Assignment 4

Deadline:	Hand in by 5pm on Friday 8th June 2012 (1 week after end of Semester1)
Evaluation:	40 marks – which is 40% 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 make use of the graphics libraries and ideas you have learned.

Problem to solve:

Pick **ONE** of the project ideas below.

Requirements:

Develop an appropriate **graphical program** using the libraries/graphics system of your choice that demonstrates an understanding of the problem involved.

Be prepared to demonstrate your visual simulation/animation to the rest of the class.

Hand-in: Submit your program, some screendump images, source code and documentation (a zip file is acceptable) by email to k.a.hawick@massey.ac.nz

Marks will be allocated for: visual impact, correctness, sensible use of graphics libraries and packages, fitness of purpose, utility, style, use of sensible **comments and program documentation**, and general appearance and panache! Good comments will help me to award you marks even if your code is not quite perfect.

Pick ONE of these ideas overleaf. Recall this assignment is worth 40 % so let that be a guide on how much time to spend on it. As usual, use and adapt any source code you wish but make sure you attribute it and that your project documentation makes it clear what your contribution is.

If you have any questions about this assignment, please ask the lecturer.

Project 1

Extend the Diffusion Limited Aggregation Assignment 2 to cope with multiple clusters. This is known as cluster-cluster aggregation in the literature. Start with an empty 3d space and randomly add a smallish fraction of filled sites. Allow all filled sites to "walk" randomly until they meet another, after which they join into a cluster. Whole clusters now move around randomly and if two clusters touch they join into a single larger cluster. So the number of clusters in the system will start off the same as the number of walkers, and could (in principle if you run the simulation long enough) end up with one giant super cluster. You can experiment with what to do if the boundary conditions are fixed or periodic. What does the cluster size distribution look like? Long tail? Flat? Can you colour the clusters according to size? Track the largest? Render as cubes or spheres or…? What if clusters are allowed to randomly break up if they get too big?

<u>Project 2</u>
Simulate and animate a self -organized criticality (SOC) sandpile model on a <u>hexagonal grid</u>.

The SOC Sandpile model was invented by Bak et al in 1987 to model avalanches and earthquakes and critical phenomena. The original model starts with an empty square mesh and a source of falling sand grains (like the DLA cells). The rules are that grains are added (at arbitrary random) positions and that each site is characterized by a height **h**. Adding a grain means the height of that site goes to **h+1**, but if at any time a site or column exceeds some critical height **h_c** it topples and the grains are redistributed on the neighbouring site.

Build an animate d graphical simulation that allows various mesh sizes, and choices of **h_c** and random starting seeds. A good animation will let you rotate around the avalanching system, but will also let you choose different ways to add sand grains – eg at a single site or at multiple sites or randomly. Sand grains can fall off the edge of the grid.

This is a well-studied system on square meshes – but for novelty try it on a hexagonal mesh (edge length between 20 and 200), so that sites have six neighbouring sites and not just four as they would on a square mesh. Think of interesting ways to colour the grains including the falling ones, and render some measured properties such as a histogram of the site heights. You might also colour sites and cells in interesting ways such as by height or number of neighbours or by time since last avalanche at that site. Visualising details is important - light and shade the system so the growth details are visible to the user. (See Google "Per Bak, Sandpile")

<u>Project 3</u>
Extend your version of assignment 3 into a "First Person Shooter" game.

Take an model scene – which can be made up of any model files you find on the Internet or a geometrically generated scene of your own devising – and extend the idea of the user flying through the scene into a more interactive scenario where you need to "shoot" or "collect" some objects and "avoid" certain other objects. You will likely need a good understanding of multi threading to tackle this project since you will nee d to dynamically interact with the game, which itself needs a thread to generate the targets etc.

<u>Project 4</u>
Use the graphics rendering technologies you have learned to develop a graphical browsing tool.

You may wish to browse a database, a file system, or any other collection of information - such as images, music files, movie files… Can you make a three dimensional viewer for information? Aspects to consider are how to tackle large or highly nested or structured data. (Recall the 3-D file browser interface used to control Jurassic Park…)

<u>Project 5</u>
For any of you interested in parallel computing, build a parallel ray tracer using OpenGL, CUDA and GPUs. Come discuss this with me if interested.

<u>Project 6</u>
Impress me!

Come up with any software involving any simulation or graphics idea that has a "high wow factor" using 3d graphics and animation techniques. (You are highly recommended to discuss your ideas with me first…)