Second Semester Examinations 2017/18

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).

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. **6 marks**

B. Describe the typical functionality of a physics engine, and name at least two advantages of using a third-party physics engine and at least two advantages of using an in-house physics routine. **6 marks**

C. Discuss the difference between the traditional Artificial Intelligence discipline and Artificial Intelligence in computer games. **4 marks**

D. A computer game can be defined as a sequence of meaningful choices made by the player in pursuit of a clear and compelling goal. Justify such a definition and give a graphical representation of the classical game structure. **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) $\text{proj}_{\mathbf{V}}\mathbf{W}$ **2 marks**

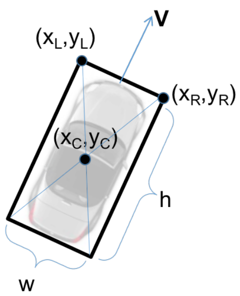
(g) $\mathbf{V} \times \mathbf{W}$ **2 marks**

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 racing game with a top view of the track. In this game a user is in control of a vehicle, which is approximated as a rectangular shape. Collision detection is implemented by checking the positions of the front corners of the vehicle.



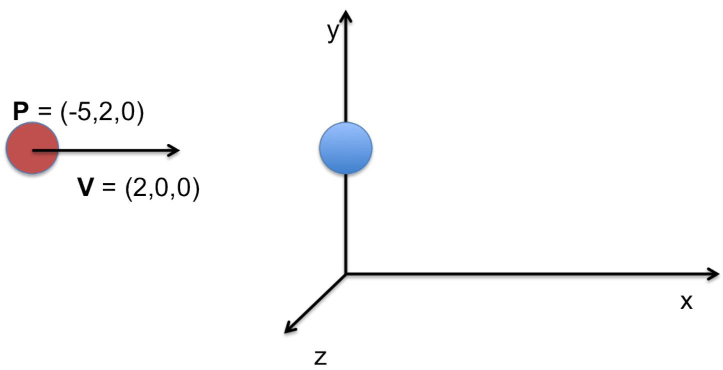
Given the dimensions h and w of the vehicle, coordinates (x_C, y_C) of its centre and its heading vector **V** determine the coordinates of the forward left corner (x_L, y_L) and forward right corner (x_R, y_R), as illustrated in the figure above. **8 marks**

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. In this module we studied two major approaches to collision detection: overlap testing and intersection testing. Define these approaches and discuss their advantages and disadvantages. **6 marks**

C. A ball with coordinates $(-5, 2, 0)$ moves uniformly with a constant speed given by vector $(2, 0, 0)$.



(a) Describe the discrete motion of the ball as a sequence of its positions using Euler steps. **4 marks**

(b) Sketch the main game loop to model the uniform motion of the ball. Your answer should take the frame rate into account. **4 marks**

(c) Suppose that there is a stationary ball of the same mass with coordinates $(0, 2, 0)$ as shown in the picture above.

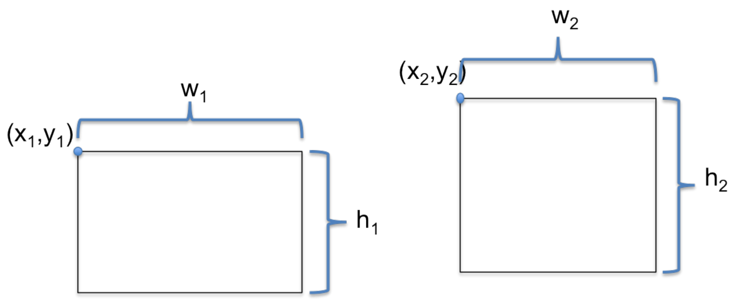
- Compute the exact moment of time when the collision between the balls happens; **4 marks**
- What will be the outcome of the collision? **4 marks**

Question 4

A. 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.

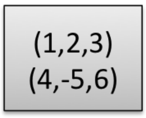
(a) Simple bounding volume shapes include Axis Aliened Bounding Boxes (AABBs) and Oriented Bounding Boxes (OBBs). What are the advantages of OBBs over AABBs? Are there any significant disadvantages? **5 marks**

(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.



