

# Gunther's Quest Alpha

0.2

Gerado por Doxygen 1.8.0

Domingo, 17 de Junho de 2012 23:27:58



# Sumário

<b>1</b>	<b>Índice das Estruturas de Dados</b>	<b>1</b>
1.1	Estruturas de Dados . . . . .	1
<b>2</b>	<b>Índice dos Arquivos</b>	<b>3</b>
2.1	Lista de Arquivos . . . . .	3
<b>3</b>	<b>Estruturas</b>	<b>5</b>
3.1	Referência da Estrutura _jogador . . . . .	5
3.1.1	Descrição Detalhada . . . . .	5
3.2	Referência da Estrutura _ladrilho . . . . .	5
3.2.1	Descrição Detalhada . . . . .	6
3.3	Referência da Estrutura _objeto . . . . .	6
3.3.1	Descrição Detalhada . . . . .	6
3.4	Referência da Estrutura _tabuleiro . . . . .	6
3.4.1	Descrição Detalhada . . . . .	6
3.5	Referência da Estrutura struct_objeto . . . . .	7
3.5.1	Descrição Detalhada . . . . .	7
3.6	Referência da Estrutura struct_vetor . . . . .	7
3.6.1	Descrição Detalhada . . . . .	7
<b>4</b>	<b>Arquivos</b>	<b>9</b>
4.1	Referência do Arquivo include/gunther.h . . . . .	9
4.1.1	Descrição Detalhada . . . . .	9
4.1.2	Funções . . . . .	10
4.1.2.1	gunther_menu . . . . .	10
4.1.2.2	inicio . . . . .	10
4.1.2.3	manual . . . . .	10
4.2	Referência do Arquivo include/rkefisica.h . . . . .	10
4.2.1	Descrição Detalhada . . . . .	10
4.2.2	Funções . . . . .	11
4.2.2.1	rke_adiciona_objeto . . . . .	11
4.2.2.2	rke_conta_objetos . . . . .	11

4.2.2.3	<a href="#">rke_get_objeto</a>	11
4.2.2.4	<a href="#">rke_set_arrasto</a>	11
4.2.2.5	<a href="#">rke_set_delta_t</a>	11
4.2.2.6	<a href="#">rke_set_numero_objetos</a>	11
4.2.2.7	<a href="#">rke_set_vetor_mundo</a>	12
4.2.2.8	<a href="#">rke_simula</a>	12
4.3	<a href="#">Referência do Arquivo include/rkegraficos.h</a>	12
4.3.1	<a href="#">Descrição Detalhada</a>	12
4.3.2	<a href="#">Funções</a>	13
4.3.2.1	<a href="#">rke_abre_janela</a>	13
4.3.2.2	<a href="#">rke_aplica_clip_a_mapa</a>	13
4.3.2.3	<a href="#">rke_aplica_clip_duplo</a>	13
4.3.2.4	<a href="#">rke_aplica_tela</a>	13
4.3.2.5	<a href="#">rke_carrega_BMP</a>	13
4.3.2.6	<a href="#">rke_carrega_clip_duplo</a>	14
4.3.2.7	<a href="#">rke_libera_tela</a>	14
4.4	<a href="#">Referência do Arquivo include/rkerender.h</a>	14
4.4.1	<a href="#">Descrição Detalhada</a>	15
4.4.2	<a href="#">Definições dos tipos</a>	15
4.4.2.1	<a href="#">Jogador</a>	15
4.4.2.2	<a href="#">Ladrilho</a>	15
4.4.2.3	<a href="#">Objeto</a>	15
4.4.2.4	<a href="#">Tabuleiro</a>	15
4.4.3	<a href="#">Funções</a>	16
4.4.3.1	<a href="#">rke_carrega_fase</a>	16
4.4.3.2	<a href="#">rke_carrega_jogador</a>	16
4.4.3.3	<a href="#">rke_carrega_objetos</a>	16
4.4.3.4	<a href="#">rke_carrega_terreno</a>	16
4.4.3.5	<a href="#">rke_move_jogador</a>	17
4.4.3.6	<a href="#">rke_render</a>	17
4.5	<a href="#">Referência do Arquivo include/rketypes.h</a>	17
4.5.1	<a href="#">Descrição Detalhada</a>	17
4.5.2	<a href="#">Definições dos tipos</a>	18
4.5.2.1	<a href="#">objeto</a>	18
4.5.2.2	<a href="#">vetor</a>	18
4.6	<a href="#">Referência do Arquivo src/main.c</a>	18
4.6.1	<a href="#">Descrição Detalhada</a>	18
4.7	<a href="#">Referência do Arquivo src/menu/menu.c</a>	19
4.7.1	<a href="#">Descrição Detalhada</a>	19
4.7.2	<a href="#">Funções</a>	19

