



Sistemi Multi-agente

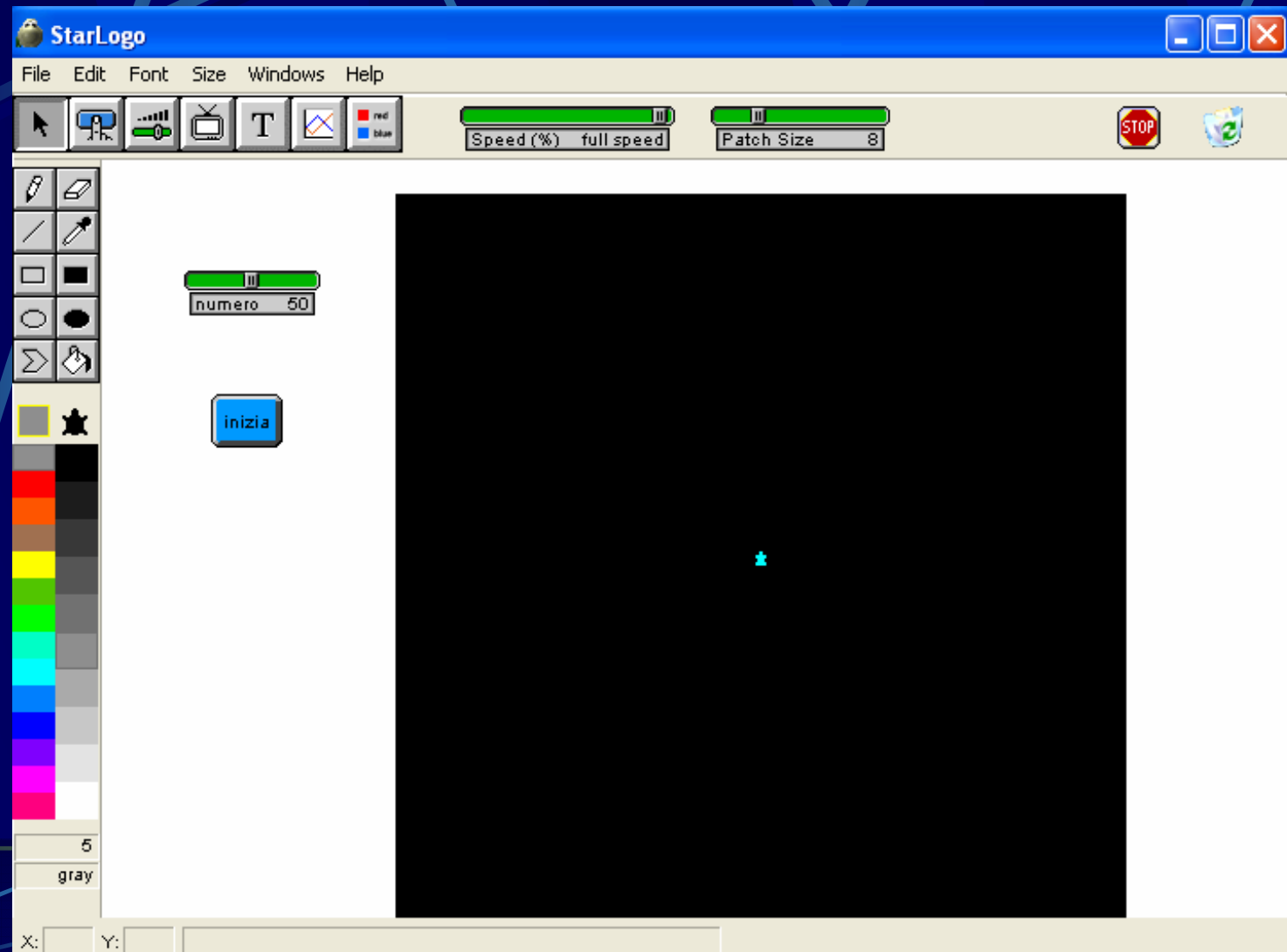
Multi-agente

- Il concetto di **comunicazione** con altri agenti porta ai sistemi multi-agente
- I sistemi attualmente più diffusi sono
 - StarLogo
 - NetLogo

StarLogo

- Si basa su Logo. Si programmano tre entità:
 - tartarughe (anche di specie diversa)
 - patches (dove vivono le tartarughe)
 - osservatore (che può controllarli entrambi).

