



Η LOGO

στην εκπαιδευτική διαδικασία

Τ. Μικρόπουλος - Τ. Λαδιάς



Πανεπιστήμιο Ιωαννίνων













1997

ISBN 960-233-040-6

7 Η ουσία των πραγμάτων

Η χελώνα ζωγραφίζει τη λέξη «ΓΗ»

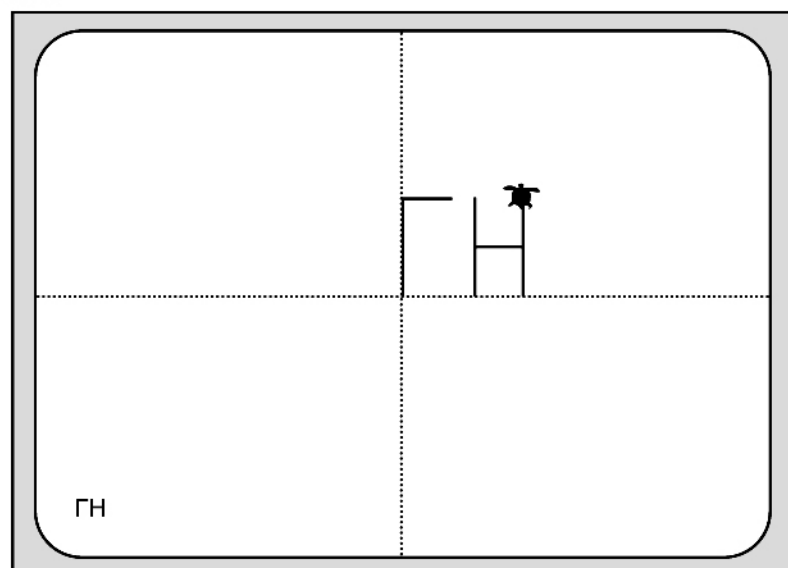
Η χελώνα εκτελώντας τις οδηγίες μας ζωγραφίζει τα γράμματα Γ και Η.

 FORWARD 40	<p>Οι οδηγίες προς τη χελώνα στη γλώσσα logo λέγονται εντολές. Το FORWARD είναι μια εντολή που διατάζει τη χελώνα να κινηθεί προς τα εμπρός. Ο αριθμός που ακολουθεί (το 40), δηλώνει τα βήματα που θα κάνει η χελώνα.</p> <p>Μια άλλη εντολή είναι η RIGHT που στρίβει τη χελώνα δεξιά. Ο αριθμός που ακολουθεί (το 90), δηλώνει τη γωνία σε μοίρες.</p> <p>Η χελώνα αφήνει ίχνη που φανερώνουν τη διαδρομή που έκανε. Η χελώνα μπορεί να μην αφήνει ίχνη με την εντολή PENUP.</p> <p>Με την εντολή PENDOWN η χελώνα μπορεί να αφήνει ίχνη. Η γλώσσα logo έχει ένα συγκεκριμένο λεξιλόγιο, δηλαδή διαθέτει προκαθορισμένες εντολές. (FORWARD, RIGHT, PENUP, PENDOWN).</p>	 RIGHT 180
 RIGHT 90		 FORWARD 20
 FORWARD 20		 RIGHT 90 FORWARD 20
 PENUP FORWARD 10		 RIGHT 90 FORWARD 20
 RIGHT 90		 RIGHT 180
 PENDOWN FORWARD 40		 FORWARD 40

Η αναγκαιότητα της διαδικασίας

Αν χρειαστεί να ξανασχεδιάσουμε τη ΓΗ θα πρέπει να επαναλάβουμε μια-μια όλες τις προηγούμενες εντολές. Για να το αποφύγουμε μπορούμε να φτιάξουμε μια δικιά μας διαταγή που να σχεδιάζει το ΓΗ. Η διαταγή που ορίζεται μ' αυτόν τον τρόπο λέγεται διαδικασία (procedure). Μια διαδικασία χρησιμοποιώντας το λεξιλόγιο της γλώσσας, πρέπει να δίνει σαφείς και αναλυτικές οδηγίες στον υπολογιστή για τον τρόπο που θα εκτελέσει κάποιο έργο.

```
TO ΓΗ
FORWARD 40
RIGHT 90
FORWARD 20
PENUP
FORWARD 10
PENDOWN
RIGHT 90
FORWARD 40
RIGHT 180
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 180
FORWARD 40
END
```



Ο ορισμός μιας διαδικασίας ξεκινάει με το «TO» και το όνομά της (στη συγκεκριμένη περίπτωση το ΓΗ).

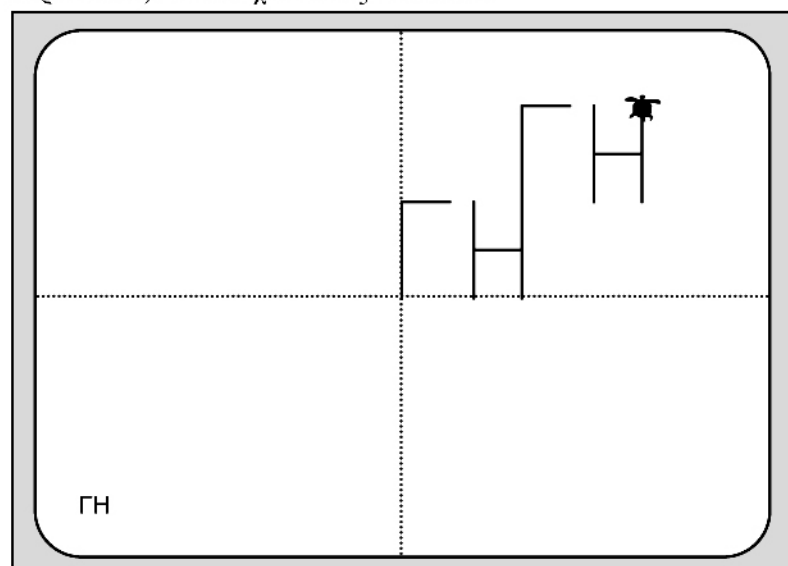
Στη συνέχεια ακολουθεί ένα τμήμα με εντολές που περιγράφουν τι πρέπει να κάνει η χελώνα ώστε να φέρει σε πέρας το συγκεκριμένο έργο που θα πραγματοποιεί η διαδικασία.

Η διαδικασία τελειώνει με την εντολή END.

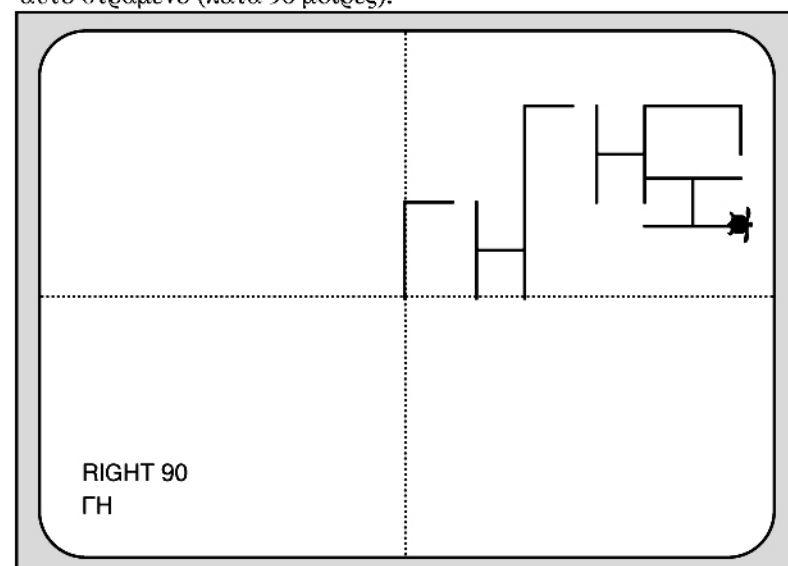
Από εδώ και στο εξής όταν θα "λέμε" στη χελώνα ΓΗ, αυτή θα σχεδιάζει το ΓΗ.

"Εκτελώντας" τη διαδικασία ΓΗ

Προηγουμένως δώσαμε τη διαταγή ΓΗ και η χελώνα ξεκίνησε από εκεί που ήταν και σχεδίασε το ΓΗ. Αν ξαναδώσουμε τη διαταγή ΓΗ, η χελώνα θα ξεκινήσει από το σημείο που βρίσκεται (στο πάνω δεξιά άκρο του Η) και θα σχεδιάσει ξανά το ΓΗ.



Αν στρίψουμε τη χελώνα δεξιά (90 μοίρες) και τη διατάξουμε να εκτελέσει τη διαταγή ΓΗ, τότε το ΓΗ που θα προκύψει θα είναι και αυτό στραμένο (κατά 90 μοίρες).



Καθαρίζοντας την οθόνη

Αν στην οθόνη υπάρχουν κάποια σχέδια και θέλουμε να "καθαρίσει η οθόνη" χρησιμοποιούμε την εντολή CG. Στο εξής όμως θα προσπαθήσουμε να "εξελληνίσουμε" τις εντολές και τις διαδικασίες που θα χρησιμοποιούμε. Έτσι θα αντικαταστήσουμε τις εντολές της Logo (όσες μπορούμε) με διαδικασίες που θα φτιάξουμε εμείς και θα έχουν ελληνικά ονόματα που θα είναι περισσότερο κατανοητά.

Ας φτιάξουμε τώρα μια διαδικασία που θα καθαρίζει την οθόνη σβήνοντας ό,τι σχήμα υπάρχει σ' αυτή.

ΤΟ ΣΒΗΣΕ_ΕΙΚΟΝΑ

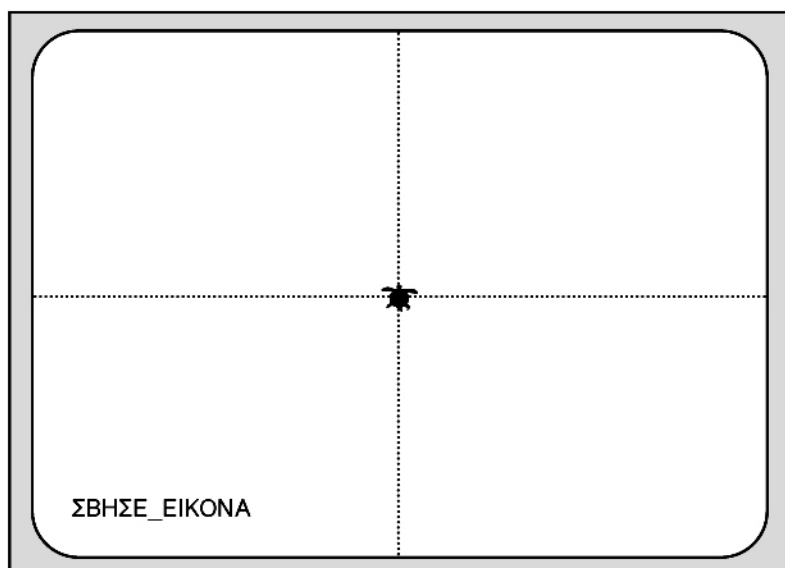
CG

END

Η εντολή CG προέρχεται από τα αρχικά της ισοδύναμης εντολής Clear Graphics που στα αγγλικά σημαίνει καθαρίσε την οθόνη.

Από εδώ και στο εξής όταν θέλουμε να καθαρίσουμε την οθόνη θα πληκτρολογούμε «ΣΒΗΣΕ_ΕΙΚΟΝΑ» όπως φαίνεται στο κάτω μέρος της οθόνης.

Αν το όνομα της διαδικασίας αποτελείται από περισσότερες από μια λέξεις αυτές πρέπει να συνδέονται με τον χαρακτήρα της υπογράμμισης.



