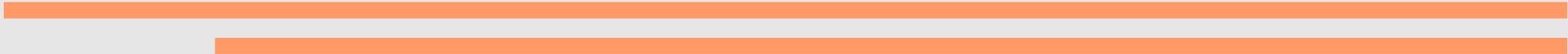
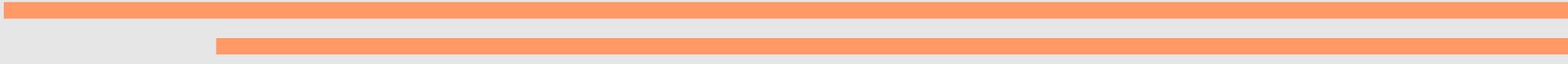
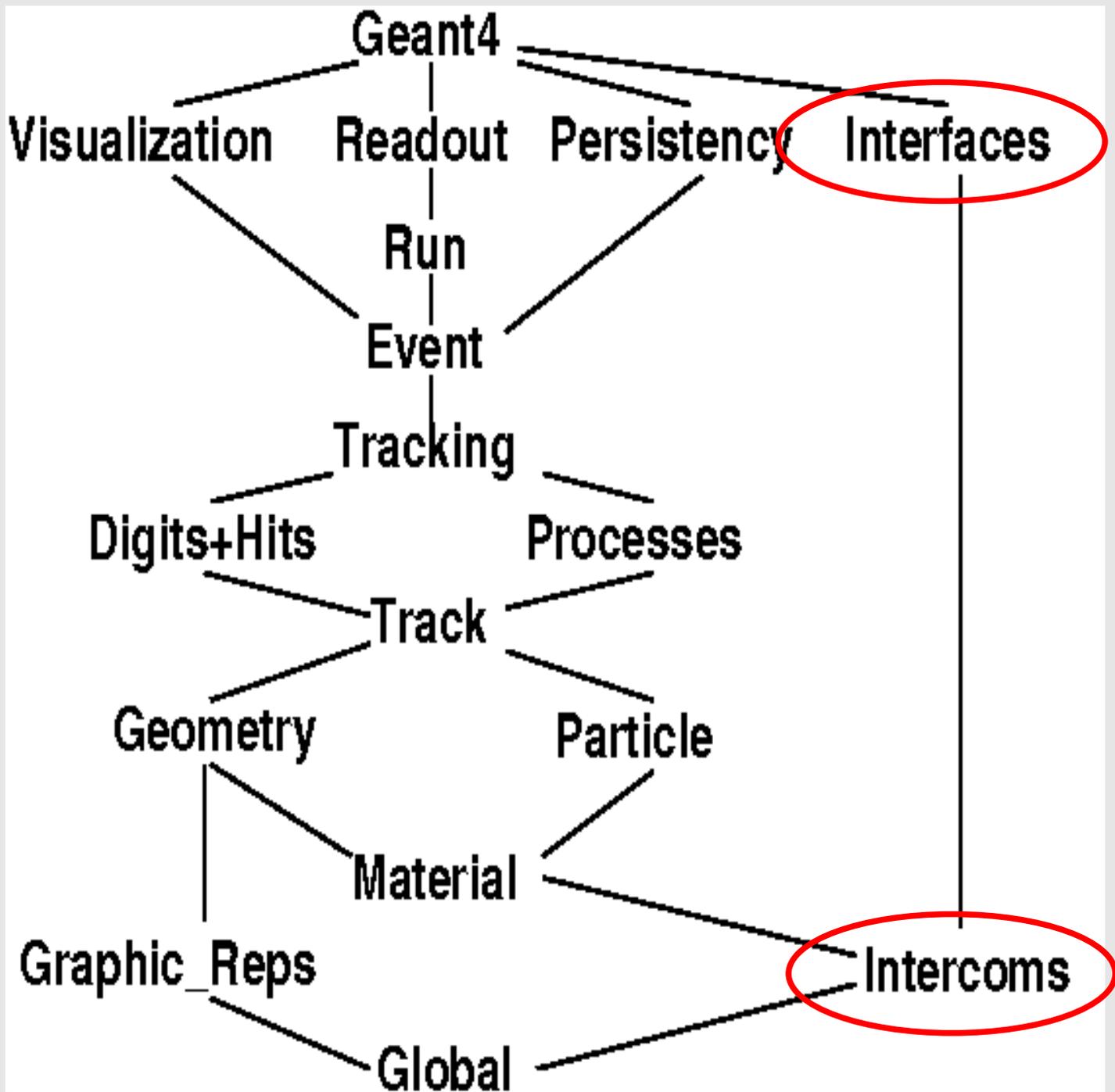
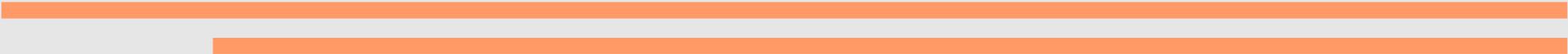


