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

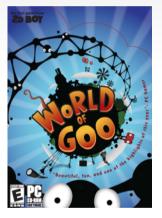
- Código aberto
 - Freeciv, SuperTux

- 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=r3wDnOAjrtk

- Indie Game: The Movie
 - http://vimeo.com/ondemand/indiegamethemovie/25268139

Hello World

```
#include <SDL2/SDL.h>
int main (int argc, char* args[])
    /* INITIALIZATION Hello World!
    SDL Init(SDL INIT
    SDL Window* windo
    SDL Renderer* ren
    /* EXECUTION */
    SDL SetRenderDraw
    SDL RenderFillRec
    SDL Rect r = \{ 20 \}
    SDL SetRenderDraw
    SDL RenderFillRec
    SDL Delay(5000);
    /* FINALIZATION *
    SDL DestroyRender
    SDL DestroyWindow
    SDL Quit();
}
```

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

```
# compile
> cd /opt/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/