Computer Organization and Assembly Language Assignment #3

Box Filter

Virtual Boy Advance

- An emulator for Nintendo Gameboy Advance
- GBA - ARM7 architecture

Develop Environment

- HAM GBA Devkit
  - http://www.ngine.de/index.jsp?pageid=3504
  - recommended

- DevKit Advance
  - http://devkitadv.sourceforge.net/
  - Lightweight and simple
  - Base knowledge in writing Makefile required

Preparing

- Download ham-280-full-win32.exe
- Follow the instructions to install it

- Download template files
  - http://w.csie.org/~r96051/hw3_template.rar
  - Extract to wherever you want

Replacing Makefile

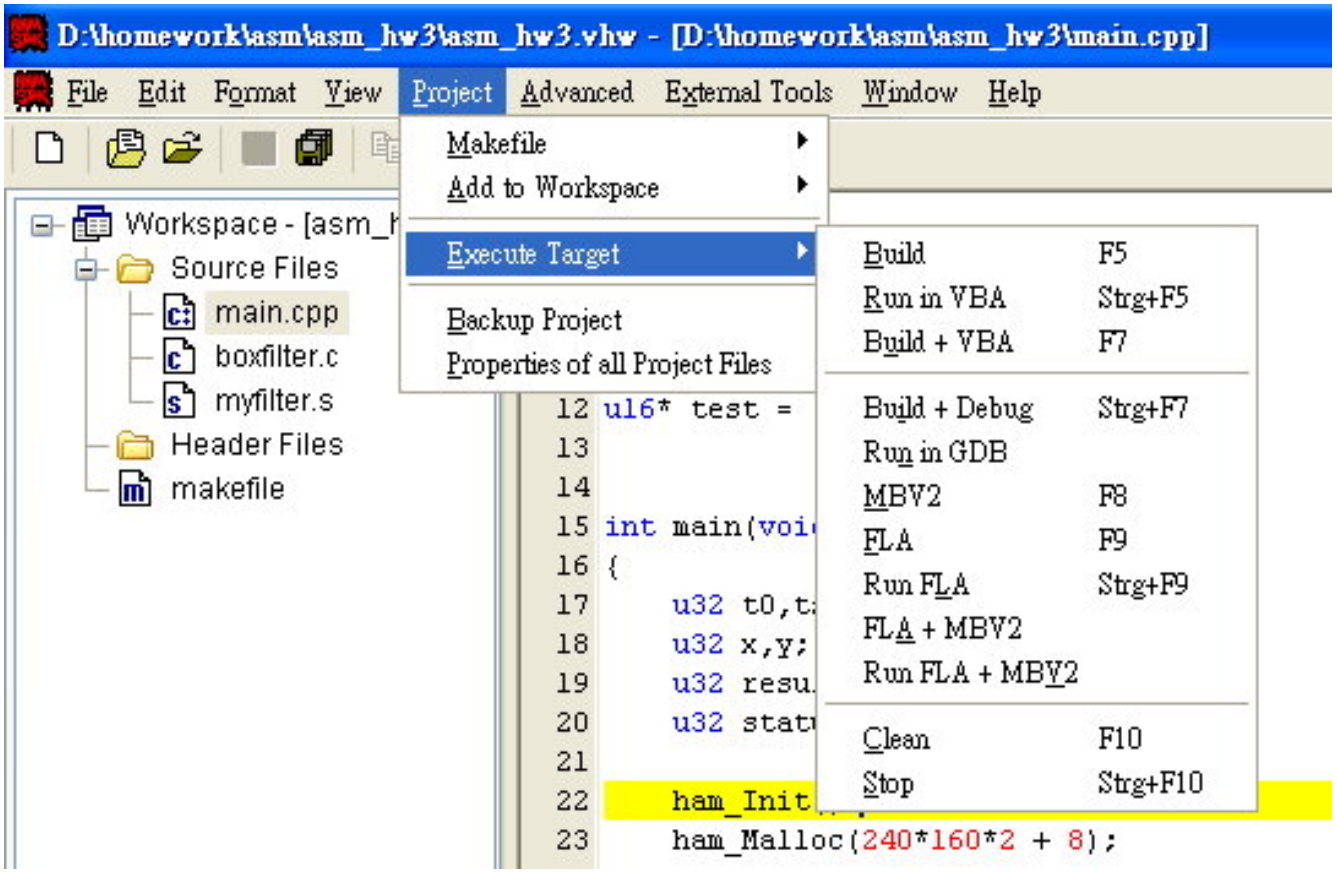
- In ($HAM_dir)/system/
- Replacing
  - master.mak & standard-targets.mak

- With these two files:
  - http://w.csie.org/~r96051/master.mak
  - http://w.csie.org/~r96051/standard-targets.mak

Getting start

- Run ($HAM_dir)/vham/VHAM.exe
- File -> Open workspace
  - open asm_hw3.vhw
- Press ‘F7’ to Build and Run

