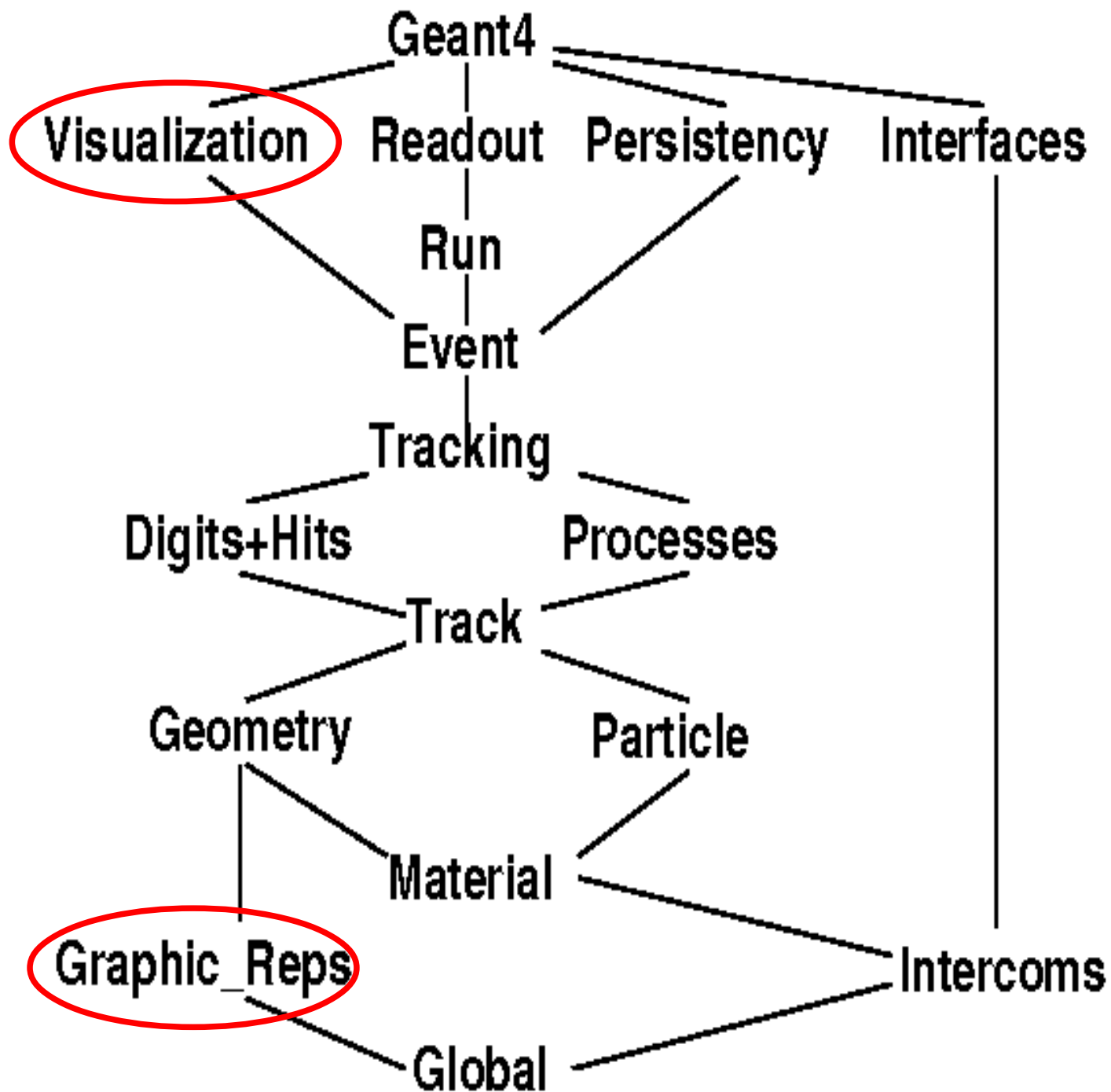


# Визуализация детектора и событий



# Что можно визуализировать

## *Элементы модели*

- Объемы детектора или отдельных подсистем
- Траектории частиц
- Срабатывания в чувствительных объемах
- Значения счетчиков

