GeoModelExplorer: improving interactive 3D geometry visualization tool based on Qt3D

Mentors: Riccardo Maria Bianchi, Edward Moyse, Joseph Boudreau
Organization: CERN/HSF

1. Student Information

Background:

Name: Huajian Qiu
University: Swiss Federal Institute of Technology in Lausanne(EPFL)
Major: Master student in Computational Science and Engineering

Open source experience:
Favo

- Initiative in a projected-based course: Software Development Project
- Connect users in their area through asking & receiving favours for common daily scenarios
- Six contributors work in weekly sprint: https://github.com/favo-sdp

Contact Information:

Email: huajian1069@gmail.com
Phone: +41 078 323 99 88
Github: https://github.com/huajian1069

About me

My name is Huajian, I have been a developer with a focus on scientific computing and visualisation for years. In the past, I usually spent time on projects inside the campus and this is my first time taking a stab at approaching the CERN/HSF community. I've enjoyed building custom visualisation tools to collect insights behind complex processes, with a big picture in mind.

I speak C as my mother language, but I also have rich experience in other programming languages such as C++, Python, and Java.

I have worked in many projects in digital 3D geometry processing, virtual reality, applied data analysis, and other courses. I know my way in graphics source code. One of my relevant 3D graphics experiences is extending the Projective Dynamics and ShapeUp Framework, which are built on Nanogui, OpenGL, Eigen, OpenMP. I added a bone constraint feature during my course project[1,2]

[1]. Code repository: https://github.com/huajian1069/digitalGeometryProcessing.git
[2]. Demo slides: Spider Skeleton

Availability

I will be based in Lausanne, Switzerland during the summer. Therefore, I will be working in the GMT+2 time zone. And I will be available **full time during summer.** Since our summer vacation is from June to September, I believe that I have enough time to complete the project.

2. Project idea

Synopsis

Interactive visualization can provide intuitive insights for experiment setups and results in many domains, including High Energy Physics. Therefore, Virtual Point1 (VP1) was developed by ATLAS for 3D event display. Nevertheless, it seems a pity that it is based on an old graphics engine(OpenInventor) and resides in a big project(full ATLAS experiment software stack) with many dependencies.

To get free of these limitations, GeoModelExplorer, a super-lighter version of VP1 has been developed, with a strong focus on the visualization of geometry data. Some of the important next steps are to replace the old 3D engine, improve existing functions and add more features. All of them are waiting for development.

In this project, I will take these steps one by one, integrate the modern graphics engine Qt3D, improve the geometry rendering effects, extend the mesh I/O abilities, and add smooth animation for view inspections. There will be a large amount of design and engineering work but I will take extreme care. This project will eventually make contributions as part of the current software modernization effort in ATLAS.

Benefits

Easier Integration with Qt GUI framework

The graphics engine Qt3D will be natively integrated with the Qt GUI framework. OpenInventor/Coin and SoQt will no longer be needed, therefore making dependencies further reduced, concentrated, and easy to manage.

General applicability with experiment-independent character

The new Qt3D-based prototype for GMEX will be compiled and run standalone, without the need for the full ATLAS software framework. It will be developed with the possibility to extend to new scenarios in mind, such as different types of new-added detectors, or even not only detectors.

Better geometry handling capabilities

Special geometries like Boolean volumes, polycones, and tessellated solids have their own nice properties. Special handling of these objects will improve the effect of common operations on them like picking, rendering, and other predefined operations. Interactive picking will be supported for all volumes.

Better rendering and visibility setting

Rendering of 3D volumes for detector geometry will be more distinguishable and free from bugs. The volumes at different levels will be organized in a hierarchy with clear parent and child relationships and uniform class type(i.e., composite design pattern). It will simplify the setting of visibility and other properties for detector volumes.

Better view inspection and transition experience

Abrupt transitions between views of different volumes will be prevented; instead, a smooth animation will be accompanied.

