

Classe imagem

imagem::imagem (void)

Cria uma imagem.

void imagem::carregar (string *arquivo*)

Carrega do arquivo de imagem (bmp, png, tga, jpg, gif, psd, hdr, pic)

Parâmetros:

<i>arquivo</i>	Caminho do arquivo da imagem a ser carregado
----------------	--

void imagem::setVelocidade (unsigned int *velocidade*)

Define a velocidade dos frames.

Parâmetros:

<i>velocidade</i>	valor da velocidade dos frames
-------------------	--------------------------------

int imagem::getVelocidade ()

Retorna a velocidade definida para os frames.

int imagem::getResX ()

Retorna a largura da imagem.

int imagem::getResY ()

Retorna a altura da imagem.

void imagem::desenha (int *x*, int *y*, unsigned int *angulo* = 0, bool *borda* = false)

Desenha a imagem.

Parâmetros:

<i>x</i>	posição X
<i>y</i>	posição Y
<i>angulo</i>	angulo
<i>borda</i>	desenhar borda na imagem

bool imagem::colide ([imagem](#) *img*)

Verifica se houve colisão com outra imagem.

Parâmetros:

<i>img</i>	Imagem a ser verificada se houve colisão
------------	--

int imagem::getPosX ()

Retorna a posição da imagem no eixo X.

int imagem::getPosY ()

Retorna a posição da imagem no eixo Y.

Classe musica

musica::musica (void)

Cria uma música.

void musica::carregar (string *arquivo*)

Carrega um arquivo de música (ogg, wav, flac, aiff, au, raw, paf, svx, nist, voc, ircam, w64, mat4, mat5 pvf, htk, sds, avr, sd2, caf, wve, mpc2k, rf64).

Parâmetros:

<i>arquivo</i>	Caminho do arquivo da música a ser carregado.
----------------	---

void musica::tocar (int *volume* = 10, bool *repetir* = false)

Começa a tocar a música.

Parâmetros:

<i>volume</i>	Define o volume da música.
<i>repetir</i>	Define se a música deve repetir

void musica::pausar ()

Pausa a música.

void musica::parar ()

Para de tocar a música.

Classe som

som::som (void)

Cria um arquivo de som.

void som::carregar (string *arquivo*)

Carrega um arquivo de som (ogg, wav, flac, aiff, au, raw, paf, svx, nist, voc, ircam, w64, mat4, mat5 pvf, htk, sds, avr, sd2, caf, wve, mpc2k, rf64)

Parâmetros:

<i>arquivo</i>	Caminho do arquivo de som a ser carregado
----------------	---

void som::tocar (int *volume* = 10, bool *repetir* = false)

Começa a tocar o som.

Parâmetros:

<i>volume</i>	Define o volume do som.
<i>repetir</i>	Define se o som deve repetir

void som::pausar ()

Pausa o som.

void som::parar ()

Para de tocar o som.

Classe texto

texto::texto (void)

Cria um texto.

void texto::carregar (string *arquivo*)

Carrega um arquivo de fonte para o texto (TrueType, Type 1, CFF, OpenType, SFNT, X11 PCF, Windows FNT, BDF, PFR, Type 42).

Parâmetros:

<i>arquivo</i>	Caminho do arquivo de fonte a ser carregado.
----------------	--

void texto::desenha (string *texto*, int *x*, int *y*, unsigned int *tamanho* = 10, unsigned int *R* = 255, unsigned int *G* = 255, unsigned int *B* = 255, unsigned int *A* = 255, bool *negrito* = false, bool *italico* = false, bool *sublinhado* = false)

Desenha o texto.

Parâmetros:

<i>texto</i>	texto a ser desenhado
<i>x</i>	posição X
<i>y</i>	posição Y
<i>tamanho</i>	tamanho da fonte
<i>R</i>	Cor vermelho (0 - 255)
<i>G</i>	Cor verde (0 - 255)
<i>B</i>	Cor azul (0 - 255)
<i>A</i>	Nível de transparencia (0 - 255)
<i>negrito</i>	Define se é negrito ou não
<i>italico</i>	Define se é itálico ou não
<i>sublinhado</i>	Define se é sublinhado ou não