:
78                         game->setGameRunning(GAME_MENU);
79                         break;
80                     case BUTTON_BACK_GAME:
81                         game->setGameRunning(GAME_PLAY);
82                     }
83                 break;
84             }
85         }
86     }
87 }
```

5.3.2.7 setButtonType()

```
void Button::setButtonType (
             ButtonType type )
```

Parameters

type	

Definition at line 166 of file Button.cpp.

```
166                                         {
167     buttonType = type;
168 }
```

5.3.2.8 setGeneralButtonHeight()

```
void Button::setGeneralButtonHeight (
             int height )
```

Parameters

height	

Definition at line 121 of file Button.cpp.

```
121                                             {
122     button_height = height;
123 }
```

5.3.2.9 setGeneralButtonWidth()

```
void Button::setGeneralButtonWidth (
             int width )
```

Parameters

width	

Definition at line 112 of file Button.cpp.

```
112                                           {
```

```
113     button_width = width;
114 }
```

5.3.2.10 setPositionSizeType()

```
void Button::setPositionSizeType (
             int x,
             int y,
             int width,
             int height,
             ButtonType type )
```

Parameters

x	
y	
width	
height	
type	

Definition at line 37 of file Button.cpp.

```
37                                                                                     {
38     position_x = x;
39     position_y = y;
40     button_width = width;
41     button_height = height;
42     buttonType = type;
43 }
```

5.3.2.11 setPositionX()

```
void Button::setPositionX (
             int x )
```

Parameters

x	

Definition at line 94 of file Button.cpp.

```
94                                {
95     position_x = x;
96 }
```

5.3.2.12 setPositionY()

```
void Button::setPositionY (
             int y )
```

Parameters

y	

Definition at line 103 of file Button.cpp.

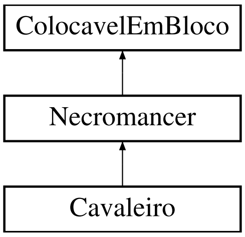
```
103                              {
104     position_y = y;
105 }
```

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

- include/Button.hpp
- src/Button.cpp

5.4 Cavaleiro Class Reference

Inheritance diagram for Cavaleiro:



Public Member Functions

- Cavaleiro ()

  Construct a new Cavaleiro:: Cavaleiro object.

Additional Inherited Members

5.4.1 Detailed Description

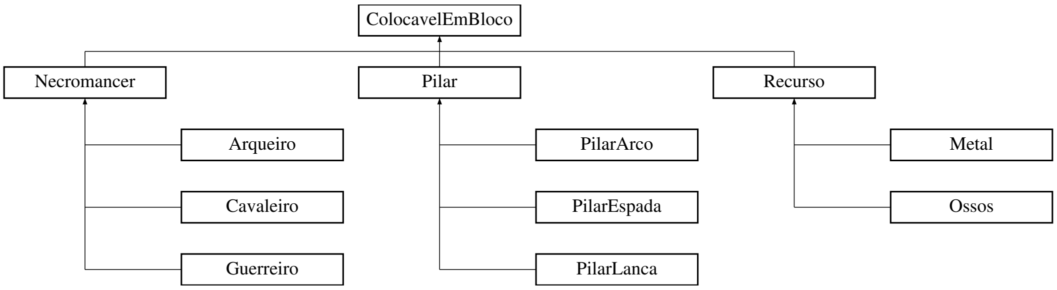
Definition at line 63 of file Necromancer.hpp.

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

- include/Necromancer.hpp
- src/Necromancer.cpp

5.5 ColocavelEmBloco Class Reference

Inheritance diagram for ColocavelEmBloco:



Public Member Functions

- ColocavelEmBloco ()

  Construct a new Colocavel Em Bloco:: Colocavel Em Bloco object.
- bool mata ()
- bool revive ()
- void setAtivo (bool a)

Public Attributes

- bool **vivo**
- unsigned short **x**
- unsigned short **y**
- unsigned short **time**
- TipoConteudoBloco **tipo**
- bool **ativo**

5.5.1 Detailed Description

Definition at line 26 of file Bloco.hpp.

5.5.2 Member Function Documentation

5.5.2.1 mata()

```
bool ColocavelEmBloco::mata ( )
```

Returns

true
false

Definition at line 87 of file Bloco.cpp.

Referenced by Player::Player().

```
87                            {
88     if (this->vivo) {
89         this->vivo = false;
90         this->x = -1;
91         this->y = -1;
92         return true;
93     }
94     return false;
95 }
```

5.5.2.2 revive()

```
bool ColocavelEmBloco::revive ( )
```

Returns

true
false

Definition at line 103 of file Bloco.cpp.

```
103                              {
104     if (this->vivo)
105         return false;
106 
107     this->vivo = true;
108     return true;
109 }
```

5.5.2.3 setAtivo()

```
void ColocavelEmBloco::setAtivo (
              bool a )
```

Parameters

a	

Definition at line 77 of file Bloco.cpp.

Referenced by Controlador::novo_jogo().

```
77                                  {
78     ativo = a;
79 }
```

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

- include/Bloco.hpp
- src/Bloco.cpp

5.6 Controlador Class Reference

Public Member Functions

- Controlador ()

  Construct a new Controlador:: Controlador object.
- bool novo_jogo (bool, bool)
- bool **salvar_jogo** (std::string path)
- bool **carregar_jogo** (std::string path)
- bool criar_pilar (Player ∗, TipoPilar, unsigned short, unsigned short)
- bool fortalecer_pilar (Player ∗, TipoPilar)
- bool criar_necromancer (Player ∗, TipoNecromancer, unsigned short, unsigned short)
- bool fortalecer_necromancer (Player ∗, TipoNecromancer)
- bool matar (unsigned short, unsigned short)
- bool pode_movimentar (Player ∗, unsigned short, unsigned short, unsigned short, unsigned short)
- bool movimentar (Player ∗, unsigned short, unsigned short, unsigned short, unsigned short)
- void **processa_jogada** ()
- bool alguem_ganhou ()
- void **muda_vez** ()
- void verifica_combate (unsigned short, unsigned short)
- void realiza_combate (unsigned short, unsigned short, unsigned short, unsigned short)
- bool gerou_combate (unsigned short, unsigned short, unsigned short)
- void **print_recursos** ()
- void **print_mapa** ()

Public Attributes

- Mapa **mapa**
- Player **jogador**
- Player **computador**
- Posicao **cursor**
- std::list< Recurso > **recursos**
- bool **jogo_terminou**
- int **ganhou**
- bool **computador_joga**
- unsigned short **vez**
- int **vezes**

5.6.1 Detailed Description

Definition at line 26 of file Controlador.hpp.

5.6.2 Member Function Documentation

5.6.2.1 alguem_ganhou()

```
bool Controlador::alguem_ganhou ( )
```

Returns

true
false

Definition at line 332 of file Controlador.cpp.

References Player::perdeu_jogo(), and Player::pontuacao().