Ability to import and export to more 3D formats *

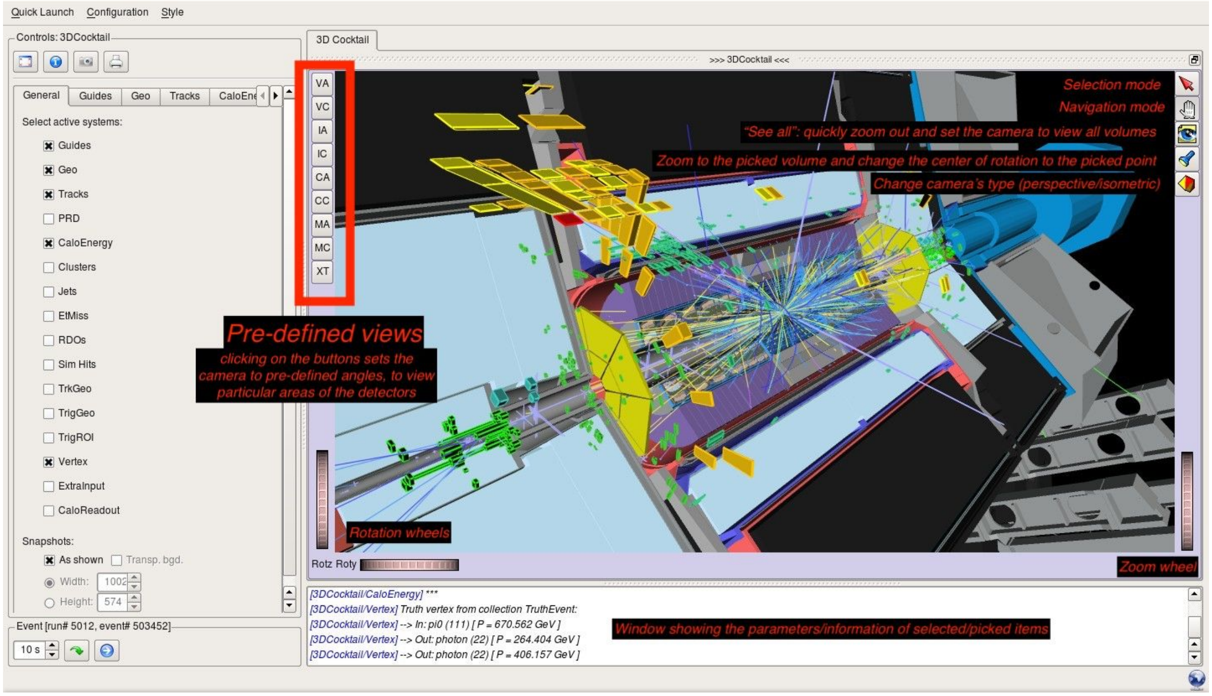
The support of more formats will ease the cooperation between different graphics software. 3D-converter is a potential library to handle this. (*This is an additional benefit if time allows further development)

Deliverables

1. Minimum Viable Product: a standalone "examiner viewer"

- It will allow users to flexibly select and inspect the geometry volumes of the detector. It will include a viewport of the 3D scene, an information window/console to show the properties of the selected items, a toolbar to select predefined views and another toolbar to change perspective mode and cursor mode between selection and navigation.

- Besides those, I will also design the mapping between other operations, such as rotate, translate, scale, and keyboard/mouse events. A survey of user habits of other popular graphics software will be beneficial before design.
- Users will also have buttons to set visibility of different levels, parts of volumes for the detector.
- The UI should allow users to snapshot and bookmark a particular view so that they can revisit it quickly.



