A Sampler of Turtle Projects (in 2D)
(define (left-dragon size level)
  (cond ((= level 0) (fd size))
    (else (left-dragon size (- level 1))
      (lt 90)
      (right-dragon size (- level 1))))
)

(define (right-dragon size level)
  (cond ((= level 0) (fd size))
    (else (left-dragon size (- level 1))
      (rt 90)
      (right-dragon size (- level 1))))
)

(left-dragon 4 8)
(left-dragon 2 11)
(define (left-hilbert side level)
  (cond ((= level 0) 0)
        (else
         (left 90)
         (right-hilbert side (- level 1))
         (fd side)
         (right 90)
         (left-hilbert side (- level 1))
         (fd side)
         (left-hilbert side (- level 1))
         (right 90)
         (fd side)
         (right-hilbert side (- level 1))
         (left 90)))))

(define (right-hilbert side level)
  (cond ((= level 0) 0)
        (else
         (right 90)
         (left-hilbert side (- level 1))
         (fd side)
         (left 90)
         (right-hilbert side (- level 1))
         (fd side)
         (right-hilbert side (- level 1))
         (left 90)
         (fd side)
         (left-hilbert side (- level 1))
         (right 90)))))
(right-hilbert 4 5)
Inventing a new recursive pattern:

\[
\text{(define (shape1 side level)}
\begin{array}{l}
\quad (\text{cond (}(= \text{level} 0) 0) 0) \\
\quad (\text{else}) \\
\quad \quad (\text{fd (/ side 2)}) \\
\quad \quad (\text{rt 45}) \\
\quad \quad (\text{shape2}) \\
\quad \quad \quad (*) (\text{sqrt 2}) (/ side 2) (- \text{level} 1)) \\
\quad \quad (\text{lt 45}) \\
\quad \quad (\text{bk (/ side 2)}) \\
\quad \quad (\text{repeat 4}) \\
\quad \quad \quad (\text{shape3 (/ side 4) (- level} 1)) \\
\quad \quad \quad (\text{fd side}) (\text{rt 90})))))
\end{array}
\text{)}
\]

\[
\text{(define (shape2 side level)}
\begin{array}{l}
\quad (\text{cond (}(= \text{level} 0) 0) 0) \\
\quad (\text{else}) \\
\quad \quad (\text{repeat 2}) \\
\quad \quad \quad (\text{shape3 (/ side 2) (- level} 1)) \\
\quad \quad \quad (\text{fd (/ side 2)}) \\
\quad \quad \quad (\text{shape1 (/ side 2) (- level} 1)) \\
\quad \quad \quad (\text{fd (/ side 2)}) \\
\quad \quad \quad (\text{rt 90}) \\
\quad \quad \quad (\text{fd side}) \\
\quad \quad \quad (\text{rt 90}))))
\end{array}
\text{)}
\]

\[
\text{(define (shape3 side level)}
\begin{array}{l}
\quad (\text{cond (}(= \text{level} 0) 0) 0) \\
\quad (\text{else}) \\
\quad \quad (\text{repeat 4}) \\
\quad \quad \quad (\text{fd (/ side 3)}) \\
\quad \quad \quad (\text{shape1 (/ side 3) (- level} 1)) \\
\quad \quad \quad (\text{fd (* 2 (/ side 3))) (rt 90)})))))
\end{array}
\text{)}
\]
Superposing Turtle Walks

Imagine that we want to combine two turtle walks: a straight-line walk, for which each tiny step looks like this:

\[(\text{fd} \ 1)\]

And a circle walk for which each tiny step looks like this:

\[(\text{fd} \ 1) \ (\text{rt} \ 1)\]

What do we mean by "combine"? Think of one turtle walking on the back of the other, each maintaining its own independent compass.
Another way of thinking about this. Imagine we are combining these two motions in a wheel: the straight-line turtle is walking along carrying the picture of the wheel on its back.