Η χελώνα κινείται σε ευθεία γραμμή

Ας μάθουμε τον υπολογιστή να κινεί τη χελώνα. Θα ορίσουμε μια διαδικασία που θα κάνει τη χελώνα να κινείται σε ευθεία γραμμή. Αυτό που χρειαζόμαστε είναι να καθορίζουμε (κάθε φορά που διατάζουμε την χελώνα να κινηθεί) πόσα βήματα να κάνει πριν σταματήσει. Χρειάζεται λοιπόν κατά τον ορισμό της διαδικασίας εκτός από το όνομά της, να της δώσουμε και το όνομα μιας μεταβλητής (το ΜΗΚΟΣ) όπου θα φυλάει ο υπολογιστής το πλήθος των βημάτων που θα του λέμε να προχωρήσει η χελώνα κάθε φορά.

ΤΟ ΕΥΘΕΙΑ :ΜΗΚΟΣ

FD :ΜΗΚΟΣ

END

Η εντολή FD της logo προέρχεται από την ισοδύναμη εντολή **For-warD** που στα αγγλικά σημαίνει εμπρός.

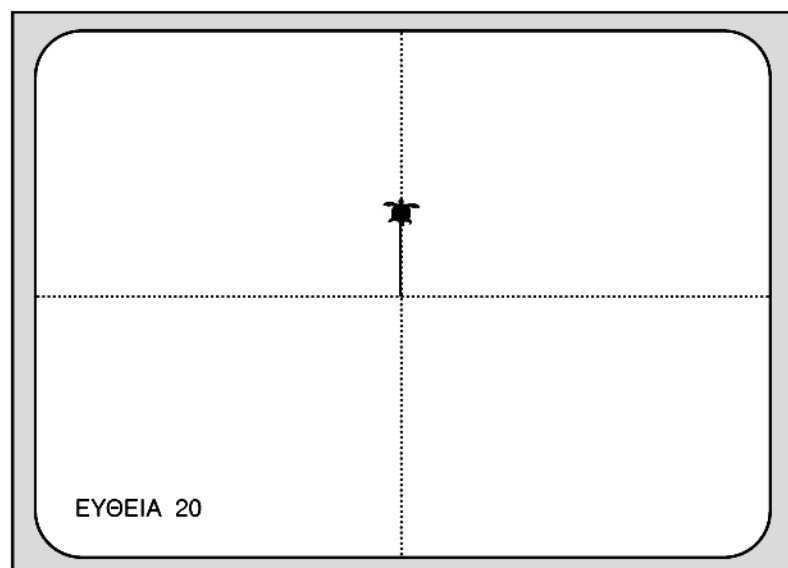
Πρέπει να ξεκαθαριστεί ότι άλλο το όνομα μιας μεταβλητής και άλλο η τιμή της.

Το 20 είναι η τιμή της μεταβλητής (το περιεχόμενό της) και αναφερόμαστε σ' αυτή μέσα στο πρόγραμμα με το :ΜΗΚΟΣ.

«Μια βασική μαθηματική έννοια, της οποίας η κατανόηση διευκολύνεται με τη χελώνα είναι η ιδέα της μεταβλητής: η ιδέα της χρήσης ενός συμβόλου για να ονομαστεί μια άγνωστη οντότητα»
Seymour Papert

ΜΗΚΟΣ

20



Όταν διατάξουμε τον υπολογιστή να εκτελέσει τη διαδικασία ΕΥΘΕΙΑ αυτή θα πρέπει να συνοδεύεται και με τον αριθμό των βημάτων που θα κάνει η χελώνα.

Μαθαίνοντας τη χελώνα να στρίβει δεξιά

Το επόμενο βήμα για να κατευθύνουμε τη χελώνα μέσα από το πρόγραμμα, είναι να ορίσουμε μια διαδικασία που θα την κάνει να στρίβει.

Όπως στη διαδικασία ΕΥΘΕΙΑ πρέπει να πούμε στον υπολογιστή πόσα βήματα να κάνει, έτσι και στη διαδικασία της στροφής θα πρέπει να του πούμε πόσο να στρίψει. Το όνομα της μεταβλητής που θα περιέχει το πόσο θα στρίψει η χελώνα είναι MOIRES.

ΤΟ ΔΕΞΙΑ :MOIRES

RT :MOIRES

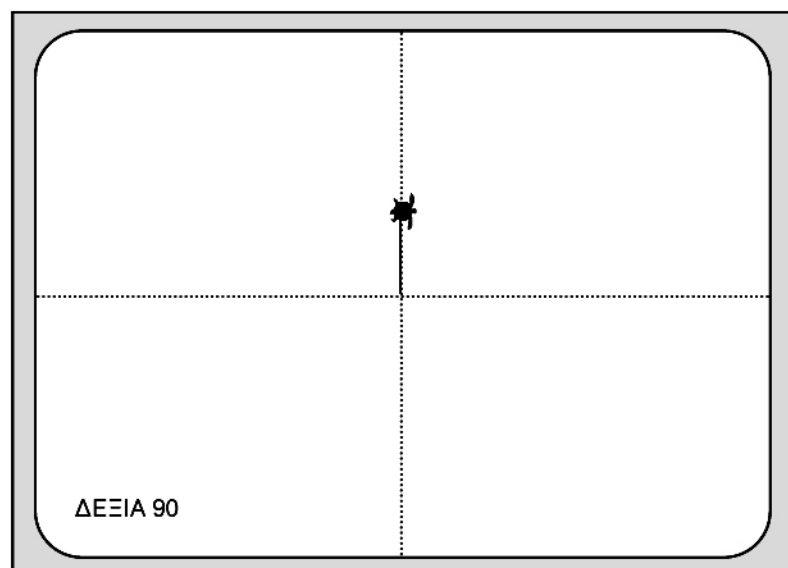
END

Η εντολή RT της logo, προέρχεται από την ισοδύναμη εντολή **Right** που στα αγγλικά σημαίνει δεξιά.

Η μονάδα μέτρησης που χρησιμοποιείται εδώ είναι οι μοίρες.

«Ο ενήλικος θα αναγνωρίσει γρήγορα αυτούς τους αριθμούς ως μετρήσεις γωνίας σε μοίρες.

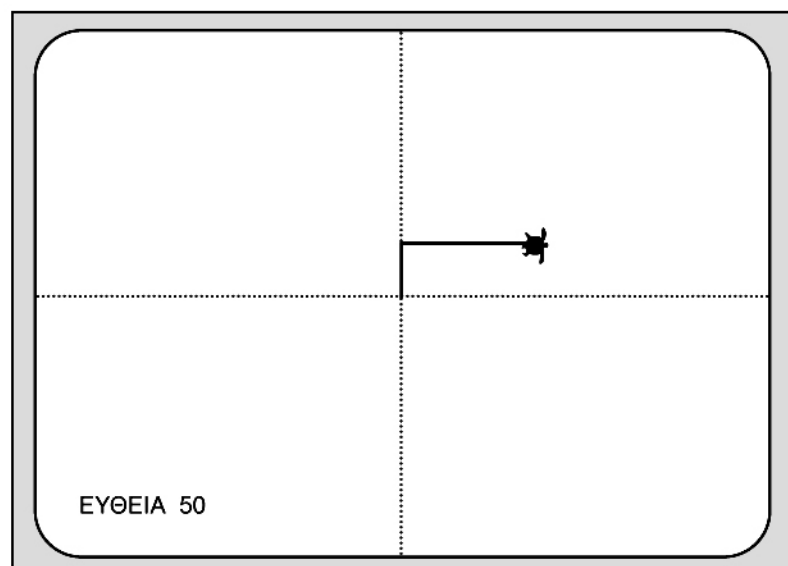
Για τα περισσότερα παιδιά αυτοί οι αριθμοί πρέπει να εξερευνηθούν και αυτό είναι μια διασκεδαστική διαδικασία παιχνιδιού.»
Seymour Papert



Όταν δώσουμε τη διαταγή **ΔΕΞΙΑ**, η χελώνα στρίβει προς τα δεξιά. Επειδή έχουμε διατάξει η στροφή να είναι 90 μοιρών, η χελώνα θα στραφεί κατά μια ορθή γωνία. Η χελώνα δεν μετακινήθηκε αλλά περιστράφηκε.

Πόσο να στρίψει η χελώνα;

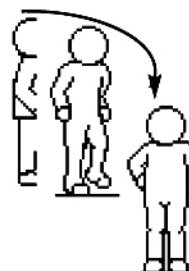
Μετά την προηγούμενη στροφή που έκανε η χελώνα, αν τη διατάξουμε να κινηθεί σε ευθεία γραμμή συνολικά η διαδρομή που θα σχηματισθεί από το ίχνος της θα είναι μια ορθή γωνία.



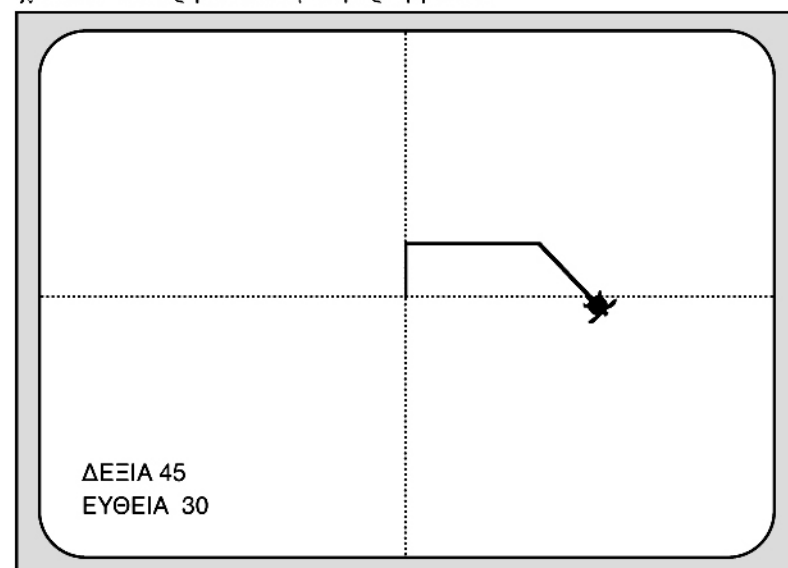
Μια στροφή 90 μοιρών αντιστοιχεί σε στροφή μιας ορθής γωνίας.

MOIRES

90



Οι στροφές που μπορεί να κάνει η χελώνα μπορούν να είναι και μικρότερες των 90 μοιρών. Με τη διαταγή ΔΕΞΙΑ 45 βλέπουμε τη χελώνα να στρίβει κατά μισή ορθή γωνία.



Διαπιστώνουμε ότι εφόσον ορίσαμε κάποια διαδικασία π.χ. την ΕΥΘΕΙΑ ή τη ΔΕΞΙΑ, μπορούμε να την καλούμε όποτε θέλουμε βάζοντας ό,τι τιμή θέλουμε στη μεταβλητή που συνοδεύει το όνομά της, π.χ. ΕΥΘΕΙΑ 20 ή ΕΥΘΕΙΑ 50 ή ΕΥΘΕΙΑ 30.

Η χελώνα στρίβει αριστερά

Η στροφή που έμαθε η χελώνα να κάνει είναι προς τα δεξιά. Θα της μάθουμε τώρα να πραγματοποιεί στροφές και προς τα αριστερά. Αυτό γίνεται με το να ορίσουμε μια νέα διαδικασία, τη **ΑΡΙΣΤΕΡΑ**, που όπως και η **ΔΕΞΙΑ** χρειάζεται μια μεταβλητή για τον προσδιορισμό του μεγέθους της στροφής.

