

# **Arquitecturas y Entornos de desarrollo para Videoconsolas**

## **Seminario: entorno de desarrollo y ejemplos**

Grado de Ingeniero en Informàtica  
Escola Tècnica Superior d'Enginyeria Informàtica  
Curso 2020/2021

# Objetivos

- Contexto: Seminario + introducción de conceptos para abordar la PL01
- Seminario
  - Introducir el desarrollo de una aplicación bajo el SDK Homebrew para NDS y 3DS
    - Mostrar desarrollos *NDS/Tutorial Day*
- PL01
  - Compilación y ejecución ejemplos del SDK sobre NDS y 3DS
  - Reconstruir el ejemplo del tutorial “*The Bouncing Ball*” en NDS

# Índice

## 1. *Seminario: Entorno de desarrollo*

### 1. *Introducir el desarrollo para NDS bajo el SDK Homebrew para NDS*

1. *Estructura de directorios y Makefile*

2. *Introducir ejemplos sobre NDS: NDS/Tutorial Day*

### 2. *Introducir el desarrollo para 3DS bajo el SDK Homebrew*

## 2. *Introducción a la PL01*

1. *Compilación y ejecución ejemplos del SDK sobre NDS y 3DS*

2. *Reconstruir el ejemplo del tutorial “The Bouncing Ball” en NDS*

# Entorno de desarrollo

- Instalación

- Herramientas de desarrollo + SDK

- *ToolChain vs ToolChain + MSYS*

- SDK

- *Librería base para la plataforma Hw. (+ ejemplos)*
    - *Otras librerías portadas*



- Ejecución

- Emulador

- Máquina real → Cartuchos vs *The Homebrew launcher*

- <http://smealum.github.io/3ds/>
    - *Does this let me run roms ?*

\* No. These exploits run strictly in userland, meaning that we do not have full control over the console, and therefore cannot run pirated software. That being said, we have the same amount of access as regular apps and games to the console's hardware, and so we can make some ... homebrew!



# Entorno de desarrollo NDS

- Instalación: Herramientas de desarrollo
  - ToolChain vs *ToolChain* + *MSYS*
    - “On Unix-like platforms such as Linux/macOS/WSL, there's devkitPro pacman”  
[[https://devkitpro.org/wiki/Getting\\_Started](https://devkitpro.org/wiki/Getting_Started)]

En versiones de Linux derivadas de Debian: (1) instalar el paquete .deb

```
$ wget https://github.com/devkitPro/pacman/releases/download/devkitpro-pacman-1.0.1/devkitpro-pacman.deb
```

```
$ sudo dpkg -i devkitpro-pacman.deb
```

```
$ dkp-pacman -U https://downloads.devkitpro.org/devkitpro-keyring-r1.787e015-2-any.pkg.tar.xz
```

```
** Listar paquetes      `dkp-pacman -Sl`
```

```
** Instalar un paquete `sudo dkp-pacman -S <name of package>`
```

```
** Borrar un paquete  `sudo dkp-pacman -R <name of package>`
```

```
** Instalar todo      `sudo dkp-pacman -Sy`
```

```
$ sudo dkp-pacman -S nds-dev 3ds-dev wii-dev switch-dev
```



# Entorno de desarrollo NDS

## ● Instalación: Herramientas de desarrollo (Unix)

En versiones de Linux derivadas de Debian: (2) instalar los paquetes de desarrollo

```
$ sudo dkp-pacman -S nds-dev 3ds-dev wii-dev switch-dev
```

```
:: There are 14 members in group nds-dev:
```

```
:: Repository dkp-lib
```

```
1) default-arm7 2) dswifi 3) libfat-nds 4) libfilesystem 5) libnds 6) maxmod-nds  
7) nds-examples
```

```
:: Repository dkp-linux
```

```
8) devkit-env 9) devkitARM 10) dstools 11) general-tools 12) grit 13) mmutil 14) ndstool
```

```
...
```

```
:: There are 11 members in group 3ds-dev:
```

```
:: Repository dkp-lib
```

```
1) 3ds-examples 2) citro2d 3) citro3d 4) libctru
```

```
:: Repository dkp-linux
```

```
5) 3dslink 6) 3dstools 7) devkit-env 8) devkitARM 9) general-tools 10) picasso 11) tex3ds
```

```
...
```

```
:: There are 6 members in group switch-dev:
```

```
:: Repository dkp-lib
```

```
1) libnx 2) switch-examples
```

```
:: Repository dkp-linux
```

```
3) devkit-env 4) devkitA64 5) general-tools 6) switch-tools
```



# Entorno de desarrollo NDS

- **Instalación: Herrramientas de desarrollo (Unix)**

En versiones de Linux derivadas de Debian: (2b) comprobar la instalación

```
$ sudo dkp-pacman -S nds-dev 3ds-dev wii-dev switch-dev
$ ll /opt/devkitpro/
total 44K
drwxr-xr-x 7 root  root  4,0K jul 21 02:44 ..
drwxr-xr-x 8 magusti magusti 4,0K jul 21 02:44 pacman
drwxr-xr-x 4 root  root  4,0K jul 21 03:08 libnds
drwxr-xr-x 8 root  root  4,0K jul 21 03:08 devkitARM
drwxr-xr-x 3 root  root  4,0K jul 21 03:08 tools
drwxr-xr-x 4 root  root  4,0K jul 21 03:08 libctru
drwxr-xr-x 4 root  root  4,0K jul 21 03:08 libnx
drwxr-xr-x 5 root  root  4,0K jul 21 03:08 examples
drwxr-xr-x 11 magusti magusti 4,0K jul 21 03:08 .
drwxr-xr-x 8 root  root  4,0K jul 21 03:08 devkitA64
drwxr-xr-x 12 root  root  4,0K jul 21 03:08 licenses
$
```

# Entorno de desarrollo NDS

## ● Instalación: Herramientas de desarrollo (Unix)

En versiones de Linux derivadas de Debian: (3) variables y emuladores

```
$ sudo dkp-pacman -S nds-dev 3ds-dev wii-dev switch-dev  
$ export DEVKITPRO=/opt/devkitpro  
$ export DEVKITARM=/opt/devkitpro/devkitARM  
$ export DEVKITA64=/opt/devkitpro/devkitA64
```

```
$ sudo apt-get install -y desmume
```

[[ Citra / Yuzu / Ryujinx no tienen instalador ]]

