1. Εισαγωγή στη NetLogo

Αποκεντρωμένος Υπολογισμός και Μοντελοποίηση

24 Οκτωβρίου 2022

Στοιχεία επικοινωνίας

Ιστότοπος: `https://eclass.upatras.gr/courses/CEID1220/`

Υπεύθυνος μαθήματος: Κωνσταντίνος Τσίχλας, Επίκουρος Καθηγητής ΤΜΗΥΠ
e-mail: ktsichlas@ceid.upatras.gr
Τηλ: 2610-996908
Γραφείο: 1ος όροφος, Γρ. 4

Βασίλης Θωμόπουλος, Υποψήφιος Διδάκτορας ΤΜΗΥΠ
e-mail: thomopul@ceid.upatras.gr
Γραφείο: Εργαστήριο 2Ε2, 2ος Όροφος

NetLogo για Μοντελοποίηση Σύνθετων Συστημάτων

- Περιβάλλοντα μοντελοποίησης για προσομοίωση που είναι προγραμματισμένα για φυσικά και κοινωνικά φαινόμενα.
- Κατάλληλη για μοντελοποίηση σύνθετων συστημάτων, τα οποία εξελίσσονται με την πάροδο του χρόνου.
- Εκατοντάδες ή και χιλιάδες ανεξάρτητοι πράκτορες λειτουργούν ταυτόχρονα.
- Εξερεύνηση της σύνδεσης μεταξύ της συμπεριφοράς μικροεπίπεδου των μεμονωμένων ατόμων και τα μοτίβα μακρο-επιπέδου, όπως προκύπτουν από την αλληλεπίδραση πολλών ατόμων.

Εύκολο στη χρήση περιβάλλον ανάπτυξης εφαρμογών

- Άνοιγμα των προσομοιώσεων και πειραματισμός με αυτές.
- Δημιουργία προσαρμοσμένων μοντέλων: γρήγορος έλεγχος υποθέσεων για συστήματα που οργανώνονται από μόνα τους.
- Βιβλιοθήκη μοντέλων: μεγάλη συλλογή από προκαθορισμένες προσομοιώσεις σε φυσικές και κοινωνικές επιστήμες που δύνανται να χρησιμοποιηθούν και να τροποποιήσουν με απλή γλώσσα scripting.
- Γραφικό περιβάλλον, φιλικό προς το χρήστη.

Σύντομη Ιστορία

1. Η θεωρία με εκπαίδευση LOGO [Papert, 1993] βασίζεται στον κονστρουκτιβισμό του Piaget (πρακτική δημιουργία και δοκιμή εννοιών). Απλή γλώσσα που προέρχεται από το LISP. Γραφήματα τύπου χελώνας και εξερεύνηση των «μικρόκοσμων».
2. StarLogo [Resnick, 1991], OpenStarLogo, MacStarLogo, StarLogoT, StarLogo TNG Agent-based γλώσσα προσομοίωσης. Διερεύνηση της συμπεριφοράς αποκεντρωμένων συστημάτων μέσω ταυτόχρονου προγραμματισμού εκατοντάδων χελωνών.
3. NetLogo [Wilensky, 2015, Tisue, 2004]. Περαιτέρω επέκταση της StarLogo (συνεχείς συντεταγμένες turtles, πολλαπλές πλατφόρμες, δικτύωση, κ.α.). Το πιο δημοφιλές σήμερα (αυξανόμενη συνεταιριστική βιβλιοθήκη μοντέλων).

Γιατί NetLogo;

- Εύκολη στη χρήση
- Ισχυρή
- Εξαιρετική τεκμηρίωση
- Πιθανότητα αλληλεπίδρασης με μοντέλο σε αληθινό χρόνο
- Αυτόματη εξερεύνηση παραμετρικού χώρου.
- Ανοικτού κώδικα και ελεύθερη
- Πολυπλατφορμική με ικανότητα εκτέλεσης μοντέλων online
- Ισχυρή υποστήριξη και ενεργής κοινότητα
- Αφθονία ποιοιτικών πηγών
- Ικανότητα επέκτασης για συμπλήρωση εξειδικευμένων αναγκών
- Εύκολη να συνδεθεί με άλλα λογισμικά.

