Referência para loja\_digital\_OS

Por Bruno Dalagnol, todos os direitos reservados exclusivamente a Bruno Dalagnol.

Criado em 7 de Maio de 2021.

Atualizado em 7 de Maio de 2021.

Sumário

1. Arquivos (Pg. 3).

* Por ordem de inserção
* Por nível de acesso

1. Vector.h(Pg. 3).
2. Alloc\_functions.h(Pg. 4).
3. File\_list.h(Pg. 4).
4. Graphic.h(Pg. 5).
5. Draw\_manager.h(Pg. 6).
6. Graphic\_constructor.h(Pg. 0).
7. Graphic\_methods.h(Pg. 0).
8. Graphic\_draw.h(Pg. 0).
9. Graphic\_manager.h(Pg. 0).

Arquivos

A seção de arquivos aborda os arquivos usados para construir o sistema, ela é separada em duas partes as quais descrevem os mesmos e suas funcionalidades, a primeira parte mostra a construção hierárquica dos arquivos e sua sequência de compilação, já a segunda parte descreve brevemente a funcionalidade de cada arquivo, sendo separados pelo seu nível de construção e acesso ao arquivo principal.

**Por ordem de inserção**

**Por nível de acesso**

**vector.h**

Cria funções que são usadas como argumento em métodos para informar as cores RGB (vector3) e para as configurações de posicionamento, borda, margem, espaçamento e outros (vector4).

Includes:

#include <iostream>

#include <cstdlib>

Funções:

template <class T> T\* vector4(T p1, T p2, T p3, T p4);

template <class T> T\* vector4(T p1, T p2);

template <class T> T\* vector4(T p);

template <class T> T\* vector3(T p1, T p2, T p3);

template <class T> T\* vector3(T p);

**alloc\_functions.h**

Cria funções que são usadas internamente para alocar e gerenciar vetores de dados, bem como funções para verificar possíveis erros e tratá-los devolvendo uma informação ao console, e caso os erros forem de mais alto nível, trazendo uma mensagem direta ao usuário.

Includes:

#include <iostream>

#include <cstdlib>

#include <cstring>

using namespace std;

Funções:

void alloc\_memory\_error(void);

unsigned int alloc\_str\_vector(char\*\*\* buffer, unsigned int lenght, const char\* str);

template <class T> unsigned int alloc\_vector(T\*\* buffer, unsigned int lenght, T value);

**file\_list.h**

Cria classes para uso interno que terão como propósito armazenar arquivos de forma simplificada.

Includes:

#include "alloc\_functions.h"

Classes:

class File\_list;

class Font\_list : public File\_list;

Variáveis internas:

File\_list::char\*\* files = nullptr;

File\_list::char\*\* ids = nullptr;

File\_list::unsigned int lenght = 0;

Font\_list::unsigned int\* sizes = nullptr;

Font\_list::int\* flags = nullptr;

Métodos:

void File\_list::add(const char\* file, const char\* id);

File\_list::~File\_list();

void Font\_list::add\_font(const char\* font, const char\* id, unsigned int size, int flags);

Font\_list::~Font\_list();

**graphic.h**

Cria uma classe global para uso exclusivo a criar e gerenciar o sistema gráfico interno, bem como alocar todos os tipos de arquivos quaisquer que sejam eles (imagens, fontes, etc.).

Includes:

#include <allegro5/allegro.h>

#include <allegro5/allegro\_native\_dialog.h>

#include <allegro5/allegro\_primitives.h>

#include <allegro5/allegro\_ttf.h>

#include <allegro5/allegro\_font.h>

#include <allegro5/allegro\_color.h>

#include <allegro5/allegro\_image.h>

#include "file\_list.h"

Classes:

class Allegro\_routines;

Variáveis internas:

Allegro\_routines::int display\_size[2] = { 0, 0 };

Allegro\_routines::int fps = 0;

Allegro\_routines::bool runing = true;

Allegro\_routines::ALLEGRO\_DISPLAY\* display = nullptr;

Allegro\_routines::ALLEGRO\_EVENT evnt;

Allegro\_routines::ALLEGRO\_EVENT\_QUEUE\* event\_queue = nullptr;

Allegro\_routines::ALLEGRO\_EVENT mouse\_evnt;

Allegro\_routines::ALLEGRO\_EVENT\_QUEUE\* mouse\_event\_queue = nullptr;

Allegro\_routines::ALLEGRO\_TIMER\* timer = nullptr;

Allegro\_routines::ALLEGRO\_BITMAP\*\* image = nullptr;

Allegro\_routines::char\*\* image\_id = nullptr;

Allegro\_routines::unsigned int image\_lenght = 0;

Allegro\_routines::ALLEGRO\_FONT\*\* font = nullptr;

Allegro\_routines::char\*\* font\_id = nullptr;

unsigned int font\_lenght = 0;

Métodos:

Allegro\_routines::Allegro\_routines(int display\_x, int display\_y, double fps, File\_list\* image\_file\_list = nullptr, Font\_list\* font\_file\_list = nullptr);

Allegro\_routines::bool \_flip();

Allegro\_routines::bool \_timed();

Allegro\_routines::void \_add\_images(File\_list\* file\_list);

Allegro\_routines::void \_add\_fonts(Font\_list\* file\_list);

**file\_list.h**

Cria classes para uso interno que terão como propósito armazenar arquivos de forma simplificada.

Includes:

#include "alloc\_functions.h"

Classes:

class File\_list;

class Font\_list : public File\_list;

Variáveis internas:

File\_list::char\*\* files = nullptr;

File\_list::char\*\* ids = nullptr;

File\_list::unsigned int lenght = 0;

Font\_list::unsigned int\* sizes = nullptr;

Font\_list::int\* flags = nullptr;

Métodos:

void File\_list::add(const char\* file, const char\* id);

File\_list::~File\_list();

void Font\_list::add\_font(const char\* font, const char\* id, unsigned int size, int flags);

Font\_list::~Font\_list();