## *Кроме того, по желанию пользователя*

- Линии, например оси координат
- Маркеры
- Текст
- Масштабные линейки

# Графические системы Geant4

Своей особенной графической системы в Geant4 нет. Вместо этого есть набор модулей-драйверов к имеющимся внешним пакетам:

*Визуализация напрямую из Geant4*

- OpenGL
- OpenInventor
- Qt

*Создание файла + внешняя программа просмотра*

- HepRep/WIRED
- DAWN
- VRML
- gMocrenFile

*Только геометрия детектора*

- RayTracer
- AsciiTree

# OpenGL

- Требует библиотек GL
- Реалистичное 3D изображение
- Ограниченная интерактивность
- Высокая скорость рисования (особенно с аппаратным ускорением)
- Возможны поворот, смещение, увеличение
- Возможно сохранение в Encapsulated Postscript (векторный или растровый)

# Qt

- Требует библиотек Qt и GL
- Реалистичное 3D изображение
- Высокая интерактивность
- Высокая скорость рисования (особенно с аппаратным ускорением)
- Возможны поворот, смещение, увеличение
- Возможно сохранение в различных форматах (векторных или растровых)
- Можно делать анимационные фильмы

# OpenInventor

- Требуется OpenInventor (только в Linux)
- Реалистичное 3D изображение
- Высокая интерактивность
- Высокая скорость рисования (особенно с аппаратным ускорением)
- Возможны поворот, смещение, увеличение
- Возможно сохранение в различных форматах (векторных или растровых)

# HerRep

- Требуется внешняя программа-просмотрщик (HerRApp, FRED, WIRED4)
- Высокая интерактивность
- Отрисовка прямыми линиями и заливкой цветом
- Дополнительная информация по клику мыши (энергия, импульс и т.д.)
- Просмотр иерархии объемов
- Сохранение картинки в векторные форматы (PS, PDF)



# DAWN

- Требуется внешняя программа-просмотрщик DAWN
- Интерактивности нет
- Реалистичное 3D-изображение
- Сохраняет Postscript высокого качества

# VRML

- Требуется внешняя программа-просмотрщик VRML
- Ограниченная интерактивность
- Возможны поворот, смещение, увеличение
- Сохранение в растровые форматы

# RayTracer

- Для рисования используется трекинг фотонов средствами Geant4
- Можно визуализировать только объемы (любой сложности)
- Результат сохраняется в виде JPEG

# gMosren

- Требуется внешнее приложение - программа gMosren (только Windows)
- Реалистичное 3D изображение
- В основном применяется в медицинских приложениях

# ASCIITree

- Печать иерархии объемов
- Графический вывод отсутствует
- Можно вычислить массу и объем любой группы объемов
- Пример:

```
"World":0
```

```
  "Target":0
```

```
  "Tracker":0
```

```
    "Chamber":0
```

