



Tutoriais Allegro 5

Por: Humberto Vieira e Mario de Castro

Instalação

Janela

Primitives

Imagens

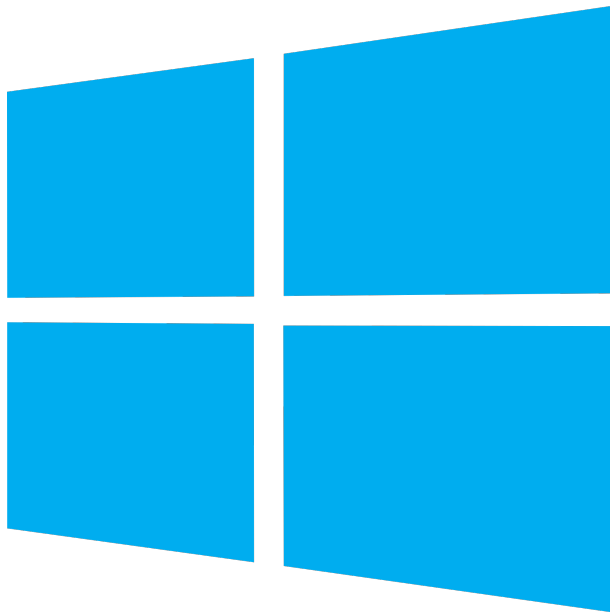
Eventos + Teclado

Introdução à Structs

Organização do código



Instalação



Fix by Proximity

WikiAllegro



Rafael Toledo

Wiki Allegro



Wiki Allegro

Janela

```
8  /*Inclui Biblioteca Allegro*/  
9  #include <allegro5/allegro.h>
```

Criação da janela

Declaração

Inicialização

Definindo o tamanho

Tratando erros

Retirando da memória

```
3  
4  /*Declaração e Inicialização da Variável de Janela*/  
5  ALLEGRO_DISPLAY *janela = NULL;  
6
```

```
13  
14  /*Inicializa Allegro5*/  
15  al_init();  
16
```

Janela

```
1  /*Define tamanho da Tela*/  
2  #define LARG 1280  
3  #define ALT 720  
4
```

Criação da janela

```
13  
14  /*Inicializa Allegro5*/  
15  if(!al_init()) {  
16      fprintf(stderr, "Falha ao iniciar Allegro5!\n");  
17      return -1;  
18  }  
19
```

Declaração

Inicialização

```
21  
22  /*Cria Janela de Jogo*/  
23  janela = al_create_display(1280, 720);  
24  if (!janela)  
25  {  
26      fprintf(stderr, "Falha ao criar janela!\n");  
27      return -1;  
28  }  
29
```

Definindo o tamanho

Tratando erros

Retirando da memória

Janela

```
4  /*Destroi Variavel de Janela*/  
5  al_destroy_display(janela);  
6
```

Retirando da memória

```
7  /*Preenche a tela com a cor desejada*/  
8  al_clear_to_color(al_map_rgb(0, 0, 0));  
9
```

Alterando o fundo

```
10 /*Atualiza tela*/  
11 al_flip_display();  
12
```

Atualizando a tela

```
13 /*Pausa execucao*/  
14 al_rest(5);  
15
```

Pausando a tela

Resultado final

Janela

```
5
6 #include <stdio.h>
7
8 /*Inclui Biblioteca Allegro*/
9 #include <allegro5/allegro.h>
10
11 /*Define tamanho da Tela*/
12 #define LARG 1280
13 #define ALT 720
14
15 int main(void) {
16
17     /*Declaracao e Inicializacao da Variavel de Janela*/
18     ALLEGRO_DISPLAY *janela = NULL;
19
20     /*Inicializa Allegro5*/
21     if(!al_init()) {
22         fprintf(stderr, "Falha ao iniciar Allegro5!\n");
23         return -1;
24     }
25
26     /*Cria Janela de Jogo*/
27     janela = al_create_display(LARG, ALT);
28     if (!janela)
29     {
30         fprintf(stderr, "Falha ao criar janela!\n");
31         return -1;
32     }
33
34     /*Preenche a tela com a cor desejada*/
35     al_clear_to_color(al_map_rgb(0, 0, 0));
36
37     /*Atualiza tela*/
38     al_flip_display();
39
40     /*Pausa execucao*/
41     al_rest(5);
42
43     /*Destroi Variavel de Janela*/
44     al_destroy_display(janela);
45
46     return 0;
47 }
```

