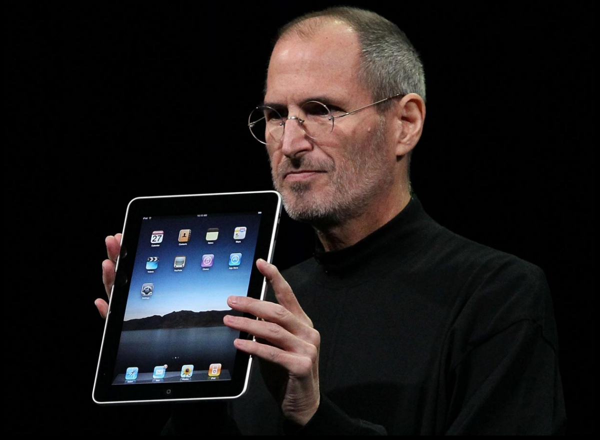
Exploring Geant4 on iOS and Android

Geant4 workshop 2018 at Lund

G.Barrand, CNRS/IN2P3/LAL

2010

- iPad :

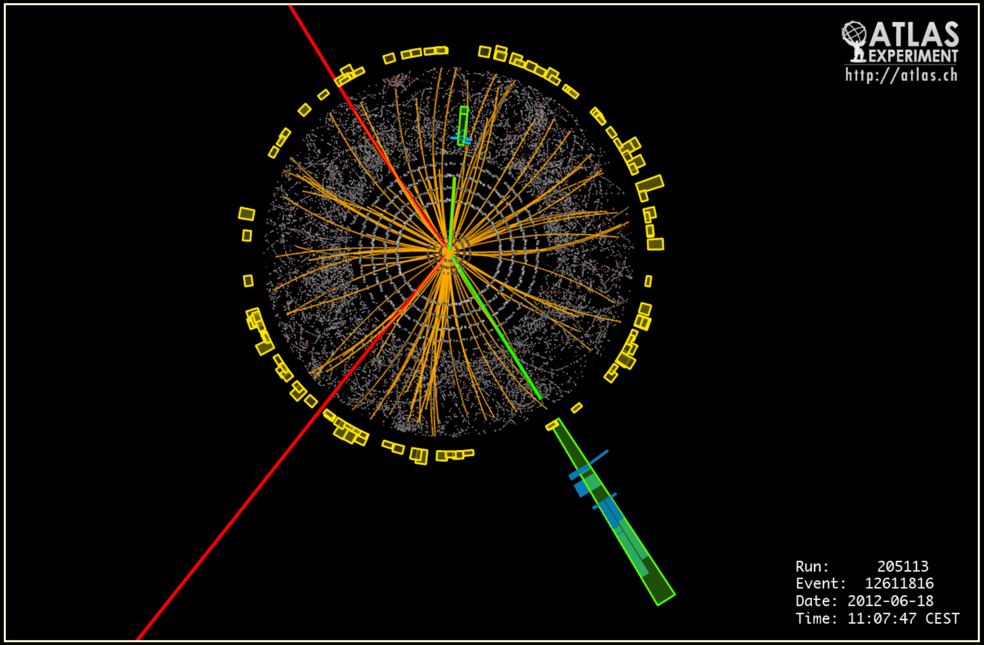


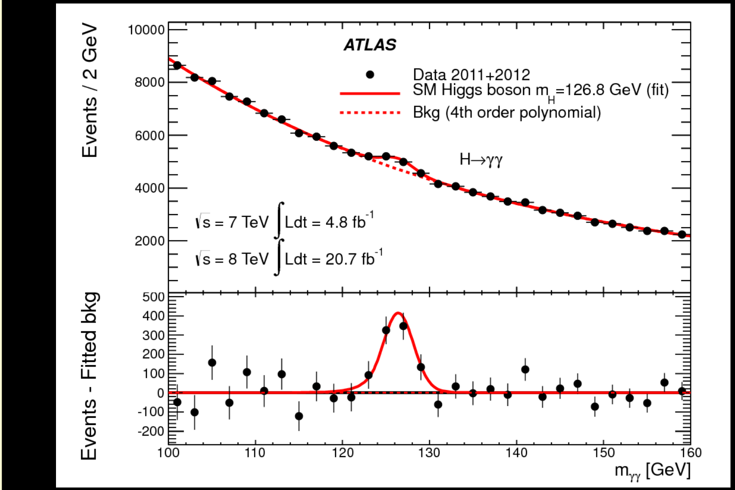
- Android tablets shortly after.
- The « app store » : Apple AppStore, GooglePlay.
- Little wall of screens at LAL :

I reconsider the way I do graphics and interactivity in general around my activity in HEP.

Redo the graphics ?

- In fact the kind of graphics that we do in HEP experiments :





- Then squares, lines and points with, let’s be crazzy, some text !

mid-2010 / begin 2011 : at work !

- Before : OpenGL, Coin3D, Motif, gtk, Qt, scripting (tcl, python) : none of these were available on iOS & Android.