# Una copia local para experimentar

```
$ cp -rp /opt/devkitpro/examples/ .  
$ cd examples/  
$ cd nds/  
$ make  
$ desmume bin/sprite_rotate.nds  
$ cd ../3ds  
$ make  
$ citra-qt.sh bin/hello-world.3dsx  
$ cd ../switch  
$ make  
$ yuzu.sh graphics/printing/hello-world/hello-world.nro
```



# Entorno de desarrollo NDS

## ● Instalación: Emuladores (Unix)

#DeSmuME ← NDS

\$ sudo apt-get install -y desmume

[[ Citra / Yuzu / Ryujinx no tienen instalador ]]

#Citra ← 3DS

\$ citra-qt.sh bin/hello-world.3dsx

# Yuzu ← Switch

De <https://yuzu-emu.org/downloads/> --> "Nightly Build"

\$ wget <https://github.com/yuzu-emu/yuzu-nightly/releases/download/nightly-550/yuzu-linux-20180720-d4104c7.tar.xz>

\$ tar xvf yuzu-linux-20180720-d4104c7.tar.xz

\$ cd yuzu-linux-20180720-d4104c7/; ./yuzu

# Ryujinx ← Switch

# <https://ryujinx.org/#/> --> Download the latest build ==> <https://ryujinx.org/#/Build>

\$ tar xvf ryujinx-1.0.1514-linux\_x64.tar.gz

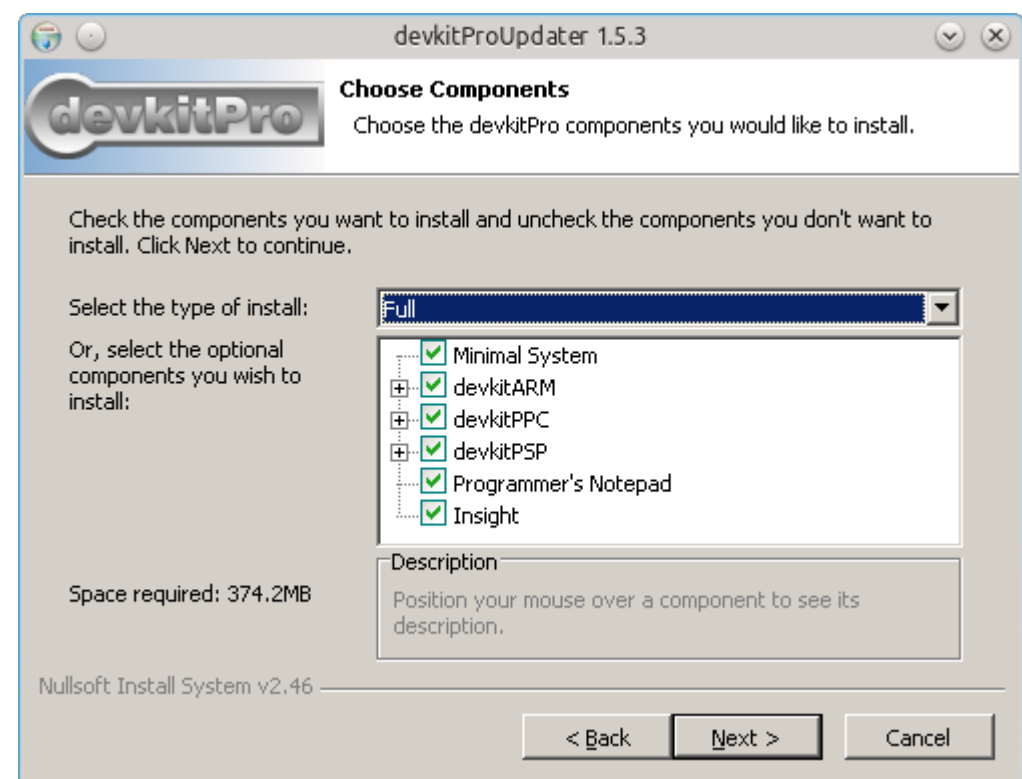
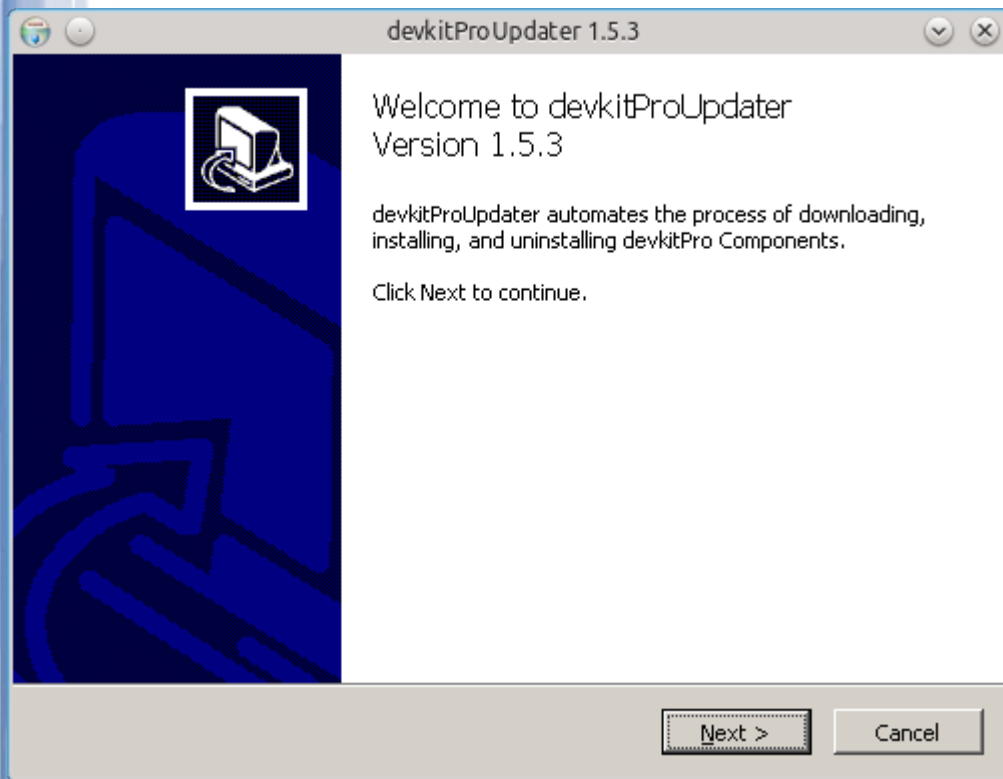
\$ publish/Ryujinx

