CS 638: Computer Game Technology

Sample Solutions

Dec 14, 2001

Question 1:

You wish to design an AI that responds to the player on roughly half of the occasions they meet, but should ignore the player on the other occasions. What techniques could you use to design such a character? (Hint: There are at least two options.)

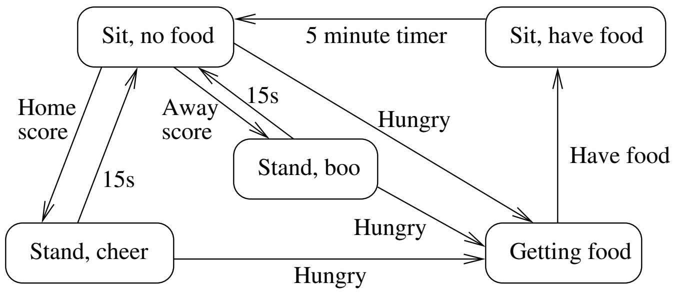
- *You could use a non-deterministic finite state machine. There would be a node for "respond to player" and an event "see player". There would be a "see player" link out of every state, with all of those links leading to the "respond" state. Finally, the probability associated with all of those links would be 1/2.*
- *You could use rules with a random number function in the conditions. For instance you might have a rule "if SeePlayer and Random ¿ 0.5 then respond."*
- *You could use regular finite state machines but only choose to process a "see player" event half of the time it is received. This might be difficult to implement and break the state machine in other ways.*
- *You could* ***not*** *use fuzzy logic, because it gives a deterministic answer.*

Question 2:

A sports spectator is to have the following behaviors:

- By default they sit and watch the game.
- If their team scores, they stand and cheer.
- If the opposing team scores, they stand and boo.
- After 15 seconds of standing, for any reason, they sit down again.
- If they are hungry at any time, they go to the food stand.
- If they have food, they sit a and eat it, and do not stand for any reason.
- It takes 5 minutes to eat food, after which they no longer have food.
- The spectator is never hungry while they have food.

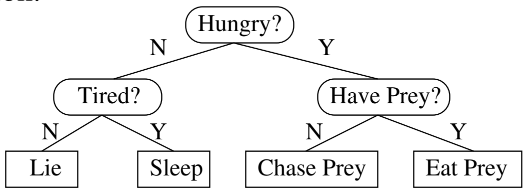
Design a finite state machine that will demonstrate the behavior described.



"Sit, no food" is the start state.

Question 3:

Convert the decision tree below into a set of rules for use in a rule-based system. It is intended to model the behavior of a lion.



- *If **Hungry** and textbfHavePrey then **EatPrey***
- *If **Hungry** and textbfnot HavePrey then **ChasePrey***
- *If **not Hungry** and textbfTired then **Sleep***
- *If **not Hungry** and textbfnot Tired then **Lie***

Question 4:

Imagine an environment with pairs of “teleport” waypoints that take you to their partners instantaneously at no cost.

a. Why is Euclidian distance not an admissible heuristic in this situation?

   Paths through the teleport waypoints might be shorter than the Euclidian distance. Hence, the Euclidian distance heuristic does not always underestimate the true cost, and is inadmissible.

b. How would you modify the waypoint graph and/or the heuristic to generate a new heuristic that is admissible.

   To fix the problem, first merge each teleport pair into a single waypoint. This makes the waypoint graph explicitly reflect the fact that you can get between the teleport pair at no cost. When you merge, all the edges into either of the original nodes now go into the merged node.

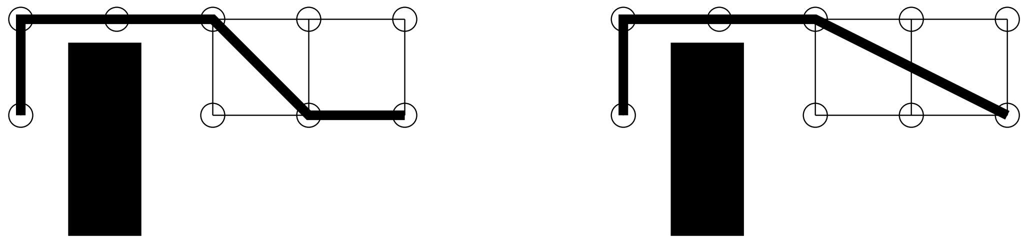
   Now, however, you have a node that is in two places at once (each teleport merged node is located at both of the original pair locations). The heuristic must be modified to account for this. When estimating the cost of getting to the goal from the merged teleport node, you compute the distance to the goal from both of the original teleport waypoints, and use the shorter one as the heuristic estimate.

