

# *Introdução à Allegro*

Tiago Alves Macambira  
em 20/05/2004

# *Introdução<sup>2</sup>*

- ◆ Onde pegar a allegro:
  - ◆ <http://alleg.sf.net>
- ◆ Onde pegar a allegro PARA o Dev-C++
  - ◆ <http://www.devpaks.org>
- ◆ E onde pegar o original desse tutorial?
  - ◆ [http://www.loomsoft.net/resources/alltut/alltut\\_inc](http://www.loomsoft.net/resources/alltut/alltut_inc)

# *Iniciando*

- ◆ Antes de poder usar qualquer coisa da Allegro, vc deve iniciá-la:
  - ◆ `allegro_init()`
- ◆ O allegro possui suporte a vários dispositivos, tais como, mouse, som, gráficos, joystick
  - ◆ O suporte a cada um desses dispositivos deve ser instalado
  - ◆ `install_<dispositivo>`

## *Iniciando (2/2)*

### ◆ Vamos ao código:

```
#include <allegro.h>
```

```
int main(int argc, char* argv[])
```

```
{  
    allegro_init();  
    install_keyboard();  
    set_gfx_mode(GFX_AUTODETECT_WINDOWED,640,480,0,0);  
    readkey(); // Espera uma tecla ser pressionada para...  
    return(0); // sair...  
}  
END_OF_MAIN();
```

# Gráficos (ou “o que importa é a aparência”)

- ◆ Boa parte do seu trabalho será desenhar imagens (.bmp) ou *sprites*
  - ◆ Você pode utilizar as funções de desenho de figuras geométricas básicas para fazer tudo. O problema é seu.
- ◆ Profundidade de cores
  - ◆ Para colocar imagens na tela deve-se antes especificar a profundidade de cores da tela (*color depth*): 8, 15, 16, 24, 32 bits
  - ◆ Você deve informar a profundidade de cores **antes** de iniciar a própria tela

# *Gráficos (ou “o que importa é a aparência”) (2)*

## ◆ Imagens

### ◆ Declarando-as:

- ◆ BITMAP\* imagem;

### ◆ Carregando-as:

- ◆ `imagem = load_bitmap(“bola.bmp”,NULL);`

- ◆ Pode-se omitir a palheta de cores utilizada passando-se NULL

### ◆ **Cuidado**

- ◆ Se a imagem não estiver acessível no diretório atual o programa travará

# *Gráficos (ou “o que importa é a aparência”) (3)*

- ◆ Desenhando Imagens
  - ◆ Lembre-se: antes de iniciar o processo de desenho de várias imagens adquira a posse da tela e depois libere-a:
    - ◆ `acquire_screen()`
    - ◆ `release_screen()`
  - ◆ Use a função `blit` para desenhar coisas na tela:
  - ◆ `blit(o que, onde, clip x, clip y, posição x , posição y, largura, altura)`
- ◆ Liberando memória:
  - ◆ Após o uso, destrua as imagens com `destroy_bitmap()`

# *Gráficos (ou “o que importa é a aparência”) (4)*

## ◆ Ao Código:

```
int main(int argc, char* argv[])
{
    BITMAP* imagem;

    allegro_init();
    install_keyboard();
    set_color_depth(16);
    set_gfx_mode(GFX_AUTODETECT_WINDOWED,640,480,0,0);
    imagem = load_bitmap("bola.bmp",NULL);
    acquire_screen();
    blit(imagem, screen, 0,0,0,0,20,20);
    release_screen();
    readkey();
    destroy_bitmap(imagem);
    return(0);
}
END_OF_MAIN();
```

# *Transparências*

- ◆ Quando desenhamos elementos do jogo, queremos que eles “casem” com o fundo (*background*)
- ◆ A cor RGB (255,0,255) é a cor “transparente” no allegro
- ◆ Podemos desenhar imagens com transparência usando:
  - ◆ `masked_blit(o que, onde, ...)` // igual a `blit`
  - ◆ `draw_sprite(onde, o que, x, y)`

# *Movendo coisas na tela*

- ◆ Variável global **key[]**
  - ◆ permite ver quais teclas estão pressionadas.
  - ◆ Podemos usá-lo dentro de ifs
    - ◆ `if (key[KEY_UP]) ...`

# *Movendo coisas na tela (2)*

## ◆ Código de exemplo:

```
void processa_teclado(int* x, int* y){
    if (key[KEY_UP]){
        *y = *y + 1;
    } else if (key[KEY_DOWN]){
        *y = *y - 1;
    } else if (key[KEY_RIGHT]){
        *x = *x + 1;
    } else if (key[KEY_LEFT]){
        *x = *x - 1;
    }
}
```

# Movendo coisas na tela (3)



```
int main(int argc, char* argv[])
{
    int x = 0;
    int y = 0;
    ...
    imagem = load_bitmap("bola.bmp",NULL);
    while(! key[KEY_ESC]){
        processa_teclado(&x,&y);
        acquire_screen();
        draw_sprite(screen,imagem,x,y);
        release_screen();
        clear_bitmap(screen);
    }
    destroy_bitmap(imagem);
    return(0);
}
END_OF_MAIN();
```

*Fim ?*

◆ Por enquanto sim...