# Entorno de desarrollo NDS

- Instalación

- Herramientas de desarrollo

- *ToolChain* vs *ToolChain + MSYS*

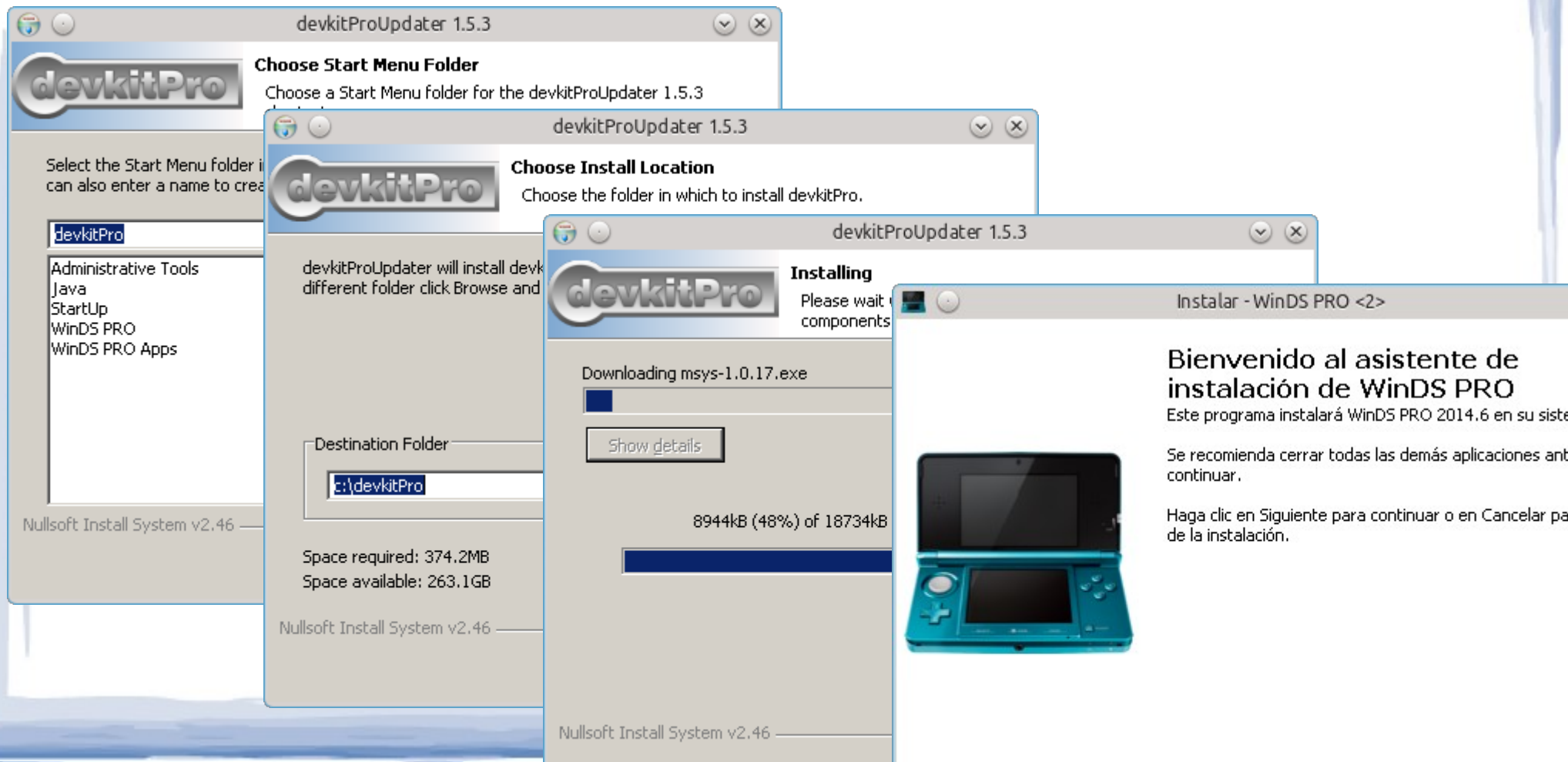


# Entorno de desarrollo NDS

- Instalación

- Herramientas de desarrollo

- *ToolChain* vs *ToolChain* + *MSYS*



# Entorno de desarrollo NDS

- Instalación

- Herramientas de desarrollo

- *ToolChain vs ToolChain + MSYS*

- SDK NDS

- Secciones → p. ej. bajo nivel (ndstypes.h)

- \* *2D engine API*

- *double, float → float64, float32, int16, ...*

- \* *3D engine API*

- \* *Audio API*

- *byte, u8, int8, s8, vs8, ...*

- \* *Memory*

- Declaraciones y definiciones

- \* *System*

- \* *User Input/output*

- *define*

- \* *Utility*

- *aligned vs packed*

- \* *Custom Peripherals*

- *const vs static*

- \* *Debugging*

- *volatile vs register*

# Entorno de desarrollo NDS

- Instalación

- Herramientas de desarrollo

- *ToolChain vs ToolChain + MSYS*

- SDK NDS

- *libnds (+ ejemplos)*

- *Otras librerías incluidas: libfilesystem, libfat, Imm, libdswifi*

- *¿Otras librerías? ¡Portarlas!*

- \* *sqlite, libpng, freetype, jpeg, zip, xml, helix y tremor.*

# Entorno de desarrollo 3DS y Switch

- Instalación

- Herramientas de desarrollo
  - *ToolChain vs ToolChain + MSYS*
- SDK NDS
- SDK 3DS
  - *libctru (+ ejemplos)*
  - *Otras librerías incluidas: libcitro3d*
  - *¿Otras librerías? ¡Portarlas!*
- *SDK Switch*
  - *Libndx*
  - *¿Otras librerías? ¡Portarlas!*



# Seminario: Entorno de desarrollo

- Ejecución

- Emulador

- GNU/Linux o OS X

- \* (NDS) DesMuME, no\$gba + Wine, Ideas, Dualis,  
...

- \* (3DS) Citra

- Sobre Android ...