Controls: 3DCocktail

Select active systems:

☒ Guides
☒ Geo
☒ Tracks
□ PRD
☒ CaloEnergy
□ Clusters
□ Jets
□ EtMiss
□ RDOs
□ Sim Hits
□ TrkGeo
□ TrigGeo
□ TrigROI
☒ Vertex
□ ExtraInput
□ CaloReadout

Snapshots:

☒ As shown □ Transp. bgd.
● Width: 1002
○ Height: 574

Event [run# 5012, event# 503452]

10 s

```
[3DCocktail/CaloEnergy] ***
[3DCocktail/Vertex] Truth vertex from collection TruthEvent:
[3DCocktail/Vertex] --> In: pi0 (111) [ P = 670.562 GeV ]
[3DCocktail/Vertex] --> Out: photon (22) [ P = 264.404 GeV ]
[3DCocktail/Vertex] --> Out: photon (22) [ P = 406.157 GeV ]
```

Selection mode

Navigation mode

"See all": quickly zoom out and set the camera to view all volumes

Zoom to the picked volume and change the center of rotation to the picked point

Change camera's type (perspective/isometric)

Pre-defined views

clicking on the buttons sets the camera to pre-defined angles, to view particular areas of the detectors

Rotation wheels

Zoom wheel

Window showing the parameters/information of selected/picked items

Figure 1 - VP1 GUI

2. Feature: support of interactive picking geometry volumes

The picking method supported in Qt3D is based on ray-casting. Based on previous warm-up exercises, this can be implemented by adding a QtObjectPicker object to the geometry volume entity. The key to success is to choose a suitable picking method, which is specified by QPickingSettings globally, and the bounding volume of geometry, which is specified by boundingVolumePositionAttribute.

3. Feature: novel views inspection and animation effect

- Smooth transitions between views of different parts of the detector geometry
- For volumes with complex shapes and those that are more interesting to users, a set of control orbits of the camera, for rotations, zoom, dolly, and other predefined movements, will be carefully designed.
- Parts of detector volumes will have a clickable button to trigger animations.

- For a set of volumes, if they distribute very flat around a plane or narrow and long around a line(i.e., a reduced dimension space), it will appear intelligent to automatically place the camera to inspect the maximum-spanned plane or line. This can be conducted by principal components analysis(PCA) of mesh vertices. It is a quick operation when mesh vertex numbers are not extremely high.

Potential additional tasks

If the time allows for further development, I will add the following tasks:

Optimisation: improve rendering of 3D volumes for detector geometry
Special attention will be given to the correct rendering of imported meshes into the new Qt3D-based version of GMEX.

Feature: built-in 3D mesh format I/O converter
Users will be able to import the common 3D file formats OBJ, 3DS, MA, MB, XSI, LWO, DXF, STL, MAT, DAE and export the file to any other format. This will be implemented by integrating an external library, such as 3D-converter.

Related work

Virtual Point1 (VP1)

VP1 is an interactive 3D event display for the ATLAS experiment at CERN. VP1 provides useful tools for the understanding of the physics hidden inside proton-proton collision events. My project aims at continuing the effort to provide similar geometry visualization functions as VP1, but much easier to use, and by the use of modern graphics libraries.

GeoModelExplorer

This project is the context of my development in this GSoC project. It is a super-lightweight version of VP1(Light), without event display. It is still under very active development. I will be involved in the development team of this project if selected into GSoC.

Project schedule

May 4 - June 1

1. Familiarize myself with the code base of GeoModelExplorer
2. Read documents and do experiments with Qt3D
3. Survey needed functionality and look for the existence of viable equivalents in Qt3D.

Actual start of the Work Period
June 1 - June 29:

1. Build a Qt3D testbed
2. Handle detector volumes meshes, build suitable entity representation in class
3. Implement wish list of MVP: examiner viewer, design UI

June 29 - July 3: First evaluation

- MVP: examiner viewer should be ready

July 3 - July 27:

1. Develop interactive picking of 3D geometry volumes
2. Check and Improve the rendering of the geometry volumes

July 27 - July 31: Second evaluation

- Rendering should be free from bugs in edge case and satisfactory
- Volumes picking should be ready

July 31 - August 24:

1. Implement wish list of novel views inspection and animation effect
2. Integrate 3D mesh format I/O converter, add UI
3. Write documentation

August 24 - August 31: Submit code and final evaluations

- Users should be able to inspect geometry volumes efficiently
- views with animation and smooth transitions.
- 3D data should be able to be exported to common formats.
- Complete user’s documentation