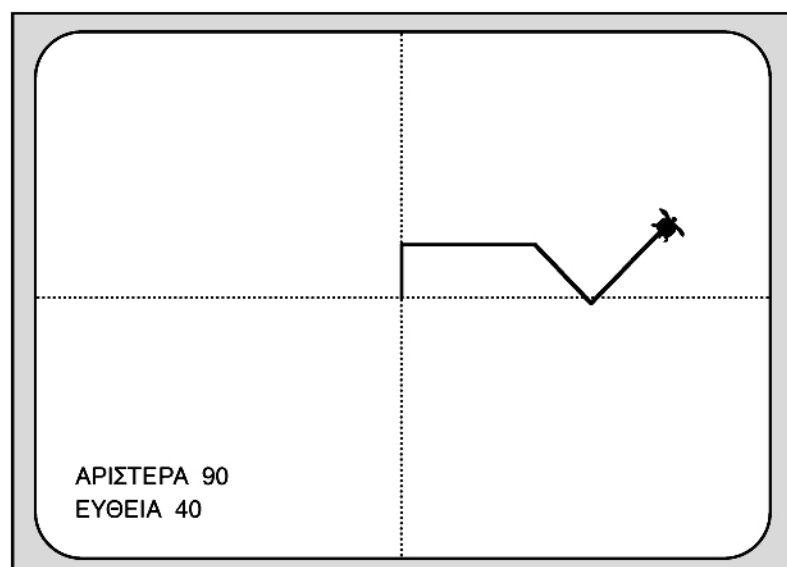
ΤΟ ΑΡΙΣΤΕΡΑ :MOIRES

LT :MOIRES

END

Η εντολή LT της logo προέρχεται από την ισοδύναμη εντολή **LefT** που στα αγγλικά σημαίνει αριστερά.

Προτείνουμε να πειραματιστείτε καλώντας την εντολή **ΑΡΙΣΤΕΡΑ** με διάφορες τιμές για τη γωνία όπως 180, 270, 360, 45, 30, 60, 135, 120, 150, 315, 330, 300, 450, 390, 420, 720. Ποιές από αυτές προκαλούν το ίδιο αποτέλεσμα;



Επιβεβαιώστε πειραματιζόμενοι ότι αριστερή στροφή 270 μοιρών και δεξιά στροφή 90 μοιρών έχουν το ίδιο τελικό αποτέλεσμα.

Πως θα στρίψουμε τη χελώνα 120 μοίρες αριστερά αν διαθέτουμε μόνο την διαταγή **ΔΕΞΙΑ**;

Η χελώνα στρίβει και προχωρεί

Η διαδικασία ΕΥΘΕΙΑ κινεί τη χελώνα σε ευθεία γραμμή στην κατεύθυνση προς την οποία «κοίταγε» η χελώνα. Αν επιθυμούμε να κινηθεί προς άλλη κατεύθυνση θα πρέπει να στρίψουμε τη χελώνα με τη διαδικασία ΔΕΞΙΑ. Μπορούμε να ενσωματώσουμε αυτές τις δύο διαδικασίες σε μια που θα στρίβει την χελώνα προς την κατεύθυνση που θέλουμε και στη συνέχεια θα την προχωράει όσα βήματα της πούμε.

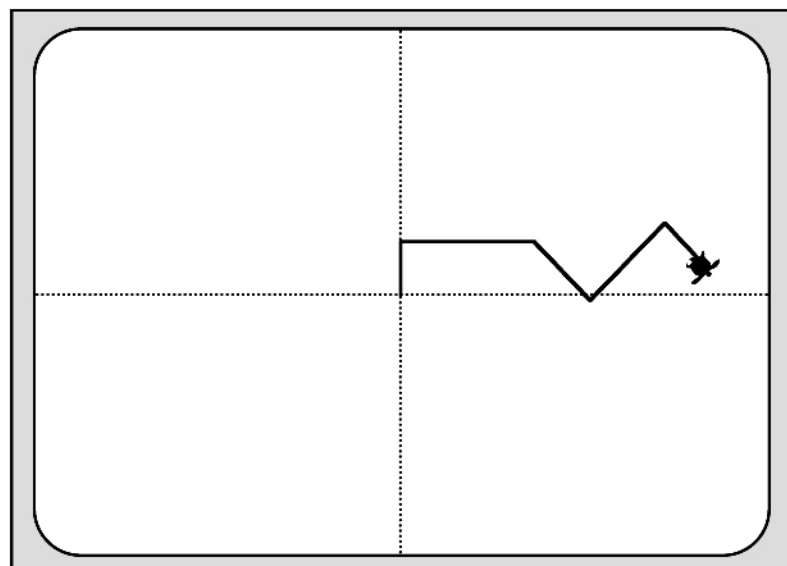
Ας ονομάσουμε αυτή τη διαδικασία ΓΡΑΜΜΗ. Από τα προηγούμενα είναι προφανές ότι θα χρειάζεται τιμή και για το μέγεθος της στροφής και για το πόσα βήματα θα κάνει.

```
TO ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ
ΔΕΞΙΑ :ΜΟΙΡΕΣ
ΕΥΘΕΙΑ :ΜΗΚΟΣ
END
```

Στον ορισμό μιας διαδικασίας μπορούν να χρησιμοποιηθούν και άλλες διαδικασίες (όπως οι εντολές της Logo) που έχουν ορισθεί προηγουμένως.

Δεν είναι απαραίτητο (επειδή κατά τον ορισμό π.χ. της διαδικασίας ΕΥΘΕΙΑ χρησιμοποιήθηκε το όνομα ΜΗΚΟΣ για τη μεταβλητή) και στο νέο ορισμό να χρησιμοποιηθεί το ίδιο όνομα. Για παράδειγμα ο προηγούμενος ορισμός θα μπορούσε να είναι:

```
TO ΓΡΑΜΜΗ :F :X
ΔΕΞΙΑ :F
ΕΥΘΕΙΑ :X
END
```



Τα ονόματα των διαδικασιών που χρησιμοποιούνται μέσα στη διαδικασία πρέπει να είναι όπως ακριβώς έχουν ορισθεί προηγουμένως.

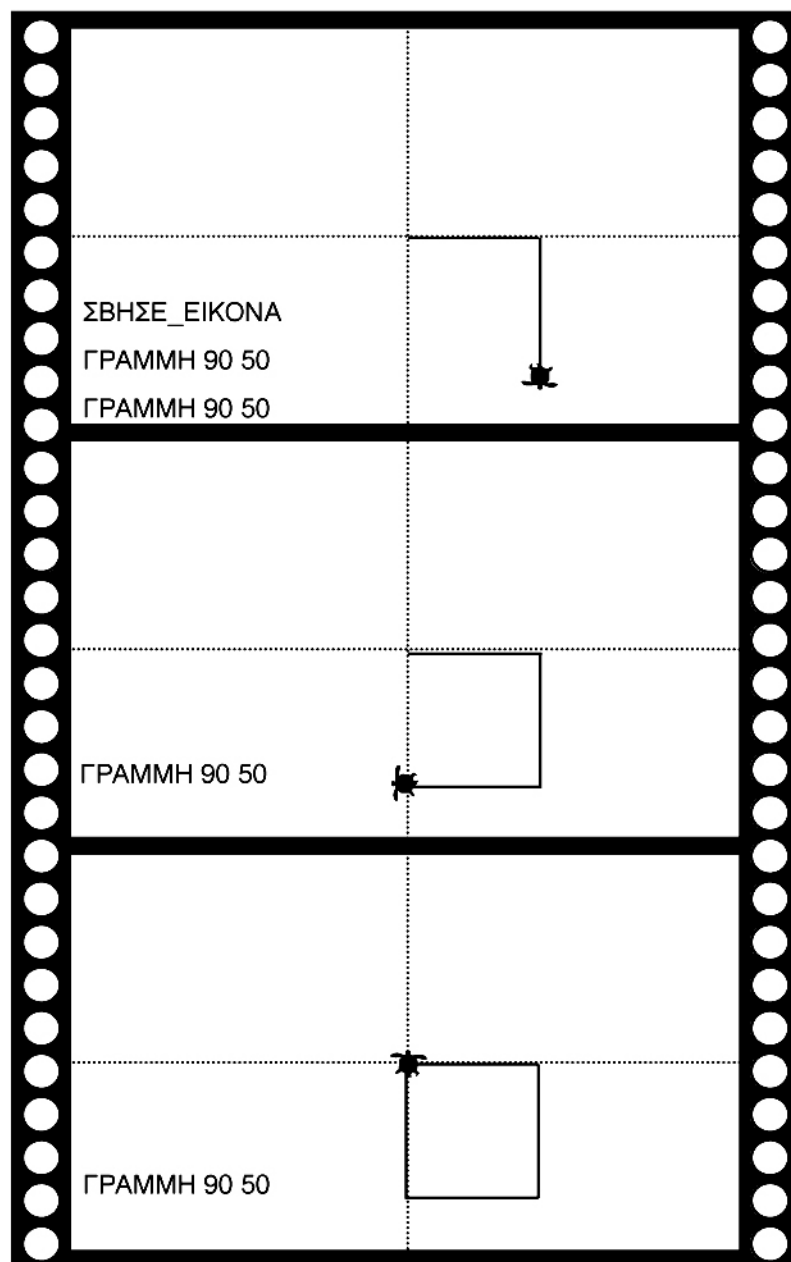
Τα ονόματα των μεταβλητών που χρησιμοποιούνται μέσα στη διαδικασία πρέπει να είναι τα ίδια με αυτά που δηλώνονται στην επικεφαλίδα.

Ποιές από τις παρακάτω διαταγές είναι λάθος και γιατί;

```
ΓΡΑΜΜΗ
ΓΡΑΜΜΗ 45
ΓΡΑΜΜΗ 0 0
ΓΡΑΜΜΗ 1 1
ΓΡΑΜΜΗ 0 0
ΓΡΑΜΜΗ 20 9
ΓΡΑΜΜΗ 4523
ΓΡΑΜΜΗ 90 K7
ΓΡΑΜΜΗ 90 50 3
```


Η χελώνα χαράζει ένα τετράγωνο

Μπορούμε δίνοντας κάθε φορά μια εντολή να βάλουμε τη χελώνα να σχηματίσει ένα τετράγωνο.



Αν η χελώνα στρίψει δεξιά 90 μοίρες και προχωρήσει κάποια απόσταση και μετά στρίψει πάλι δεξιά 90 μοίρες και προχωρήσει την ίδια απόσταση η τροχιά της θα σχηματίσει τις πλευρές μιας ορθής γωνίας.

Η χελώνα με τη διαταγή ΓΡΑΜΜΗ 90 50 συνεχίζει και ξαναστρίβει δεξιά 90 μοίρες και προχωράει την ίδια απόσταση.

Τέλος η χελώνα στρίβει δεξιά 90 μοίρες και προχωράει την ίδια απόσταση. Το αποτέλεσμα είναι να βρεθεί στην αρχική της θέση έχοντας κάνει μια πλήρη περιστροφή με διαδρομή σε σχήμα τετραγώνου.

Η χελώνα κινείται σε τετράγωνο

Τώρα θα διδάξουμε τη χελώνα να κινείται μόνη της και να σχηματίζει το τετράγωνο. Ας φτιάξουμε μια νέα διαδικασία, την ΤΕΤΡΑΓΩΝΟ, που θα κάνει τη χελώνα να διαγράψει ένα τετράγωνο.

```
ΤΟ ΤΕΤΡΑΓΩΝΟ :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
END
```

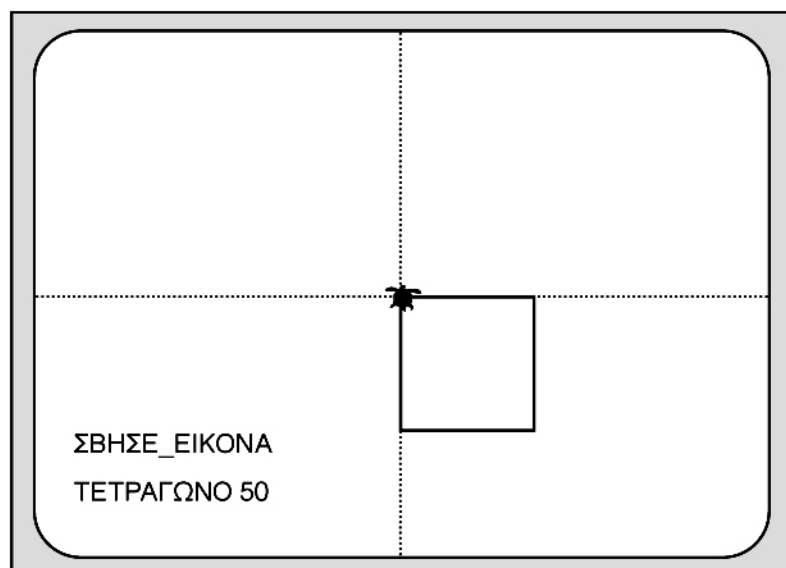
Παρατηρούμε ότι η διαδικασία ΓΡΑΜΜΗ 90 :PLEYRA χρησιμοποιείται τέσσερις φορές στη διαδικασία τετράγωνο.