- Sobre Windows

- Máquina real

- Cartuchos

- *HomeBrew Launcher*



# Seminario: Estructura del proyecto

## ● Estructura de directorios y *Makefile*

Makefile de \${DEVKITPRO}/examples/nds/templates/arm9/Makefile

<https://github.com/devkitPro/nds-examples/blob/master/templates/arm9/Makefile>

```
#-----  
# TARGET is the name of the output  
# BUILD is the directory where object files & intermediate files will be placed  
# SOURCES is a list of directories containing source code  
# INCLUDES is a list of directories containing extra header files  
# DATA is a list of directories containing binary files embedded using bin2o  
# GRAPHICS is a list of directories containing image files to be converted with grit  
# AUDIO is a list of directories containing audio to be converted by maxmod  
# ICON is the image used to create the game icon, leave blank to use default rule  
# NITRO is a directory that will be accessible via NitroFS  
#-----
```

## ● Personalizar el título en la NDS

Personalizar cómo se muestra en la NDS

```
# These set the information text in the nds file  
GAME_TITLE := Advanced ARM9 template  
GAME_SUBTITLE1 := built with devkitARM  
GAME_SUBTITLE2 := http://devkitpro.org
```

# Seminario: Estructura del proyecto (II)

## ● Incluir otras librerías <> SDK

```
#-----  
# any extra libraries we wish to link with the project (order is important)  
#-----  
LIBS := -lnds9  
  
# automatically add libraries for NitroFS  
ifneq ($(strip $(NITRO)),)  
LIBS := -lfilesystem -lfat $(LIBS)  
endif  
# automatically add maxmod library  
ifneq ($(strip $(AUDIO)),)  
LIBS := -lmm9 $(LIBS)  
endif  
  
#-----  
# list of directories containing libraries, this must be the top level containing  
# include and lib  
#-----  
LIBDIRS := $(LIBNDS) $(PORTLIBS)
```

# Seminario: Estructura del proyecto (y III)

- Un Makefile de un proyecto ya incluye otras definiciones

— *`$(DEVKITARM)/base_rules`*

```
iwram, itcm, ...
```

```
...
```

```
#-----
```

```
# add portlibs path
```

```
#-----
```

```
export PORTLIBS_PATH := $(DEVKITPRO)/portlibs
```

— *`$(DEVKITARM)/base_tools`*

```
export PATH := $(DEVKITARM)/bin:$(PATH)
```

```
#-----
```

```
# the prefix on the compiler executables
```

```
#-----
```

```
PREFIX := arm-none-eabi-
```

```
export CC := $(PREFIX)gcc
```

```
export CXX := $(PREFIX)g++
```

```
...
```

# Seminario: Hello, World!

- *Plantilla:*

... *\$DEVKITPRO/examples/nds/templates/arm9/source/main.c*

```
/*-----  
    Basic template code for starting a DS app  
-----*/  
#include <nds.h>  
#include <stdio.h>  
//-----  
int main(void) {  
    //-----  
    consoleDemolnit();  
    iprintf("Hello World!");  
    while(1) {  
        swiWaitForVBlank();  
        scanKeys();  
        int pressed = keysDown();  
        if(pressed & KEY_START) break;  
    }  
}
```

# Seminario: Ejemplos

- *PoliformaT / Aev: Recursos / Prácticas*  
...  
*practica1\_instalacion*  
*recursosParaAplicacionesSobreNDS*
  - J. D. Jaén. (2015). *Tutorial práctico para desarrollo de videojuegos sobre plataforma Nintendo NDS*
  - Reconstruir ejemplos de código de “NDS/Tutorials – Dev-Scene”
- *Obtener una copia del libro de*
  - F. Moya y M. J. Santofimia. 2007. *Laboratorio de Estructura de Computadores empleando videoconsolas Nintendo DS.*

## NDS/Tutorials

< [NDS](#)

### Contents

- [1 Day 1: Setting up](#)
- [2 Day 2: NDS Introduction](#)
- [3 Day 3: 2D Raster Graphics and Input](#)
- [4 Day 4: 2D Tile Graphics](#)
- [5 Day 5: 2D Sprites](#)
- [6 Day 6: 2D Hardware Effects](#)
- [7 Day 7: Sound and Music](#)
- [8 Day 8: 3D Introduction](#)
- [9 Day 9: Wifi and Networking](#)
- [10 Day 10: A bit more advanced 3D Graphics](#)
- [11 Game Programming 101: Collision Detection](#)
- [12 Game Programming 101: Animation](#)
- [13 Tiling, Sprites and Animation](#)



# Seminario: Ejemplos

- Day 1 "First demo".

...

- We will initialize our DS and paint one of the screens a wonderful red color.

```
#include <nds.h>
#include <stdio.h>

int main(void)
{
    int i;

    consoleDemoInit();

    videoSetMode(MODE_FB0);

    vramSetBankA(VRAM_A_LCD);

    printf("Hello World!\n");
    printf("www.Drunkencoders.com");

    for(i = 0; i < 256 * 192; i++)
        VRAM_A[i] = RGB15(31, 0, 0);

    while(1){
        swiWaitForVBlank();
    }

    return 0;
}
```



# Seminario: Ejemplos

- Day 3 “Talking the keypad”
  - Acceso directo a registro vs libnds

```
#include <nds.h>
#include <stdio.h>

int main(void)
{
    consoleDemoInit();

    while(1)
    {
        if(REG_KEYINPUT & KEY_A)
            printf("Key A is released");
        else
            printf("Key A is pressed");

        swiWaitForVBlank();

        consoleClear();
    }

    return 0;
}
```

```
#include <nds.h>
#include <stdio.h>

int main(void)
{
    consoleDemoInit();

    while(1)
    {
        scanKeys();
        int held = keysHeld();

        if( held & KEY_A)
            printf("Key A is pressed\n");
        else
            printf("Key A is released\n");

        if( held & KEY_X)
            printf("Key X is pressed\n");
        else
            printf("Key X is released\n");

        if( held & KEY_TOUCH)
            printf("Touch pad is touched\n");
        else
            printf("Touch pad is not touched\n");

        swiWaitForVBlank();

        consoleClear();
    }

    return 0;
}
```

