PAPER CODE NO.	EXAMINER : Boris Konev
COMP222	DEPARTMENT : Computer Science Tel. No. 0151 795 4260

UNIVERSITY OF LIVERPOOL

Second Semester Examinations 2016/17

Principles of Computer Game Design and Implementation

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer FOUR questions.

If you attempt to answer more questions than the required number of questions (in any section), the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

UNIVERSITY OF LIVERPOOL

Question 1

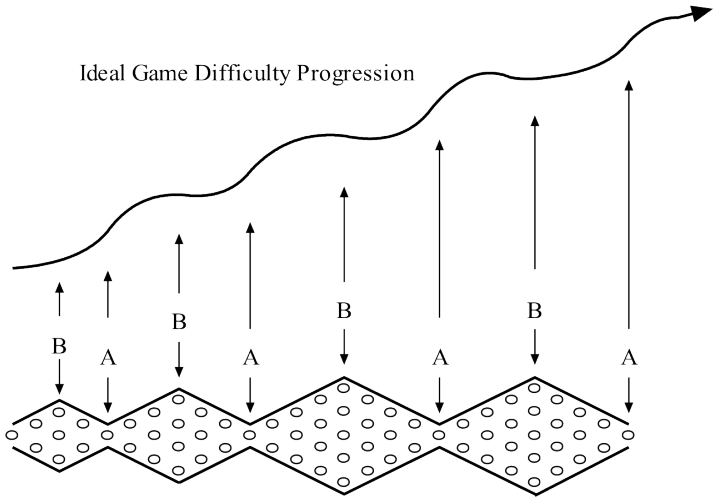
A. Describe the code structure of a modern computer game. Your answer should mention *game-specific* code, *game engine*, and *in-house tools*. You should also cover a typical game architecture to include *initialisation*, *main game loop* and *cleanup*. Give a diagrammatic representation of a typical game architecture. **7 marks**

B. Classify every one of the following

- rendering,
- behaviour specific to zombies,
- message passing,
- sound playback

as a part of game engine code or game-specific code. **2 marks**

C. In your opinion, why do game developers often design games so that periods of increased difficulty follow more relaxed periods, as shown in the diagram below?



In your answer discuss

- the link between the difficulty progression and the definition of a computer game as a sequence of meaningful choices made by the player in pursuit of a clear and compelling goal;
- how this form of difficulty progression fits the classical game structure;
- what points A and B in the diagram correspond to.

7 marks

D. Name at least two advantages and at least two disadvantages of using a scripting language such as Python or Lua for programming game-specific code. **4 marks**

E. Games are driven by a game loop that performs a series of tasks every frame. Traditionally, games only featured one game loop. Some game architectures suggest running multiple game loops in different threads. What is the reason for doing this? Name at least one advantage and one disadvantage of such an approach to multithreading. **5 marks**

Question 2

A. Let $\mathbf{V} = (3, 2, 1)$ and $\mathbf{W} = (4, 2, -6)$ be 3D-vectors. Compute (and show your working)

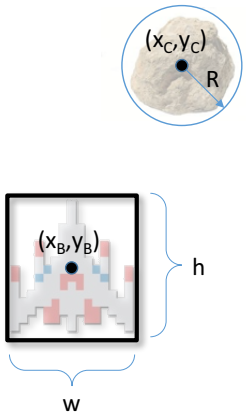
- **(a)** $\mathbf{V} + \mathbf{W}$ **1 marks**
- **(b)** $2\mathbf{V}$ **1 marks**
- **(c)** $\mathbf{V} - 2\mathbf{V}$ **1 marks**
- **(d)** $\mathbf{V} \cdot \mathbf{W}$ **1 marks**
- **(e)** $|\mathbf{V}|$ **1 marks**
- **(f)** $\text{proj}_{\mathbf{V}}\mathbf{W}$ **1 marks**
- **(g)** $\mathbf{V} \times \mathbf{W}$ **2 marks**

B. Recall that the 2D rotation matrix representing the counter-clockwise rotation about the origin by an angle θ is as follows

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Give the 3D rotation matrix representing the counter-clockwise rotation about the Z-axis through an angle α combined with counter-clockwise rotation about the X-axis through an angle β. **8 marks**

C. Assume you are implementing a simple 2D space shooter arcade game. In this game a user is in control of a spaceship, which is approximated by an axis-aligned bounding box. You need to implement collision detection with obstacles that are approximated as circles.



