Hall A event visualization in Python

Tyler Kutz
Stony Brook University

June 26, 2018
Hall A/C software workshop

Contents

1. Introduction

2. Tools

3. Visualization code

Goal

Need a program to visualize events and detector response in the Hall A HRS

This requires tools to:

- Build and visualize detector geometry (VPython)
- Read event data from ROOT files (PyROOT)
- Visualize event and detector response (VPython)

VPython

VPython is the Python language plus the Visual 3D graphics module

Basics:

- Construct objects from set of simple volumes (`sphere, box, cylinder...`)
- More complex objects possible using `extrusion`
- Each object has a set of...
  - ...geometric attributes (`pos, length, radius...`)
  - ...optical attributes (`color, opacity, material...`)

```
# create a red cube at origin with unit side length
from vpython import *
myBox = box(pos=vector(0,0,0), length=1, width=1, height=1)
myBox.color = color.red
```

Reading event data with PyROOT

Normal (C++) method for reading data from a ROOT tree `T`:

```
double var;
T->SetBranchAddress("branch", &var);
T->GetEntry(i);
```

Python doesn't have pointers! Python method is:

```
T.GetEntry(i)
var = T.branch
```

But...our branch names have periods in them (e.g., `R.tr.n`).
Python interprets this as the dot operator!

Cleanest solution I found:

```
T.GetEntry(i)
ntr = getattr(T, "R.tr.n")
```

Alternate suggestions are welcome!

Structure of visualization code

Each detector is built its own file `detector.py`:

- Contains single function `build_detector` that constructs geometry
- Function must return object/array of objects composing that detector

Main function `event_vis.py`:

- Draws coordinate system axes
- Sets scene lighting, camera focus and initial angle
- Imports and executes all detector build functions
- Opens ROOT file (PyROOT) and loops over events
  - Draws electron track
  - Adjusts detector appearance to reflect detector response

Demonstration

- Current visualization contains the RHRS preshower/shower calorimeters
- VDC is shown as visual reference but does not respond to event data
- Tracks are shown as green lines
- Calorimeter blocks in shower cluster are colored red
    - Saturation adjusted based on ADC signal

Summary

- VPython is a simple tool allowing 3D graphics visualization
- PyROOT allows Python programs to directly access data from ROOT files
- Combining these tools allows visualization of detector geometry and event data
- Future improvements:
  - Define complete detector geometry
  - Implement more quantitative visualization of detector response
  - Create GUI for cycling events, displaying event parameters, etc.

Currently tracked in the `replay/scripts` directory of the tritium repository:
github.com/JeffersonLab/HallA-Online-Tritium