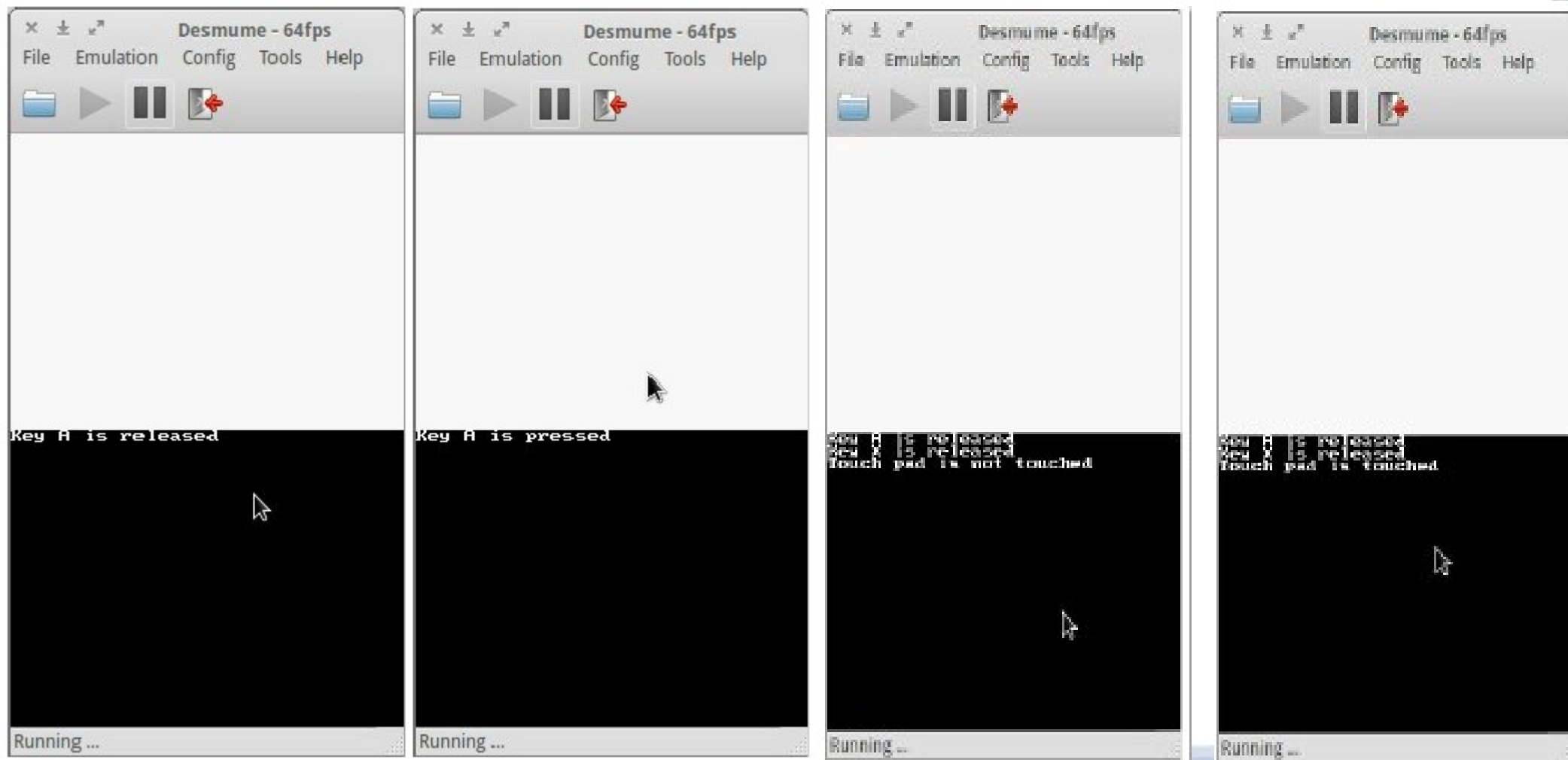
# Seminario: Ejemplos

- Day 3 “Talking the keypad”

– Acceso directo a registro

vs

libnds



# Seminario: Ejemplos

- Day 3 “Frame Buffer”