Given the dimensions h and w of the box, coordinates (x_B, y_B) of its centre, coordinates (x_C, y_C) of the circle and the circle radius R

- Draw a diagram representing when a collision happens; **2 marks**
- Clearly identify the cases you need to consider; **2 marks**
- Sketch a method which determines whether the box overlaps the circle. **5 marks**

Question 3

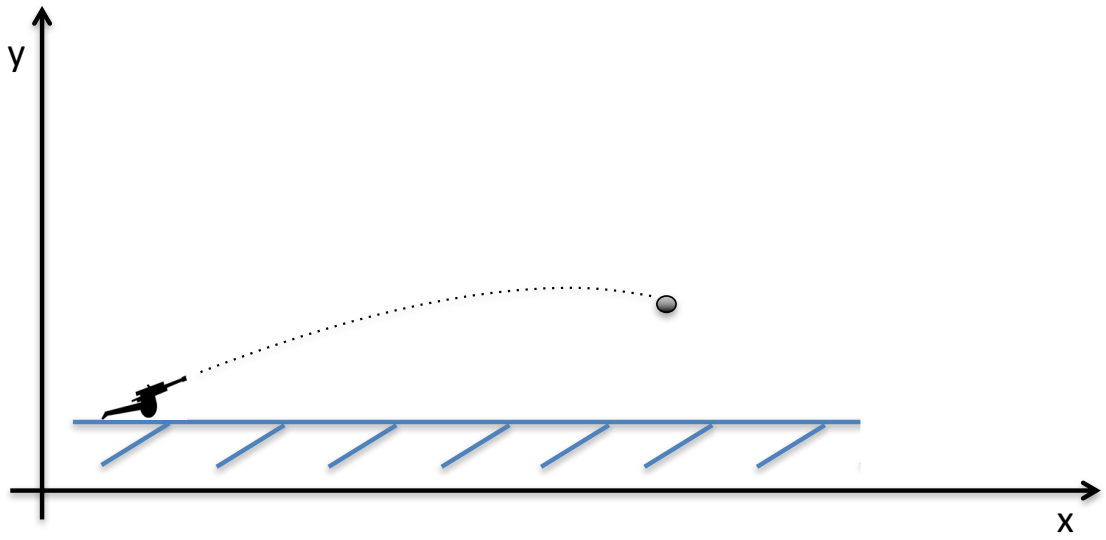
A. Modern computer games commonly use scene graphs to represent a graphical scene. Name at least three advantages of this form of data representation as compared with unstructured collections of geometries, light sources, textures, etc. **3 marks**

B. Describe the role of a renderer in a game engine. **2 marks**

C. Collision detection based on overlap testing may lead to penetration and tunnelling. Explain what is meant by these terms. Name at least two methods to avoid penetration and tunnelling. **4 marks**

D. Is it possible to combine animation-based collision response with a physics-based collision response in a game? If yes, give an example; if not provide a justification. **4 marks**

E. Consider a 2D game, in which a gun fires a cannonball. As part of the gameplay, you are modelling the effect of the air resistance on the cannonball. The mass of the cannonball is 50kg. The initial speed vector for the cannonball is $(100, 50)$.



Assuming the linear model of drag,

(a) give a graphical representation of all the forces acting on the cannonball as it flies through the air; **2 marks**

(b) describe the discrete motion of the cannonball as a sequence of its positions using Euler steps; **5 marks**

(c) sketch the `simpleUpdate()` method that implements the described motion in jMonkeyEngine. You are not required to write finished working code, but you must clearly convey the idea. **5 marks**

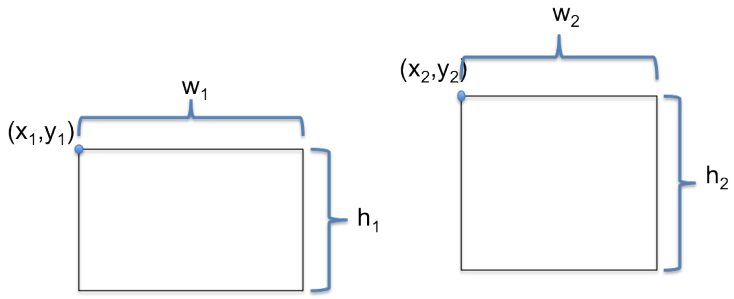
Question 4

A. Poor collision detection can lead to artefacts in computer games. Name at least two undesirable implications of poor collision detection in computer games. **4 marks**

B. Overlap testing in computer games is often approximated with the help of *bounding volumes*: a real shape is being embedded into a simplified geometry, and if two bounding volumes do not overlap, one does not perform an (expensive) triangle-level overlap test.