Τι θα συνέβαινε αν η διαδικασία ΤΕΤΡΑΓΩΝΟ είχε ορισθεί ως εξής;

```
ΤΟ ΤΕΤΡΑΓΩΝΟ :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
ΓΡΑΜΜΗ 90 :PLEYRA
```

Στη διαδικασία ΓΡΑΜΜΗ (που χρησιμοποιείται στον ορισμό της διαδικασίας ΤΕΤΡΑΓΩΝΟ) η τιμή της μεταβλητής MOIRES είναι 90 ενώ η μεταβλητή PLEYRA θα παίρνει τιμή τη στιγμή που θα διατάζουμε τη χελώνα να κινηθεί σε τετράγωνο.

Το άθροισμα των γωνιών των τεσσάρων στροφών που θα κάνει η χελώνα για να σχηματίσει τετράγωνο είναι:
 $90+90+90+90=$
 $=360$ μοίρες.



Η τιμή 50 είναι το μήκος της πλευράς του τετραγώνου. Το 50 πηγαίνει ως τιμή στη μεταβλητή PLEYRA της διαδικασίας ΓΡΑΜΜΗ κάθε φορά που τη χρησιμοποιεί η διαδικασία ΤΕΤΡΑΓΩΝΟ.

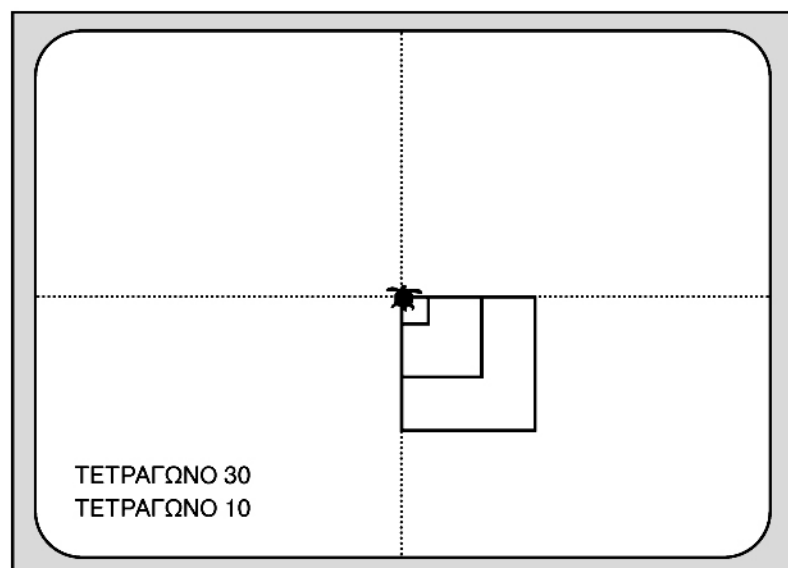
Η διαδικασία δίνει τιμή σε μια μεταβλητή

Προηγουμένως η στροφή που έκανε η χελώνα ξεκινώντας τη διαδικασία ΓΡΑΜΜΗ ήταν 90 μοίρες. Γενικά μπορούμε να βάλουμε μέσα στη μεταβλητή MOIRES την τιμή 90 με την εντολή MAKE και να χρησιμοποιούμε τη διαδικασία ΓΡΑΜΜΗ όπως φαίνεται στον παρακάτω νέο ορισμό της διαδικασίας ΤΕΤΡΑΓΩΝΟ.

```
TO ΤΕΤΡΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 90  
ΓΡΑΜΜΗ :MOIRES :PLEYRA  
ΓΡΑΜΜΗ :MOIRES :PLEYRA  
ΓΡΑΜΜΗ :MOIRES :PLEYRA  
ΓΡΑΜΜΗ :MOIRES :PLEYRA  
END
```

Η εντολή MAKE της logo, στα αγγλικά σημαίνει κάνει και εννοεί να δώσει την τιμή (το 90) ως περιεχόμενο σε κάποια μεταβλητή (την MOIRES).

Όταν αναφερόμαστε στην τιμή (δηλαδή το περιεχόμενο) της μεταβλητής MOIRES, χρησιμοποιούμε το :MOIRES, ενώ όταν αναφερόμαστε στην μεταβλητή χρησιμοποιούμε το "MOIRES.