# ***Интерфейс пользователя***





# ***Взаимодействие пользователя с программой моделирования***

- Пакетный режим
  - Пакетный режим, управляемый сценарием
  - Интерактивный режим с командной строкой
  - Интерактивный режим с графическим интерфейсом
- 

# Управление программой моделирования

- Geant4 имеет встроенный набор команд, управляющих моделированием
    - уровень диагностики
    - изменение геометрических параметров модели
    - изменение списка учитываемых процессов
    - ...
  - Этот набор может быть расширен пользователем
  - Команды могут использоваться в интерактивном режиме, считываться из файла сценария или быть запрограммированы в коде
- 
-

# Категории команд

/control	управление интерфейсом
/units	система единиц
/geometry	навигация и проверка перекрытий объемов
/tracking	управление объектами TrackingManager и SteppingManager
/event	управление объектом EventManager
/cuts	управление порогами рождения частиц
/run	управление сеансом
/random	управление генератором случайных чисел
/material	просмотр списка и свойств материалов
/particle	изменение списков и свойств частиц
/process	изменение списка активных процессов
/vis	визуализация
/gun	управление генератором первичной вершины
/hits	управление детектирующими объемами
/score	работа со счетчиками

---

---

# ***/control***

- 1) **execute** \* Execute a macro file.
  - 2) **loop** \* Execute a macro file more than once.
  - 3) **foreach** \* Execute a macro file more than once.
  - 4) **suppressAbortion** \* Suppress the program abortion caused by G4Exception.
  - 5) **verbose** \* Applied command will also be shown on screen.
  - 6) **saveHistory** \* Store command history to a file.
  - 7) **stopSavingHistory** \* Stop saving history file.
  - 8) **alias** \* Set an alias.
  - 9) **unalias** \* Remove an alias.
  - 10) **listAlias** \* List aliases.
  - 11) **shell** \* Execute a (Unix) SHELL command.
  - 12) **manual** \* Display all of sub-directories and commands.
  - 13) **createHTML** \* Generate HTML files for all of sub-directories and commands.
  - 14) **maximumStoredHistory** \* Set maximum number of stored UI commands.
- 
-

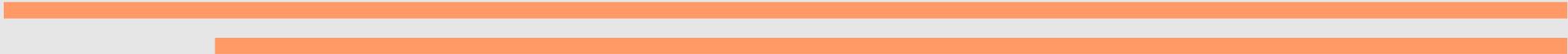
# ***/control/execute***

Выполнение сценария:

**/control/execute имя\_файла\_сценария**

Например

**/control/execute ~/run.mac**



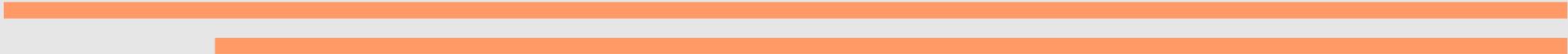
# ***/control/shell***

Выполнение команды оболочки Unix

**/control/shell команда**

Например

**/control/shell ls -la**



## ***/units/list***

Пример:  
Idle> /units/list

----- The Table of Units -----

**Вывод списка  
используемых  
единиц**

**category: Length**  
parsec ( pc) = 3.08568e+19  
kilometer ( km) = 1e+06  
meter ( m) = 1000  
centimeter ( cm) = 10  
millimeter ( mm) = 1  
micrometer (mum) = 0.001  
nanometer ( nm) = 1e-06  
angstrom (Ang) = 1e-07  
fermi ( fm) = 1e-12

**category: Surface**  
kilometer2 ( km2) = 1e+12  
meter2 ( m2) = 1e+06

. . .

---

---

# */geometry*

Два подкаталога

*/geometry/navigator*

*/geometry/navigator/reset* - отладка трекинга

*/geometry/navigator/verbose*

*/geometry/test* - проверка геометрии на перекрытие  
объемов

*/geometry/test/tolerance* (по умолчанию 0.1 микрон)

*/geometry/test/position*

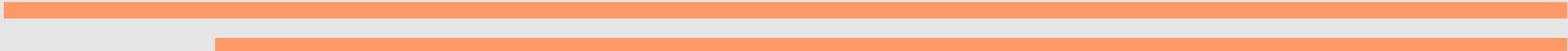
*/geometry/test/direction*

*/geometry/test/run*

*/geometry/test/line\_test* */geometry/test/recursive\_test*

*/geometry/test/grid\_test*

*/geometry/test/cylinder\_test*

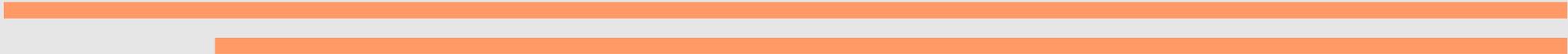


# *Пример проверки геометрии*

Idle> /geometry/test/grid\_test

GeomTest: no daughter volume extending outside  
mother detected.

