COMPUTER SCIENCE
ASSESSMENT DESCRIPTION 2019/20

MODULE DETAILS:

Module Number:	700106/700120	Trimester:	1
Module Title:	Real-Time Graphics / C++ Programming and Design		
Lecturer:	Qingde Li / Warren Viant / Xinhui Ma		

COURSEWORK DETAILS:

Assessment Number:	1 1	of	1 (700106) 1 (700120)		
Title of Assessment:	Software Portfolio				
Format:	Program	Report	Demonstration		
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	120	and	150	hours on this assessment
Length of Submission:	This assessment should be **no** more than: *(over-length submissions **will be** penalised as per University policy)*	2500 **words** *(excluding diagrams, appendices, references, source code)*			

PUBLICATION:

Date of issue:	23rd Sept 2019

SUBMISSION:

ONE copy of this assessment should be handed in via:	Canvas		If Other (state method)	
Time and date for submission:	**Time**	2pm	**Date**	See Below
If **multiple hand–ins** please provide details*:*	Source Code – 7th January 2020 Report and Video – 10th January 2020 Demonstration – w/b 14th January 2020			
Will submission be scanned via TurnitinUK?	No	If submission is via TurnitinUK students MUST only submit Word, RTF or PDF files. Students MUST NOT submit ZIP or other archive formats. Students are reminded they can **ONLY** submit **ONE** file and must ensure they upload the correct file.		

The assessment must be submitted **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: http://www2.hull.ac.uk/student/registryservices/currentstudents/usefulforms.aspx

If submission is via TurnitinUK within Canvas staff must set resubmission as standard, allowing students to resubmit their work, though only the last assessment submitted will be marked and if submitted after the coursework deadline late penalties will be applied.

MARKING:

Marking will be by:	Student Name

ASSESSMENT:

The assessment is marked out of:	100	and is worth	100	% of the module marks
N.B If multiple hand-ins please indicate the marks and percentage apportioned to each stage above, i.e. Stage 1–50, Stage 2–50. It is these marks that will be presented to the exam board.				

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:

700106

LO	Learning Outcome	Method of Assessment
1	*Show evidence of a systematic and comprehensive understanding of real-time computer-graphics algorithms, techniques, and architectures. Adapt approaches, including some at the forefront of the discipline, and identify possibilities for originality or creativity.*	Program, Report
2	*Implement an efficient, robust, real-time computer-graphics application.*	Program
3	*Apply selected mathematical techniques of linear algebra and projective geometry to real-time graphics.*	Program

700120

LO	Learning Outcome	Method of Assessment
1	*Show evidence of a systematic and comprehensive understanding of object-oriented principles by producing a design that meets identifiable requirements and standards. Adapt approaches including some at the forefront of the discipline and identify possibilities for originality or creativity.*	Program, Report
2	*Show evidence of a systematic and comprehensive understanding of the C++ programming language by producing source code to identifiable standards.*	Program
3	*Use appropriate development tools and processes to create, debug, test and optimize an efficient, robust, real-time, C++ application based on an object-orientated design.*	Program

Assessment Criteria	Contributes to Learning Outcome	Mark
Quality of implementation / graphics features	1,2,3	80
Quality of portfolio	1	20

FEEDBACK

Feedback will be given via:	Verbal (via demonstration)	Feedback will be given via:	Mark Sheet
Exemption			
Feedback will be provided no later than 4 'teaching weeks' 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, which is available on Canvas - https://canvas.hull.ac.uk/courses/17835/files/folder/Student-Handbooks-and-Guides.

In particular, please be aware that:

- Your work has a 10% penalty applied if submitted up to 24 hours late
- Your work has a 10% penalty applied and is capped to 40 (50 for level 7 modules) if submitted more than 24 hours late and up to and including 7 days after the deadline
- Your work will be awarded zero if submitted more than 7 days after the published deadline.
- The over-length penalty applies to your written report (which includes bullet points, and lists of text you have disguised as a table. It does not include contents page, graphs, data tables and appendices). Your mark will be awarded zero if you exceed the word count by more than 10%.