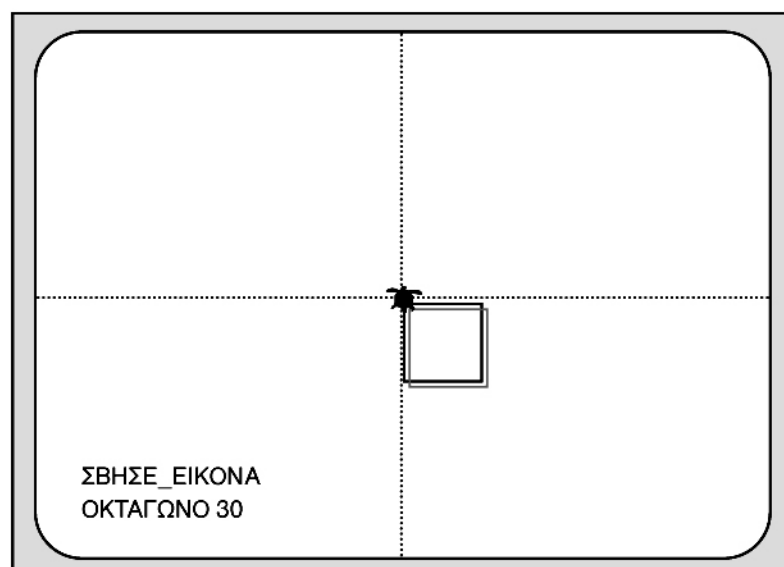


Προσπάθεια για περισσότερες πλευρές

Στη διαδικασία ΤΕΤΡΑΓΩΝΟ χρειάστηκε να χρησιμοποιήσουμε τέσσερις φορές τη διαδικασία ΓΡΑΜΜΗ. Για να φτιάξουμε ένα κανονικό σχήμα με οκτώ πλευρές, το οκτάγωνο, θα πρέπει να χρησιμοποιήσουμε τη διαδικασία ΓΡΑΜΜΗ οκτώ φορές. Ας γράψουμε λοιπόν τη διαδικασία ΟΚΤΑΓΩΝΟ.

```
ΤΟ ΟΚΤΑΓΩΝΟ :PLEYRA
MAKE "MOIRES 90
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

Θα διαπιστώσουμε ότι δε σχηματίστηκε ένα σχήμα με οκτώ πλευρές αλλά ότι η χελώνα διέτρεξε δύο φορές τις τέσσερις πλευρές ενός τετραγώνου. Τι δεν πήγε καλά;



Το πρόγραμμα που φαίνεται δίπλα είναι λάθος. **Λάθι κάνομαι όλυ. Αξοία αίχι να**

δηδασκώμασται από αφτά.

Μελετώντας τα λάθη μας μπορούμε να εντοπίσουμε τα αδύνατα σημεία του τρόπου που αντιμετωπίσαμε το πρόβλημα και να

αναζητήσουμε τρόπους όχι μόνο να διορθώσουμε το συγκεκριμένο λάθος αλλά και να γενικεύσουμε την εμπειρία μας ώστε να μην το επαναλάβουμε.

Το άθροισμα των γωνιών των οκτώ στροφών που κάνει η χελώνα είναι:
 $90+90+90+90+90+90+90+90 = 720$ μοίρες
ενώ μια πλήρης περιστροφή είναι 360 μοίρες.

Περισσότερες πλευρές, μικρότερη γωνία

Όπως είδαμε, η χελώνα έκανε οκτώ στροφές των 90 μοιρών πραγματοποιώντας δύο περιστροφές δηλαδή γωνία 720 μοιρών ενώ θέλουμε να κάνει μόνο μια περιστροφή (δηλαδή 360 μοίρες) στρίβοντας 8 φορές. Αυτό θα το πετύχουμε αν κάθε φορά αντί να στρίβει 90 μοίρες στρίβει το μισό, 45 μοίρες. Μ' αυτήν την παρατήρηση ας διορθώσουμε το πρόγραμμα της διαδικασίας ΟΚΤΑΓΩΝΟ καταχωρώντας ως τιμή στη μεταβλητή MOIRES το 45.

ΤΟ ΟΚΤΑΓΩΝΟ :PLEYRA

MAKE "MOIRES 45

GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

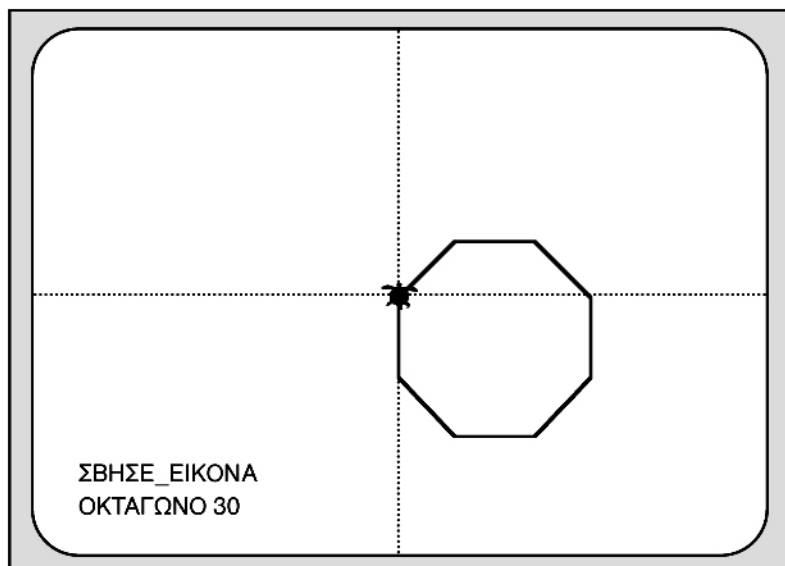
GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

GRAMMH :MOIRES :PLEYRA

END

«...Το να μάθεις
να είσαι
έμπειρος
προγραμματιστή
σημαίνει να
μάθεις πως να
γίνεις ικανός
στην
απομόνωση και
διόρθωση
"σφαλμάτων",
σημείων που
αποτρέπουν την
εκτέλεση του
προγράμματος.
Η ερώτηση
σχετικά με ένα
πρόγραμμα δεν
είναι το αν είναι
σωστό ή
λανθασμένο,
αλλά αν
διορθώνεται.»
Seymour Papert



Το άθροισμα
των γωνιών των
οκτώ στροφών
που κάνει η
χελώνα για να
σχηματίσει
οκτάγωνο είναι:
 $45+45+45+45+45+45+45+45 = 360$ μοίρες.

Γωνία και πλήθος πλευρών

Το επόμενο βήμα είναι να μάθουμε τη χελώνα να σχεδιάζει ένα κανονικό εξάγωνο. Φυσικά θα χρησιμοποιήσουμε τη διαδικασία ΓΡΑΜΜΗ έξι φορές. Δε μένει παρά να υπολογίσουμε τη γωνία που θα στρίβει η χελώνα κάθε φορά.

Είδαμε ότι για να κάνει η χελώνα μια πλήρη περιστροφή (είτε διαγράφοντας τετράγωνο είτε οκτάγωνο) το άθροισμα των γωνιών είναι 360 μοίρες.

Δηλαδή

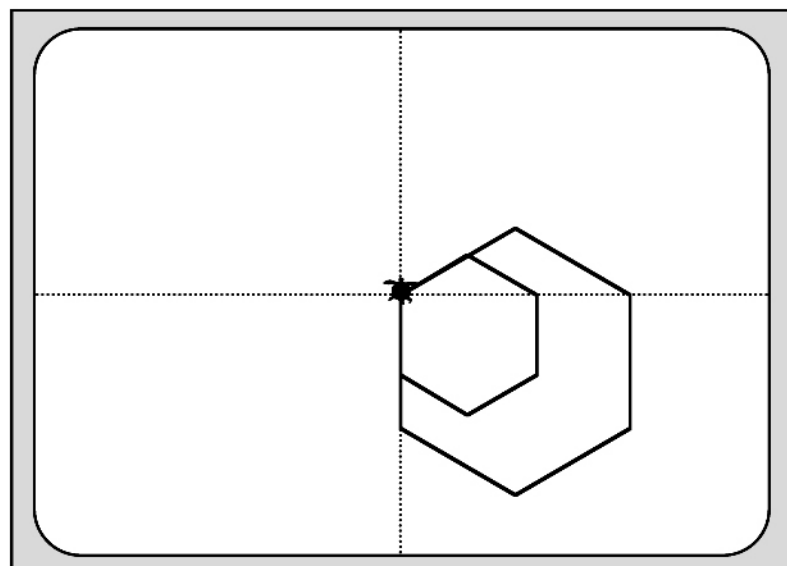
$$90+90+90+90 = 4 \times 90 = 360 \text{ ή } 360/4 = 90 \text{ μοίρες}$$

$$45+45+45+45+45+45+45+45 = 8 \times 45 = 360 \text{ ή } 360/8 = 45 \text{ μοίρες}$$

Άρα και η γωνία για το εξάγωνο θα είναι $360/6 = 60$ μοίρες.

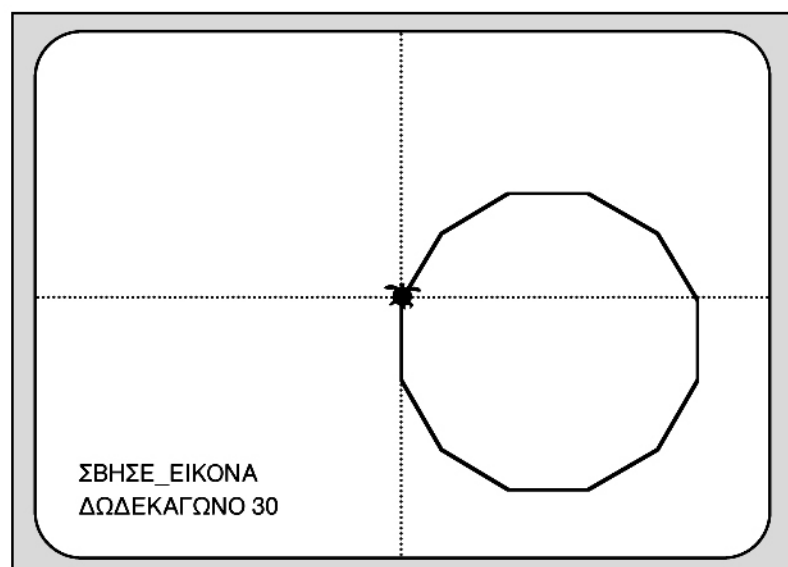
```
TO ΕΞΑΓΩΝΟ :PLEYRA
MAKE "MOIRES ( 360 / 6 )
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
ΓΡΑΜΜΗ :MOIRES :PLEYRA
END
```

Στην εντολή
MAKE
υπολογίζεται το
αποτέλεσμα της
παράστασης
 $360/6$ και
καταχωρείται
ως τιμή στη
μεταβλητή
"MOIRES.



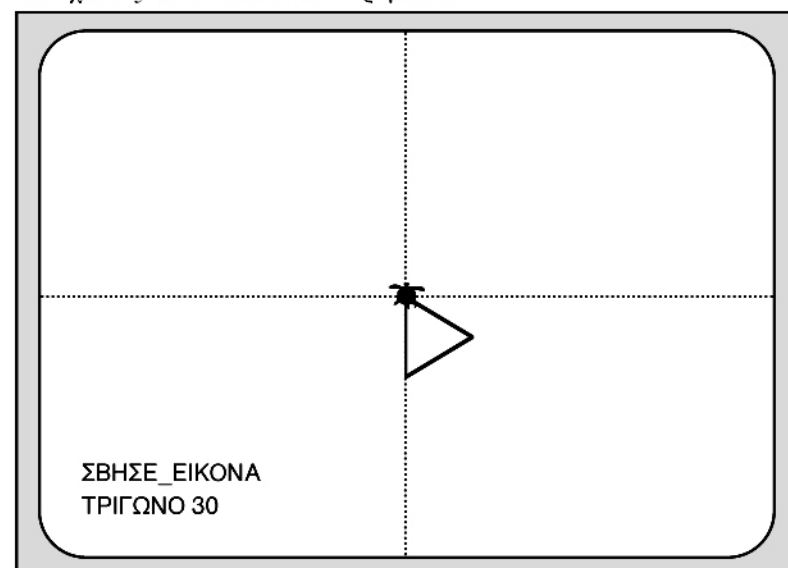
Δωδεκάγωνο και τρίγωνο

Γράψτε το πρόγραμμα της διαδικασίας που θα μαθαίνει τη χελώνα να σχεδιάζει ένα κανονικό δωδεκάγωνο. Τι τιμή θα δώσετε στην μεταβλητή MOIRES και γιατί;



«Για να λύσεις
ένα πρόβλημα
ψάξε να βρεις
κάτι που να
μοιάζει με αυτό
το οποίο ήδη
καταλαβαίνεις»
Seymour Papert

Γράψτε το πρόγραμμα της διαδικασίας που θα μαθαίνει τη χελώνα να σχεδιάζει ένα "κανονικό" τρίγωνο.



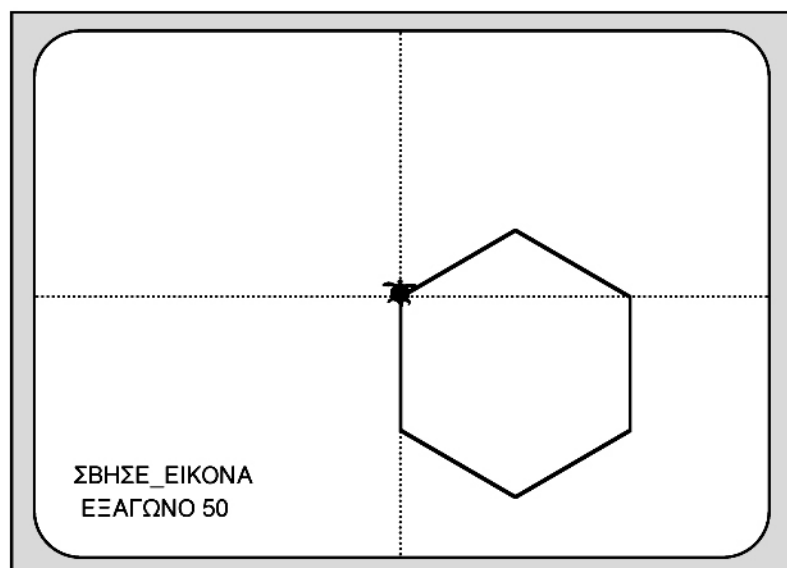
Ένας τρόπος επανάληψης

Στο πρόγραμμα για το δωδεκάγωνο, η ίδια διαδικασία (ΓΡΑΜΜΗ) επαναλαμβάνεται 12 φορές. Το να γράφεις 12 φορές το ίδιο πράγμα μάλλον θυμίζει τιμωρία παρά δημιουργική εργασία. Όλες οι γλώσσες προγραμματισμού διαθέτουν μια τουλάχιστον εντολή με τη βοήθεια της οποίας μπορούμε εύκολα να επαναλάβουμε κάποιο σύνολο εντολών.

Η Logo διαθέτει την εντολή REPEAT. Ας δούμε πως μπορεί να γραφεί το πρόγραμμα της διαδικασίας ΕΞΑΓΩΝΟ χρησιμοποιώντας την.

```
TO ΕΞΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES ( 360 / 6 )  
REPEAT 6 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

Η εντολή REPEAT της Logo στα αγγλικά, σημαίνει επανάλαβε. Μετά τη λέξη REPEAT ακολουθεί ένας ακέραιος αριθμός που καθορίζει πόσες φορές θα γίνει η επανάληψη. Στη συνέχεια μέσα σε αγκύλες υπάρχει η εντολή ή οι εντολές που θα επαναληφθούν.



Ίδιες διαδικασίες με τη βοήθεια του repeat

Ας δούμε πως μπορούν να ξαναορισθούν οι διαδικασίες ΤΡΙΓΩΝΟ, ΤΕΤΡΑΓΩΝΟ, ΟΚΤΑΓΩΝΟ και ΔΩΔΕΚΑΓΩΝΟ με τη βοήθεια της εντολής REPEAT.

```
TO ΤΡΙΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 3  
REPEAT 3 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

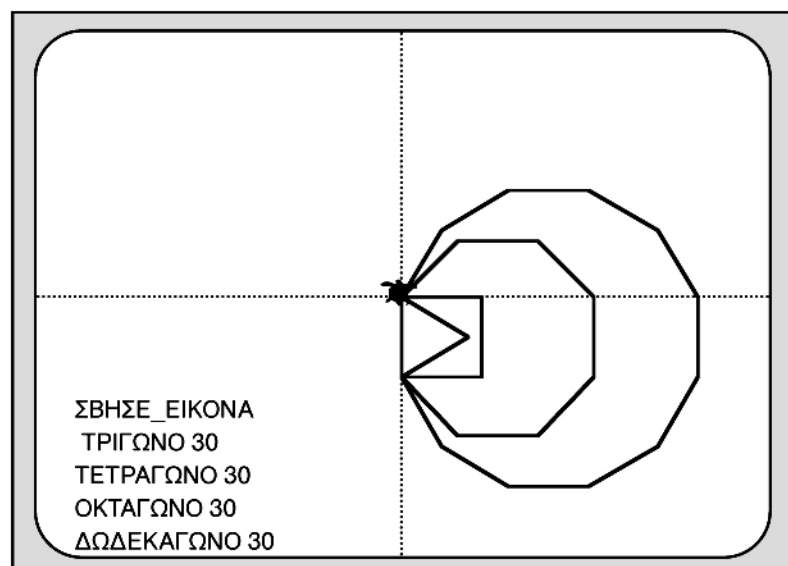
```
TO ΤΕΤΡΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 4  
REPEAT 4 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

```
TO ΟΚΤΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 8  
REPEAT 8 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

```
TO ΔΩΔΕΚΑΓΩΝΟ :PLEYRA  
MAKE "MOIRES 360 / 12  
REPEAT 12 [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

Τα τέσσερα αυτά προγράμματα διαφέρουν μόνο ως προς τον αριθμό που εκφράζει το πλήθος των πλευρών (και των γωνιών) του κανονικού σχήματος. Δηλαδή τα προγράμματα όλων των κανονικών σχημάτων έχουν την ίδια **δομή**.