GeomTest: no overlapping daughters detected.



# ***/tracking***

*/tracking/abort* - прервать моделирование  
текущего трека

*/tracking/resume* - возобновить моделирование  
текущего трека

*/tracking/storeTrajectory* – сохранять траектории  
частиц

*/tracking/verbose* – уровень диагностики трекинга

---

---

# ***/tracking/verbose***

***/tracking/verbose*** уровень\_детализации

***Уровни детализации диагностики***

0 : Сообщения отсутствуют

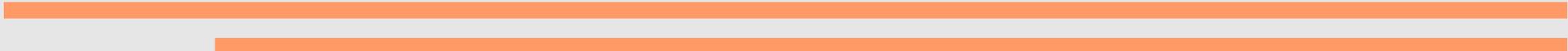
1 : Минимальная информация о каждом шаге

2 : Уровень=1 + информация о вторичных частицах

3 : Дополнительно информация о состоянии в начале и конце каждого шага

4 : Дополнительно информация о состоянии в начале и конце каждого шага с детализацией по отдельным процессам

5 : Дополнительно информация об оценках длины шага согласно каждому отдельному процессу

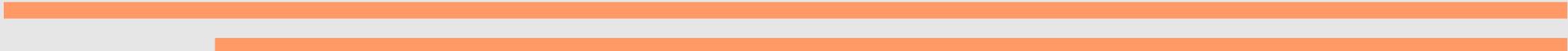


# ***/tracking/storeTrajectory***

*/tracking/storeTrajectory 0* - траектории частиц не сохраняются

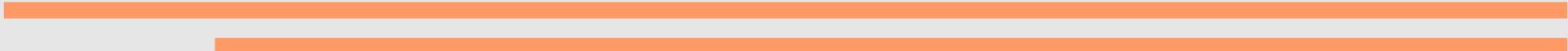
*/tracking/storeTrajectory 1* – траектории частиц сохраняются и могут быть визуализированы

***В случае если частиц очень много (например, моделируется адронный ливень), может наступить переполнение оперативной памяти.***



# ***/event***

- `/event/stack/status` – вывести подробную  
информацию о стеках треков в событии
- `/event/stack/clear` - сбросить стеки
- `/event/abort` - прекратить моделирование  
текущего события
- `/event/verbose` - диагностика
  - `/event/verbose 0` - нет сообщений
  - `/event/verbose 1` - вывод информации о стеках



# */run*

Подкаталог :

1) **/run/particle/** управление списком физических процессов

Команды :

2) **initialize** \* Инициализация ядра Geant4

3) **beamOn** \* Начало сеанса

4) **verbose** \* Уровень диагностики объекта G4RunManager.

5) **dumpRegion** \* информация об объектах G4Region

6) **dumpCouples** \* вывод таблицы соответствия “материал-порог”

7) **optimizeGeometry** \* флаг оптимизации геометрии

8) **breakAtBeginOfEvent** \* пауза в начале каждого события

9) **breakAtEndOfEvent** \* пауза в конце каждого события

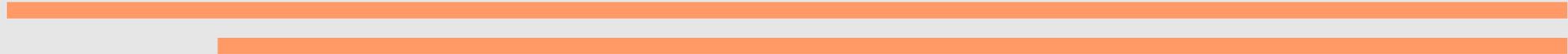
10) **abort** \* прервать текущий сеанс

11) **abortCurrentEvent** \* прервать текущее событие

12) **geometryModified** \* заново создать модель детектора

13) **physicsModified** \* пересчитать таблицы физических процессов

14) **cutoffModified** \* - не применяется



## */run (продолжение)*

15) **randomNumberStatusDirectory** \* каталог для сохранения файла с начальным значением генератора случайных чисел

16) **storeRandomNumberStatus** \* установить сохранение начального значения генератора в начале сеанса и в начале события

17) **restoreRandomNumberStatus** \* считать начальное значение генератора из файла

18) **setCut** \* установить значение порога по умолчанию

19) **setCutForRegion** \* установить значение порога для объекта G4Region



# ***/run/beamOn***

Основная команда при моделировании

*Примеры*

**/run/beamOn 1000** - Начать моделирование 1000  
событий

**/run/beamOn 1000 event.mac 100**

Начать моделирование 1000 событий, и после каждого  
из первых 100 событий выполнить сценарий event.mac