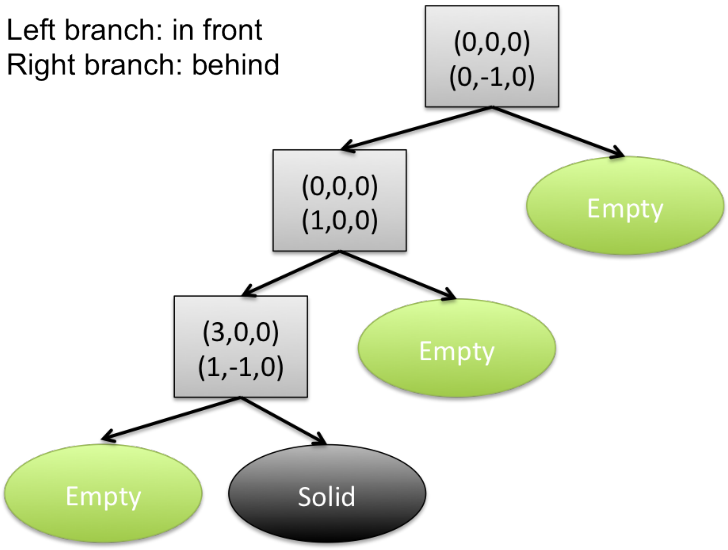
7 marks

B. Recall that a node of a solid-leaf BSP tree can be *solid*, *empty*, or it can be an internal node associated with the plain that partitions the space. In the diagram below, the plain associated with an internal (shown as a box) node is determined by a position vector (first three numbers) and a normal vector (the second line). For example, for the internal node



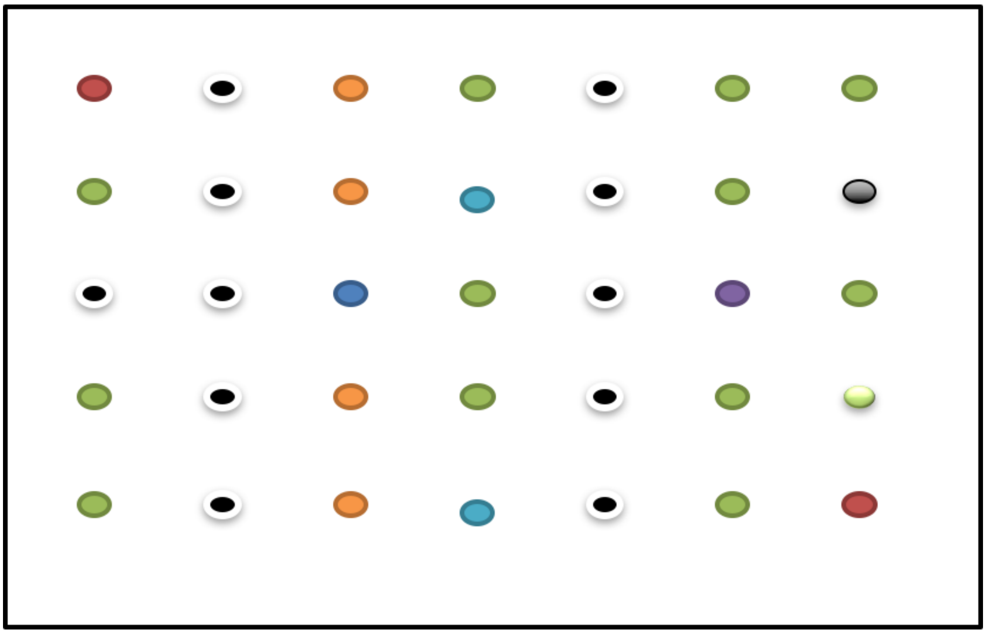
the position vector is (1,2,3) and the plain normal is (4,-5,6).

Sketch the geometrical shape defined by the solid-leaf BSP tree shown below.

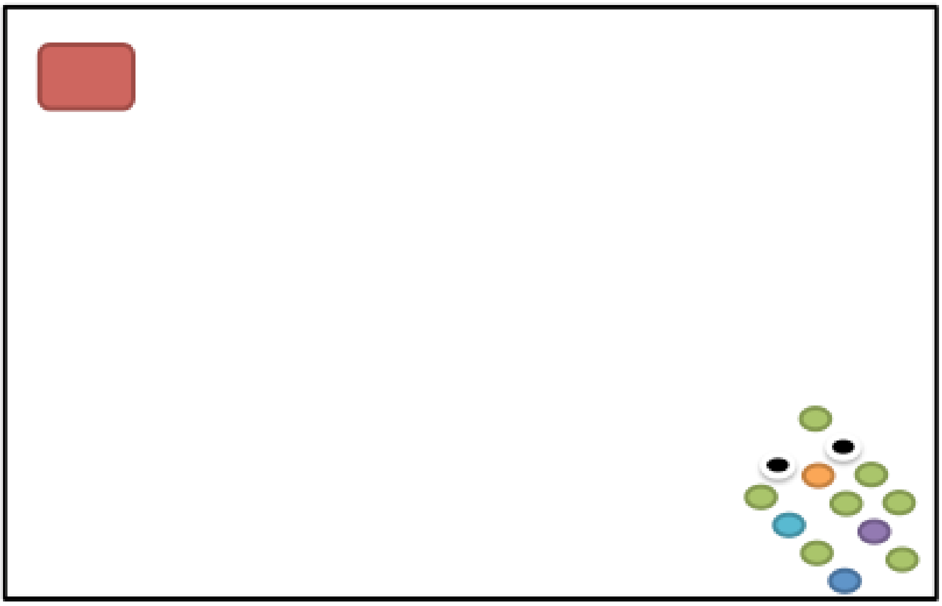


Mark clearly on your drawing the position and normal vectors for each plain. **7 marks**

