UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

INTRODUCTION TO MODERN C++

Lecture / Lab 7

Rémi Géraud

March 17, 2016

École Normale Supérieure de Paris

PROJECTS!

Projects:

- Don’t forget to send me a project description **for approval**
- TERs: Last opportunity to get your papers signed!

EXAM HIGHLIGHTS

Exam highlights:

- Exams have been graded. I just need to enter each grade.
- Who believes that C++ is simpler than C? Related to it?
- `g++ [source] [-o output] [–std=standard]`

Lecture 7
Graphics with SDL
The revenge of Angry Birds.

SITUATION & HISTORY

SITUATION

So far, we wrote to the terminal or files. That’s nice.

Most applications don’t need more.

Indeed, more than 90% of programs don’t.

But, Sir, we see graphical applications all the time! Tip of the iceberg.

GRAPHICS

Computers (and phones) were not originally designed with graphics in mind.

- Hardware was not really capable / talented at it
- Software was not available / practical
- Engineers didn’t have much choice

GRAPHICS

Engineers didn't have much choice and resorted to DIY

- Hacks
- Custom-made, non-standard hardware
- Incompatible approaches

Most of these solutions appeared and died out quickly.

GRAPHICS

It changed in the early 2000’s:

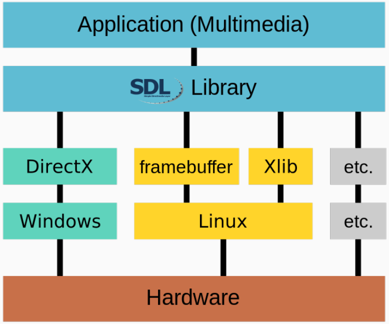
- Dedicated hardware (GPUs, screens)
- Cross-platform, cross-vendor APIs (OpenGL)
- Standards (Collada, etc)

The main problem: We still have widely different hardware.

SDL: THE SIMPLE DIRECTMEDIA LAYER

SDL: AN ABSTRACTION LAYER

SDL was originally developped by Sam Lantinga (former lead software engineer at Blizzard, now Valve).



Works on Android, iOS, Linux, Mac OS X, Raspberry Pi, PSP, and even Windows.

DEVELOPPING WITH SDL (OR ANY LIBRARY FOR THAT MATTER)

Three steps:

- **Install** the SDL library, and development headers (`.h` files)
- Write code **using** SDL functionality

  Note: All SDL-related material starts with `SDL_`
- Compile the program, **linking** to the library

  Note: With `g++` we use the `-lSDL2` option.

The only hard part: Learn how to use SDL functionality.

SOME GRAPHICS NOTIONS

SOME GRAPHICS NOTIONS: GPU VS CPU

On modern systems, tasks are divided between

- CPU: Logical tasks (computation, branching, looping)
  aka “The Brain”
- GPU: Data tasks (copying data, simple repetitive operations)
  aka “The Hand”

Most graphics operations consist in **copying** data from A to B.

The CPUs are really bad at that. We want to use the GPUs as much as possible.

SOME GRAPHICS NOTIONS: COMPOSITION

Remember: **Graphics don’t happen on the screen**.

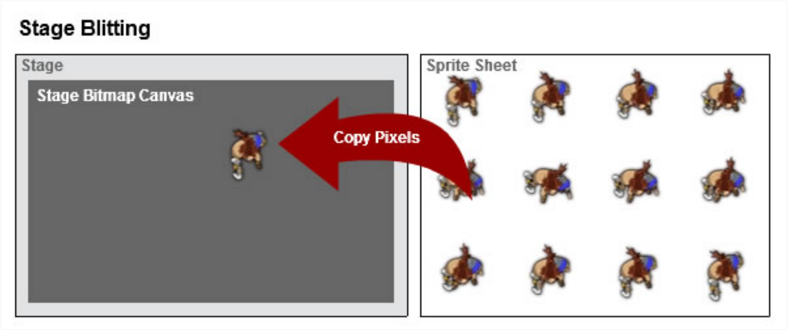
The screen is the **last** destination of data.

Pictures are assembled on the GPU memory first, then copied.

You can think of a “hidden canvas” that is showed after composition.

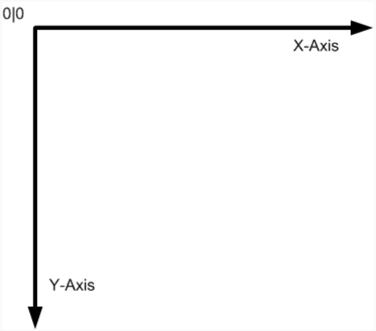
SOME GRAPHICS NOTIONS: 2D COMPOSITION BY BLITTING

To compose pictures, we copy and paste from rectangle regions:



SOME GRAPHICS NOTIONS: 2D COORDINATES

Beware: Coordinate systems are upside-down! (Why?)

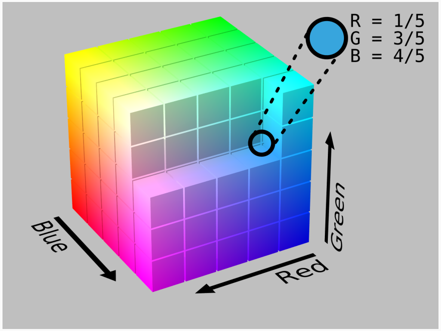


Coordinates are **integers**.

SOME GRAPHICS NOTIONS: PIXELS AND COLOURS

Each coordinate on the screen is called a pixel (short for “picture element”). It has a colour.

Every colour is a certain combination of Red, Green, and Blue.



An image is just a collection of coloured pixels.

SOME GRAPHICS NOTIONS: COLOURS AND ALPHA

With SDL, the quantity of each primary colour varies **from 0 to 255**.

There is also an additional quantity called “alpha” (also from 0 to 255) which controls `transparency`.

Examples:

- RGBA = (255, 0, 0, 255) is red, fully opaque
- RGBA = (0, 255, 0, 127) is green, half-transparent
- RGBA = (255, 255, 0, 255) is yellow, fully opaque
- RGBA = (32, 128, 255, 255) is teal, fully opaque

SOME GRAPHICS NOTIONS: SUMMARY

You follow essentially three steps:

1. Initialise everything, load resources
2. Compose picture by blitting
3. Present the finished picture to the screen

It’s a bit more detailed in 3D, but we won’t see it today.

THE LAB SESSION

THE LAB SESSION: WHAT WE'LL DO

- Install SDL (easy on Linux)
- Initialise the GPU and present data to screen
- Load pictures on the GPU to compose a scene
- Animations

THE LAB SESSION: TARGET

Questions?

Lab : The Revenge of Angry Birds