(a) Simple bounding volume shapes include Axis Aliened Bounding Boxes (AABBs) and Oriented Bounding Boxes (OBBs). What are the advantages of OBBs over AABBs? Are there any significant disadvantages? Your answer should contain definitions of OBBs and AABBs. **4 marks**

(b) Sketch a method which, given the coordinates of *upper left* corners of two 2-dimensional axis-aligned boxes (x_1, y_1) and (x_2, y_2) and their widths w_1, w_2 and heights h_1, h_2, respectively, determines whether these boxes intersect.

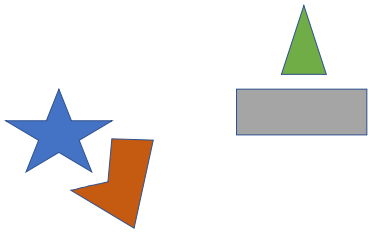


7 marks

C. Recall that bounding volume hierarchies are used to better support collision detection.

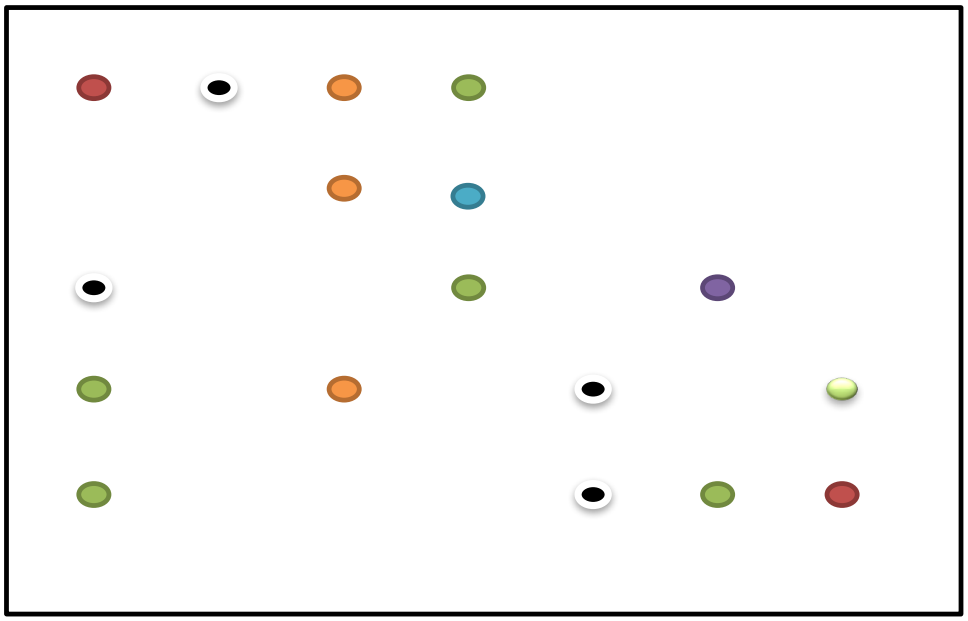
(a) Explain how bounding volume hierarchies are used in collision detection. **2 marks**

(b) Sketch a bounding volume hierarchy based on **bounding spheres** for the shape given below and use it as an example to illustrate your answer.



