UNIVERSITY OF LIVERPOOL

Second Semester Examinations 2014/15

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

1. What is the difference between game engine code and game-specific code? **4 marks**

Most games make a distinction between game-specific code and game-engine code. 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-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. Illustrate your answer with positioning each of

2. Classify every one of the following

- rendering,
- behaviour specific to zombies,
- message passing,
- level implementation, and
- sound playback

as a part of game engine code or game-specific code. **5 marks**

Rendering, message passing and sound playback belong go game engine.
Behaviour of zombies and level implementation are a part of game-specific code.

3. 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

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

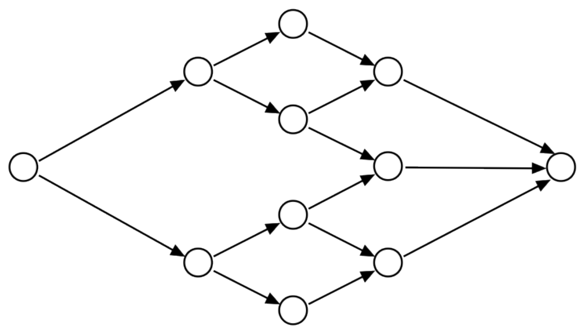
In academia, some AI researchers are motivated by philosophy: understanding the nature of thought and the nature of intelligence and building software to model how thinking might work. Some are motivated by psychology: understanding the mechanics of the human brain and mental processes. Others are motivated by engineering: building algorithms to perform human like tasks.
Games developers are primarily interested in only the engineering side: building algorithms that make game characters appear human or animal-like.
In academia, AI is about the matter. In games AI is about the appearance.

5. 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**

Justification:

- *Must have choice, or it is not interactive*
- *Must be a series of choices or it is too simple to be a game*
- *Must have a goal or it is a software toy*

Graphical representation:



Starts with a single choice, widens to many choices, returns to a single choice

6. 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

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

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

$$\boldsymbol{V} + \boldsymbol{W} = (1 + 2, 3 + 4, 5 + 6) = (3, 7, 11)$$

(b) $2\mathbf{V}$ **2 marks**

$$2\boldsymbol{V} = (2, 6, 10)$$

(c) $\mathbf{V} - 2\mathbf{V}$ **2 marks**

$$\boldsymbol{V} - 2\boldsymbol{V} = -\boldsymbol{V} = (-1, -3, -5)$$

(d) $\mathbf{V} \cdot \mathbf{W}$ **3 marks**

$$(1, 3, 5) \cdot (2, 4, 6) = 1 \cdot 2 + 3 \cdot 4 + 5 \cdot 6 = 2 + 12 + 30 = 44$$

(e) $|\mathbf{V}|$ **3 marks**

$$|\boldsymbol{V}| = \sqrt{1^2 + 3^2 + 5^2} = \sqrt{35}$$

(f) $\mathsf{proj}_{\mathbf{V}}\mathbf{W}$ **3 marks**

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

so

$$proj_{\boldsymbol{V}}\boldsymbol{W} = \frac{44}{35}(2, 4, 6) = \left(\frac{88}{35}, \frac{176}{35}, \frac{264}{35}\right).$$

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

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

2. Let $\mathbf{M} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\mathbf{M}' = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. Compute (and show your working)

(a) The product of the two matrices, $\mathbf{MM}'$. **3 marks**

$$\boldsymbol{MM}' = \begin{bmatrix} 1 * 5 + 2 * 7 & 1 * 6 + 2 * 8 \\ 3 * 5 + 4 * 7 & 3 * 6 + 4 * 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

(b) The product of the first matrix and the vector, $\mathbf{MV}$. **3 marks**

$$\boldsymbol{MV} = \begin{bmatrix} 1 * 1 + 2 * 2 \\ 3 * 1 + 4 * 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

Question 3

1. 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. **5 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.*

