# *Machina Speculatrix* Project
# Part II

In this continuation of the project, you will now make the robots exhibit animal-like behavior. The best document to consult for a description of the basic behaviors you must implement is the web site http://wiki.solarbotics.net/pmwiki/pmwiki.php/Main/MachinaSpeculatrix, especially the lower half of the web page. You do not have to understand the circuitry and are not expected to model the circuitry; you just have to make your robots have most of the operation modes mentioned, namely, the search, move, dazzle, and collision modes. These modes are achieved mainly by having the steering motor and drive motor run at full speed, half speed or off, in various combinations. You will have to determine two light threshold levels that will determine what mode the robot is in. Some of the behaviors require a robot to change its operation mode even though external signals have not changed. In the original circuit, these changes were accomplished by discharging capacitors and by feedback loops; in your program, they will be accomplished by timers, i.e., counters that are set to some initial value, get decremented once by each tick message sent to the robot, and when the counter reaches zero, the robot changes to another mode or setting of its motor prameters.

Note: When the drive motor is operating at half speed, that means that the parameter `ds` is set to half its normal value. When the steering motor is operating at half speed, that means that the parameter `dbeta` is set to half its normal value. A motor is off when the corresponding parameter is set to 0.

Below is a brief description of the four modes based on the description of the above web site.

## Search mode
The robot is searching for light. The drive motor is half speed, the steering motor is full speed.

## Move mode
Light has been detected at a level between the lower light threshold and the upper light threshold. The steering motor is off, the drive motor is full speed. After a certain number of cycles of the simulation (use a timer as described above), the robot switches back to search mode. (When the light is first detected and this mode is entered, the robot is probably turning. It will continue to turn when in the move mode, so it will likely move to a new position where the light level is below the lower threshold, but when it switches back to search mode, it will soon find the light again and move a little closer to it.)

## Dazzle mode
When the detected light level exceeds the upper threshold, the robot enters dazzle mode. First the robot sets the steering motor to half speed and the drive motor to full speed, then after a short period of time (use a timer), the steering motor is set to full speed and the drive motor is set to half speed. The robot stays with these settings for a short period of time (use a second timer). During both of these phases of the dazzle mode, light levels are ignored. Then it switches back to search mode or move mode. The duration of the first phase will be short, but the duration of the second phase may be in seconds.

## Collision mode
This mode takes precedence over the other modes when a robot collides with a barrier, mirror or another robot. The robot goes into dazzle mode and repeats dazzle mode if the robot is still in contact with something at the end of the dazzle mode.

You will want to modify your code so that each time the robot calculates a new position (new value for `alpha`, `x` and `y`), it checks to see whether the robot would be in contact with anything if it made this move. If it will not be in contact with anything, the three variables are updated (the robot makes the move). If it will be in contact with something, it does not move (the three variables are not changed), and goes into collision mode if it is not already in collision mode. The parameter `beta` is still updated, however, since this represents the rotation of the steering wheel. This modified calculation of where the robot is and its orientation is done each time that the robot motion is calculated regardless of which mode the robot is in.

## The pilot light

A robot's pilot light is on when either the steering wheel or the drive wheel is running at half speed. If neither wheel is running at half speed, it is off. You should show that the pilot light is off by making the small circle in the image of the robot change to black. A robot can see the pilot light of another robot only when that robot's pilot light is on.

## What to turn in

Modify your code to implement the above described modes. Experiment with different values of the parameters (thresholds, timing lengths, speeds, etc., maybe even the size of your robots and graphics window to give the robots enough room to maneuver in) until your robots produce the desired behavior as described in the various web site documents at the URLs I sent by email last month. Add the following functions to your code. These functions set up the environment in the graphics window for several experiments as described. After each function is called, the call `(run n)` will do the experiment by running the simulation for `n` cycles. Email your program to both the TA and me. In your email or in a separate attached document, describe the main problems you ran into when doing this project (both parts), any comments you want to make about the project and any suggestions you may want to make to improve the project.

You may want to put barriers all the way around the edge of the graphics window so that the robots stay on screen, but this is optional.

## Demo1

The call `(demo1)` should set up the graphics window with one robot and one light. After this call, running the simulation should make the robot wander a bit, discover the light and go towards it, get close enough to it to go into dazzle mode, and it should then circle the light (all the time looking for another light), all as a natural consequence of the operation modes as described above.

## Demo2

This is like `demo1` except that there are now two lights. The robot discovers the first light, gets close and starts circling it, then discovers the other light while circling the first, and goes to the second light. I would guess that if the simulation is run long enough, the robot will start circling the second light, discover the first light and go back it. The robot will shuttle back and forth between the two lights in this fashion.

## Demo3

In this demo, there is a robot, a barrier, and a light on the other side of the barrier. When the experiment is run, the robot will discover the light, go toward it but run into the barrier, and will work its way around

the barrier before continuing toward the light.

## Demo4

In this demo, there is a robot and a mirror. When the experiment is run, the robot will come near the mirror, see its own reflection (the pilot light) in the mirror and move towards the reflection. Since the pilot light goes out when it is moving towards its reflection, it will dance in front of the mirror.

## Demo5

In this demo, there are two robots. In the experiment, they will see each other's pilot light and move towards each other. We will have to decide whether they appear to be dancing with each other or fighting!

## Demo6

In this demo, there are four robots. The expectation is that they will wander around the graphics window as a "social" group because they are attracted to each other from time to time. The question here is how this behavior will appear different from that found in the two robot experiment.