

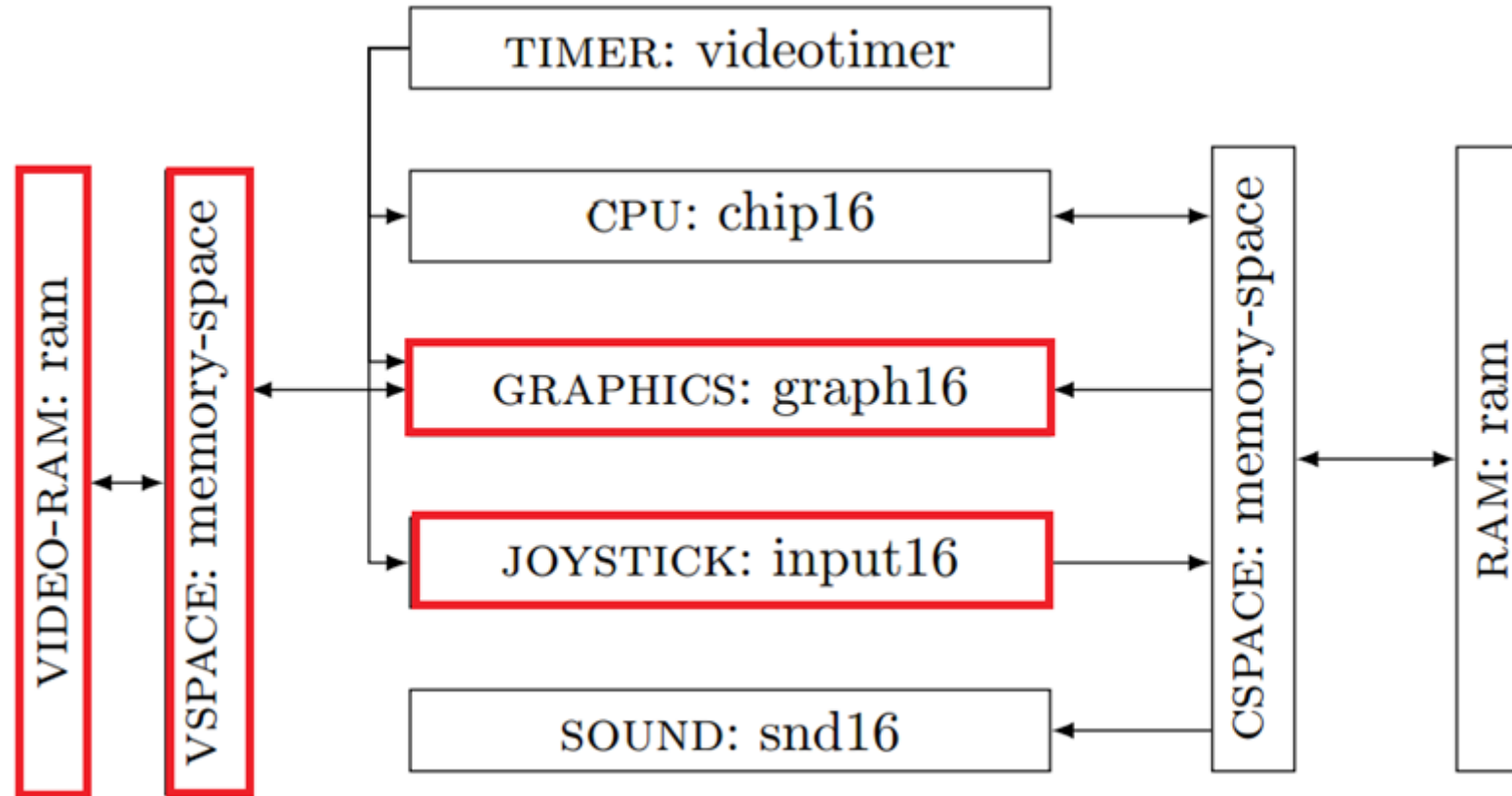
Chip16: Graphics and Input

Valery Konychev

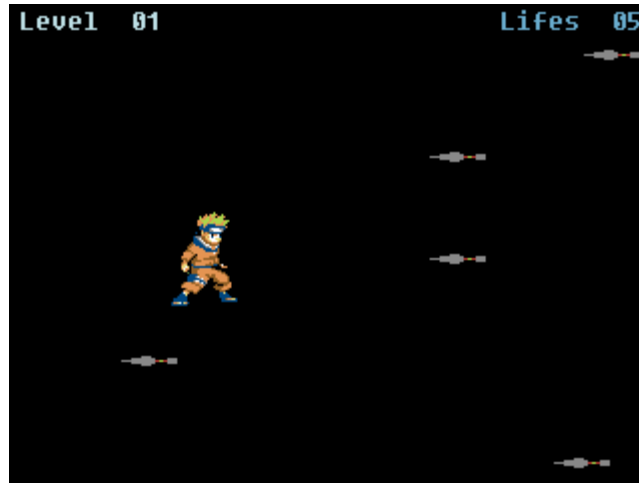


На лекции:

CHIP16



Graphics



Screen details:

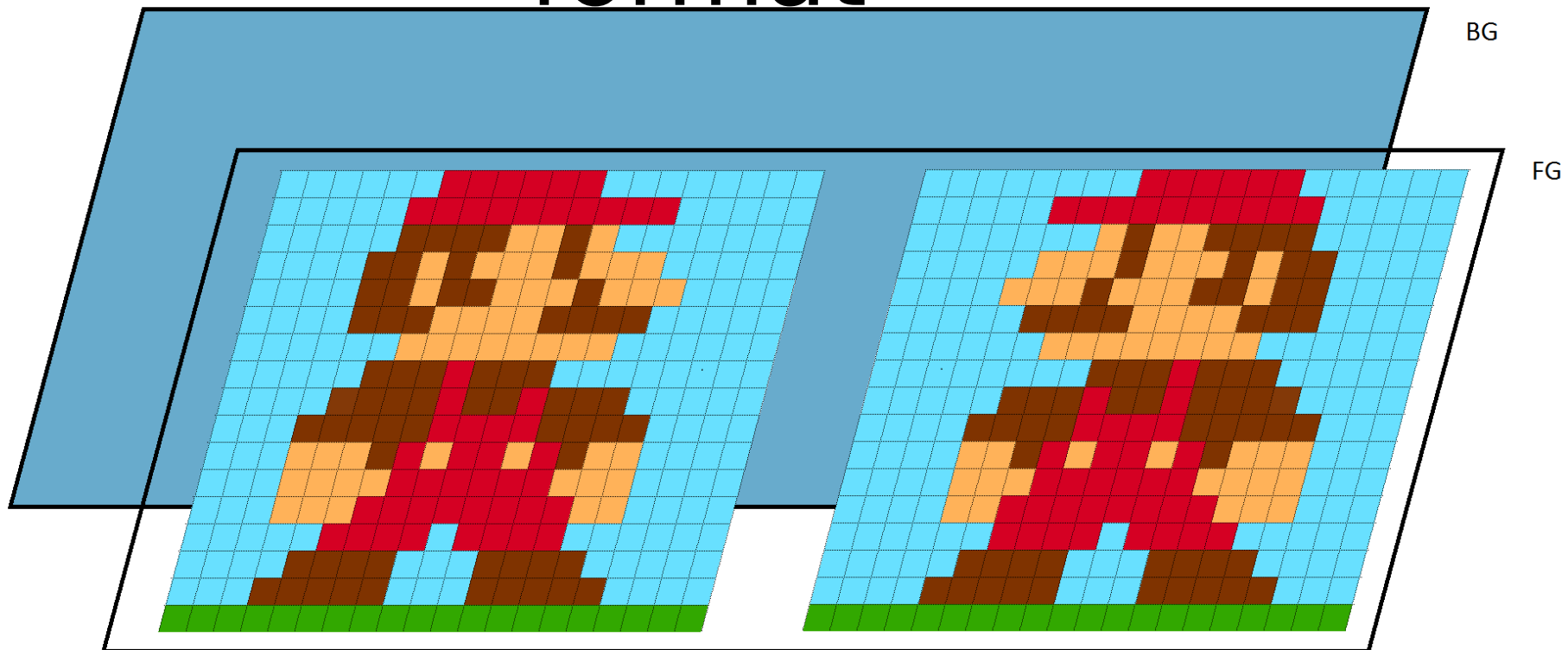
- 320x240 - screen resolution
- Refresh frequency of 60 Hz (~16.67ms)
- VBLNK (Vblank)