2. Describe the role of a renderer in a game engine. **4 marks**

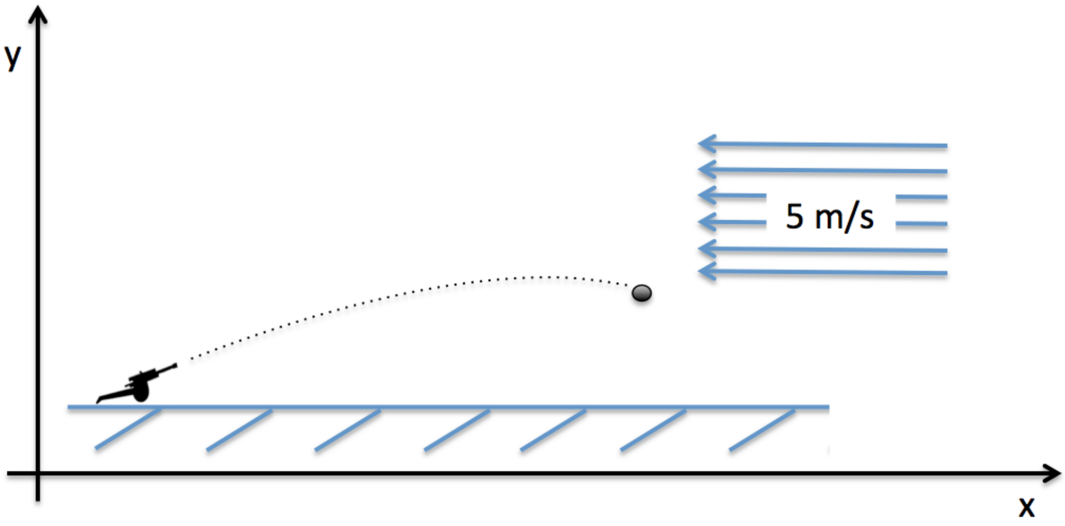
   A renderer

   - *Transforms geometry from world space to screen space*
   - *Eliminates hidden objects*
   - *Draws the transformed scene*

3. 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**

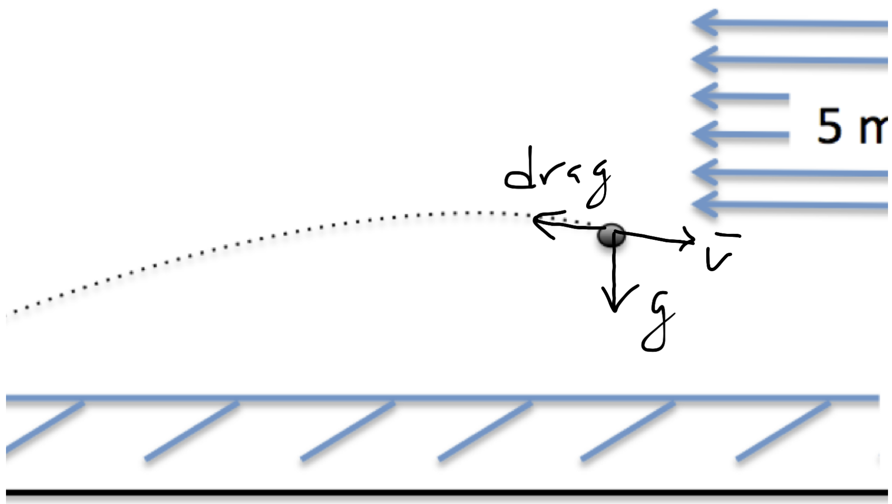
The main difference is that overlap testing detects whether a collision has already occurred, and intersection testing predicts if a collision will occur in the future.
The idea of overlap testing is that at every simulation step, each pair of objects will be tested to determine if they overlap with 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.

4. 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.

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

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

(c) Suppose you are running a computer simulation of the cannonball flight. What method can you use to determine the initial speed vector of the cannonball so that it hits the ground 10 meters from the gun? **3 marks**

While it is possible to integrate the motion equation and get an analytic solution, for a computer symulation scenario one can run several symulation using the divide and conquer strategy to determine the initial speed vector.

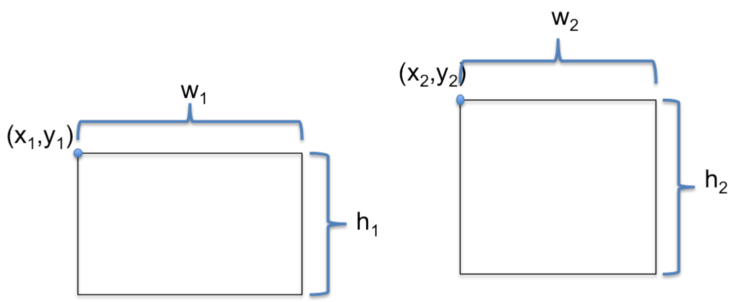
Question 4

1. 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**
   - *Players/objects falling through the floor;*
   - *Projectiles passing through targets;*
   - *Players getting where they should not get;*
   - *Players missing a trigger boundary.*

2. 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? **4 marks**
   OBBs fit the real geometry tighter. The main disadvantage is that it is harder to check if the bounding volumes overlap; however, this disadvantage is outweighed by better collision detection offered by OBBs.

   (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.

