UNIVERSITY OF LIVERPOOL

Second Semester Examinations 2012/13

Principles of Computer Game Design and Implementation

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

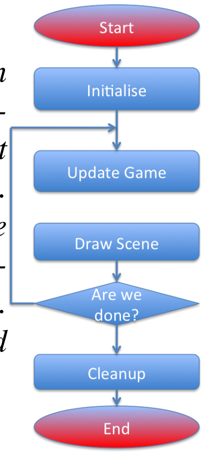
Answer FOUR questions.

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 role and functionality of the main game loop. Give a diagrammatic representation of the key steps of the main game loop. **7 marks**

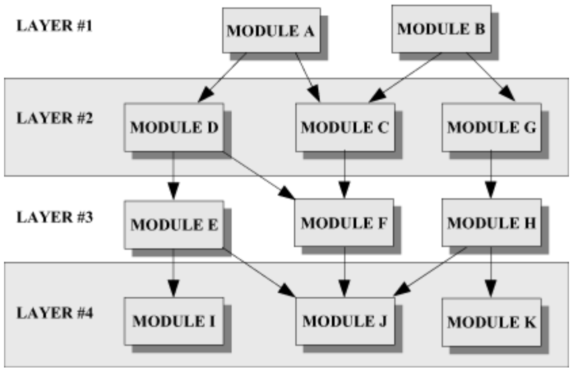
Main Game Loop: At their heart, games are driven by a game loop that performs a series of tasks every frame. By doing those tasks every frame, we put together the illusion of an animated, living world. Sometimes, games will have separate loops for the front end and the game itself, since the front end usually involves a smaller subset of tasks than the game. Other times, games are organised around a unified main loop.



B. Most modern games make a distinction between game-specific code and game-engine code. In your opinion, for each of the two distinctive parts of code, what kind of programming language is best suited for it? In your answer, discuss the role of game-specific code and game-engine code and reflect on what programming languages features make them a better candidate for the corresponding code part. **8 marks**

Game-engine code is the foundation on top of which the game-specific code is built. It has no concept of the specifics of the game being developed, and deals with generic concepts that apply to any project: rendering, message passing, sound playback, collision detection, or network communication. Game engine code typically is implemented in a high-performance *programming language such as C, C++ and, to some extent, Java.*
Game-specific code is, as the name implies, tailored to the current game being developed. It involves the implementation of specific parts of the game domain itself, such as the behaviour of zombies or spaceships, tactical reasoning for a set of units, or the logic for a front-end screen. This code is not intended to be generically reused in any other game in the future other than possibly direct sequels. Game-specific code can also be implemented in traditional languages, but increasingly scripting languages are used for this purpose. Scripting languages benefit from: ease of development; no need to re-compile the project to see the effect of changes; and, quite importantly, game designers not programmers can edit the scripts.

C. Describe a *Layered Architecture* used to organise modules in the game code. Illustrate your description with a diagram. In your answer, mention at least one advantage and one disadvantage of a layered architecture. Give an example of a game subsystem which is well suited to a layered architecture. **8 marks**



In a layered architecture specific subsystems of the game are clearly identified and separated into modules or libraries. Modules can vary in how they interface with the rest of the game. In addition, modules are arranged in rigid layers, and a module can only interact with modules in the layer directly below.

Advantages:

- *Improvesportability*
- *Best for code reuse*

Disadvantages:

- *Can lead to code or interface duplication*
- *Heavyweight*

Best suited for network communication, which perform many serial operations from layer to layer.

D. The *golden path* in a game in the optimum path for a player to take through the game to experience the game as intended and to get the maximum rewards. Name two methods used by computer game designers to keep a player on the golden path. **2 marks**

Characters refuse to obey, internal monologue, Attractions on the way.