4.7.2.1	gunther_menu . . . . .	19
4.7.2.2	inicio . . . . .	19
4.7.2.3	manual . . . . .	19
4.8	Referência do Arquivo src/rkegraficos/rkegraficos.c . . . . .	20
4.8.1	Descrição Detalhada . . . . .	20
4.8.2	Funções . . . . .	20
4.8.2.1	rke_abre_janela . . . . .	20
4.8.2.2	rke_aplica_clip_a_mapa . . . . .	20
4.8.2.3	rke_aplica_clip_duplo . . . . .	20
4.8.2.4	rke_aplica_tela . . . . .	21
4.8.2.5	rke_carrega_BMP . . . . .	21
4.8.2.6	rke_carrega_clip_duplo . . . . .	21
4.8.2.7	rke_libera_tela . . . . .	21
4.9	Referência do Arquivo src/rkerender/rkerender.c . . . . .	21
4.9.1	Descrição Detalhada . . . . .	22
4.9.2	Funções . . . . .	22
4.9.2.1	rke_carrega_fase . . . . .	22
4.9.2.2	rke_carrega_jogador . . . . .	22
4.9.2.3	rke_carrega_objetos . . . . .	22
4.9.2.4	rke_carrega_terreno . . . . .	23
4.9.2.5	rke_move_jogador . . . . .	23
4.9.2.6	rke_render . . . . .	23



# Capítulo 1

## Índice das Estruturas de Dados

### 1.1 Estruturas de Dados

Aqui estão as estruturas de dados, uniões e suas respectivas descrições:

<a href="#">_jogador</a>	5
<a href="#">_ladrilho</a>	5
<a href="#">_objeto</a>	6
<a href="#">_tabuleiro</a>	6
<a href="#">struct_objeto</a>	7
<a href="#">struct_vetor</a>	7





## Capítulo 2

# Índice dos Arquivos

### 2.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

include/ <a href="#">gunther.h</a>	
Arquivo header geral do jogo . . . . .	9
include/ <a href="#">rkefisica.h</a>	
Arquivo header da biblioteca de funções físicas . . . . .	10
include/ <a href="#">rkegraficos.h</a>	
Arquivo header da parte gráfica . . . . .	12
include/ <a href="#">rkerender.h</a>	
Arquivo header do renderizador . . . . .	14
include/ <a href="#">rketypes.h</a>	
Arquivo header de tipos e defines do Red Knife Engine . . . . .	17
src/ <a href="#">main.c</a>	
Ponto de entrada do jogo . . . . .	18
src/menu/ <a href="#">menu.c</a>	
Implementação do menu principal . . . . .	19
src/rkegraficos/ <a href="#">rkegraficos.c</a>	
Utilitários gráficos . . . . .	20
src/rkerender/ <a href="#">rkerender.c</a>	
Renderizador de fases . . . . .	21



## Capítulo 3

# Estruturas

### 3.1 Referência da Estrutura `_jogador`

```
#include <rkerender.h>
```

#### Campos de Dados

- `SDL_Rect` **retangulo** [8]
- `int` **x**
- `int` **y**
- `int` **direcao**

#### 3.1.1 Descrição Detalhada

Struct Jogador

##### Parâmetros

<i>retangulo</i>	Vetor com os 8 retângulos das visões do jogador
<i>x</i>	Posição x do jogador
<i>y</i>	Posição y do jogador
<i>direcao</i>	Direção que o jogador está olhando

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `include/rkerender.h`

### 3.2 Referência da Estrutura `_ladrilho`

```
#include <rkerender.h>
```

#### Campos de Dados

- `SDL_Rect` **retangulo**
- `int` **passavel**

### 3.2.1 Descrição Detalhada

Struct Ladrilho

Parâmetros

<i>retangulo</i>	Retângulo do ladrilho
<i>passavel</i>	Indica se o jogador pode andar em cima do ladrilho

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `include/rkrender.h`

## 3.3 Referência da Estrutura \_objeto

```
#include <rkrender.h>
```

### Campos de Dados

- `SDL_Rect` **retangulo**

### 3.3.1 Descrição Detalhada

Struct Objeto

Parâmetros

<i>retangulo</i>	Retângulo do objeto
------------------	---------------------

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `include/rkrender.h`

## 3.4 Referência da Estrutura \_tabuleiro

```
#include <rkrender.h>
```

### Campos de Dados

- `char **` **terreno**
- `char **` **objetos**

### 3.4.1 Descrição Detalhada

Struct Tabuleiro

Parâmetros

<i>terreno</i>	Matriz que armazena como o terreno é construído
<i>objetos</i>	Matriz que armazena a localização dos objetos

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- include/[rkerender.h](#)

## 3.5 Referência da Estrutura struct\_objeto

```
#include <rketypes.h>
```

### Campos de Dados

- int **id**
- double **x**
- double **y**
- double **v\_x**
- double **v\_y**
- double **massa**
- double **tempo**

#### 3.5.1 Descrição Detalhada

Struct objeto

##### Parâmetros

<i>id</i>	Identificador único
<i>x</i>	Componente x
<i>y</i>	Componente y
<i>v_x</i>	Componente x da velocidade do objeto
<i>v_y</i>	Componente y da velocidade do objeto
<i>massa</i>	Massa do objeto
<i>tempo</i>	Tempo de vida do objeto

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- include/[rketypes.h](#)

## 3.6 Referência da Estrutura struct\_vetor

```
#include <rketypes.h>
```

### Campos de Dados

- double **x**
- double **y**

#### 3.6.1 Descrição Detalhada

Struct vetor

##### Parâmetros

<i>x</i>	Componente x
<i>y</i>	Componente y

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `include/rketypes.h`

# Capítulo 4

## Arquivos

### 4.1 Referência do Arquivo include/gunther.h

Arquivo header geral do jogo.

```
#include <SDL/SDL.h>
#include "rkegraficos.h"
#include "rkefisica.h"
#include "rkerender.h"
```