iOS : Objective-C, UIKit, GL-ES.

Android : java, View, GL-ES.

- But C/C++ and GL-ES are common : we build on that. Also available on Linux, macOS, Windows !

- Huge expertise of OpenInventor (Coin3D)
- I redo (refactor) a « scene graph manager ».
- We do also the GUI with that ! (then on GL-ES).
- 99% of the code is common for all platforms.

Then…

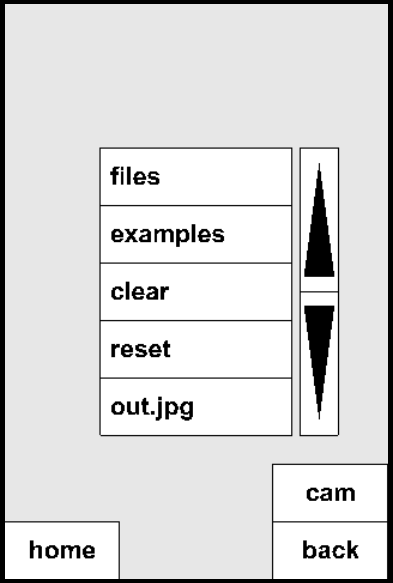
- iOS : we do a minimum of Objective-C.
- Android : we do a minimum of java (a maximum of NDK).

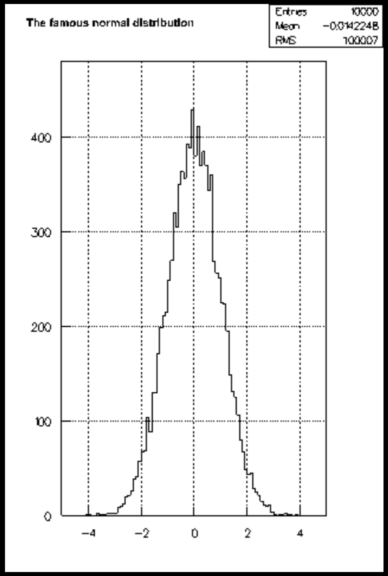
- To Tim, Sundar, Satya, Linus I ask only : one C++ compiler, one area to do some GL-ES and a way to get "touch" (or "pick") events and that's all !