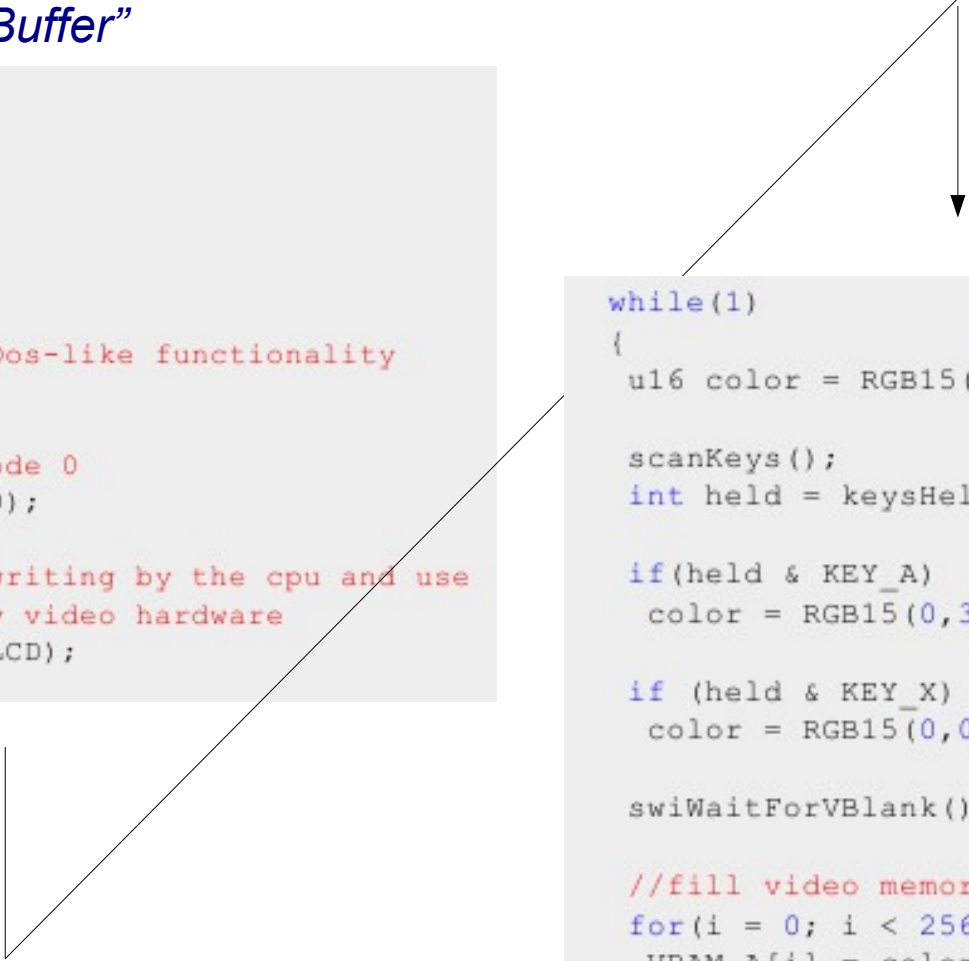
```
#include <nds.h>
#include <stdio.h>

int main(void)
{
    int i;

    //initialize the DS Dos-like functionality
    consoleDemoInit();

    //set frame buffer mode 0
    videoSetMode(MODE_FB0);

    //enable VRAM A for writing by the cpu and use
    //as a framebuffer by video hardware
    vramSetBankA(VRAM_A_LCD);
```



```
while(1)
{
    u16 color = RGB15(31,0,0); //red

    scanKeys();
    int held = keysHeld();

    if(held & KEY_A)
        color = RGB15(0,31,0); //green

    if (held & KEY_X)
        color = RGB15(0,0,31); //blue

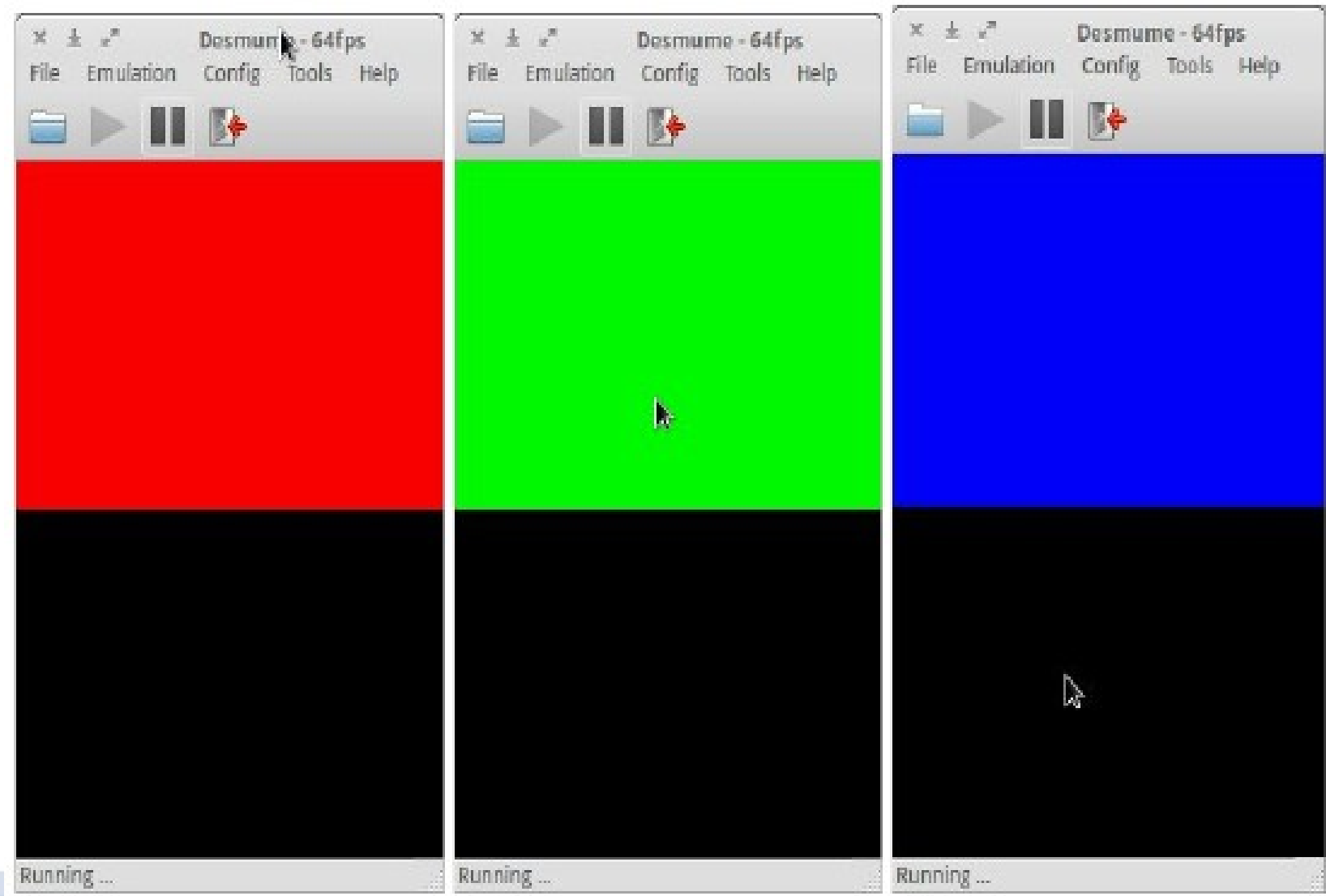
    swiWaitForVBlank();

    //fill video memory with the chosen color
    for(i = 0; i < 256*192; i++)
        VRAM_A[i] = color;
}

return 0;
}
```

# Seminario: Ejemplos

- Day 3 “Frame Buffer”



# Seminario: Ejemplos

- Day 3 “Pixels and Things (stars)”

```
#include <nds.h>
#include <stdlib.h>

#define NUM_STARS 40

typedef struct
{
    int x;
    int y;
    int speed;
    unsigned short color;
}Star;

Star stars[NUM_STARS];

void MoveStar(Star* star)
{
    star->x += star->speed;

    if(star->x >= SCREEN_WIDTH)
    {
        star->color = RGB15(31,31,31);
        star->x = 0;
        star->y = rand() % 192;
        star->speed = rand() % 4 + 1;
    }
}
```

```
void ClearScreen(void)
{
    int i;

    for(i = 0; i < 256 * 192; i++)
        VRAM_A[i] = RGB15(0,0,0);
}

void InitStars(void)
{
    int i;

    for(i = 0; i < NUM_STARS; i++)
    {
        stars[i].color = RGB15(31,31,31);
        stars[i].x = rand() % 256;
        stars[i].y = rand() % 192;
        stars[i].speed = rand() % 4 + 1;
    }
}

void DrawStar(Star* star)
{
    VRAM_A[star->x + star->y * SCREEN_WIDTH] = star->color;
}

void EraseStar(Star* star)
{
    VRAM_A[star->x + star->y * SCREEN_WIDTH] = RGB15(0,0,0);
}
```

```
int main(void)
{
    int i;

    irqInit();
    irqEnable(IRQ_VBLANK);

    videoSetMode(MODE_FB0);
    vramSetBankA(VRAM_A_LCD);

    ClearScreen();
    InitStars();

    //we like infinite loops in console dev!
    while(1)
    {
        swiWaitForVBlank();

        for(i = 0; i < NUM_STARS; i++)
        {
            EraseStar(&stars[i]);

            MoveStar(&stars[i]);

            DrawStar(&stars[i]);
        }

        return 0;
    }
}
```