```
332                                {
333     if (this->recursos.size() == 0) {
334         if (this->jogador.pontuacao() > this->computador.pontuacao()) {
335             this->jogo_terminou = true;
336             this->ganhou = 0;
337             return true;
338         }
339         if (this->jogador.pontuacao() < this->computador.pontuacao()) {
340             this->jogo_terminou = true;
341             this->ganhou = 1;
342             return true;
343         }
344 
345         this->jogo_terminou = true;
346         this->ganhou = -1;
347         return true;
348     }
349 
350     if (this->jogador.perdeu_jogo()) {
351         this->jogo_terminou = true;
352         this->ganhou = 1;
353         return true;
354     }
355     if (this->computador.perdeu_jogo()) {
356         this->jogo_terminou = true;
357         this->ganhou = 0;
358         return true;
359     }
360 
361     return false;
362 }
```

5.6.2.2 criar_necromancer()

```
bool Controlador::criar_necromancer (
              Player * jog,
              TipoNecromancer nec,
              unsigned short x,
              unsigned short y )
```

Parameters

jog	
nec	
x	
y	

Returns

true
false

Definition at line 113 of file Controlador.cpp.

References Player::criar_necromancer(), Mapa::inserir(), Player::necromancer(), and Mapa::vazio().

```
113                                                                                                  {
114     if (jog->necromancer(nec)->vivo) return false;
115     if (!this->mapa.vazio(x, y)) return false;
116     if (!jog->criar_necromancer(nec)) return false;
117
118     this->mapa.inserir(jog->necromancer(nec), x, y);
119
120     return true;
121 }
```

5.6.2.3 criar_pilar()

```
bool Controlador::criar_pilar (
             Player * jog,
             TipoPilar pil,
             unsigned short x,
             unsigned short y )
```

Parameters

jog	
pil	
x	
y	

Returns

true
false

Definition at line 94 of file Controlador.cpp.

References Player::criar_pilar(), Mapa::inserir(), Player::pilar(), and Mapa::vazio().

```
94                                                                                      {
95     if (jog->pilar(pil)->vivo) return false;
96     if (!this->mapa.vazio(x, y)) return false;
97     if (!jog->criar_pilar(pil)) return false;
98     this->mapa.inserir(jog->pilar(pil), x, y);
99
100     return true;
101 }
```

5.6.2.4 fortalecer_necromancer()

```
bool Controlador::fortalecer_necromancer (
            Player * jog,
            TipoNecromancer nec )
```

Parameters

jog	
nec	

Returns

true
false

Definition at line 145 of file Controlador.cpp.

References Player::criar_necromancer(), and Player::necromancer().

```
145                                                                      {
146     if (!jog->necromancer(nec)->vivo) return false;
147
148     return jog->criar_necromancer(nec);
149 }
```

5.6.2.5 fortalecer_pilar()

```
bool Controlador::fortalecer_pilar (
            Player * jog,
            TipoPilar pil )
```

Parameters

jog	
pil	

Returns

true
false

Definition at line 131 of file Controlador.cpp.

References Player::criar_pilar(), and Player::pilar().

```
131                                                          {
132     if (!jog->pilar(pil)->vivo) return false;
133 
134     return jog->criar_pilar(pil);
135 }
```

5.6.2.6 gerou_combate()

```
bool Controlador::gerou_combate (
              unsigned short time,
              unsigned short x,
              unsigned short y )
```

Parameters

time	
x	
y	

Returns

true
false

Definition at line 243 of file Controlador.cpp.

References Mapa::posicao_valida(), Mapa::vazio(), and Mapa::ver().

Referenced by verifica_combate().

```
243                                                                                {
244     if (!this->mapa.posicao_valida(x, y))
245         return false;
246     if (this->mapa.vazio(x, y))
247         return false;
248     if (this->mapa.ver(x, y)->tipo == TipoConteudoBloco::RECURSO)
249         return false;
250 
251     return this->mapa.ver(x, y)->time != time;
252 }
```

5.6.2.7 matar()

```
bool Controlador::matar (
              unsigned short X,
              unsigned short Y )
```

Parameters

X	
Y	

Returns

true
false

Definition at line 159 of file Controlador.cpp.

References Mapa::vazio(), and Mapa::ver().

```
159                                                           {
160     if (this->mapa.vazio(X, Y))
161         return false;
162     if (this->mapa.ver(X, Y)->tipo == TipoConteudoBloco::RECURSO)
163         return false;
164     ColocavelEmBloco * vitima = this->mapa.retirar(X, Y);
165     vitima->mata();
166     return true;
167 }
```

5.6.2.8 movimentar()

```
bool Controlador::movimentar (
               Player * jog,
               unsigned short x_orig,
               unsigned short y_orig,
               unsigned short x_dest,
               unsigned short y_dest )
```

Parameters

jog	
x_orig	
y_orig	
x_dest	
y_dest	

Returns

true
false

Definition at line 212 of file Controlador.cpp.

References pode_movimentar(), Mapa::retirar(), Mapa::vazio(), and Mapa::ver().

```
                           {
    if ( !this->pode_movimentar(jog, x_orig, y_orig, x_dest, y_dest) )
        return false;
    ColocavelEmBloco * unidade_movida = this->mapa.retirar(x_orig, y_orig);
    if (!this->mapa.vazio(x_dest, y_dest)) {
        if (this->mapa.ver(x_dest, y_dest)->tipo == TipoConteudoBloco::RECURSO) {
            Recurso * rec = ((Recurso *) mapa.retirar(x_dest, y_dest));
            jog->captar_recurso(rec->tipo_recurso);
            for (std::list<Recurso>::iterator i = this->recursos.begin(); i != this->recursos.end(); ++i)
                if (i->x == x_dest && i->y == y_dest) {
// std::cout << "recurso retirado de  " <<i->x << " " << i->y << std::endl;
                    this->recursos.erase(i);
                    break;
                }
        }
    }
    this->mapa.inserir(unidade_movida, x_dest, y_dest);

    this->processa_jogada();
    return true;
}
```

5.6.2.9 novo_jogo()

```
bool Controlador::novo_jogo (
              bool recursos_aleatorios,
              bool computador_joga )
```

Parameters

recursos_aleatorios	
computador_joga	

Returns

true
false

Definition at line 59 of file Controlador.cpp.

References Mapa::inserir(), and ColocavelEmBloco::setAtivo().

