

# Nintendo DS Homebrew Development

An introduction

Hugh Cole-Baker

[sigmaris@gmail.com](mailto:sigmaris@gmail.com)

# What's a Nintendo DS?

- Handheld console from Nintendo
- Released in 2004, DS Lite in 2006, DSi (hardware upgrade) in 2008
- Dual screen, with one touch screen



Thursday, 18 February 2010

Talk generally deals with DS and DS Lite, DSi not so interesting to homebrew developers as its protection has not been fully circumvented

# Tech Specs

- Dual processors - one 66MHz ARM9 and one 33MHz ARM7
- No floating point unit - use fixed-point math instead
- Dual 256x192 screens
- 4MB Ram
- Hardware 2D
  - Scrolling, scaling and rotating backgrounds (x4) and sprites (x128)
- Hardware 3D
  - Texture mapped, 2048 triangles per frame
- Speakers, Microphone, Touchscreen, WiFi - No WPA :(
- Graphics capabilities are similar to the N64

# DS Homebrew

- ✦ Unofficial, hobbyist development
- ✦ Not approved by Nintendo
- ✦ Closed platform, so some 'hacking' was required

Thursday, 18 February 2010

Protection against running homebrew on DS is now fully understood and can be circumvented  
The same cannot be said for DSi.

# Why DS Homebrew?

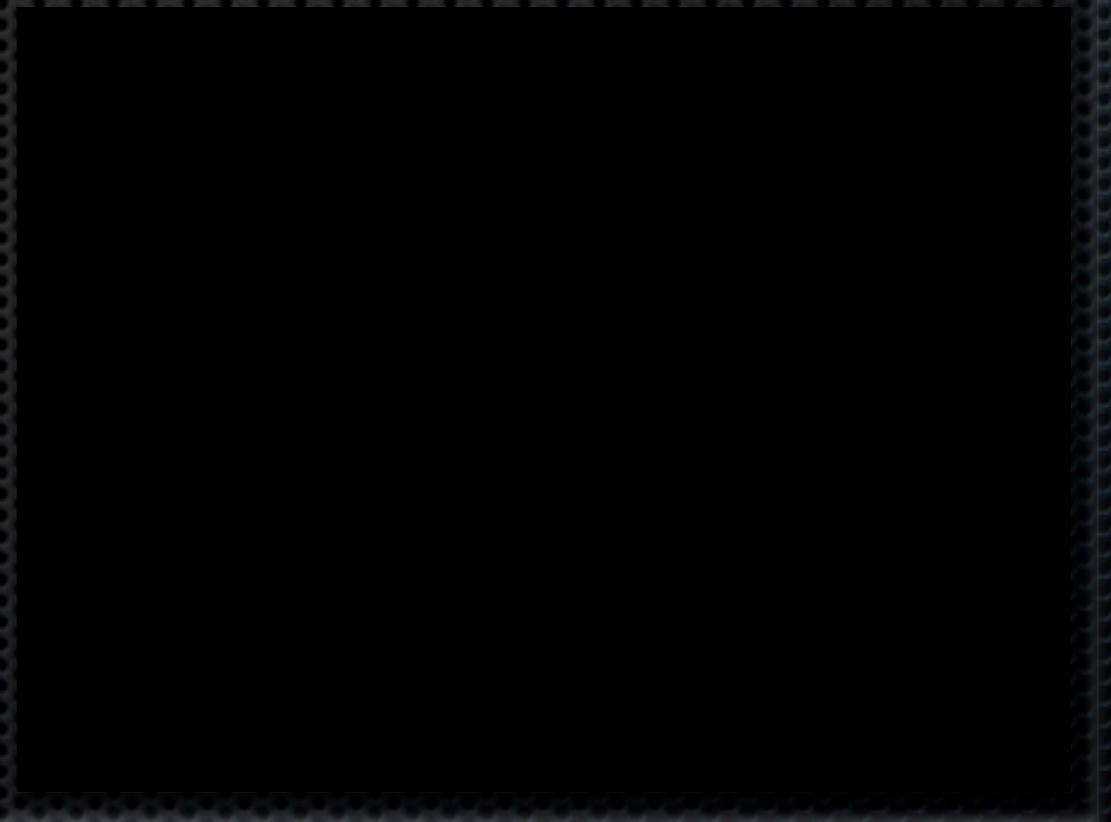
- ✦ System is accessible to homebrew coders
- ✦ Wide audience - Many people own a DS, a sizeable proportion of them have a flash card
- ✦ Designed for games (but much more is possible!)
- ✦ Touchscreen allows for easy and fun interaction

Thursday, 18 February 2010

Point 1: Flash cards used to store your programs (on SD card, etc), Hardware is 90% documented for homebrewers, Development kit for Windows, Mac and Linux

# Why contd.

- ✦ The challenge!
  - ✦ Programming within the limits of a small device
  - ✦ Squeezing the last % of performance from the hardware
  - ✦ Coding for concurrent dual CPUs
- ✦ Nostalgia?