Retirando da memória

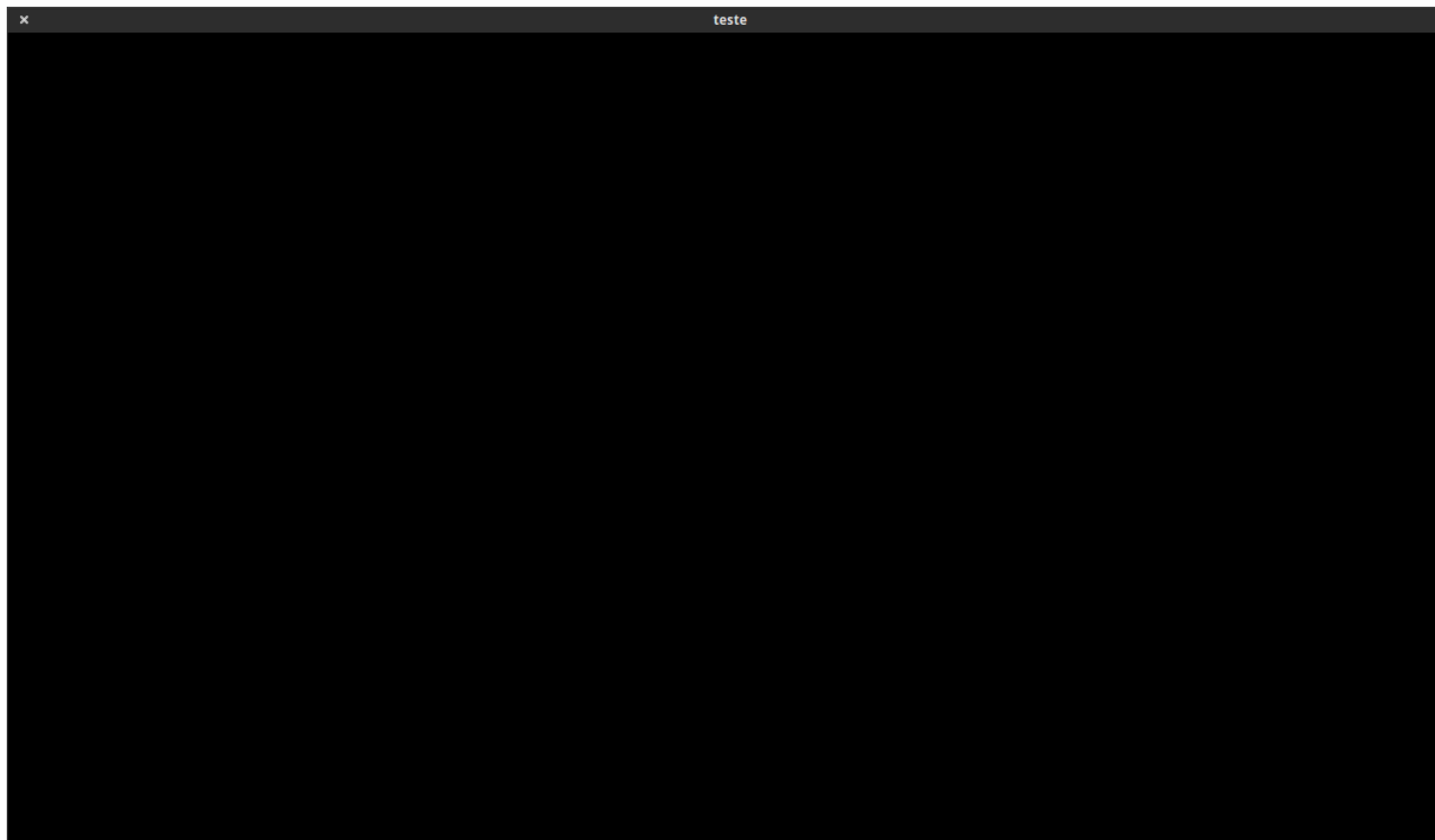
Alterando o fundo

Atualizando a tela

Pausando a tela

Resultado final

Janela



Retirando da memória

Alterando o fundo

Atualizando a tela

Pausando a tela

Resultado final

Primitives

```
/*Inclui biblioteca do Allegro Primitives*/  
#include <allegro5/allegro_primitives.h>
```

```
/*Inicia extensão de Primitives*/  
al_init_primitives_addon();
```

```
/*Exemplos de Primitives*/  
al_draw_line(400, 300, 600, 100, al_map_rgb(0, 255, 0), 3);  
al_draw_filled_triangle(800, 100, 700, 200, 850, 300, al_map_rgb(255, 255, 0));  
al_draw_rounded_rectangle(300, 500, 500, 600, 30, 30, al_map_rgb(255, 0, 0), 4);  
al_draw_filled_circle(800, 500, 75, al_map_rgb(0, 0, 255));
```

Importando

Inicializando

Exemplos

Resultado final

Visualização

Primitives

```
#include <stdio.h>
#include <allegro5/allegro.h>

/*Inclui biblioteca do Allegro Primitives*/
#include <allegro5/allegro_primitives.h>

#define LARG 1280
#define ALT 720

int main(void) {
    ALLEGRO_DISPLAY *janela = NULL;

    if(!al_init()) {
        fprintf(stderr, "Falha ao iniciar Allegro5!\n");
        return -1;
    }

    janela = al_create_display(LARG, ALT);
    if (!janela)
    {
        fprintf(stderr, "Falha ao criar janela!\n");
        return -1;
    }

    /*Inicia extensão de Primitives*/
    al_init_primitives_addon();

    al_clear_to_color(al_map_rgb(0, 0, 0));

    /*Exemplos de Primitives*/
    al_draw_line(400, 300, 600, 100, al_map_rgb(0, 255, 0), 3);
    al_draw_filled_triangle(800, 100, 700, 200, 850, 300, al_map_rgb(255, 255, 0));
    al_draw_rounded_rectangle(300, 500, 500, 600, 30, 30, al_map_rgb(255, 0, 0), 4);
    al_draw_filled_circle(800, 500, 75, al_map_rgb(0, 0, 255));

    al_flip_display();

    al_rest(5);

    al_destroy_display(janela);
    return 0;
}
```