Getting start

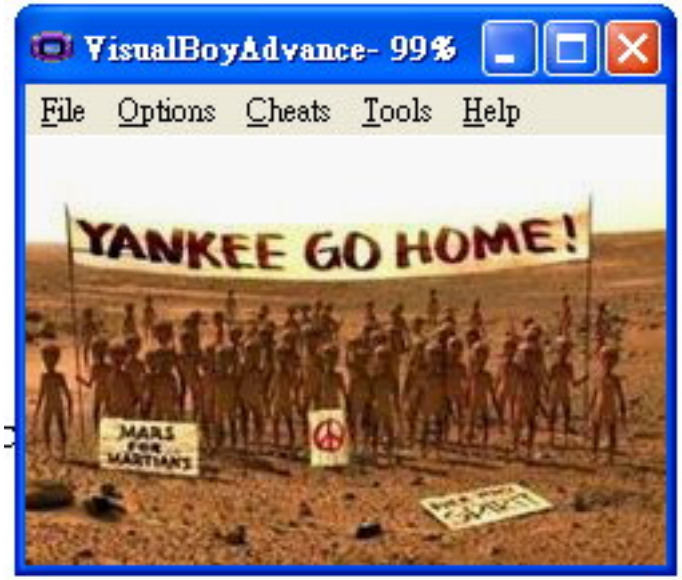


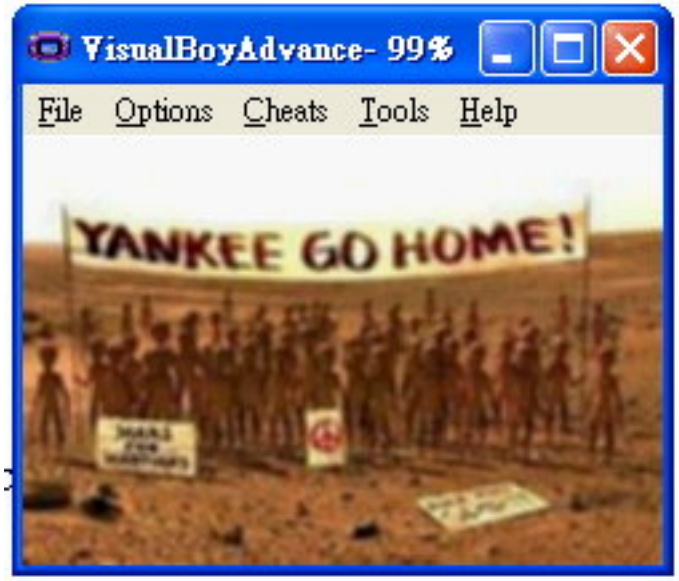
Box Filter

- A spatial domain filter that simply averaging pixel values in kernel.
- Fixed 3x3 kernel for this assignment.
- Notice: There are 2x2 kernel on corners, and 2x3 kernel on edges

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Box Filter


VisualBoyAdvance- 99%
File Options Cheats Tools Help
YANKEE GO HOME!


VisualBoyAdvance- 99%
File Options Cheats Tools Help
YANKEE GO HOME!

Assignment

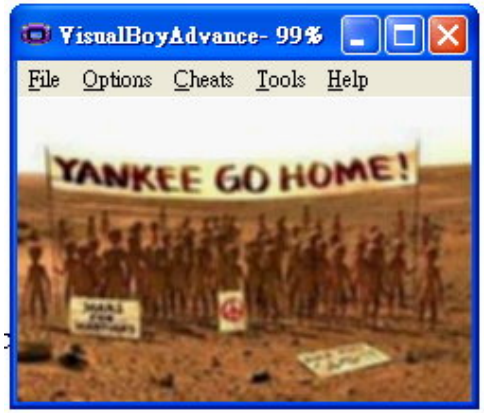
- Modify myfilter.s as a box filter
- void myfilter(u16* ret,const u16* ori);
  - ret and ori are two pointer to original and return images
  - ret in r0, ori in r1

myfilter.s

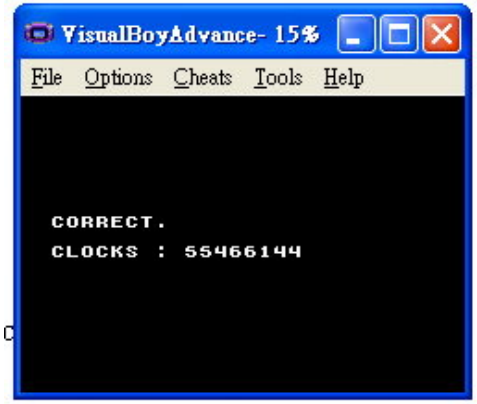
```
1      .file   "myfilter.c"
2      .text
3      .align  2
4      .global myfilter
5      .type   myfilter, %function
6  myfilter:
7
8      ldrh    r2, [r1, #0]     @ movhi @ * ori
9      strh    r2, [r0, #0]     @ movhi     @ * ret
10
11     bx  lr
12
13     .size   myfilter, .-myfilter
14     .ident  "GCC: (GNU) 3.3.2"
```

Auto Judge

- Press Left, Up and Down to switch between original image, filtered image and result.



