Faculty of Science and Technology

2018/2019

Assignment Brief		
Course/s: BSc (Hons) Games Software Engineering	**Unit Name:** Physics for Games **Unit Level:** Level 5	
Assignment set by: Wen Tang	**QA:** Leigh McLoughlin	
Assignment Issued: Monday 8th October 2018	**Recommended time to complete this assignment:** 35 hours	
Date Due: Friday 11h January 2019	**Unit Weighting:** 70%	**Assignment number:** 2 of 2
SUBMISSION METHOD(S) **Electronic Submission** Assignment is to be electronically submitted by 12:30 pm on the due date (please allow sufficient time to upload files before the deadline) via the **Large File Submission Link on Brightspace**		

The Assessment Title: Rigid Body Simulation in 3D

This In Course Assignment is designed to assess your understanding of physics simulation and your programming skills for digital games. A small scale physics engine software program in C++ is to be submitted with a physics engine architecture designed for games. You are required to use the OpenGL graphics API for the display of a rigid body simulation scenario in a 3D simulated environment to showcase your physics engine features (please see the list below). You must also provide a technical report along with your code that describes your engine architecture and development process. This assignment assesses the following skills and knowledge:

1. Demonstrate critical thinking with respect to the analysis, design and production of physics engines for games;
2. Design and structure a physics simulation engine framework and core elements;
3. Develop a rigid body simulation scenario in 3D to demonstrate functionalities of your physics engine;
4. Evaluate the basic algorithms and techniques that will affect the performance of the physics engine.

By the end of your assignment you should have at least a physics engine for 3D games with a number of features as well as a rigid body simulation demonstration to showcase your physics engine capabilities.

The following list illustrates the functionalities your physics engine should provide and the indicative program component mark they may achieve based on quality and completeness. This list is for illustrative purposes only and does not guarantee a grade.

40%-49%	Provide a physics simulation computation framework that can perform following tasks: • Render multiple 3D objects • Load simulation parameters as a text file • Correct computation of rigid body dynamics with linear motion and Euler integration • Have the physics engine as a separate system to the rendering engine
50-59%	In addition to above functionalities, your physics simulation framework can perform: • Collision detections • Impulse based collision responses • Rigid body dynamics both linear and angular motions • Good coding standard and physics engine structure
60-69%	Provide a physics simulation computation framework that is capable of: • Handling multiple object collision detection and response using both broad and narrow phase detections • More numerical integration algorithms tested and evaluated with rigid body dynamics • Good design and implementation of simulation scenario
70%- 79%	Provide a physics simulation computation framework that is capable of: • Advanced collision with spatial partitioning • More advanced numerical integration algorithms tested and evaluated with rigid body dynamics • For each object, attempt at handling of multiple contact points • Excellent and creative simulation demo with game features
80%+	• Successful attempt at LCP • Attempt at advanced physics simulation beyond rigid body dynamics (particle simulation, cloth simulation, hair simulation etc.) • Excellent and creative simulation demo with game features

For any questions, please ask the unit leader: Prof Wen Tang.

The Deliverables:
You must submit the following items for assessment:

This is assignment is a DIGITAL SUBMISSION. Submit a compressed **zip.** Submissions must be under 100 Megabytes unless you have good reason and permission is granted prior via email. Your zip file must contain the following:

- **Code** - A Windows **executable** (compiled in release and debug modes) with any required asset files, libraries and **complete source code with appropriate build files for compiling**.
- **A video trailer of your physics simulation.** Maximum HD (1080), 10s- 20s duration, must run on VLC in the lab computers
- **Report** - A PDF document containing a 1000 word report including:
    - An overview of your project and an initial specification of what it is supposed to do
    - Any research you undertook whilst writing the program (including looking at what work others have done in this area), and details of mathematical solutions or algorithms you used. Try to use references here.
    - A high-level description of your program describing how it works and justifying the design choices you made. UML and other diagrams would be useful here.

- An analysis of your physics simulation performance, key features of your physics engine, identifying potential strengths and weaknesses, concluding with proposals for future improvements.

- **User Guide** – Just a simple text file called README.txt in your project directory to explain how to build, run and use your program. Any on-screen instructions in the program itself will also be very helpful.