Importando

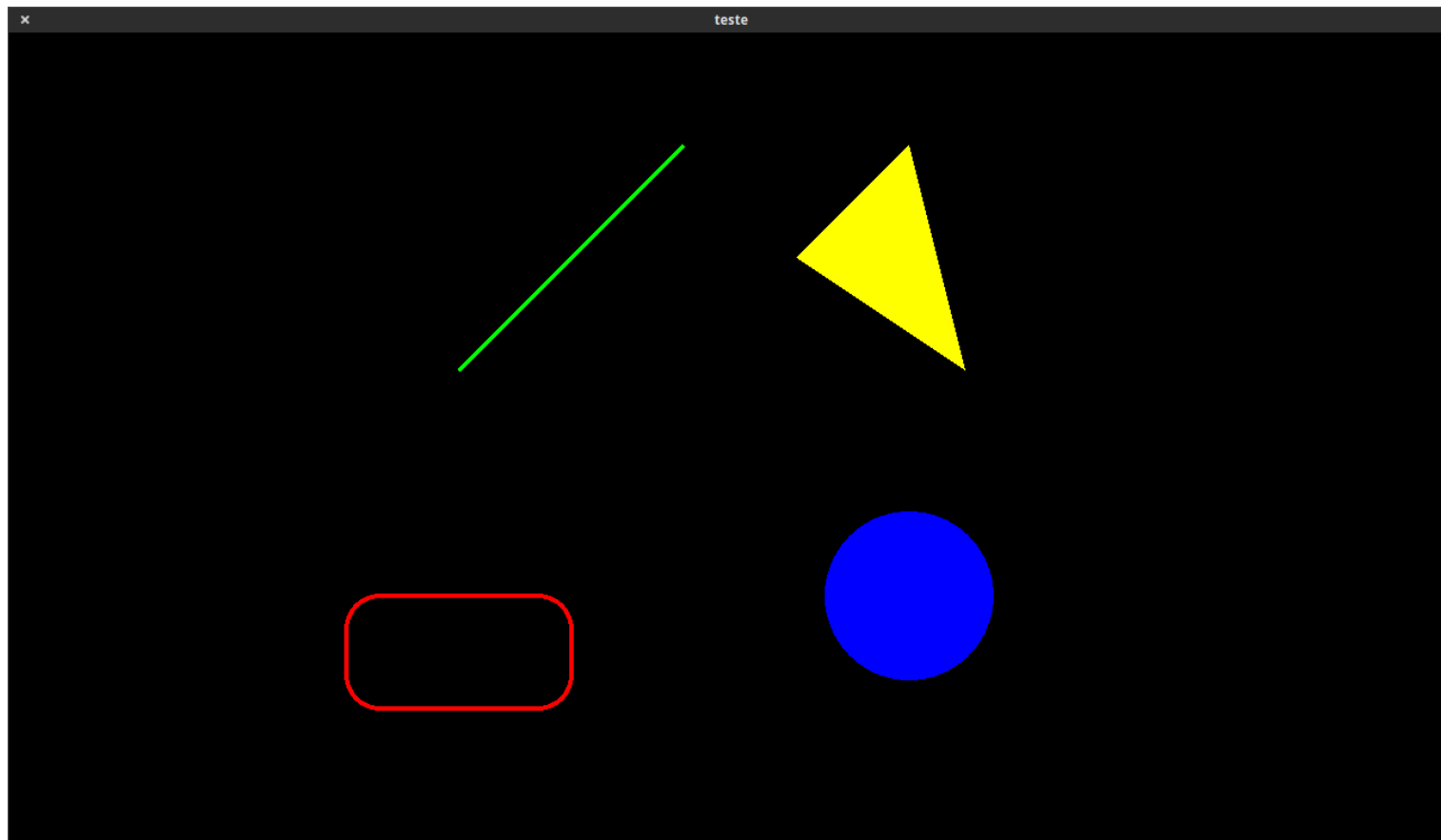
Inicializando

Exemplos

Resultado final

Visualização

Primitives



Importando

Inicializando

Exemplos

Resultado final

Visualização

Imagens

```
/*Inclui biblioteca de imagens allegro*/  
#include <allegro5/allegro_image.h>
```

```
/*Declara e inicializa variavel de imagem*/  
ALLEGRO_BITMAP *imagem = NULL;
```

```
/*Inicia extensao de imagens*/  
al_init_image_addon();
```

```
/*Carrega imagem*/  
imagem = al_load_bitmap("imagen.png");  
if (!imagem)  
{  
    fprintf(stderr, "Falha ao carregar imagem!\n");  
    return -1;  
}
```

Importação

Declaração/Inicialização

Tratamento de erros

Exemplo

Resultado final

Visualização

Imagens

```
/*Desenha a imagem*/  
al_draw_bitmap(imagem, LARG / 2 - (al_get_bitmap_width(imagem) / 2),  
    ALT / 2 - (al_get_bitmap_height(imagem) / 2), 0);
```

```
/*Destroi Variavel de imagem*/  
al_destroy_bitmap(imagem);
```

Importação

Declaração/Inicialização

Tratamento de erros

Exemplo

Retirando da memória

Resultado final

Visualização

Imagens

```
#include <stdio.h>
#include <allegro5/allegro.h>