# Seminario: Ejemplos

- Day 3 *"Pixels and Things (stars)"*



# Seminario: Ejemplos

- Day 3 *“Touching things”*

```
#include<nds.h>
#include<stdlib.h>

int main(void)
{
    touchPosition touch;

    videoSetMode(MODE_FB0);
    vramSetBankA(VRAM_A_LCD);

    //notice we make sure the main graphics engine renders
    //to the lower lcd screen as it would be hard to draw if the
    //pixels did not show up directly beneath the pen

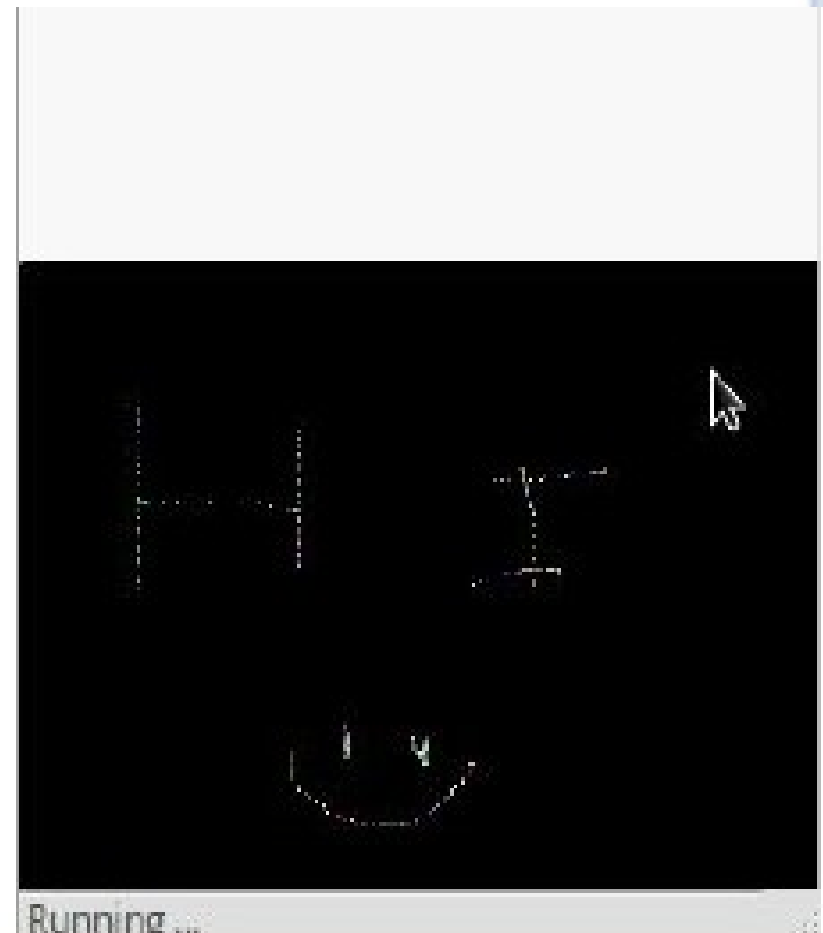
    lcdMainOnBottom();

    while(1)
    {
        scanKeys();

        if(keysHeld() & KEY_TOUCH)
        {
            // write the touchscreen coordinates in the touch variable
            touchRead(&touch);

            VRAM_A[touch.px + touch.py * 256] = rand();
        }
    }

    return 0;
}
```



# Seminario: Ejemplos

- Day 4 “First Map Demo”
  - Teselaciones (tiles) → mapas

```
#include <nds.h>

//create a tile ca
u8 redTile[64] =
{
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
};

//create a tile ca
u8 greenTile[64] =
{
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
    2,2,2,2,2,2,2,2,
};

//-----
int main(void) {
//-----
    int i;

    //set video mode and map vram to the background
    videoSetMode(MODE_0_2D | DISPLAY_BG0_ACTIVE);
    vramSetBankA(VRAM_A_MAIN_BG_0x06000000);

    //get the address of the tile and map blocks
    u8* tileMemory = (u8*)BG_TILE_RAM(1);
    u16* mapMemory = (u16*)BG_MAP_RAM(0);

    //tell the DS where we are putting everything and set 256 color mode and
    REG_BG0CNT = BG_32x32 | BG_COLOR_256 | BG_MAP_BASE(0) | BG_TILE_BASE(1);

    //load our palette
    BG_PALETTE[1] = RGB15(31,0,0);
    BG_PALETTE[2] = RGB15(0,31,0);

    //copy the tiles into tile memory one after the other
    swiCopy(redTile, tileMemory, 32);
    swiCopy(greenTile, tileMemory + 64, 32);

    //create a map in map memory
    for(i = 0; i < 32 * 32; i++)
        mapMemory[i] = i & 1;
    while(1)
        swiWaitForVBlank();

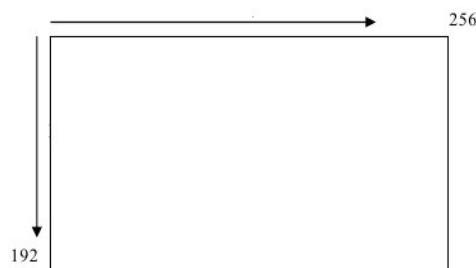
    return 0;
}
```



# Esquema de bloques (y V)

- Modos gráficos → “aceleradores”
  - 2 motores gráficos 2D + 1 motor 3D
  - 2 pantallas: LCD + táctil

## ○ Resolución



## ● Otros:

- 16 canales de sonido, micrófono y altavoces estéreo,
- 2x4 temporizadores, aceleradores para las operaciones de división y raíz cuadrada, etc.