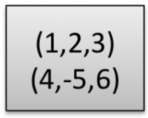


7 marks

There is an elegant solution to this problem based on the check of when boxes do not *overlap.*

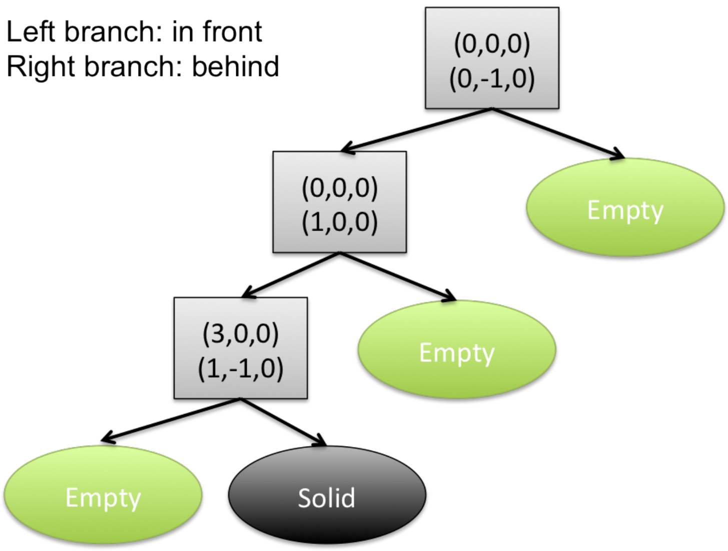
```
if((x1 + w1 < x2) || (x2 + w2 < y2) ||
   (y1 + h1 > y2) || (y2 + w2 > y1)) {
        return false;
}
else {
        return true;
}
```

3. 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



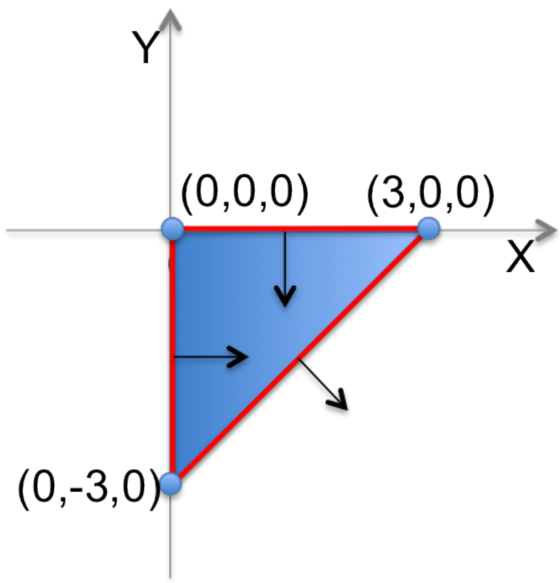
the position vector is (1,2,3) and the plain normal is (4,-5,6).

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

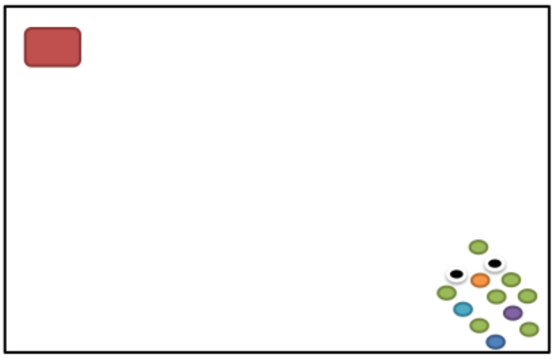


Mark clearly on your drawing the position and normal vectors for each plain. **7 marks**

The BPS tree determines the following shape:



4. In your opinion, 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.



3 marks

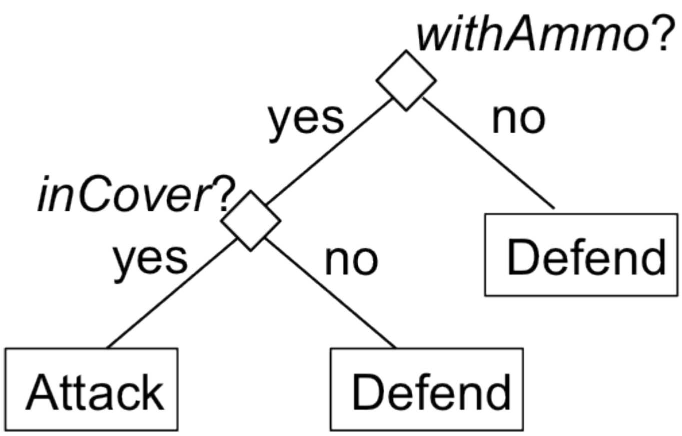
It's best to use non-uniform grid based data structure such as a quad tree, a k-d tree of a BSP tree.