/*Inclui biblioteca de imagens allegro*/
#include <allegro5/allegro_image.h>

#define LARG 1280
#define ALT 720

int main(void) {
    ALLEGRO_DISPLAY *janela = NULL;

    /*Declara e inicializa variavel de imagem*/
    ALLEGRO_BITMAP *imagem = NULL;

    if(!al_init()) {
        fprintf(stderr, "Falha ao iniciar Allegro5!\n");
        return -1;
    }

    janela = al_create_display(LARG, ALT);
    if (!janela)
    {
        fprintf(stderr, "Falha ao criar janela!\n");
        return -1;
    }

    /*Inicia extensao de imagens*/
    al_init_image_addon();

    /*Carrega imagem*/
    imagem = al_load_bitmap("imagem.png");
    if (!imagem)
    {
        fprintf(stderr, "Falha ao carregar imagem!\n");
        return -1;
    }

    al_clear_to_color(al_map_rgb(0, 0, 0));

    /*Desenha a imagem*/
    al_draw_bitmap(imagem, LARG / 2 - (al_get_bitmap_width(imagem) / 2), ALT / 2 - (al_get_bitmap_height(imagem) / 2), 0);

    al_flip_display();

    al_rest(5);

    al_destroy_display(janela);

    /*Destroi Variavel de imagem*/
    al_destroy_bitmap(imagem);

    return 0;
}
```

