

Il Mondo della Tartaruga - Programmazione LOGO

Prof. Domenico Beneventano
Università di Modena e Reggio Emilia

6 maggio 2003

Un pó di storia

Il Logo é un ambiente di apprendimento, non solo uno dei piú potenti linguaggi di programmazione oggi disponibili per personal computer.

Il piú popolare dei prodotti Logo é la Turtle, la tartaruga. Originariamente era un piccolo robot azionato da motori che la facevano muovere su spessi fogli di cartone e su cui disegnava con una penna che poteva essere abbassata o alzata dal foglio.

Era comandata attraverso una piccola tastiera collegata ad computer, con semplici istruzioni .



1

Presupposti Teorici del LOGO

I presupposti teorici del Logo si trovano nella teoria dell'apprendimento elaborata da Jean Piaget.

- **Ambiente Egosintonico*** Questo linguaggio si propone di offrire all'allievo un ambiente "egosintonico" sfruttando alcune idee e creazioni (es il LISP) dell'intelligenza artificiale.
- **Ambiente Stimolante** al cui interno é possibile *imparare* attraverso la manipolazione di una serie di materiali e di strumenti, realizzando cosí una serie di esperienze che sfociano nella costruzione di concetti profondi, acquisiti *sperimentando*.
- **Costruzione di Algoritmi** trasferendo le capacità di riflessione acquisite sperimentando, si arriva alla costruzione di algoritmi che, proposti dapprima all'automa Turtle, vengono successivamente trasferiti nella costruzione di piú astratte strutture informatiche che fanno uso di parole, stringhe e vettori.

*In accordo con il corpo

La Tartaruga ai tempi nostri

La tartaruga, Turtle, é diventata un oggetto grafico sullo schermo del computer, vive una *vita elettronica animata* istruita attraverso il linguaggio di programmazione Logo .



Geometria Tradizionale V_s Geometria della Tartaruga

La geometria tradizionale é costruita su astrazioni:

- **Un punto senza dimensioni**
- **Una linea che ha una lunghezza ma non ha spessore**

concetti difficili da afferrare per i bambini.

Geometria Tradizionale V_s Geometria della Tartaruga

La tartaruga é invece un oggetto reale che puó essere manipolato, chiamiamolo piú esattamente un oggetto di *transizione* tra il concreto, il robot, e l'astratto, apparendo come un piccolo triangolo sullo schermo che puó essere comandato e istruito ad eseguire compiti diversi, legati soprattutto alla creazione di disegni di vario tipo e colorati.

Geometria Tradizionale V_s Geometria della Tartaruga

- **Geometria analitica** si fonda su un sistema esterno di riferimento e la posizione del punto é individuata dalle coordinate.
- **Geometria della tartaruga** é egosintonica, in accordo con il corpo. La tartaruga si muove come tu ti muovi, pensi al tuo movimento e puoi ordinare alla tartaruga di compiere di simili.

La geometria della tartaruga non vuole sostituire la geometria tradizionale, é un punto d'ingresso alternativo alle astrazioni geometriche e matematiche. Si presta ugualmente bene ad apprendere alcuni concetti di base sia per i bambini che per gli adulti.

Facciamo muovere la tartaruga

Digita PCS e premi Enter per visualizzare la tartaruga

La tartaruga si trova nella sua HOME nel centro dello schermo.

Per far muovere la tartaruga dobbiamo dare:

- **Comando:** Cosa fare.
- **Argomento:** Quante volte farlo.
- **Enter:** Muoviti.

Per far scrivere / non far scrivere la tartaruga dobbiamo dare:

- **GIU** : Penna giù.
- **SU** : Penna su.

Comandi di Movimento

Avanti (AV) Destra (DE)
Indietro (IN) Sinistra (SI)

In Micromondi, si può scrivere **indifferentemente**

- il comando per intero (esempio: Avanti 100) **oppure** la sua abbreviazione (esempio: AV 100).
- in maiuscolo (esempio: AVANTI 100) oppure in minuscolo (esempio: avanti 100)

IMPORTANTE : In queste dispense si usano a volte queste ulteriori abbreviazioni:

Avanti (A), Destra (D), Indietro (I) e Sinistra (S)

Tali abbreviazioni **non sono accettate** da Micromondi: quindi in Micromondi occorre sostituirle (ad esempio, A 100 deve essere sostituita con avanti 100 oppure AV 100)

N.B. ricordati di premere Enter dopo ogni comando!!