«Ο καλύτερος
τρόπος
κατανόησης της
μάθησης είναι
να καταλάβουμε
πρώτα τις
ειδικές
επιλεγμένες
περιπτώσεις και
μετά να
νοιαστούμε για
το πως θα
γενικευτεί αυτή
η
κατανόηση.»
Seymour Papert



Γιατί όλα τα
σχήματα έχουν
μια κοινή
πλευρά;

Τρι-τετρα-εξα-οκτα...γωνο ή πολύγωνο

Από τα προηγούμενα προγράμματα ας κρατήσουμε ό,τι είναι κοινό σε όλα:

```
ΤΟ όνομα σχήματος :PLEYRA  
MAKE "MOIRES 360 / πλήθος γωνιών  
REPEAT πλήθος πλευρών [ ΓΡΑΜΜΗ :MOIRES :PLEYRA ]  
END
```

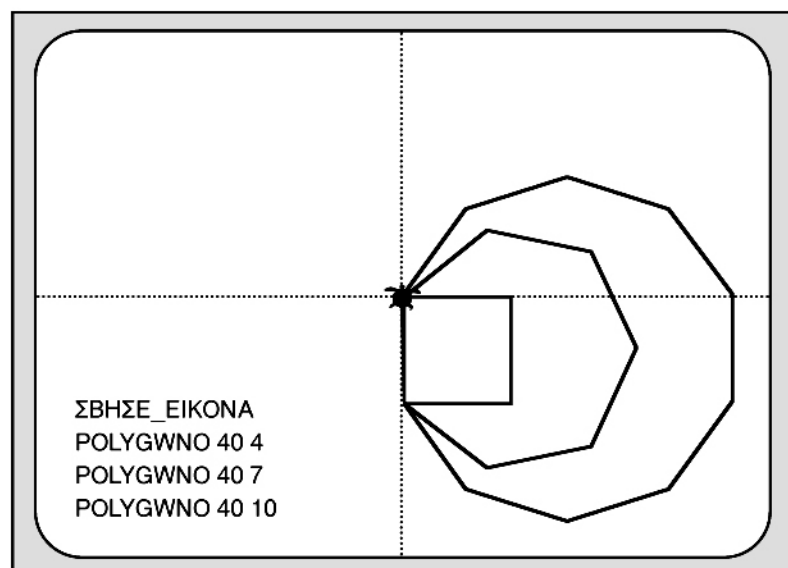
Το πλήθος των γωνιών είναι ίδιο με το πλήθος των πλευρών.

Με τη διαδικασία ΤΕΤΡΑΓΩΝΟ η χελώνα μαθαίνει να σχεδιάζει ένα τετράγωνο με μέγεθος ανάλογο προς το μήκος της πλευράς που θα της πούμε όταν τη διατάξουμε.

Ας προχωρήσουμε ένα βήμα πιο κάτω. Ας διδάξουμε τη χελώνα να σχεδιάζει οποιοδήποτε κανονικό πολύγωνο, δηλώνοντας και το πλήθος των πλευρών ή γωνιών του. Χρειάζεται λοιπόν κατά τον ορισμό της διαδικασίας αυτής να δίνεται στη χελώνα εκτός από το μήκος της πλευράς και το πλήθος των γωνιών.

```
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS  
MAKE "MOIRES ( 360 / :PLHQOS )  
REPEAT :PLHQOS [ΓΡΑΜΜΗ :MOIRES :PLEYRA]  
END
```

Η παράσταση (360/:PLHQOS) βρίσκεται μέσα σε παρενθέσεις για να είναι περισσότερο κατανοητό στον προγραμματιστή ότι στην μεταβλητή MOIRES εκχωρείται το αποτέλεσμα της διαίρεσης και όχι μόνο το 360.



Η δομή του προγράμματος αναπαριστάται στη μνήμη του υπολογιστή τη δομή του φυσικού κόσμου.

«Διδάσκοντας τον υπολογιστή πως να σκέφτεται, τα παιδιά ξεκινούν για μια εξερεύνηση του δικού τους τρόπου σκέψης.

Η εμπειρία μπορεί να είναι μεθυστική. Η σκέψη για τη σκέψη κάνει το παιδί επιστημολόγο, μια εμπειρία άγνωστη ακόμα και στους περισσότερους ενήλικες.»
Seymour Papert

Παρατηρούμε ότι με την ίδια διαδικασία σχηματίζονται διαφορετικά σχήματα.

Προσανατολισμένο πολύγωνο

Όλα τα προηγούμενα σχήματα δημιουργούνται δεξιά και κάτω από το σημείο που ξεκινά η χελώνα. Αυτό οφείλεται στο ότι η πρώτη εντολή που κινεί τη χελώνα είναι η ΓΡΑΜΜΗ και περιέχει την εντολή ΔΕΞΙΑ.

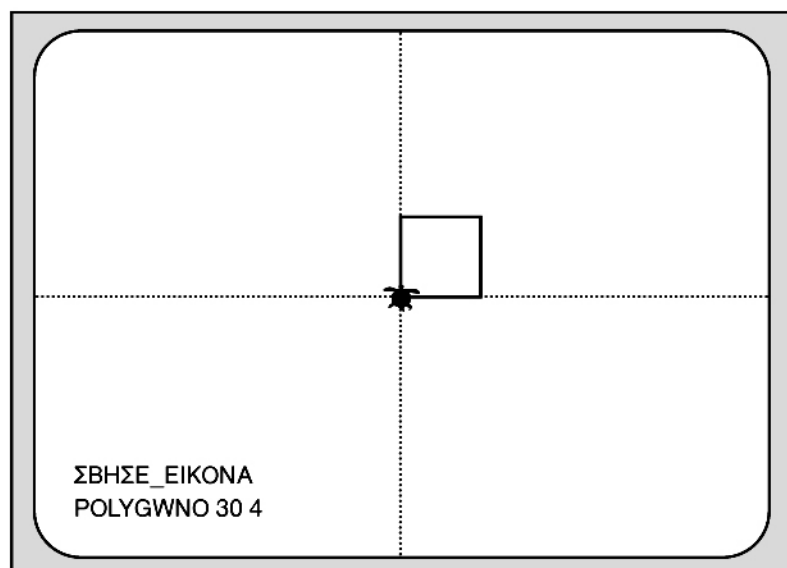
Αν θέλουμε η αρχή των σχημάτων να είναι πάνω από το σημείο αφετηρίας της χελώνας θα πρέπει να "εξουδετερώσουμε" αυτή τη στροφή δεξιά με μια στροφή 90 μοιρών προς τα αριστερά.

Στη συνέχεια αφού σχηματισθεί το πολύγωνο, θα πρέπει να αποκαταστήσουμε τον αρχικό προσανατολισμό της χελώνας στρέφοντάς την 90 μοίρες προς τα δεξιά.

Με βάση αυτές τις παρατηρήσεις ας βελτιώσουμε τη διαδικασία POLYGWNO ώστε να δημιουργεί σχήματα που η αρχή τους να στρέφεται προς τα πάνω.

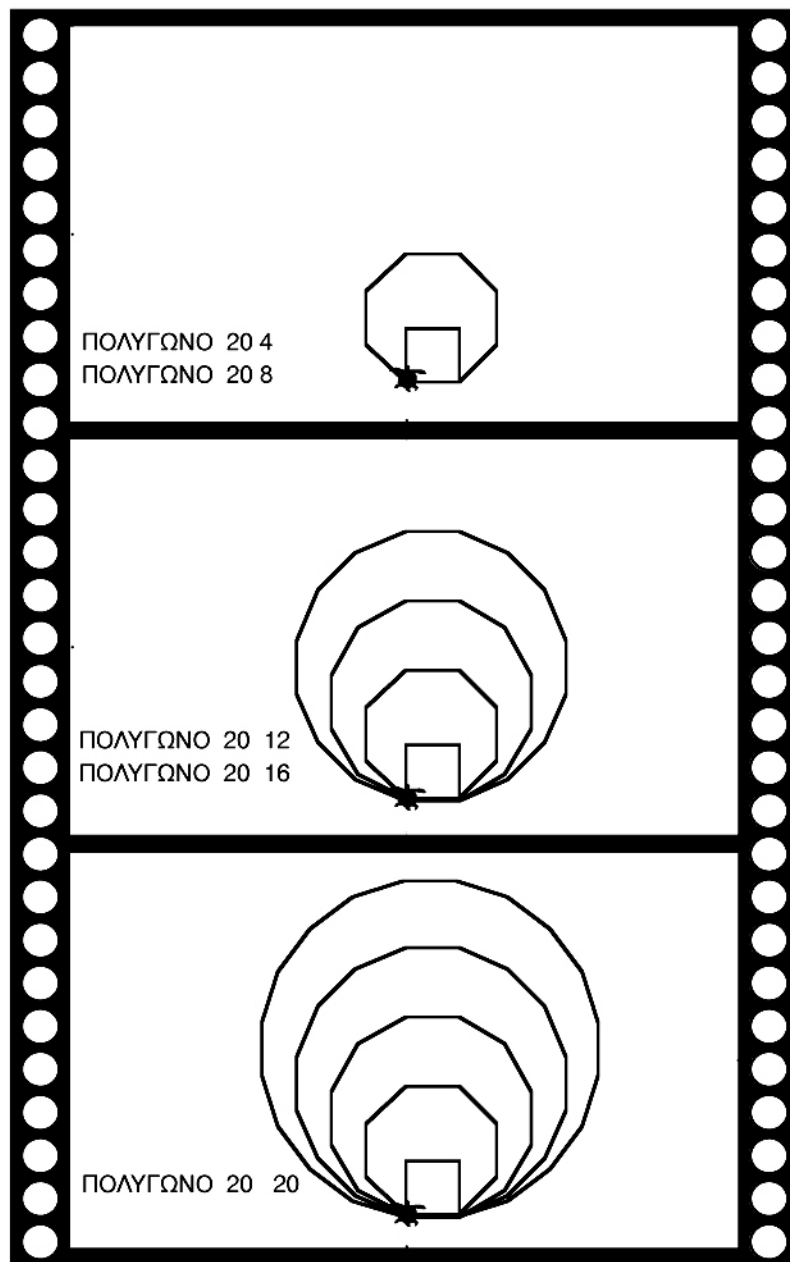
```
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES ( 360 / :PLHQOS )
REPEAT :PLHQOS [ΓΡΑΜΜΗ :MOIRES :PLEYRA]
ΔΕΞΙΑ 90
END
```

Πως αλλιώς
μπορεί να
προσανατολιστεί
το πολύγωνο;



Το πλήθος των πλευρών τείνει στο άπειρο

Στα παρακάτω σχήματα, αυξάνεται διαδοχικά το πλήθος των πλευρών της διαδικασίας ΠΟΛΥΓΩΝΟ.



Η χελώνα σχεδιάζει ένα τετράγωνο και ένα οκτάγωνο με μήκος πλευράς 20 βημάτων. Για να το επιβεβαιώσετε μετρήστε πόσες γωνίες έχει το κάθε σχήμα.