The Submission Deadline(s):
You must submit your work by the following deadline(s):

12:30 pm on Friday 11th January 2019

The Marking Scheme:
Your assignment will be assessed using the following mark scheme criteria:

Element	Contribution
Program	60%
Code	10%
Report	30%

Program Deliverable	**0% - 39%**	**40% - 49%**	**50% - 59%**	**60% - 69%**	**70% - 100%**
Design quality (15%)	Little evidence of purposeful design. Messy code structure.	Basic design and structure, low complexity. No clear separation between physics data and visualisation parts. Some evidence of using of appropriate design techniques, e.g object oriented techniques	Demonstration of good design and structure, medium complexity. Clear separation between physics data and visualisation parts. Some evidence of using of appropriate design techniques, e.g object oriented techniques	Demonstration of good design and structure, medium to high complexity. Clear separation between physics data and visualisation parts. Evidence of using of appropriate design techniques, e.g object oriented techniques and modular approaches	Demonstration of good design and structure, high complexity. Clear separation between physics data and visualisation parts. Evidence of using of appropriate design techniques, e.g object oriented techniques and modular/component based approaches
Underlying physics theory (15%)	Errors in physics equations. Incomplete physics computation.	Potentially wide range of bugs. Physics simulation may be with some errors but basic physics equations and numerical integration are correctly implemented.	Program performs rigid body simulation in real-time with multiple objects in 3D. Implementation of linear and angular dynamics	Physics engine computes both linear and angular motions with efficient collision detections and responses.	Program performs rigid body simulation task extremely well with a full set of very advanced features and extensions with very few bugs.
Implementation quality (15%)	Code does not compile / program does not run	Program runs, loads and displays one or more objects.	Evidence of attempts at some minor advanced features or extensions, with some bugs, such as realistic collision responses. Simple rigid body simulation scenario well designed.	Program performs rigid body simulation task well with a good range of advanced features and extensions, but with a few bugs. Good design of the simulation scenario and some user interactions involved.	Physics engine computes both linear and angular motions with efficient collision detections and responses. Excellent and creative design of the simulation scenario via a simple game demo etc.
Performance testing (15%)	No testing on performance metrics e.g. fps	Testing on performance metrics e.g. fps	Testing on performance metrics e.g. fps, multiple objects	Testing on performance metrics e.g. fps, multiple object collisions	A range of testing on performance metrics e.g. fps, multiple object collisions and responses

Code Deliverable	**0% - 39%**	**40% - 49%**	**50% - 59%**	**60% - 69%**	**70% - 100%**
Adherence to coding standards **5%***	No attempt at following coding standards.	Some attempt at following coding standards with many mistakes.	Acceptable attempt at following coding standards, with some mistakes.	Good attempt at following coding standards with few mistakes.	Excellent adherence to coding standards.

Quality, Readability, Comments 5%	No real code structure or attempts at organisation. Poor code with many errors. No comments in code.	Poor code quality with some clear errors, badly formatted, difficult to read, poor attempts at structure. Minor attempt at code comments.	Acceptable code quality with some errors, mostly readable, some structure and organisation, but inconsistencies in style. Some meaningful code comments.	Good code quality, readable and consistent in style, with a clearly defined structure and good attempt at design. Good and clear comments at appropriate places which enhances code readability.	Excellent code quality, clearly designed structure, well organised and written to a consistent standard. Excellent code comments, clearly enhancing code readability, written to a standard for documentation (e.g. Doxygen).

Report Deliverable	**0% - 39%**	**40% - 49%**	**50% - 59%**	**60% - 69%**	**70% - 100%**
*Introduction, overview and specification **5%***	No real project introduction or overview. No description of what the program is supposed to do.	Little understanding of project provided in overview. Unclear description of program.	Some understanding of project in overview and introduction. Brief but clear description of program.	Good understanding of project in overview and introduction, with an initial specification.	Excellent understanding of project in overview and introduction, showing contextual understanding and detailed specification.
*Research, algorithmic solutions, Problem Solving **5%***	No real evidence of research of algorithmic understanding. No documentation of problem solving.	Little evidence of research or some attempts at algorithms or problem solving.	Some relevant research with multiple sources and mostly correct algorithms.	Good evidence of research with survey of several relevant sources, with correct algorithmic formulations for fully described tasks	Excellent and comprehensive literature review of relevant research and correct algorithmic formulations of complex tasks
*Program design **15%***	No real design of program, no evidence of how it works.	Brief design of program, providing some description of how some of it works.	Adequate design of program providing clear description of how most of it works. May include some attempt at diagrams.	Good design clearly describing program and how it all works, some justifications of design choices. May include some useful diagrams.	Excellent design clearly describing all aspects of program, how it works, with analysis and justification of design choices, with a number of diagrams.
*Analysis and conclusion **5%***	No real analysis or conclusion.	Brief project analysis or conclusion. Mostly discussing experiences.	Project analysis identifying some strengths and weaknesses; brief conclusion.	Clear analysis showing strengths and weaknesses and identifying their causes; good conclusion identifying potential future work.	Excellent project analysis comprehensively identifying strengths and weaknesses together with their causes and potential solutions; excellent conclusion with clear roadmap for future work.