Question 2

A. Let $\mathbf{V} = (3, 1, 2)$ and $\mathbf{W} = (6, 5, 4)$ be 3D-vectors. Compute (and show your working)

(a) $\mathbf{V} \cdot \mathbf{W}$ **2 marks**

$$(3, 1, 2) \cdot (6, 5, 4) = 3 \cdot 6 + 1 \cdot 5 + 2 \cdot 4 = 18 + 5 + 8 = 32$$

(b) $\mathsf{proj}_{\mathbf{V}}\mathbf{W}$ **2 marks**

$$proj_{\boldsymbol{V}}\boldsymbol{W} = \frac{\boldsymbol{W} \cdot \boldsymbol{V}}{\|\boldsymbol{V}\|^2}\boldsymbol{V},$$

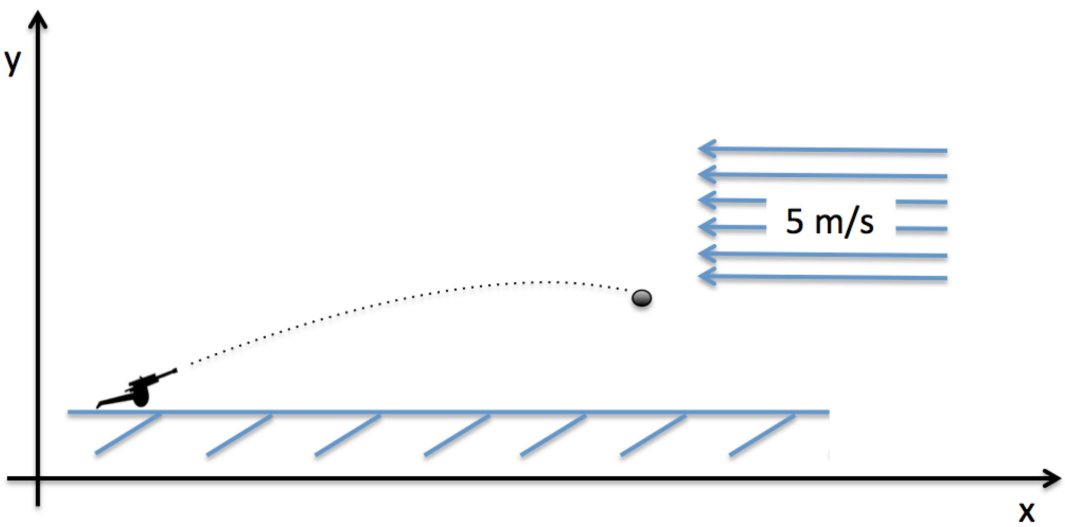
so

$$proj_{\boldsymbol{V}}\boldsymbol{W} = \frac{32}{9+4+1}(3, 1, 2) = \left(\frac{48}{7}, \frac{16}{7}, \frac{32}{7}\right).$$

(c) $\mathbf{V} \times \mathbf{W}$ **4 marks**

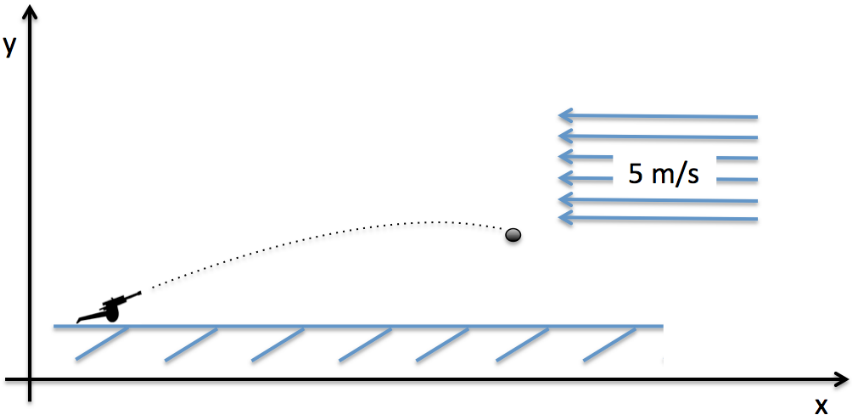
$$(3, 1, 2) \times (6, 5, 4) = (1 * 4 - 2 * 5, 2 * 6 - 3 * 4, 3 * 5 - 1 * 6) = (-6, 0, 9)$$

B. 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. Additionally, the cannonball moves against the 5m/s wind. 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**

