# Staffordshire University logoSchool of Computing and Digital Technologies

# Module Code: COSE60587

# Module Title: Advanced Graphics and Real Time Rendering

# Module Leader: David White

# Title: Human Motion on Terrain

# Year: 2019-2020 Level 6, Semester 1

# Weighting: 35% of module marks

**Submission**

**Hand-in: 1st May 2020 @ 3.30pm**

### Module Learning Outcomes for This Assignment

|  |  |
| --- | --- |
| 1. CRITICALLY EVALUATE HIGH-LEVEL DATA STRUCTURES AND ALGORITHMS IN TERMS OF THEIR EFFICIENCY AND PERFORMANCE IN DELIVERY OF ADVANCED GRAPHICAL REQUIREMENTS. | Reflection |
| 1. DOCUMENT THE DESIGN, CONSTRUCTION AND OPTIMISATION OF 3D GRAPHICS AND GAME-RELATED ARTEFACTS INCORPORATING PROCESSED IMAGES | Communication |
| 1. SYSTEMATICALLY UNDERSTAND AND EXPLAIN REAL-TIME RENDERING AND ANIMATION CONCEPTS, TECHNIQUES AND ALGORITHMS. | Knowledge & Understanding |
| 1. RESEARCH AND APPLY SUITABLE TECHNIQUES FOR ADVANCED 3D SIMULATION. | Application |

### Assignment Process Requirements

This assignment is worth 35% of the module mark overall.

You are required to develop an application incorporating graphical, animation and game concepts taught in this module. You must submit your source code, executable program and a report by the deadline given above, and also demonstrate your work in a video.

**Video Format**

The video should be in avi or mp4 format and playable on all common video file players (such as VLC).

During the video demonstration you should cover each of the points *in order* in the marking scheme – if you haven’t implemented a feature, then state this. You should use voice over or subtitles. You do not need to discuss your report in your video but can refer to this if required.

It is your responsibility to find a screen recorder to capture video to as well as audio / subtitles, but OBS Studio might be one (free) option:

<https://obsproject.com/>

More details of the assignment task are given in the following pages.

# Scenario

‘Open World Games’ is a local small games company who specialize in creating bespoke games for a variety of game publishers. They program games for the PC platform. They have developed several genres of 3D games to run on these platforms.

All the games they develop share common features, they work only on Open World style games and frequently have the need to generate and render 3D terrain, with characters animating and moving realistically over this terrain.

They are concerned with the underlying engine efficiency and design, and require you to create a flexible and efficient program that is capable of generating terrain using a number of different methods, as well as rendering it as efficiently as possible given the time constraints on the project deadline.

# Task Requirements

Your task is in two parts.

**Part 1**

You should create a version of the engine specified above which contains the following:

* The engine should be written in C++ using an API of your choice, but must be DirectX 11, 12, OpenGL 2 (or above) or Vulcan
* A graphics engine which controls the loading and rendering of an animated 3d humanoid
* A mechanism for loading terrain data which can be used to represent the game world
* A component to allow the humanoid character to ‘walk’ on the terrain
* The walking motion may be a simple cyclical animated sequence which has no interaction component to effect changes based on terrain features but there will be additional marks for
  1. Implementing adaptive step sequences to follow terrain
  2. Inverse kinematics
  3. Other relevant technique
* Advanced topics such as Voxelization, Level of Detail, etc

**Part 2**

You must submit a written report that accompanies your implementation.

The report (excluding formulas, tables, diagrams, appendices and references) of around 2000 words.

It must include the following items:

* An introduction.
* An explanation for the API you chose, a **critical analysis** and a **critical discussion** of an alternative API
* A block-diagram of the graphics pipeline, or a class diagram, depicting the architecture of the software that you have implemented. The diagram must include the names of key functions that make up the software components implemented by you; the diagram must not show components that you have not implemented.

* A description of the data structures, algorithms or techniques that you have implemented, together with justifications for their choice. Where applicable, the description must include the mathematical formulas that underpin your implementation. You must analyse and justify any formula simplification or algorithm simplification that you made.
* A critical appraisal of the functionality and performance of your software. The appraisal should discuss issues such as performance bottlenecks and computational costs, together with implemented or potential solutions. You must also discuss the positive and negative effect(s) of any formula or algorithm simplification that you made.
* A conclusion.
* A list of bibliographic references to some authoritative computer graphics publications that you consulted for your assignment. You should use the Harvard system of referencing.

**Marking scheme**

In each case marks are awarded for quality as well as implementation. You must be able to explain code correctly, answer questions correctly, and have created a flawless implementation in order to receive full marks.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Marks** | **Criteria** |
|  | System Architecture and code style | 10 | The architecture of the system implemented should incorporate OO design considerations and be programmed with extension and change in mind. |
| Terrain | Terrain  construction | 5 | A terrain rendered from a proprietary format input data file, and rendered correctly |
| Advanced Terrain generation | 10 | The generated terrain has features such as caves and overhangs. Voxels can be used. |
| LOD terrain | 10 | The terrain renders at different levels of resolution for viewing parameters. This must be demonstrated with a camera and visible and effective LOD. |
| Procedural  Terrain generation | 10 | The terrain is generated using diamond square, fault line, and one other algorithm (2.5 marks each) |
| Animation | Human skeletal structure | 10 | Simple skeletal structure consisting of jointed arms and legs using an appropriate data structure for traversing the skeleton.  This must be hand coded – i.e. all code is written by you, you may not use a pre-defined animation code library. |
| Walking animation | 10 | An animated sequence which either reads (a pre-computed set) or computes a set of joint angle and position changes to simulate walking motion, and renders the skin of the character  This must be hand coded – i.e. all code is written by you, you may not use a pre-defined animation code library. |
| Loading a humanoid character | 5 | Loading a skinned model of humanoid character from a suitably formatted file |
| Adaptive motion sequence | 10 | The ‘walking’ motion is adapted to the terrain features such that gait parameters are dynamically calculated |
| Report | Report | 20 | The report as discussed in the assignment, written in a suitable style with appropriate diagrams and explanation for each of the sections specified.  The selection of techniques is reflected upon through referenced researched. |

You should also make yourself aware of University regulations regarding ethics, plagiarism, late hand-ins and other issues relating to assessment and your general conduct.