Updates on QtViewer / vis

GEANT4
A SIMULATION TOOLKIT

Laurent Garnier IRISA / INS2I / CNRS

Geant4 collaboration meeting, documentation working session

Well…

- **No major developments on QtDriver**
- **Lot of bug fixes**

Vis work plan

- **OpenGL drivers**
  - Adapt to newer OpenGL versions, exploit new functionalities and replace deprecated calls such as glBegin/glEnd (2/*)
- **Qt OpenGL driver:**
  - Continue development (*)
  - Improve toolBar by adding useful icons (viewpoints,search…)
- **Other drivers**
  - OpenInventor: Continue development of G4OpenInventorXtExtendedViewer (*)
  - gMocrenFile and gMocren: Continue development (updating in order to support visualization attributes and other information) (*)
  - G4DAE exporter: Creation of a new driver in order to export in Collada format (*)
  - iOS and Android Devices: Develop visualization solutions for these devices (*)
- **Others**
  - Vis driver for ParaView: Creation of a new driver in order to export in a format readable by Paraview (1/2)
  - Documentation: Major overhaul of visualization documentation (2/*)
  - Wt driver: Rewrinting code (2)
  - New Transparent Visualization Tool: Develop new tool to support high resolution transparent visualization with ability to rotate and zoom, able to show, for example, trajectories and hits within a HEP detector - (*)
  - Add PDF3D driver (*)
  - GDML: Study addition of visualization attributes to this exchange format (*)
  - Multi-threading: Support user-drawn primitives in MT mode (*)
  - Driver for Satoshi's renderer?

Other vis topics

- **Apple: Migration to Metal ? When / How ?**

Deprecation of OpenGL and OpenCL

Apps built using OpenGL and OpenCL **will continue to run in macOS 10.14**, but these legacy technologies are **deprecated in macOS 10.14.** Games and graphics-intensive apps that use OpenGL **should now adopt Metal**. Similarly, apps that use OpenCL for computational tasks should now adopt Metal and Metal Performance Shaders.

Metal is designed from the ground up to provide the best access to the modern GPUs on iOS, macOS, and tvOS devices. Metal avoids the overhead inherent in legacy technologies and exposes the latest graphics processing functionality. Unified support for graphics and compute in Metal lets your apps efficiently utilize the latest rendering techniques. For information about developing apps and games using Metal, see the developer documentation for Metal, Metal Performance Shaders, and MetalKit. For information about migrating OpenGL code to Metal, see Mixing Metal and OpenGL Rendering in a View.

Questions ?