To use Euler steps, we need to upate the forces, acceleration, velocity and position of the cannonball at every frame as follows.

$$\begin{aligned} \boldsymbol{F}_{i+1} &= -b \cdot \boldsymbol{V}_i - b \cdot \boldsymbol{V}_{wind} \\ \boldsymbol{a}_{i+1} &= \boldsymbol{g} + \frac{1}{m} \cdot \boldsymbol{F}_{i+1} \\ \boldsymbol{V}_{i+1} &= \boldsymbol{V}_i + tpf \cdot \boldsymbol{a}_{i+1} \\ \boldsymbol{P}_{i+1} &= \boldsymbol{P}_i + tpf \cdot \boldsymbol{V}_{i+1} \end{aligned}$$

where $\mathbf{V}_{wind}$ *is the* $(-5, 0)$ *vector, the speed of wind.*

(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. **3 marks**

```
Vector3f force,acceleration;
Vector3f velocity = new Vector3f(100,50,0);
Vector3f windVelocity = new Vector3f(-5,0,0);
protected void simpleUpdate() {
     force = (velocity.mult(-b)).add(windVelocity.mult(-b));
     acceleration = gravity.add(force.divide(m));
     velocity = velocity.add(acceleration.mult(tpf));
     s.setLocalTranslation(s.getLocalTranslation().
              add(velocity.mult(tpf)));
}
```

C. Let $\mathbf{V} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ be a vertical representation of a 2D vector and $\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ be a 2x2 matrix.

(a) Compute **MV**, the result of multiplication of **M** and **V**. **3 marks**

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 * 2 + 0 * 3 \\ 0 * 2 + (-1) * 3 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \end{bmatrix}$$

(b) In lay terms, what transformation does **M** define? **4 marks**

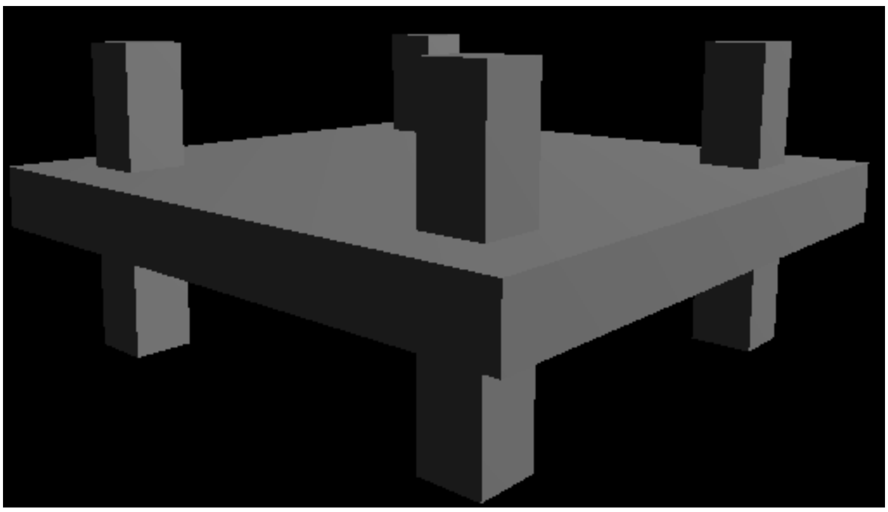
Central reflection with a centre in the origin.