5 marks

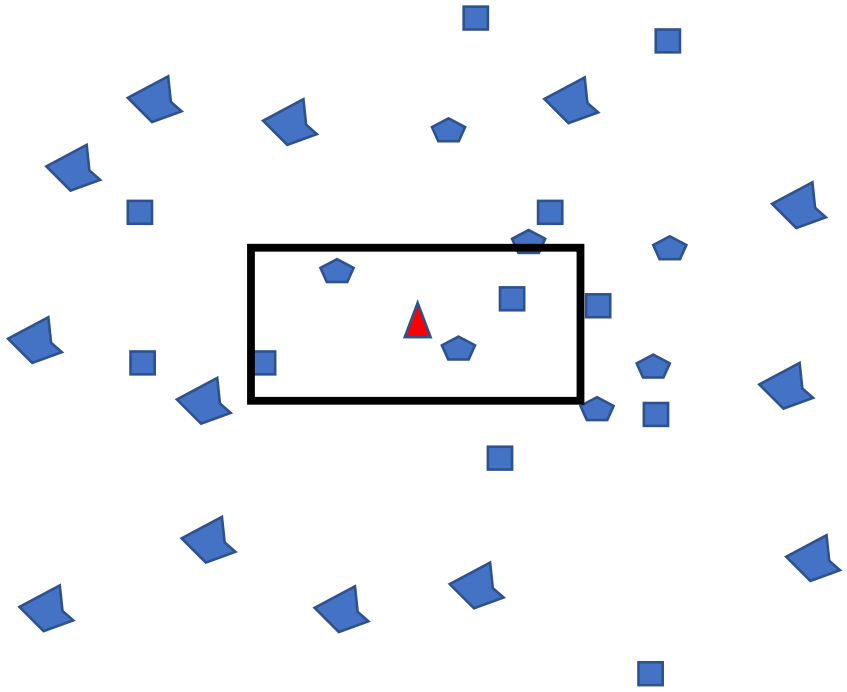
D. In your opinion, what data structure is most suitable to reduce the number of pairwise collision detection tests in the scene shown below? Explain your reasoning.



3 marks

Question 5

A. You are implementing a 2D asteroids field video game targeting a very weak computational platform. In your game, you want to have in excess of 1000 asteroids floating freely in the space. You want to model how asteroids collide and, on impact, break up into smaller ones. In the centre of your viewport (shown as a frame in the image below) you have a spaceship (shown as a triangle). As the spaceship moves in the space, the viewport follows.



Having implemented accurate collision detection based on the low-level overlap test, you discover that running it on every pair of asteroids leads to very poor performance. What techniques can you suggest to speed-up the game? **10 marks**

B. In computer game AI one can often identify two actors: a virtual player and a game agent.

- Define what they are and what role they play in computer games.
- Give an example of both.
- What is the difference between them?
- Give examples of when a computer game has a virtual player but no game agents and when a computer game has game agents but no virtual player.
- How do a virtual player and game agents collaborate? Give an example.

4 marks

C. Consider the following behaviour of a fighter game agent. The agent can be in three possible states: *idle*, *patrol*, or *attack*. In the *idle* state the agent remains motionless, in the *patrol* state the agent moves to the next checkpoint, and in the *attack* state the agent attacks another player. If the agent sees the other player, it goes into the *attack* state; otherwise,

from being idle it changes, on a timeout, to the *patrol* state and, once completed the move to the next checkpoint, returns to the *idle* state. If the enemy unit is destroyed, the agent goes from the *attack* state to the *idle* state.

(a) Give a graphical representation of the FSM that represents the agent behaviour. Indicate clearly conditions under which one state changes into another. 5 **marks**

(b) Assume now that you want the agent to show more complicated behaviour: in the *patrol* state the agent patrols four stations S_1,...,S_4 in the order $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1 \rightarrow \ldots$ and in the *attack* state the agent goes through three consecutive stages: *approach*, *aim*, *fire*.