Thursday, 18 February 2010

Limited memory to work with, limited CPU. The challenge is to do the most with what you have.

# What You Need

- ✦ A DS, DS Lite or DSi
- ✦ A flash card
- ✦ A development kit - devkitARM and libnds from [www.devkitpro.org](http://www.devkitpro.org)
- ✦ The first two items could be replaced by an emulator.

Thursday, 18 February 2010

Flash card simulates a DS game card with programs loaded off SD card. Recommended models at the time of writing are CycloDS or DSTopToy for the original, Acekard 2i for the DSi.

# Coding for the DS - languages

- ✦ devkitARM includes C and C++ cross-compilers for Win, Mac and Linux
- ✦ J2ME port - <http://bit.ly/ds-java> - run mobile Java apps
- ✦ Lua - <http://microlua.risike.com>
- ✦ Python - <https://www.develer.com/trac/dspython/>
- ✦ Generally, to fully utilise the hardware you need C/C++

Thursday, 18 February 2010

C/C++: Compile with example Makefile from devkitARM, or MS Visual Studio templates



# Coding for the DS - libraries

- ✦ libnds - controls most of the hardware
- ✦ dswifi - can use the WiFi capability
- ✦ maxmod - for playing music, samples
- ✦ libfat - provides access to storage devices

Thursday, 18 February 2010

DSWifi contains a full TCP stack.

PALib simplifies things a little, but is no longer maintained (and missing some support e.g. 3D)

If you concentrate on PALib you may have trouble when looking to use more advanced features.

Woopsi is a C++ library for UI elements.

# Coding for the DS - technique

- ✦ ARM9 handles graphics and heavy-duty computation
- ✦ ARM7 handles input, wifi, and sound
- ✦ Don't have to code for the ARM7, can just use default ARM7 code from libnds
- ✦ Make use of hardware support (scrolling, sprites, 3D, math coprocessor) whenever you can
- ✦ Conserve RAM

Thursday, 18 February 2010

Only 4MB RAM, and 16K DTCM for stack.

# Coding for the DS - deploying

- ✦ Compile your program to produce a .nds file
- ✦ Copy the file to an SD card
- ✦ Insert SD card in flash card, insert flash card in DS
- ✦ Power on, select your program, play!