Graphics Modes

Main 2D Engine				
Mode	BG0	BG1	BG2	BG3
Mode 0	Text/3D	Text	Text	Text
Mode 1	Text/3D	Text	Text	Rotation
Mode 2	Text/3D	Text	Rotation	Rotation
Mode 3	Text/3D	Text	Text	Extended
Mode 4	Text/3D	Text	Rotation	Extended
Mode 5	Text/3D	Text	Extended	Extended
Mode 6	3D	-	Large Bitmap	-
Frame Buffer	Direct VRAM display as a bitmap			
Sub 2D Engine				
Mode	BG0	BG1	BG2	BG3
Mode 0	Text	Text	Text	Text
Mode 1	Text	Text	Text	Rotation
Mode 2	Text	Text	Rotation	Rotation
Mode 3	Text	Text	Text	Extended
Mode 4	Text	Text	Rotation	Extended
Mode 5	Text	Text	Extended	Extended

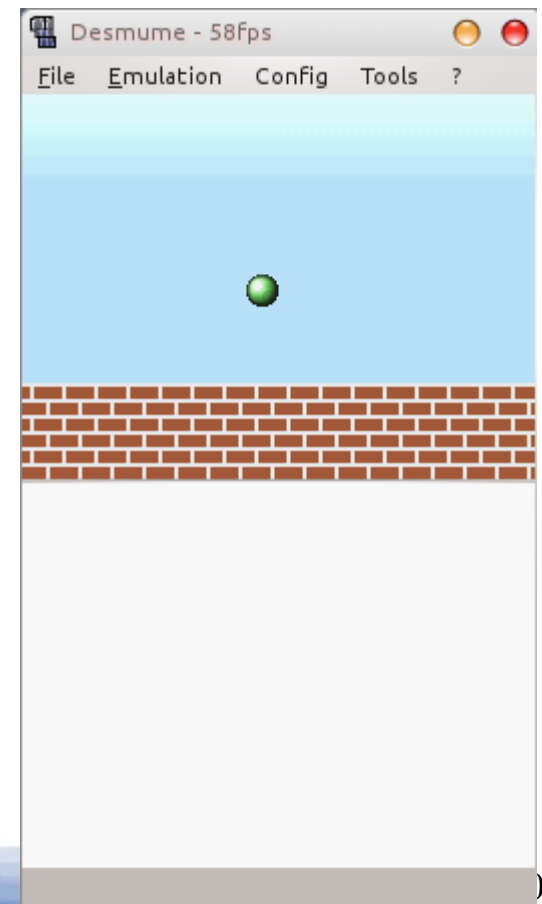
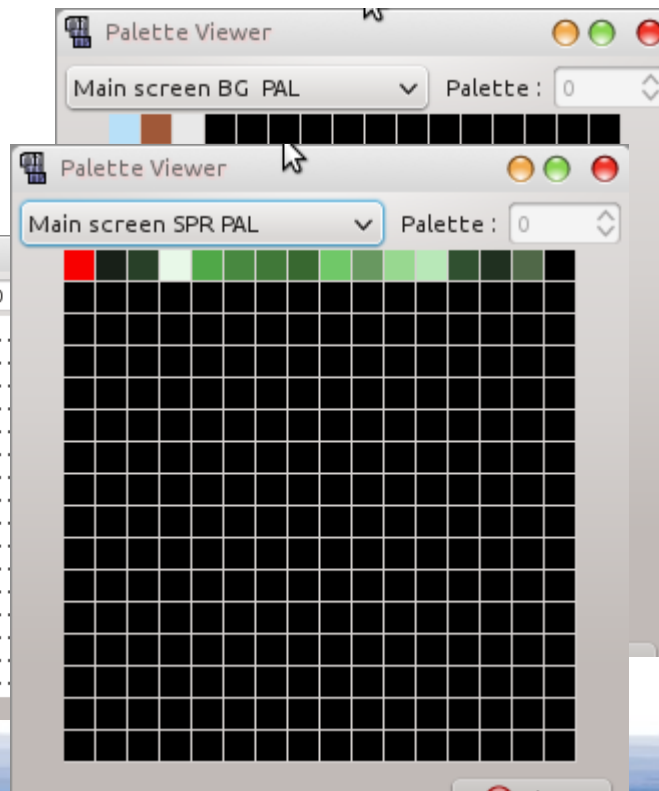
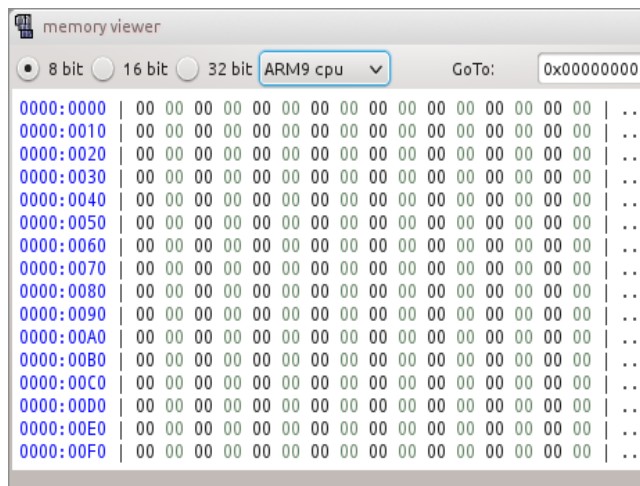
# Práctica 1: Un ejemplo sobre NDS

- *Instalación SDK*
- Lanzar la PL01: Instalación, ejemplos y un ejemplo *"The Bouncing Ball"*

— PoliformaT | Prácticas | *practica1\_instalacion*

`$ desmume`

`$ desmume-glade`



# Seminario: Bibliografía

- *F. Moya y M. J. Santofimia. 2007. Laboratorio de Estructura de Computadores empleando videoconsolas Nintendo DS.*
- *J.J. Jaen. (2015). Tutorial practico para desarrollo de videojuegos sobre plataforma Nintendo NDS. [En linea]. Disponible en <<https://riunet.upv.es/handle/10251/56433>>.*
- *Francisco Sivianes, J. Barros y A. Martín. 2009. Entorno Desarrollo para NDS. Periféricos e Interfaces. 3º Ingeniería Técnica en Informática, especialidad en Sistemas Físicos*  
*<[http://www.dte.us.es/tec\\_inf/itis/peri\\_int/EvolucionInicio/Trabajo\\_NDS.pdf](http://www.dte.us.es/tec_inf/itis/peri_int/EvolucionInicio/Trabajo_NDS.pdf)>*
- 3DBrew <<http://3dbrew.org/wiki>>
- Neimod y Martin Korth. 2013. DSTek: Nintendo DS Technical Information
- GbaTek. Technical information from no\$gba
- Märten Tonisoo. 2010. Nintendo DS game console.
  - <[http://www.martentonisoo.com/documents/Nintendo\\_DS\\_game\\_console.pdf](http://www.martentonisoo.com/documents/Nintendo_DS_game_console.pdf)>
- ThomasWorld. Baby Steps In Nintendo DS Homebrew Hacking.