

基本思路介绍：

1 在 Geant4 的 SteppingAction 中进行数据抽取；

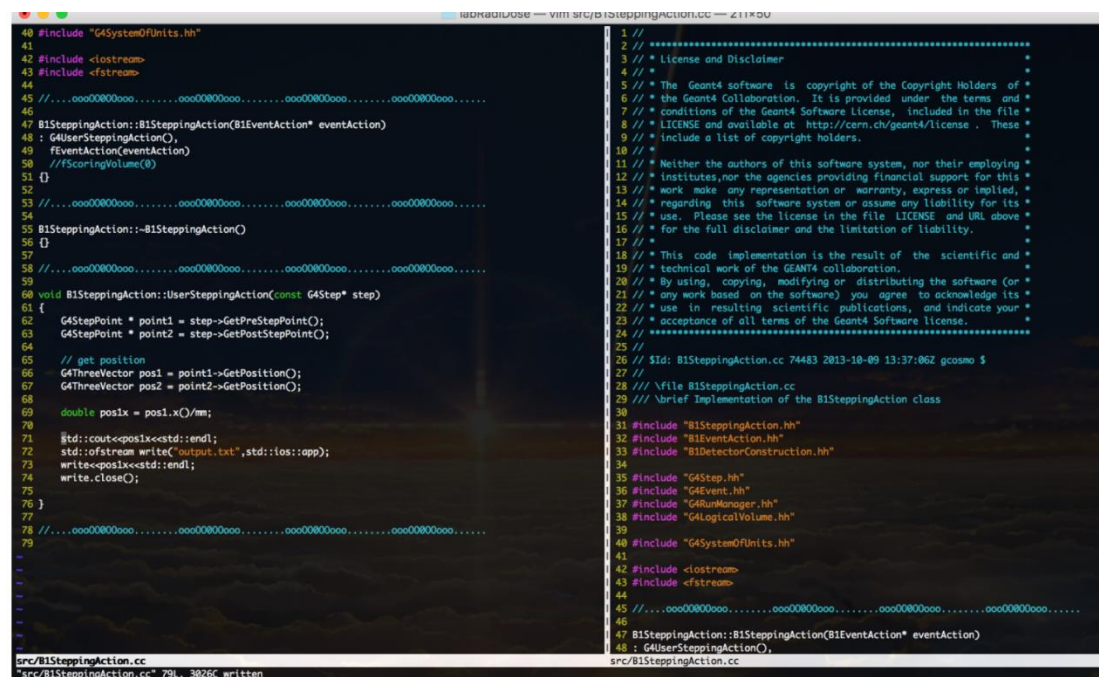
2 在数据抽取完成时，使用 C++ 的 ofstream 打开文本并进行文本写入，待本 step 中所有抽取的数据都写入之后执行关闭文件；

也就是说，每个 Step 都进行数据写入。

需要注意：

1 ofstream 的默认写入方式是清空文本再写入数据，而这里是要在每个 step 中反复进行文本的打开、写入和关闭，因此需要使用 ofstream 的不清空原有数据的方式，如"ios::app"

2 每次运行 Geant4 之前，需要手动删除文本文件，否则上一次运行 Geant4 的数据依然会存在。（有兴趣的同学可以考虑如何智能化地解决这个问题，可以考虑在 RunAction 的 Begin 函数中执行文件清空哦，也就是使用清空原有数据的方式打开文本）



```
40 #include "G4SystemOfUnits.hh"
41
42 #include <ostream>
43 #include <fstream>
44
45 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
46
47 B1SteppingAction::B1SteppingAction(B1EventAction* eventAction)
48 : G4UserSteppingAction(),
49   fEventAction(eventAction)
50 //fScoringVolume(0)
51 {}
52
53 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
54
55 B1SteppingAction::~B1SteppingAction()
56 {}
57
58 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
59
60 void B1SteppingAction::UserSteppingAction(const G4Step* step)
61 {
62     G4StepPoint * point1 = step->GetPreStepPoint();
63     G4StepPoint * point2 = step->GetPostStepPoint();
64
65     // get position
66     G4ThreeVector pos1 = point1->GetPosition();
67     G4ThreeVector pos2 = point2->GetPosition();
68
69     double pos1x = pos1.x()/mm;
70
71     std::cout<<pos1x<<std::endl;
72     std::ofstream write("output.txt",std::ios::app);
73     write<<pos1x<<std::endl;
74     write.close();
75 }
76
77
78 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
79
src/B1SteppingAction.cc
src/B1SteppingAction.cc" 79L, 3026C written

2 // *****
3 // * License and Disclaimer
4 // *
5 // * The Geant4 software is copyright of the Copyright Holders of *
6 // * the Geant4 Collaboration. It is provided under the terms and *
7 // * conditions of the Geant4 Software license, included in the file *
8 // * LICENSE and available at http://cern.ch/geant4/license . These *
9 // * include a list of copyright holders.
10 // *
11 // * Neither the authors of this software system, nor their employing *
12 // * institutes,nor the agencies providing financial support for this *
13 // * work make any representation or warranty, express or implied, *
14 // * regarding this software system or assume any liability for its *
15 // * use. Please see the license in the file LICENSE and URL above *
16 // * for the full disclaimer and the limitation of liability.
17 // *
18 // * This code implementation is the result of the scientific and *
19 // * technical work of the GEANT4 collaboration.
20 // * By using, copying, modifying or distributing the software (or *
21 // * any work based on the software) you agree to acknowledge its *
22 // * use in resulting scientific publications, and indicate your *
23 // * acceptance of all terms of the Geant4 Software license.
24 // *****
25 //
26 // $Id: B1SteppingAction.cc 74483 2013-10-09 13:37:06Z gcosmo $
27 //
28 /// \file B1SteppingAction.cc
29 /// \brief Implementation of the B1SteppingAction class
30
31 #include "B1SteppingAction.hh"
32 #include "B1EventAction.hh"
33 #include "B1DetectorConstruction.hh"
34
35 #include "G4Step.hh"
36 #include "G4Event.hh"
37 #include "G4RunManager.hh"
38 #include "G4LogicalVolume.hh"
39
40 #include "G4SystemOfUnits.hh"
41
42 #include <ostream>
43 #include <fstream>
44
45 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
46
47 B1SteppingAction::B1SteppingAction(B1EventAction* eventAction)
48 : G4UserSteppingAction(),
```

