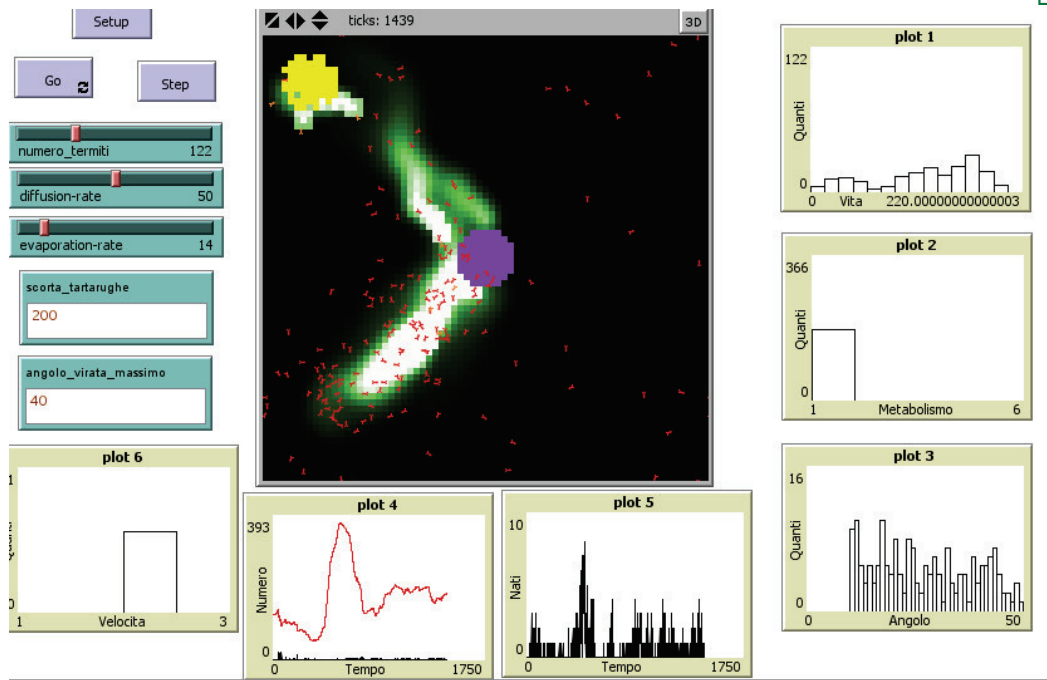


# Evoluzione



□ Esempio di interfaccia, compresi alcuni parametri di controllo

← Scorta

← Metabolismo

← Angolo virata

↑ velocità    ↑ vivi e morti    ↑ nati

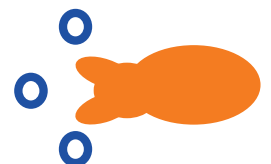
# Le formiche

□ Ogni agente (formica) ha come stati interni:

- angolo\_virata    - angolo di virata
- velocita        - numero passi di «vagare»  $\in \{1, 2\}$
- scorta          - deve essere  $> 0$  per sopravvivere
- **sgl\_scorta**    - «ricchezza» minima per fare figli
- **eta**            - un contatore, che vale 0 appena nati
- metabolismo    - viene sottratto alla scorta,  $\in \{1, 2, 3, 4, 5\}$

□ Gli agenti ora

- se hanno **scorta**  $< 0$  vengono eliminati
- se uno porta cibo (**color** = orange) ed entrambi hanno **scorta**  $>$  **sgl\_scorta** e hanno **più di 30 anni** tentano di fare un figlio
  - se ci riescono, la **loro** scorta viene calata di **scorta\_tartarughe/10**



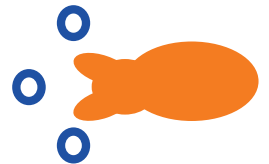
## Le formiche

```
if scorta < 0
[
  set morti morti + 1
  die
]
```

```
if color = orange [
  if scorta > sgl_scorta and eta > 30 [
    ]
  ]
```

### □ Gli agenti ora

- se hanno **scorta** < 0 vengono eliminati
- se uno porta cibo (**color** = *orange*) ed **entrambi** hanno **scorta** > **sgl\_scorta** e **hanno più di 30 anni** tentano di fare un figlio
  - se ci riescono, la **loro** scorta viene calata di *scorta\_tartarughe*/10



