

Probability Machine

You are to produce three dimensional graphics, and two or three dimensional physical simulation of a probability machine. **This coursework is worth 100% of a 20 credit module.**

A probability machine consists of a number of pegs arranged between two flat boards. Balls are introduced at a random point in a delivery shoot at the top of the machine. The balls fall, bouncing in between the pegs down to the bottom of the machine where they are collected in a number of buckets at the bottom of the machine. The amount of balls in the buckets should form a bell curve. A diagram is shown below in Figure 1.1.1.

Code should be written in C++ or C# and OpenGL and should be compiled and linked using Visual Studio 2008.

The following items should be submitted via E-Bridge as a single zipped folder:

- A brief report on of **what has been achieved, how to use the software, important techniques and algorithms** used and **potential improvements** (no more than three pages)
- Source code, associated VS.Net 2008 solution and project files, resource files and an executable program

Probability Machine Marking Scheme/Deliverables	Percentage
1.0 Graphics Deliverables	
1.1 Render 3D Probability Machine	20%
1.2 Add multiple cameras	5%
1.3 Add textures	5%
1.4 Add lights	5%
2.0 Simulation Deliverables	
2.1 Kinetics and collision detection	10%
2.2 Collision response	10%
2.3 Multiple balls	10%
2.4 3D Collision and response	10%
3.0 Common Deliverables	
3.1 Object Oriented Design	5%
3.2 Optimized Maths Library	5%
3.3 Achieve desired bell curve	5%
4.0 Additional Deliverables	
4.1 Quality of report	5%
4.2 Quality of demonstration	5%
Total	100%

Detailed Deliverables

This section expands upon the individual deliverables. The deliverables have been split into four sections. This does not represent the order in which they should be attempted. For example, deliverable 3.1 Object Oriented Design should be tackled at the start of the project and should be considered throughout. A suggested schedule to work to is presented at the end of this document.

1.0 Graphics Deliverables

This section describes the graphics deliverables required for a successful piece of work.

1.1 Render 3D Probability Machine

A diagram of the probability machine is shown in figure 1.1.1

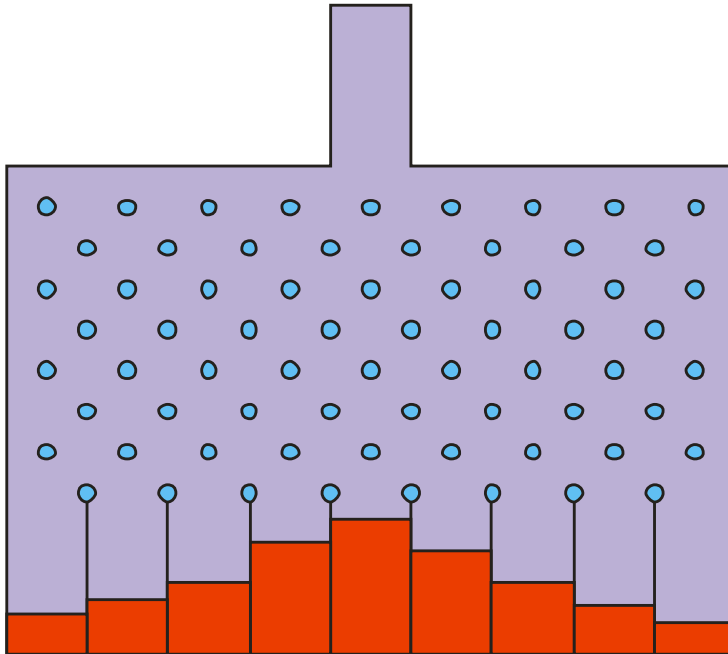


Figure 1.1.1 *The probability machine*

The probability machine consists of 68 pegs arranged into 8 rows as shown in Figure 1.1.1

The pegs should have a diameter of 2cm and should be arranged such that the centre of pegs that are vertically or horizontally aligned should be 10 centimetres apart.

The balls should have a diameter of 2.5cm.

The boards should be 10cm apart.

Buckets should be 10cm wide and 20cm deep.