```
59                                                                           {
60     this->mapa.inserir(&jogador.guerreiro, X_NECROMANCER_PLAYER, Y_NECROMANCER_PLAYER);
61     this->mapa.inserir(&jogador.pilar_espada, X_PILAR_PLAYER, Y_PILAR_PLAYER);
62     this->mapa.inserir(&computador.guerreiro, X_NECROMANCER_COMPUTADOR, Y_NECROMANCER_COMPUTADOR);
63     this->mapa.inserir(&computador.pilar_espada, X_PILAR_COMPUTADOR, Y_PILAR_COMPUTADOR);
64 
65     jogador.guerreiro.setAtivo(true);
66     computador.guerreiro.setAtivo(true);
67 
68     #ifdef PROD
69     ativo_x_jog = X_NECROMANCER_PLAYER;
70     ativo_y_jog = Y_NECROMANCER_PLAYER;
71     ativo_x_cpu = X_NECROMANCER_COMPUTADOR;
72     ativo_y_cpu = Y_NECROMANCER_COMPUTADOR;
73     #endif
74 
75     if (recursos_aleatorios)
76         this->preenche_recursos_iniciais();
77 
78     this->computador_joga = computador_joga;
79 
80     this->computador.muda_time();
81     return true;
82 }
```

5.6.2.10 pode_movimentar()

```
bool Controlador::pode_movimentar (
             Player * jog,
             unsigned short x_orig,
             unsigned short y_orig,
             unsigned short x_dest,
             unsigned short y_dest )
```

Parameters

jog	
x_orig	
y_orig	
x_dest	
y_dest	

Returns

true
false

Definition at line 180 of file Controlador.cpp.

References Mapa::posicao_valida(), Mapa::vazio(), and Mapa::ver().

Referenced by movimentar().

```
180                                   {
181     if (!(this->mapa.posicao_valida(x_orig, y_orig) && this->mapa.
      posicao_valida(x_dest, y_dest)))
182         return false;
183     if (abs(x_dest - x_orig) > RANGE_MOVIMENTO || abs(y_dest - y_orig) > RANGE_MOVIMENTO)
184         return false;
185     if (this->mapa.vazio(x_orig, y_orig))
186         return false;
187     if (this->mapa.ver(x_orig, y_orig)->tipo != TipoConteudoBloco::UNIDADE)
188         return false;
189     if (this->mapa.ver(x_orig, y_orig)->time != jog->time)
190         return false;
191     if (this->mapa.vazio(x_dest, y_dest))
192         return true;
193     if (this->mapa.ver(x_dest, y_dest)->tipo == TipoConteudoBloco::UNIDADE)
194         return false;
195     if (this->mapa.ver(x_dest, y_dest)->tipo == TipoConteudoBloco::PREDIO)
196         return false;
197 
198     return true;
199 }
```

5.6.2.11 realiza_combate()

```
void Controlador::realiza_combate (
             unsigned short x_atac,
             unsigned short y_atac,
             unsigned short x_vit,
             unsigned short y_vit )
```

Parameters

x_atac	
y_atac	
x_vit	
y_vit	

Definition at line 262 of file Controlador.cpp.

References Mapa::ver().

Referenced by verifica_combate().

```
262 {
263     unsigned short dano_golpe;
264     std::cout<<  "realizando combate em: "<< x_atac << " " << y_atac << " "<< x_vit << " " << y_vit << std::endl;
265     Necromancer *atacante = (Necromancer *)this->mapa.ver(x_atac, y_atac);
266 
267     ColocavelEmBloco *vitima = this->mapa.ver(x_vit, y_vit);
268     TipoNecromancer tipo_vitima_nec = ((Necromancer *) vitima)->tipo_necromancer;
269     TipoPilar tipo_vitima_pil = ((Pilar *) vitima)->tipo_pilar;
270     switch ((int) this->mapa.ver(x_vit, y_vit)->tipo) {
271         case (int) TipoConteudoBloco::UNIDADE:
272             dano_golpe = DANO_DE_ATQ * atacante->multiplicador(tipo_vitima_nec);
273             if (((Necromancer *) vitima)->mp <= dano_golpe) {
274                 this->matar(x_vit, y_vit);
275                 ((Necromancer *) vitima)->mp = 0;
276             } else {
277                 ((Necromancer *) vitima)->mp = ((Necromancer *) vitima)->mp - dano_golpe;
278             }
279             break;
280         case (int) TipoConteudoBloco::PREDIO:
281             dano_golpe = DANO_DE_ATQ * atacante->multiplicador(tipo_vitima_pil);
282             if (((Pilar *) vitima)->hp <= dano_golpe) {
283                 this->matar(x_vit, y_vit);
284             } else {
285                 ((Pilar *) vitima)->hp = ((Pilar *) vitima)->hp - dano_golpe;
286             }
287             break;
288         default:
289             break;
290     }
291     return;
292 }
```

5.6.2.12 verifica_combate()

```
void Controlador::verifica_combate (
            unsigned short x,
            unsigned short y )
```

Parameters

x	
y	

Definition at line 300 of file Controlador.cpp.

References gerou_combate(), realiza_combate(), and Mapa::ver().

```
                                                                     {
        // std::cout<<  "verificando combate em: "<< x << " " << y << std::endl;
        unsigned short time = this->mapa.ver(x, y)->time;
        // Só é combate se tiver na vez do atacante
        if (time != this->vez)
            return;
        // procura adversarios vizinhos e realiza combate se tiver
        for (int i=x-RANGE_COMBATE; i <= x+RANGE_COMBATE; i++)
            for (int j = y-RANGE_COMBATE; j <= y+RANGE_COMBATE; j++)
                if (this->gerou_combate(time, i, j))
                    this->realiza_combate(x, y, i, j);
    }
```

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

- include/Controlador.hpp
- src/Controlador.cpp

5.7 Graphics Class Reference

Public Member Functions

- bool **init** ()
- bool **loadMedia** ()
- void **close** ()

5.7.1 Detailed Description

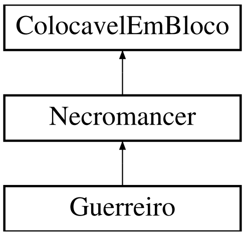
Definition at line 26 of file Graphics.hpp.

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

- include/Graphics.hpp
- src/Graphics.cpp

5.8 Guerreiro Class Reference

Inheritance diagram for Guerreiro:



Public Member Functions

- Guerreiro ()

    Construct a new Guerreiro:: Guerreiro object.

Additional Inherited Members

5.8.1 Detailed Description

Definition at line 56 of file Necromancer.hpp.

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

- include/Necromancer.hpp
- src/Necromancer.cpp

5.9 Mapa Class Reference

Public Member Functions

- Mapa (unsigned short, unsigned short)

    Construct a new Mapa:: Mapa object.
- bool posicao_valida (unsigned short, unsigned short)
- bool vazio (unsigned short, unsigned short)
- bool inserir (ColocavelEmBloco ∗, unsigned short, unsigned short)
- ColocavelEmBloco ∗ ver (unsigned short, unsigned short)
- ColocavelEmBloco ∗ retirar (unsigned short, unsigned short)

Public Attributes

- Posicao **cursor**
- MapaDeBlocos **mapa**

5.9.1 Detailed Description

Definition at line 25 of file Mapa.hpp.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Mapa()

```
Mapa::Mapa (
            unsigned short X,
            unsigned short Y )
```

Construct a new Mapa:: Mapa object.

Parameters

X	
Y	

Definition at line 24 of file Mapa.cpp.