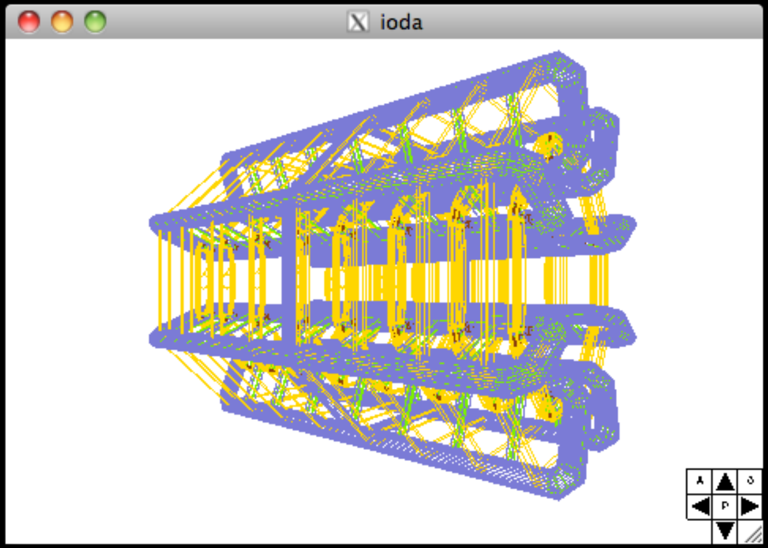
- And with that (and four devices)
  I redo my needed world.
  (which is not all the world)

Begin 2011 / ioda

- IODA : first « app » on the stores oriented « analysis » : visualisation of histograms from a local file at the CERN-ROOT format and a little bit of detector (format Saclay/fog) :







- First of all amazingly easy world wide deployement through the Apple Store and Android Market (now Google Play) : huge progress !
- 99% of the code common to all platforms.

ioda / softinex

- IODA gently evolves toward a viewer of various file formats :
  - HEP : root (histograms, ntuples, geometries), gdml (geometries Geant4 read with expat), heprep (geometries).
  - Astro : fits (astro images, tables)
  - hdf5 (histograms, ntuples).
  - Also : VRML (with Coin3D finally ported on iOS and Android), png, jpeg.

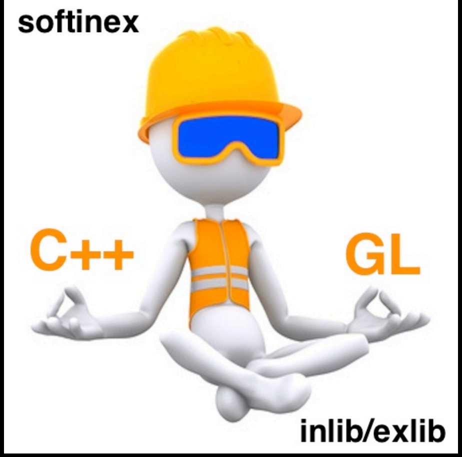
- I do the port of : cfitsio, hdf5, Coin3D, png, jpeg, etc...

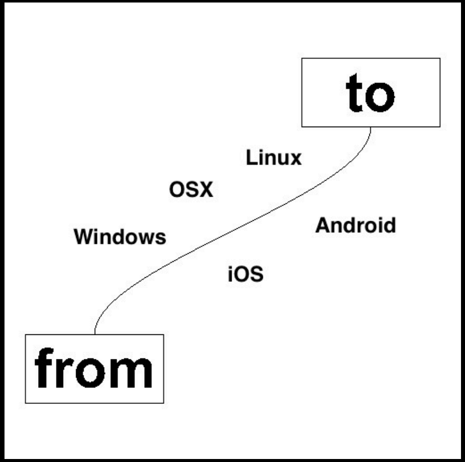
- We build a code toolbox of all that in the softinex libraries (inlib, exlib, ourex). http://softinex.lal.in2p3.fr (http://gbarrand.github.io)
- 2018 GitHub : http://github.com/gbarrand/ioda.git

softinex

- inlib : code C/C++ « pure header » highly portable.
- exlib : code C/C++ doing the relationship with « external packages ».
- ourex : versions of « critical » external packages as freetype, expat, png, jpeg, zlib, zip for which we master, embark the sources. But also cfitsio, hdf5, Coin3D, lua, Python (2.7) and… Geant4 (yes, yes).