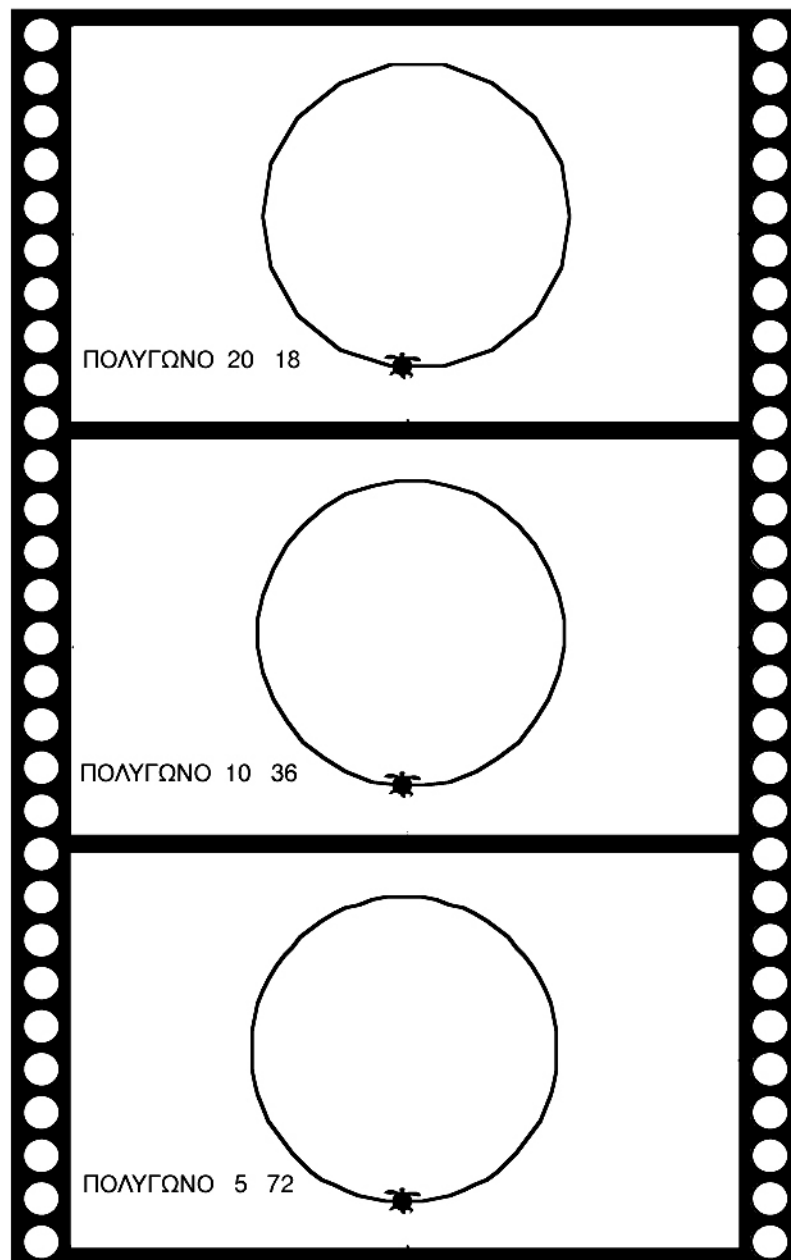
Στη συνέχεια σχηματίζουμε ένα δωδεκάγωνο και ένα δεκαεξάγωνο. Προσπαθήστε πάλι να επιβεβαιώσετε μετρώντας τις γωνίες τους, ότι έχουν τον σωστό αριθμό γωνιών.

Τέλος, ζητάμε να σχηματιστεί ένα εικοσάγωνο. Είναι εύκολο να μετρήσετε τώρα το πλήθος των γωνιών του σχήματος;

Με τι μοιάζει το σχήμα; Φτιάξτε πολύγωνο με εικοσιτέσσερες γωνίες.

Προσεγγίζοντας μια περιφέρεια κύκλου

Ας φτιάξουμε τρία πολύγωνα με 18, 36 και 72 πλευρές αντίστοιχα και ας παρατηρήσουμε τις διαφορές τους.



Φαίνεται ότι πλησιάζουμε τα όρια της διακριτικής ικανότητας του μέσου απεικόνισης αλλά και της ίδιας της αναπαράστασης. Σιγά-σιγά, μειώνεται η ικανότητα διάκρισης μεταξύ εικόνων δύο αντικειμένων που έχουν πολύ μικρές διαφορές.

Μεταξύ του 18-γωνου και του 36-γωνου υπάρχουν κάποιες μικρές διαφορές.

Μεταξύ του 36-γωνου και του 72-γωνου δεν υπάρχουν σημαντικές διαφορές στα σχήματα όπως εμφανίζονται στην οθόνη.

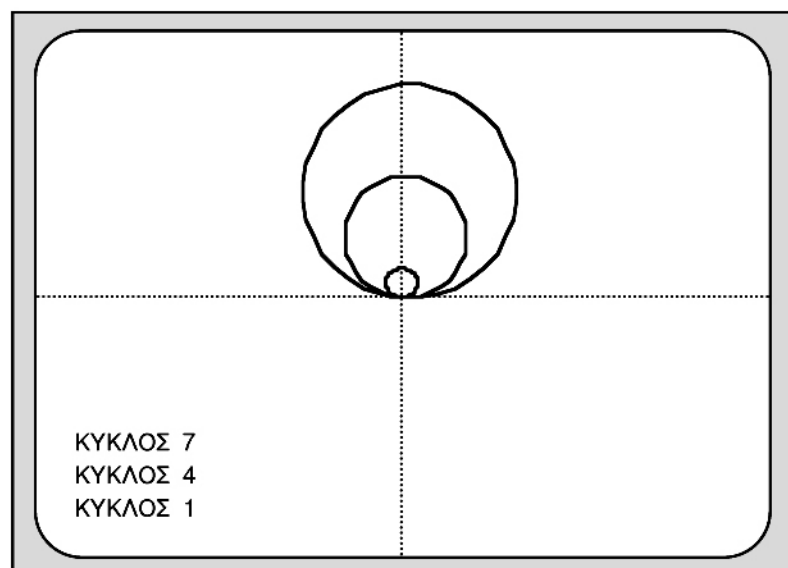
Σχήμα που μοιάζει με περιφέρεια κύκλου

Από τα σχήματα της προηγούμενης σελίδας καταλήγουμε στο συμπέρασμα ότι το 36-γωνο είναι μια ικανοποιητική προσέγγιση του κύκλου.

Επειδή ο κύκλος είναι ένα σχήμα που χρειάζεται συχνά, το ορίζουμε ξεχωριστά.

Ένας κομψός τρόπος είναι να χρησιμοποιήσουμε την υπάρχουσα διαδικασία του πολυγώνου, οπότε η διαδικασία ΚΥΚΛΟΣ ορίζεται ως εξής:

```
ΤΟ ΚΥΚΛΟΣ :PLEYRA  
ΠΟΛΥΓΩΝΟ :PLEYRA 36  
END
```

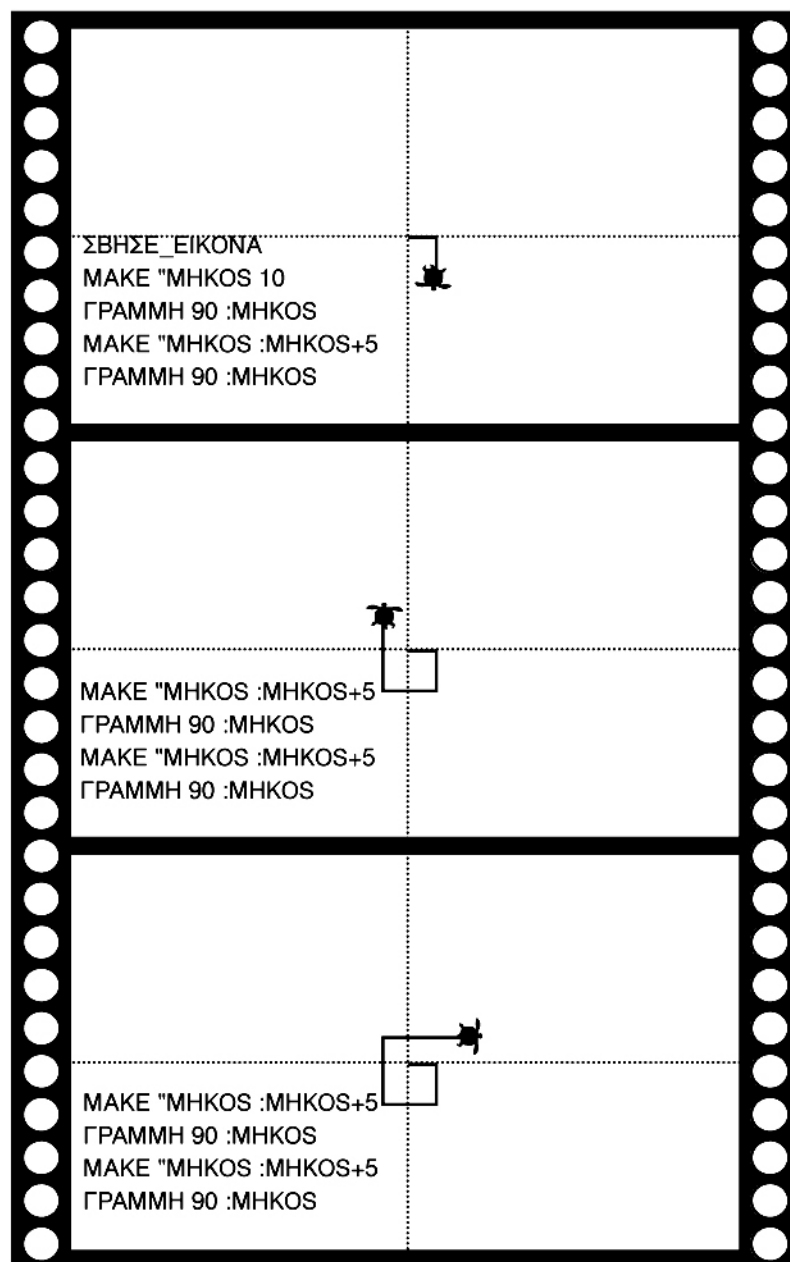


«Τα παιδιά... αν έρτιαξαν μόλις έναν κύκλο δίνοντας εντολές στη χελώνα να κάνει μικρά μπροστινά βήματα και μικρές δεξιές στροφές, είναι έτοιμα να επιχειρηματολογήσουν με το δάσκαλο ότι ο κύκλος είναι στην πραγματικότητα ένα πολύγωνο. Κάποιος που ακούει μια τέτοια συνομιλία στην πέμπτη τάξη της logo, δεν απομακρύνεται χωρίς να εντυπωσιαστεί από την ιδέα ότι η αλήθεια ή ψευδότητα της θεωρίας είναι δευτερεύουσας σημασίας ως προς τη συμβολή της στη μάθηση».

Seymour Papert

Επαναλαμβάνοντας τον εαυτό μας

Αυξάνοντας διαδοχικά την τιμή της μεταβλητής ΜΗΚΟΣ στη διαδικασία ΓΡΑΜΜΗ, το αποτέλεσμα μοιάζει με σπείρα.



Ο κόσμος του αύριο θα είναι ο κόσμος της τεχνητής νοημοσύνης. Μια τεχνική που χρησιμοποιείται ευρέως στον τομέα αυτό είναι η αναδρομή. Εμείς οι μεγάλοι καταλάβαμε την αναδρομή σε προχωρημένη ηλικία. Η αναδρομή δεν μας είναι οικεία όπως η επανάληψη. **Πιστεύουμε ότι αν μπολιάσουμε τον τρόπο σκέψης της γενιάς που θα ζήσει με τις νοήμονες μηχανές με την έννοια της αναδρομής, αυτό θα την εξοικειώσει με την τεχνητή νοημοσύνη.**

Μια αναδρομική διαδικασία

Η σειρά των διαταγών προς τη χελώνα αποτελείται από όμοια τμήματα δύο εντολών (με εξαίρεση τις πρώτες δύο).

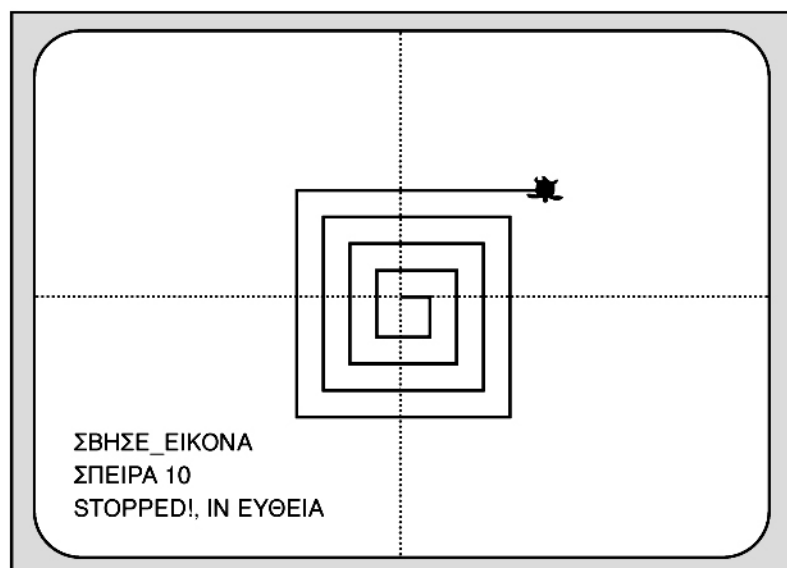
```
MAKE "ΜΗΚΟΣ 10
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
```

Η τεχνική κατά την οποία μέσα στον ορισμό μιας διαδικασίας χρησιμοποιείται η ίδια η διαδικασία ονομάζεται **αναδρομή**.