Importação

Declaração/Inicialização

Tratamento de erros

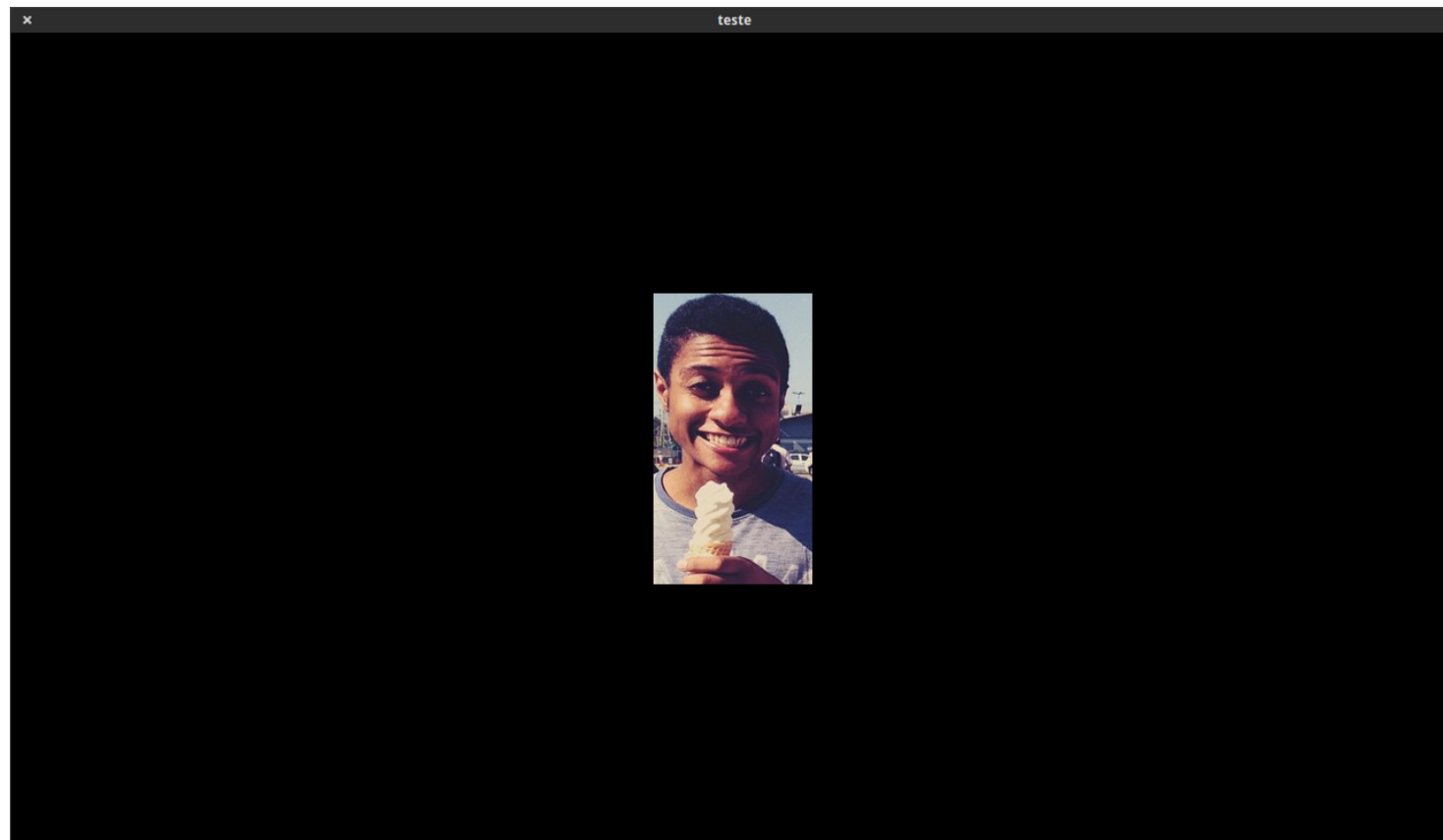
Exemplo

Retirando da memória

Resultado final

Visualização

Imagens



Importação

Declaração/Inicialização

Tratamento de erros

Exemplo

Retirando da memória

Resultado final

Visualização

Fila de eventos + Teclado

```
/*Declara e inicializa variavel da fila de eventos*/  
ALLEGRO_EVENT_QUEUE *fila_de_eventos = NULL;
```

```
/*Instala extensoes de mouse e teclado*/  
al_install_keyboard();  
al_install_mouse();
```

```
/*Cria fila de eventos*/  
fila_de_eventos = al_create_event_queue();  
if (!fila_de_eventos)  
{  
    fprintf(stderr, "Falha ao criar fila de eventos!\n");  
    return -1;  
}
```

Declaração

Inicialização

Tratamento de erros

Exemplos

Criação de eventos

Evento do teclado

Retirando da memória

Fila de eventos + Teclado

```
/*Registra fontes de eventos*/  
al_register_event_source(fila_de_eventos, al_get_keyboard_event_source());  
al_register_event_source(fila_de_eventos, al_get_mouse_event_source());  
al_register_event_source(fila_de_eventos, al_get_display_event_source(janela));
```

Declaração

```
/*Controle do Loop Principal*/  
int fim_do_jogo = 0;
```

Inicialização

```
/*Loop principal do Jogo*/  
while (!fim_do_jogo) {  
  
}
```

Tratamento de erros

Exemplos

Criação de eventos

```
/*Cria variavel de evento e a associa a fila de eventos*/  
ALLEGRO_EVENT evento;  
al_wait_for_event(fila_de_eventos, &evento);
```

Evento do teclado

Retirando da memória

Fila de eventos + Teclado