8

Esercitiamoci

Proviamo a disegnare due semplici poligoni:



Notate delle similitudini nei comandi necessari a disegnare le due figure ?

9

Il Comando RIPETI

Il comando RIPETI ci aiuta nel disegno di poligoni regolari, vediamo la sintassi:

RIPETI < n > [Comando1 Arg1 Comando2 Arg2 ComandoN ArgN]

Poligono	Lati	Angolo	Istruzioni
	3		RIPETI 3 [AV DE]
	4		RIPETI 4 [AV DE]

10

Il comando RIPETI

Riesci a vedere una similitudine tra Lati e Angolo, quale relazione esiste tra il numero di lati e l'angolo per ogni poligono ?

Se N = numero di lati, allora Angolo = $360/N!$ semplice no!!

11

Procedure

Definire una procedura vuol dire, in parole semplici, assegnare un nome ad una sequenza di istruzioni, per poterle richiamare in un secondo momento digitando il nome della procedura.

```
PER SQUARE
RIPETI 4 [ A 50 D 90]
FINE
```

Per richiamare la procedura basta digitare:
SQUARE (e dare Enter)

12

Variabili

Una procedura può essere richiamata con dei parametri: dei valori da associare ad una variabile presente nella procedura.

Una variabile è un'area di memoria a cui viene associato un nome e nella quale si può memorizzare un valore.

```
PER TRIANGOLO :SIZE
RIPETI 3 [A :SIZE D 120]
FINE
```

La prima riga (PER TRIANGOLO :SIZE) dice al computer di associare ad un'area di memoria il nome SIZE e di memorizzare in essa il numero che trova accanto a TRIANGOLO nella chiamata della procedura.

Il comando AV :SIZE dice alla tartaruga di muoversi in avanti di tanti passi quanto è il valore che si trova nell'area di memoria chiamata SIZE.

13

Ancora sulle procedure

Una procedura può essere richiamata all'interno di un'altra procedura:

```
PER MODELLO
RIPETI 3 [TRIANGOLO 100 D 120]
FINE
```

In questo esempio la procedura MODELLO richiama la procedura TRIANGOLO passandogli come parametro d'ingresso 100.

14

Lavorare con più variabili

Una procedura può essere definita con più di 1 parametro d'ingresso. I parametri devono seguire il nome della procedura ed avere nomi distinti:

```
PER RETTANGOLO :H :W
RIPETI 2 [A :H D 90 A :W D 90]
FINE
```

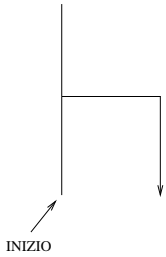
In questo esempio la procedura RETTANGOLO viene definita con due parametri H (height = altezza) e W (Width = larghezza) e ripete 2 volte il disegno:

```
PER RETTANGOLO :H :W
RIPETI 2 [A :H D 90 A :W D 90]
FINE
```

15

Divertiamoci!!!

Proviamo a disegnare una sedia, come in figura:



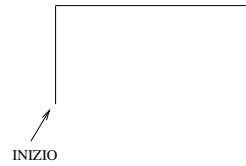
Proviamo a definire la procedura **sedia**:

```
PER SEDIA
AVANTI 40
INDIETRO 20
DESTRA 90
AVANTI 16
DESTRA 90
AVANTI 20
FINE
```

16

Divertiamoci!!!!

Dopo la sedia ci vuole un tavolo:



Per disegnarlo dobbiamo prima allontanarci dalla sedia:

```
SU
SINISTRA 90
AVANTI 24
SINISTRA 90
GIU
```

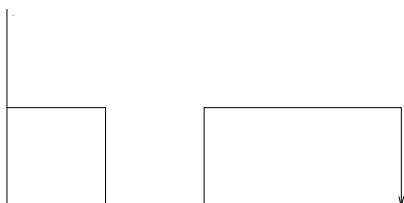
17

Divertiamoci!!!!