Question 5

1. Given the following table representing decisions taken by a human player,

Health	Cover	Ammo	Decision
Healthy	In Cover	With Ammo	Attack
Hurt	In Cover	With Ammo	Attack
Healthy	In Cover	Empty	Defend
Hurt	In Cover	Empty	Defend
Hurt	Exposed	With Ammo	Defend

and always considering attributes in the order **Ammo**, **Cover**, **Health**, apply the decision tree learning algorithm studied in the lectures to construct the decision tree that, based on attribute values, gives the same decision specified in the table. **6 marks**



2. 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?

6 marks

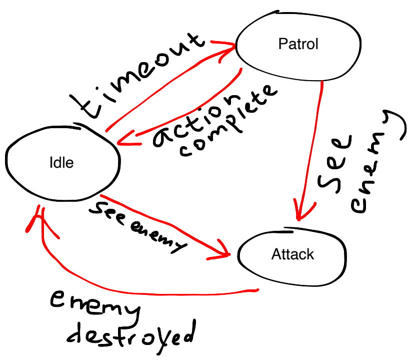
Game agents are autonomous entities that observe and act upon an environment. They often are associated with game characters (enemies, companions, computer car drivers etc.). Early agents did not show much of intelligent behaviour, often their choices were random. In modern computer games they can learn and react to the environment in an intelligent way.
A virtual player takes place of a human opponent in a game. The virtual player performs the same operations as the human player. The intelligence of the virtual player is perceived through the moves it makes and the results of choices. For example, a chess player is a virtual player.
Many first-person shooters have game agents (enemies) but no virtual player. Chess has a virtual player but no game agents.
In real-time strategies, the computer-controlled side is a virtual player (thus, there might be more than one virtual player in a game), while individual units are game agents, which often can take decisions on their own in order to follow orders.

3. 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

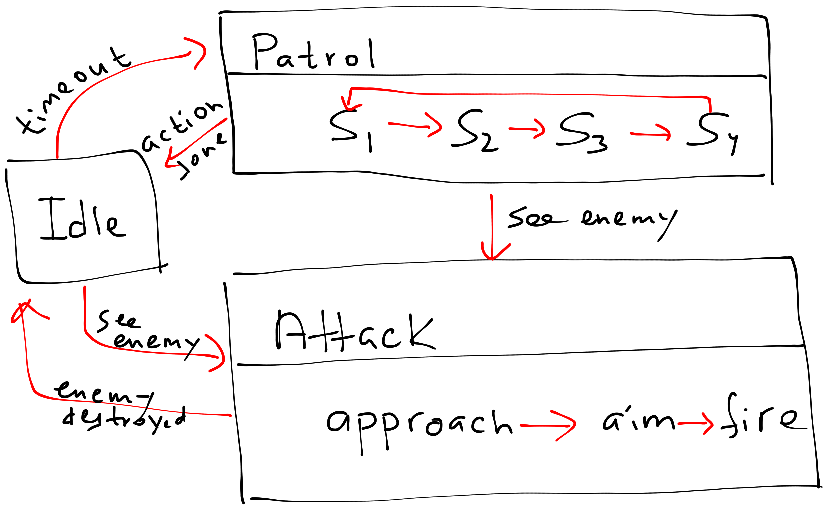
(b) Give a graphical representation of this model of agent behaviour. Indicate clearly conditions under which one state changes into another. 5 **marks**



(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. **6 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.*



Question 6

1. 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**

   Steering techniques allow a computer character to navigate from one position into another provided there are no (or few simple) obstacle on the way. Pathfinding is used whenever a computer entity needs to find a way to a goal avoiding obstacles.
   Advantages:

   - *steering is closer to game engine an often requires integration with the physics engine; pathfinding is closer to the decision taking level. Keeping them separate leads to a cleaner code and better task distribution.*
   - *Pathfinders can be reused in a different kind of game even of another genre.*