**Note on adherence to coding standards: you must use the coding standard introduced in your L4 programming units*

The Learning Outcomes:

This assignment will assess the following ILOs:

1. Demonstrate the ability to create a rigid body simulation program;
2. Compare and categorise implementations of collision detection algorithms with rigid body simulations;
3. Show an understanding of numerical integration algorithms and can implement Euler integrations;
4. Demonstrate knowledge of the state of the art physics simulation algorithms (i.e. soft body simulation; fluid simulation) and real-time approaches of the algorithms for interactive applications, such as games.

Confirmation that this assignment assesses the relevant ILOs:
Yes

ASSIGNMENT GUIDANCE NOTES – Academic Year 2018-2019

You must keep a copy of your assignment – the university will not take responsibility for lost assignments. Please make sure you back up your work carefully.

Submission Deadlines:

All Written assignments must be submitted before 12:30pm on the date due – unless otherwise stated on the assignment brief.

Electronic submission time will be 12:30pm on the due date following the above assignment detail, note this deadline is the **time for the upload of the assignment to be completed**, you are advised to begin submission AT LEAST 1 hour before.

For submission of physical assignments/artefacts/USB memory sticks etc, the assignment submission box for Creative Technology can be found on the first floor of Christchurch House, in the corridor near C114. Physical assignments must be submitted in the correct submission box before 12:30pm on the date stated on the assignment brief.

Late Submissions:

Please note that as per the **Standard Assessment Regulations** for any coursework that is submitted within 72 hours after the deadline, the maximum mark that can be awarded is 40%.

Please note that **the 72 hour Late Submission regulation only applies to the first submission of a given assignment** – and **does not apply to resubmissions**, unless they are being undertaken as a first attempt due to approved exceptional circumstances.

The 72 hour Late Submission regulation only applies to coursework submissions and does not apply to examinations.

If coursework is submitted **more than 72 hours after the deadline, a mark of zero (0%)** will be awarded.

Capped assignments will be considered by the Assessment Board and cannot be retrospectively uncapped by Academic Staff.

Exceptional Circumstances:

If you have any valid **exceptional circumstances** which mean that you cannot meet an assignment submission deadline and you wish to request an extension, you will need to complete and submit the **Exceptional Circumstances Form** for consideration to your Programme Support Officer (based in C114) together with appropriate supporting evidence (e.g, GP note) **before the coursework deadline**.

Further details on the procedure and the exceptional circumstances form can be found on **Brightspace** or via the link below:

https://www1.bournemouth.ac.uk/students/help-advice/looking-support/exceptional-circumstances

For further guidance on exceptional circumstances please see your Programme Leader or PSO.

Avoiding Plagiarism:

Plagiarism is the act of copying the work or ideas of others without proper acknowledgement of this work. Plagiarism also includes self-plagiarism or duplication: the inclusion in coursework, or a dissertation, or project, of any material which is identical or substantially similar to material which has already been submitted for any other individual assessment within the University or elsewhere.

Avoiding plagiarism is best achieved through the use of proper academic referencing and minimising direct quotations (i.e. re-write others' ideas in your own words, but still provide the reference of where these ideas came from). Further information can be found via the following links:

http://libguides.bournemouth.ac.uk/bu-referencing-harvard-style

https://www1.bournemouth.ac.uk/discover/library/using-library/how-guides/how-avoid-academic-offences

Further Information related to assessment can be found with the SciTech Faculty area of Brightspace, or via the following link:

https://brightspace.bournemouth.ac.uk/d2l/le/content/6633/viewContent/62565/View

Accessing Learning Support:

Student with Additional Learning needs are advised to contact the Learning Support team. Further details can be found via the following link:

www.bournemouth.ac.uk/als