

Módulo de Física - RKE

Gerado por Doxygen 1.8.0

Sexta, 18 de Maio de 2012 09:33:30



# Sumário

<b>1</b>	<b>Índice dos Componentes</b>	<b>1</b>
1.1	Lista de Componentes . . . . .	1
<b>2</b>	<b>Índice dos Arquivos</b>	<b>3</b>
2.1	Lista de Arquivos . . . . .	3
<b>3</b>	<b>Classes</b>	<b>5</b>
3.1	Referência da Estrutura struct_objeto . . . . .	5
3.1.1	Descrição Detalhada . . . . .	5
3.2	Referência da Estrutura struct_vetor . . . . .	5
3.2.1	Descrição Detalhada . . . . .	6
<b>4</b>	<b>Arquivos</b>	<b>7</b>
4.1	Referência do Arquivo include/rkefisica.h . . . . .	7
4.1.1	Descrição Detalhada . . . . .	7
4.1.2	Funções . . . . .	7
4.1.2.1	rke_adiciona_objeto . . . . .	7
4.1.2.2	rke_get_objeto . . . . .	8
4.1.2.3	rke_set_arrasto . . . . .	8
4.1.2.4	rke_set_delta_t . . . . .	8
4.1.2.5	rke_set_numero_objetos . . . . .	8
4.1.2.6	rke_set_vetor_mundo . . . . .	8
4.1.2.7	rke_simula . . . . .	8
4.2	Referência do Arquivo include/rketypes.h . . . . .	8
4.2.1	Descrição Detalhada . . . . .	9
4.2.2	Definições dos tipos . . . . .	9
4.2.2.1	objeto . . . . .	9
4.2.2.2	vetor . . . . .	9
4.3	Referência do Arquivo src/fisica.c . . . . .	9
4.3.1	Descrição Detalhada . . . . .	10
4.3.2	Funções . . . . .	10
4.3.2.1	itera_posicao . . . . .	10

4.3.2.2	<a href="#">rke_adiciona_objeto</a>	10
4.3.2.3	<a href="#">rke_get_objeto</a>	11
4.3.2.4	<a href="#">rke_set_arrasto</a>	11
4.3.2.5	<a href="#">rke_set_delta_t</a>	11
4.3.2.6	<a href="#">rke_set_numero_objetos</a>	11
4.3.2.7	<a href="#">rke_set_vetor_mundo</a>	11
4.3.2.8	<a href="#">rke_simula</a>	12
4.4	<a href="#">Referência do Arquivo src/main.c</a>	12
4.4.1	<a href="#">Descrição Detalhada</a>	12

# Capítulo 1

## Índice dos Componentes

### 1.1 Lista de Componentes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

<a href="#">struct_objeto</a>	.....	5
<a href="#">struct_vetor</a>	.....	5



## 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="#">rkefisica.h</a>	Arquivo header da biblioteca de funções físicas . . . . .	7
include/ <a href="#">rketypes.h</a>	Arquivo header de tipos e defines do Red Knife Engine . . . . .	8
src/ <a href="#">fisica.c</a>	Esta é a biblioteca de funções que lidam com a física do Red Knife Engine . . . . .	9
src/ <a href="#">main.c</a>	Este é o arquivo que implementa as funções descritas na <a href="#">fisica.c</a> . Aqui, carrega-se um arquivo texto com as condições iniciais e escreve um arquivo "saida.out" com as informações após as iterações . . . . .	12





## Capítulo 3

# Classes

### 3.1 Referência da Estrutura struct\_objeto

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

#### Atributos Públicos

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

#### 3.1.1 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](#)

### 3.2 Referência da Estrutura struct\_vetor

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

### Atributos Públicos

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

### 3.2.1 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/rkefisica.h

Arquivo header da biblioteca de funções físicas.

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)
- void [rke\\_simula](#) ()

#### 4.1.1 Descrição Detalhada

Arquivo header da biblioteca de funções físicas.

##### Autor

João da Silva, Marina Salles, Ricardo Macedo

#### 4.1.2 Funções

4.1.2.1 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

#### 4.1.2.2 objeto rke\_get\_objeto ( int *i* )

Retorna o *i*-ésimo objeto.

##### Parâmetros

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

#### 4.1.2.3 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
---------------------	--------------------------

#### 4.1.2.4 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
------------	-----------------------

#### 4.1.2.5 void rke\_set\_numero\_objetos ( int *numero* )

Configura o número total de objetos a serem simulados.

##### Parâmetros

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

#### 4.1.2.6 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

#### 4.1.2.7 void rke\_simula ( )

Simula todos os objetos por um quanta de tempo.

