# Chapter 1

# Roadmap

## 1.1 SimpleOpenCL Version 0.010 27/02/2013

- v1 Incremented with increased functionality up to a certain goal or amount of changes (new functions)
- vv2 Incremented with bug corrections and functionality additions.
- date Set at the date of bug correction or any v1 or vv2 level modification.

## 1.2 Actual version 0.08

- Non finished but working version. A first finished version is scheduled for version 1.0
- It provides two structs to simplify the handling of OpenCL objects. They are conceptually grouped in hardware and software.
- It provides simplified Device memory allocation and copy functionalities.
- It also provides a function that creates a list of devices using the "clHard" struct for each device, and creates a context for that devices. The context will be the same for same type and same memory capacity devices. If any of those differ, then different contexts will be created for each device.
- Other functions select the desired devices from the list.
- In order to get information of OpenCL errors, there is a function that prints the OpenCL error flags returned by the OpenCL functions.
- Functions to load Device source code, compile it etc are present.
- Functions to enqueue or execute kernels, etc...

- A main 1.0 version goal is already implemented as "sclManageArgsLaunchKernel". It can with only host pointers, sclHard and sclSoft variables NDRange dimensions and a string containing the info of what to do with the pointers etc, execute the kernel and update the results on the host pointers. All in a single function call.

## 1.3 Next version

Goals for version 1.0:

- All hardware selection functions must use the sclGetAllHardware function first if a list of hardware is not passed as an argument and it has NULL value. Then, the functions must return the desired hardware following the function criteria expressed in its own name.
- The sclManageArgsLaunchKernel can/must be improved in the following ways:

1. Optional: a version that hides sclHard and sclSoft so the user only cares about which host pointers will be read and written from the device in OpenCL C code and which to use exclusively on the device. Nothing else. Think of when to initialize hardware? Possibly needing a global variable pointing to the sclHard and sclSoft objects to avoid repeating the software/hardware initialization process on each kernel execution.
2. Mandatory: the function must have the ability to schedule work across all the devices available. So maybe the function will need more info from the user to know which data can be partitioned, and which can not. The possibility of internally using something similar to GMAC would be wonderfull. Pagination of the pointers in my opinion is the most efficient automatic method to do that work, but repeating a work already done would be frustrating.