Πώς να μάθεις τα βασικά της NetLogo;

Το εύρος του σταδίου αυτού, εκτείνεται από μερικές ώρες έως και δύο μέρες ανάλογα με την ικανότητα στον προγραμματισμό.

Μπορεί να υλοποιηθεί ακολουθώντας τις παρακάτω εργασίες: Κατεβάζεις και εγκαθιστάς τη NetLogo ακολουθώντας τις οδηγίες: `https://ccl.northwestern.edu/netlogo/`.

Κάνε τα τρία tutorials στο εγχειρίδιο χρήσης της NetLogo, δηλαδή:

- Tutorial #1: Models Tutorial
- Tutorial #2: Commands Tutorial
- Tutorial #3: Procedures

Βασικές έννοιες της NetLogo

Κάθε patch είναι μέρος του φόντου ή του «τοπίου». Οι χελώνες είναι οντότητες που κινούνται γύρω ή και πάνω από τα patches.

Πράκτορας (Agent)

Πράκτορας είναι μια οντότητα με (τουλάχιστον) τα ακόλουθα χαρακτηριστικά / ιδιότητες:

1. Αυτονομία
2. Αντιδραστικότητα (Reactivity)
3. Προληπτικότητα (Pro-activity)
4. Κοινωνικές δεξιότητες – Κοινωνικότητα (Sociability)

Πράκτορας: δεύτερος ορισμός

Πράκτορας είναι μια φυσική ή εικονική οντότητα που:

1. είναι σε θέση να ενεργεί μέσα σε ένα περιβάλλον
2. μπορεί να αλληλεπιδρά άμεσα με άλλα πράκτορες,
3. καθοδηγείται από ένα σύνολο αντιφάσεων (μεμονωμένοι στόχοι, ικανοποίηση ή ζωντανές λειτουργίες για βελτιστοποίηση).
4. κατέχει τις δικές του πηγές.
5. είναι σε θέση να αντιληφθεί το περιβάλλον περιορισμένης έκτασης,
6. έχει μερική γνώση για το περιβάλλον (πιθανώς καμιά)
7. κατέχει δεξιότητες και προσφέρει υπηρεσίες.
8. μπορεί να αναπαραχθεί μόνος του και
9. έχει μια συμπεριφορά που στοχεύει στην ικανοποίηση των στόχων του πράκτορα, λαμβάνοντας υπόψη πόρους, δεξιότητες, αντίληψη και επικοινωνίες.

Τέσσερις τύποι πρακτόρων της NetLogo

1. Οι **χελώνες** μετακινούνται πάνω από τα patches, όχι απαραίτητα στο κέντρο τους. Έχουν δεκαδικές συντεταγμένες (xcor, ycor) και προσανατολισμό (κατεύθυνση).
2. Τα **patches** δεν μετακινούνται, σχηματίζουν ένα wrap-around 2D grid. Έχουν ακέραιες συντεταγμένες (pxcor, pycor).
3. Οι **Σύνδεσμοι** (προσανατολισμένοι ή μη) ανάμεσα σε δύο χελώνες. Δεν παρουσιάζεται σε αυτό το εργαστήριο
4. Ο **παρατηρητής - χρήστης**, μπορεί να δημιουργήσει νέες χελώνες. Μπορεί να έχει πρόσβαση ανάγνωσης/εγγραφής όλους τους πράκτορες και τις μεταβλητές.

Μεταβλητές (Variables)

Μεταβλητή: χώρος αποθήκευσης τιμών, όπως αριθμοί ή κείμενο.

Τύποι μεταβλητών Τρεις τύποι μεταβλητών:

1. Καθολικές μεταβλητές (Global variables)
2. Μεταβλητές πράκτορα (Agent variables)
3. Τοπικές μεταβλητές (Local variables)

Καθολικές μεταβλητές (Global Variables)

Ορισμός: Σε μια καθολική μεταβλητή μπορούν να έχουν πρόσβαση όλοι οι πράκτορες. Αυτό πρέπει να δηλωθεί πριν από όλες τις διαδικασίες. Παράδειγμα:

```
globals [
    max_energy ;; maximum energy
    speed ;; traversed cells per simulation step
]
```

Σχόλια (comments)

Ένα σχόλιο ξεκινά με τους χαρακτήρες «;;» και τελειώνει στο τέλος της γραμμής.