Please be reminded that you are responsible for reading the University Code of Practice on the use of Unfair means (http://www2.hull.ac.uk/student/studenthandbook/academic/unfairmeans.aspx) and must understand that unfair means is defined as any conduct by a candidate which may gain an illegitimate advantage or benefit for him/herself or another which may create a disadvantage or loss for another. You must therefore be certain that the work you are submitting contains no section copied in whole or in part from any other source unless where explicitly acknowledged by means of proper citation. 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.

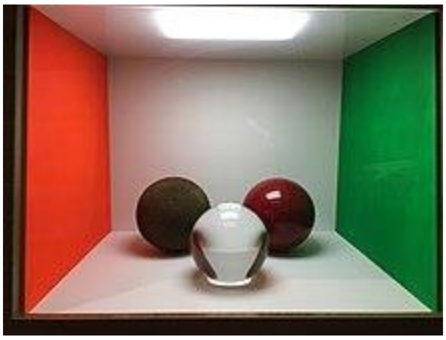
In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility, not the Department's, to produce the assignment in question.

ASSIGNMENT DETAILS

The two modules 700106 Real-Time Graphics and 700120 C++ Programming and Design are assessed through a single portfolio of work, allowing you to develop a substantial piece of software and demonstrating your ability and knowledge of (real-time interactive) 3D computer graphics.

CONCEPT

The aim is to create a variant of the Cornell Box; a box containing a small number of simple geometry objects.



Cornell Box (https://en.wikipedia.org/wiki/Cornell_box)

MANDATORY FEATURES

These features, when perfectly implemented, will provide 50% of the final mark. In addition, these features are a guide to the necessary building blocks, as well as the expected functionality.

Box
A box with the opaque walls when viewed from inside and transparent walls when viewed from outside.

Geometry
At least two spheres and one cuboid to be placed within the box. The rendering technique for each object is to be randomly selected from the list of techniques below. No two objects should be rendered using the same technique.

1. Gouraud shading
2. Phong shading
3. Texture mapping
4. Bump mapping (advanced feature, optional)
5. Displacement mapping (advanced feature, optional)
6. Toon shading (advanced feature, optional)
7. Transparency (advanced feature, optional)
8. Environment mapping of just the box (advanced feature, optional)
9. Glowing using a post-processing technique (advanced feature, optional)

Lighting
The box should have the following lighting options

1. Ambient + single local light
2. Ambient + single local light rotating around the top of the box
3. Ambient + four spot lights (one in each of the top corners of the box)

Shadows
All geometry within the box should exhibit the following shadow effects. Note that the shadows will change based on the current lighting mode. Each object will have a single shadow when illuminated by a single directional light, or four shadows when illuminated by the four spot lights.

1. Static shadow object on floor only (not required to work with dynamic lights)
2. Basic shadow mapping (advanced feature, optional)
3. Advanced shadow mapping (advanced feature, optional)

Explosion
Each shape is assigned a unique integer value. If one of the numeric keys is pressed, the corresponding shape explodes into a cloud of dust, represented by a particular system
The dust must exhibit the following properties:

- Be subject to gravity and air resistance
- Collide with each of the objects, coming to rest on any flat surfaces

Ink
The box is to be filled with a dilute quantity of blue ink. The level of the ink can be controlled using the keys specified below. The effect of the ink is to temporarily change the colour of the parts of the box and objects that are submerged within the ink. (advanced feature, optional)

Cameras
The following cameras are to be adjustable via keyboard controls (see below)

- Camera1: Initially looking at the front of the box.
- Camera2: Initially looking at the top of the box from overhead.