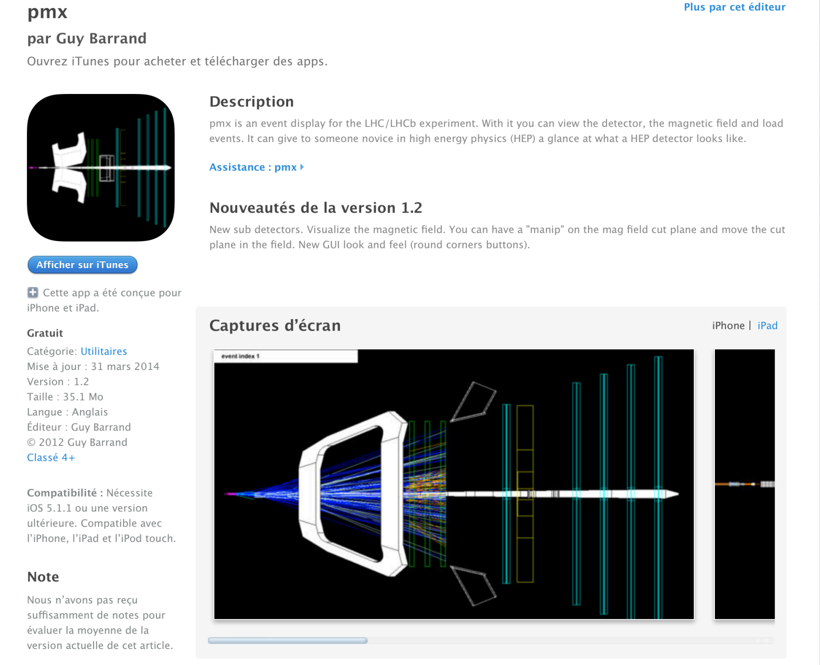
- In softinex, there are strong choices about the developments... (close to a philosophy ☺)





pmx

Demonstrator app that can show the LHCb detector (read from a root file) and can show some tracks of data from a dst root file.



pmx

Plus par cet éditeur

par Guy Barrand

Ouvrez iTunes pour acheter et télécharger des apps.

Description

pmx is an event display for the LHC/LHCb experiment. With it you can view the detector, the magnetic field and load events. It can give to someone novice in high energy physics (HEP) a glance at what a HEP detector looks like.

Assistance : pmx ›

Nouveautés de la version 1.2

New sub detectors. Visualize the magnetic field. You can have a "manip" on the mag field cut plane and move the cut plane in the field. New GUI look and feel (round corners buttons).

Cette app a été conçue pour iPhone et iPad.

Gratuit

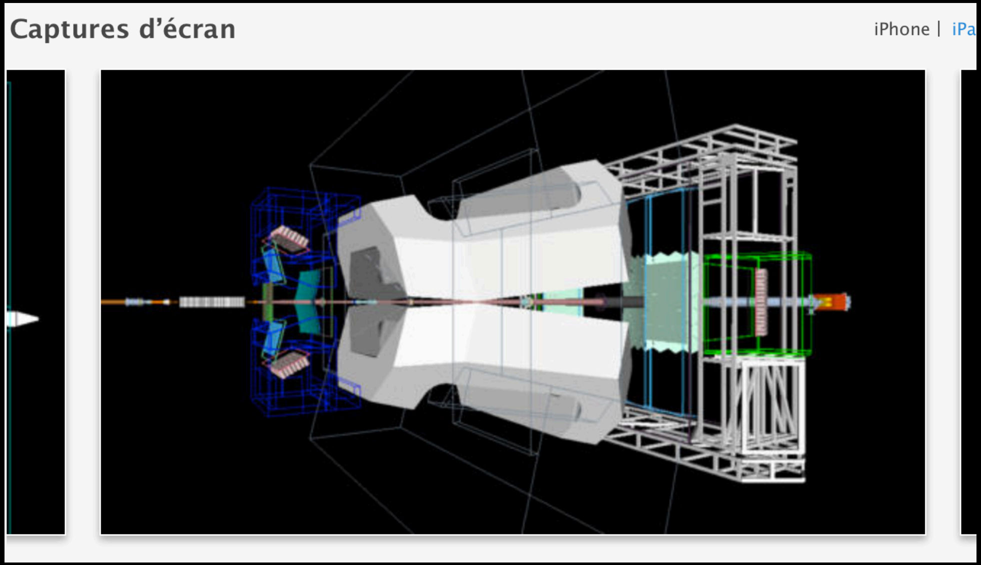
Catégorie: Utilitaires
Mise à jour : 31 mars 2014
Version : 1.2
Taille : 35.1 Mo
Langue : Anglais
Éditeur : Guy Barrand
© 2012 Guy Barrand
Classé 4+