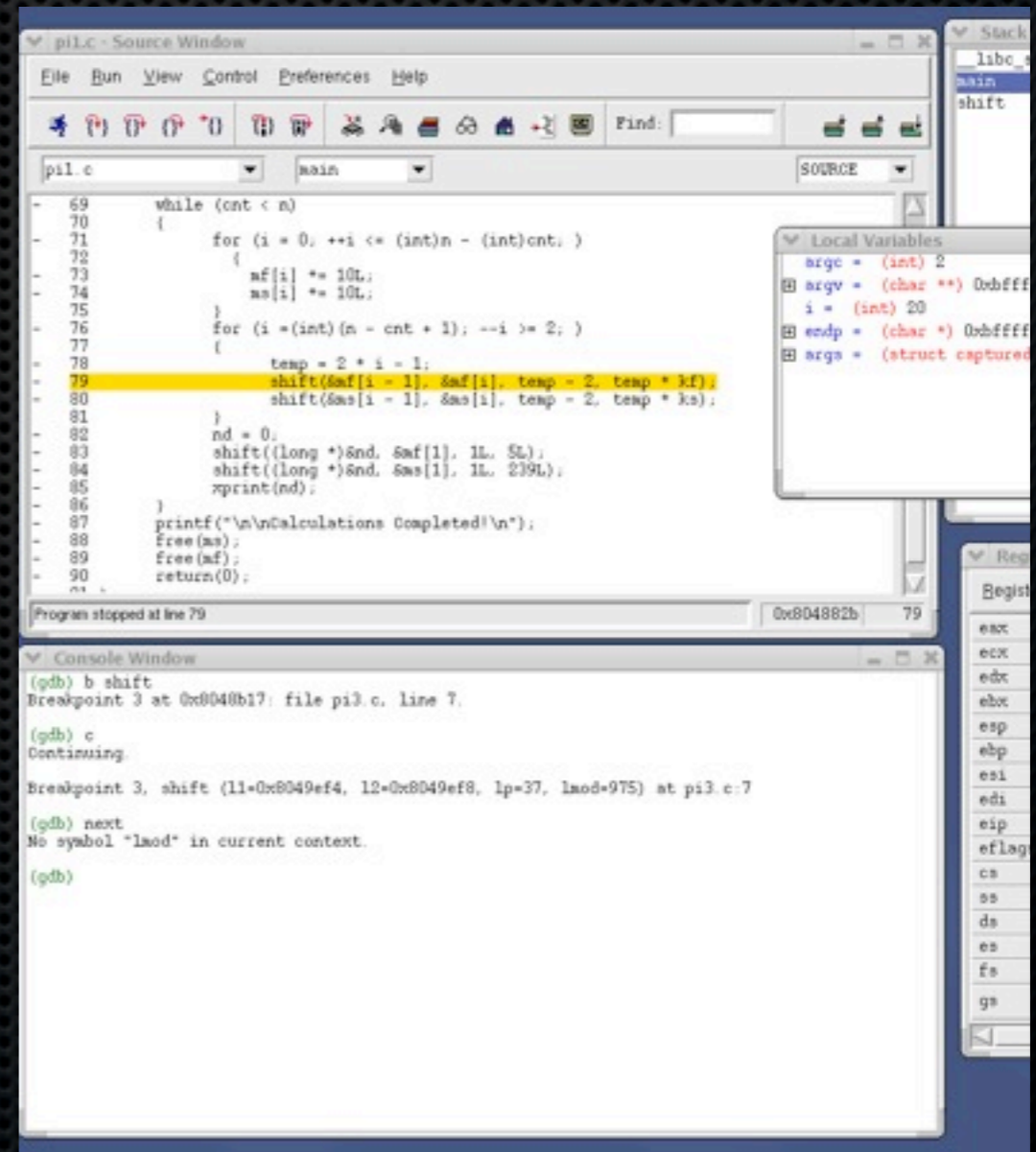


Thursday, 18 February 2010

Compiling – use makefile from nds examples or visual studio, etc

# Coding for the DS - debugging

- ✦ Use GDB over WiFi - devkit for Windows includes graphical frontend called Insight.
- ✦ Use an emulator with debugger
- ✦ Use printf statements!



The screenshot shows a debugger window with the following components:

- Source Window:** Displays C code for a function named 'shift'. Line 79 is highlighted: `shift(&af[i - 1], &af[i], temp - 2, temp * kf);`
- Local Variables:** A panel on the right showing variables like `argc = (int) 2`, `argv = (char **) 0xbffff`, `i = (int) 20`, `endp = (char *) 0xbffff`, and `args = (struct captured`.
- Console Window:** Shows GDB commands and output:

```
(gdb) b shift
Breakpoint 3 at 0x8040b17: file pi3.c, line 7.
(gdb) c
Continuing.
Breakpoint 3, shift (11*0x8049ef4, 12*0x8049ef8, 1p=37, laod=975) at pi3.c:7
(gdb) next
No symbol "laod" in current context.
(gdb)
```

Thursday, 18 February 2010

devkitARM includes GDB for ARM.

Set up connection with your home computer running GDB host.

masscat's gdb stub: [http://masscat.afraid.org/ninds/debug\\_stub.php](http://masscat.afraid.org/ninds/debug_stub.php)

DesMuMe also has GDB stub

No\$GBA and iDeas have debugger versions

# Where to start / find out more

- ✦ [www.devkitpro.org/devkitarm](http://www.devkitpro.org/devkitarm) has the devkit, forums, and NDS example programs
- ✦ [drunkencoders.com](http://drunkencoders.com) has good tutorials
- ✦ GBATEK (google it) - pretty complete hardware docs
- ✦ [forum.gbadev.org](http://forum.gbadev.org) has a DS section
- ✦ [www.liranuna.com](http://www.liranuna.com) and [www.patater.com/manual](http://www.patater.com/manual) are good resources too
- ✦ IRC channel #dsdev on [irc.blitzed.org](http://irc.blitzed.org)

Thursday, 18 February 2010

Drunkencoders now have screencasts

# Questions?

- ✦ If you want to get in touch: [sigmaris@gmail.com](mailto:sigmaris@gmail.com)
- ✦ There are a couple posts about DS development on my blog: <http://sigmaris.info>