   But that’s still not enough. The Euclidian distance does not take into account the fact that you can jump distances through teleporters. So you need to build the heuristic by taking the minimum cost of all the paths through all the potential teleport sequences. That is, for every combination of teleporters, sum the Euclidian distances between them, and take the cheapest path.

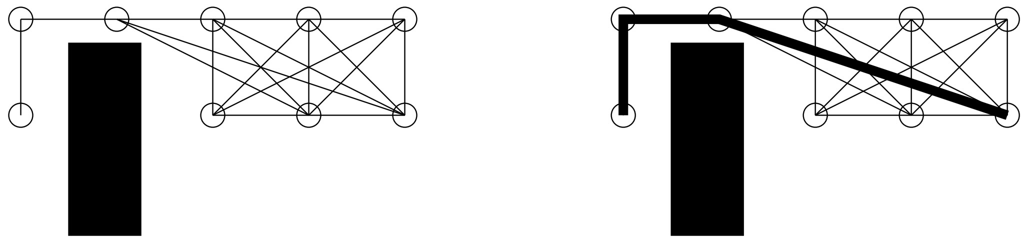
Question 5:

This question examines the effectiveness of the "Option 1" greedy path smoothing approach, in which points on the path are eliminated in a greedy fashion until the next point cannot be reached with a straight line, and the process repeats on the remainder of the path.

a. Show the result of greedy path straightening on the following path and obstacles. The light lines are the grid on which planning was done.



b. Show the optimal path found on the waypoint graph below, which has been augmented with links between any pair of nodes that can see each other.



c. What are the advantages of planning on the augmented waypoint graph compared to greedy path straightening?

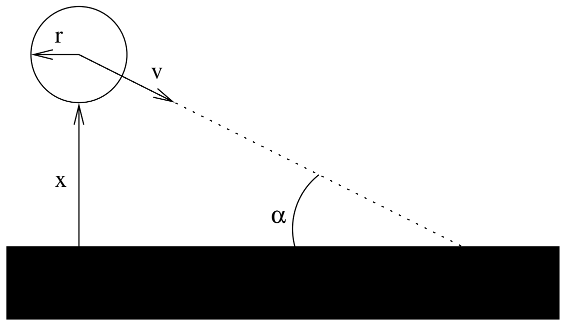
Planning on the augmented graph will give you the true shortest path. Not only will this path look better, it will also remove any need for path straightening.

d. What are the disadvantages?

The augmented graph may be very much larger than the simple grid graph. In fact, in open space it is of size $O(n^2)$ for n waypoints. The simple grid is only of size $O(n)$.

Question 6:

Consider the situation illustrated below, in which a sphere is approaching an infinite plane. The sphere has radius r and is moving with constant velocity v at an angle of α to the plane. Currently the distance between the ball and the plane is x. Write an expression for the time at which the ball will hit the plane, given that the current time is $t = 0$.

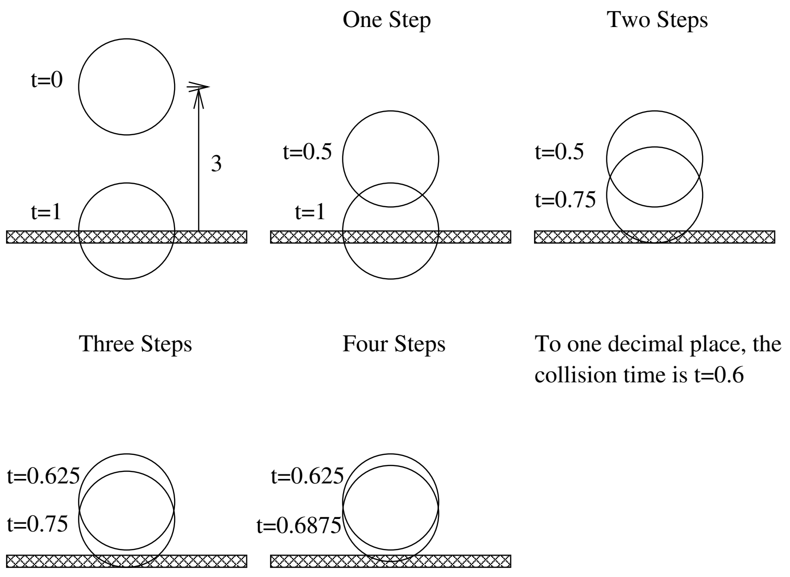


When the sphere must travel x units perpendicular to the plane before it hits. The component of the speed in the direction perpendicular to the plane is $v \sin \alpha$. So the time to hit the plane will be $x/(v \sin \alpha)$. The component of the speed tangential to the plane has no impact on the collision time.

