6.836 Embodied Intelligence – Problem Set 1

Massachusetts Institute of Technology

Due: Friday, February 23rd, 2001

General Problem Set Disclaimer: This is a graduate class, and as such, the problem sets are very open ended; there will not be much handholding, and there definitely will not be great levels of clarification of precisely how to respond. Typical "problem/solution" matching is not what we are looking for here. Whereas a typical problem set rewards finding a unique "right" answer, these problem sets reward demonstrating (sometimes creatively) an understanding of how to find a workable solution. Scores on problems will reflect the perceived depth of understanding in the suggested area, though we will be as explicit as possible regarding our grading metrics. The policy on late problem sets can be found on the web at http://www.ai.mit.edu/courses/6.836/news.html.

Problem Set 1 Disclaimer: These exercises require thinking but not much writing (less than a page for each sub-problem). It can be handled in a boring way or you can have some fun. Extra resources have been suggested on the web (http://www.ai.mit.edu/courses/6.836/handouts/handouts.html), but they exist solely for your hacking pleasure: your grade is not raised or lowered by turning in code.

☞ Please turn in each **Problem** on separate sheets of paper for grading purposes.

Problem 1 Consider modifying Braitenberg's vehicles *2a*, *2b*, *3a*, and *3b* so that the connections between the sensors and the actuators are non-monotonic. Design vehicles that perform the following tasks (being sure to state explicitly any environmental assumptions that come into play):

(a) Circle a single light source
(b) Go back and forth between two light sources
(c) Make a figure eight with the light source inside one of the lobes

While it is possible to trivialize these problems by choosing the "right" physical assumptions, doing so limits your ability to demonstrate understanding, thereby lowering your score.

Problems 2, 3, 4 Consider modifying the Braitenberg vehicles again, this time by adding a light source to the top of each vehicle so that other vehicles can sense its presence. Imagine a pair of each type of vehicle otherwise alone on a flat plane. Analyze the behavior of each pair under each of the following conditions:

	No additional conditions	Light is restricted*	Light must exceed threshold#	Light is both restricted and must exceed threshold
Vehicle 2a	**2 (a)**	**3 (a)**	**4 (a)**	**4 (e)**
Vehicle 2b	**2 (b)**	**3 (b)**	**4 (b)**	**4 (f)**
Vehicle 3a	**2 (c)**	**3 (c)**	**4 (c)**	**4 (g)**
Vehicle 3b	**2 (d)**	**3 (d)**	**4 (d)**	**4 (h)**

* The light on the vehicles is not omnidirectional, but visible only over some angle range in the front of the vehicle instead.

The total light level measured by the two front-pointing sensors of a vehicle must exceed some threshold for the vehicle's lights to turn on.

OVER →

Problem 5 (This question is based on the material in Chapter 2 and will be covered in the February 16th class.)

a) Come up with a network of AFSM's as in Figure 2.6 of Chapter 2 that could control the behavior of the video game character Pac Man (point your browser at http://www.classicgaming.com/pac-man/ if you're not familiar with Pac Man). You should use the environment of the game (i.e., don't worry about how Pac Man actually moves), but imagine designing a controller for Pac Man, the creature.

You may assume the action commands present in the game (moving forward, backward, left and right, automatically eating whatever is in your path, etc.), however, you will need to imagine some simple sensors for things such as a ghosts, walls, pellets, etc. Don't worry about the technology of the sensors, but do describe your assumptions about their behavior and limitations.

Your final answer should have the same notation as figure 2.6 and include text descriptions of how each box works. Your system should include at least 6 AFSM's and at least one instance each of suppression, inhibition, and defaulting.

b) Describe the behavior of your system, including any faults and/or irregularities that might arise from your particular implementation.