Overview

Your task for this assignment is to explore decision making and learning methods. Working alone and using Processing (http://www.processing.org/), your task is to implement the algorithms described below and to analyze your results in a 3–5 page writeup. Note that for some of these tasks you will be integrating your new solutions with your code from Assignments 1 and 2.

First Steps

For this assignment, you will be using your movement and pathfinding code from Assignments 1 and 2 to demonstrate higher-level decision making. To start, ensure that you have functioning pathfinding algorithms and basic movement algorithms including seek and pathfollowing.

Additionally, design an environment in which you will run your experiments for this assignment. Like for Assignment 2, your environment should have at least three rooms and a number of obstacles. You may reuse your environment from the second Assignment if you wish. Lastly, incorporate a “monster” into your environment using the behavior tree and movement algorithms from your previous assignment. Your monster’s task should be to wander the environment and try to eat your character. Hint, you need two behaviors here: 1) wander and 2) seek/arrive/pursue. Any challenges in getting your monster to work? Write about it!

Decision Trees

Implement a decision tree algorithm. Recall from class that a decision tree is slightly different than a behavior tree. In a behavior tree we have choice nodes and sequence nodes whereas in a decision tree we only have choice nodes. That is, your decision tree should output a single behavior that will run either for a fixed amount of time or until completion. How you implement that is your choice. In order to facilitate decisions, you will need to parameterize the state space. You can use a binary- or nominal-valued vector as your state representation. Examples of attributes include the room the character is in, the number of other characters in the room, distance to the closest obstacle in a particular direction, etc. Construct your decision tree using those attributes.

Make sure you clearly define your attributes in your writeup, as well as depict your decision tree. Take screenshots or make videos of your character moving through the space and changing behaviors according to the decision tree you constructed and the attributes that evolve in the environment. Write about it!

Markov Processes

Model your environment as states in a Markov process. These states should include all relevant information for your environment including the location of your “monster”. Note that six states to indicate each of your three rooms both with and without the monster is the minimum requirement. Construct a probabilistic transition matrix to govern the movement of your character from state to state (and therefore room to room in the concrete environment).
Begin by putting a single character in the environment. What happens? How does it work? Now add additional characters, what looks different? What changes? Can you tweak the transition matrix to achieve a stable distribution of characters?

Note that the transitions are actually movement goals that need to get realized either through seek or pathfinding.

**Decision Tree Learning**

Run your Markov process code and record detailed information about how the characters move through the environment. Each time a character takes a transition, record the attribute values you defined in the first decision tree section of the assignment along with the movement behavior or target that is selected. Use this data to learn a decision tree. How much data did you collect? Why? Now execute that decision tree. What is your character doing? Why? How does it look different than the Markov process that generated the data you used to learn your decision tree? Go back, and try a different transition matrix with your Markov process, collect a new set of data, and learn a new decision tree. How does this one result in different behavior than your last one? Why?

**Extra Credit: Reinforcement Learning**

Using the same set of states for your Markov process, create a MDP. In addition to the states, you will need to define a set of movement actions, a new transition matrix that takes into account the starting state as well as the action taken, and (most importantly) a reward signal. Describe all of those components in detail.

How long does it take to learn a policy for your MDP? What method did you use? Why did you choose the actions you chose? Be specific in your description of your implementation and your results. As always, screenshots and videos should be included.

**Writeup**

Now that you have implemented and evaluated a number of algorithms, write a 3–5 page paper summarizing your findings. It is strongly suggested that you do not limit yourself to only answering those questions posed in this assignment. Think creatively about what you have done. What other parameters can you tweak and what effect do they have on the results? The most successful writeups will contain evidence that you have thought deeply about these algorithms and what they can produce and have gone beyond what is written in the assignment.

As an appendix to your paper, please include all relevant screenshots to support your analysis. The appendix does not count toward your 3–5 page requirement.

**What to submit**

By the start of class on 4/18/11, please upload a zip archive to moodle. This archive should contain all of your appropriately labeled files from each part of the assignment, a README file, plus your writeup in pdf format. For this homework, rather than require a specific naming convention for each of your files, please include a README with instructions for compiling and running your code. In the README, you should list which files correspond to which parts of your homework assignment.