CSC3222 Project 1

Welcome to the practical classes for CSC3222. It is essential that we, as programmers, apply the theories we learn in a variety of engineering situations. Understanding the theory is an essential skill, but understanding the limitations of the theory in an implementation sense is just as important as you move forward with your software development career. The purpose of these practicals is to provide some experience of that, and to generate a platform from which you can launch your own exploration of the concepts discussed in the lecture series.

Module Assessment

There are 3 pieces of assessment for this module:

1. **Project 1 which is submitted to NESS. (35)**
2. Project 2 which is submitted to NESS. (15)
3. An exam in June. (50)

Project 1 (Deadline: 11:45pm, Wednesday 13th April, 2016)

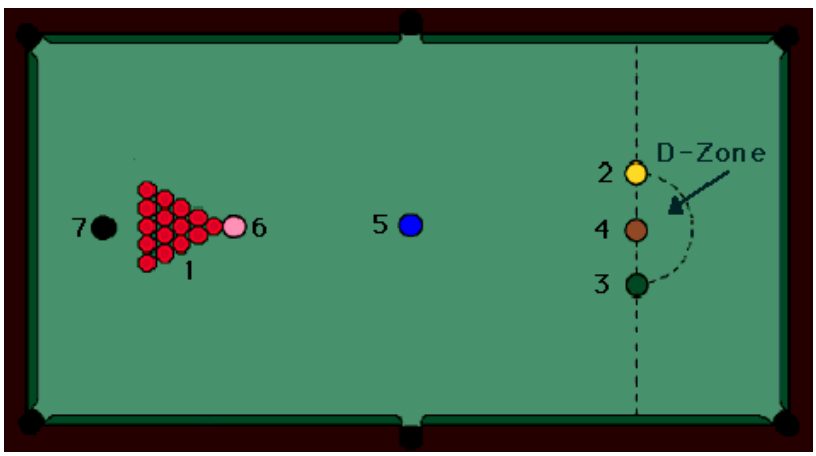
Aims

Produce a graphical simulation allowing circles to collide (bounce off each other) within a bounded cuboid arena resembling a snooker table. This will be achieved via a Newtonian physics implementation using numerical integration. The simulation will detect when a circle moves over a corner region (a pocket) and remove it from the simulation.

Minimum Deliverable Software Specification (Up to 25 Marks)

The capability to process physical interactions between virtual entities in real-time is a fundamental element of almost all game engines. Understanding the processes which underpin those technologies is a very valuable asset to the aspiring game programmer. This Project explores this issue in detail.

Using C++, in Microsoft Visual Studio, implement a graphical simulation in which coloured circles of uniform size are arranged in the following configuration, and begin the simulation at rest:



A white circle will be positioned somewhere on the table by the player such that it does not intersect with any other circle. The player must be able to input an initial velocity which will be applied to the white circle. If the white circle collides with another circle, the collision should be resolved with Newtonian mechanics assuming conservation of momentum. If that collision leads onto another collision, this too must be resolved, and so on.

Friction will be modelled in some fashion, to eventually bring the circles to rest. If any circle collides with a table edge, the circle should bounce back off it. A region near each corner of the table should be defined as a 'pocket'; a circle whose centre passes over a pocket should leave the simulation. If that circle is the white circle, the player will be able to place it back on the table and again 'fire' it by inputting an initial velocity.

Code must be commented and functions described. Code will make good use of object orientation, and variables shall use appropriate and transparent naming conventions. A display of the framerate will be toggled with the 'F' key. The submitted software must compile and execute; keep in mind that this means you must employ relative pathing when linking libraries, etc., and it is advisable to test the software on another PC before uploading it to NESS.

Additional Software Features (Up to 10 Marks):

You are encouraged to extend the simulation into a more fully featured snooker simulator through the inclusion of additional features. Such features should be documented and explained in code comments. Marks are awarded based on the complexity of the features implemented. Some example features you might include are:

- The inclusion of an arrow giving an indication of the direction and magnitude of the velocity the player is applying to the white circle before the player confirms their decision.
- The inclusion of a turn-based play mode and scoring system.
- The inclusion of middle pockets.
- The requirement that the player place the white circle within the 'D-Zone' (see diagram).

Deliverables

Zipped Microsoft Visual Studio Project, Source Code and Executable (submitted via NESS)
Demonstration of Working Software (to take place on Thursday 14th April at 11AM in the Rack, or earlier by individual arrangement)

Mark Scheme

Submitted Software:	35
Total:	35