Question 3

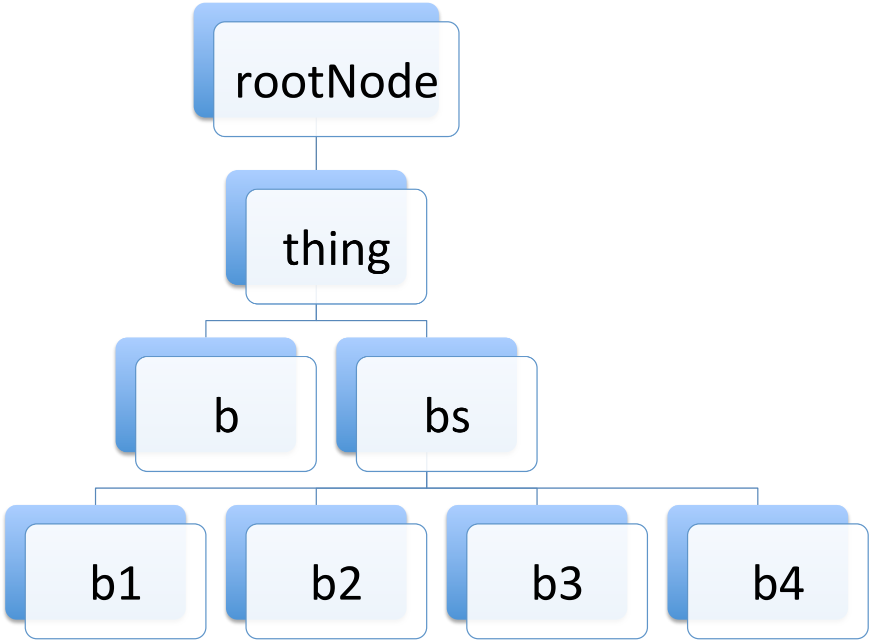
A. Consider the following jMonkeyEngine code fragment:

```
Box b  = new Box("B", Vector3f.ZERO, 10, .3f, 10);
Box b1 = new Box("B1", Vector3f.ZERO, 1,5,1);
Box b2 = new Box("B2", Vector3f.ZERO, 1,5,1);
Box b3 = new Box("B2", Vector3f.ZERO, 1,5,1);
Box b4 = new Box("B2", Vector3f.ZERO, 1,5,1);

b1.setLocalTranslation( 7, 0, 7);
b2.setLocalTranslation(-7, 0, 7);
b3.setLocalTranslation( 7, 0,-7);
b4.setLocalTranslation(-7, 0,-7);

Node bs = new Node("BS");
bs.attachChild(b1);
bs.attachChild(b2);
bs.attachChild(b3);
bs.attachChild(b4);

Node thing = new Node("Thing");
thing.attachChild(b);
thing.attachChild(bs);

rootNode.attachChild(table);
```

(a) Draw coordinate axis and sketch the resulting scene indicating the coordinate origin and coordinates of all geometries. **7 marks**



(b) Draw the scene graph specified by this code fragment. Name at least three advantages of this form of data representation as compared with unstructured collections of geometries, light sources, textures, etc. **7 marks**



Some advantages:

- *Scene graphs provide an intuitive way to manage large amounts of geometric and rendering data*
- *The data needed for rendering, which is associated with the scene graph nodes, can be kept separate from the rendering code.*
- *Hierarchical animated models are easier to deal with*
- *View frustum culling can be supported by using bounding volumes at the nodes.*

B. What is a physics engine? 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**

A physics engine is computer software that provides an approximate simulation of certain simple physical systems, such as rigid body dynamics (including collision detection), soft body dynamics, and fluid dynamics,
Advantages of game engines:
- Complete solution from day 1
- Proven, robust code base (in theory)
- Lower costs
Advantages of home-grown solutions:
- Choose only the features you need
- Opportunity for more game-specific optimisations
- Greater opportunity to innovate

C. Sketch the main game loop to model the uniform motion of a particle starting from $(0, 0, 0)$ in the direction specified by vector $(1, 2, 3)$ with a constant speed of 5 units per second. You answer should take the frame rate into account. **5 marks**

- *Let* $pos := (0, 0, 0)$
- *Let* $U := (1, 2, 3).normalize()$
- *While (loop not finished)*
  - $pos := pos + 5 * tpf * U$
- *EndWhile*

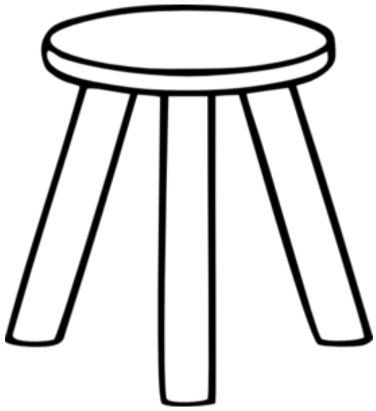
where tpf is the time per frame corresponding to the current frame rate.

