

SDL

- *Simple DirectMedia Layer*
- Acesso “baixo nível” ao áudio, teclado, mouse, joystick, gráficos, etc.
- Jogos, vídeos e aplicações gráficas
- Multi-plataforma:
 - *Windows, Linux, MacOS*
 - *Android, iOS*
- Implementado em C
 - Bibliotecas para *Python, Lua*, etc.
 - Game engines: *Pygame, Löve*

Jogos

- Código aberto
 - *Freeciv, SuperTux*
- Independente (Indie)
 - *World of Goo, FEZ*
- Profissional (AAA)
 - *Quake 4, Unreal*



Artigos & Videos - 04

- Open Source Tools for Game Development

- <https://www.youtube.com/watch?v=r3wDn0Ajrtk>

- Indie Game: The Movie

- <http://vimeo.com/ondemand/indiegamethemovie/25268139>

Hello World

```
#include <SDL2/SDL.h>
int main (int argc, char* args[])
{
    /* INITIALIZATION */
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Window* window = SDL_CreateWindow(
        "Hello World!",
        SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED,
        640, 480,
        SDL_WINDOW_OPENGL);

    SDL_Renderer* renderer = SDL_CreateRenderer(
        window, -1,
        SDL_RENDERER_ACCELERATED);

    /* EXECUTION */
    SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
    SDL_RenderFillRect(renderer, &r);
    SDL_Rect r = { 200, 200, 100, 100 };
    SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
    SDL_RenderFillRect(renderer, &r);
    SDL_Delay(5000);

    /* FINALIZATION */
    SDL_DestroyRenderer(renderer);
    SDL_DestroyWindow(window);
    SDL_Quit();
}
```

Hello World!



Hello World - Input

```
/* INITIALIZATION */

/* EXECUTION */
SDL_Rect r = { 200,200, 50, 50 };
SDL_Event e;
while (1) {
    while (SDL_PollEvent(&e) == 0);
    if (e.type == SDL_QUIT) {
        break;
    } else if (e.type == SDL_KEYDOWN) {
        switch (e.key.keysym.sym) {
            case SDLK_UP:
                r.y -= 10;
            case SDLK_DOWN:
                r.y += 10;
            case SDLK_LEFT:
                r.x -= 10;
            case SDLK_RIGHT:
                r.x += 10;
        }
    }
    SDL_SetRenderDrawColor(renderer, 0xFF,0xFF,0xFF,0x00);
    SDL_RenderFillRect(renderer, NULL);
    SDL_SetRenderDrawColor(renderer, 0x00,0x00,0xFF,0x00);
    SDL_RenderFillRect(renderer, &r);
    SDL_RenderPresent(renderer);
}

/* FINALIZATION */
```


O “loop” de eventos

```
/* INITIALIZATION */  
  
/* EXECUTION */  
while (1) {  
    /* events */  
    /* logic */  
    /* redraw */  
}  
  
/* FINALIZATION */
```

- Modelo síncrono

O “loop” de eventos

```
/* INITIALIZATION */

/* EXECUTION */
while (1) {
    /* events */
    SDL_PollEvent(&e);
    SDL_WaitEvent(&e);
    SDL_WaitEventTimeout(&e, 25);

    /* logic */
    /* redraw */
}

/* FINALIZATION */
```

Exercício - “Animação”

- Mover dois retângulos “em quadrados” com velocidades diferentes
- Parar o retângulo quando clicado com o mouse

SDL_GetTicks

Use this function to get the number of milliseconds since the SDL library initialization.

Contents

- [SDL_GetTicks](#)
- [Syntax](#)
- [Return Value](#)
- [Code Examples](#)
- [Remarks](#)
- [Related Functions](#)

Syntax

[Toggle line numbers](#)

```
Uint32 SDL_GetTicks(void)
```

Return Value

Returns an unsigned 32-bit value representing the number of milliseconds since the SDL library initialized.

```
switch (e.type) {  
    <...>  
    SDL_MOUSEBUTTONDOWN:  
        SDL_MouseButtonEvent* me =  
            (SDL_MouseButtonEvent*) &e;  
        me->x;  
        me->y;  
        <...>  
        break;  
}
```

Animações

```
int speed = 10;

while (1) {
    SDL_PollEvent(&e);

    // eventos
    switch (e.type) {
        <...>
    }

    // animacoes
    x += speed;

    // redesenho
    <...>
}
```

```
int speed = 10;

while (1) {
    SDL_PollEvent(&e);

    // eventos
    switch (e.type) {
        <...>
    }

    // animacoes
    x += DT*speed;
        // DT: diferenca de tempo
        // entre dois frames (SDL_Ticks)

    // redesenho
    <...>
}
```


Compilando / Executando

```
# clone
> git clone https://github.com/fsantanna-uerj/reativos

# update
> cd reativos/
> git pull

# compile
> cd reativos/code/sdl
> gcc 00_hello.c -lSDL2 -o 00_hello # (01_input.c 01_input)

# run
> ./00_hello # (./01_input)
```

Documentação

- Lazy Foo Tutorial

- <http://lazyfoo.net/tutorials/SDL/index.php>

- Manual de referência

- <https://wiki.libsdl.org/APIByCategory>

- Fórum do SDL

- <http://forums.libsdl.org/>

SDL + Céu

- Hello World! (00_hello.ceu)
- Input (01_input.ceu)
- SDL (02_sdl.ceu)
- Animações (ex_01.ceu)
- Animações + code/await (ex_01_code.ceu)

game.ceu

- sprites
- salto (vy)
- aceleração (ax,ay)
- **objetos dinâmicos**
- **colisão**
- ???

Compilando / Executando

```
$ su - arduino    # (Password: ArdDev16)

$ git clone https://github.com/fsantanna/reativos
# OU
$ cd ~/reativos/
$ git pull

$ cd ~
$ tar xvf /home/extra/fsantanna/ceu.tgz
$ exec bash
$ ceu

# game.ceu
$ cd ~/ceu/ceu-sdl
$ make CEUFILE=<repo-reativos>/reativos/code/sdl/game.ceu
$ <repo-reativos>/code/sdl/game.exe

# birds.ceu
$ cd ~/ceu/ceu-sdl
$ make CEUFILE=<repo-reativos>/reativos/code/sdl/birds.ceu
$ cd <repo-reativos>/code/sdl/ && ./birds.exe
```

SDL + Céu

- Configurar o ambiente (toda vez que abrir um terminal):

```
$ . /opt/reativos/login.sh # note o espaço entre "." e "/"
```

- Compilar e gravar:

```
$ cd /opt/reativos/ceu-sdl/  
$ make CEU_SRC=<caminho-arquivo-ceu>
```

- http://github.com/fsantanna-uerj/reativos/code/*.ceu

Compilando / Executando

```
$ su - arduino    # (Password: ArdDev16)

$ git clone https://github.com/fsantanna/reativos
# OU
$ cd ~/reativos/
$ git pull

$ cd ~
$ tar xvf /home/extra/fsantanna/ceu.tgz
$ exec bash
$ ceu

$ cd ~/ceu/ceu-sdl
$ make CEUFILE=<repo-reativos>/reativos/code/sdl/00_hello.ceu
$ <repo-reativos>/code/sdl/00_hello.exe
```