C. 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.



Moreover, what data structure is most suitable to reduce the number of pairwise collision detection tests in a scene where there is one large static object in one corner and a number of small static object in the other as shown below? Explain your reasoning



6 marks

Question 5

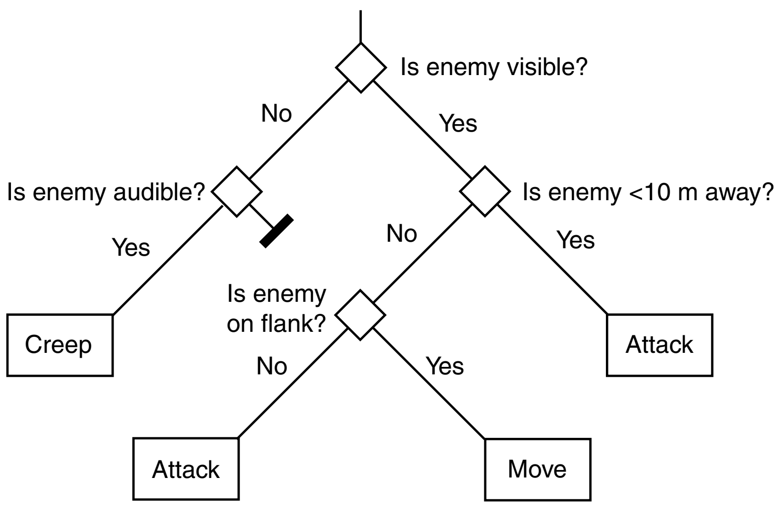
A. In agent-based computer game AI, intelligent agents continually step through the *sense-think-act* cycle. In your opinion, what is the necessity of the *sense* step of the cycle since the game world is always entirely represented inside the game and perfect information about the state of the world is always available? **5 marks**

B. 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)** What AI technique is best suitable to represent the behaviour of such an agent? **2 marks**
- **(b)** Give a graphical representation of this model of agent behaviour. Indicate clearly conditions under which one state changes into another. 5 **marks**
- **(c)** Assume now that you want the agent to show more complicated behaviour: in the *patrol* state the agent patrols four stations $S_1,\ldots,S_4$ in the order $S_1 \to S_2 \to S_3 \to S_4 \to S_1 \to \ldots$ 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. **8 marks**

C. Describe in plain English the agent strategy given diagrammatically by the following decision tree.

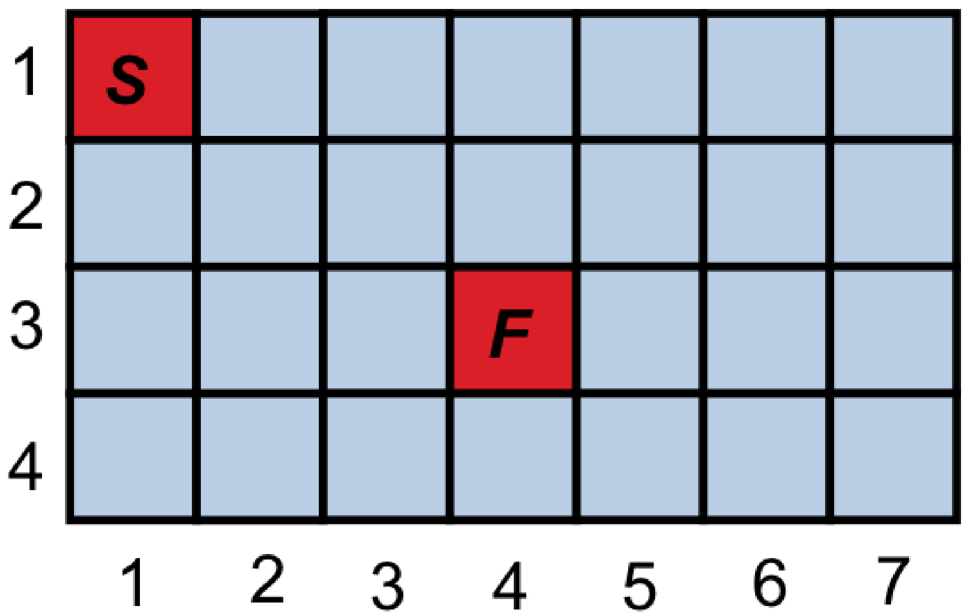


5 marks

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. Consider the following tile-based map.



The only permitted movements are up-down and left-right (if the adjacent tile exists).

(a) Define class `Node` referring to a tile on the map, that could be used by a pathfinding algorithm. **3 marks**

(b) Using Manhattan block distance as heuristics

i. Apply the A* algorithm to find a path from the start tile (marked with ***S***) to the finish tile (marked with ***F***). **6 marks**

ii. Shade the tiles which will be considered by the A* algorithm with Manhattan block distance heuristic. **6 marks**

(c) Suggest a different heuristic to reduce the number of nodes explored by the A* algorithm. Will this heuristic be admissible? **6 marks**