The delivery shoot should be 10cm wide and 20cm deep. Marbles should be able to be randomly produced anywhere in this tube.

1.2 Add multiple cameras

Be able to view the scene using a variety of cameras. An excellent piece of work will feature a default camera that takes in the whole machine. A user controlled camera that can move around the scene and a zoomed camera that tracks the paths of a ball from a close up shot.

1.3 Add textures

Render the scene using one or more textures. An excellent piece of work will feature more than one texture and will ensure that disposal of the textures is handled correctly.

1.4 Add lights

Add lighting, surface normal's and material properties that enhances the scene. An excellent piece of work will use a variety of lights (positional, directional and spotlight) in an appropriate way.

2.0 Simulation Deliverables

This section describes the simulation deliverables required for a successful piece of work.

2.1 Kinetics and collision detection

The project should be able to animate falling balls, and consider the impact of running the simulation on different pieces of hardware (i.e. be processor independent). An excellent project will implement a superior method of collision detection.

2.2 Correct collision response

Balls should respond to collisions appropriately. An excellent project will take into account factors such as size and density when calculating the response to collisions.

2.3 Multiple balls

An excellent project will be able to have several balls running through the machine simultaneously, performing collision detection with one another and calculating the correct response. Although initially all balls should be the same, an excellent project will be able to include balls of varying size, density.

2.4 3D collisions and response

An excellent project will be able to perform all calculations in three dimensions rather than being restricted to a 2 dimensional plane.

3.0 Common Deliverables

This section describes deliverables required for a successful piece of work that can be considered common across both the graphics and simulation aspects of the project.

3.1 Object oriented design

An excellent project will be laid out in an appropriate way, and will use objects beyond those introduced in tutorials. An excellent project will use appropriate objects, functions and inheritance, as well as other good programming practice such as naming conventions, appropriate use of constants, static variables, objects, functions and inheritance.

3.2 Maths library

The project should include a set of maths classes. Maths classes that are used frequently should be heavily optimized. A maths library might include vectors in 2 and 3 dimensions, lines in 2 and 3 dimensions and matrices of various appropriate dimensions.

3.3 Achieve desired bell curve

At the bottom of the machine there should be nine buckets. Once a ball falls beyond the bottom of a bucket it should disappear. However, a record should be kept as to how many balls have ended in each bucket. The number of balls that have fallen into each bucket should be represented by the height of a panel on the front of each bucket, forming a bar chart. An excellent project will scale the contribution of each ball to the height of each panel such that no panel ends up so high that it obscures the main part of the machine. An excellent project will also exhibit a bell curve effect across the panels on each bucket.

4.0 Additional Deliverables

This section describes additional deliverables required as part of the submission of the work.

4.1 Quality of report

The project should be accompanied by a brief report on of what has been achieved, how to use the software, important techniques and algorithms used and potential improvements. The report should include appropriate screen shots demonstrating what has been achieved and should be no more than three pages long.

4.2 Quality of demonstration

Students are required to demonstrate their work in week 12. Failure to attend your demonstrating will result in a 40% cap of your work in line with departmental policy. Students should demonstrate their work on the lab machines, but may also use their own equipment, however, students are reminded that graphics can often appear significantly different on different hardware, so there is a risk associated with demonstrating work on machines that it was not developed on. Students are also reminded that there are no extensions given as a result of loss of work from your own hardware. An excellent demonstration will be able to show off all the implemented features of the simulation effectively, and should incorporate additional features where appropriate to do this. An excellent student will be able to set up scenarios to show features that might not be evident from the normal running of the simulation; and have the ability to slow down or pause the simulation as required.

Suggested Timetable

Students are reminded that this piece of coursework is worth **100%** of a 20 credit module. Students should place their time appropriately. A timetable for work is suggested below. A progress check will be carried out in week 7. By this time the majority of the graphics deliverables should be in place. The weekly tutorials are intended to help you stick to this schedule. It is intended that the weekly tutorials should give you the skill necessary to produce an excellent piece of work, and that both the tutorials and the course work be completed in parallel. If you have any questions or feel like you're struggling talk to a demonstrator or email S.Grey@hull.ac.uk.

[illegible]