Il mondo StarLogo



Tartarughe



Ecco lo stato di una tartaruga
who, shape, color, ...

Possiamo intervenire nel centro
di comando e farla operare.
Ma è più interessante dare
ordini a tutte le tartarughe

Le tartarughe vivono su una
griglia di patch

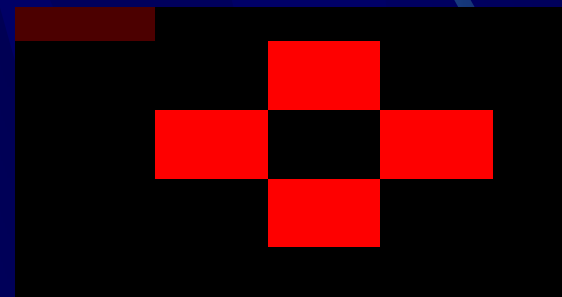
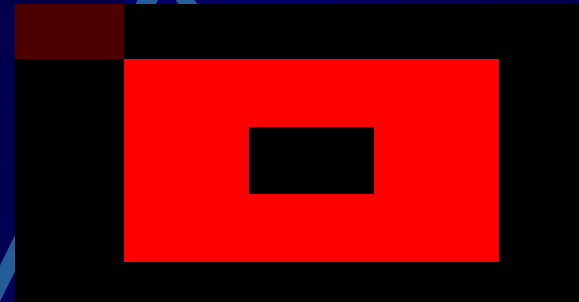
Patch



- Ecco un patch e un comando che crea una tartaruga verde
- I patch hanno coordinate. Al centro è (0,0)
- L'ampiezza del patch può essere variata
- *screen-height* e *screen-width* forniscono la dimensione in numero di patch

patch

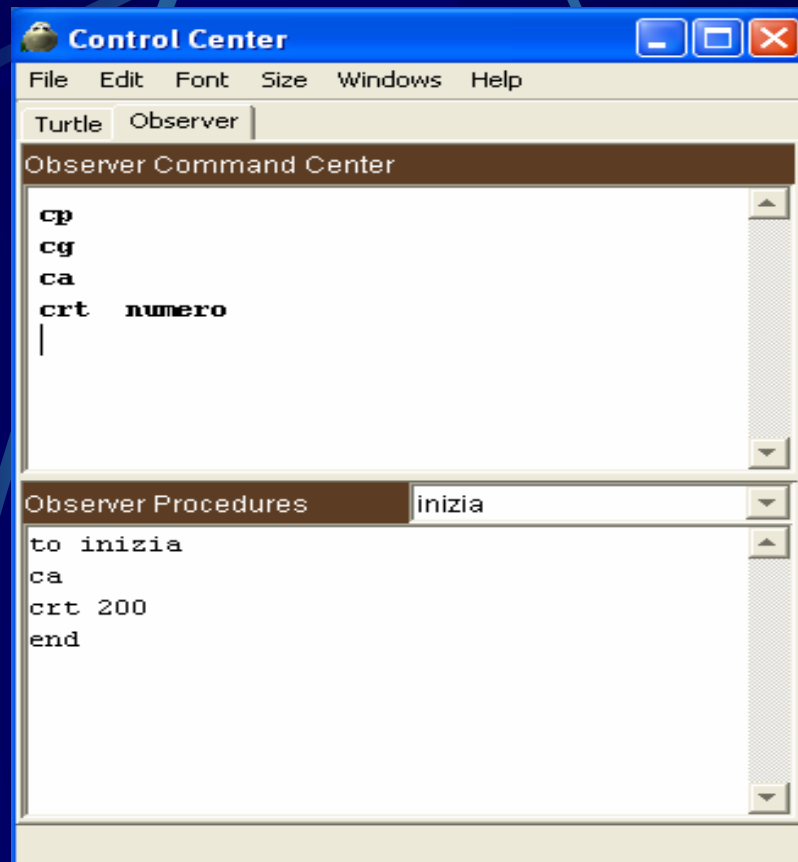
- Ogni patch ha un intorno di altri patch (8 o 4 secondo il nostro interesse)
- Essendo il mondo un “toro” ai bordi l’intorno si espande



