

GEANT-4 GPU Port:

Test Plan

Stuart Douglas – dougls2
Matthew Pagnan – pagnanmm
Rob Gorrie – gorrierw
Victor Reginato – reginavp

Version 0
October 25, 2015

Contents

1	General Information	1
1.1	Summary	1
1.2	Risks	1
1.3	Constraints	1
1.4	Definitions and Acronyms	1
2	Test Types	1
2.1	Manual and Automatic Testing	1
2.2	System Testing	2
2.3	Structural Testing (Stress Testing)	2
2.4	Unit Testing	2
2.5	Code Testing	2
3	Testing Factors	2
3.1	Factors to be Tested	2
3.2	Description of Factor	2
3.2.1	Correctness	2
3.2.2	Performance	3
4	Test Items	3
4.1	Requirements Testing	3
4.2	Code Testing	3
4.3	User Manual Testing	3
4.4	Error Handling Testing	3
5	Automated Testing Plans	4
6	Schedule	4
6.1	Testing Schedule	4
6.2	Deliverables	4

Revision History

All major edits to this document will be recorded in the table below.

Table 1: Revision History

Description of Changes	Author	Date
Initial draft of document	Stuart, Matthew, Rob, Victor	2015-10-26

1 General Information

1.1 Summary

1.2 Risks

The following table outlines the major risks associated with the testing of the product. A more detailed analysis of each of the risks follows the table.

Table 2: Risks

Risk #	Summary	Severity
1	differing order of random numbers on GPU could lead to difficulty comparing results with simulations run on CPU	Very High
2	isolating GEANT4 methods to test with unit tests may be too difficult	High
3	running time of tests will be too long to run them frequently	High

Risk 1 – Random Numbers:

The GEANT4 project is heavily dependent on random numbers. Random numbers are used to determine attributes about particles (independent of all other particles) as they move through the system. By parallelizing the workload, the order in which the particles are evaluated may change, causing it to draw a different random number from the sequence, leading to different results.

1.3 Constraints

1.4 Definitions and Acronyms

2 Test Types

2.1 Manual and Automatic Testing

Manual testing needs to be conducted by people, where test cases and inspections are manually performed. On the other hand, automated testing relies less on people and

performs tests quickly and effectively returning feedback on results not meeting expectations. Manual testing is more flexible; However, it is far more time consuming.

Our System testing shall be done manually since there is some graphical output. Most of the system tests will have a unique output format that our tester will have to verify that it is correct. Our unit testing shall be mostly automated.

2.2 System Testing

There are several examples that come with GEANT4 which we shall use along with other simple test files we will create ourselves. There are some specific test files the engineering physics department has run on the non-GPU GEANT4 system that we will also use for testing. Due to the time it takes for computation for some of these test files on the non-GPU system we will not run many of those files again on the non-GPU system in order to compare them with the GEANT4-GPU computation times. In those cases we will run those files only on GEANT4-GPU to ensure that those files are still able to run.

2.3 Structural Testing (Stress Testing)

Structural tests focus on the program's internal structure by evaluating the structure and the non-functional requirements of the system, particularly on any abnormal or extreme behavior.

We will have several tests where there will be an extremely high number of particles to process. We will test to see how this affects the computation time compared to the non-GPU code. We will also check to ensure that the large number of particles does not crash the system when it doesn't on the non-GPU code.

2.4 Unit Testing

2.5 Code Testing

3 Testing Factors

3.1 Factors to be Tested

3.2 Description of Factor

3.2.1 Correctness

Tests run on GEANT4-GPU should provide the same resulting output as tests run on the non-GPU GEANT4

Rationale

In order for GEANT4-GPU to be of use it must still be able to compute the correct outputs.

Methods of testing

- The output of tests run on GEANT4-GPU will be compared with the output of tests run on the non-GPU GEANT4

3.2.2 Performance

Tests run on GEANT4-GPU should take less time to compute compared to the same tests run on the non-GPU GEANT4

Rationale

The focus of this project is to reduce the computation time for GEANT4 programs by using a GPU.

Methods of testing

- Compare the computation times of tests run on GEANT4-GPU with the computation times of test run on the non-GPU GEANT4 to see if the GEANT4-GPU computation times are smaller.
- Stress test GEANT4-GPU by having it compute programs that would take an infeasible amount of time to compute on non-GPU GEANT4.

4 Test Items

4.1 Requirements Testing

4.2 Code Testing

4.3 User Manual Testing

4.4 Error Handling Testing

Error handling indicates to the extent at which users can recover or know the basis of an error occurring in the system. Error handling would imply trying to find ways of committing errors, such as inputting too many or too few arguments or wrong data in general. For examples, if users try to call one of the new functions and provide it with the wrong types of data, they must be clearly shown the reason for their failure and, if possible, present a solution. Error handling would be performed both, manually and

in an automated setting, once a set of potential errors are discussed and figured out amongst the members of the testing team.

5 Automated Testing Plans

6 Schedule

6.1 Testing Schedule

6.2 Deliverables