Adesso definiamo la procedura TAVOLO:

```
PER TAVOLO
AVANTI 50
DESTRA 90
AVANTI 100
DESTRA 90
AVANTI 50
FINE
```

Adesso abbiamo una sedia ed un tavolo facile no



18

Domanda e Risposta

In Micromondi possiamo interagire con la tartaruga, lei fa le domande con il comando:

```
DOMANDA [ testo della domanda ]
```

La nostra risposta sarà contenuta nella variabile speciale `risposta`, vediamo un esempio (cliccate Invio dopo ogni comando):

```
DOMANDA [ETA' ?]
MOSTRA RISPOSTA
```

dopo il comando `DOMANDA [ETA' ?]` appare una finestra, immettendo un valore e cliccando su OK sulla console comandi (quella in basso a destra) apparirà il valore da noi immesso.

19

Disegniamo un Poligono

Abbiamo già visto come disegnare un poligono:

```
PER POLIGONO :NLATI :LUNG
RIPETI :NLATI [ AV :LUNG DE 360 / :NLATI ]
FINE
```

Adesso vogliamo fare in modo che sia la tartaruga a chiederci il numero di lati e la lunghezza di ciascun lato del nostro poligono:

20

Disegniamo un Poligono

La tartaruga deve quindi formulare due domande:

- **Quanti lati ?** DOMANDA [*Quanti lati ?*]
- **Dimensione del lato :** DOMANDA [*Dimensione lato :*]

21

Disegniamo un Poligono

Vediamo di costruire la procedura CHIEDIPOLIGONO, definiamo l'istruzione di inizio e fine procedura:

```
PER CHIEDIPOLIGONO
```

```
FINE
```

All'interno dobbiamo far sì che la tartaruga formuli 2 domande; poiché la nostra risposta andrà sempre nella variabile *risposta*, prima di formulare la seconda dobbiamo memorizzare il suo valore in una variabile da noi definita.

22

Disegniamo un Poligono

```
PER CHIEDIPOLIGONO
DOMANDA [NUMERO LATI ?]
ASSEGNA "NUMEROLATI RISPOSTA
...
..
.
FINE
```

Cosa abbiamo fatto qui ?, la tartaruga ci chiede il numero di lati e poi assegna alla variabile di nome NUMEROLATI (notare l'uso delle ") il valore di *risposta* (che essendo una variabile speciale non ha bisogno dei :)

23

Disegniamo un Poligono

Il principio é lo stesso per chiedere la dimensione del lato, per cui scriveremo all'interno della procedura CHIEDIPOLIGONO:

```
PER CHIEDIPOLIGONO
DOMANDA [NUMERO LATI ?]
ASSEGNA "NUMEROLATI RISPOSTA
DOMANDA [DIMENSIONE LATO :]
ASSEGNA "DIMENSIONELATO RISPOSTA
...
FINE
```

Se non avessimo scritto

ASSEGNA "NUMEROLATI RISPOSTA, dopo la seconda domanda avremmo perso il numero di lati.

24

Disegniamo un Poligono

A questo punto abbiamo tutti i dati necessari per chiamare la procedura POLIGONO:

```
PER CHIEDIPOLIGONO
DOMANDA [NUMERO LATI ?]
ASSEGNA "NUMEROLATI RISPOSTA
DOMANDA [DIMENSIONE LATO :]
ASSEGNA "DIMENSIONELATO RISPOSTA
POLIGONO :NUMEROLATI :DIMENSIONELATO
FINE
```

25

Differenza fra Nome e Valore di una variabile

La tartaruga di Micromondi distingue il nome di una variabile dal valore ad essa associato attraverso l'uso delle " e del :

- **Nome variabile** "miavariabile
- **Valore variabile** :miavariabile

E' facile capire quando usare le " o i :

Usiamo le " nell'istruzione ASSEGNA
(es: ASSEGNA "MIAVARIABILE 2)

Usiamo i : nelle

- **Espressioni Matematiche** :miavariabile - 1
- **Definizioni di procedure con parametri**
PER POLIGONO :NLATI :LUNG.
- **Nelle chiamate di procedure con parametri**
POLIGONO :NLATI :LUNG
(vedi esempio in CHIEDIPOLIGONO)

26

Esercitiamoci con i cerchi

Un cerchio lo possiamo vedere come un poligono di 360 lati, per cui possiamo definirne la procedura nel seguente modo:

```
PER CERCHIO :RAGGIO
RIPETI 360 [AV :RAGGIO DE 1]
FINE
```

Per far si che la tartaruga disegni tanti cerchi uno dentro l'altro, dobbiamo variare il raggio all'interno del ripeti

Quello che vogliamo fare é partire da un valore iniziale di raggio e cambiarne il valore ad ogni cerchio disegnato.

