DEPARTMENT OF COMPUTER SCIENCE

COURSEWORK ASSESSMENT SPECIFICATION

MODULE DETAILS:

Module Number:	08227	Semester:	1 and 2
Module Title:	Advanced Programming		
Lecturer:	PMC / WJV		

COURSEWORK DETAILS:

Coursework Assessment Number:	1	of	1		
Title of Assignment:	The Gibraltar Invasion				
Format:	Program	Report			
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	100	and	125	hours on this assessment

PUBLICATION:

Date of issue:	Week 4

SUBMISSION:

ONE copy of this assignment should be handed in via:	Class Server	Other (please state method)	
Time and date for submission:	**9:30**	Friday 9th May 2008	
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* 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 attached to the assignment on submission (or given to the lecturer in the case of electronic submission).

MARKING:

Marking will be by:	Student Name

BEFORE submission, each student must complete the **correct** departmental coursework cover sheet 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	60	% 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}*
2	*Demonstrate understanding of game development techniques*	Program / Report
3	*Develop a robust efficient and real-time C++ application*	Program
4	*Deploy an Integrated Development Environment (IDE) to create, debug and optimise a well structured implementation in C++*	Program

Assessment Criteria	Contributes to Learning Outcome	Mark
Minimum Specification	2, 3, 4	45
Bonus Specification	2, 3, 4	25
Quality of Code	3, 4	25
Report	2, 3	5

FEEDBACK

Feedback will be given via:	Feedback Sheet	Other (please state method)	
Other feedback (if appropriate) will be given via:			
Feedback will be provided no later than: *(please state date, week or month)*	June 2008		

Questions
If you have any questions regarding this assessment you **MUST** speak to the lecturer as soon as possible.

You are advised to read the **NOTES** regarding Late Penalties, Use of Unfair means and Quality Assurance on the department's student intranet at:
http://intra.net.dcs.hull.ac.uk/sites/home/student/ACW%20Cover%20Sheets/Forms/AllItems.aspx

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 attached)*

The Gibraltar Invasion

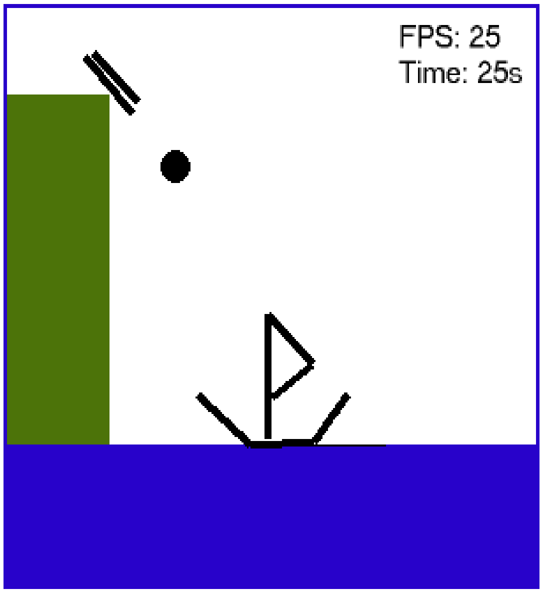

Fig 1 – Illustration of the ACW specification

The objective of this ACW is to develop a program using C++ in order to produce the following specification (depicted in fig 1):

Minimum Specification (45%)

1. A display window consists of a cliff face (shown in green), the sea (shown in blue), and a cannon (shown on top of the cliff).
2. The boat moves horizontally on the water from side to side in the window (e.g. from left to right, then right to left, and so on).
3. The cannon's elevation/pitch is controllable by using the up/down cursor keys.

4. A cannonball can be fired from the cannon in the direction that the cannon is pointing by using the space bar. For this minimum specification, the cannon ball can travel in a straight line. The cannonball class must be extended from a base class called NewtonianObject.

5. The NewtonianObject class has at least a pure virtual function called Update(float dt). This function must be used to update the Cannonball's position etc.

6. If the cannonball hits the ship then the ship will sink. At which point another ship will sail into view from one of the sides of the screen.

7. The program should use a clock so that there is processor independent animation. For example, the position of the cannonball per frame would equate to ***cannonBallPos += velocity*dt***

8. You should also display a frames per second counter and also a timer which shows the amount of times that has elapsed since the program started.

9. The boat moves at random speeds as it moves across the water.

Bonus Specification (25%):

1. Provide the user with the ability to change the power of shot by using the left/right cursor keys. Also provide a visual system to show the user the power (e.g. display the power value as text, implement a power bar).

2. Add more realistic movement to the cannonball's motion (modelling gravity etc).

3. Multiple boats based on a base Ship class containing virtual draw functions. For example two different ships.

Quality of Code (25%):

1. Design approach and class design (e.g. Object Orientated Design). Include a class diagram in the report (see below).

Short Report (5%):

1. Maximum of two sides of A4.

2. Summary of the main algorithms that you have used to tackle the ACW.

3. Include a list of all bonus and novel elements implemented.

4. Design approach and class design. Include a class diagram.