There are other ways to derive that result, including computing how far the center of the sphere must travel in its current direction of motion, and then simply dividing that by v.

Question 7:

The state of a ball is shown at time $t = 0$, and again at time $t = 1$, when it is penetrating the plane. Use interval halving to find the collision time accurate to 1 decimal place. (Assume the ball is moving with constant velocity).



To one decimal place, the collision time is t=0.6

Question 8:

Assume that you have two choices for broad-phase collision detection: bounding boxes and bounding spheres. You will perform tests between bounds for all n^2 object pairs and only do narrow-phase (accurate) collision detection between pairs whose bounds overlap.

Say that the bounding box overlap test costs 1 unit to perform, and eliminates %85 of the potentially colliding pairs. The sphere test costs 2 units to perform, but eliminates %90 of the pairs.

a. If the cost of the narrow-phase test is 100 units, which of the bounding options is best?

   Let N_b be the total number of object-object pairs that must be tested. With the box test, we must perform $N_n = 0.15N_b$ narrow-phase tests. So the total cost is $C = N_bC_b + N_nC_n = N_b + 0.15 \times N_b \times 100 = 16N_b$.

   With the sphere test, we must perform $N_n = 0.1N_b$ narrow-phase tests. So the total cost is $C = N_b \times 2 + 0.1 \times N_b \times 100 = 12N_b$.

   Hence, the sphere bound gives better performance.

b. At what narrow-phase cost do the two bounding options result in the same overall cost?

   We are looking for the value of C_n at which $N_b + 0.15 \times N_bC_n = N_b \times 2 + 0.1 \times N_bC_n$. Simplifying gives $N_b(1 + 0.15C_n) = N_b(2 + 0.1C_n)$. Solving for C_n gives $C_n = 20$.

   The intuition is that a cheaper narrow-phase test makes it more important to have a cheap broad-phase test, and less important how effective that test is. An expensive narrow-phase test makes it worth the effort to do very good broad-phase testing.

Question 9:

You are considering using oriented bounding cubes for a bounding volume. These are cubes that are rotated and sized to best fit the object.

a. How many separating planes/axes must you test to be sure that two oriented cubes collide?

   You must do 15 separating axis tests - 6 for the six distinct face orientations, and $3 \times 3 = 9$ for the edge pair combinations.

b. Will bounding cubes perform better overall than OBBs. Why or why not?

   Oriented cubes will be worse than OBBs. The collision test is a little bit cheaper because all the side lengths of a cube are equal, but this saving will not out-weigh the poor tightness of a cube bound.

Question 10:

Which protocol, UDP or TCP/IP, would you prefer for each of the following types of game data?

a. Player state *UDP*

b. AI position information *UDP*

c. AI state machines transitions *TCP*

d. Customized player geometry *TCP*

e. AI orientation information *UDP*

f. A player obtaining a power up *TCP*

Question 11:

You decide to use packet compression in your multi-player game. With this technique, every outgoing packet is compressed using a standard compression algorithm, and every incoming packet is decompressed.

a. What impact does this technique have on bandwidth requirements for the game? Explain your answer.

   This technique will reduce the bandwidth requirements, because each packet will be smaller. The extent of the reduction will depend on the quality of the compression.

b. Which sources of latency are affected by this technique, why, and in what way?

   This will ***increase*** *the latency in the both the client and server frame time because it takes some time to compress or decompress the packets before sending them to the operating system or using their contents for the game. It will* ***decrease*** *the latency due to the* ***OS layer*** *because the packets are smaller (and take less time to copy, for instance). It will* ***decrease*** *the latency through the physical network because the bandwidth required will be lower and the packet will be shorter.*

c. How is game security impacted by this technique?

   Packet compression might make your game a little more secure, because it makes the packets harder to read. On the other hand, it is an easy measure to defeat, because the decompression code is available on the player's machine inside the client executable.

   Also, the interplay between compression and encryption is not good. The whole point of encryption is to introduce hard to guess information into the encrypted data. But this explicitly reduces the amount of redundant information in the data, which makes it harder to compress (in fact, the better the encryption, the worse the compression). You don't need to know about the relationships between compression and encryption for this class.