Assignment 3

Deadline:	Hand in by 5pm on Friday 17th October 2014 (end of semester)
Evaluation:	30 marks – which is 30% of your final grade
Late Submission:	5 Marks off per day late
Work	This assignment is to be done **individually** – your submission may be checked for plagarism against other assignments and against Internet repositories. If you adapt material from the Internet acknowledge your source.
Purpose:	To give you an opportunity to use advanced programming methods in a more extensive and complex simulations project.

Problem to solve:
Develop and document a simple particle simulation and visualisation in C++, Java, Python (or another language if approved by the lecturer).

Requirements:
As a minimum requirement your assignment should simulate and visualise a number of particles in two-dimensions that collide with rectangular boundaries and other particles. In addition to this requirements you should choose and implement at least one of the following extensions (or other approved by lecturer):

1. Simulate particles in three-dimensions instead of two.
2. Non-rectangular boundaries.
3. Particles with different masses and radii
4. Uniform force on the particles 'wind' or 'gravity'
5. Potential between particles
6. More advanced collision methods

Develop your code in **C++**, **Java**, **Python** or another approved language and make sure you document what you have done and how it is implemented. Be prepared to demonstrate your simulation to the class.

Hand-in: Submit your **program and documentation** including discussion of your results (a zip file is acceptable) by email to d.p.playne@massey.ac.nz

Marks will be allocated for: correctness, fitness of purpose, utility, style, use of sensible **comments and program documentation**, and general elegance. Good comments will help me to award you marks even if your code is not quite perfect. You will also get marks for having writing up some discussion/conclusions – ie signs that you have really thought about the problem.

Additional Notes on the Technical Specification and Hints:

What class structure are you going to use for your simulation?

How can the simulation be optimized for speed?

What software mechanisms might you use to maximize code reuse in your system?

How are you going to document your simulation design?