University of the West of England
BRISTOL

MODULAR PROGRAMME

COURSEWORK ASSESSMENT SPECIFICATION

Module Details

Module Code UFCFG5-30-2	**Run** 14SEP/FR/JUN15/1	**Module Title** SIMULATED WORLDS
Module Leader Simon Scarle	**Module Coordinator**	**Module Tutors**
Component and Element Number B: CW2		**Weighting: (% of the Module's assessment)** 45
Element Description PRACTICAL ASSIGNMENT - IMPLEMENTATION (Practical Assignment - Implementation)		**Total Assignment time** **50 hours**

Dates

Date Issued to Students	**Date to be Returned to Students**
Submission Place **Coursework Hub** (Level 1 A Block Underpass)	**Submission Date** 16/04/2015
	Submission Time **2.00 pm**

Deliverables

- USB Memory stick containing all required files for implementation
- Digital copy of a development report
- Logbook of progress discussions

Module Leader Signature

SIMON SCARLE

Assignment Specification – Component B1

UFCFG5-30-2 – Simulated Worlds

Information

This module will be assessed in two components.

Component A will be a 2 hour exam on the mathematical and theoretical aspects of this module, and will comprise 25% of the overall mark for the module. This will be sat in the pre-summer exam period.

The remaining 75% will be the practical **Component B**. 40% of which will be a Level design task using UDK to be handed in before Christmas and marked by a demonstration of the level created to the module leader, and 60% will be an implementation in C++ of some component of a simulation of your choice. This will be handed in near the end of the academic year, and will also be in-part marked by a demonstration of the simulation created.

Part of the marks for each practical section will be based on progress reports given in person to the module team.

Component B1

A brief is given below of the implementation task to be completed. The majority of the second semester tutorial sessions for SW will be given over to time to work on this task. At these sessions you are expected to give at least three progress reports to the tutor, which need to be signed off on a logbook form supplied with this specification. This will then form one of the deliverables for this component.

The other deliverables will be a USB memory stick containing all files for your implementation, and a digital copy of a development report.

During the week following the hand in, you will be expected to give a demonstration of your implementation.

Progress Report Logbook

Each student will keep a progress report logbook throughout the module. Students are expected to discuss their work and progress with a course tutor at least every two weeks.

The role of these discussions is to keep work on track, keep the project in scope and to set reasonable targets for subsequent weeks.

Each student should keep a progress report logbook sheet which the course tutor will sign after each discussion. After completion of a brief summary of what discussed and the agreed targets.

A total of 3 progress discussions must be logged over the course of the semester.

A logbook template has been provided on Blackboard, to be completed and signed by the tutor after each progress discussion.

Submission of the completed logbook sheet should be as part of the main level design task submission to the coursework hub by the date specified on the header sheet.

The logbook MUST use the template provided on Blackboard.

Marks for this part of the assignment are awarded on a pass/fail basis. Any student who has satisfactorily completed and signed off all 3 progress discussions in a timely manner will be awarded the full 10%.

Demo Sessions

Part of the marking for this implementation task will be ascertained via a demonstration of the implementation to take place the week after the hand in date.

A list of available demo slots will be published on the door of 2Q20 the week of the hand in.

If you cannot attend any of the slots available, you must contact the module leader before the hand in date to make alternative arrangements.

Brief: Implementation Task

Brief Description

Taking as inspiration one of four suggested simulations, you shall produce a tech demo style implementation of that simulation suitable for inclusion into a wider game engine. This should be implemented in DirectX possibly using the supplied code base. Whilst the simulation might not take place in 3D, any visualization of it should do. You will also be required to produce a development report which outlines your research into developing this implementation and how you went about producing it.

Detailed Breakdown

Completed Progress Reports (10%) As highlighted above an all or nothing 10% for 3 completed progress reports.

Research (25%) The introductory part of the report should detail background research and outline the techniques that are expected to be used to produce the simulation, as well as highlighting competing techniques and rationale on why you choose to use the ones you did. All sources should be suitably referenced. Students should reflect on the expected suitability of the simulation in a games production pipeline. 5% of the marks for this will be based on the completion of the library service's library skills workbook activity.

Implementation (35%) Your code should ideally be structured as though it is part of a wider game engine and have clear "handles" where by it could be extended and/or customised. Essentially, the parameters of the simulation should be accessible to change how it behaves. It should also be clearly commented and a suitable coding standard employed. The principle section of the report will constitute a discussion of this implementation and will be 10% of the marks for this section. This should also discuss the class structure used and how the selected techniques were implemented in code.

Debugging Information / HUD (5%) A rudimentary HUD system is expected which details currents parameters of the underlying simulation and allows them to be altered on the fly. Additional graphical components which highlight the underlying functions of the simulation should also be toggle-able.

Shading & Creation of Assets (10%) It is expected that all assets used to display your simulation have their vertices created in code, and that a suitable selection of shaders are implemented which demonstrate different effects or display debugging information.

Critical Evaluation (10%) This final section of the report should be used to showcase the understanding gained throughout the process of your project. It should discuss any flaws in the implementation of your simulation and they could be potentially be improved, and also highlight particularly strong aspects of your implementation. A discussion as to its suitability for use in a games development is also expected.

Demo (5%) Reflection and demonstration of an understanding of the functioning of the simulation implemented, as shown by the demo.

Grading Criteria

40%	Student shows a basic understanding of the material presented in lectures. Code goes some way towards solving the given problems, but there is little evidence that the student is capable of writing structured code and use language features and data structures appropriately. There is neither clear factoring of code nor any indication of how the code could be extended. Discussions show little or no evidence of being able to relate theory to practice, and there is no evidence of reading outside of the material presented in lectures. There is some evidence of the student attempting to comment the code appropriately. The language used in the report is not always appropriate for an academic setting.
50%	Student shows a developing understanding of the material presented in lectures. Code generally solves the given problems, and there is some evidence that the student is capable of writing structured code and use language features and data structures appropriately. There are the beginnings of appropriate code factoring and some indication of how the code could be extended. Discussions show some evidence of being able to relate theory to practice, and there is some reference to background reading. The student largely comments the code appropriately. The language used in the report is mostly appropriate for an academic setting.
60%	Student shows a good understanding of the material presented in lectures. Code solves the given problems, and is generally well structured, using language features and data structures appropriately. Code has been factored to collect function efficiently in the class structure and clear handles are present to extend beyond what has been implemented. Discussions show that the student is clearly able to relate theory to practice, and there is clear evidence of reading outside of the material presented in lectures which is mostly referenced. The student comments the code in an appropriate style. The language used in the report is mostly appropriate for an academic setting.
70%+	Student shows an excellent understanding of the material presented in the lectures. Code solves the given problems and is very well structured, using language features and data structures clearly appropriate within the given context. Code has been factored to collect function efficiently in the class structure and clear handles are present to extend beyond what has been implemented. Discussions show that the student has no problems relating theory to practice, with clear references to a range of suitable material outside of that presented in lectures. The student comments the code in an appropriate style. The language used is appropriate for an academic setting.