基本思路：

1 利用 C++ 的 ofstream 保存文本

2 进入当前 step，准备好要存储的数据

3 打开文本，但并不删除文本原有内容，将准备好的数据以追加的方式写入文本

4 关闭文本

5 进入下一轮 step

代码:

只需要在 SteppingAction 中添加代码

```
1 // *****
2 // * License and Disclaimer *
3 // *
4 // *
5 // * The Geant4 software is copyright of the Copyright Holders of *
6 // * the Geant4 Collaboration. It is provided under the terms and *
7 // * conditions of the Geant4 Software license, included in the file *
8 // * LICENSE and available at http://cern.ch/geant4/license. These *
9 // * include a list of copyright holders. *
10 // *
11 // * Neither the authors of this software system, nor their employing *
12 // * institutes, nor the agencies providing financial support for this *
13 // * work make any representation or warranty, express or implied, *
14 // * regarding this software system or assume any liability for its *
15 // * use. Please see the license in the file LICENSE and URL above *
16 // * for the full disclaimer and the limitation of liability. *
17 // *
18 // * This code implementation is the result of the scientific and *
19 // * technical work of the GEANT4 collaboration. *
20 // * By using, copying, modifying or distributing the software (or *
21 // * any work based on the software) you agree to acknowledge its *
22 // * use in resulting scientific publications, and indicate your *
23 // * acceptance of all terms of the Geant4 Software license. *
24 // *****
25 //
26 // $Id: B1SteppingAction.cc 74483 2013-10-09 13:37:06Z gcosmo $
27 //
28 // \file B1SteppingAction.cc
29 // \brief Implementation of the B1SteppingAction class
30 //
31 #include "B1SteppingAction.hh"
32 #include "B1EventAction.hh"
33 #include "B1DetectorConstruction.hh"
34 //
35 #include "G4Step.hh"
36 #include "G4Event.hh"
37 #include "G4RunManager.hh"
38 #include "G4LogicalVolume.hh"
39 //
40 //.....0000000000.....0000000000.....0000000000.....0000000000.....
41 //
42 B1SteppingAction::B1SteppingAction(B1EventAction* eventAction)
43 : G4UserSteppingAction(),
44 fEventAction(eventAction),
45 fScoringVolume(0)
46 {}
47 //.....0000000000.....0000000000.....0000000000.....0000000000.....
48 //.....0000000000.....0000000000.....0000000000.....0000000000.....
49
50 B1SteppingAction::~B1SteppingAction()
51 {}
52 //.....0000000000.....0000000000.....0000000000.....0000000000.....
53 //.....0000000000.....0000000000.....0000000000.....0000000000.....
54
55 void B1SteppingAction::UserSteppingAction(const G4Step* step)
56 {
57     if (!fScoringVolume) {
58         const B1DetectorConstruction* detectorConstruction
59             = static_cast<const B1DetectorConstruction*>
60             (G4RunManager::GetRunManager()->GetUserDetectorConstruction());
61         fScoringVolume = detectorConstruction->GetScoringVolume();
62     }
63
64     // get volume of the current step
65     G4LogicalVolume* volume
66         = step->GetPreStepPoint()->GetTouchableHandle()
67         ->GetVolume()->GetLogicalVolume();
68
69     // check if we are in scoring volume
70     if (volume != fScoringVolume) return;
71
72     // collect energy deposited in this step
73     G4double edepStep = step->GetTotalEnergyDeposit();
74     fEventAction->AddEdep(edepStep);
75
76     std::ofstream energy;
77     energy.open("Energy.data", std::ios::app);
78     if (energy.is_open())
79     {
80         energy<<edepStep<<G4endl;
81     }
82     energy.close();
83
84     //std::ofstream position;
85     //position.open("position.data", std::ios::app);
86     //if (position.is_open())
87     //{
88     //    position<<"<<xPre<<"<<yPre<<"<<zPre<<"
89     //    <<"<<xPost<<"<<yPost<<"<<zPost<<"<<edepStep<<"
90     //    <<G4endl;
91     //}
92     //position.close();
93 }
94 //.....0000000000.....0000000000.....0000000000.....0000000000.....
95 //.....0000000000.....0000000000.....0000000000.....0000000000.....
96
97 src/B1SteppingAction.cc
```