Gráfico de dependência de inclusões para gunther.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

#### Definições e Macros

- **#define DIR\_IMAGENS** "data/imagens/"
- **#define LADRILHO\_ALTURA** 48
- **#define LADRILHO\_LARGURA** 48
- **#define TELA\_LARGURA** 15
- **#define TELA\_ALTURA** 10
- **#define SAIR** -1
- **#define NOVOJOGO** 0
- **#define CARREGAJOGO** 1
- **#define OPCOES** 2
- **#define MANUAL** 3
- **#define CREDITOS** 4
- **#define INICIO** 5
- **#define COMMANDOS** 6

#### Funções

- void [gunther\\_menu](#) (SDL\_Surface \*tela)
- int [inicio](#) (SDL\_Surface \*screen)
- int [manual](#) (SDL\_Surface \*screen, int section)

#### 4.1.1 Descrição Detalhada

Arquivo header geral do jogo.

## Autor

João da Silva, Marina Salles, Ricardo Macedo

## 4.1.2 Funções

### 4.1.2.1 void gunther\_menu ( SDL\_Surface \* tela )

Renderiza o menu baseado na seleção

#### Parâmetros

<i>tela</i>	Tela onde o menu será renderizado
-------------	-----------------------------------

### 4.1.2.2 int inicio ( SDL\_Surface \* screen )

Menu principal

#### Parâmetros

<i>tela</i>	Tela onde o menu será renderizado
-------------	-----------------------------------

### 4.1.2.3 int manual ( SDL\_Surface \* screen, int mode )

Seção "Manual" do menu

#### Parâmetros

<i>tela</i>	Tela onde a seção será renderizada
-------------	------------------------------------

## 4.2 Referência do Arquivo include/rkefisica.h

Arquivo header da biblioteca de funções físicas.

```
#include "rketypes.h"
```

Gráfico de dependência de inclusões para rkefisica.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

## Funções

- void [rke\\_set\\_delta\\_t](#) (double d\_t)
- void [rke\\_set\\_arrasto](#) (double coef\_arrasto)
- void [rke\\_set\\_vetor\\_mundo](#) (double x, double y)
- void [rke\\_set\\_numero\\_objetos](#) (int numero)
- void [rke\\_adiciona\\_objeto](#) (int id, double x, double y, double v\_x, double v\_y, double massa, double tempo)
- [objeto rke\\_get\\_objeto](#) (int i)
- int [rke\\_conta\\_objetos](#) ()
- void [rke\\_simula](#) ()

### 4.2.1 Descrição Detalhada

Arquivo header da biblioteca de funções físicas.



## Autor

João da Silva, Marina Salles, Ricardo Macedo

## 4.2.2 Funções

**4.2.2.1 void rke\_adiciona\_objeto ( int *id*, double *x*, double *y*, double *v\_x*, double *v\_y*, double *massa*, double *tempo* )**

Adiciona um objeto à lista de objetos a serem simulados.

## Parâmetros

<i>id</i>	Identificador único do objeto
<i>x</i>	Posição x do objeto
<i>y</i>	Posição y do objeto
<i>v_x</i>	Velocidade em x do objeto
<i>v_y</i>	Velocidade em y do objeto
<i>massa</i>	Massa do objeto
<i>tempo</i>	Tempo de vida do objeto em segundos

**4.2.2.2 int rke\_conta\_objetos ( )**

Conta o número de objetos ativos, ou seja, com tempo de vida diferente de zero.

**4.2.2.3 objeto rke\_get\_objeto ( int *i* )**

Retorna o i-ésimo objeto.

## Parâmetros

<i>i</i>	Índice do objeto
----------	------------------

**4.2.2.4 void rke\_set\_arrasto ( double *coef\_arrasto* )**

Configura o coeficiente de arrasto da superfície.

## Parâmetros

<i>coef_arrasto</i>	Coeficiente de 0.0 a 1.0
---------------------	--------------------------

**4.2.2.5 void rke\_set\_delta\_t ( double *d\_t* )**

Indica a resolução da simulação. Este é o tamanho do quanta de tempo.

## Parâmetros

<i>d_t</i>	Resolução em segundos
------------	-----------------------

**4.2.2.6 void rke\_set\_numero\_objetos ( int *numero* )**

Configura o número total de objetos a serem simulados.

## Parâmetros

<i>numero</i>	Número de objetos
---------------	-------------------

## 4.2.2.7 void rke\_set\_vetor\_mundo ( double x, double y )

Configura o vetor base que registrará todos os objetos do mundo.

## Parâmetros

<i>x</i>	Componente x
<i>y</i>	Componente y

## 4.2.2.8 void rke\_simula ( )

Simula todos os objetos por um quanta de tempo.

