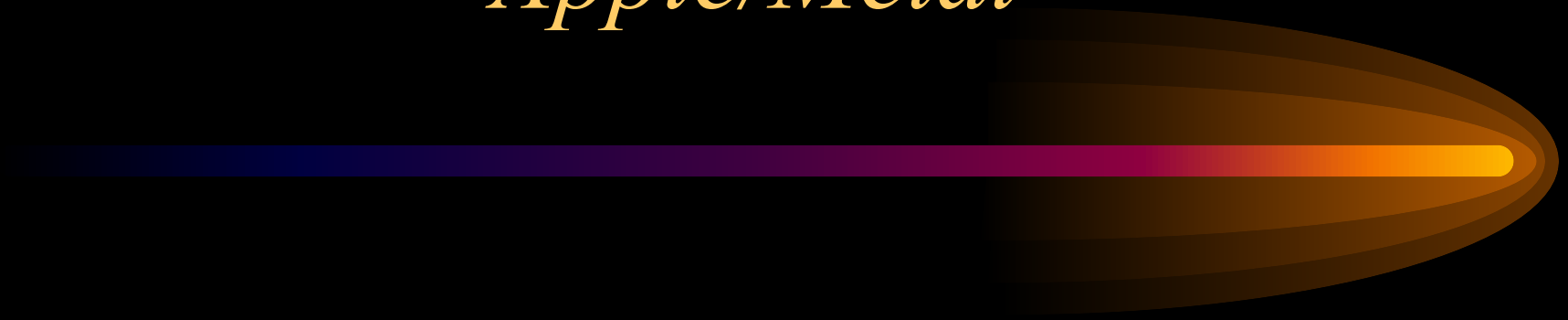
Apple/Metal



IJCLab dev meeting
15 September 2020

2018 :-(

WWDC June 2018 : Apple, in a // session, announced that the Apple/OpenGL is deprecated.

2018 :-(

- Bad news for people looking for a standard to do visualisation.
- Bad new for me and Geant4, and a lot of scientific software.
- Due to the impact of Apple concerning interactivity, we can't ignore that…
- Apple promotes their proprietary Metal in remplacement of OpenGL on their devices. We have to look!

- (No date given about a strong removal of Apple/OpenGL on macOS and iOS)

inlib/exlib/sg scene graph logic

- In spirit, same logic as the great OpenInventor.
- A scene is described by a graph of “nodes” in which, for example, a rotated red cube is described by a matrix (rotation) node, a color node and then a shape node.
- A graph is rendered on screen (or offscreen!) by using an implementation of a “renderer” for a given technology, for example OpenGL.
- See softinex at http://gbarrand.github.io

inlib/exlib/sg renderers

- GL-ES. It permits (today) with SAME CODE to visualise on Linux, macOS, Windows, iOS, Android.
- offscreen : to produce a .png, .jpeg, .ps, .pdf file without having to be tied to any graphics system (it is pure C++ code based on the std/stl libs).
- wasm : a web assembly version (using WebGL). It permits to display in most web browser.
- Then I have to provide a renderer for Apple/Metal…

Not so easy to do !

- API is in Objective-C or in Swift (the “better than Python” Apple language, dixit Apple).
- Apple examples are in Swift buildable from Xcode.
- Nothing in C++ buildable from a “simple make”.
- Stucked…

…up to the end of June 2020

- Some googling gave a hit on GitHub : naleksiev/mtlpp
- mtlpp : a C++ wrapper around Metal
- With an example to draw a triangle buildable with make: bingo!

- (As says a famous quote : “give me a triangle and I visualise the world”).

Summer 2020 at the forge…

- After two months of very painful coding, I have now one app (the ESSnu display) that works on macOS.
- And this by using straight the Objective-C Metal API from C++ (Apple clang permits to mix both languages).
- No extra libs involved.
- (It follows my “software least action principle”).
- Painful because the logic of Metal is not similar than GL-ES (even if ideas of rendering pipline, buffers, etc… are the same). We have to rethink a new renderer (which was not the case for offscreen and wasm ones).

Summer 2020 at the forge… (2)

- I have correct 3D rendering for basic primitives (points, lines, segments, triangles, triangle-fan and strip).
- I have lighting.
- I have texture mapping.
- With that I can my apps working on Metal.

And be sure it had not be easy to get !

Then "ouf" !

- An iOS version has to be done (I am on it).
- In principle I am now ready for what Apple prepares for the future.
- (I strongly suspect that they are going to remove their OpenGL when the macOS major release, running on their own Ax processors, is going to come).

Can it help for Geant4 ?

- My apps g4exa, g4view should run with Metal and I am going to release versions of these.
- But it is not based on the "G4 vis system" largely used now.
- The G4 vis system is in principle designed to handle multiple heterogenous graphics systems (= drivers).
- For example there is an OpenInventor driver and some offscreen ones (HepRep, VRML).

Can it help for Geant4 ? (2)

- Right now what is promoted by Geant4 is Qt for the GUI and the old OpenGL for the graphics.
- Not so clear how to move…
- Can we mix Qt with some QWidget doing Metal ? It would need special programming for macOS/iOS : lot of work needed.
- Some Qt vis lib having a hidden QMetal renderer on macOS, iOS? This would need to have a “Q-vis-lib” driver : lot of work needed.
- An inlib/exlib driver ? lot of work needed. (Knowing that time is running for me).

Waiting that…

- I can surely release my code and apps. (There is probably few scientific C++ apps running straight on Metal right now).
- It would deserve some paper in some future C(OVID)HEP (probably my last “R&D” one).
- What is sure is that I am eager to see what Apple is going to do in the next years…
- And, good timing, there is keynote today.