```
24                                              {
25     this->tam_x = X;
26     this->tam_y = Y;
27     int i, j;
28     for (i = 0; i < X; i++)
29         for (j = 0; j < Y; j++)
30             this->mapa[std::make_pair(i, j)] = Bloco();
31 }
```

5.9.3 Member Function Documentation

5.9.3.1 inserir()

```
bool Mapa::inserir (
              ColocavelEmBloco * item,
              unsigned short X,
              unsigned short Y )
```

Parameters

item	
X	
Y	

Returns

true
false

Definition at line 68 of file Mapa.cpp.

References vazio().

Referenced by Controlador::criar_necromancer(), Controlador::criar_pilar(), and Controlador::novo_jogo().

```
68                                                                              {
69     if (!this->vazio(X, Y)) return false;
70 
71     this->mapa[std::make_pair(X, Y)].preenche(item);
72     item->x = X;
73     item->y = Y;
74     return true;
75 }
```

5.9.3.2 posicao_valida()

```
bool Mapa::posicao_valida (
               unsigned short X,
               unsigned short Y )
```

Parameters

X	
Y	

Returns

true
false

Definition at line 41 of file Mapa.cpp.

Referenced by Controlador::gerou_combate(), Controlador::pode_movimentar(), retirar(), vazio(), and ver().

```
41                                                        {
42     return X < this->tam_x && Y < this->tam_y;
43 }
```

5.9.3.3 retirar()

```
ColocavelEmBloco * Mapa::retirar (
               unsigned short X,
               unsigned short Y )
```

Parameters

X	
Y	

Returns

ColocavelEmBloco∗

Definition at line 98 of file Mapa.cpp.

References posicao_valida(), and vazio().

Referenced by Controlador::movimentar().

```
98                                                              {
99     if (this->vazio(X, Y) || !this->posicao_valida(X, Y)) return nullptr;
100     ColocavelEmBloco *conteudo = this->mapa[std::make_pair(X, Y)].conteudo;
101     this->mapa[std::make_pair(X, Y)].limpa();
102 
103     return conteudo;
104 }
```

5.9.3.4 vazio()

```
bool Mapa::vazio (
            unsigned short X,
            unsigned short Y )
```

Parameters

X	
Y	

Returns

true
false

Definition at line 53 of file Mapa.cpp.

References posicao_valida().

Referenced by Controlador::criar_necromancer(), Controlador::criar_pilar(), Controlador::gerou_combate(), inserir(), Controlador::matar(), Controlador::movimentar(), Controlador::pode_movimentar(), retirar(), and ver().

```
                                                {
    if (!this->posicao_valida(X, Y)) return false;

    return this->mapa[std::make_pair(X, Y)].vazio;
}
```

5.9.3.5 ver()

```
ColocavelEmBloco * Mapa::ver (
            unsigned short X,
            unsigned short Y )
```

Parameters

X	
Y	

Returns

ColocavelEmBloco∗

Definition at line 84 of file Mapa.cpp.

References posicao_valida(), and vazio().

Referenced by Controlador::gerou_combate(), Controlador::matar(), Controlador::movimentar(), Controlador::pode_movimentar(), Controlador::realiza_combate(), and Controlador::verifica_combate().

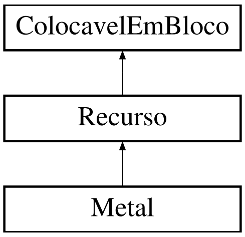
```
                                                           {
    if (this->vazio(X, Y) || !this->posicao_valida(X, Y)) return nullptr;
    ColocavelEmBloco *conteudo = this->mapa[std::make_pair(X, Y)].conteudo;

    return conteudo;
}
```

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

- include/Mapa.hpp
- src/Mapa.cpp

5.10 Metal Class Reference

Inheritance diagram for Metal:



Public Member Functions

- Metal ()

  Construct a new Metal:: Metal object.

Additional Inherited Members

5.10.1 Detailed Description

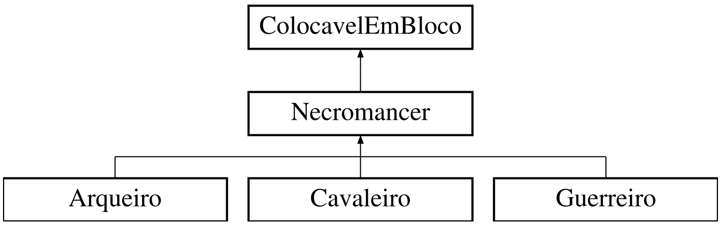
Definition at line 37 of file Recurso.hpp.

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

- include/Recurso.hpp
- src/Recurso.cpp

5.11 Necromancer Class Reference

Inheritance diagram for Necromancer:



Public Member Functions

- Necromancer ()

  Construct a new Necromancer:: Necromancer object.
- unsigned short multiplicador (TipoPilar tipo)
- unsigned short multiplicador (TipoNecromancer tipo)
- bool **handleEvent** (SDL_Event ∗e, int x, int y)

Public Attributes

- unsigned short **mp**
- TipoNecromancer **tipo_necromancer**

5.11.1 Detailed Description

Definition at line 39 of file Necromancer.hpp.

5.11.2 Member Function Documentation

5.11.2.1 multiplicador() [1/2]

```
unsigned short Necromancer::multiplicador (
              TipoPilar tipo )
```

Parameters

tipo	

Returns

unsigned short

Definition at line 97 of file Necromancer.cpp.