## 4.3 Referência do Arquivo include/rkegraficos.h

Arquivo header da parte gráfica.

```
#include "SDL/SDL.h"
```

Gráfico de dependência de inclusões para rkegraficos.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

## Definições e Macros

- #define **bool** int
- #define **true** 1
- #define **false** 0
- #define **TAMMAX** 100
- #define **IMAGEDIR** "data/imagens/"
- #define **KEY\_VERMELHO** 0
- #define **KEY\_VERDE** 0xFF
- #define **KEY\_AZUL** 0xFF

## Funções

- SDL\_Surface \* [rke\\_abre\\_janela](#) (int largura, int altura)
- void [rke\\_aplica\\_tela](#) (SDL\_Surface \*destino, SDL\_Surface \*origem)
- SDL\_Surface \* [rke\\_carrega\\_BMP](#) (char \*arquivo)
- void [rke\\_aplica\\_clip\\_duplo](#) (SDL\_Surface \*destino, SDL\_Surface \*origem, SDL\_Rect clip[ ][2], int selecao)
- void [rke\\_carrega\\_clip\\_duplo](#) (char \*arquivo, SDL\_Rect clip[ ][2])
- void [rke\\_libera\\_tela](#) (SDL\_Surface \*tela)
- void [rke\\_aplica\\_clip\\_a\\_mapa](#) (SDL\_Surface \*destino, SDL\_Surface \*origem, int mapa\_x, int mapa\_y, SDL\_Rect clip)

### 4.3.1 Descrição Detalhada

Arquivo header da parte gráfica.

## Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.3.2 Funções

#### 4.3.2.1 SDL\_Surface\* rke\_abre\_janela ( int largura, int altura )

Abre uma nova janela

## Parâmetros

<i>largura</i>	Largura em pixels
<i>altura</i>	Altura em pixels

#### 4.3.2.2 void rke\_aplica\_clip\_a\_mapa ( SDL\_Surface \* destino, SDL\_Surface \* origem, int mapa\_x, int mapa\_y, SDL\_Rect clip )

Aplica um retângulo a uma tela

## Parâmetros

<i>destino</i>	A tela que receberá a imagem
<i>origem</i>	A tela de origem
<i>mapa_x</i>	A posição X (em ladrilhos) no destino
<i>mapa_y</i>	A posição Y (em ladrilhos) no destino
<i>clip</i>	O retângulo a ser aplicado

#### 4.3.2.3 void rke\_aplica\_clip\_duplo ( SDL\_Surface \* destino, SDL\_Surface \* origem, SDL\_Rect clip[ ][2], int selecao )

Copia um retângulo em uma tela para um outro retângulo em outra tela

## Parâmetros

<i>destino</i>	Tela de destino
<i>origem</i>	Tela de origem
<i>clip</i>	Vetor com os retângulos
<i>selecao</i>	Opção selecionada no menu

#### 4.3.2.4 void rke\_aplica\_tela ( SDL\_Surface \* destino, SDL\_Surface \* origem )

Aplica uma tela guardada em memória em outra

## Parâmetros

<i>destino</i>	Tela de destino
<i>origem</i>	Tela de origem

#### 4.3.2.5 SDL\_Surface\* rke\_carrega\_BMP ( char \* arquivo )

Carrega um arquivo Bitmap em uma tela

## Parâmetros

<i>arquivo</i>	O nome do arquivo
----------------	-------------------

4.3.2.6 void rke\_carrega\_clip\_duplo ( char \* *arquivo*, SDL\_Rect *clip*[[2]] )

Carrega retângulos a partir de um arquivo

## Parâmetros

<i>arquivo</i>	O arquivo a ser carregado
<i>clip</i>	O vetor de retângulos que conterá as informações

4.3.2.7 void rke\_libera\_tela ( SDL\_Surface \* *tela* )

Libera a memória de uma tela

## Parâmetros

<i>tela</i>	A tela em questão
-------------	-------------------

## 4.4 Referência do Arquivo include/rkrender.h

Arquivo header do renderizador.

```
#include "rkegraficos.h"
```

Gráfico de dependência de inclusões para rkrender.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

### Estruturas de Dados

- struct [\\_ladrilho](#)
- struct [\\_tabuleiro](#)
- struct [\\_objeto](#)
- struct [\\_jogador](#)

### Definições e Macros

- #define **N** 0
- #define **NE** 1
- #define **E** 2
- #define **SE** 3
- #define **S** 4
- #define **SO** 5
- #define **O** 6
- #define **NO** 7

### Definições de Tipos

- typedef struct [\\_ladrilho](#) Ladrilho
- typedef struct [\\_tabuleiro](#) Tabuleiro
- typedef struct [\\_objeto](#) Objeto
- typedef struct [\\_jogador](#) Jogador

## Funções

- void **rke\_render** (char \*fase, char \*imagens, char \*img\_jogador, int larg, int alt, int largura\_ladrilho, int altura\_ladrilho)
- void **rke\_carrega\_terreno** (char \*arquivo, **Ladrilho** terrenos[], int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_carrega\_objetos** (char \*arquivo, **Objeto** objetos[], int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_carrega\_fase** (char \*arquivo, **Tabuleiro** \*tabuleiro, int \*jogador\_x, int \*jogador\_y)
- void **rke\_carrega\_jogador** (**Jogador** \*jogador, int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_move\_jogador** (**Jogador** \*jogador, **Tabuleiro** tabuleiro, **Ladrilho** \*terrenos, **Objeto** \*objetos, int delta\_x, int delta\_y)

### 4.4.1 Descrição Detalhada

Arquivo header do renderizador.

Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.4.2 Definições dos tipos

#### 4.4.2.1 typedef struct \_jogador Jogador

Struct Jogador

Parâmetros

<i>retangulo</i>	Vetor com os 8 retângulos das visões do jogador
<i>x</i>	Posição x do jogador
<i>y</i>	Posição y do jogador
<i>direcao</i>	Direção que o jogador está olhando

#### 4.4.2.2 typedef struct \_ladrilho Ladrilho

Struct Ladrilho

Parâmetros

<i>retangulo</i>	Retângulo do ladrilho
<i>passavel</i>	Indica se o jogador pode andar em cima do ladrilho

#### 4.4.2.3 typedef struct \_objeto Objeto

Struct Objeto

Parâmetros

<i>retangulo</i>	Retângulo do objeto
------------------	---------------------

#### 4.4.2.4 typedef struct \_tabuleiro Tabuleiro

Struct Tabuleiro

## Parâmetros

<i>terreno</i>	Matriz que armazena como o terreno é construído
<i>objetos</i>	Matriz que armazena a localização dos objetos

## 4.4.3 Funções

4.4.3.1 void rke\_carrega\_fase ( char \* *arquivo*, Tabuleiro \* *tabuleiro*, int \* *jogador\_x*, int \* *jogador\_y* )

Carrega as informações do tabuleiro da fase

## Parâmetros

<i>arquivo</i>	O arquivo com as informações
<i>tabuleiro</i>	O lugar para armazenar as informações
<i>jogador_x</i>	Argumento que devolve a posição X do jogador
<i>jogador_y</i>	Argumento que devolve a posição Y do jogador

4.4.3.2 void rke\_carrega\_jogador ( Jogador \* *jogador*, int *larg\_ladrilho*, int *alt\_ladrilho* )

Carrega os retângulos da imagem do jogador

## Parâmetros

<i>O</i>	lugar para armazenar as informações
<i>Largura</i>	em pixels do ladrilho
<i>Altura</i>	em pixels do ladrilho

4.4.3.3 void rke\_carrega\_objetos ( char \* *arquivo*, Objeto *objetos*[], int *larg\_ladrilho*, int *alt\_ladrilho* )

Carrega as informações dos objetos de um arquivo

## Parâmetros

<i>arquivo</i>	O arquivo com as informações
<i>objetos</i>	O lugar para armazenar as informações
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

4.4.3.4 void rke\_carrega\_terreno ( char \* *arquivo*, Ladrilho *terrenos*[], int *larg\_ladrilho*, int *alt\_ladrilho* )

Carrega as informações dos elementos de terreno de um arquivo

## Parâmetros

<i>arquivo</i>	O arquivo com as informações
<i>terrenos</i>	O lugar para armazenar as informações
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

4.4.3.5 void rke\_move\_jogador ( Jogador \* *jogador*, Tabuleiro *tabuleiro*, Ladrilho \* *terrenos*, Objeto \* *objetos*, int *delta\_x*, int *delta\_y* )

Função de movimentação do jogador.

#### Parâmetros

<i>jogador</i>	Informações do jogador
<i>tabuleiro</i>	Informações do tabuleiro
<i>terrenos</i>	Informações dos elementos de terreno
<i>objetos</i>	Informações dos objetos
<i>delta_x</i>	Movimentação no eixo X
<i>delta_y</i>	Movimentação no eixo Y

4.4.3.6 void rke\_render ( char \* *fase*, char \* *imagens*, char \* *img\_jogador*, int *largura*, int *altura*, int *larg\_ladrilho*, int *alt\_ladrilho* )

Função principal do renderizador de fases

#### Parâmetros

<i>fase</i>	Arquivo de fase
<i>imagens</i>	Arquivo clipboard com as imagens para terreno e objetos
<i>img_jogador</i>	Arquivo clipboard com as imagens do jogador
<i>largura</i>	Largura em ladrilhos da tela
<i>altura</i>	Altura em ladrilhos da tela
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

## 4.5 Referência do Arquivo include/rketypes.h

Arquivo header de tipos e defines do Red Knife Engine.

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

### Estruturas de Dados

- struct [struct\\_vetor](#)
- struct [struct\\_objeto](#)

### Definições e Macros

- #define **BARCOID** -1
- #define **ESTATICO** -1.0

### Definições de Tipos

- typedef struct [struct\\_vetor](#) vetor
- typedef struct [struct\\_objeto](#) objeto

#### 4.5.1 Descrição Detalhada

Arquivo header de tipos e defines do Red Knife Engine.

## Autor

João da Silva, Marina Salles, Ricardo Macedo

## 4.5.2 Definições dos tipos

### 4.5.2.1 typedef struct struct\_objeto objeto

Struct objeto

## Parâmetros

<i>id</i>	Identificador único
<i>x</i>	Componente x
<i>y</i>	Componente y
<i>v_x</i>	Componente x da velocidade do objeto
<i>v_y</i>	Componente y da velocidade do objeto
<i>massa</i>	Massa do objeto
<i>tempo</i>	Tempo de vida do objeto

### 4.5.2.2 typedef struct struct\_vetor vetor

Struct vetor

## Parâmetros

<i>x</i>	Componente x
<i>y</i>	Componente y

## 4.6 Referência do Arquivo src/main.c

Ponto de entrada do jogo.

```
#include <stdio.h>
#include <stdlib.h>
#include "gunther.h"
```

Gráfico de dependência de inclusões para main.c:

### Funções

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

### Variáveis

- int **largura** = 720
- int **altura** = 480

### 4.6.1 Descrição Detalhada

Ponto de entrada do jogo.