Μεταβλητές πρακτόρων (Agent Variables)

Κάθε χελώνα και patch έχει το δικό του σύνολο μεταβλητών, που ονομάζεται μεταβλητές πρακτόρων. Η τιμή μιας μεταβλητής μπορεί να διαφέρει από πράκτορα σε πράκτορα. Παράδειγμα:

```
turtles-own [
    life
]
```

Τοπικές μεταβλητές (Local Variables)

Ορισμός: Μια μεταβλητή ορίζεται και είναι προσβάσιμη μόνο μέσα σε ένα διαδικασία.

Πεδίο εφαρμογής: οι πιο στενές αγκύλες, ή η ίδια η διαδικασία.
Παράδειγμα:

```
to permuter [val1 val2]
    let tmp val1
    set val1 val2
    set val2 tmp
end
```

Προκαθορισμένες μεταβλητές πράκτορα (Predefined Agent Variables)

Για τις χελώνες

- χρώμα
- κατεύθυνση - προσανατολισμός σε μοίρες
- ετικέτα - όνομα
- σχήμα
- μέγεθος
- pxcor - συντεταγμένη κατά μήκος του άξονα x
- pycor - συντεταγμένη κατά μήκος του άξονα y
- ποιος – αναγνωριστικό

Προκαθορισμένες μεταβλητές πράκτορα (Predefined Agent Variables)

Για τα patches

- pcolor - χρώμα
- pxcor - συντεταγμένη κατά μήκος του άξονα x
- pycor - συντεταγμένη κατά μήκος του άξονα y
- plabel - ετικέτα

Χρώματα (Colors)

Τα χρώματα καθορίζονται από:

1. έναν αριθμό που κυμαίνεται από [0 - 140]
2. τον κωδικό RGB
3. ένα προκαθορισμένο όνομα (π.χ. green)

	black = 0										white = 9.9
gray = 5	0	1	2	3	4	5	6	7	8	9	9.9
red = 15	10	11	12	13	14	15	16	17	18	19	19.9
orange = 25	20	21	22	23	24	25	26	27	28	29	29.9
brown = 35	30	31	32	33	34	35	36	37	38	39	39.9
yellow = 45	40	41	42	43	44	45	46	47	48	49	49.9
green = 55	50	51	52	53	54	55	56	57	58	59	59.9
lime = 65	60	61	62	63	64	65	66	67	68	69	69.9
turquoise = 75	70	71	72	73	74	75	76	77	78	79	79.9
cyan = 85	80	81	82	83	84	85	86	87	88	89	89.9
sky = 95	90	91	92	93	94	95	96	97	98	99	99.9
blue = 105	100	101	102	103	104	105	106	107	108	109	109.9
violet = 115	110	111	112	113	114	115	116	117	118	119	119.9
magenta = 125	120	121	122	123	124	125	126	127	128	129	129.9
pink = 135	130	131	132	133	134	135	136	137	138	139	139.9

Εντολές (Commands)

Ενέργειες που πρέπει να πραγματοποιήσουν οι πράκτορες. Η εντολή δεν επιστρέφει μια τιμή (void συναρτήσεις).
Δήλωση:

```
to <command_name> [<parameter1> <parameter2> . . . ]
    <commands>
end
```

Κλήση

```
<command_name> <argument1> <argument2>
```

Πάραδειγμα (Command)

```
to INITIALISE
    clear-all ;; destroy all agents
    create-turtles 100 ;; create 100 turtles
end
```

Reporters

Ορισμός: Σύνολο διαδικασιών για τον υπολογισμό μιας τιμής. Ο ρεπόρτερ απαντά σε μια τιμή (αληθινές συναρτήσεις). Η τιμή αναφέρεται με την λέξη - κλειδί αναφοράς (ισοδύναμη με την εντολή «return» σε άλλες γλώσσες).