```
/*Verifica se a janela foi fechada*/  
if (evento.type == ALLEGRO_EVENT_DISPLAY_CLOSE) {  
    fim_do_jogo = 1;  
}
```

```
/*Verifica pressionamento de tecla do teclado*/  
else if (evento.type == ALLEGRO_EVENT_KEY_DOWN) {  
    /*Verifica qual tecla foi pressionada*/  
    switch (evento.keyboard.keycode)  
    {  
        case ALLEGRO_KEY_ESCAPE:  
            fim_do_jogo = 1;  
            break;  
        case ALLEGRO_KEY_ENTER:  
            al_draw_bitmap(imagem, LARG / 2 - (al_get_bitmap_width(imagem) / 2),  
                ALT / 2 - (al_get_bitmap_height(imagem) / 2), 0);  
            al_flip_display();  
            fim_do_jogo = 1;  
            break;  
    }  
}
```

```
/*Destroi Variavel da fila de eventos*/  
al_destroy_event_queue(fila_de_eventos);
```

Declaração

Inicialização

Tratamento de erros

Exemplos

Criação de eventos

Evento do teclado

Retirando da memória

Introdução à Structs

```
3 struct exemplo{  
4     int numero = 4;  
5     float qut = 3.4;  
6     char sexp = 'm';  
7     ALLEGRO_BITMAP *imagem = NULL;  
8 };|
```

Criação

Apelidos

```
| struct exemplo nome;|
```

Colocando valores

```
3 typedef struct {  
4     int numero = 4;  
5     float qut = 3.4;  
6     char sexp = 'm';  
7     ALLEGRO_BITMAP *imagem = NULL;  
8 }exemplo;  
9  
0 exemplo nome;
```

Introdução à Structs

```
1  
2 nome.numero = 10;  
3 nome.sexo = 's';
```

Criação

Apelidos

```
3 exemplo jobson;  
4 exemplo *doisberto = &jobson;  
5  
6 (*doisberto).numero = 10;  
7 doisberto->numero = 10;
```