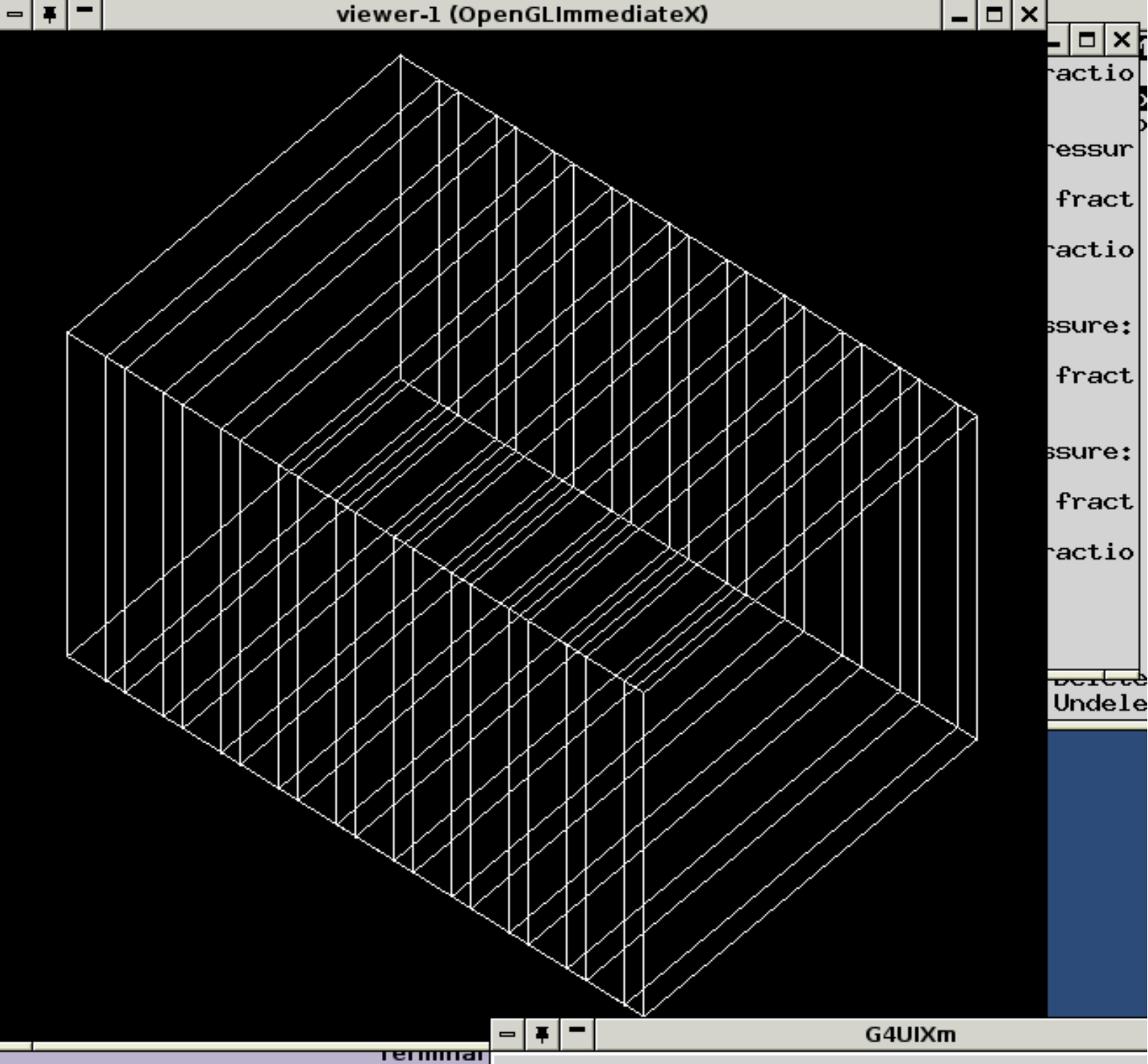
```
    "Chamber":1
```

```
    "Chamber":2
```

```
    "Chamber":3
```

```
    "Chamber":4
```

```
G4ASCIITreeSceneHandler::EndModeling
```