Question 4

A. 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. **7 marks**

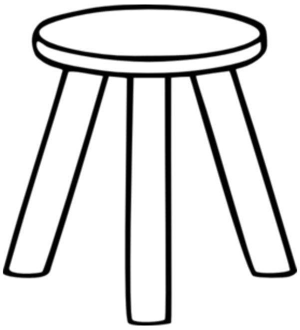
The main difference is that overlap testing detects whether a collision has already occurred, and intersection testing predicts if a collision will occur in the future.
The idea of overlap testing is that at every simulation step, each pair of objects will be tested to determine if they overlap with each other. If two objects overlap, they are in collision. This is known as a discrete test since only one particular point in time is being tested.
The biggest advantage of overlap testing is that it is easy to implement. It's biggest disadvantage is that it handles poorly objects travelling fast. For overlap testing to always work, the speed of the fastest object in the scene multiplied by the time step must be less than the size of the smallest collidable object in the scene. This implies a design constraint on the game to keep objects from moving too fast relative to the size of other objects.
Intersection testing tests the geometry of an object swept in the direction of travel against other swept geometry. Whatever geometry the object is composed of, it must be extruded over the distance of travel during the simulation step and tested against all other extruded geometry.
The disadvantages of overlap testing include a poor handling of networked games. The issue is that future predictions rely on knowing the exact state of the world at the present time. Due to packet latency in a networked game, the current state is not always coherent, and erroneous collisions might result. Therefore, predictive methods aren't very compatible with networked games because it isn't efficient to store enough history to deal with such changes and, in practise, running clocks backward to repair coherency issues rarely works well.
One more potential problem for intersection testing is that it assumes a constant velocity and zero acceleration over the simulation step. This might have implications for the physics model or the choice of integrator, as the predictor must match their behaviour for the approach to work.

B. Bounding volume hierarchies are used in computer games to facilitate collision detection. How are they used? Sketch a bounding volume hierarchy based on OBBs for the shape given below and use it as an example to illustrate your answer. Your answer should contain a definition of OBBs and state the main principles underpinning the construction of bounding volume hierarchies.



8 marks

Collision detection 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. OBBs stands for oriented bounding boxes.*
Although the tests themselves have been simplified, the same number of pairwise tests are still being performed. It's a hierarchical structure, in which the root node completely encapsulates the object; Children give a "tighter fit" for the shape; and Recursive / iterative algorithms to are used toconstruct and navigateBVHs.



C. What will be the speed vector of a particle moving with the speed given by vector $(1, 1, 1)$ after the collision with a plain given by its normal $(0, 0, -1)$? **3 marks**

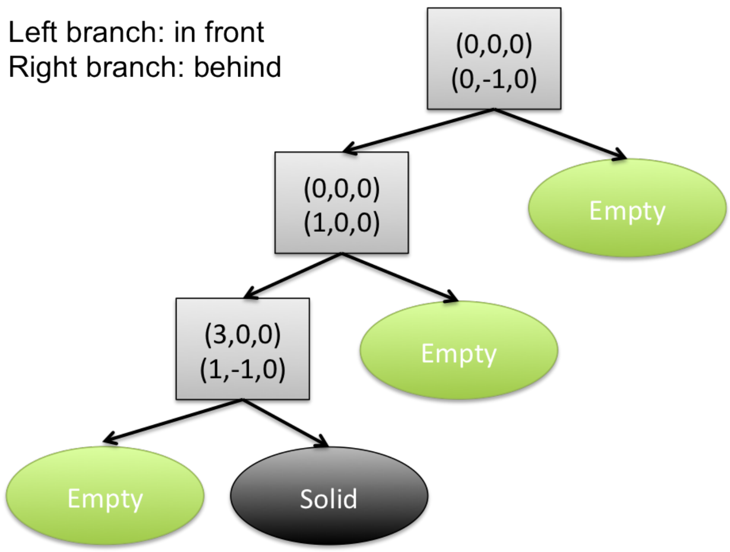
$(1, 1, -1)$

D. 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

(1,2,3)
(4,-5,6)