## 4.2 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:

## Componentes

- 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](#)

### 4.2.1 Descrição Detalhada

Arquivo header de tipos e defines do Red Knife Engine.

#### Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.2.2 Definições dos tipos

#### 4.2.2.1 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

#### 4.2.2.2 typedef struct [struct\\_vetor](#) [vetor](#)

Struct vetor

##### Parâmetros

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

## 4.3 Referência do Arquivo src/fisica.c

Esta é a biblioteca de funções que lidam com a física do Red Knife Engine.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../include/rketypes.h"
Gráfico de dependência de inclusões para fisica.c:
```

## 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 `itera_posicao` (`objeto` \*`obj`, `vetor` `forca`)
- 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`)
- void `rke_simula` ()

## Variáveis

- double `delta_t`
- double `percentual_atrito`
- double `arrasto`
- `vetor` `mundo`
- int `ult_objeto` = 0
- int `num_objetos` = 0
- `objeto` \* `objetos`

### 4.3.1 Descrição Detalhada

Esta é a biblioteca de funções que lidam com a física do Red Knife Engine.

#### Autor

João da Silva, Marina Salles, Ricardo Macedo

### 4.3.2 Funções

#### 4.3.2.1 void `itera_posicao` ( `objeto` \* `obj`, `vetor` `forca` )

De acordo com a velocidade do objeto, o tempo de vida dele, o vetor mundo e a massa, calcula a posição seguinte no próximo quanta de tempo.

#### Parâmetros

<i>obj</i>	Endereço do objeto
<i>forca</i>	Força a ser aplicada

#### 4.3.2.2 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

4.3.2.3 `objeto rke_get_objeto ( int i )`

Retorna o i-ésimo objeto.

## Parâmetros

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

4.3.2.4 `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
---------------------	--------------------------

4.3.2.5 `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
------------	-----------------------

4.3.2.6 `void rke_set_numero_objetos ( int numero )`

Configura o número total de objetos a serem simulados.

## Parâmetros

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

4.3.2.7 `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

#### 4.3.2.8 void rke\_simula ( )

Simula todos os objetos por um quanta de tempo.

## 4.4 Referência do Arquivo src/main.c

Este é o arquivo que implementa as funções descritas na [fisica.c](#). Aqui, carrega-se um arquivo texto com as condições iniciais e escreve um arquivo "saida.out" com as informações após as iterações.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/rketypes.h"
#include "../include/rkefisica.h"
Gráfico de dependência de inclusões para main.c:
```

### Funções

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

#### 4.4.1 Descrição Detalhada

Este é o arquivo que implementa as funções descritas na [fisica.c](#). Aqui, carrega-se um arquivo texto com as condições iniciais e escreve um arquivo "saida.out" com as informações após as iterações.

### Autor

João da Silva, Marina Salles, Ricardo Macedo



# Índice Remissivo

## fisica.c

- itera\_posicao, [10](#)
- rke\_adiciona\_objeto, [10](#)
- rke\_get\_objeto, [11](#)
- rke\_set\_arrasto, [11](#)
- rke\_set\_delta\_t, [11](#)
- rke\_set\_numero\_objetos, [11](#)
- rke\_set\_vetor\_mundo, [11](#)
- rke\_simula, [11](#)

include/rkefisica.h, [7](#)

include/rketypes.h, [8](#)

itera\_posicao  
fisica.c, [10](#)

## objeto

rketypes.h, [9](#)

## rke\_adiciona\_objeto

fisica.c, [10](#)  
rkefisica.h, [7](#)

## rke\_get\_objeto

fisica.c, [11](#)  
rkefisica.h, [7](#)

## rke\_set\_arrasto

fisica.c, [11](#)  
rkefisica.h, [8](#)

## rke\_set\_delta\_t

fisica.c, [11](#)  
rkefisica.h, [8](#)

## rke\_set\_numero\_objetos

fisica.c, [11](#)  
rkefisica.h, [8](#)

## rke\_set\_vetor\_mundo

fisica.c, [11](#)  
rkefisica.h, [8](#)

## rke\_simula

fisica.c, [11](#)  
rkefisica.h, [8](#)

## rkefisica.h

- rke\_adiciona\_objeto, [7](#)
- rke\_get\_objeto, [7](#)
- rke\_set\_arrasto, [8](#)
- rke\_set\_delta\_t, [8](#)
- rke\_set\_numero\_objetos, [8](#)
- rke\_set\_vetor\_mundo, [8](#)
- rke\_simula, [8](#)

## rketypes.h

objeto, [9](#)  
vetor, [9](#)

src/fisica.c, [9](#)

src/main.c, [12](#)

struct\_objeto, [5](#)

struct\_vetor, [5](#)

## vetor

rketypes.h, [9](#)