---

---

## ***/random***

**Действия с начальным значением генератора случайных чисел**

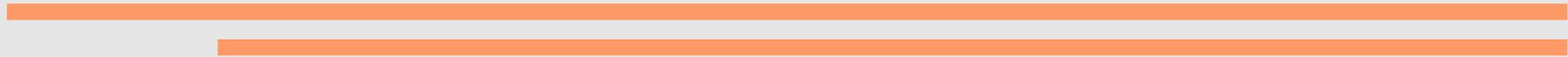
`/random/setDirectoryName [fileName]`

`/random/setSavingFlag [flag]`

`/random/saveThisRun`

`/random/saveThisEvent`

`/random/resetEngineFrom [fileName]`



# ***/material***

/material/nist/printElement

аргумент – СИМВОЛ ЭЛЕМЕНТА либо all

/material/nist/printElementZ

аргумент - Z

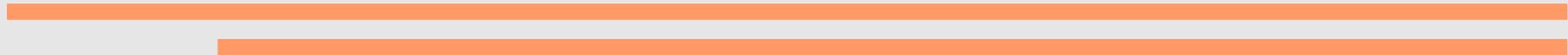
/material/nist/listMaterials

аргумент- simple, compound, hep,all

/material/g4/printElement

/material/g4/printMaterial

/material/verbose



# Пример

```
Idle> /material/nist/printElementZ 12
```

```
/material/nist/printElementZ 12
```

```
Nist Element: <Mg> Z= 12 Aeff(amu)= 24.305 18 isotopes:
```

```
    N: 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
```

```
    mass(amu): 18647.5 19572.3 20492.5 21418.9 22341.9 23274.1 24202.6
```

```
25135.7 26066.8 27002.7 27936 28873.1 29807.1 30744.5 31679.3 32618.6
```

```
33554.7 34494.3
```

```
    abanbance: 0 0 0 0 0.7899 0.1 0.1101 0 0 0 0 0 0 0 0 0 0 0
```

```
Idle> /material/nist/printElement Mg
```

```
/material/nist/printElement Mg
```

```
Nist Element: <Mg> Z= 12 Aeff(amu)= 24.305 18 isotopes:
```

```
    N: 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
```

```
    mass(amu): 18647.5 19572.3 20492.5 21418.9 22341.9 23274.1 24202.6
```

```
25135.7 26066.8 27002.7 27936 28873.1 29807.1 30744.5 31679.3 32618.6
```

```
33554.7 34494.3
```

```
    abanbance: 0 0 0 0 0.7899 0.1 0.1101 0 0 0 0 0 0 0 0 0 0 0
```

# Еще пример

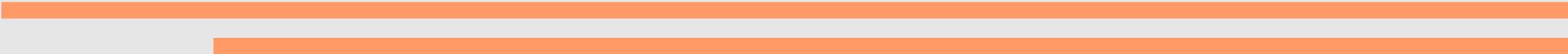
```
Idle> /material/nist/listMaterials hep
```

```
/material/nist/listMaterials hep
```

```
=====
```

###	HEP & Nuclear Materials		##
Ncomp	Name	ChFormula	density(g/cm^3) I(eV)
1	G4_IH2	0.0708	21.8
1	G4_IN2	0.807	82
1	G4_IO2	1.141	95
1	G4_IAr	1.396	188
1	G4_IKr	2.418	352
1	G4_IXe	2.953	482
3	G4_PbWO4	8.28	0
	8	0.140637	
	82	0.455366	
	74	0.403998	
1	G4_Galactic	1e-25	21.8

```
=====
```



# *Самые главные команды*

Idle> **help**

Command directory path : /

Sub-directories :

1) /control/ UI control commands.

2) /units/ Available units.

.....

Type the number ( 0:end, -n:n level back ) :

0

Exit from HELP.

Idle> **exit**

---

---

# Как добавить свои команды (I)

Создается класс — Messenger:

```
class ExN03DetectorMessenger: public G4UImessenger
{
public:
    ExN03DetectorMessenger(ExN03DetectorConstruction* );
    ~ExN03DetectorMessenger();
    void SetNewValue(G4UIcommand*, G4String);

private:
    ExN03DetectorConstruction* ExN03Detector;
    G4UIDirectory*              N03Dir;
    G4UIDirectory*              detDir;
    G4UIcmdWithAString*         AbsMaterCmd;
    G4UIcmdWithADoubleAndUnit* AbsThickCmd;
    G4UIcmdWithAnInteger*      NbLayersCmd;
    G4UIcmdWithADoubleAndUnit* MagFieldCmd;
    G4UIcmdWithoutParameter*   UpdateCmd;
    ...
};
```

---

---

# Как добавить свои команды (II)

Создаются объекты - «команды»:

```
ExN03DetectorMessenger::ExN03DetectorMessenger( ExN03DetectorConstruction*  
ExN03Det):ExN03Detector(ExN03Det)
```

```
{  
  N03Dir = new G4UIdirectory("/N03/");  
  N03Dir->SetGuidance("UI commands of this example");
```

```
  detDir = new G4UIdirectory("/N03/det/");  
  detDir->SetGuidance("detector control");
```

```
  NbLayersCmd = new G4UIcmdWithAnInteger("/N03/det/setNbOfLayers",this);  
  NbLayersCmd->SetGuidance("Set number of layers.");  
  NbLayersCmd->SetParameterName("NbLayers",false);  
  NbLayersCmd->SetRange("NbLayers>0 && NbLayers<500");  
  NbLayersCmd->AvailableForStates(G4State_PreInit,G4State_Idle);
```

```
  ...  
}
```