```
                                                      {
  if (this->tipo_necromancer == TipoNecromancer::ARQUEIRO) {
      if (tipo == TipoPilar::ESPADA)
           return ARCO_ESPADA_ATQ_MULTIPLICADOR;

       if (tipo == TipoPilar::LANCA)
           return ARCO_LANCA_ATQ_MULTIPLICADOR;

       if (tipo == TipoPilar::ARCO)
           return ARCO_ARCO_ATQ_MULTIPLICADOR;
   }
   if (this->tipo_necromancer == TipoNecromancer::CAVALEIRO) {
       if (tipo == TipoPilar::ESPADA)
           return LANCA_ESPADA_ATQ_MULTIPLICADOR;

       if (tipo == TipoPilar::LANCA)
           return LANCA_LANCA_ATQ_MULTIPLICADOR;
```

```
114
115         if (tipo == TipoPilar::ARCO)
116             return LANCA_ARCO_ATQ_MULTIPLICADOR;
117     }
118     if (this->tipo_necromancer == TipoNecromancer::GUERREIRO) {
119         if (tipo == TipoPilar::ESPADA)
120             return ESPADA_ESPADA_ATQ_MULTIPLICADOR;
121
122         if (tipo == TipoPilar::LANCA)
123             return ESPADA_LANCA_ATQ_MULTIPLICADOR;
124
125         if (tipo == TipoPilar::ARCO)
126             return ESPADA_ARCO_ATQ_MULTIPLICADOR;
127     }
128     return 0;
129 }
```

5.11.2.2 multiplicador() [2/2]

```
unsigned short Necromancer::multiplicador (
              TipoNecromancer tipo )
```

Parameters

tipo	

Returns

unsigned short

Definition at line 137 of file Necromancer.cpp.

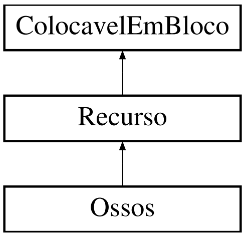
```
137                                                               {
138     if (this->tipo_necromancer == TipoNecromancer::ARQUEIRO) {
139         if (tipo == TipoNecromancer::GUERREIRO)
140             return ARCO_ESPADA_ATQ_MULTIPLICADOR;
141
142         if (tipo == TipoNecromancer::CAVALEIRO)
143             return ARCO_LANCA_ATQ_MULTIPLICADOR;
144
145         if (tipo == TipoNecromancer::ARQUEIRO)
146             return ARCO_ARCO_ATQ_MULTIPLICADOR;
147     }
148     if (this->tipo_necromancer == TipoNecromancer::CAVALEIRO) {
149         if (tipo == TipoNecromancer::GUERREIRO)
150             return LANCA_ESPADA_ATQ_MULTIPLICADOR;
151
152         if (tipo == TipoNecromancer::CAVALEIRO)
153             return LANCA_LANCA_ATQ_MULTIPLICADOR;
154
155         if (tipo == TipoNecromancer::ARQUEIRO)
156             return LANCA_ARCO_ATQ_MULTIPLICADOR;
157     }
158     if (this->tipo_necromancer == TipoNecromancer::GUERREIRO) {
159         if (tipo == TipoNecromancer::GUERREIRO)
160             return ESPADA_ESPADA_ATQ_MULTIPLICADOR;
161
162         if (tipo == TipoNecromancer::CAVALEIRO)
163             return ESPADA_LANCA_ATQ_MULTIPLICADOR;
164
165         if (tipo == TipoNecromancer::ARQUEIRO)
166             return ESPADA_ARCO_ATQ_MULTIPLICADOR;
167     }
168     return 0;
169 }
```

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

- include/Necromancer.hpp
- src/Necromancer.cpp

5.12 Ossos Class Reference

Inheritance diagram for Ossos:



Public Member Functions

- Ossos ()

  Construct a new Ossos:: Ossos object.

Additional Inherited Members

5.12.1 Detailed Description

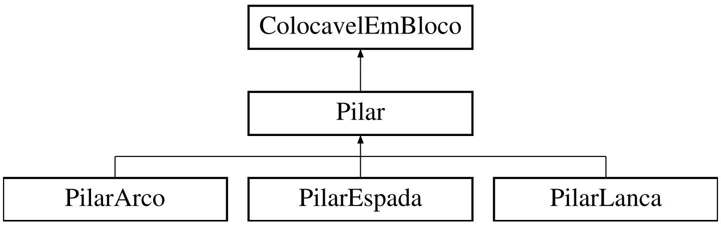
Definition at line 44 of file Recurso.hpp.

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

- include/Recurso.hpp
- src/Recurso.cpp

5.13 Pilar Class Reference

Inheritance diagram for Pilar:



Public Member Functions

- Pilar ()

  Construct a new Pilar:: Pilar object.
- bool **handleEvent** (SDL_Event ∗e, int position_x, int position_y)

Public Attributes

- unsigned short **hp**
- TipoPilar **tipo_pilar**

5.13.1 Detailed Description

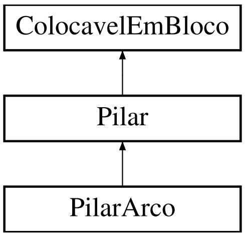
Definition at line 28 of file Pilar.hpp.

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

- include/Pilar.hpp
- src/Pilar.cpp

5.14 PilarArco Class Reference

Inheritance diagram for PilarArco:



Public Member Functions

- PilarArco ()

  Construct a new Pilar Arco:: Pilar Arco object.

Additional Inherited Members

5.14.1 Detailed Description

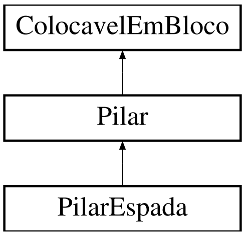
Definition at line 52 of file Pilar.hpp.

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

- include/Pilar.hpp
- src/Pilar.cpp

5.15 PilarEspada Class Reference

Inheritance diagram for PilarEspada:



Public Member Functions

- PilarEspada ()

  Construct a new Pilar Espada:: Pilar Espada object.

Additional Inherited Members

5.15.1 Detailed Description

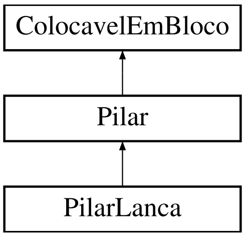
Definition at line 38 of file Pilar.hpp.

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

- include/Pilar.hpp
- src/Pilar.cpp

5.16 PilarLanca Class Reference

Inheritance diagram for PilarLanca:



Public Member Functions

- PilarLanca ()

  Construct a new Pilar Lanca:: Pilar Lanca object.

Additional Inherited Members

5.16.1 Detailed Description

Definition at line 45 of file Pilar.hpp.

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

- include/Pilar.hpp
- src/Pilar.cpp

5.17 Player Class Reference

Public Member Functions

- Player ()

  Construct a new Player:: Player object.
- bool criar_necromancer (TipoNecromancer)
- bool criar_pilar (TipoPilar)
- bool tem_pilar (TipoPilar)
- bool tem_necromancer (TipoNecromancer)
- Pilar ∗ pilar (TipoPilar)
- Necromancer ∗ necromancer (TipoNecromancer)
- bool captar_recurso (TipoRecurso)
- void **muda_time** ()
- bool perdeu_jogo ()
- unsigned short pontuacao ()
- void print_recursos (const char ∗)

Public Attributes

- unsigned short **time**
- unsigned short **metal**
- unsigned short **ossos**
- Guerreiro **guerreiro**
- Cavaleiro **cavaleiro**
- Arqueiro **arqueiro**
- PilarEspada **pilar_espada**
- PilarArco **pilar_arco**
- PilarLanca **pilar_lanca**

5.17.1 Detailed Description

Definition at line 24 of file Player.hpp.

5.17.2 Member Function Documentation

5.17.2.1 captar_recurso()

