

```

;; =====
;; Larghezza del corridoio nelle farfalle REALE - Nota 7 Cap 10
;; =====

globals []      ;; variabile globale
patches-own [elevazione usata?] ;; variabili proprie
turtles-own [casella-partenza ax ay ]

;; ho creato una slider con variabile globale numero-farfalle
;; ho creato una slider con variabile globale p , probabilità (vedi info)

to setup
  ca
  file-open "ElevationData.txt"
  while [not file-at-end?]
  [
    let prossimo-X file-read
    let prossimo-Y file-read
    let prossima-elevazione file-read
    ask patch prossimo-X prossimo-Y [set elevazione prossima-elevazione]
  ]
  file-close

  ask patches
  [
    let min-elevazione min [elevazione] of patches
    let max-elevazione max [elevazione] of patches
    set pcolor scale-color green elevazione min-elevazione max-elevazione ;;asigna alla casella un verde di tonalità dal
    chiaro allo scuro, in proporzione al valore di elevazione (min, max)
    set usata? false
  ]
  crt numero-farfalle      ;; crea le farfalla da ingresso
  [
    set size 3      ;; dimensione delle farfalle 3
    setxy (Posizione-iniziale + random 10) (Posizione-iniziale + random 10)      ;; le inizializza nelle coordinate distanti
    al massimo 10 da Posizione iniziale
    set casella-partenza patch-here
    if traccia? = true [pen-down]      ;; lascia la traccia degli spostamenti. Disabilitata per vedere in chiaro il corridoio
    ( interruttore on/off )
  ]
  reset-ticks
end

to go
  ask turtles [move]
  plot larghezza-corridoio
  tick      ;; conta un passo nel tempo . Spuntare il bottone go in forever
  if ticks >= 1000      ;; ferma il tempo a 1.000 passi
  [
    let larghezza-corridoio-finale larghezza-corridoio
    output-print word "Larghezza del corridoio: " larghezza-corridoio-finale ;; stampa su monitor il valore finale
    stop]
end

```

```

to move          ;; sottomodello. Ho realizzato una slider nell'interfaccia con la variabile q per modificarla a
piacere da ingresso
  ;;if elevazione >=
  ;; [elevazione] of max-one-of neighbors [elevazione]
  ;;[stop]
  ;; q è la probabilità che la farfalla si muova alla casella vicina più alta
  ifelse random-float 1 < q  ;; se il numero causale generato < q
  [uphill elevazione]      ;; la farfalla si muove verso una casella vicina di più alto valore
  [move-to one-of neighbors] ;; altrimenti si muove in modo casuale - vedi Info
  set usata? true          ;;
  set pcolor white
end
to-report larghezza-corridoio
  let caselle-visitate count patches with [usata?]
  let distanza-media mean [distance casella-partenza] of turtles
  if distanza-media = 0 [set distanza-media 1]
  report caselle-visitate / distanza-media
end

```