

Módulo de Física - RKE

0.1

Gerado por Doxygen 1.8.0

Sexta, 18 de Maio de 2012 14:25:32

Sumário

1	Índice das Estruturas de Dados	1
1.1	Estruturas de Dados	1
2	Índice dos Arquivos	3
2.1	Lista de Arquivos	3
3	Estruturas	5
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
4	Arquivos	7
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_conta_objetos	8
4.1.2.3	rke_get_objeto	8
4.1.2.4	rke_set_arrasto	8
4.1.2.5	rke_set_delta_t	8
4.1.2.6	rke_set_numero_objetos	8
4.1.2.7	rke_set_vetor_mundo	8
4.1.2.8	rke_simula	8
4.2	Referência do Arquivo include/rketypes.h	9
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	10
4.3.1	Descrição Detalhada	10
4.3.2	Funções	10

4.3.2.1	itera_posicao	10
4.3.2.2	rke_adiciona_objeto	11
4.3.2.3	rke_conta_objetos	11
4.3.2.4	rke_get_objeto	11
4.3.2.5	rke_set_arrasto	11
4.3.2.6	rke_set_delta_t	11
4.3.2.7	rke_set_numero_objetos	11
4.3.2.8	rke_set_vetor_mundo	12
4.3.2.9	rke_simula	12
4.4	Referência do Arquivo src/main.c	12
4.4.1	Descrição Detalhada	12

Capítulo 1

Índice das Estruturas de Dados

1.1 Estruturas de Dados

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

struct_objeto	5
struct_vetor	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/ rkefisica.h	Arquivo header da biblioteca de funções físicas	7
include/ rketypes.h	Arquivo header de tipos e defines do Red Knife Engine	9
src/ fisica.c	Esta é a biblioteca de funções que lidam com a física do Red Knife Engine	10
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	12

Capítulo 3

Estruturas

3.1 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**

3.1.1 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>
```

Campos de Dados

- 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)
- int [rke_conta_objetos](#) ()
- 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 int rke_conta_objetos ()

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

4.1.2.3 objeto rke_get_objeto (int *i*)

Retorna o *i*-ésimo objeto.

Parâmetros

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

4.1.2.4 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
---------------------	--------------------------

4.1.2.5 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
------------	-----------------------

4.1.2.6 void rke_set_numero_objetos (int *numero*)

Configura o número total de objetos a serem simulados.

Parâmetros

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

4.1.2.7 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

4.1.2.8 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:

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](#)

4.2.1 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)
- int `rke_conta_objetos` ()
- 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

O arrasto é modelado como resistência viscosa, com um número de Reynolds desprezível.

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 int rke_conta_objetos ()

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

4.3.2.4 objeto rke_get_objeto (int *i*)

Retorna o i-ésimo objeto.

Parâmetros

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

4.3.2.5 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
---------------------	--------------------------

4.3.2.6 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
------------	-----------------------

4.3.2.7 void rke_set_numero_objetos (int *numero*)

Configura o número total de objetos a serem simulados.

Parâmetros

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

4.3.2.8 void rke_set_vetor_mundo (double x, double y)

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

Parâmetros

x	Componente x
y	Componente y

4.3.2.9 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_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, 11
 rke_simula, 12

include/rkefisica.h, 7
include/rketypes.h, 9
itera_posicao
 fisica.c, 10

objeto
 rketypes.h, 9

rke_adiciona_objeto
 fisica.c, 10
 rkefisica.h, 7

rke_conta_objetos
 fisica.c, 11
 rkefisica.h, 8

rke_get_objeto
 fisica.c, 11
 rkefisica.h, 8

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, 12
 rkefisica.h, 8

rkefisica.h
 rke_adiciona_objeto, 7
 rke_conta_objetos, 8
 rke_get_objeto, 8
 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, 10
src/main.c, 12
struct_objeto, 5
struct_vetor, 5

vetor
 rketypes.h, 9