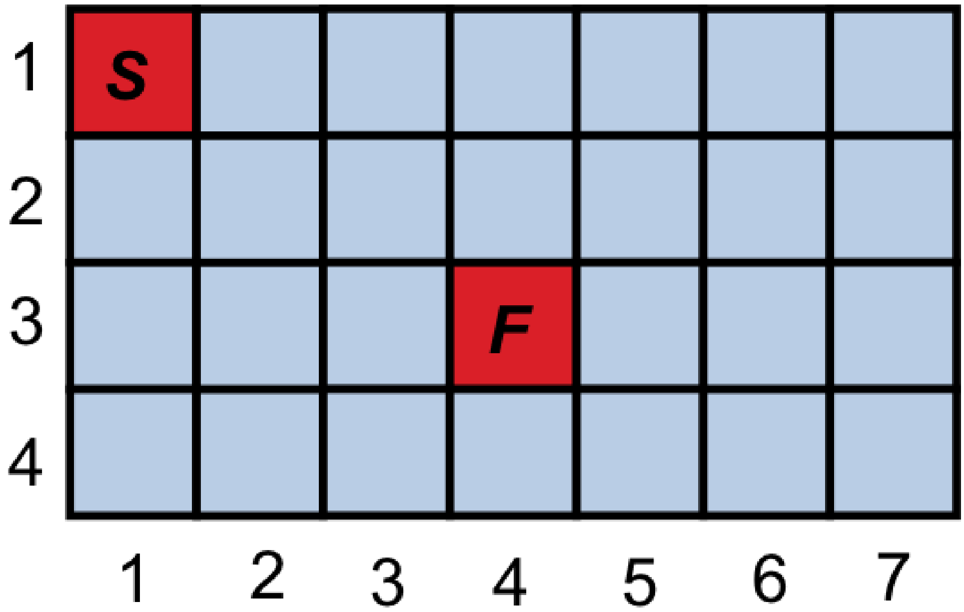
2. For a turn-based strategy game such as Civilization, and for a first-person shooter game like Quake, which of the following space search structures would you use and why? For the same two games, which one will you not use and why?

   - Regular grids
   - Waypoint Graph Based
   - Navigation Meshes

   3 marks

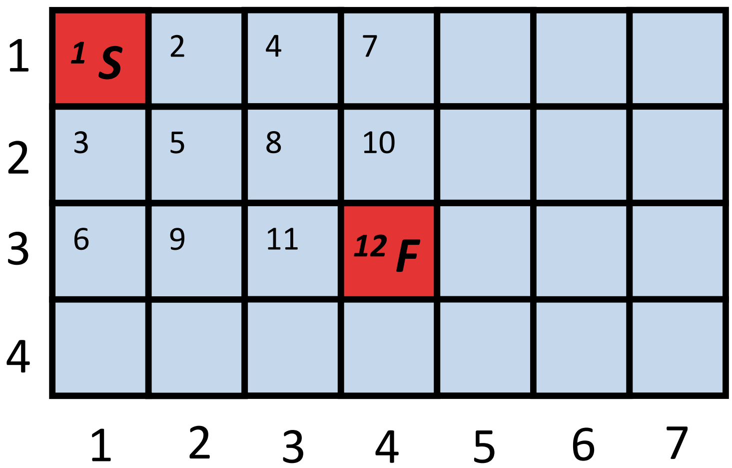
   For a TBS game use Regular Grids. It fits naturally the game nature that units typically occupy a cell of the grid. When entities navigate, they move from cell to cell. Waypoints are not suitable for TBS since they restrict movements of entities so that if a path is blocked, it is next to impossible to amend the plan. For an FPS game use waypoint graphs or navigation meshes. In a 3D game an agent occupy any position. Both navigation mashes and waypoints are user-controlled, which allows for a computer agent to choose a natural path in the 3D space. Regular grids are not suitable for 3D shooters since they generate chunky paths and the shape of rooms may not fit well the grid making it harder to detect which cells are passable and which are not.

3. Consider the following tile-based map.



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

(a) Using Manhattan block distance as heuristic, number the tiles of the map in a way consistent with how the A* algorithm explores the search space to find a path from the start tile (marked with ***S***) to the finish tile (marked with ***F***). **4 marks**
There is more than one way to explore the tiles. Here is one example



(b) Suggest a different heuristic to reduce the number of nodes explored by the A* algorithm. Will this heuristic be admissible? **6 marks**
Manhattan block distance multiplied by a small number. Although this heuristic is inadmissible, it is more goal-oriented and would make the A search more greedy.*

4. 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.

5. 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

After performing the action

- *Drink-soda the overall utility is* $2 + 3 + 3 = 8$
- *Visit-Bathroom the overall utility is* $3 + 3 + 0 = 6$
- *Eat-dinner the overall utility is* $0 + 3 + 2 = 5$
- *Take a nap the overall utility is* $3 + 1 + 2 = 6$

Therefore, the Eat-dinner action should be taken.

6. In computer games physics simulation often runs at a different speed compared with the drawing updates. Discuss advantages and disadvantages of this approach. **3 marks**

Advantages: by running physics simulation at a lower speed it is possible to keep the vide update rate consistently high. One disadvantage is the fact that slower physics simulation can lack detail and lead to abnormal behaviour of physics objects.