Palette

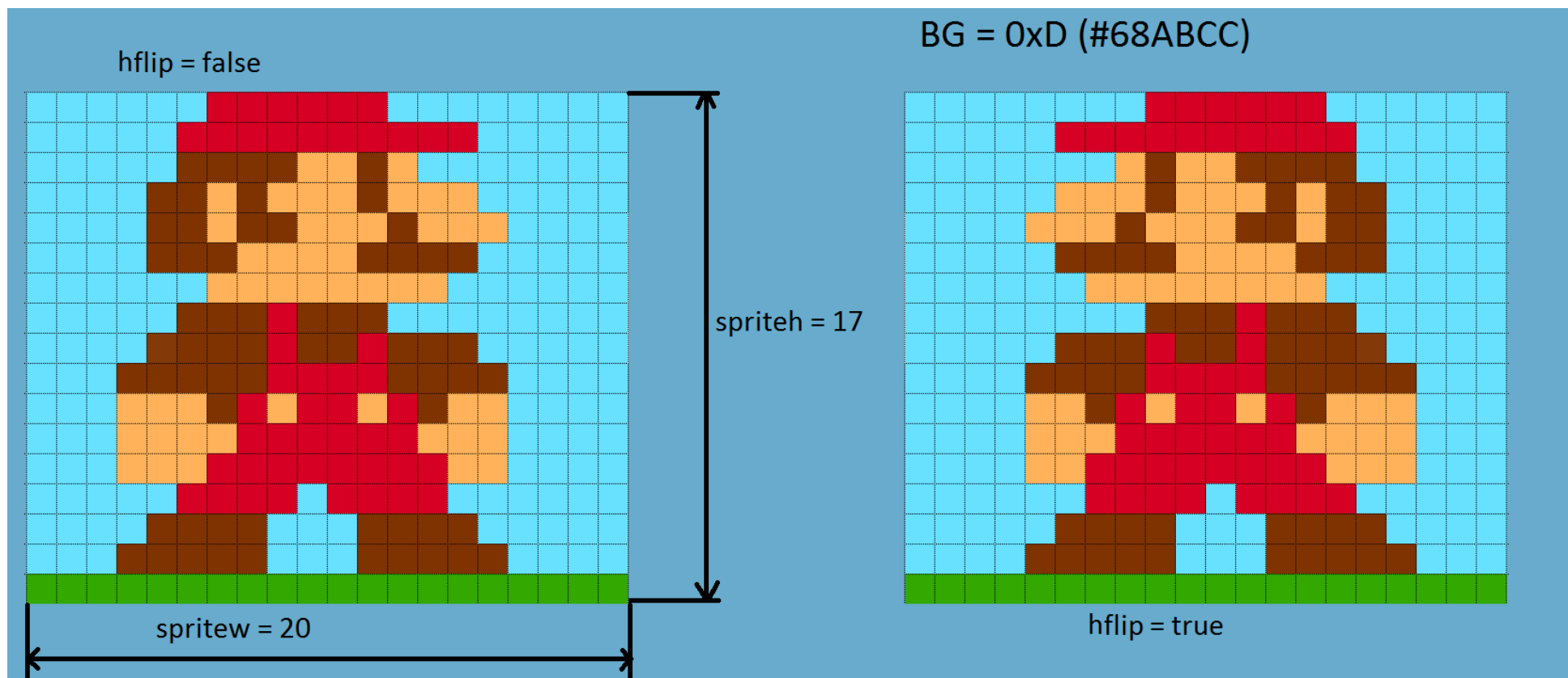
Индекс в палитре	Шестнадцат. значение	Цвет
0x0	0x000000	Черный, прозрачный на слое фона
0x1	0x000000	Черный
0x2	0x888888	Серый
0x3	0xBF3932	Красный
0x4	0xDE7AAE	Розовый
0x5	0x4C3D21	Темно-коричневый
0x6	0x905F25	Коричневый
0x7	0xE49452	Оранжевый
0x8	0xEAD979	Желтый
0x9	0x537A3B	Зеленый
0xA	0xABD54A	Светло-зеленый
0xB	0x252E38	Темно-синий
0xC	0x00467F	Синий
0xD	0x68ABCC	Светло-синий
0xE	0xBCDEE4	Sky blue
0xF	0xFFFFFFFF	Белый

Opcode (Hex)	Mnemonic	Usage	Flags affected	Introduced
D0 00 LL HH	PAL HHLL	Load palette from[HHLL]		1.1
D1 0X 00 00	PAL RX	Load palette from[RX]		1.1

Display format



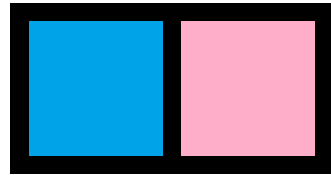
State registers



Register	Type
bg	Nibble
spritew	Unsigned byte

spriteh	Unsigned byte
hflip	Boolean
vflip	Boolean

Min Sprite:



0xD4

Example:

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

0xF – White
0x1 - Black

Example

0xF – White

0x1 - Black

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

F1 FF 1F

F1 FF 1F

F1 FF 1F

FF FF FF

1F FF F1

F1 11 1F

□ $36 / 2 = 18$ bytes

Example

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

01 00 00 00 (CLS) – clear display

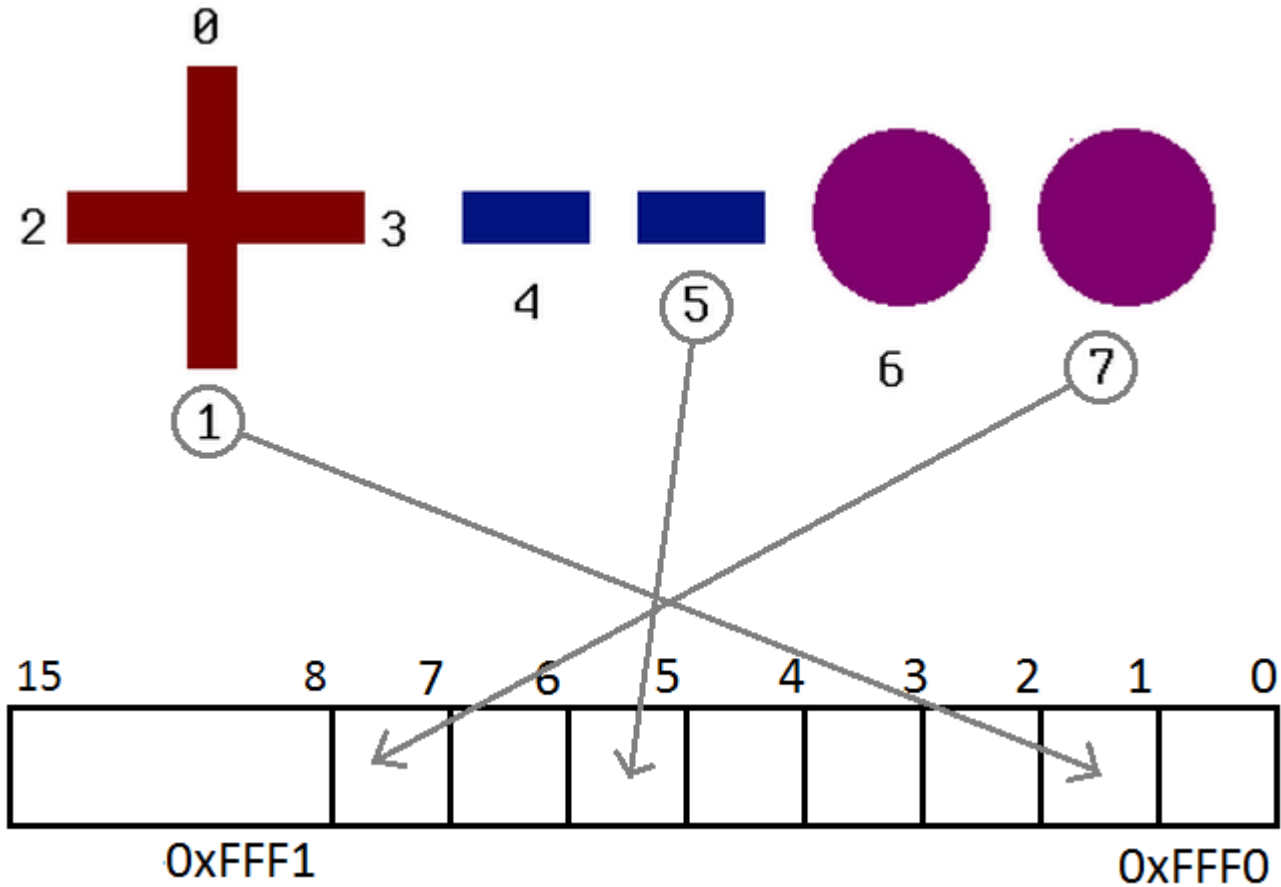
04 00 06 06 (SPR 06 06) – set H and L of sprite