27

Esercitiamoci con i cerchi

```
PER CERCHI :ncerchi
ASSEGNA "raggio 2
RIPETI :ncerchi [ cerchio :raggio assegna "raggio :raggio - .3 ]
FINE
```

- **PER CERCHI :ncerchi** Definiamo una procedura di nome CERCHI che ha bisogno del *valore* (quindi :) del parametro ncerchi.
- **ASSEGNA "raggio 2** Assegniamo alla variabile di nome raggio il valore di partenza 2.
- **RIPETI :ncerchi** Ripetiamo un numero di volte pari al **valore** di ncerchi (il parametro d'ingresso) le istruzioni racchiuse tra parentesi quadre.
 - **cerchio :raggio** richiamiamo la procedura cerchio passandole il **valore** della variabile raggio.
 - **assegna "raggio :raggio - 0.3** Assegniamo alla variabile di nome raggio il **valore** di raggio meno 0.3 (:raggio -0.3 é un'espressione matematica)

28

dettiamo delle condizioni alla tartaruga

In LOGO é possibile condizionare il comportamento della tartaruga al verificarsi o meno di una condizione.

Una condizione é un espressione di cui é possibile affermare se sia vera o falsa (es: $5 > 2$, $2 + 1 = 3$)

La tartaruga riesce a valutare una condizione con il comando SE

```
SE condizione [ istruzioni se condizione é vera ]
```

Ad esempio:

```
PER SETIMUOVI
DOMANDA [NUMERO PASSI ?]
SE RISPOSTA > 50 [AV RISPOSTA]
FINE
```

Se alla tartaruga diamo un numero di passi maggiore di 50 la condizione RISPOSTA > 50 sar  vera e la vedremo muoversi in avanti di un numero di passi pari al valore da noi dato come risposta, altrimenti rimarr  ferma perch  non sapr  cosa fare

29

Alternative

con il comando SE la tartaruga fa qualcosa solo se la condizione é vera, per dargli un'alternativa dobbiamo usare il comando SEALTRIMENTI

```
SEALTRIMENTI condizione
[ istruzioni se condizione é vera ]
[ istruzioni se condizione é falsa ]
```

Ad esempio:

```
PER SETIMUOVI2
DOMANDA [NUMERO PASSI ?]
SEALTRIMENTI RISPOSTA > 50
[AV RISPOSTA]
[IN RISPOSTA]
FINE
```

La tartaruga valuta la condizione risposta > 50, se é vera va in avanti altrimenti va indietro di un numero di passi pari al valore da noi dato come risposta.

30

Esempio avanzato

Ipotizziamo di avere tre numeri: num1, num2 e num3 e di volerne individuare il maggiore:

Prima di tutto dobbiamo valutare chi tra num1 e num2 é il pi  grande :

```
SEALTRIMENTI :num1 > :num2
```

nel caso in cui sia num1 il pi  grande (condizione vera) dobbiamo confrontarlo con num3 :

```
SEALTRIMENTI :num1 > :num2
[
  SEALTRIMENTI :num1 > :num3
```

```
]

```

31

Esempio avanzato

Se la condizione é vera mostriamo num1 se é falsa mostriamo num3:

```
SEALTRIMENTI :num1 > :num2
[
  SEALTRIMENTI :num1 > :num3
  [MOSTRA :num1]
  [MOSTRA :num3]
]
```

32

Esempio avanzato

Finora abbiamo esaminato il caso in cui la condizione :num1 > :num2 fosse vera, dobbiamo vedere cosa fare se é falsa:

```
SEALTRIMENTI :num1 > :num2
[
  SEALTRIMENTI :num1 > :num3
  [MOSTRA :num1]
  [MOSTRA :num3]
]
[
  :num1 < :num2
]
```

33

Esempio avanzato

In questo caso dobbiamo valutare num2 con num3 come fatto in precedenza con num1:

```
SEALTRIMENTI :num1 > :num2
[
  SEALTRIMENTI :num1 > :num3
  [MOSTRA :num1]
  [MOSTRA :num3]
]
[
  SEALTRIMENTI :num2 > :num3
  [MOSTRA :num2]
  [MOSTRA :num3]
]
```