## La riproduzione

### □ Ogni agente (formica) che rispetta le condizioni per poter fare figli deve:

- controllare se intorno ci sono altre formiche
- ordinarle per scorta
- verificare se la prima con più di 30 anni supera la soglia (non importa se carica o meno di cibo)

### □ Se va tutto bene

- deve chiedere alla patch di fare nascere un'altra formica con
  - **metabolismo** scelto a caso fa «padre» e «madre»
  - **velocita** scelta a caso fa «padre» e «madre»
- perde un decimo della scorta iniziale

## La riproduzione

- Ogni agente (formica) che rispetta le condizioni per poter fare figli deve:
  - controllare se intorno ci sono altre formiche
  - ordinarle per scorta
  - verificare se la prima con più di 30 anni supera la soglia (non importa se carica o meno di cibo)

```
if any? other turtles-here with [eta > 30] [  
  if [scorta] of item 0 sort-on [scorta] other turtles-here with [eta > 30] > sgl_scorta[  
  
  ]  
]
```

- Ordine utile:

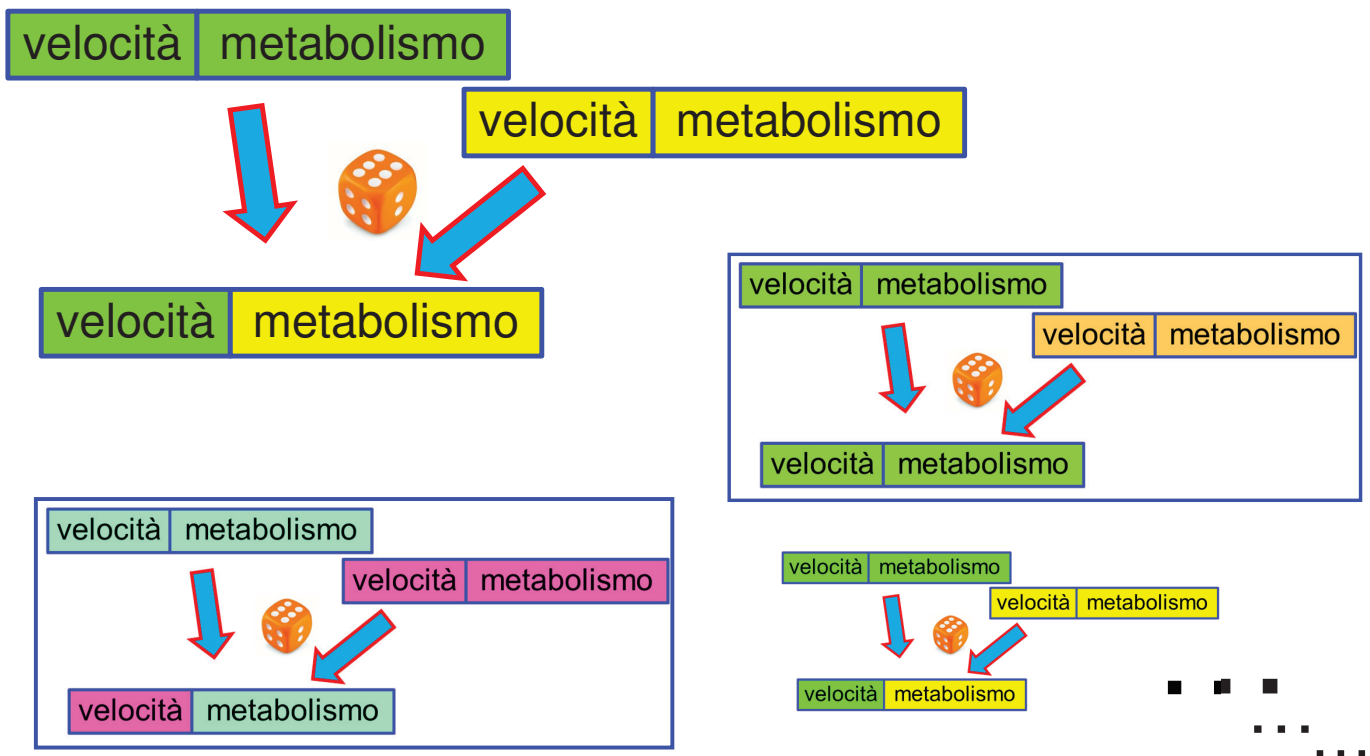
```
show [scorta] of item 0 sort-on [scorta] other turtles-here with [eta > 30]
```

