Physics

Generated by Doxygen 1.9.2

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Data Structure Documentation

3.1 SistemaEQ_ Struct Reference

Data Fields

- Vetor **vento**
- Vetor **aceleracao**
- Vetor **velocidade**
- Vetor **posicao**
- Vetor **aceleracaoInicial**
- Vetor **velocidadeInicial**
- Vetor **posicaoInicial**
- Tempo **tempoAtual**
- err_t(∗ **calcularAceleracao**)(struct SistemaEQ_, Vetor)
- err_t(∗ calcularPosicao)(struct SistemaEQ_, Vetor)

  Callback para calcular aceleracao
- err_t(∗ calcularVelocidade)(struct SistemaEQ_, Vetor)

  Callback para calcular posicao
- err_t(∗ iniciarLancamento)(struct SistemaEQ_, Angulo, double, Vetor)

  Callback para calcular velocidade
- err_t(∗ sortearVento)(struct SistemaEQ_ ∗)

  Callback para calcular velocidade inicail
- err_t(∗ incrementarTempo)(struct SistemaEQ_ ∗, Tempo)

  Callback para calcular sortear vento
- err_t(∗ calcularTempoNoAr)(struct SistemaEQ_, double ∗)

  Callback para incrementar um dt

The documentation for this struct was generated from the following file:

- physics.h

Chapter 4

File Documentation

4.1 physics.c File Reference

```
#include "physics.h"
```

Functions

- err_t ModuloVetor (const Vetor inputVetor, double *ptrResult)
- err_t criarSistemaEQ (SistemaEQ *ptrSistemaEQ)
- err_t liberarSistemaEQ (SistemaEQ *ptrSistemaEQ)
- err_t iniciarLancamento_ (const SistemaEQ ptrSistemaEQ, const Angulo tetha, const double velocidade↩ Saida, Vetor result)

4.1.1 Function Documentation

4.1.1.1 criarSistemaEQ()

```
err_t criarSistemaEQ (
            SistemaEQ * ptrSistemaEQ )
```

Função que cria uma estrutura com os vetores necessários E preenche as callbacks para as funções com funções padrões Que podem ser modificadas depois

Parameters

in	*ptrSistemaEQ*	Ponteiro para a estrutura de sistemas
out	*errorCode*	

4.1.1.2 iniciarLancamento_()

```
err_t iniciarLancamento_ (
            const SistemaEQ ptrSistemaEQ,
            const Angulo tetha,
            const double velocidadeSaida,
            Vetor result )
```

Função auxiliar padrão que calcula a velocidade inicial do sistema

Parameters

in	*ptrSistemaEQ*	Ponteiro para a estrutura de sistemas
out	*errorCode*	

4.1.1.3 liberarSistemaEQ()

```
err_t liberarSistemaEQ (
            SistemaEQ * ptrSistemaEQ )
```

Função libera toda a estrutura para evitar vazamento de memória

Parameters

in	*ptrSistemaEQ*	Ponteiro para a estrutura de sistemas
out	*errorCode*	

4.1.1.4 ModuloVetor()

```
err_t ModuloVetor (
            const Vetor inputVetor,
            double * ptrResult )
```

Calcula o modulo de um vetor Formula: raiz(x^2 + y^2 + ...^{2})

Parameters

in	*inputVetor*	Vetor de entrada
in	*ptrResult*	Ponteiro para uma variável double que vai armazenar o resultado
out	*errorCode*	

4.2 physics.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <string.h>
#include "global.h"
#include "debug.h"
```

Data Structures

- struct SistemaEQ_

Macros

- #define VECTOR_DIMENSION 2

  Constante que define as dimensões dos vetores.
- #define **X** 0
- #define **Y** 1
- #define GRAVITY 9.80665

  Constante que define o valor da gravidade.
- #define MAX_WIND 20

  Constante que define o range maximo que o vento podera ter.
- #define MIN_WIND 0

  Constante que define o range minimo que o vento podera ter.
- #define PI 3.14159265

  Constante matemática PI.

Typedefs

- typedef double * **Vetor**
- typedef double **Tempo**
- typedef double **Angulo**
- typedef struct SistemaEQ_ **SistemaEQ**

Functions

- err_t criarSistemaEQ (SistemaEQ *ptrSistemaEQ)
- err_t liberarSistemaEQ (SistemaEQ *ptrSistemaEQ)
- err_t ModuloVetor (const Vetor inputVetor, double *ptrResult)

4.2.1 Function Documentation

4.2.1.1 criarSistemaEQ()

```
err_t criarSistemaEQ (
            SistemaEQ * ptrSistemaEQ )
```

Função que cria uma estrutura com os vetores necessários E preenche as callbacks para as funções com funções padrões Que podem ser modificadas depois

Parameters

in	*ptrSistemaEQ*	Ponteiro para a estrutura de sistemas
out	*errorCode*	

4.2.1.2 liberarSistemaEQ()

```
err_t liberarSistemaEQ (
            SistemaEQ * ptrSistemaEQ )
```

Função libera toda a estrutura para evitar vazamento de memória

Parameters

in	*ptrSistemaEQ*	Ponteiro para a estrutura de sistemas
out	*errorCode*	

4.2.1.3 ModuloVetor()

```
err_t ModuloVetor (
            const Vetor inputVetor,
            double * ptrResult )
```

Calcula o modulo de um vetor Formula: raiz(x^2 + y^2 + ...^{2})

Parameters

in	*inputVetor*	Vetor de entrada
in	*ptrResult*	Ponteiro para uma variável double que vai armazenar o resultado
out	*errorCode*	