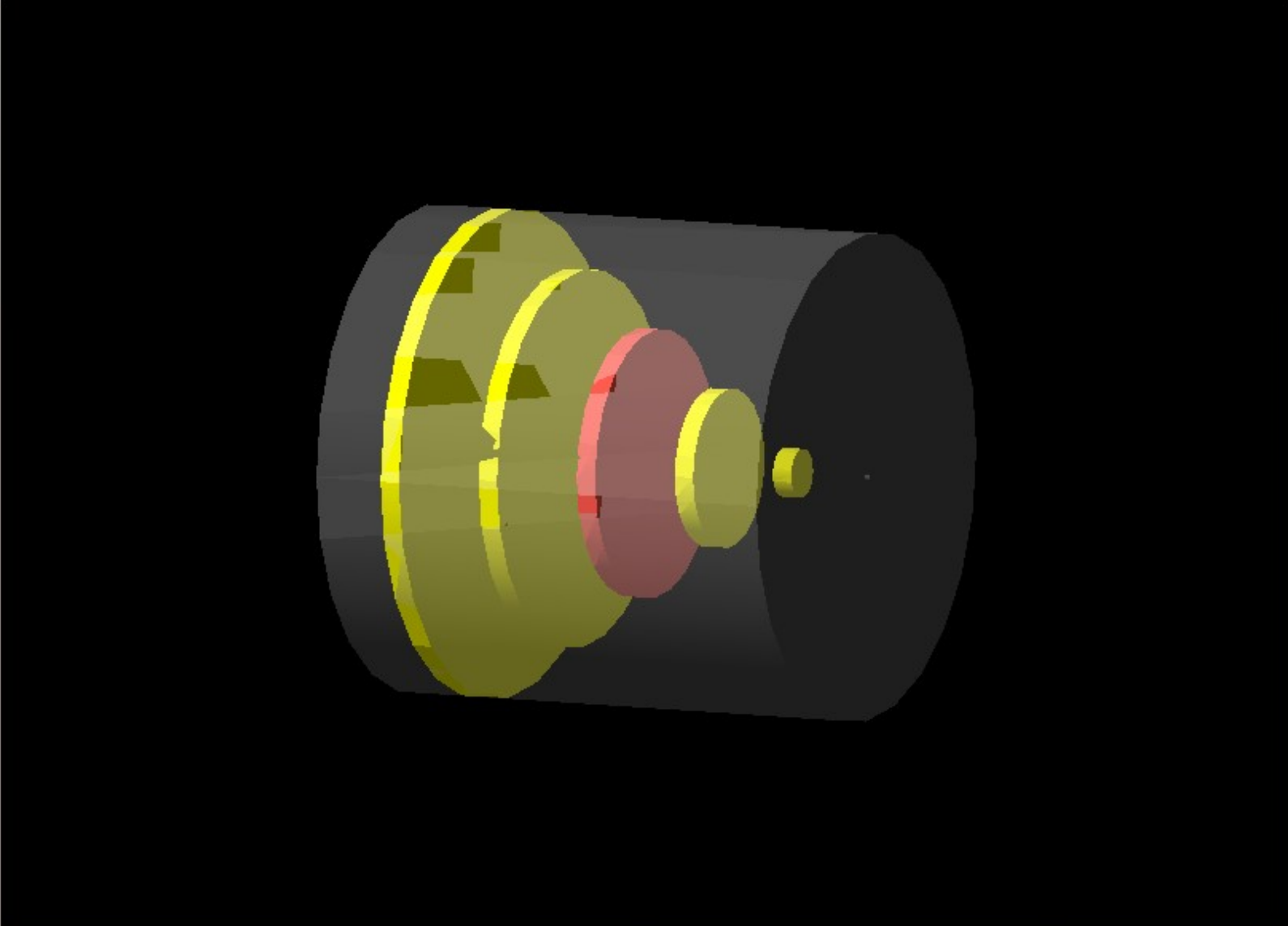
Scene tree Help History

viewer-0 (OpenGLStoredQt) [X]

viewer-0 (OpenGLStoredQt)

Scene tree : viewer-0 (OpenGLSto

- [-]  Touchables
  - [-]  World [0]
    - Target [0]
    - [-]  Tracker [0]
      - Chamber [0]
      - Chamber [1]
      - Chamber [2]
      - Chamber [3]
      - Chamber [4]



Touchable slider



Search :  select item(s)

Session :

# Включение визуализации в программу моделирования

- Наличие драйвера графической системы
- Инициализация объекта VisManager в программе моделирования
- Выбор элементов модели, которые должны быть визуализированы
- Управление визуализацией командами категории /vis
- (Использование внешней программы просмотра)



