```
;;;; Template-03.rtf, cr 11 march 2012 by rha on pro
;;;; code tab of Template.nlogo, rev 5.0, original comments stripped, new brief comments added
;;;; look in the NetLogo Dictionary under H to read about 23 hubnet commands

breed [ students student ] ;; server-side proxies for clients

students-own
[
  user-id    ;; useful for all hubnet activities
  step-size ;; specific to this activity
]

to startup
  hubnet-reset ;; starts the system, includes login process (look up in dictionary)
end

to setup ;; avoid clear-all
  clear-patches
  clear-drawing
  clear-output
  ask turtles ;; that is, students (proxies for clients)
  [
    set step-size 1
    hubnet-send user-id "step-size" step-size ;; server sends value to tag of named client (see dictionary)
  ]
  reset-ticks
end

to go
  listen-clients ;; waits for messages from clients (defined below)
  every 0.1 [ tick ] ;; 10x/second, increments tick counter
end

;; HubNet Procedures (6)

to listen-clients
  while [ hubnet-message-waiting? ] ;; look for message from a client in the queue, if there is:
  [
    hubnet-fetch-message            ;; get the message (3 parts) from the queue (dictionary)
    ifelse hubnet-enter-message?  ;; checks for a new client  (dictionary)
    [ create-new-student ]            ;; (defined below)
    [
      ifelse hubnet-exit-message?  ;; checks if client has departed  (dictionary)
        [ remove-student ]              ;; (defined below)
        [ ask students with [user-id = hubnet-message-source] ;; if client still active (dictionary)
          [ execute-command hubnet-message-tag ] ;; (defined below)
      ]
    ]
  ]
end

to create-new-student
  create-students 1
  [
    set user-id hubnet-message-source ;; reports which client  (dictionary)
    set label user-id
    set step-size 1
    send-info-to-clients   ;; (defined below)
  ]
end
```

```
to remove-student
  ask students with [user-id = hubnet-message-source] [ die ]
end

to execute-command [command]
  if command = "step-size"
    [
      set step-size hubnet-message       ;; uses command and value sent by client
      stop
    ]
  if command = "up" [ execute-move 0 stop ]
  if command = "down" [ execute-move 180 stop ]
  if command = "right" [ execute-move 90 stop ]
  if command = "left" [ execute-move 270 stop ]
end

to send-info-to-clients ;; turtle procedure (server-side proxy sends info to client)
  hubnet-send user-id "location" (word "(" pxcor "," pycor ")") ;; note 3 parts
end

to execute-move [new-heading]
  set heading new-heading
  fd step-size
  send-info-to-clients
end

;;;; end of code
```