**Autor**

João da Silva, Marina Salles, Ricardo Macedo

## 4.7 Referência do Arquivo src/menu/menu.c

Implementação do menu principal.

```
#include "gunther.h"
```

Gráfico de dependência de inclusões para menu.c:

**Funções**

- void [gunther\\_menu](#) (SDL\_Surface \*tela)
- int [inicio](#) (SDL\_Surface \*screen)
- int [manual](#) (SDL\_Surface \*screen, int mode)

### 4.7.1 Descrição Detalhada

Implementação do menu principal.

**Autor**

João da Silva, Marina Salles, Ricardo Macedo

### 4.7.2 Funções

#### 4.7.2.1 void [gunther\\_menu](#) ( SDL\_Surface \* *tela* )

Renderiza o menu baseado na seleção

**Parâmetros**

<i>tela</i>	Tela onde o menu será renderizado
-------------	-----------------------------------

#### 4.7.2.2 int [inicio](#) ( SDL\_Surface \* *screen* )

Menu principal

**Parâmetros**

<i>tela</i>	Tela onde o menu será renderizado
-------------	-----------------------------------

#### 4.7.2.3 int [manual](#) ( SDL\_Surface \* *screen*, int *mode* )

Seção "Manual" do menu

**Parâmetros**

<i>tela</i>	Tela onde a seção será renderizada
-------------	------------------------------------

## 4.8 Referência do Arquivo src/rkegraficos/rkegraficos.c

Utilitários gráficos.

```
#include "gunther.h"
```

Gráfico de dependência de inclusões para rkegraficos.c:

### Funções

- `SDL_Surface * rke_abre_janela` (int largura, int altura)
- void `rke_aplica_tela` (SDL\_Surface \*destino, SDL\_Surface \*origem)
- `SDL_Surface * rke_carrega_BMP` (char \*arquivo)
- void `rke_aplica_clip_duplo` (SDL\_Surface \*destino, SDL\_Surface \*origem, SDL\_Rect clip[ ][2], int selecao)
- void `rke_carrega_clip_duplo` (char \*arquivo, SDL\_Rect clip[ ][2])
- void `rke_libera_tela` (SDL\_Surface \*tela)
- void `rke_aplica_clip_a_mapa` (SDL\_Surface \*destino, SDL\_Surface \*origem, int mapa\_x, int mapa\_y, SDL\_Rect clip)

### 4.8.1 Descrição Detalhada

Utilitários gráficos.

Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.8.2 Funções

#### 4.8.2.1 `SDL_Surface* rke_abre_janela ( int largura, int altura )`

Abre uma nova janela

Parâmetros

<i>largura</i>	Largura em pixels
<i>altura</i>	Altura em pixels

#### 4.8.2.2 `void rke_aplica_clip_a_mapa ( SDL_Surface * destino, SDL_Surface * origem, int mapa_x, int mapa_y, SDL_Rect clip )`

Aplica um retângulo a uma tela

Parâmetros

<i>destino</i>	A tela que receberá a imagem
<i>origem</i>	A tela de origem
<i>mapa_x</i>	A posição X (em ladrilhos) no destino
<i>mapa_y</i>	A posição Y (em ladrilhos) no destino
<i>clip</i>	O retângulo a ser aplicado

#### 4.8.2.3 `void rke_aplica_clip_duplo ( SDL_Surface * destino, SDL_Surface * origem, SDL_Rect clip[ ][2], int selecao )`

Copia um retângulo em uma tela para um outro retângulo em outra tela

## Parâmetros

<i>destino</i>	Tela de destino
<i>origem</i>	Tela de origem
<i>clip</i>	Vetor com os retângulos
<i>selecao</i>	Opção selecionada no menu

## 4.8.2.4 void rke\_aplica\_tela ( SDL\_Surface \* destino, SDL\_Surface \* origem )

Aplica uma tela guardada em memória em outra

## Parâmetros

<i>destino</i>	Tela de destino
<i>origem</i>	Tela de origem

## 4.8.2.5 SDL\_Surface\* rke\_carrega\_BMP ( char \* arquivo )

Carrega um arquivo Bitmap em uma tela

## Parâmetros

<i>arquivo</i>	O nome do arquivo
----------------	-------------------

## 4.8.2.6 void rke\_carrega\_clip\_duplo ( char \* arquivo, SDL\_Rect clip[][2] )

Carrega retângulos a partir de um arquivo

## Parâmetros

<i>arquivo</i>	O arquivo a ser carregado
<i>clip</i>	O vetor de retângulos que conterá as informações

## 4.8.2.7 void rke\_libera\_tela ( SDL\_Surface \* tela )

Libera a memória de uma tela

## Parâmetros

<i>tela</i>	A tela em questão
-------------	-------------------

## 4.9 Referência do Arquivo src/rkrender/rkrender.c

Renderizador de fases.

```
#include "gunther.h"
```

```
#include <string.h>
```

Gráfico de dependência de inclusões para rkrender.c:

## Funções

- void **rke\_render** (char \*fase, char \*imagens, char \*img\_jogador, int largura, int altura, int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_move\_jogador** (**Jogador** \*jogador, **Tabuleiro** tabuleiro, **Ladrilho** \*terrenos, **Objeto** \*objetos, int delta\_x, int delta\_y)
- void **rke\_carrega\_terreno** (char \*arquivo, **Ladrilho** terrenos[], int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_carrega\_objetos** (char \*arquivo, **Objeto** objetos[], int larg\_ladrilho, int alt\_ladrilho)
- void **rke\_carrega\_fase** (char \*arquivo, **Tabuleiro** \*tabuleiro, int \*jogador\_x, int \*jogador\_y)
- void **rke\_carrega\_jogador** (**Jogador** \*jogador, int larg\_ladrilho, int alt\_ladrilho)

### 4.9.1 Descrição Detalhada

Renderizador de fases.

Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.9.2 Funções

**4.9.2.1 void rke\_carrega\_fase ( char \* arquivo, Tabuleiro \* tabuleiro, int \* jogador\_x, int \* jogador\_y )**

Carrega as informações do tabuleiro da fase

Parâmetros

<i>arquivo</i>	O arquivo com as informações
<i>tabuleiro</i>	O lugar para armazenar as informações
<i>jogador_x</i>	Argumento que devolve a posição X do jogador
<i>jogador_y</i>	Argumento que devolve a posição Y do jogador

**4.9.2.2 void rke\_carrega\_jogador ( Jogador \* jogador, int larg\_ladrilho, int alt\_ladrilho )**

Carrega os retângulos da imagem do jogador

Parâmetros

<i>O</i>	lugar para armazenar as informações
<i>Largura</i>	em pixels do ladrilho
<i>Altura</i>	em pixels do ladrilho

**4.9.2.3 void rke\_carrega\_objetos ( char \* arquivo, Objeto objetos[], int larg\_ladrilho, int alt\_ladrilho )**

Carrega as informações dos objetos de um arquivo

Parâmetros

<i>arquivo</i>	O arquivo com as informações
<i>objetos</i>	O lugar para armazenar as informações
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

**4.9.2.4 void rke\_carrega\_terreno ( char \* *arquivo*, Ladrilho *terrenos*[], int *larg\_ladrilho*, int *alt\_ladrilho* )**

Carrega as informações dos elementos de terreno de um arquivo

**Parâmetros**

<i>arquivo</i>	O arquivo com as informações
<i>terrenos</i>	O lugar para armazenar as informações
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

**4.9.2.5 void rke\_move\_jogador ( Jogador \* *jogador*, Tabuleiro *tabuleiro*, Ladrilho \* *terrenos*, Objeto \* *objetos*, int *delta\_x*, int *delta\_y* )**

Função de movimentação do jogador.

**Parâmetros**

<i>jogador</i>	Informações do jogador
<i>tabuleiro</i>	Informações do tabuleiro
<i>terrenos</i>	Informações dos elementos de terreno
<i>objetos</i>	Informações dos objetos
<i>delta_x</i>	Movimentação no eixo X
<i>delta_y</i>	Movimentação no eixo Y

**4.9.2.6 void rke\_render ( char \* *fase*, char \* *imagens*, char \* *img\_jogador*, int *largura*, int *altura*, int *larg\_ladrilho*, int *alt\_ladrilho* )**

Função principal do renderizador de fases

**Parâmetros**

<i>fase</i>	Arquivo de fase
<i>imagens</i>	Arquivo clipboard com as imagens para terreno e objetos
<i>img_jogador</i>	Arquivo clipboard com as imagens do jogador
<i>largura</i>	Largura em ladrilhos da tela
<i>altura</i>	Altura em ladrilhos da tela
<i>larg_ladrilho</i>	Largura em pixels do ladrilho
<i>alt_ladrilho</i>	Altura em pixels do ladrilho

