DUBLIN INSTITUTE OF TECHNOLOGY

DT211C BSc. (Honours) Degree in Computer Science (Infrastructure)

Year 4

WINTER EXAMINATIONS 2016/2017

GAMES ENGINES 1 [CMPU4030]

DR. CATHY ENNIS
DR. DEIRDRE LILLIS
MR. ALAN FAHEY

MONDAY 16TH JANUARY 9.30 A.M. – 11.30 A.M.

DURATION 2 HOURS

ANSWER **QUESTION 1** AND **TWO OTHER QUESTIONS.**

QUESTION 1 IS WORTH 40 MARKS. ALL OTHER QUESTIONS ARE WORTH 30 MARKS EACH. YOU MAY USE CODE, PSEUDOCODE, MATHEMATICAL EQUATIONS AND DETAILED ENGLISH DESCRIPTIONS IN YOUR ANSWERS.

1. The two models drawn in the scene in Figure 1 are encapsulated into C++ classes that have the members given in Figure 2. Assume that the model on the left is represented as an object called `ship1` and that the model on the right is represented as an object called `ship2` With respect to Figure 1 and Figure 2 answer the following:

(a) Construct an efficient algorithm for calculating if `ship2` is in front of or behind `ship1`.

(8 marks)

(b) `ship1` needs to know if `ship2` is less than 10 units away and also inside of its 45 degree field of view (FOV). Construct an algorithm for evaluating this.

(10 marks)

(c) Construct an efficient algorithm for calculating the world matrix for `ship2`, assuming it must always be directed towards `ship1`.

(12 marks)

(d) Describe the function of the view and projection matrices and how to calculate them in C++.

(10 marks)

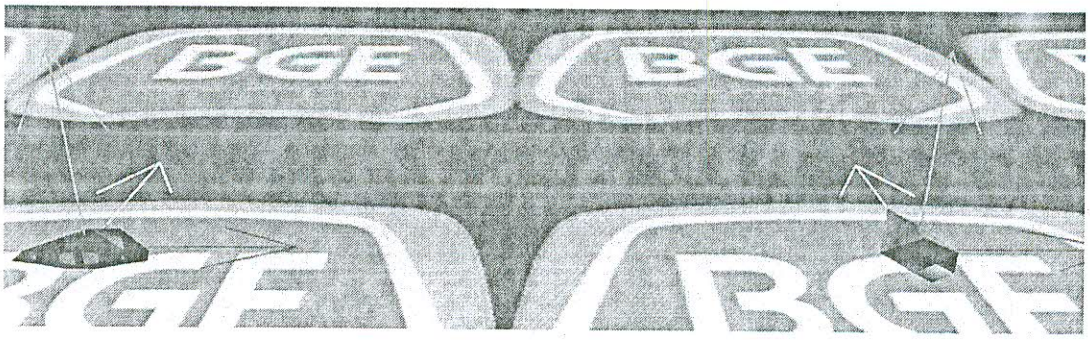

Figure 1

```
glm::vec3 position;
glm::vec3 look;
glm::vec3 up;
glm::vec3 right;
glm::vec3 scale;
glm::mat4 world;
glm::quat orientation;
```

Figure 2

2. **(a)** Describe the following Visual Studio components and how they can be used in Game Programming.

(i) shared_ptr

(4 marks)

(ii) STL

(4 marks)

(b) Describe the operation of a component based games engine architecture. In your answer, discuss, using an example, the advantage of this design pattern over object-oriented game engine architectures.

(12 marks)

(c) Propose an approach for implementing camera yaw and pitch in a FPS game.

(10 marks)

3. **(a)** What is the difference between force and torque? Write the code for a function that calculates the resulting torque when force is applied to a point on a rigid body.

(7 marks)

(b) Distinguish between static, dynamic and kinematic rigid bodies in the Bullet physics engine. Provide examples of these types of rigid bodies from a commercial game with which you are familiar.

(8 marks)

(c) Describe the main stages in a real time graphics pipeline, using examples to describe what happens at each stage.

(15 marks)

4. **(a)** Compare Euler angle sequences with quaternions as a means of representing the orientation of an object in a 3D game engine. Include in your answer a description and a diagram illustrating the problem of Gimbal lock.

(15 marks)

(b) A character in a 3D computer game encapsulates the member variables given in **Error! Reference source not found.**. The `turnRate` variable is the rate at which the character rotates and is given in radians per second. The characters model points down the positive X axis when no rotation is applied. Write the implementation for the characters `Update` function so that it gradually turns to face a character located at the member variable `endTarget`. It should rotate at its `turnRate`.

(15 marks)

```
glm::vec3 position;
glm::quat orientation;
float turnRate;
glm::vec3 initialTarget;
glm::vec3 endTarget;
```

Figure 3

Down with Paper

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF: DT211C/4 and DT228/4 VENUE: _____

SUBJECT: Games Engines 1 [CMPU4030]

DATE: Monday 16th January 2017

TIME: 9.30 a.m. – 11.30 a.m.

SIGNED: Dave Carroll

INSTRUCTIONS: Q 4 (b): The error message 'Error! Reference source not found' should refer to Figure 3, as shown below.

4. (b) A character in a 3D computer game encapsulates the member variables given in **Figure 3**. The `turnRate` variable is the rate at which the character rotates and is given in radians per second. The characters model points down the positive X axis when no rotation is applied. Write the implementation for the characters `Update` function so that it gradually turns to face a character located at the member variable `endTarget`. It should rotate at its `turnRate`.

(15 marks

```
glm::vec3 position;
glm::quat orientation;
float turnRate;
glm::vec3 initialTarget;
glm::vec3 endTarget;
```

Figure 3