---

---

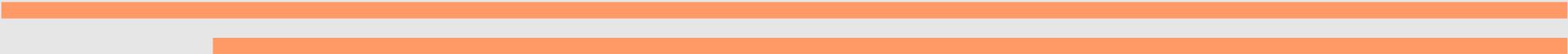
# Как добавить свои команды (III)

Обработка команд:

```
void ExN03DetectorMessenger::SetNewValue(G4UIcommand* command,G4String
newValue)
{
if( command == NbLayersCmd )
    { ExN03Detector->SetNbOfLayers(NbLayersCmd->GetNewIntValue(newValue));}

if( command == UpdateCmd )
    { ExN03Detector->UpdateGeometry(); }

if( command == MagFieldCmd )
    { ExN03Detector->SetMagField(MagFieldCmd->GetNewDoubleValue(newValue));}
...
}
```



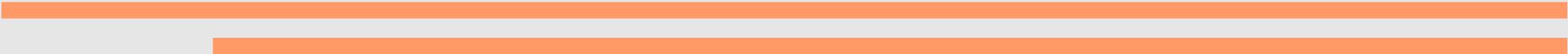
# Как добавить свои команды (IV)

```
Idle> /N03/det/update  
/N03/det/update
```

-----  
---> The calorimeter is 10 layers of: [ 10mm of Lead + 5mm of liquidArgon ]  
-----

Obsolete world logical volume is removed from the default region.  
World is registered to the default region.

...



# ***Взаимодействие пользователя с программой моделирования***

- **Пакетный режим**
  - **Пакетный режим, управляемый сценарием**
  - **Интерактивный режим с командной строкой**
  - **Интерактивный режим с графическим интерфейсом**
- 
-

```
int main()
{
// Construct the default run manager
G4RunManager* runManager = new G4RunManager;

// set mandatory initialization classes
runManager->SetUserInitialization(new ExN01DetectorConstruction);
runManager->SetUserInitialization(new ExN01PhysicsList);

// set mandatory user action class
runManager->SetUserAction(new ExN01PrimaryGeneratorAction);

// Initialize G4 kernel
runManager->Initialize();

// start a run
int numberOfEvent = 1000;
runManager->BeamOn(numberOfEvent);

// job termination
delete runManager;
return 0;
}
```

# ***Взаимодействие пользователя с программой моделирования***

- **Пакетный режим**
  - **Пакетный режим, управляемый сценарием**
  - **Интерактивный режим с командной строкой**
  - **Интерактивный режим с графическим интерфейсом**
- 
-

```
int main(int argc, char** argv) {
    // Construct the default run manager
    G4RunManager * runManager = new G4RunManager;

    // set mandatory initialization classes
    runManager->SetUserInitialization(new MyDetectorConstruction);
    runManager->SetUserInitialization(new MyPhysicsList);

    // set mandatory user action class
    runManager->SetUserAction(new MyPrimaryGeneratorAction);

    // Initialize G4 kernel
    runManager->Initialize();

    //read a macro file of commands
    G4UImanager * UI = G4UImanager::getUIpointer();
    G4String command = "/control/execute ";
    G4String fileName = argv[1];
    UI->applyCommand(command+fileName);

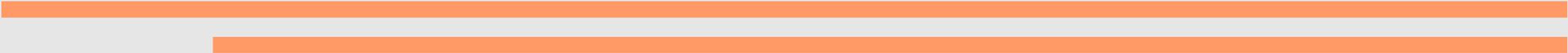
    delete runManager;
    return 0;
}
```

---

---

# Пример файла сценария

```
#  
# Macro file for "myProgram.cc"  
#  
# set verbose level for this run  
#  
/run/verbose 2  
/event/verbose 0  
/tracking/verbose 1  
#  
# Set the initial kinematic and run 100 events  
# electron 1 GeV to the direction (1.,0.,0.)  
#  
/gun/particle e-  
/gun/energy 1 GeV  
/run/beamOn 100
```



# ***Взаимодействие пользователя с программой моделирования***

- **Пакетный режим**
  - **Пакетный режим, управляемый сценарием**
  - **Интерактивный режим с командной строкой**
  - **Интерактивный режим с графическим интерфейсом**
- 
-

```
int main() {
    // Construct the default run manager
    G4RunManager * runManager = new G4RunManager;

    // set mandatory initialization classes
    runManager->SetUserInitialization(new MyDetectorConstruction);
    runManager->SetUserInitialization(new MyPhysicsList);

    // set user action classes
    runManager->SetUserAction(new MyPrimaryGeneratorAction);

    // Initialize G4 kernel
    runManager->Initialize();

    // Define UI terminal for interactive mode
    G4UIsession * session = new G4UIterminal;
    session->SessionStart();
    delete session;

    // job termination
    delete runManager;
    return 0;
}
```