```
bool Player::captar_recurso (
            TipoRecurso rec )
```

Parameters

rec	

Returns

true
false

Definition at line 79 of file Player.cpp.

```
79                                          {
80     if (rec == TipoRecurso::METAL) {
81         this->metal = this->metal + QUANTIDADE_METAL_POR_ITEM;
82     }
83     if (rec == TipoRecurso::OSSOS) {
84         this->ossos = this->ossos + QUANTIDADE_OSSOS_POR_ITEM;
85     }
86     return true;
87 }
```

5.17.2.2 criar_necromancer()

```
bool Player::criar_necromancer (
               TipoNecromancer nec )
```

Parameters

nec	

Returns

true
false

Definition at line 205 of file Player.cpp.

Referenced by Controlador::criar_necromancer(), and Controlador::fortalecer_necromancer().

```
205                                                   {
206     if (nec == TipoNecromancer::GUERREIRO) {
207         if (!this->pilar_espada.vivo)
208             return false;
209         if (this->ossos < OSSOS_CRIAR_GUERREIRO)
210             return false;
211 
212         this->ossos -= OSSOS_CRIAR_GUERREIRO;
213         if (!this->guerreiro.vivo) {
214             this->guerreiro.revive();
215             this->guerreiro.mp = MP_INICIAL_GUERREIRO;
216         } else {
217             this->guerreiro.mp += MP_INICIAL_GUERREIRO;
218         }
219     }
220 
221     if (nec == TipoNecromancer::CAVALEIRO) {
222         if (!this->pilar_lanca.vivo)
223             return false;
224         if (this->ossos < OSSOS_CRIAR_CAVALEIRO)
225             return false;
226 
227         this->ossos -= OSSOS_CRIAR_CAVALEIRO;
228 
229         if (!this->cavaleiro.vivo) {
230             this->cavaleiro.revive();
231             this->cavaleiro.mp = MP_INICIAL_CAVALEIRO;
232         } else {
233             this->cavaleiro.mp += MP_INICIAL_CAVALEIRO;
234         }
235     }
236 
237     if (nec == TipoNecromancer::ARQUEIRO) {
238         if (!this->pilar_arco.vivo)
239             return false;
240         if (this->ossos < OSSOS_CRIAR_ARQUEIRO)
241             return false;
242 
243         this->ossos -= OSSOS_CRIAR_ARQUEIRO;
244         if (!this->arqueiro.vivo) {
245             this->arqueiro.revive();
246             this->arqueiro.mp = MP_INICIAL_ARQUEIRO;
247         } else {
248             this->arqueiro.mp += MP_INICIAL_ARQUEIRO;
249         }
250     }
251 
252     return true;
253 }
```

5.17.2.3 criar_pilar()

```
bool Player::criar_pilar (
            TipoPilar pil )
```

Parameters

pil	

Returns

true
false

Definition at line 156 of file Player.cpp.

Referenced by Controlador::criar_pilar(), and Controlador::fortalecer_pilar().

```
156                                       {
157     if (pil == TipoPilar::ARCO) {
158         if (this->metal < METAL_CRIAR_PILAR_ARCO)
159             return false;
160 
161         this->metal -= METAL_CRIAR_PILAR_ARCO;
162         if (!this->pilar_arco.vivo) {
163             this->pilar_arco.revive();
164             this->pilar_arco.hp = HP_INICIAL_PILAR_ARCO;
165         } else {
166             this->pilar_arco.hp += HP_INICIAL_PILAR_ARCO;
167         }
168     }
169 
170     if (pil == TipoPilar::LANCA) {
171         if (this->metal < METAL_CRIAR_PILAR_LANCA)
172             return false;
```

```

        this->metal -= METAL_CRIAR_PILAR_LANCA;
        if (!this->pilar_lanca.vivo) {
            this->pilar_lanca.revive();
            this->pilar_lanca.hp = HP_INICIAL_PILAR_LANCA;
        } else {
            this->pilar_lanca.hp += HP_INICIAL_PILAR_LANCA;
        }
    }

    if (pil == TipoPilar::ESPADA) {
        if (this->metal < METAL_CRIAR_PILAR_ESPADA)
            return false;

        this->metal -= METAL_CRIAR_PILAR_ESPADA;
        if (!this->pilar_espada.vivo) {
            this->pilar_espada.revive();
            this->pilar_espada.hp = HP_INICIAL_PILAR_ESPADA;
        } else {
            this->pilar_espada.hp += HP_INICIAL_PILAR_ESPADA;
        }
    }
    return true;
}
```

5.17.2.4 necromancer()

```
Necromancer * Player::necromancer (
              TipoNecromancer nec )
```

Parameters

nec	

Returns

Necromancer∗

Definition at line 114 of file Player.cpp.

Referenced by Controlador::criar_necromancer(), Controlador::fortalecer_necromancer(), and tem_necromancer().

```
114                                                    {
115     if (nec == TipoNecromancer::GUERREIRO)
116         return &this->guerreiro;
117
118     if (nec == TipoNecromancer::ARQUEIRO)
119         return &this->arqueiro;
120
121     if (nec == TipoNecromancer::CAVALEIRO)
122         return &this->cavaleiro;
123
124     return nullptr;
125 }
```

5.17.2.5 perdeu_jogo()

```
bool Player::perdeu_jogo ( )
```

Returns

true
false

Definition at line 289 of file Player.cpp.

Referenced by Controlador::alguem_ganhou().

```
                        {
    return !(this->guerreiro.vivo || this->pilar_espada.vivo ||
    this->arqueiro.vivo || this->pilar_arco.vivo ||
    this->cavaleiro.vivo || this->pilar_lanca.vivo);
}
```

5.17.2.6 pilar()

```
Pilar * Player::pilar (
              TipoPilar pil )
```

Parameters

pil	

Returns

Pilar∗

Definition at line 95 of file Player.cpp.

Referenced by Controlador::criar_pilar(), Controlador::fortalecer_pilar(), and tem_pilar().