Controls
Note: the following key assignments are non-negotiable; they are expected to work in each submission to ease the demonstration process:

- 'ESC' exits the application
- F1 to F2 will select Camera1 and Camera2, respectively, as explained above
- Cameras are controlled by the cursor keys:
    - 'left'/'right'/'up'/'down' rotate left/right/up/down, respectively
    - CTRL + 'left'/'right'/'up'/'down'/'page up'/'page down' panning to left/right/forward/backward/up/down, respectively
    - Alternatively, w-a-s-d or i-j-k-l keys may be used instead of the "cursor" keys
- F5 re-renders each object using a different random effect
- F6 switches between lighting modes
- F7 switches between shadow types
- F8 switches between scene level rendering modes (advanced feature, optional)
- '1', '2', etc. explodes the corresponding shape
- 't'/'T' decrease/increase a factor that globally slows/speeds-up time-dependent effects
- 'f'/'F' decrease/increase the level of ink in the box
- 'r' resets the scene

Configuration File
Scene elements and their animations paths as well as lights and camera configurations are read from a configuration file.

OPTIONAL/ADVANCED FEATURES

Each advanced feature has an associated difficultly point. The maximum mark available for each feature is this difficulty point. Only 30 marks are available for implementing advanced features. These 30 marks can be gained either by perfectly implementing features worth 30 points, or by partially implementing features worth up to 40 points. Any features implemented beyond 40 points will not be marked. Marks for advanced features will be capped at 30.

Shading
[4 points] bump mapping of geometry
[2 points*] displacement mapping of geometry [*builds on bump mapping]
[4 points] transparent geometry
[4 points] toon shading of geometry (e.g., toon/cel shading)
[4 points] environment cube mapping on all geometry

Shadows
[8 points] Shadow mapping
[4 points*] Advanced shadow mapping (e.g., PCF, VSM) [*builds on shadow mapping]

Scene level rendering modes
[8 points] Deferred rendering, including the ability to switch the visualization for the various MRT buffers
[6 points] Glowing effect
[4 points] High-dynamic range rendering and tone-mapped post-processing

Features

[6 points] Ink

Hints

The following may help you to successfully complete your work:

- Start now!
- Produce a paper visualization (e.g., story-board sketches) of your project and check it with a member of the module team
- Produce a top-level software design and check it with a member of the module team.
- Prototype your ideas to help produce a more detailed design
- Test your software at each stage of development
- Document as you go

IMPLEMENTATION

The final implementation must be able to run on a standard PC in the post-graduate graphics lab, i.e. FEN-177. Only the following libraries and APIs are permitted:

- DirectX 11 SDK
- GLM or DirectMath
- AssImp (http://www.assimp.org/)

The use of any language feature within C++ 11, 14 or 17 is permitted (please remember the final product needs to run on a standard PC in FEN-177).
The use of Boost Libraries, DirectXTk or WinRT are not permitted.
Prior approval is required for any other APIs and features, not specifically mentioned above.

REPORT

You are required to produce a single document containing the following sections:

Design (700120) – [word limit 1000]

- Class diagram(s) containing main classes
- Class diagram(s) containing service/utility classes
- A textual description giving the name, role and responsibilities of each class
- Interaction diagram(s) for significant components of the software design
- A critique of the design, including details on:
- The merits of the design?
- Weaknesses of the design?
- What has changed in the design?
- What would you do in a different way?

Graphics (700106) – [word limit 1000]

- Document and critique the algorithms used, including:
- Geometry representation and processing
- Shading and lighting
- Shadow generation
- Particle system
- Explain how application objects and their graphics representation are connected; discuss the advantages/disadvantages of your approach
- Explain how application-object behavior is updated and how these updates are propagated to the graphics representation
- Discuss potential extensions (especially with respect to non-implemented items from the advanced feature list) as well as potential scalability issues
- What feature (not necessarily mentioned before) would you have liked to add?

Project Management (700106 & 700120) – [word limit 500]

- Discuss your project management

Please note, that your report should adhere to the document standards for the Department of Computer Science and Technology. Marks will be lost if you exceed word limits; see handbook for over-length penalties!

VIDEO

A 3 to 5 minute video with audio commentary. The video should highlight each of the key elements of your implementation. It is strongly suggested that you work through the list of mandatory and optional features within the specification, showing each in turn.

DEMONSTRATION

You will be required to both demonstrate your solution, and take part in a code review (RESIT only). The code review will consist of your examiners selecting parts of your code and requiring you to explain the implementation and the rationale behind the design.