---

---

jemtchou: ~/geant4/bin/Linux-g++\$ ./prog01

\*\*\*\*\*

Geant4 version Name: geant4-08-00-patch-01 (10-February-2006)

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

\*\*\*\*\*

Idle>

Idle> /tracking/verbose 1

Idle> /control/execute run01.mac



# ***Доступные интерактивные сессии***

## **G4UItterminal и G4UItcsh**

командная строка (простой терминал и эмулятор tcsh)

## **G4UIXm, G4UIXaw и G4UIWin32**

графический интерфейс с использованием модулей Motif, Athena или Win32

## **G4UIGAG и G4UIGainServer**

Geant4 Adaptive GUI – графический интерфейс написанный на Java

*<http://erpc1.naruto-u.ac.jp/~geant4/>*

**G4Qt** – экспериментальная разработка

---

---

# ***Включение в программу моделирования***

```
#include "G4UIxxx.hh"
```

```
...
```

```
G4UIsession* session = new G4UIxxx;  
session->SessionStart();  
delete session;
```

***В случае G4UITcsh***

```
G4UIsession* session = new G4UITerminal(new G4UITcsh);
```

---

---

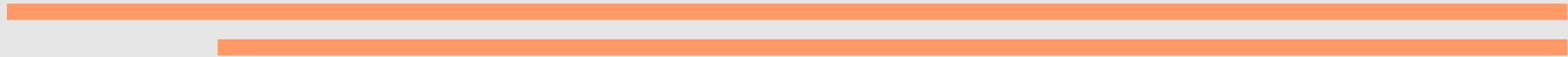
# ***Задание графического интерфейса – команды /gui***

`/gui/addMenu run Run`

`/gui/addButton run Init /run/initialize`

`/gui/addButton run "Set gun" "/control/execute gun.g4m"`

`/gui/addButton run "Run one event" "/run/beamOn 1"`



File Run Gun Vis Viewer

You have successfully chosen to use the following graphics systems.

Current available graphics systems are:

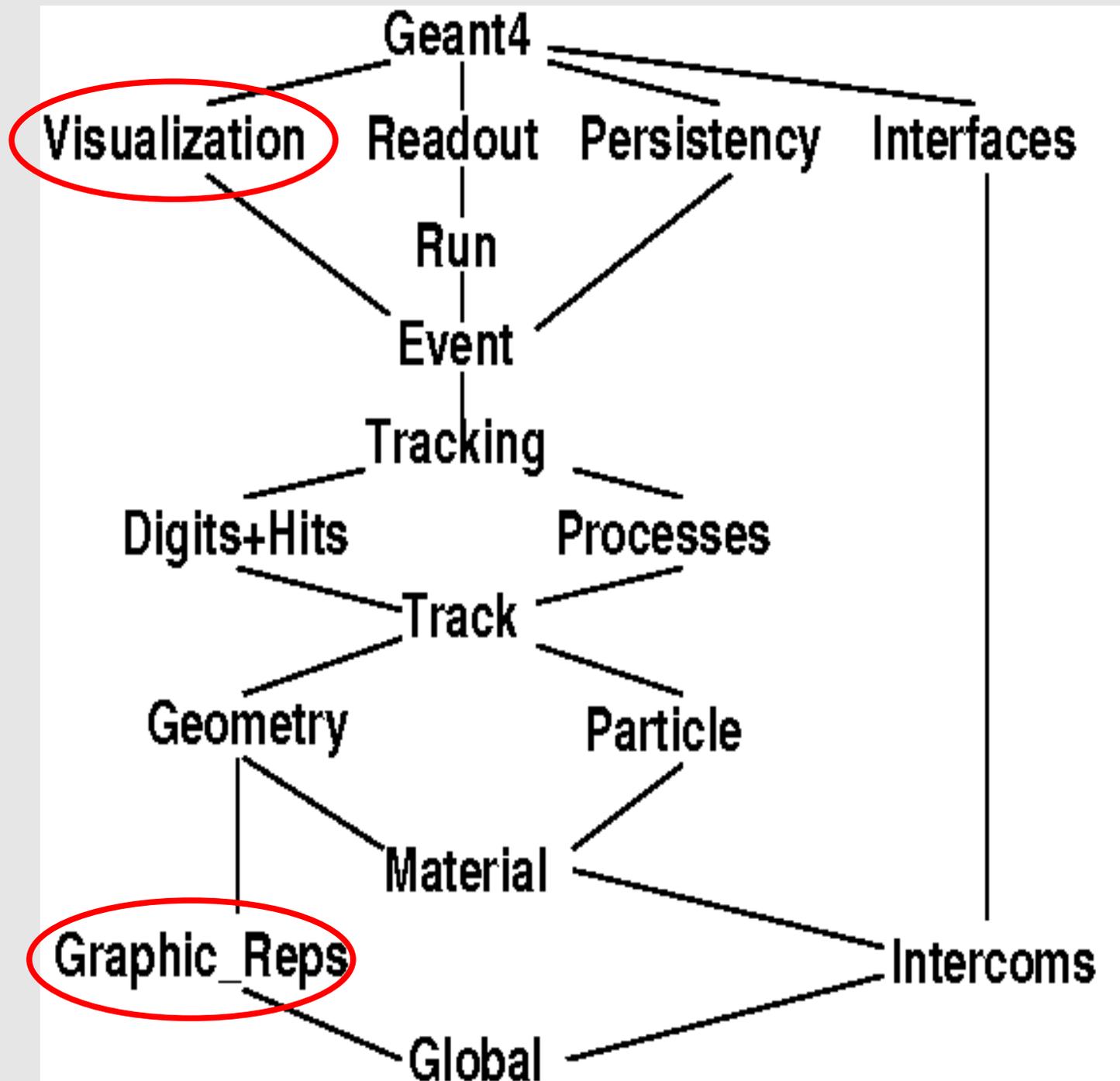
- ASCIITree (ATree)
- DAWNFILE (DAWNFILE)
- GAGTree (GAGTree)
- G4HepRep (HepRepXML)
- G4HepRepFile (HepRepFile)
- RayTracer (RayTracer)
- VRML1FILE (VRML1FILE)
- VRML2FILE (VRML2FILE)
- FukuiRenderer (DAWN)
- OpenGLImmediateX (OGLIX)
- OpenGLStoredX (OGLSX)
- OpenGLImmediateXm (OGLIXm)
- OpenGLStoredXm (OGLSXm)
- RayTracerX (RayTracerX)
- VRML1 (VRML1)