Πάραδειγμα (Reporters)

```
to-reprt abs [ nb ] ;; Replies the absolute value of nb
    if else nb >= 0
    [ report nb ]
    [ report (-nb ) ]
end
```

Primitive Procedures

Ενσωματωμένες εντολές ή ρεπόρτερ (λέξεις-κλειδιά γλώσσας). Μερικές έχουν μια συντομευμένη μορφή: create-turtles $< - >$ crt, clear-all $< - >$ ca

Asks

Ορισμός: Καθορίστε τις εντολές που θα εκτελούνται από χελώνες ή patches.

- Asking all turtles: ask turtles [fd 50]
- Asking all patches: ask patches [<commands>]
- Asking one turtle: ask turtle <id> [<commands>]
- Asking one patch: ask patch <x> <y> [<commands>]

Agent Sets

Ορισμός υποσυνόλου πρακτόρων. Δημιουργία ενός υποσυνόλου με πράκτορες που επικυρώνουν μια δεδομένη συνθήκη.

Παραδείγματα Agent Sets

- Υπό συνθήκη <agentset> with [<condition>] π.χ. όλες οι κόκκινες χελώνες: turtles with [color = red]
- Απόσταση <agentset> in-radius<distance> π.χ. όλες οι χελώνες εντός 3 patches: turtles in-radius 3
- Με σχετική θέση <agentset> at-points[[x1 y1] [x2 y2] ...]

Λειτουργίες σε Agent Sets

1. Ζητήστε από τέτοιους πράκτορες να εκτελέσουν μια εντολή: ask <agentset> [...]. Παράδειγμα: όλες οι κόκκινες χελώνες 1 βήμα μπροστά: ask turtles with [color = red] [fd 1]
2. Ελέγξτε αν υπάρχουν πράκτορες στο σετ: any? <agentset>. Παράδειγμα: εμφάνισε αν υπάρχουν κόκκινες χελώνες: show any? turtles [color = red]
3. Μετρήστε τους πράκτορες μέσα σε ένα σετ: count <agentset>. Παράδειγμα: εμφάνισε όλους τις κόκκινες χελώνες: show count turtles [color = red]

Breeds

Ένα «φυσικό» είδος σετ πρακτόρων (άλλα είδη εκτός από τις χελώνες).
Για δήλωση νέου Breed ακολουθούμε τη σύνταξη:
breed [<breeds> <breed>]
Η πρώτη τιμή είναι ο πληθυντικός και η δεύτερη ο ενικός.
Παράδειγμα δήλωσης νέου Breed ποντικιών: breed [mice mouse]

Αυτόματα Primitives για Breeds

- create-<breeds> <number> Δημιουργία του δοσμένου αριθμού πρακτόρων του συγκεκριμένου Breed
- <breed>-own Δήλωση μεταβλητών για όλα τα μέλη της συγκεκριμένης φυλής (breed).
- κ.α.

Δείτε περισσότερα εδώ:
`http://ccl.northwestern.edu/netlogo/docs/dict/breed.html`

Γραφική διεπαφή της NetLogo

Τρεις καρτέλες:

1. Διεπαφή: εργαλεία προβολής και UI για την προσομοίωση.
2. Πληροφορίες: Πληροφορίες κειμένου για το προσομοιωμένο μοντέλο.
3. Κώδικας: πηγαίος κώδικας NetLogo του προσομοιωμένου μοντέλου.

Model Library

Η NetLogo παρέχει πολλά μοντέλα έτοιμα προς χρήση από τη Βιβλιοθήκη της. Επιλέξτε το στοιχείο μενού: > File > Models Library. Κάντε διπλό κλικ στο επιλεγμένο μοντέλο για φόρτωση. Αντίστοιχα και για τη NetLogo online.

Βασικά του σχεδιασμού μοντέλων (1)

1. Ορίζουμε τις:
   - διαφορετικές φυλές χελωνών
   - παγκόσμιες μεταβλητές
   - μεταβλητές πράκτορα

Βασικά του σχεδιασμού μοντέλων (2)

2 Ορίζουμε τη διαδικασία έναρξης:
   - Η εντολή συνήθως ονομάζεται setup
   - Η εντολή συνήθως καλείται από το κουμπί setup
   - Η εντολή δημιουργεί τους πράκτορες και ενεργοποιεί τις μεταβλητές.

