Programmation des systémes
Working with Graphics

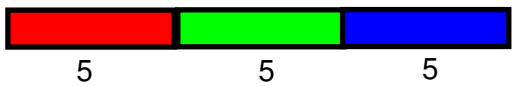
March 21, 2016

Basics

- No OS or any other library, we are on our own..
- place the bits in the right place(registers/memory locations) and get things done (as specified by hardware)
- Video processing can be done in different modes (mode 3,4 and 5)
- Choice of the mode depends on your application

Screen, Pixels, Colors and Video Buffer

- Screen is represented by **pixels**, where a pixel is assigned a color
- Color is represented by 16 bits (mode 3), 5 Red, 5 Green, 5 Blue, 1 unused



Screen, Video Buffer and Pixels

- Depending on the mode, resolution of the pixel grid is defined
- Graphic you want to draw (in the form of pixels) has to be written to the **video buffer**

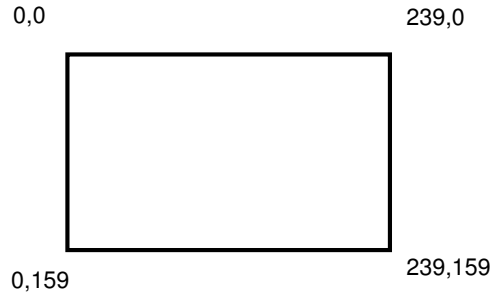


Figure : Mode 3 screen resolution **240*160**

- Video buffer is on the memory, starting from a specific memory address

Drawing Pixels

- Initialize the video mode (we work on mode 3 or mode 4 for this assignment)
  - The register responsible for this resides in 0x4000000 address
  - **#define REG_DISPCNT *(unsigned short *) 0x4000000**
- Assign the value to this register
  - **#define BG2_ENABLE 0x400**
  - **#define MODE_4 0x4**
  - **#define MODE_3 0x3**
  - **REG_DISPCNT = MODE_4 | BG2_ENABLE**
- Just assign a color to the respective pixel location (i, j) on the screen
- **videoBuffer[i*240+j]=color**

Some more on graphics

- Mode 4 has different resolution, color representation and video buffer structure
- Colors are defined in a palette, indexed by 8 bit integers
- Offers the possibility of **double buffering** to make animations more smoother
- Have a look at the GBA book chapter on graphics for more details