Squaring the circle

This demonstration illustrates what is meant by saying that behavior is the control of perception (meaning, behavior is the visible part of the process by which we control our perceptions). A mouse is used to draw a square on the screen. But between the mouse and the screen there is a transformation which requires that the mouse move in a circle or a triangle—no matter what figure is being drawn!

This program is a demonstration of a basic fact about people organized as negative feedback control systems: they vary their actions as required to control the perceptual effects of those actions. Here, a person moves a mouse to make a spot trace out a square on the computer screen (the controlled perception), but the movement required is to move the mouse either in a triangular pattern or a circular pattern—not a square pattern.

Operating the demo

Start the program. You will see a screen showing a large square with a small circle in it, and a dot over on the right. If the small circle starts to move, hit the space bar to reset it.

Across the top of the screen you will see

CIRCLE PERSON CONTROLLING TRACE ON

Below that are the instructions for giving commands (upper or lower case):

c,t: select circle or triangle
m,p: Model or Person traces square

and in the next row down,

n: toggle random noise on and off (when model is controlling spot)
s: toggle trace on and off (leave record of track on left or just show small circle)

The next commands are not described on the screen:

space: reset without changing settings
q: quit

Let's postpone the discussion. You can now start moving the spot (the small circle) around the square by using the mouse. Move it very slowly – this is not easy. Be as accurate as you can. When you have traced completely around the square, look at the record of the mouse movements on the right side of the screen. It will be a circle! To try it again, just hit the space bar to reset.

Now press the "t" (or 'T') to select the triangle. Repeat the trace around the square. This is even harder, but when you're done, look at the mouse trace on the right. The mouse moved in a triangle this time. (Typing 'c' or 'C' restores the circle pattern)

To convince yourself that the mouse trace is really showing how you moved the mouse, type 'l' or 'L' to turn off the left side of the screen (you can also leave the left side showing if you like, or do this any time during a run). Move the mouse in any arbitrary pattern, and you'll see the mouse dot moving exactly the same way. The trace on the right always shows exactly how the mouse moved. Press the same key again to turn the left side back on, if you turned it off.

You can also draw other shapes on the left side, as long as they stay away from the center of the square (the program gets confused if you move the spot too close to the center). For example, with the figure set to "triangle," you can draw a diamond by connecting the midpoints of the square with straight lines. You can even draw an upside-down triangle. In either case the mouse record will still show a right-side-up triangle! This works best if you move the spot slowly and have a clear idea of the figure you're going to draw with the spot.

Paradox??

Some people think that controlling is a process of figuring out the actions needed to produce a wanted result, and then executing the actions. In the present case, that would mean figuring out how to move the mouse to create a square pattern on the left side of the screen. It would be very hard to figure out what actions are needed in this demonstration. Even if you look at the circle or triangle pattern of the mouse movements after one run and then reproduce the movements as best you can remember, you would not get a square (or diamond or upside-down triangle) on the left as a result. Don't bother trying this: the next paragraph should settle the question.

Even if you could see the exact shape of the required figure, tracing it would not reproduce the figure on the left. You can try this by typing 'f' to show the required figure, and then moving the mouse trace around its edge as accurately as you can. You will most definitely not get any particular shape on the left. Yet you can leave the figure on the right part of the screen and make the spot move around the square, and you'll see that your hand made the mouse-trace move around exactly the same figure.

We appear to have a paradox. Whatever you draw on the left side, you will see the intended pattern on the left and either a circle or a triangle on the right, showing how the mouse moved. But if you try to move the mouse as exactly as you can in the same way as that pattern on the right, even with an accurate picture of that figure as a guide, you will not get the figure you just drew on the left.

Explanations

This strange behavior is achieved by the action of an invisible control system inside the computer that tries to keep the mouse on the edge of a stored hidden pattern, the triangle or circle. This control system senses where a line from the center of the hidden pattern through the mouse crosses the boundary of the figure. If the mouse is inside the boundary on this line, the control system pushes on the spot being controlled by the person. It pushes, in this case, toward the center of the square that the person is tracing, so if the person did not resist, the spot would move inside the square. As this deviation begins to develop, the person detects it and moves the mouse to correct the error. The movement of the mouse required to correct the error moves the mouse radially outward from the center of the hidden triangle or circle. Since the mouse was initially inside the boundary of the hidden pattern, it therefore moves toward the boundary.

The same thing, only in reverse, happens if the mouse is outside the boundary of the hidden figure. The hidden control system pushes radially outward on the spot that the person is controlling, and to correct the error the person moves the mouse radially inward toward the edge of the hidden pattern.

In this way, the invisible control system uses disturbances of the spot that the person is controlling to make the mouse move radially inward or outward, thus keeping it close to the edge of the hidden pattern. Movements around the center of the hidden pattern are not resisted – only radial movements that would take the mouse off the edge of the hidden pattern are controlled.

The hidden control system uses "proportional-integral control", which means that in addition to the action being proportional to the error, the action continues to increase as long as there is any error at all. This action consists of pushing on the spot that the person is controlling, so if you displace the mouse a little and then let go, you will see the spot move quickly at first, and then keep moving slowing as long as no corrective motion of the mouse is made. The only way to stop the motion of the spot is to put the mouse dot exactly on the perimeter of the figure on the right (turn the figure on with 'f' to check this).

This is why you can't reproduce the mouse pattern exactly enough to recreate the shape on the left. The slightest error will make the spot move indefinitely far unless an error of the opposite sign is created. And the current position of the spot on the left depends on the cumulative history of mouse position errors, so you would have to recreate not only the current position error, but the whole history of position errors, including those you caused yourself by slight wobbles of your hand. You would be trying to reproduce exactly those same wobbles using the same wobbly hand, so it's hopeless. Can you make all the same tiny mistakes during two runs in a row?

A model of the person

If you type 'm' or 'M', the program will stop reading mouse positions from the mouse, and start reading simulated modeled mouse positions from an internal simulation of the person's control system (a first approximation – a much better model could be constructed). As soon as control is given to the model, the spot begins to trace around the square just as if a person were doing it. There are natural wobbles due to rounding errors and dynamic interactions between the model and the hidden control system; more random noise can be toggled on and off by typing 'n' or 'N'. The model simply substitutes for the person, with no change in the rest of the program..

This model shows some of the same problems that a real person has, because it is designed to be slightly unstable yet have high sensitivity to error. In order to match the model's errors more closely to the human errors, it would be necessary to adjust the model's parameters for the best fit to the human performance.

Also, if a real person tried many runs the performance would improve from run to run. The current model has no ability to improve its performance, but it could be given this ability quite easily. If we adjusted the model parameters to fit each of a series of runs by the same person, we would see some of the best-fit model parameters changing from one run to the next. We could then imitate the learning performance by simply having the model change its parameters from one run to the next, or we could try to simulate the learning process itself, making the parameters depend on comparisons of achieved performance with desired performance. Anyone is welcome to use this program as the basis for a research project of this kind (or any other).

Addendum

When 'm' is pressed, "NOISE" or "NO NOISE" is added to the top line on the screen, and noise can be turned on and off. The run is reset by toggling the noise on or off. This noise is a small random number added to the simulated mouse position.