DUBLIN INSTITUTE OF TECHNOLOGY

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

Year 4

DT228 BSc. (Honours) Degree in Computer Science

Year 4

SUMMER EXAMINATIONS 2015/2016

GAMES ENGINES 2 [CMPU4031]

DR. BRYAN DUGGAN
DR. DEIRDRE LILLIS
MR. ALAN FAHEY
MR. KEVIN FOLEY

FRIDAY 20TH MAY 4.00PM - 6.00PM

2 HOURS

INSTRUCTIONS TO CANDIDATES

ANSWER QUESTION 1 (COMPULSORY) AND ANY 2 FROM THE REMAINING QUESTIONS
QUESTION 1 IS WORTH 40 MARKS, THE REMAINING QUESTIONS ARE WORTH 30 MARKS EACH

Question 1

Figure 1 shows a screenshot from a Unity3D project. This scene depicts a dolphin wandering through the scene.

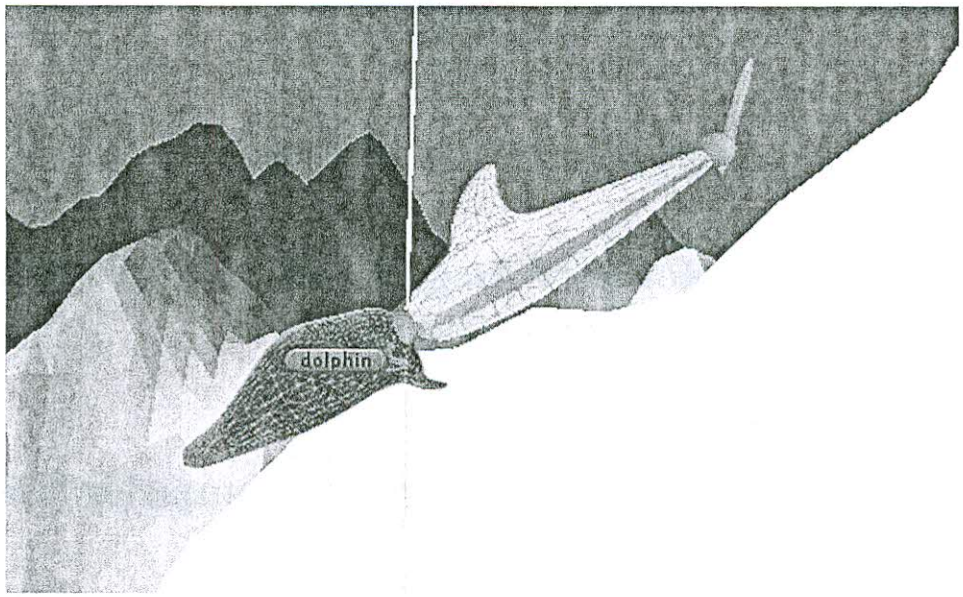

Figure 1

(a) How would you program the dolphin to wander purposefully? Include a diagram in your solution.

(13 marks)

(b) How would you constrain the dolphin to stay within range of its starting position?

(10 marks)

(c) How would you combine these two behaviours together so that constraining the dolphin's movement takes priority over wandering?

(13 marks)

(d) How is the generated force integrated into the boids state?

(4 marks)

Question 2

Figure 2 shows a screenshot from a Unity3D project that implements a boid with a *dynamic obstacle avoidance* behaviour.

