

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

BSc Computer Science

Year 4

SUMMER EXAMINATIONS 2006

GAMES PROGRAMMING

Mr. Bryan Duggan

Mr. Hugh McAtamney

Dr. Brendan O' Shea

Dr. K. Beckoum

25 May 2006 2.00-5.00

Time 3 Hours

Attempt any TWO Questions from Section A and any TWO questions from Section B

All questions carry equal marks

SECTION A

1. (a) What is meant by the following terms: *resolution*, *anti-aliasing*, and *bump mapping*. In your answer, describe the effect these settings have on a game's *image quality* and *frame rate*.

(5 marks)

- (b) The position of a character in a computer game is represented by a vector C . The character starts at position $(10\ 0\ 10)$. The character moves to a position **offset** from its start position by $(10\ 0\ 9)$. It then moves to an offset of $(-8\ 0\ -20)$ and finally moves to an offset of $(8\ 0\ -9)$. The character always faces the direction it is heading (a vector L). When it arrives at its final destination, it rotates to face an enemy E located at $(20\ 10\ 8)$.

Formulate an approach for calculating the angles the character must *yaw* and *pitch* in order to face the enemy, once it has reached its destination. In your answer include:

- i. An appropriate diagram to illustrate the problem. (4 marks)
- ii. A detailed description of a suitable approach. (8 marks)
- iii. Any mathematics/equations required. (3 marks)

Note it is **not** necessary to actually perform the calculations.

- (c) Assuming the character in part (b) is represented by a C++ class holding the data members listed in Figure 1 and is drawn using a mesh, write the DirectX code to calculate the *view matrix* required to *position* and *orientate* the character in a 3D world.

```
D3DXVECTOR3 _pos; \\ The entities position
D3DXVECTOR3 _up; \\ Up
D3DXVECTOR3 _look; \\ The direction the entity is heading
D3DXVECTOR3 _right; \\ A vector orthogonal to _up and _look
float _yaw; \\ The angle to yaw in radians
float _pitch; \\ The angle to pitch in radians
```

Figure 1

(5 marks)

2. (a) 3D games require *assets*. Describe what you understand by this term and describe a coding strategy for managing assets in a game engine. Include in your answer a simple example of how this is achieved in C++. (6 marks)

- (b) Figure 2 shows classes that exist in a computer game engine:

```
// Represents a rectangular wall
class Wall
{
    D3DXVECTOR3 _lowerBound;
    D3DXVECTOR3 _upperBound;
};

// Represents the player
class Player
{
    D3DXVECTOR3 _pos; // The players position in the world
};

// Represents the enemy
class Enemy
{
    D3DXVECTOR3 _look; // The direction the enemy is looking
    D3DXVECTOR3 _pos; // The enemy's position in the world
    World * _world; // The world
    float _fov; // the enemy's field of view in radians
};

class World
{
    std::vector<Wall*> _walls; // A vector of walls
    Player * _player;
    Enemy * _enemy;
};
```

Figure 2

Formulate an approach for detecting if the game character represented by the Enemy class can *see* the game character represented by the Player class. In your answer include:

- i. A diagram illustrating a possible scenario. (3 marks)
 - ii. A detailed description of any algorithms you propose implementing. (6 marks)
 - iii. Any new API's required with a brief description of the purpose of each new API. (4 marks)
 - iv. A critique of your approach. (1 marks)
- (c) Compare and contrast *physics based gameplay* with *scripted gameplay* using examples from commercial games. (5 marks)

3. (a) In relation to path finding in computer games, what is meant by a *heuristic*? Outline any 2 heuristics you are familiar with. Which of the heuristics you describe is likely to give the fastest execution times? Justify your answer.

(5 marks)

- (b) Figure 3 presents an abstraction of a game world, where *S* represents the starting position and *D* represents the destination of a game character. An *X* represents non traversable nodes.

| | | | | | | |
|--|--|---|---|---|---|--|
| | | | | | | |
| | | S | X | X | | |
| | | | X | | D | |
| | | | X | | | |
| | | | | | | |

Figure 3

Draw the state of the game world for the first 3 iterations of the A* algorithm using any **one** of the heuristics you outlined in part (a). In your answer include:

- The *f*, *g* and *h* scores for each node considered.
- Nodes that are expanded and nodes that are no longer to be considered.

(10 marks)

- (c) Compare and contrast pre-generated navigation graphs with graphs where nodes are added on the fly for pathfinding using the A* algorithm.

(5 marks)

- (d) Choose appropriate optimal data structures from the STL to hold the open and closed lists in an implementation of the A* algorithm in C++. Justify your answer.

(5 marks)

SECTION B

4. (a) “The *Atari Video Computer System (VCS)* was the most crucial computer game system in the history of computer games”
- Discuss this statement. Illustrate your answer using examples of both current and early generation gaming systems. (8 marks)
- (b) Summarise the 6 elements described in Jesper Juul’s Classic Game Definition. (6 marks)
- (c) What is *game balance*? Illustrate your answer with examples. (4 marks)
- (d) Differentiate between *player versus gameplay balance* and *gameplay versus gameplay balance*. (7 marks)
5. (a) Describe in detail how *visibility determination variables*, *attribute objects*, the *perception co-efficient* and *visibility* are used as part of the visual sensors in the Crytek engine’s AI perception system. (13 marks)
- (b) List three properties that can be manipulated in the Crytek Sandbox Editor to change the calculation of the alert status of a computer AI character. (3 marks)
- (c) What are Crytek *game entities*? How might you go about creating a game entity in Crytek? (5 marks)
- (d) What *visual metaphor* is used in the FarCry game to inform the player of the computer AI character’s alert status? Describe briefly how it works. (4 marks)
6. (a) The CryAI system uses a *non directed graph* as the basis of its spatial representation. Outline the *general properties* of this graph and discuss how the nodes it creates are used in the *triangulation* process. (10 marks)
- (b) In the context of the CryAI system, what is *beautification* and why is it used? (4 marks)
- (c) Differentiate between *tactical events* and *game events* in FarCry. (6 marks)
- (d) Identify the advantages of using the scripting language LUA over C++ in the creation of events and triggers. (5 marks)