# Драйверы графической системы

<i>Драйвер</i>	<i>Графическая система</i>	<i>Платформа</i>
OpenGL-Xlib	OpenGL	Linux, Mac + Xlib
OpenGL-Motif	OpenGL	Linux, Mac + Motif
OpenGL-Win32	OpenGL	Windows
Qt	OpenGL	Linux, Mac, Windows
OpenInventor-X	OpenInventor, OpenGL	Linux, Mac+Xlib или Motif
OpenInventor-Win32	OpenInventor, OpenGL	Windows
HepRep	WIRED (JAS)	Linux, Mac, Windows
DAWNFILE	Fukui Renderer DAWN	Linux, Mac, Windows
DAWN-Network	Fukui Renderer DAWN	Linux
VRMLFILE	просмотр VRML	Linux, Mac, Windows
VRML-Network	просмотр VRML по сети	Linux
RayTracer	просмотр JPEG	Linux, Mac, Windows
ASCIITree	нет	Linux, Mac, Windows
GAGTree	GAG	Linux, Mac, Windows
XMLTree	просмотр XML	Linux, Mac, Windows
gMocrenFile	gMocren	Windows only

# Дополнительная информация о графических системах

- **OpenGL**

*<http://www.opengl.org>*

- **OpenInventor**

*<http://www.sgi.com/software/inventor>*

- **Qt**

*<http://geant4.in2p3.fr/spip.php?article60&lang=en>*

- **HepRep/WIRED**

*<http://www.slac.stanford.edu/BFROOT/www/Computing/Graphics/Wired/>*

- **VRML**

*<http://www.vrmlsite.com/>*

- **DAWN**

*[http://geant4.kek.jp/~tanaka/DAWN/About\\_DAWN.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html)*

- **gMocren**

*<http://geant4.kek.jp/gMocren/>*

# Создание объекта VisManager в программе моделирования

//----- C++ source codes: Instantiation and initialization of G4VisManager

```
.....  
// Your Visualization Manager  
#include "G4VisExecutive.hh"  
.....  
  
// Instantiation and initialization of the Visualization Manager  
#ifndef G4VIS_USE  
G4VisManager* visManager = new G4VisExecutive;  
visManager -> Initialize ();  
#endif  
  
.....  
#ifndef G4VIS_USE  
delete visManager;  
#endif  
  
//-----
```

# Визуализация детектора

В каждом логическом объеме можно задать параметры визуализации, содержащиеся в объекте класса `G4VisAttributes`

```
VolumeLog->SetVisAttributes(G4VisAttributes* pObj);
```

# G4VisAttributes - параметры ВИЗУАЛИЗАЦИИ

```
G4VisAttributes * myVisAtt = new G4VisAttributes();
```

```
myVisAtt -> SetVisibility(true);
```

```
myVisAtt -> SetColour(G4Colour(1.0, 1.0, 1.0)); // RGB
```

*или*

```
myVisAtt -> SetColour(1.0, 0.0, 0.0);
```

*или*

```
myVisAtt -> SetColour(G4Colour::Red());
```

```
myVisAtt -> SetForceWireframe(true);
```

# Взаимодействие визуализируемых объектов и графической системы

**Строится по следующей схеме:**

- **Scene (сцена)** – набор 3D элементов, которые нужно отобразить
- **SceneHandler (обработчик сцены)** – производит на основе сцены графическое представление
- **Viewer (проектор)** – прорисовывает графическое представление, используя функции графической системы

# Процедура визуализации

- Создаются проектор и обработчик сцены
- Создается пустая сцена
- К сцене добавляются объекты, которые нужно визуализировать (геометрия, траектории, примитивные формы и т.д.)
- Сцена передается обработчику
- Задаются параметры проектора (положение камеры, стиль прорисовки линий, прозрачность и т.д.)
- Проектор рисует объекты сцены
- Обновляется экран проектора

# Как это выглядит в командной строке

- `/vis/open OGL`
- `/vis/drawVolume`
- `/vis/scene/add/axes 0 0 0 500 mm`
- `/vis/scene/add/trajectories`
- `/vis/scene/add/text 0 0 0 mm 40 -100 -200 Text`
- `/vis/scene/add/hits`
- `/vis/viewer/flush`
- `/vis/viewer/set/viewpointThetaPhi 20 20`
- `/vis/viewer/refresh`