Compatibilité : Nécessite iOS 5.1.1 ou une version ultérieure. Compatible avec l'iPhone, l'iPad et l'iPod touch.

Note

Nous n'avons pas reçu suffisamment de notes pour évaluer la moyenne de la version actuelle de cet article.

Captures d'écran



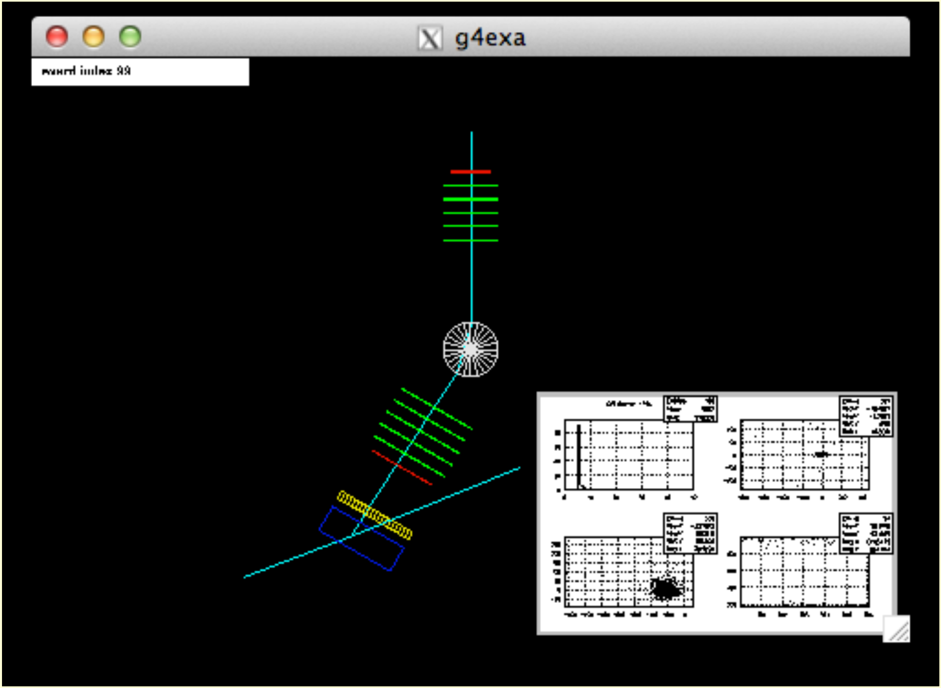
Clearly a « wow effect » from people to whom I show that.

Geant4…

- Started with the code of the 9.3 release. Now 10.03.p01.
- give all .cc and include access paths to the Android/SDK and Xcode build systems. Have a couple of -DG4 (-DG4VERBOSE, –DG4_STORE_TRAJECTORY).
- iOS : clash with Apple that does not authorize the usage of "system()" here. Used in a couple of place in the core code, but not needed for what we want to do. #if TARGET_OS_IPHONE to disconnect them.
- Else go on ! No huge resistance to build the Geant4 core here.
- WARNING : we do not seek to build libs ! but build apps, moreover by doing static linking.
- (By the way I did the port of Lund/pythia8 ! SUSY resisted a lot).