- Both accuracy and efficiency will be encountered when scoring.



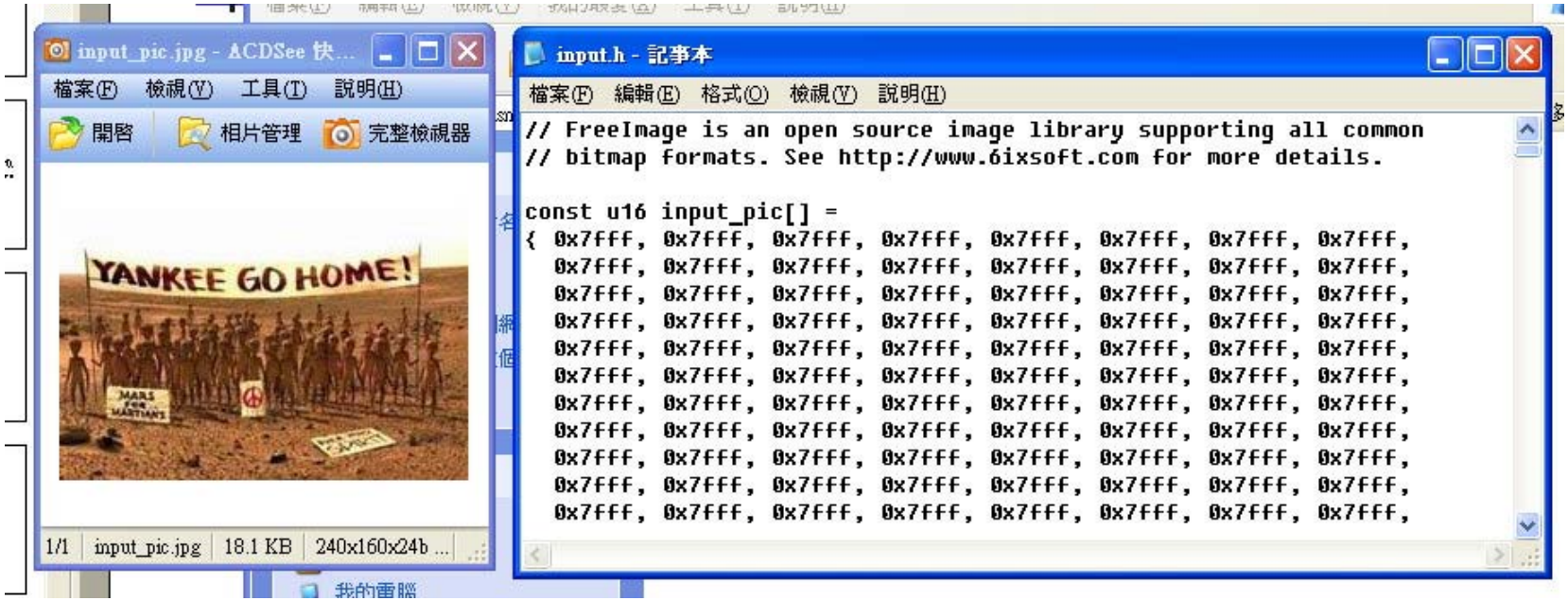
Changing input

- Replacing input.h for different input image.

- You can use Gfx2gba making your own input image.
  - Gfx2gba 1.03
  - http://www.gbadev.org/tools.php?showinfo=143

Gfx2gba

- Converting image file to C header file
  - See ReadMe.txt for usage information



```
// FreeImage is an open source image library supporting all common
// bitmap formats. See http://www.6ixsoft.com for more details.

const u16 input_pic[] =
{ 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
  0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff, 0x7fff,
```

BGR5 Format

- 15 bits for a pixel
- 5 bits for each BGR channel
  - Red channel at the lowest 5 bits
  - Green channel at bit6 to bit10
  - Blue channel at bit11 to bit15
- Separating channels when blending

BGR5 Format

- A straightforward C code for blending

```
            ncolor = ori[ny*240+nx];
            cc++;
            r+= (ncolor&0x001f);
            g+= ((ncolor&0x03e0)>>5);
            b+= ((ncolor&0x7c00)>>10);
        }
    }
    r = r/cc;
    g = g/cc;
    b = b/cc;

    ret[y*240+x] = (b<<10)+ (g<<5) + r;
```

- Is there a faster way?

Efficiency issue

- Loop unrolling?
- GBA memory :
  - 96KB VRAM
  - 32KB in-chip RAM
  - 256KB on-board RAM
  - Can we tiling?
- Space and time : trade off
  - Are there duplicate adding? Can we reuse it?

Reference

- More about GBA development:
- GBADEV.org
  - http://www.gbadev.org/
- Jonathan S. Harbour's blog
  - http://theharbourfamily.com/jonathan/?page_id=89