

UNIVERSITY OF ESSEX

Undergraduate Examinations 2011

GAMES CONSOLE PROGRAMMING

Time allowed: **TWO** hours

Candidates must answer **ALL** questions.

The paper consists of **FOUR** questions.

The questions are of equal weight.

The percentages shown in brackets provide an indication of the proportion of the total marks for the **PAPER** which will be allocated.

**Please do not leave your seat unless you are given permission by an invigilator.
Do not communicate in any way with any other candidate in the examination room.
Do not open the question paper until told to do so.
All answers must be written in the answer book(s) provided.
All rough work must be written in the answer book(s) provided. A line should be drawn through any rough work to indicate to the examiner that it is not part of the work to be marked.
At the end of the examination, remain seated until your answer book(s) have been collected and you have been told you may leave.**

Candidates must answer ALL questions.

Question 1

- (a) With the aid of a diagram briefly describe the standard XNA flow of control (i.e. from the start-up to the shutdown of a game, including the game loop). State which methods would normally perform the following tasks: [10%]
- (i) Checking for user input
 - (ii) Loading in a 3D model
- (b) With the aid of examples explain how the following C# / XNA features are used within 3D games programming: [8%]
- (i) Structs
 - (ii) Operator overloading
 - (iii) Explicit call by reference
 - (iv) SpriteBatch objects
- (c) Briefly describe the steps involved in creating a rectangle with an image tiled on it in a 2x2 arrangement using XNA. Include in your answer any XNA classes used and also the type of primitive used to draw the rectangle. [7%]

Question 2

- (a) When detecting collisions between two 3D objects, explain how bounding sphere calculations can be used hierarchically to provide reasonable efficiency and accuracy, and briefly describe the support that XNA offers for this approach in its `Model` class. [6%]
- (b) Briefly describe the strengths and weaknesses of the data structures listed below for storing game objects when performing collision detection. For each approach state the time complexity of probing the data structure to find whether any game objects contain a particular point in 3D space. [9%]
- (i) List
 - (ii) Grid
 - (iii) Binary Space Partition (BSP) Tree
- (c) Figure 1 below illustrates three lines in the order they are to be added to a BSP Tree. Construct a possible BSP Tree that would store these lines. Draw the tree, and also draw a new version of the line diagram that indicates where each line has been split (if at all) and labels each line segment with its corresponding tree node. Your diagram should also indicate the front normal vector chosen for each line. [10%]



**Figure 1: Line diagram to be represented in BSP Tree.
Lines should be added to the tree in the order 0, 1, 2.**

Question 3

- (a) This part of the question is about particle effects.
- (i) Give three examples of particle effects commonly used within 3D games. [3%]
 - (ii) Given a uniform random number generator, outline an algorithm for specifying the initial velocity of each particle in a spherical explosion effect (the observed explosion must not have any cuboid artefacts). Mention any XNA vector or matrix functions that would be used. [7%]
 - (iii) A particle effect may involve drawing thousands of primitives. Explain how you can ensure this is done efficiently in XNA. [4%]
- (b) This part of the question is about *billboarding* as used in 3D games.
- (i) Explain why billboarding may be used, and what is meant by cylindrical and spherical billboarding respectively. Give one example of how each may be used. [5%]
 - (ii) With the aid of snippets of C# / XNA code explain how you would apply the necessary vector and/or matrix transformations for the cylindrical billboarding of a single object. [6%]

Question 4

- (a) Using pseudo-code outline the A* (A-Star) algorithm. You may assume the existence of an efficient Priority Queue data structure. [10%]
- (b) State the criterion for an A* heuristic to be *admissible*. Given a grid-based 2D navigation map with eight-way connectivity and non-traversable grid-squares, give one example of an admissible heuristic and one example of an inadmissible heuristic. [5%]
- (c) When used for path-finding the A* algorithm is usually used to find a path between two points. Explain how it can be modified to find an optimal path that visits multiple points. [5%]
- (d) Describe how the A* algorithm can be applied to control the low-level actions of an agent which moves in continuous space according to a simple physics model. Briefly explain a potential problem with this approach, and a possible solution. [5%]

END OF PAPER CE318-6-AU