```
                                   {
    if (pil == TipoPilar::ARCO)
        return &this->pilar_arco;

    if (pil == TipoPilar::LANCA)
        return &this->pilar_lanca;

    if (pil == TipoPilar::ESPADA)
        return &this->pilar_espada;

    return nullptr;
}
```

5.17.2.7 pontuacao()

```
unsigned short Player::pontuacao ( )
```

Returns

unsigned short

Definition at line 300 of file Player.cpp.

Referenced by Controlador::alguem_ganhou().

```
300                                  {
301     unsigned short total = 0;
302 
303     if (this->pilar_espada.vivo)
304         total+=this->pilar_espada.hp;
305     if (this->pilar_arco.vivo)
306         total+=this->pilar_arco.hp;
307     if (this->pilar_lanca.vivo)
308         total+=this->pilar_lanca.hp;
309     if (this->guerreiro.vivo)
310         total+=this->guerreiro.mp;
311     if (this->arqueiro.vivo)
312         total+=this->arqueiro.mp;
313     if (this->cavaleiro.vivo)
314         total+=this->cavaleiro.mp;
315 
316     return total;
317 }
```

5.17.2.8 print_recursos()

```
void Player::print_recursos (
               const char * nome )
```

Parameters

nome	

Definition at line 260 of file Player.cpp.

```
260                                             {
261     using namespace std;
262 
263     #ifdef PROD
264     std::ostringstream metal_s, bones_s;
265 
266     metal_s << "METAL: " << this->metal;
267     std::string m = metal_s.str();
268 
269     bones_s << "BONES: " << this->ossos;
270     std::string o = bones_s.str();
271 
272     textMetal .loadFromRenderedText(m);
273     textMetal .render(444, 568);
274     textBones.loadFromRenderedText(o);
275     textBones.render(654, 568);
276     #else
277     cout << nome << " tem: ";
278     cout << this->metal << " de metal, ";
279     cout << this->ossos << " de ossos" << endl;
280     #endif
281 }
```

5.17.2.9 tem_necromancer()

```
bool Player::tem_necromancer (
             TipoNecromancer nec )
```

Parameters

nec	

Returns

true
false

Definition at line 145 of file Player.cpp.

References necromancer().

```
145                                           {
146     return this->necromancer(nec)->vivo;
147 }
```

5.17.2.10 tem_pilar()

```
bool Player::tem_pilar (
             TipoPilar pil )
```

Parameters

pil	

Returns

true
false

Definition at line 134 of file Player.cpp.

References pilar().

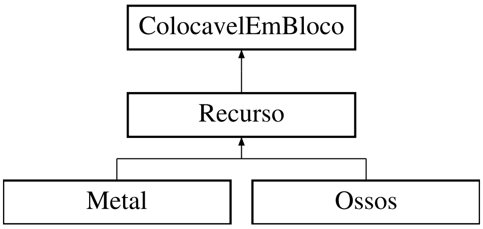
```
134                                 {
135     return this->pilar(pil)->vivo;
136 }
```

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

- include/Player.hpp
- src/Player.cpp

5.18 Recurso Class Reference

Inheritance diagram for Recurso:



Public Member Functions

- Recurso ()

  Construct a new Recurso:: Recurso object.

Public Attributes

- unsigned short **qtde**
- TipoRecurso **tipo_recurso**

5.18.1 Detailed Description

Definition at line 27 of file Recurso.hpp.

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

- include/Recurso.hpp
- src/Recurso.cpp

Chapter 6

File Documentation

6.1 include/Bloco.hpp File Reference

```
#include "Utils.hpp"
```

Classes

- class ColocavelEmBloco
- class Bloco

Enumerations

- enum **TipoConteudoBloco** { **UNIDADE** = 0, **PREDIO**, **RECURSO** }

6.1.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

6.2 include/Button.hpp File Reference

```
#include "Game.hpp"
#include "common.hpp"
```

Classes

- class Button

Macros

- #define **Button_STRUCTURE**
- #define **TOTAL_MENU_BUTTONS** 3

Enumerations

- enum **ButtonType** {
BUTTON_PLAY, **BUTTON_CREDITS**, **BUTTON_QUIT**, **BUTTON_BACK_CREDITS**,
BUTTON_BACK_GAME, **BUTTON_CRIAR_PILAR_ARCO**, **BUTTON_CRIAR_PILAR_LANCA**, **BUTTON_CRIAR_PILAR_ESPADA**,
BUTTON_CRIAR_NECRO, **BUTTON_FORT_PILAR**, **BUTTON_FORT_NECRO**, **BUTTON_CRIAR**,
CANCEL }

6.2.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.3 include/common.hpp File Reference

```
#include "Button.hpp"
#include "Texture.hpp"
#include <SDL2/SDL_ttf.h>
```

Macros

- #define **SCREEN_WIDTH** 800
- #define **SCREEN_HEIGHT** 600
- #define **SQUARE_SIZE** 40
- #define **PLAYER_AREA_START_Y** 40
- #define **PLAYER_AREA_END_Y** 560
- #define **PLAYER_AREA_START_X** 120
- #define **PLAYER_AREA_END_X** 680

Variables

- Button **menuButtons** [TOTAL_MENU_BUTTONS]
- Button **pauseButton** [2]
- Button **creditsBackButton**
- SDL_Window ∗ **window**
- SDL_Renderer ∗ **renderer**
- Texture **menu_screen**
- Texture **credit_screen**
- Texture **pause_screen**
- Texture **round_screen**
- Texture **knight** [2]
- Texture **solider** [2]
- Texture **archer** [2]
- Texture **map_screen**
- Texture **ganhou_screen** [2]
- TTF_Font ∗ **font**
- Texture **textActiveItem**
- Texture **textHP**
- Texture **textBones**
- Texture **textMetal**
- Texture **textRound**
- Texture **pilar_archer** [2]
- Texture **pilar_knight** [2]
- Texture **pilar_solider** [2]
- Texture **bones** [20]
- Texture **metal** [20]
- int **ativo_x_jog**
- int **ativo_y_jog**
- int **ativo_x_cpu**
- int **ativo_y_cpu**
- int **ganhou_time**

6.3.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.4 include/Controlador.hpp File Reference

```
#include <string>
#include <list>
#include <cmath>
#include "Recurso.hpp"
#include "Mapa.hpp"
#include "Player.hpp"
#include "Utils.hpp"
```

Classes

- class Controlador

6.4.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.5 include/Game.hpp File Reference