osservatore

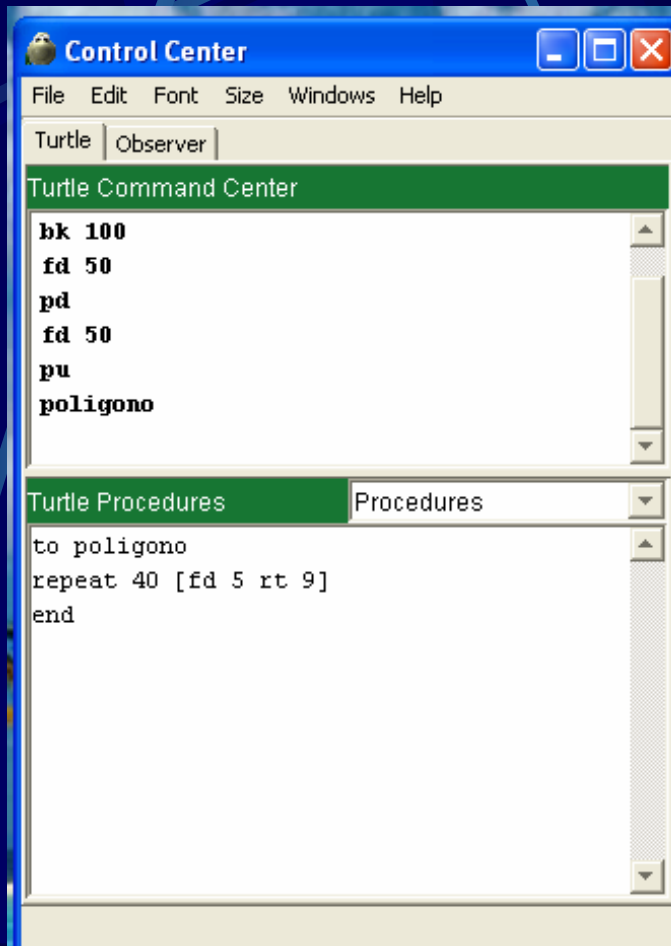
- Può creare le tartarughe e dare ordini a tartarughe e patch. Ma non è il supremo leader. L'evoluzione del mondo che viene generato non è sempre sotto il suo controllo

Interfaccia osservatore



- Divisa in due parti

Interfaccia tartarughe



- Anche qui due parti in cui si comandano tutte le tartarughe.

La metafora in StarLogo

- Tartarughe vivono sui patches: mondo bidimensionale che è un automa cellulare. L'enfasi non è sulle figure che vengono tracciate con la penna ma su quelle che le tartarughe formano con la propria presenza.

L'idea

- to starlog-circle
create-turtles 2000
setheading random 360
forward 40
end
- La sintassi del programma precedente non è corretta. Vogliamo solo sottolineare l'idea della geometria **delle** tartarughe

Primi passi

- die
- crt 20
- create-and-do 5 [fd 3]
- hatch [setc blue fd 1]
- setc blue
- if ycor < 0 [setc green]
- if color = green [pd]
- if color = green [repeat 36 [fd 1 right 10]]
- cg
- stamp red

esempi

- `seth 90` (90 = a destra. 0 = in su)
- `stamp red` - il patch sotto la tartaruga è rosso
- `setcolor green` – il colore della tartaruga
- `setxcor 0`
- `setycor screen-height` - sull'asse y
- `setycor screen-width` – sull'asse x
- `setxy mouse-xcor mouse-ycor` – sul cursore
- `seth (random 360)` - direzione random

ask-patches [...]