# Índice Remissivo

- [\\_jogador, 5](#)
  - [\\_ladrilho, 5](#)
  - [\\_objeto, 6](#)
  - [\\_tabuleiro, 6](#)
- [gunther.h](#)
  - [gunther\\_menu, 10](#)
  - [inicio, 10](#)
  - [manual, 10](#)
- [gunther\\_menu](#)
  - [gunther.h, 10](#)
  - [menu.c, 19](#)
- [include/gunther.h, 9](#)
- [include/rkefisica.h, 10](#)
- [include/rkegraficos.h, 12](#)
- [include/rkerender.h, 14](#)
- [include/rketypes.h, 17](#)
- [inicio](#)
  - [gunther.h, 10](#)
  - [menu.c, 19](#)
- [Jogador](#)
  - [rkerender.h, 15](#)
- [Ladrilho](#)
  - [rkerender.h, 15](#)
- [manual](#)
  - [gunther.h, 10](#)
  - [menu.c, 19](#)
- [menu.c](#)
  - [gunther\\_menu, 19](#)
  - [inicio, 19](#)
  - [manual, 19](#)
- [Objeto](#)
  - [rkerender.h, 15](#)
- [objeto](#)
  - [rketypes.h, 18](#)
- [rke\\_abre\\_janela](#)
  - [rkegraficos.c, 20](#)
  - [rkegraficos.h, 13](#)
- [rke\\_adiciona\\_objeto](#)
  - [rkefisica.h, 11](#)
- [rke\\_aplica\\_clip\\_a\\_mapa](#)
  - [rkegraficos.c, 20](#)
  - [rkegraficos.h, 13](#)
- [rke\\_aplica\\_clip\\_duplo](#)
  - [rkegraficos.c, 20](#)
  - [rkegraficos.h, 13](#)
- [rke\\_aplica\\_tela](#)
  - [rkegraficos.c, 21](#)
  - [rkegraficos.h, 13](#)
- [rke\\_carrega\\_BMP](#)
  - [rkegraficos.c, 21](#)
  - [rkegraficos.h, 13](#)
- [rke\\_carrega\\_clip\\_duplo](#)
  - [rkegraficos.c, 21](#)
  - [rkegraficos.h, 14](#)
- [rke\\_carrega\\_fase](#)
  - [rkerender.c, 22](#)
  - [rkerender.h, 16](#)
- [rke\\_carrega\\_jogador](#)
  - [rkerender.c, 22](#)
  - [rkerender.h, 16](#)
- [rke\\_carrega\\_objetos](#)
  - [rkerender.c, 22](#)
  - [rkerender.h, 16](#)
- [rke\\_carrega\\_terreno](#)
  - [rkerender.c, 22](#)
  - [rkerender.h, 16](#)
- [rke\\_conta\\_objetos](#)
  - [rkefisica.h, 11](#)
- [rke\\_get\\_objeto](#)
  - [rkefisica.h, 11](#)
- [rke\\_libera\\_tela](#)
  - [rkegraficos.c, 21](#)
  - [rkegraficos.h, 14](#)
- [rke\\_move\\_jogador](#)
  - [rkerender.c, 23](#)
  - [rkerender.h, 16](#)
- [rke\\_render](#)
  - [rkerender.c, 23](#)
  - [rkerender.h, 17](#)
- [rke\\_set\\_arrasto](#)
  - [rkefisica.h, 11](#)
- [rke\\_set\\_delta\\_t](#)
  - [rkefisica.h, 11](#)
- [rke\\_set\\_numero\\_objetos](#)
  - [rkefisica.h, 11](#)
- [rke\\_set\\_vetor\\_mundo](#)
  - [rkefisica.h, 12](#)
- [rke\\_simula](#)
  - [rkefisica.h, 12](#)
- [rkefisica.h](#)
  - [rke\\_adiciona\\_objeto, 11](#)
  - [rke\\_conta\\_objetos, 11](#)

- [rke\\_get\\_objeto, 11](#)
  - [rke\\_set\\_arrasto, 11](#)
  - [rke\\_set\\_delta\\_t, 11](#)
  - [rke\\_set\\_numero\\_objetos, 11](#)
  - [rke\\_set\\_vetor\\_mundo, 12](#)
  - [rke\\_simula, 12](#)
- [rkegraficos.c](#)
  - [rke\\_abre\\_janela, 20](#)
  - [rke\\_aplica\\_clip\\_a\\_mapa, 20](#)
  - [rke\\_aplica\\_clip\\_duplo, 20](#)
  - [rke\\_aplica\\_tela, 21](#)
  - [rke\\_carrega\\_BMP, 21](#)
  - [rke\\_carrega\\_clip\\_duplo, 21](#)
  - [rke\\_libera\\_tela, 21](#)
- [rkegraficos.h](#)
  - [rke\\_abre\\_janela, 13](#)
  - [rke\\_aplica\\_clip\\_a\\_mapa, 13](#)
  - [rke\\_aplica\\_clip\\_duplo, 13](#)
  - [rke\\_aplica\\_tela, 13](#)
  - [rke\\_carrega\\_BMP, 13](#)
  - [rke\\_carrega\\_clip\\_duplo, 14](#)
  - [rke\\_libera\\_tela, 14](#)
- [rkerender.c](#)
  - [rke\\_carrega\\_fase, 22](#)
  - [rke\\_carrega\\_jogador, 22](#)
  - [rke\\_carrega\\_objetos, 22](#)
  - [rke\\_carrega\\_terreno, 22](#)
  - [rke\\_move\\_jogador, 23](#)
  - [rke\\_render, 23](#)
- [rkerender.h](#)
  - [Jogador, 15](#)
  - [Ladrilho, 15](#)
  - [Objeto, 15](#)
  - [rke\\_carrega\\_fase, 16](#)
  - [rke\\_carrega\\_jogador, 16](#)
  - [rke\\_carrega\\_objetos, 16](#)
  - [rke\\_carrega\\_terreno, 16](#)
  - [rke\\_move\\_jogador, 16](#)
  - [rke\\_render, 17](#)
  - [Tabuleiro, 15](#)
- [rketypes.h](#)
  - [objeto, 18](#)
  - [vetor, 18](#)
- [src/main.c, 18](#)
- [src/menu/menu.c, 19](#)
- [src/rkegraficos/rkegraficos.c, 20](#)
- [src/rkerender/rkerender.c, 21](#)
- [struct\\_objeto, 7](#)
- [struct\\_vetor, 7](#)
- [Tabuleiro](#)
  - [rkerender.h, 15](#)
- [vetor](#)
  - [rketypes.h, 18](#)