First app : g4exa

- From the extended/A01 example of 9.3

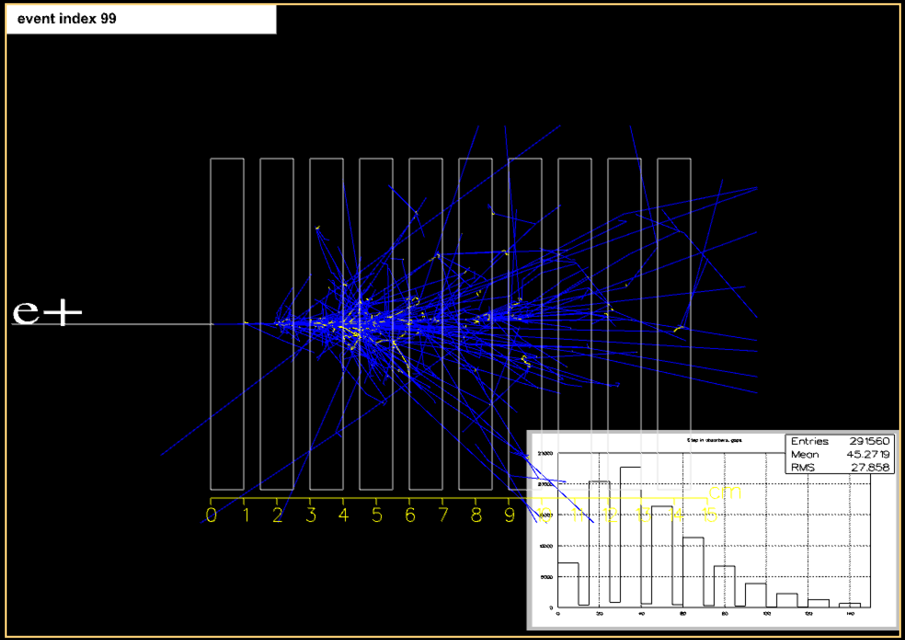


event index 99

- Rejected by Apple : « we do not want examples/demonstrators, we want apps that do something ».
- It exists on GooglePlay.
- Can serve as a template to create your own app !