- setpc red
- setpc pc + 1
- setpc (random 256)
- repeat 100 [setpc pc + 1]
- setpc xcor
- setpc xcor + ycor
- if xcor < 0 [setpc red] -
- if (distance 5 5) < 10 [setpc green] -
- if (distance mouse-xcor mouse-ycor) < 4 [setpc blue]
- ifelse ycor < 0 [setpc red] [setpc blue]

Primi passi

- ca
- crt 200
- Alternare fd 40 e bk 100 (tarta)
- ask-patches [if xcor > 0 [setpc blue]]
- pu fd 50 (tarta)
- cp

Ancora ...

- ca
- crt 50
- fd 10
- ifelse (who mod 2 = 0) [setc blue] [setc red]
- repeat 36 [fd 2 rt 10]
- ifelse (who mod 2) = 0 [repeat 36 [fd 2 rt 10]] [repeat 36 [fd 2 lt 10]]
- seth 0 e ripetere..

Colori ...

- to setup
ca
crt 10
ask-turtles [pd]
end

con patch-size 2

go ripetuto e repeat .. [go]

repeat 200 [go]

- to go
rt 1
hatch [setc who / 30
repeat 36 [fd 4 rt 10]
die]
- end

altri passi ...

- crt 100
- fd 200
- fd random 200
- setc blue
- seth 0 fd 30
- pd fd 50
- if ycor < 0 [setc green]
- turtles-own[passo]
- set passo random 30
- fd passo
- ca
- ask-patches [setpc yellow]
- ask-patches-with [xcor > 20]
[setpc green]
- ask-patches [if (distance 10
20) < 15 [setpc white]]
- ask-patches [if (distance 10
20) = 0 [setpc blue]]
- cg
- patches-own [deposito]
- ask-patches [set deposito 0]
- ask-patches [if (distance 0 0) <
15 [set deposito 30]]

diffuse qualcosa percentuale

con $0 \leq \text{percentuale} \leq 1$) Esempi:

- **ask-patch -at 0 0 [set chemical 50]**

con l'ordine dell'osservatore

diffuse chemical 1

avremo in (0,0) chemical=0, negli otto patch vicini chemical= 6.25 mentre con l'ordine

diffuse chemical 0.5

avremo in (0,0) chemical=25, negli otto patch vicini chemical= 3.125.

scale-pc color variabile min max

- scala il colore con cui si presenta il patch in funzione dei valori min e max. Se il valore di variabile eccede i limiti, allora il patch si presenta nero (variabile $<$ min) o bianco (variabile $>$ min).

Comportamenti emergenti

- Non esiste un “leader” o un “pacemaker” ma l’intelligenza sociale è diffusa in semplici regole dei singoli
 - Le formiche
 - Slime mold o *Dictyostelium discoideum*
 - Le termiti

Una popolazione

- Iniziale
 - ca
 - ask-patches [if (random 20) < 5 [setpc yellow]]
- count-patches-with [(distance 0 0) < 5]
- count-pc yellow
- count-turtles
- nsum *variabile vicini* (considera, per ogni mattonella, il valore di *variabile* nelle otto mattonelle che la circondano e li somma mettendo il risultato nella variabile *vicini*)

Primo esempio

- Le termiti. Passare a termite.ppt
 - Ipotizziamo un comportamento di questo tipo:
 - Se non ho un chicco di grano lo cerco
 - Quando lo trovo salto lontano
 - Col chicco di grano in bocca cerco un deposito, ovvero un posto dove vedo un chicco di grano cui lo affianco.
 - Poi salto lontano e ricomincio