Colocando valores

Organização do código

```
int main(void) {  
    /*  
    - Declaracao e Inicializacao de variaveis  
    - Inicializacao de elementos do Allegro  
    - Criacao de janelas, imagens, sons, etc.  
    - Criacao de fila de eventos  
    - Outras funcoes de inicialização  
    */  
    while () {  
        /*  
        LOOP DE JOGO  
  
        - Executar funcoes da logica do jogo  
        - Verificar eventos e inputs do usuário  
        - Desenhar a tela  
        */  
    }  
    /*  
    - Finalização do programa  
    - Liberação de memoria  
    */  
}
```

Explicação por partes

Criando structs

Inicializando structs

Colocando eventos

Renderização

Retirando da memória

Organização do código

```
/*Struct da imagem*/  
typedef struct {  
    int pos_x;  
    int pos_y;  
  
    ALLEGRO_BITMAP *imagem;  
}jobson;
```

```
/*Declaracao da struct*/  
objeto jobson;
```

```
/*Inicializa struct*/  
jobson.pos_x = LARG / 2;  
jobson.pos_y = ALT / 2;  
jobson.img = imagem;
```

Explicação por partes

Criando structs

Inicializando structs

Colocando eventos

Renderização

Retirando da memória

Organização do código

```
/*verifica pressionamento de tecla do teclado*/  
else if (evento.type == ALLEGRO_EVENT_KEY_DOWN) {  
    /*Verifica qual tecla foi pressionada*/  
    switch (evento.keyboard.keycode)  
    {  
        case ALLEGRO_KEY_ESCAPE:  
            fim_do_jogo = 1;  
            break;  
        case ALLEGRO_KEY_UP:  
            jobson.pos_y -= 10;  
            break;  
        case ALLEGRO_KEY_DOWN:  
            jobson.pos_y += 10;  
            break;  
        case ALLEGRO_KEY_LEFT:  
            jobson.pos_x -= 10;  
            break;  
        case ALLEGRO_KEY_RIGHT:  
            jobson.pos_x += 10;  
            break;  
    }  
}
```

Explicação por partes

Criando structs

Inicializando structs

Colocando eventos

Renderização

Retirando da memória

Organização do código

```
al_draw_bitmap(jobson.img, jobson.pos_x - (al_get_bitmap_width(jobson.img) / 2),  
    jobson.pos_y - (al_get_bitmap_height(jobson.img) / 2), 0);  
al_flip_display();  
al_clear_to_color(al_map_rgb(0, 0, 0));
```

Explicação por partes

```
/*controle de renderização da tela*/  
int renderizar = 1;
```

Criando structs

Inicializando structs

```
/*Controle de frames por segundo*/  
const int FPS = 60;
```

Colocando eventos

Renderização

Retirando da memória

Organização do código

```
/*Declara e inicializa variavel de timer*/  
ALLEGRO_TIMER *timer = NULL;
```

Explicação por partes

```
/*Cria timer de controle*/  
timer = al_create_timer(1.0 / FPS);  
if (!timer)  
{  
    fprintf(stderr, "Falha ao criar timer!\n");  
    return EXIT_FAILURE;  
}
```

Criando structs

Inicializando structs

```
/*Registra Timer como fonte de eventos*/  
al_register_event_source(fila_de_eventos,  
    al_get_timer_event_source(timer));
```

Colocando eventos

Renderização

Retirando da memória

Organização do código

```
/*Inicia Timer*/  
al_start_timer(timer);
```

Explicação por partes

```
if (evento.type == ALLEGRO_EVENT_TIMER)  
{  
    /*Permite Renderização da tela*/  
    renderizar = true;  
}
```

Criando structs

Inicializando structs

Colocando eventos

```
if (renderizar) {  
    renderizar = 0;  
    al_draw_bitmap(jobson.img, jobson.pos_x - (al_get_bitmap_width(jobson.img) / 2),  
        jobson.pos_y - (al_get_bitmap_height(jobson.img) / 2), 0);  
    al_flip_display();  
    al_clear_to_color(al_map_rgb(0, 0, 0));  
}
```

Renderização

Retirando da memória

```
/*Destroi variavel de timer*/  
al_destroy_timer(timer);
```

Tutoriais motherfuckers

Tudo em videos: Fix By Proximity

O mito: Rafael Toledo

A Biblia: Reference Manual

Obrigado! =)