

Agendas and Tomorrow's Lab

John Chilton

April 9, 2007

Data Abstraction

Working with data well requires (at least) two kinds of thinking.

Building Data Abstractions

Cons Boxes, `car`, `cdr`, `set-car!`, `set-cdr!`.

Using Data Abstractions

Don't think in terms of cons boxes, think in terms of kinds of operations available for data.

Queue procedures:

- (make-queue)
- (empty-queue? q)
- (front-queue q)
- (insert-queue! q item)
- (delete-queue! q)

One way to think about queues:

← [a b 7 5] ←

A Question

Given a problem and some data, what do you think about first, how to build the abstractions for the data or how you are going to use the abstractions for the data?

Events over Time

```
(define (dance)
  (walk 1.0)    (look-left)
  (sleep 4000)
  (walk -1.0)   (look-right)
  (sleep 4000)
  (stop)        (look-center)
  (sleep 2000)
  (walk 1.0)    (look-left)
  (sleep 4000)
  (walk -1.0)   (look-right)
  (sleep 4000)
  (stop) (look-center))
```

A more intelligent approach.

```
(define (walk-forward steps)
  (if (> steps 0)
      (begin (walk 1.0) (look-left)
              (sleep 4000)
              (walk-backward (1- steps))))))
```

```
(define (walk-backward steps)
  (if (> steps 0)
      (begin (walk -1.0) (look-right)
              (sleep 4000)
              (wait (1- steps))))))
```

```
(define (wait steps)
  (if (> steps 0)
      (begin (stop) (look-center)
              (sleep 2000)
              (walk-forward (1- steps))))))
```

A more intelligent approach.

```
(define (walk-forward steps)
  (if (> steps 0)
      (begin (walk 1.0) (look-left)
              (sleep 4000)
              (walk-backward (1- steps))))))
```

```
(define (walk-backward steps)
  (if (> steps 0)
      (begin (walk -1.0) (look-right)
              (sleep 4000)
              (wait (1- steps))))))
```

```
(define (wait steps)
  (if (> steps 0)
      (begin (stop) (look-center)
              (sleep 2000)
              (walk-forward (1- steps))))))
```

Limitiations?

Timeline Data Structure

Associate events or actions with times. Different times can have multiple events associated with it. Some procedures add actions to the agenda, and a separate procedure pulls items out of the agenda, runs them, and handles time delays.


```
(define (walk-forward)
  (walk 1.0) (look-left)
  (after-delay 4000 walk-backward))
```

```
(define (walk-backward)
  (walk -1.0) (look-right)
  (after-delay 4000 walk-forward))
```

```
(define (led1)
  (ear-led 1.0 0.0 1.0)
  (after-delay 5000 led2))
```

```
(define (led2)
  (ear-led 0.0 1.0 0.0)
  (after-delay 5000 led1))
```

Separates actions from actual timing code.

```
(define the-agenda (make-agenda))

(define (after-delay time action)
  (add-to-agenda! (+ time (current-time the-agenda))
                  action
                  the-agenda))

(define (reset-the-agenda!)
  (set! the-agenda (make-agenda)))

(define (propagate)
  (if (not (empty-agenda? the-agenda))
      (let ((first-item (first-agenda-item the-agenda)))
        (first-item)
        (remove-first-agenda-item! the-agenda)
        (propagate))))
```

What is the output?

```
(define (proc1)
  (display "PROC1")
  (newline)
  (after-delay 2 proc1))
```

```
(define (proc2)
  (display "PROC2")
  (newline)
  (after-delay 3 proc2))
```

```
(after-delay 0 proc1)
(after-delay 1 proc2)
(after-delay 6 reset-the-agenda!)
(propagate)
```

- Lab will be in room EE/CS 2-140 this week
- Must use agenda, dance-off winners will receive 1 bonus point
- 4 teams per lab - feel free to bring in laptops to divide effort

```
;; Loading helper procedures...  
(load "agenda.scm")  
(load "connect.scm")  
  
;; Put dance moves in here with after-delay  
  
;; Put a stopping time if you want  
; (after-delay ... reset-the-agenda!)  
  
;; runs the dance.  
(propagate)
```

Lab Bonus Points

- 1 Point for participating, 3-4 points for putting a movie together
- If you are interested in putting a movie together e-mail me (chil0060@umn.edu) as soon as possible, will accept first 3 or 4 people from each lab
- Must include description of algorithms and/or data structures used
- If you would like to not be filmed for whatever reason also e-mail me