Visualization Status/Development

Satoshi Tanaka
Ritsumeikan University (Rits)

Geant4 Workshop at CERN
October 2, 2002

Satoshi moved to Kyoto recently, and lives near this tower.
Overview

- Vis category is working stably.
- Some works are in progress.
  - Curved trajectory
  - Picking
  - DTREE,
  - DCUT
  - Network visualization
  - Etc
Visualization of curved trajectories
Visualization of curved trajectories (1)

- At present, we **cannot visualize smoothly curved trajectories**. Why?
- Because **visualization drivers have no access to auxiliary points**. These points are not persisted long enough for visualization to use them, hence we can’t draw smooth trajectories.
Visualization of curved trajectories (2): Solution

- Introduce a virtual method
  ```cpp
  virtual const
  G4std::vector<G4ThreeVector*>* GetAuxiliaryPoints() const { return 0; }
  ```
to `G4VTrajectoryPoint` class.

- We will introduce `G4SmoothTrajectory` and `G4SmoothTrajectoryPoint` classes derived from `G4VTrajectory` and `G4VTrajectoryPoint` abstract base classes, respectively.

- A new `DrawTrajectory()` method should be implemented to use the auxiliary point data properly.

- The draft codes of `G4VTrajectory` and `G4VTrajectoryPoint` have been committed to the head of the tracking category.
Visualization of curved trajectories (3):

Note

- We should not lose performance of Geant4 by visualizing fancy smoothly curved trajectories.
  - Auxiliary points should be generated only when user requires. (There should be a switch.)
Attribute displaying of trajectories with picking
Attribute displaying of trajectories with picking (1) Problem

- Visualization drivers have access to detailed information about geometry
  - Easy to perform the attribute displaying with picking.

- Visualization drivers, however, have no access to detailed information about trajectories (or sensitive detector hits).
  - We cannot execute the attribute displaying with picking.

- Let me present some illustration figures (created by J. Perl).
Although it is easy to implement picking on detector volumes to give detailed attributes ...
Picking on trajectory can give no details
Picking on Trajectory becomes useful, if visualization drivers had better access to trajectory information.
Attribute displaying of trajectories with picking (2) Solution

- We introduce attribute classes in `graphics_reps` category.
- `G4AttDef` defines new kinds of attributes that can then have values set for a Trajectory, Trajectory Point or Sensitive Detector Hit.
- `G4AttValues` can be attached to a Trajectory, Trajectory Point or Sensitive Detector Hit. These attributes are then made available to the end user in interactive graphics system (such as WIRED, OPACS).
- `G4AttValueList*` is returned by `G4VTrajectory` and `G4VtrajectoryPoint` by `GetAttValueList()` method.
  
  ```cpp
  virtual const G4AttValueList* GetAttValueList() const
  { return 0; }
  ```

- The draft codes are committed to the head of the tracking category.
Attribute displaying of trajectories with picking (3) Action

- Classes G4AttDef, G4AttValues, G4AttValueList will be implemented
  - The draft codes already exists at the head of the graphics_reps category
  - Work will be made as collaboration of John Allison, Joseph Perl, Jacek Generowicz, Makoto Asai, and Satoshi Tanaka.
- To be tested with HepRep driver (Joseph)
DTREE
DTREE (1): Overview

**DTREE:**
- ASCII Tree (indented) is implemented
  - Properly indented description of physical volumes.
- GAG Tree is implemented.
- HepRep visualization driver realizes DTREE synchronized to visualization (NEW!)
DTREE (2): GAG Tree
DTREE (3): HepRep driver
DTREE (3): HepRep driver
DCUT
(Slice view)
DCUT (slice view) (1)

- **Status**
  - DCUT is available with OpenGL driver.
  - But it does not re-fill section plane.
    = **Only outlines** of a section plane are drawn.

- **Projects are going on to improve the situation**
    (Evgeni, John Allison)
  - Another solution with DAWNCUT application.
    (Satoshi Tanaka)
DCUT (slice view) (2)

- DCUT with DAWNFILE driver is in plan.
  - We use DAWNCUT application (cut-away view generator)
  - It can generate fine view of section plane by filling proper polygons:
New network visualization
New network visualization

- A new project with Naruto group (Yoshida, Minamimoto)

- GAIN (Network version of GAG)
  + G4 remote visualization
    - VRML: Already implemented
    - DAWN: In plan

- More details are presented in Yoshida’s talk.
Note added
Note added (1):
Precision control in the DAWNFILE visualization driver

- The default precision is made “9”, which had been “6” for a long time.

- The precision is controllable by setting an environmental variable:
  - Example:
    ```
    setenv G4DAWNFILE_PRECISION 12
    ```

- For details see:
  ```
  geant4/source/visualization/Fukui_Renderer/CUSTOMIZE
  ```
Note added (2):
Information for users

- Visualization category uses many graphics packages, which are not included in the distributed Geant4 package.
- More user-friendly way to obtain the packages?
  - Web, new CVS category, ....