University of Huddersfield
School of Computing and Engineering
Department of Informatics
Assignment Specification

Module Title	**Game Engine Architecture**
Code	**CIT2213**
Assignment Title	**Portfolio submission 1**
Weighting	**40%**
Module Tutor	**Dr. Minsi Chen**

Assignment Aims

This assignment aims to give you an opportunity to demonstrate the following learning outcomes (see the module specification for further details)

Learning outcome (1)
: Explain approaches to the design and implementation of non-graphics game engines and game engine components using industry standard programming languages and tools

Learning outcome (3)
: Employ industry standard tools and techniques to design and implement non-graphics game engine components and tools, illustrated by a working game utilizing those components

Learning outcome (4)
: Produce clear documentation, and explain choices made between competing approaches to design and implementation of the non-graphics aspects of a game engine

Problem Statement and Brief

Commercial game engines including Unreal Engine 4, CryEngine and Unity3D are designed to provide an encompassing set of general-purpose features to support game development.

For this assignment, you are required to build a *wheeled vehicle* game in the Unreal Engine 4. You are free to choose a game genre to develop, for examples:

- Driving simulation
- Arcade racing
- Battle arena driving

Irrespective to your choice of game genre, your game should consist of at least the following elements:

- Adequate physical handling of the player vehicle - ***Physics***
- At least one computer-controlled vehicle – ***AI/Pathfinding***
- A general overarching mechanics, e.g. checkpoint racing, time attack, combat and so forth – ***Game State***
- A test level for evaluating your game mechanics, e.g. a closed arena, a racing circuit

Please note: **You have only a limited amount of time to complete the assignment so you should set a realistic set of targets.**

Your game must be programmed in C/C++ using Unreal Engine 4. Pure BluePrint implementation will not be accepted and graded, although hybrid usage is permitted where necessary. Your code should be commented appropriately and you should adhere to the coding standard specified by the game engine documentation. These include consistent and self-explanatory naming convention for both variables and methods.

Formative Checkpoints

There are two formative checkpoints where you will discuss the progress of your work with your tutor.

- **Checkpoint 1 - Week 7**: At this checkpoint, you should already have a concrete design and specification of your work. You should be familiar with the core of Unreal Engine and how to apply them to your work.
- **Checkpoint 2 – Week 11**: At this checkpoint, your work ideally should be in a completed state, i.e. they should be fully functional but perhaps require more refinement.

Please note, you must still take part in this formative checkpoint even if you think you are behind your work. You will receive appropriate guidance from your tutor.

Deliverables and Submission Process:

The deadline for submitting your work is **23:59 Friday 14/12/2018**. You will submit a single ZIP file containing the followings to the designated submission point on Brightspace:

- The entire project folder containing your game
- A PowerPoint presentation (no more than 10 slides) concisely explaining the implementation of mechanics and features.

The submitted ZIP file must be named using the format "*STUDENTID_assignment1.zip".* Substitute *STUDENTID* with your own student number.

Please note, your submitted project must build and run on the machines in our departmental computer labs. It is recommended that you check this on a regular basis.

Instructions for handing your work in will be available on Brightspace

Level of Collaboration

This is an individual assignment, collaboration is NOT permitted.

Assessment criteria

This assignment is worth 40% of the total module grade.

Explain approaches to the design and implementation of non-graphics game engines and game engine components using industry standard programming languages and tools (10 marks)	Employ industry standard tools and techniques to design and implement non-graphics game engine components and tools, illustrated by a working game utilizing those components (20 marks)	Produce clear documentation, and explain choices made between competing approaches to design and implementation of the non-graphics aspects of a game engine (10 marks)	Mark band
Excellent explanations, no weaknesses. Significant, very original thinking on the topic.	Effective use of industry standard tools and techniques, and extensive innovative use of appropriate advanced tools and techniques	In addition to the criteria for "89 - 80": Puts forward a convincing original argument to support some choices	100 – 90
Excellent explanations, no weaknesses. Some original thinking on the topic.	Effective use of industry standard tools and techniques, and some innovative use of appropriate advanced tools and techniques	In addition to the criteria for "79 - 70": Critically reviews relevant sources to when evaluating the options and justifying all choices	89 – 80
Excellent explanations, no significant weaknesses.	Effective use of industry standard tools and techniques, and some appropriate tools and techniques	Documentation is imaginatively presented resulting in clarity of message and information	79 – 70
Good explanations Weaknesses in one area compensated by good understanding elsewhere	Effective use of industry standard tools and techniques	Documentation is carefully structured with clear message	69 – 60
Reasonable explanations; some weaknesses or omissions.	Use of industry standard tools and techniques is mostly effective	Documentation included is relevant to topic and has been structured.	59 – 50
Reasonable explanations one significant misunderstanding	Some weaknesses in the use of the industry standard tools and techniques compensated	Documentation presented is relevant but lacks structure	49 – 40

	by effective use elsewhere		
Poor explanations in most areas OR More than one significant misunderstanding	Some weaknesses in the use of the industry standard tools and techniques	Documentation is structured but there are gaps, or some is not relevant	39 – 30
Poor explanations in most areas; some evidence that the topic is misunderstood.	Some of the industry standard tools and techniques are not used or used incorrectly; others used effectively	Documentation is incomplete or irrelevant and lacks structure	29 – 20
Significant evidence that the topic is misunderstood.	Some of the industry standard tools and techniques are not used or used incorrectly	Documentation is incomplete or irrelevant and it is presented in a disorganised manner	19 – 10
Has misunderstood the topic.	Little evidence that the industry standard tools and techniques can be used effectively	Little evidence of an attempt to document the solution	9 – 0

Additional Information for Clarification

- You are reminded that it is your responsibility to read and understand the University regulations regarding assessment. Your attention is directed to the regulations regarding assessment in the Student's Handbook of Regulations (available from the University website), and in particular to its guidance on academic misconduct and plagiarism.
- ***A note on using other people's program code in your work:***
  It is standard practice amongst professional developers to use pre-existing code in their programs. Correctly using such code is an important skill: you will not be marked down for using such code appropriately in your solutions. The need to properly reference such "quotations" is obvious in an academic context; however it is increasingly important in commercial contexts too, due to increased copyright and patent protection of code. This means that any code taken from a text book, or an Internet site, should be surrounded by comments that clearly indicate:
    - Where the quoted code starts and ends.
    - The original source of the quoted code, using the University's standard referencing system.
    - A brief description of any changes, simplifications or improvements you have made to the quoted code.

  Warning: It can sometimes be harder to adapt someone else's code to meet your needs than to code something from scratch. You should also be aware that there are some very bad code samples on the Web. Incorporating bad code in your own work demonstrates a lack of understanding of good coding practice, and is an inappropriate use of pre-existing code.

Hand out date: 04 October 2018	**Hand in date:** 14 December 2018