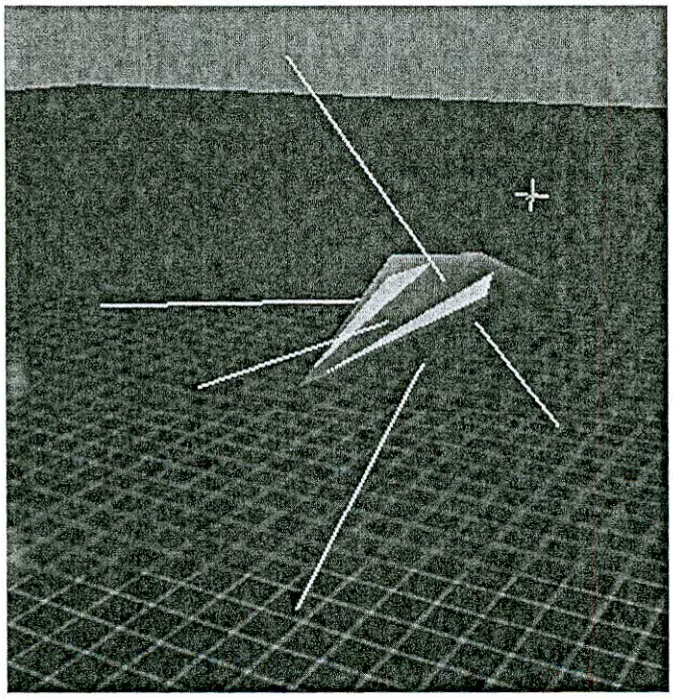

Figure 2

(a) Describe in detail how you would generate the feelers for the boid given in Figure 2 so that the feeler distance is proportional to the boid's speed.

(15 marks)

(b) How would you generate a force to steer the boid away from any obstacles in its path?

(15 marks)

Question 3

(a) Present pseudocode for the A* algorithm for calculating the lowest cost path between two nodes in a tile based navigation graph.

(10 marks)

(b) Figure 3 presents an abstraction of a 2D game world where *S* represents the starting position and *D* represents the destination of a game character. An X represents non traversable nodes.

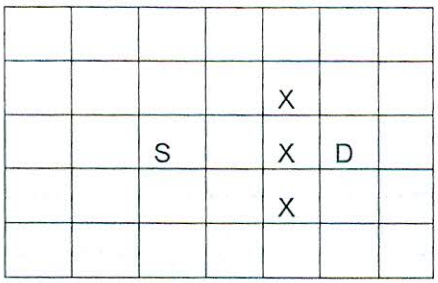

Figure 3

Using the algorithm you described in your solution to part (a), draw the state of the game world for the first 3 iterations of the A* algorithm using the **Manhattan distance** heuristic. 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) How does path finding work in Unity3D?

(10 marks)

Question 4

The simulated fish in Figure 4 exhibit the following behviours: Each fish spawns into the scene at a random position inside a rectangular fish tank. It then picks two random points inside the fish tank and will swim between these. If the fish sees a piece of food, it will chase the food and eat it if it catches it. If it doesn't manage to catch the food before another fish eats it, the fish will resume swimming between the two points. The fish get progressively hungrier until after a 120 seconds without any food, they die.

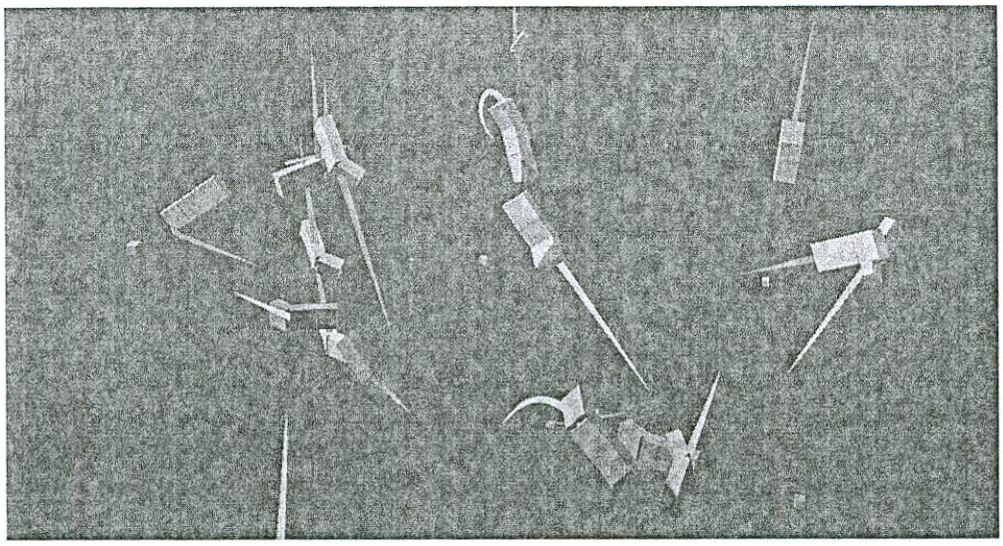

Figure 4

Answer the following questions about this scenario:

(a) Draw a state transition diagram for the fish behaviour described above.

(6 marks)

(b) Identify any state from your answer to part (a) and explain how you would implement this behaviour. Assume each fish has a game component attached that implements the *seek* steering behaviour. Include a class diagram in your answer.

(10 marks)

(c) How you would program the fish to detect the food?

(d) How would you program the Fish to get hungrier and die?

(7 marks)