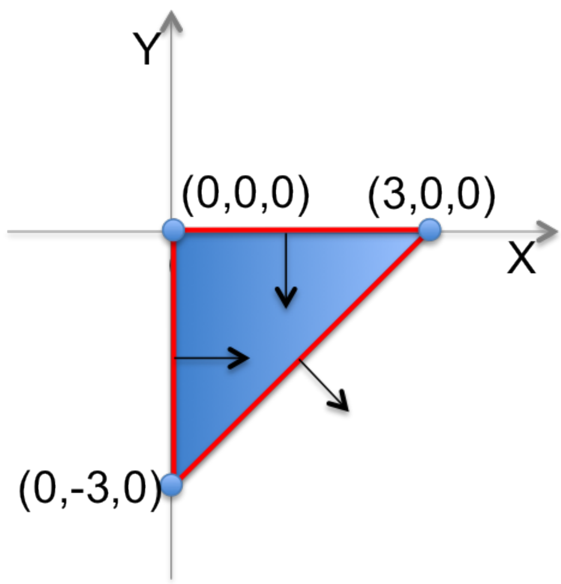
the position vector is (1,2,3) and the plain normal is (4,-5,6).

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**

The BPS tree determines the following shape:



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**

Game agents are autonomous entities associated with game characters (enemies, companions, computer car drivers etc.).
While all of this rich information exists, it may be expensive or difficult to tease out useful and pertinent information. Additionally, at any time, the game agent can query the game world representation to locate the player or other enemies, but most players would consider this cheating. Therefore, it is necessary to endow the game agent with certain limitations in terms of what it can sense. For example, game agents should typically not be able to see through walls. Game agents are usually given human limitations. They are restricted to knowing only about events or entities they have seen, heard, or perhaps were told about by other agents. Therefore, it is necessary to create models for how an agent should be able to see into the world, hear the world, and communicate with other agents.

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**

Finite state machine

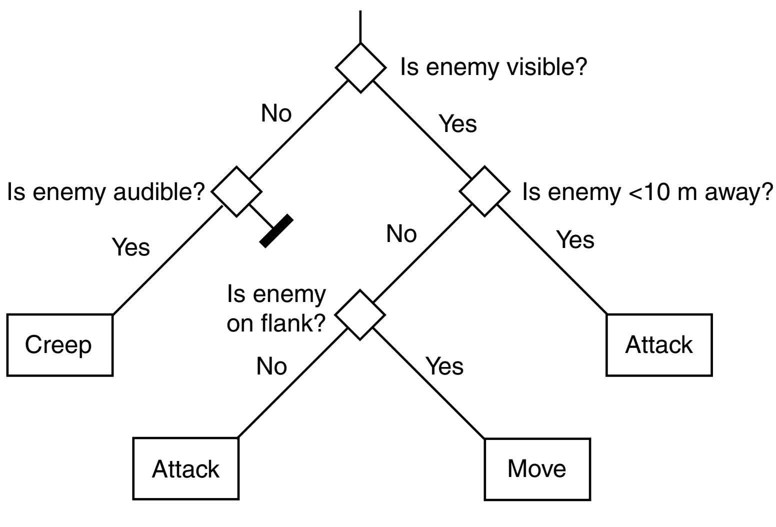
(b) Give a graphical representation of this model of agent behaviour. Indicate clearly conditions under which one state changes into another. 5 **marks**

(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**

There are two options how this can be handled. Either to add more states to the FSM, or consider a hierarchical FSM in which the patrol *state and* attack *state are FSMs as follows.*

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



5 marks

If the enemy is visible and within 10 meters away then attack. If the enemy is visible, further than 10 meters and is on flank then move. If the enemy is visible, further than 10 meters and is not on flank then attack. If the enemy is not visible but is audible then creep.

