DEPARTMENT OF COMPUTER SCIENCE

COURSEWORK ASSESSMENT DESCRIPTION

MODULE DETAILS:

Module Number:	08964	Semester:	2
Module Title:	Simulation and Concurrency		
Lecturer:	WJV / DPMW / DDM		

COURSEWORK DETAILS:

Coursework Assessment Number:	1	of	1		
Title of Assignment:	Gravity Wells				
Format:	Program	Report	Demonstration		
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	75	and	125	hours on this assessment
Length of Submission:	This assignment should be **no** more than:	1500 **words** *(excluding diagrams, appendices, bibliography, code)*			

PUBLICATION:

Date of issue:	Week 3

SUBMISSION:

ONE copy of this assignment should be handed in via:	E-Bridge	If Other (please state method)	
Time and date for submission:	**Intermediate Review** **Source Code** **Report** **Demonstration**	Week 8 during lab slot 17:00 Monday 10th May - Week 12 17:00 Monday 17th May - Week 13 Weeks 13 & 14	
If **multiple hand–ins** please provide details *(as appropriate)*:			

The assignment should be handed in **no later** than the time and date shown above, unless an extension has been authorised on a *Request for an Extension for an Assessment* (Mit Circs) form which is available from the Office or http://www.student-admin.hull.ac.uk/downloads/Mitcircs.doc. The extension form, once authorised by the lecturer concerned, should be sent to Amanda Millson.

MARKING:

Marking will be by:	Student Name

BEFORE submission, each student must complete the **correct** departmental coursework cover sheet and attach it to your work, dependant upon whether the assignment is being marked by student number, student name, group number or group name. This is obtainable from the departmental student intranet at http://intra.net.dcs.hull.ac.uk/sites/home/student/ACW%20Cover%20Sheets/Forms/AllItems.aspx

ASSESSMENT:

The assignment is marked out of:	100	and is worth	100	% of the module marks

ASSESSMENT STRATEGY AND LEARNING OUTCOMES:
The overall assessment strategy is designed to evaluate the student's achievement of the module learning outcomes, and is subdivided as follows:

LO	Learning Outcome	Method of Assessment *{e.g. report, demo}*
1	*Demonstrate research, selection and asessment of concurrent and distributed architectures and implementation techniques*	Program, Report
2	*Analyse real world problems and identify appropriate physically-based algorithms for their simulation*	Program, Report
3	*Implement distributed applications in C++*	Program
4	*Implement real world simulation, applying techniques from mathematics and physics, for use in virtual environments and computer games*	Program
5	*Use the mathematical techniques of vectors, matrices and numerical integration*	Program

Assessment Criteria	Contributes to Learning Outcome	Mark
Quality of system architecture	1	15
Quality of distribution implementation	1,3	25
Quality of completed product	1,3	10
Quality of Visualization and input/output	4	10
Performance of the collision detection algorithms	4,5	15
Performance of the collision response	4,5	10
Overall PBM design, implementation and selection of algorithms	2,4	15

FEEDBACK

Feedback will be given via:	Mark Sheet	Feedback will be given via:	Verbal (via demonstrati
Exemption (staff to explain why)			
Feedback will be provided no later than 20 working days after the submission date.			

This assessment is set in the context of the learning outcomes for the module and does not by itself constitute a definitive specification of the assessment. If you are in any doubt as to the relationship between what you have been asked to do and the module content you should take this matter up with the member of staff who set the assessment as soon as possible.

You are advised to read the **NOTES** regarding late penalties, over-length assignments, unfair means and quality assurance in your student handbook, also available on the department's student intranet at: **http://intra.net.dcs.hull.ac.uk**. In addition, **please note** that if one student gives their solution to another student who submits it as their own work, **BOTH** students are breaking the unfair means regulations, and will be investigated.

In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility, not the Department's, to produce the assignment in question.

Assignment Details

08964 ACW "Gravity Wells"

The aim of the ACW is to research, design and implement a distributed, physically-based modelling demonstrator, for a series of gravity wells.

The world consists of a very large, planar surface, surrounded by a circular wall. A large number of spheres are placed on the surface.

The simulation is run across a number of peers, each displaying a consistent view of the world. The simulation allows a user to apply a force to the balls.

Definitions

- *Region of influence*: A spherical region, centred on each peer, representing the maximum distance at which forces are applied to the balls.
- *Ownership*: A ball can only be owned by a single peer. Ownership can be transferred.
- *Contended*: A ball is contended if it has forces acting on it from more than one peer i.e. it exists within the region of influence of more than one peer.