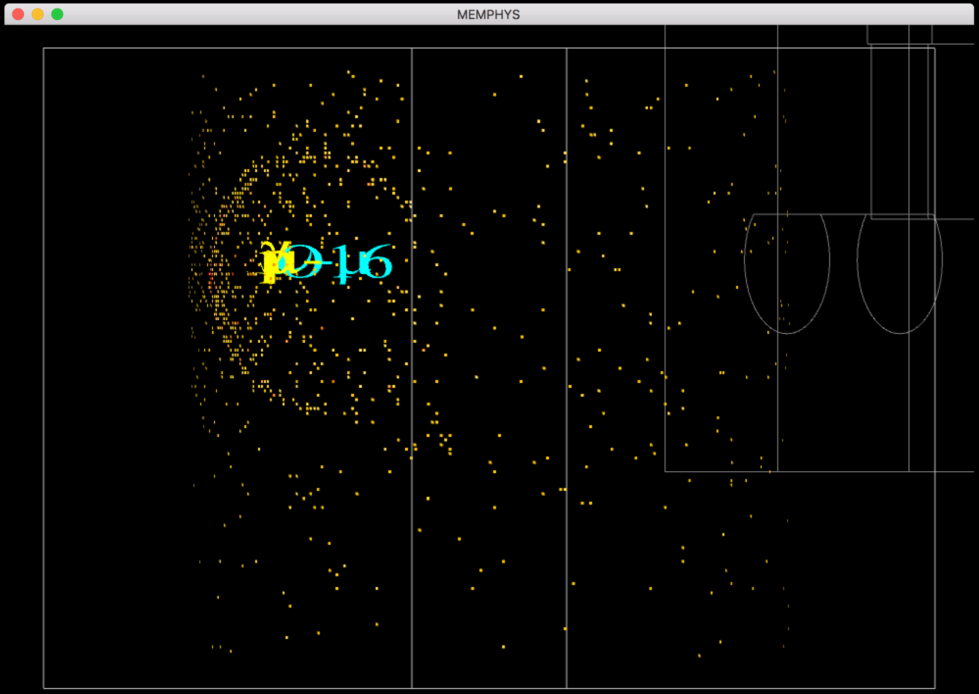
Second app : g4view

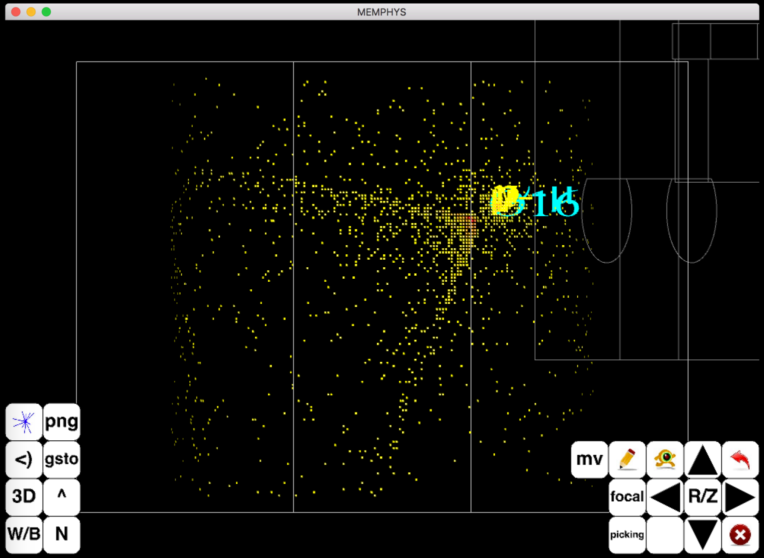
- An app presented as a « viewer » for some scientific data (then gdml files). Apple had been happy with that.
- The calorimeter example has clearly a « wow effect ».
- Comparison of what looks like an EM shower versus a proton shows interest. You can show from your pocket, the basement of proton therapy.



Third app : MEMPHYS_vis

- MEMPHYS : a water Cherenkov neutrino detector simulation revivied (long story) in relation to ESSnuSB based... at Lund! (European Spallation Source neutrino Super Beam) (!).





Main issue : G4 data files

- Size of exe and packing are limited on the stores (100 MB is a good marker).
- On private packing, we do what we want.
- For g4exa, g4view, I extracted and brought what is needed to run the app. Packings less than 50 Mbytes !
- But, due to needs around gaming, “extensions” mechanism exists so that an app on the stores can get data from elsewhere. To be explored...
- Have a “load on demand” mechanism in Geant4 ?

G4/vis

- G4/vis permits to users to setup “scenarios” not available in my apps, and this in a scripted way.
- The structure of my apps permits to integrate G4 intercoms. (I have already Python, lua, KUIP).
- But today none of the vis drivers are portable iOS/Android.
- Have OGL => OGL-ES ? (Yes, but Apple deprecates OpenGL !)
- Have a inlib/sg driver ?

Then…

- Anyway, my apps are demonstrators for the moment : but we can deal with these technologies!
- Definitely a HUGE challenge around the ergonomy.
- Hell, how far behind are we from the ergonomy of Apple apps !
- A huge potential for education/outreach.

For physics…

- For display, we have good performances and can cover a lot. (One Terabyte on the Galaxy Note 9 !)
- HEP : clearly a problem to bring data, the related detector/event models and IO reader on these devices for the today experiments. We need a new generation of frameworks thought to be portable. Can the HSF help ?
- Astronomers are on the stores for long now!
- Geant4, being portable on iOS and Android, can help to show the way !