Clear

session

# ***Визуализация детектора и событий***





# ***Что можно визуализировать***

## ***Элементы модели***

Объемы детектора или отдельных подсистем

Траектории частиц

Срабатывания в чувствительных объемах

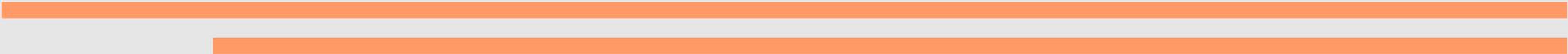
## ***Кроме того, по желанию пользователя***

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*

HerRep/WIRED

DAWN

VRML

gMocrenFile

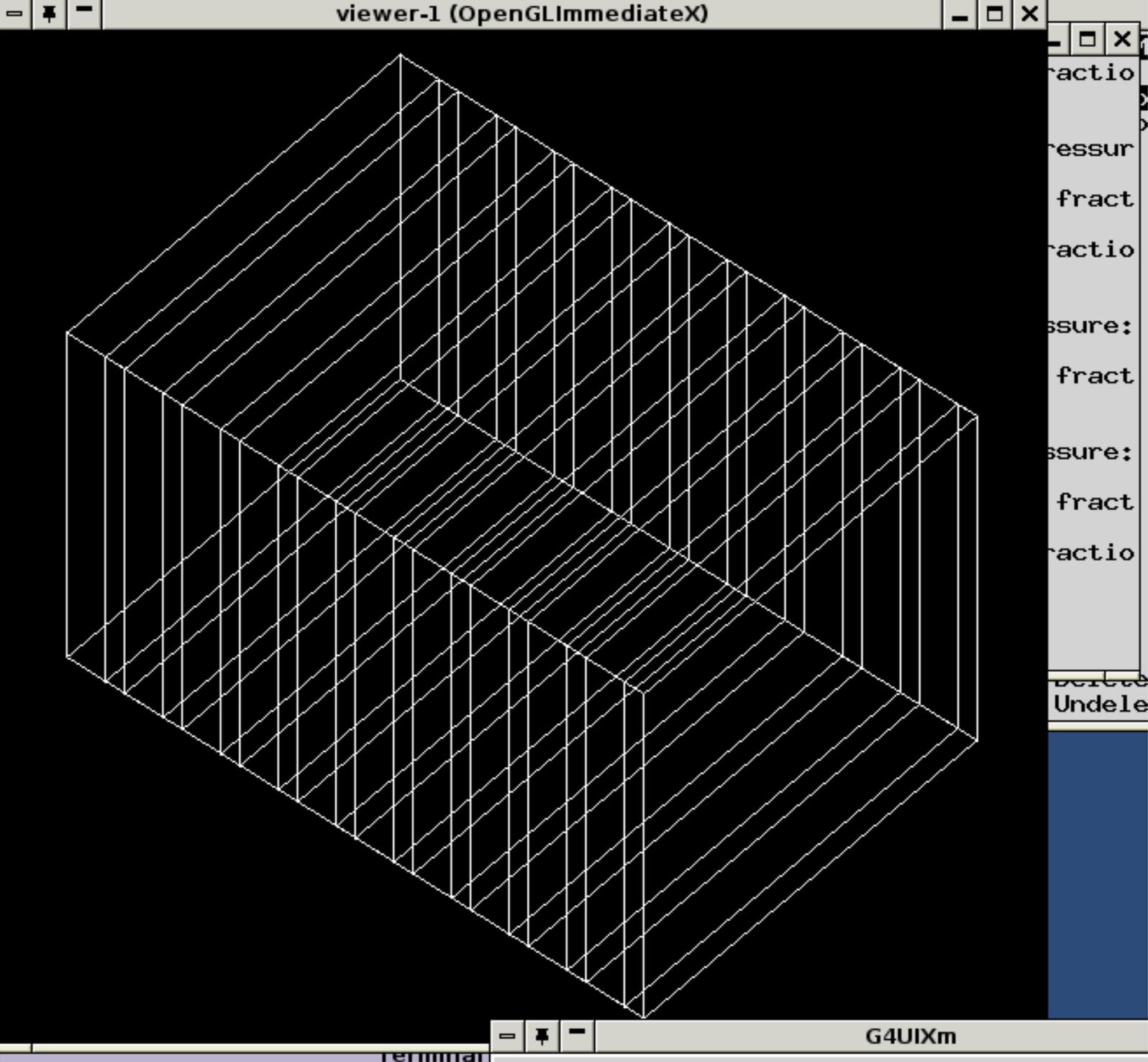
*Только геометрия детектора*

RayTracer

AsciiTree

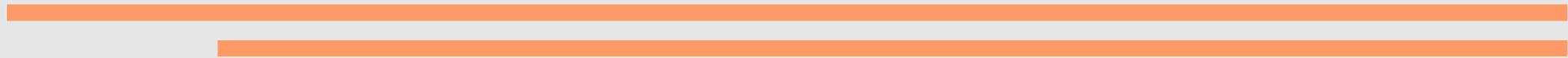
---

---



# ***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
HerRep	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/>*

---

---

Сборка драйверов и их включение в программу моделирования зависит от переменных окружения:

Сборка драйверов в пакете Geant4

```
G4VIS_BUILD_OPENGLX_DRIVER # OpenGL-Xlib driver
G4VIS_BUILD_OPENGLXM_DRIVER # OpenGL-Motif driver
G4VIS_BUILD_OIX_DRIVER      # OpenInventor-Xlib driver
G4VIS_BUILD_RAYTRACERX_DRIVER # RayTracer-XLib
G4VIS_BUILD_DAWN_DRIVER     # DAWN-Network
G4VIS_BUILD_VRML_DRIVER     # VRML-Network
G4VIS_BUILD_OPENGLQT_DRIVER # OpenGL-Qt driver