Requirements

World

The world consists of a very large, planar surface upon which are placed a large number of balls (ranging from 256 – 64k balls, defined within a configuration file). The surface is surrounded by a circular wall.

The balls are initially arranged in a regular grid on the surface.

Each ball has a mass assigned which is light, medium or heavy (you should define masses in real units and indicate this via the shading of the ball). Balls move under the forces applied by the peers, they are effected by a gravity force that either attracts or repels them. Three types of motion should be supported in your demonstrator:

1. Balls are treated purely as particles with dimension, they do not have rotation but move smoothly over the surface
2. Balls are still treated as particles as far as the physics simulation is concerned but rotates in such a ways as to mimic a real ball rolling over the surface.
3. Advanced: Balls should be modelled as rigid bodies and therefore rotation should be fully modelled within the physics equations.

In all cases, you should include a frictional force (defined in your configuration file) between the ball and the floor. You should also include elasticity (defined in the configuration file) when balls collide with each other and with the cylindrical wall.

Collision detection and response is required for ball-to-ball, ball-to-surface and ball-to-wall collisions.

Extending the concepts described above, you should also be able to define some of the balls to be soft deformable objects which are model by a spring-dashpot system or otherwise. These balls should be scattered around the plane and clearly identifiable. These soft balls should deform on impact when they collide with other balls and the walls.

Peers

The world is distributed across a number of peers. The maximum number of peers is four. Each peer stores only partial knowledge of the world. Each peer owns a subset of the total balls within the simulation.

As a peer moves across the surface, it broadcasts its new position to all of its peers. When peers receive these broadcast messages, they are responsible for checking this updated position against their knowledge of the world. Any balls within the region of influence of their peer are noted. If the ball is outside of their own region of interest, then ownership of the ball is transferred to their peer. If the ball is within both their region of interest and that of their peer, then the ball is flagged as being contended.

When a ball is contended, peers who do not have ownership of the ball are responsible for passing their force data to the ball's owner. The force data is the value of all the forces applied to the ball, originating from that peer (i.e. their gravity well and collisions with their balls). The owner of the ball is responsible for collating this force data, and calculating a new position for the ball. The new position is then communicated back to the interested peers.

When a peer is added to the world it has no knowledge of any balls. Balls will gradually migrate to this new peer during the course of the simulation. Load balancing algorithms could be considered to improve the efficiency of the simulation.

Visualisation

The view of the world is through a number of peers. Each peer has a set field of view covering a small proportion of the total surface. The view should be a 3d perspective projection at approximately 30 degrees to the horizontal.

The view should include:

- A graphical representation of the region of influence, i.e. colouring the surface.
- All balls that are within the region of influence.
- The balls should be colour coded to show ownership and any contention, and should show that the balls are rolling. You should give an indication of mass either through colouring or texturing.
- You should take care in your visualisation to ensure that balls are sufficiently large on the screen to show the accurate motion and collision detection implemented in your system. You may wish to add a zoom control to help achieve this.
- Note: all the balls have the same radius and the gravity well is always positioned at a height above the plane equal to the radius of a ball.

A HUD should be provided that shows:

- Number of balls owned by the peer
- Number of balls currently contended
- Total number of balls in the system
- Magnitude of the force being applied

Control

Control of the world is through a number of peers. Each peer is able to affect the balls within their region of influence, by applying a positive or negative vertical force at the point where their view direction vector intersects the surface.

The control scheme uses the mouse:

- Left button, applies an attractor force. The longer the button is held the greater the force
- Right button, applies a repellor force. The longer the button is held the greater the force
- Left-Right button together, cancels the force
- Moving the mouse, moves the position of the peer in a plane parallel to the surface.

Implementation

Networking

Only the Winsock library is permitted.

Fault tolerance is required. Network connections will be broken and reconnected during the final demonstration. The simulation should behave in a consistence manner.

This application is a peer-to-peer architecture. No server of any type should be used.

Threads

Only the Win32 threading library is permitted.

The quad core processors in lab 177 are considered the target platform. Threading is to be used to leverage the performance of these processors.

Research questions

The following are a few points you may wish to consider:

- What is the visual impact from the perspective of a peer on a non-owned, contended ball when the owner does not update the position fast enough?
- The simulation starts with one peer owning all the balls. As new peers are added into the simulation, how are the balls migrated to improve performance?
- What happens when a ball rolls out of the region of influence? Who is responsible for updating the position of that ball?

Report

The structure of the ACW report is as follows:

- System architecture, including where threads and networking have been used (2 pages max), plus UML diagrams
- Physics of the rolling ball and deformable ball (2 pages max)
- Project critique (1 page max)

Marks will be lost if the page limit is exceeded.