## La riproduzione

```
let met_padre metabolismo  
let met_madre [metabolismo] of item 0 sort-on [scorta] other turtles-here with  
[eta > 30]  
...  
ask patch-here [  
  sprout 1 [  
    ....  
    set nati nati + 1  
  ]  
]  
set scorta scorta - (scorta_tartarughe / 10)
```

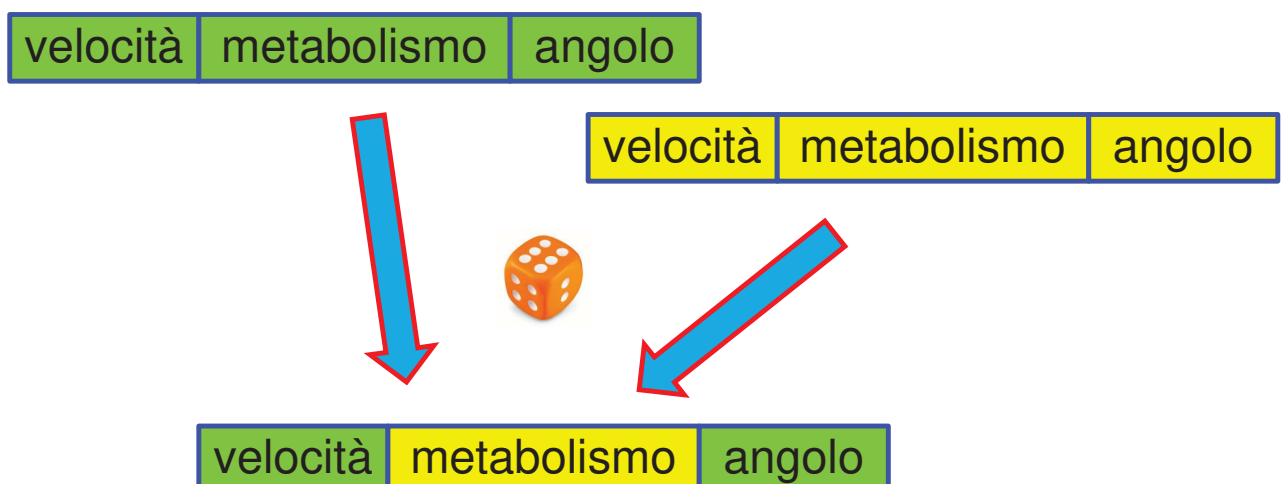
- Se va tutto bene
  - deve chiedere alla patch di fare nascere un'altra formica con
    - **metabolismo** scelto a caso fa «padre» e «madre»
    - **velocita** scelta a caso fa «padre» e «madre»
  - perde un decimo della scorta iniziale

## Riassunto situazione



## Variente (facoltativa)

- Mettere soggetto ad evoluzione anche l'angolo di virata



## *La sequenza di azioni*

- **Ad ogni passo, quindi:**
  - **Le formiche fanno la loro mossa**
    - se devono, «muoiono»
    - se possono, si riproducono
  - **L'ambiente effettua la sua dinamica**
    - diffusione (osservatore), evaporazione (ogni patch)
  - **Il cibo viene ricaricato**
    - ogni 70 passi una delle tre fonti di cibo viene rimpiazzata
    - per un ciclo completo ogni 210 passi

## *Riassunto dei valori dei parametri*

- **numero termiti iniziale:** 100
- **diffusione:** [20,50] **default: 50**
- **evaporazione:** [10,20] **default: 14**
- **scorta iniziale tartarughe:** 200
- **sgl\_scorta:** (scorta iniziale tartarughe)/1.6
- **«prezzo» per riproduzione:** (scorta iniziale tartarughe)/10
- **età minima riproduzione:** 30

### **Formiche iniziali:**

- **angolo virata :** casuale fra 10 e 40
- **Velocità:** casuale fra 1 e 2
- **metabolismo:** casuale fra 1, 2, 3, 4, 5