```

Включение в программу моделирования

```
G4VIS_USE_OPENGLX
G4VIS_USE_OPENGLQT
G4VIS_USE_OPENGLXM
G4VIS_USE_OIX
G4VIS_USE_RAYTRACERX
G4VIS_USE_DAWN
G4VIS_USE_VRML
```

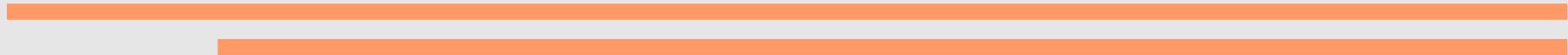
---

---

# Создание объекта *VisManager* в программе моделирования

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

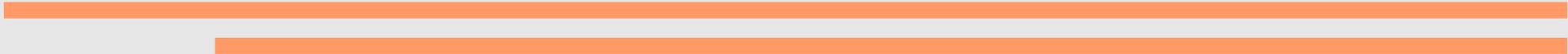
```
.....  
// Your Visualization Manager  
#include "G4VisExecutive.hh"  
.....  
  
// Instantiation and initialization of the Visualization Manager  
#ifndef G4VIS_USE  
G4VisManager* visManager = new G4VisExecutive;  
visManager -> Initialize ();  
#endif  
  
.....  
#ifndef G4VIS_USE  
delete visManager;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

---

---

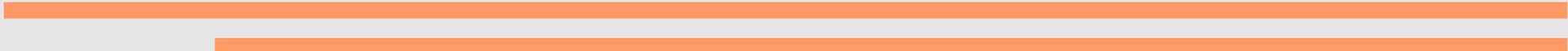
# ***Взаимодействие визуализируемых объектов и графической системы***

**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 OGLIX
```

```
/vis/scene/create
```

```
/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/viewer/flush
```

```
/vis/viewer/set/viewpointThetaPhi 20 20
```

```
/vis/viewer/refresh
```