Question 6

A. Describe the difference between Goal Oriented Behaviour (GOB) and Goal Oriented Action Planning (GOAP) as defined in the lectures. **3 marks**

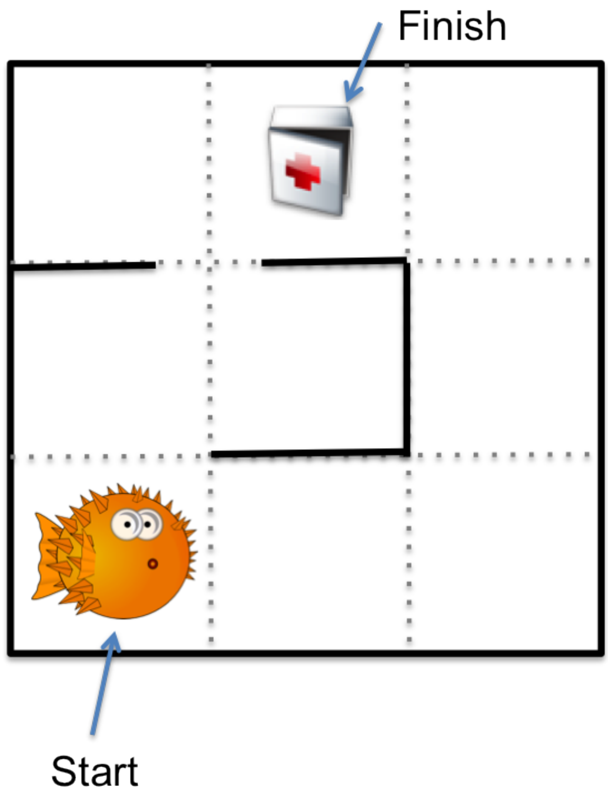
GOB requires the agent to choose one action out of several alternatives; GOAP requires the agent to select a sequence of actions.

B. Suppose that a computer character has three goals: Eat(3); Sleep(3); Go_to_bathroom(2). The insistence of every goal is given in the brackets. Which of the following actions should the character choose based on the *overall utility* approach? The effect of every action is given in the brackets.

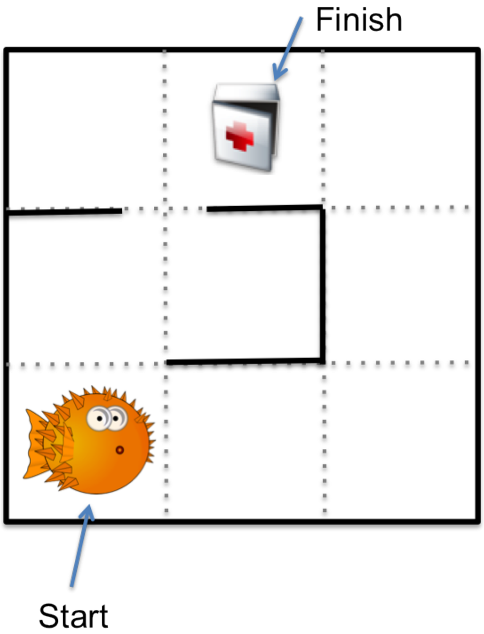
- Drink-soda (Eat – 1; Go_to_bathroom + 1)
- Visit-Bathroom (Go_to_bathroom – 4)
- Eat-dinner (Eat – 3)
- Take a nap (Sleep – 2)

2 marks

C. Consider the following floor plan of a 3x3 room. The globefish filled its stomach with water and can only move between adjacent tiles where there are no obstacles.



(a) Construct the tile-based pathfinding graph. **5 marks**



(b) Using the Manhattan block distance between tiles as a heuristic, apply the A* algorithm to the graph constructed and find a path between Start and Finish. 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**

D. Why is machine learning not used in many games? **4 marks**

Some of this is due to the relative complexity of learning techniques. The biggest problems, however, with learning are those of reproducibility and quality control. Since the algorithm learns behaviour itself it is virtually impossible to test if it performs correctly and locate bugs.

E. What is hierarchical pathfinding? Give at least one advantage and one disadvantage of hierarchical pathfinding. **6 marks**

Each stage of the path will consist of another route plan. This is done by grouping locations together to form clusters. The individual locations for a whole room, for example, can be grouped together. There may be 50 navigation points in the room, but for higher level plans they can be treated as one. This group can be treated as a single node in the pathfinder.
Advantages: faster and requires less memory. Disadvantage: may not find the best path.