Βασικά του σχεδιασμού μοντέλων (3)

3 Ορίζουμε τη δυναμική του συστήματος:
- Η εντολή ονομάζεται go
- Η εντολή καλείται με το κουμπί go
- Οι υπο-διαδικασίες καλούνται στην εντολή go

Βασικά εκτέλεσης των μοντέλων (1)

```
to setup
    clear-all
    ...
    reset-ticks
end
```

- Αρχικοποίηση της προσομοίωσης
- clear-all Κλήση όλων των συναρτήσεων αρχικοποίησης
- reset-ticks Θέτει τον χρόνο t σε 0

Βασικά εκτέλεσης των μοντέλων (2)

```
to go
    ...
    display
    tick
end
```

- Εκτέλεση ενός βήματος της προσομοίωσης (τελικά σε βρόχο).
- display Ενημέρωση της οθόνη διεπαφής NetLogo.
- tick Αυξήστε την τρέχουσα ώρα t κατά ένα.

Διαχείριση του χρόνου προσομοίωσης t

- Ο χρόνος t είναι μια πληροφορία που έχει δηλωθεί εσωτερικά από τη NetLogo. Το μοντέλο είναι υπεύθυνο για την εξέλιξη του χρόνου.
- Λήψη του χρόνου t: **ticks**
- Αύξηση του χρόνου κατά ένα: **tick**
- Αύξηση του χρόνου κατά $< dt >$: **tick-advance<dt>**
- Αρχικοποίηση του χρόνου: **reset-ticks** (συνήθως στο τέλος του setup)

Παράδειγμα - Termites

https://ccl.northwestern.edu/netlogo/models/Termites
http://www.netlogoweb.org/launch#http://ccl.northwestern.edu/netlogo/models/models/Sample%20Models/Biology/Termites.nlogo

Termites

Οι τερμίτες ακολουθούν 3 απλούς κανόνες:

1. Ψάξε γύρω σου για ένα ξυλάκι και σήκωσέ το.
2. Ψάξε γύρω σου για ένα σωρό από ξυλάκια.
3. Ψάξε γύρω σου για ένα αδειανό σημείο στο σωρό και ρίξε εκεί το ξυλάκι.

Κώδικας I

```
to setup
  clear-all
  set-default-shape turtles "bug"
  ;; randomly distribute wood chips
  ask patches
  [ if random-float 100 < density
    [ set pcolor yellow ] ]
  ;; randomly distribute termites
  create-turtles number [
    set color white
    setxy random-xcor random-ycor
    set size 5  ;; easier to see
  ]
end

to go  ;; turtle procedure
  search-for-chip
  find-new-pile
  put-down-chip
end

to search-for-chip  ;; turtle procedure -- "picks up chip" by turning orange
  ifelse pcolor = yellow
  [ set pcolor black
    set color orange
    fd 20 ]
  [ wiggle
    search-for-chip ]
end

...
```

Κώδικας II

```
...

to find-new-pile  ;; turtle procedure -- look for yellow patches
  if pcolor != yellow
  [ wiggle
    find-new-pile ]
end

to put-down-chip  ;; turtle procedure -- finds empty spot & drops chip
  ifelse pcolor = black
  [ set pcolor yellow
    set color white
    get-away ]
  [ rt random 360
    fd 1
    put-down-chip ]
end

to get-away  ;; turtle procedure -- escape from yellow piles
  rt random 360
  fd 20
  if pcolor != black
    [ get-away ]
end

to wiggle ; turtle procedure
  fd 1
  rt random 50
  lt random 50
end


; Copyright 1997 Uri Wilensky.
; See Info tab for full copyright and license.
```

Επόμενα βήματα

- Κατεβάστε τη NetLogo στον υπολογιστή σας από εδώ: `https://ccl.northwestern.edu/netlogo/download.shtml`
- Εξερευνήστε τα μοντέλα που υπάρχουν στη βιβλιοθήκη της NetLogo `https://ccl.northwestern.edu/netlogo/models/`
- Το αρχείο και το βίντεο της παρουσίασης θα είναι διαθέσιμο στο E-class
- Απορίες στο thomopul@ceid.upatras.gr

Ευχαριστώ για την προσοχή σας!