34

Esempio avanzato

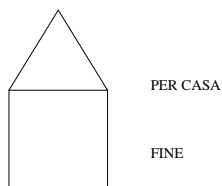
Quindi la nostra procedura massimo sará:

```
PER MASSIMO :num1 :num2 :num3
SEALTRIMENTI :num1 > :num2
[
  SEALTRIMENTI :num1 > :num3
  [MOSTRA :num1]
  [MOSTRA :num3]
]
[
  SEALTRIMENTI :num2 > :num3
  [MOSTRA :num2]
  [MOSTRA :num3]
]
FINE
```

35

ESERCIZIO: La casa

Costruiamo la nostra casetta:



36

Le mura

Le mura sono rappresentabili attraverso un quadrato:



RIPETI 4 [AV 100 DE 90]

37

Il tetto

A questo punto la tartaruga é tornata a casa, deve usare una scala per salire sul tetto:

AV 100 DE 30

Ora possiamo costruire il nostro tetto, che altro non é che un triangolo:



RIPETI 3 [AV 100 DE 120]

Dopo aver costruito il tetto dobbiamo far scendere la tartaruga dal tetto:

SI 30

IN 100

38

Conclusioni sulla casa

ricapitolando la nostra casa é definita dalla seguente procedura:

PER CASA

RIPETI 4 [AV 100 DE 90] (le mura)

AV 100 DE 30 (la salita sulle scale)

RIPETI 3 [AV 100 DE 120] (il tetto)

SI 30 IN 100 (la discesa dalle scale)

FINE

39

Varianti alla casa

Possiamo cercare di definire 2 procedure per la salita e la discesa dal tetto:

PER SALITA
AV 100
DE 30
FINE

PER DISCESA
SI 30
IN 100
FINE

A questo punto la casa é costruita cosí:

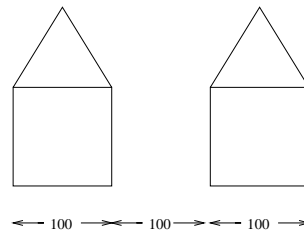
PER CASA
RIPETI 4 [AV 100 DE 90] (le mura)
SALITA (la salita sulle scale)
RIPETI 3 [AV 100 DE 120] (il tetto)
DISCESA (la discesa dalle scale)
FINE

40

Il paese

La tartaruga nella sua casa si sente sola, utilizziamo allora la procedura CASA per costruire un paese in cui possano venire ad abitare altre tartarughe:

PER PAESE
RIPETI 4 [CASA SU DE 90 AV 200 SI 90 GIU]
FINE



41

Variante al paese

Volendo possiamo definire una procedura spazio:

PER SPAZIO
SU
DE 90
AV 200
SI 90
GIU
FINE

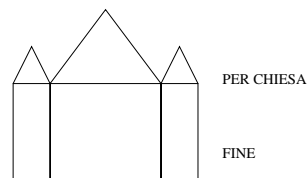
In questo modo la procedura PAESE cambia:

PER PAESE
RIPETI 4 [CASA SPAZIO]
FINE

42

La chiesa

Proviamo adesso a costruire una chiesa:



43

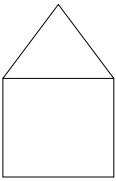
Componenti di una chiesa

La chiesa che vogliamo costruire é composta da:

- **2 Torri:**



- **1 casa:**



44

La Torre

La torre é composta da un rettangolo e da un triangolo, il triangolo lo sappiamo già disegnare non il rettangolo

PER RETTANGOLO

RIPETI 2 [AV 200 DE 90 AV 100 DE 90]

FINE

Modifichiamo le procedure SALITA e DISCESA per far salire la tartaruga sulla torre:

PER SALITA :a

AV :a

DE 30

FINE

PER DISCESA :a

SI 30

IN :a

FINE

45

La procedura TORRE

A questo punto la procedura TORRE é definita come:

PER TORRE

RETTANGOLO

SALITA 200

RIPETI 3 [AV 100 DE 90]

DISCESA 200

FINE

46

La procedura CHIESA

La chiesa puó essere cosí definita:

PER CHIESA

TORRE

DE 90 AV 100 SI 90

CASA

DE 90 DE 100 SI 90

TORRE

FINE

Notate l'utilitá delle procedure, conn 5 righe ho costruito una chiesa!!!

47