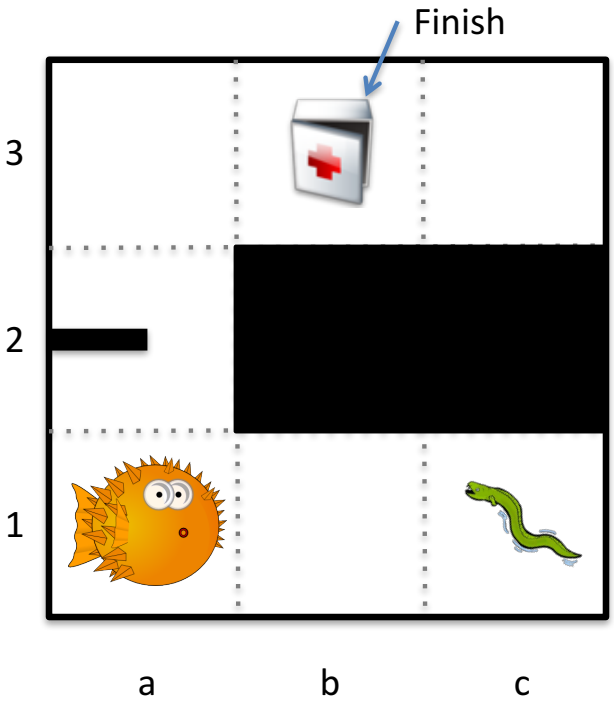
In your opinion, what is the best way to accommodate these modifications to the agent behaviour? Give a graphical representation of the new model of agent behaviour. **6 marks**

Question 6

A. Why in computer games is the character motion control routine often considered at two logical levels: steering and pathfinding? Name at least two advantages of such separation. **4 marks**

B. What is the difference between static, kinematic and dynamic physics entities in the terminology of computer games physics engine? Give an example of a static, kinematic and dynamic physics entities. **4 marks**

C. Consider the following floor plan of a 3x3 room. Tiles **2b** and **2c** are impassable. The Pufferfish has filled its stomach with water and so cannot pass through tile **2a** either; however the Eel can.

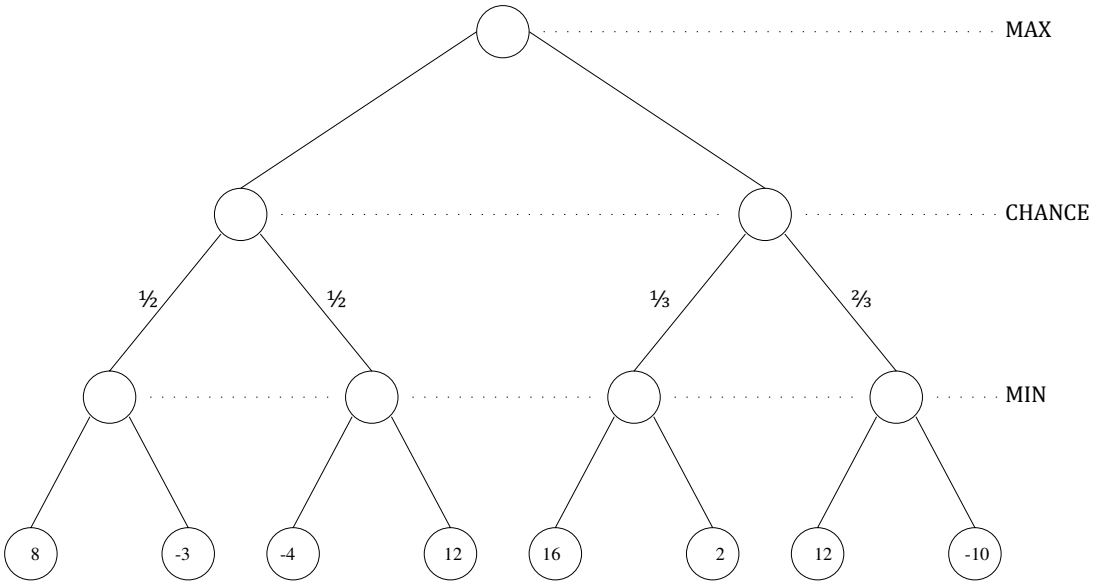


(a) Construct the tile-based pathfinding graphs for both agents (Pufferfish and Eel). **4 marks**

(b) Using the Manhattan block distance between tiles as a heuristic, which tiles and in which order the A* algorithm will explore for both agents when trying to find a path from their current position to the target tile **3b** (we assume that both agents can occupy the same tile)? Illustrate the work of the A* algorithm with a diagram. For every node of the diagram indicate clearly the cost so far and the estimated cost to the goal. **5 marks**

(c) Which techniques can be used to speed up finding that there is no path for the Pufferfish? Show how the technique of your choice works on this example. **5 marks**

D. Consider the following chance game tree. Perform the ExpectiMiniMax algorithm on this tree and compute the ExpectiMiniMax value of the root node.



3 marks