As the straight-line turtle carries the wheel along, the circle turtle walks around the perimeter of the wheel.
The combination of these two walks gives you the path taken by a point on the rim of a rolling wheel—namely, a cycloid:

```
TO SUPERPOSE-TURTLE-WALKS walk1 heading1 walk2 heading2
REPEAT FOREVER
    [Save the turtle's starting position as START]
    [Do turtle-walk1 with heading1; save the heading at the end of this step as NEWHEADING1]
    [Do turtle-walk2 with heading2; save the heading at the end of this step as NEWHEADING2]
    [Find the turtle's final position after performing both walks; draw a line from START to this position]
heading1 <-- NEWHEADING1
heading2 <-- NEWHEADING2
```
Our superposition motion combines two components -- a turning (stationary) wheel, and a sliding (non-turning) wheel.
We'll represent each turtle-walk to be combined through its constituent "turtle-step", written as a zero-argument procedure:

```scheme
(define (straight-line-step) (fd 1))
(define (circle-step)
  (fd 1) (rt 1))
```
(define (repeat-superposed-walks
    proc1 proc2 hdg1 hdg2 iterations)
    (cond ((= iterations 0) 'done)
      (else
       ;; Get the turtle's starting coordinates
       (let ((startx (getposx))
              (starty (getposy)))
         ;; Pick up the pen and do step 1
         (pu)
         (seth hdg1)
         (proc1)
         (let ((endhdg1 (geth)))
           ;; Now do step 2
           (seth hdg2)
           (proc2)
           (let ((endx (getposx))
                  (endy (getposy))
                  (endhdg2 (geth)))
             ;; Combine the two steps by vector addition
             (setpos startx starty)
             (pd)
             (setpos endx endy)
             ;; and then recursively call this procedure
             ;; with the new values for headings
             (repeat-superposed-walks
              proc1 proc2 endhdg1 endhdg2
              (- iterations 1))))))
The cycloid:

(repeat-superposed-walks
  straight-line-step circle-step 90 90 1000)
(fd 0.5) (rt 1)

(fd 1.25) (rt 1)
The **cardioid:**

$(fd\ 1)\ (rt\ 1)$ and $(fd\ 1)\ (rt\ 2)$.  

The **nephroid:**

$(fd\ 1)\ (rt\ 1)$ and $(fd\ 1)\ (rt\ 3)$
(fd 1) (rt 1)  
(fd 1) (rt 8)  
(fd 1) (rt 3.5)  
(fd 1.5) (rt 4.2)
(fd 1) (rt 2.25)
(fd 1.5) (rt 4.2)
3-fold walks

4-fold walks

5-fold walks
Superpose “spiraloids” -- a cardioid, nephroid, and spirograph spiral
Predator Prey procedure
(to be repeated N times)

1. Set the prey turtle at preyx, prey, preyheading
2. Move the prey turtle by preystep
3. Update preyx prey
4. Set the predator turtle at predx, predy
5. Set the predator turtle heading towards preyx prey
6. Move the predator turtle by predatorstep
7. Update predx predy
“Synthetic Psychology” (or maybe “synthetic biology”): building minds and animals
Figure 1
Vehicle 1, the simplest vehicle. The speed of the motor (rectangular box at the tail end) is controlled by a sensor (half circle on a stalk, at the front end). Motion is always forward, in the direction of the arrow, except for perturbations.
Figure 2.
Vehicle 2, with two motors and two sensors; otherwise like Vehicle 1. The connections differ in a, b, and c.
Figure 3
Vehicles 2a and 2b in the vicinity of a source (circle with rays emanating from it). Vehicle 2b orients toward the source, 2a away from it.
Figure 4
Vehicle 3, with inhibitory influence of the sensors on the motors.
Figure 5
A multisensorial vehicle of brand 3c.
Figure 6
A nonlinear dependence of the speed of the motor $V$ on the intensity of stimulation $I$, with a maximum for a certain intensity.
Figure 7
Trajectories of vehicles of brand 4a around or between sources.
Figure 8
Various bizarre kinds of dependence of the speed of the motor (ordinate) on the intensity of stimulation (abscissa) in Vehicle 4b.