08 00 00 00 (FLIP 0 0) – no flip

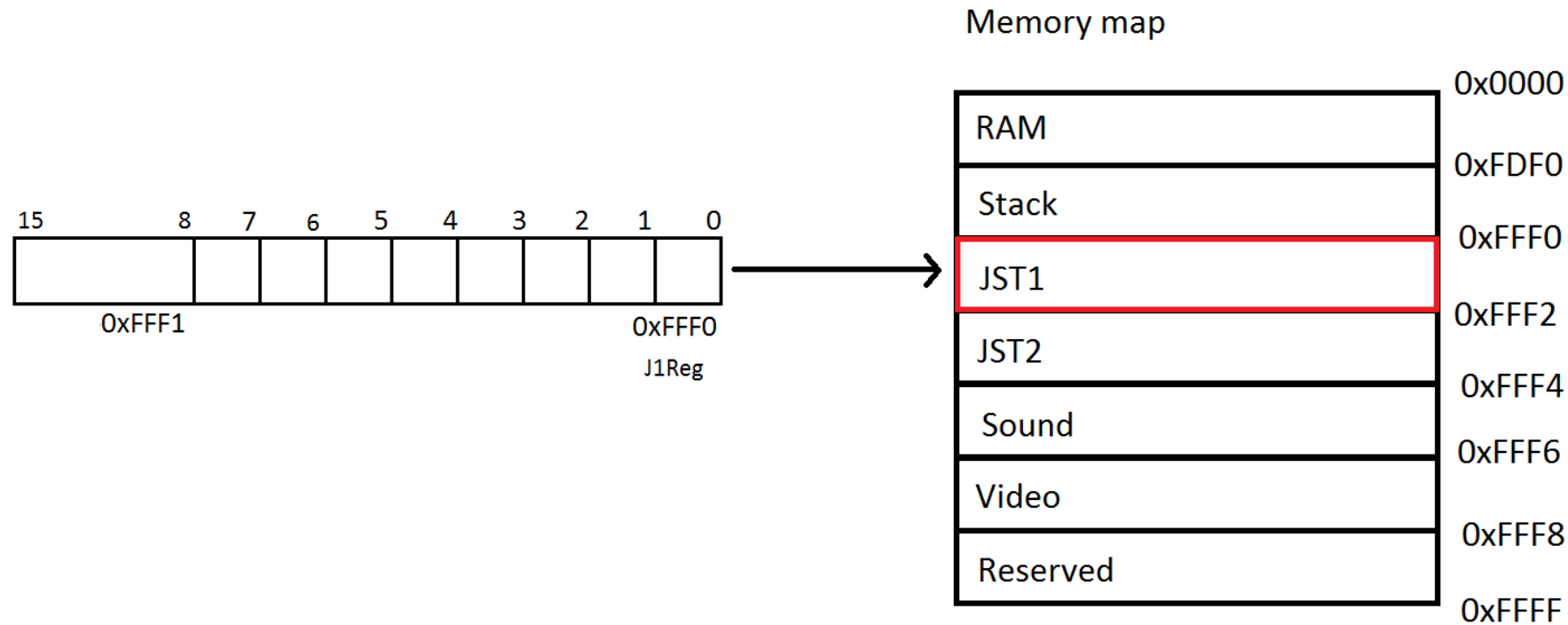
05 YX LL HH (DRW RX RY HHLL) – draw sprite
from HHLL with coordination RX RY

Input/Joystick

Controller layout



Memory mapping



SDL

Init

```
#include  
"path_to_sdl2/SDL.h"  
SDL_Init(SDL_INIT_EVERYTHI  
NG);  
/*code*/  
atexit(SDL_Quit);
```

```
if ( SDL_Init( SDL_INIT_VIDEO ) < 0 )  
{  
printf( "Unable to init SDL: %s",  
SDL_GetError ( ) );  
return 1; }
```

SDL_INIT_AUDIO

SDL_INIT_VIDEO

SDL_INIT_TIMER

SDL_INIT_CDROM

SDL_INIT_JOYSTICK

SDL_INIT_EVERYTHING — инициализация всех
подсистем

SDL_INIT_NOPARACHUTE — не ловить фатальные
сигналы

SDL_INIT_EVENTS

SDL_INIT_EVENTTHREAD — запускать менеджер
событий в отдельном потоке



Create window

```
SDL_Window* SDL_CreateWindow(  const char* title,  
                                int x,  
                                int y,  
                                int w,  
                                int h,  
                                Uint32 flags );
```

Events

SDL_Event event;

int **SDL_PollEvent**(SDL_Event* event)

Click handling

```
SDL_Event event;  
Const Uint8 * kState =  
SDL_GetKeyboardState(NULL);  
  
If (SDL_PollEvent(&event))  
    switch (event.type){  
        case event.key.keysym.sym:  
            if (SDL_SCANCODE_UP)  
                /*do smth*/
```

Scancodes

- http://wiki.libsdl.org/SDL_Scancode

Полезные ссылки по SDL/SDL2:

- <http://habrahabr.ru/post/134936/>
- <http://www.libsdl.org/>