```
#include "common.hpp"
#include <SDL2/SDL_image.h>
#include <SDL2/SDL.h>
```

6.5.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

6.6 include/Graphics.hpp File Reference

```
#include <SDL2/SDL_image.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include "Texture.hpp"
#include "common.hpp"
```

Classes

- class Graphics

6.6.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.7 include/Mapa.hpp File Reference

```
#include <tuple>
#include <map>
#include "Bloco.hpp"
```

Classes

- class Mapa

Typedefs

- typedef std::pair< unsigned short, unsigned short > **Posicao**
- typedef std::map< Posicao, Bloco > **MapaDeBlocos**

6.7.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.8 include/Necromancer.hpp File Reference

```
#include <SDL2/SDL_image.h>
#include <SDL2/SDL.h>
#include "Bloco.hpp"
#include "Pilar.hpp"
#include "Utils.hpp"
```

Classes

- class Necromancer
- class Guerreiro
- class Cavaleiro
- class Arqueiro

Enumerations

- enum **TipoNecromancer** { **GUERREIRO** = 4, **CAVALEIRO**, **ARQUEIRO** }

6.8.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.9 include/Pilar.hpp File Reference

```
#include "Bloco.hpp"
#include "Utils.hpp"
#include "common.hpp"
```

Classes

- class Pilar
- class PilarEspada
- class PilarLanca
- class PilarArco

Enumerations

- enum **TipoPilar** { **ESPADA** = 7, **LANCA**, **ARCO** }

6.9.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.10 include/Player.hpp File Reference

```
#include "Necromancer.hpp"
#include "Pilar.hpp"
#include "Recurso.hpp"
#include "Utils.hpp"
```

Classes

- class Player

6.10.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.11 include/Recurso.hpp File Reference

```
#include "Bloco.hpp"
#include "Utils.hpp"
```

Classes

- class Recurso
- class Metal
- class Ossos

Enumerations

- enum **TipoRecurso** { **METAL** = 10, **OSSOS** }

6.11.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.12 include/Texture.hpp File Reference

```
#include <SDL2/SDL_image.h>
#include <SDL2/SDL.h>
#include "common.hpp"
#include <string>
```

6.12.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.13 src/Bloco.cpp File Reference

```
#include "../include/Bloco.hpp"
#include <iostream>
```

6.13.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Date

2018-12-01

Copyright

Copyright (c) 2018

6.14 src/Button.cpp File Reference

```
#include "../include/Button.hpp"
```

6.14.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.15 src/Controlador.cpp File Reference

```
#include <string>
#include <sstream>
#include <iostream>
#include "../include/Controlador.hpp"
#include "../include/common.hpp"
```

6.15.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.16 src/Game.cpp File Reference

```
#include <cstdio>
#include <string>
#include <sstream>
#include "../include/Game.hpp"
#include "../include/Controlador.hpp"
```

Functions

- void **boxWarning** (string text)
- void **boxCriarPilar** (Controlador ∗controlador)
- void **boxCriarNecro** (Controlador ∗controlador, TipoPilar tipo_p)
- int **boxPilar** ()
- void **action_pilar_option** (int option, Controlador ∗controlador, TipoPilar tipo_p, TipoNecromancer tipo_n)
- void **handle_necro_ativo** (Controlador ∗controlador, SDL_Event ∗e, int i, int j)
- void **handle_events_elements** (Controlador ∗controlador, SDL_Event ∗e)
- void **movimentar_ativo** (Controlador ∗controlador, SDL_Event ∗e)

6.16.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.17 src/Graphics.cpp File Reference

```
#include "../include/Graphics.hpp"
```

Variables

- Button **menuButtons** [TOTAL_MENU_BUTTONS]
- Button **pauseButton** [2]
- Button **creditsBackButton**
- SDL_Window ∗ **window** = NULL
- SDL_Renderer ∗ **renderer** = NULL
- Texture **menu_screen**
- Texture **credit_screen**
- Texture **pause_screen**
- Texture **round_screen**
- Texture **knight** [2]
- Texture **solider** [2]
- Texture **archer** [2]
- Texture **map_screen**
- Texture **ganhou_screen** [2]
- TTF_Font ∗ **font**
- Texture **textActiveItem**
- Texture **textHP**
- Texture **textBones**
- Texture **textMetal**
- Texture **textRound**
- Texture **pilar_archer** [2]
- Texture **pilar_knight** [2]
- Texture **pilar_solider** [2]
- Texture **bones** [20]
- Texture **metal** [20]
- int **ativo_x_jog**
- int **ativo_y_jog**
- int **ativo_x_cpu**
- int **ativo_y_cpu**
- int **ganhou_time**

6.17.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.18 src/main.cpp File Reference

```
#include "../include/Game.hpp"
#include "../include/Graphics.hpp"
```

Functions

- int main (int argc, char ∗args[ ])

6.18.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.18.2 Function Documentation

6.18.2.1 main()

```
int main (
            int argc,
            char ∗ args[ ] )
```

Parameters

argc	
args	

Returns

int

Definition at line 26 of file main.cpp.

```
26                                  {
27     Graphics *graphics = new Graphics();
28     Game *game = new Game();
29 
30     if (!graphics->init()) {
31         printf("Falha ao inicializar!\n");
32     } else {
33         // Carrega mídia
34         if (!graphics->loadMedia()) {
35             printf("Nao foi possivel carregar a midia!\n");
36         } else {
37             game->playGame(GAME_MENU);
38         }
39         graphics->close();;
40     }
41 
42     return 0;
43 }
```

6.19 src/Mapa.cpp File Reference

```
#include "../include/Mapa.hpp"
```

6.19.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.20 src/Necromancer.cpp File Reference

```
#include "../include/Necromancer.hpp"
```

6.20.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.21 src/Pilar.cpp File Reference

```
#include "../include/Pilar.hpp"
```

6.21.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.22 src/Player.cpp File Reference

```
#include "../include/Player.hpp"
#include <iostream>
#include <string>
#include <sstream>
#include "../include/common.hpp"
```

6.22.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.23 src/Recurso.cpp File Reference

```
#include "../include/Recurso.hpp"
```

6.23.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Copyright (c) 2018

6.24 src/Texture.cpp File Reference

```
#include "../include/Texture.hpp"
```

6.24.1 Detailed Description

Author

Alex Siqueira (siqueiralex@github.com)
Alexander André (Alexander-1995@github.com)
Arthur Veiga (arthurveiga@github.com)
Matheus Veleci (matheusvsantos@github.com)
Luis Luz (lightguy875@github.com)

Version

0.1

Date

2018-12-01

Copyright

Index