Θα μπορούσαμε να ορίσουμε μια διαδικασία (τη ΣΠΕΙΡΑ), η οποία αφού κάνει τη χελώνα να στρίψει 90 μοίρες, να χαράξει ένα ευθύγραμμο τμήμα ορισμένου μήκους και να αυξήσει την τιμή της μεταβλητής ΜΗΚΟΣ μήκους κατά 5, να καλέσει τον εαυτό της να εκτελεστεί.

```
ΤΟ ΣΠΕΙΡΑ :ΜΗΚΟΣ
  ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ
  MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + 5
  ΣΠΕΙΡΑ :ΜΗΚΟΣ
END
```

Η διαδικασία ΣΠΕΙΡΑ εφόσον κληθεί θα εκτελείται αενάως. Παρουσιάζεται λοιπόν η ανάγκη να διακοπεί αυτή η εκτέλεση για να επανακτήσει ο χρήστης τον έλεγχο του υπολογιστή. Αυτό γίνεται με την εντολή STOP. Η logo εμφανίζει στην οθόνη το μήνυμα που δείχνει σε ποιο σημείο της διαδικασίας ΣΠΕΙΡΑ έγινε η διακοπή.



Μια αναδρομική διαδικασία

Αναδρομή είναι η τεχνική κατά την οποία μέσα στον ορισμό μιας διαδικασίας χρησιμοποιείται η ίδια διαδικασία. Η αναδρομή γεννιέται από τη στιγμή που ένα σχήμα κατασκευάζεται από αναπαραγωγή του ίδιου του εαυτού του σε άλλη κλίμακα ή διαφορετικό επίπεδο, και αναδρομικότητα είναι η έκφραση μέσα από την αναδρομή. Αναδρομικές διαδικασίες συναντώνται στην καθημερινή εμπειρία με παραδείγματα την ανθρώπινη γλώσσα, τις ακρογιαλιές, τις νιφάδες του χιονιού, το φύλλωμα ενός δένδρου, και όλες τις μορφές φράκταλ από τους παλμούς της ανθρώπινης καρδιάς μέχρι την κατανομή των γαλαξιών και τις χασοτικές διαδικασίες.

Είναι επόμενο λοιπόν, η λύση προβλημάτων των μαθηματικών και της πληροφορικής να βασίζονται στον αναδρομικό τρόπο σκέψης,

με την αναδρομή να αποτελεί

βασική αρχή του

προγραμματισμού για εύκολη,

γρήγορη και κομψή

αντιμετώπιση προβλημάτων.

Ως εργαλείο για την κατανόηση

και εξοικείωση με την έννοια της

αναδρομής στην εκπαίδευση,

προτείνεται το ανοικτό

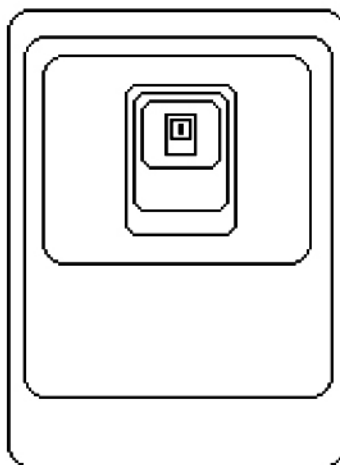
περιβάλλον της logo. Σ' αυτό

προγραμματιστικά προβλήματα

αντιμετωπίζονται με

απλούστερο τρόπο, και επιπλέον εμφανίζεται στο μαθητή ο

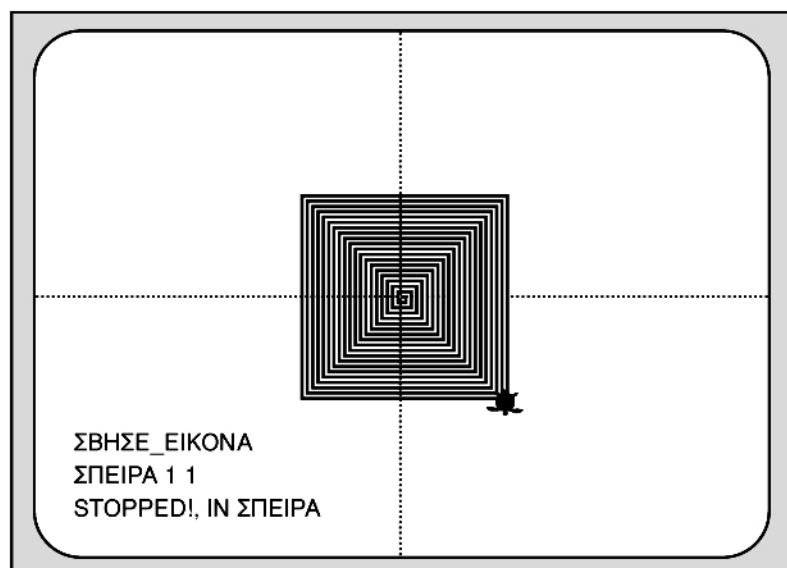
ανθρώπινος τρόπος σκέψης και η ομορφιά της φύσης.



Μορφή μέσα στη μορφή

Η προηγούμενη διαδικασία μπορεί να γίνει πιο γενική αν αντί του 5 θέσουμε μια νέα μεταβλητή, τη ΒΗΜΑ, που θα αντιστοιχεί στο ποσό που θα αλλάζει κάθε φορά το μήκος της γραμμής.

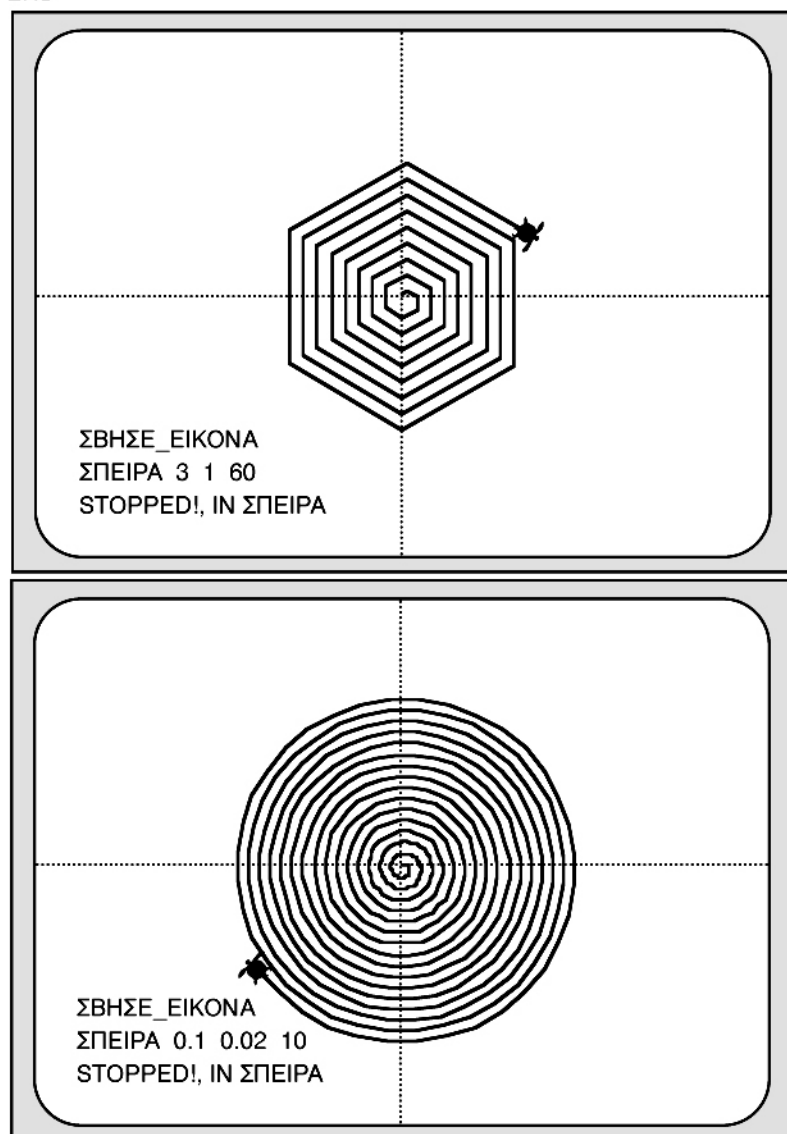
```
ΤΟ ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ  
ΓΡΑΜΜΗ 90 :ΜΗΚΟΣ  
MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ  
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ  
END
```



Δεσμώτες του ιλίγγου της αναδρομής

Μπορούμε να προσθέσουμε μια ακόμη μεταβλητή, τη MOIRES, που θα καθορίζει τη σταθερή γωνία που θα στρίβει η χελώνα.

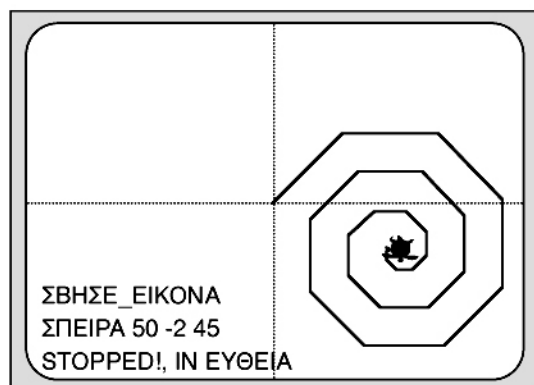
```
ΤΟ ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ  
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ  
MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ  
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ  
END
```



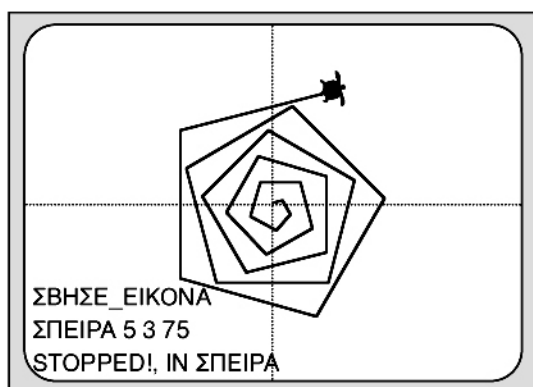
«Το μυαλό δεν μπορεί να συλλάβει στο σύνολό της την άπειρη επανάληψη της πολυπλοκότητας μέσα στον εαυτό της. Αλλά για κάποιον που σκέφτεται τη μορφή με τον τρόπο του γεωμέτρη, αυτό το είδος επανάληψης της δομής σε όλο και πιο μικρές κλίμακες μπορεί να αποκαλύψει έναν ολόκληρο κόσμο.»
James Gleick

Ασήμαντες αιτίες με μεγάλα αποτελέσματα

Παίζοντας με διάφορες τιμές στις παραμέτρους εισόδου της διαδικασίας, θα παρατηρήσουμε ποικιλία σχημάτων.



Αρνητική τιμή
στο βήμα κάνει
τη σπείρα να
στρέφεται προς
το εσωτερικό της.



«Η σύγχρονη
μελέτη του χάους
άρχισε με τη
βαθμιαία
συνειδητοποίηση,
στη δεκαετία του
1960, ότι πολύ
απλές
μαθηματικές
εξισώσεις
μπορούσαν να
απεικονίζουν
συστήματα βίαια
σαν ένα
καταρράκτη.

**Ελάχιστες
διαφορές στην
είσοδο
μπορούσαν
γρήγορα να
μετατραπούν σε
τεράστιες
διαφορές στην
έξοδο...»**

James Gleick



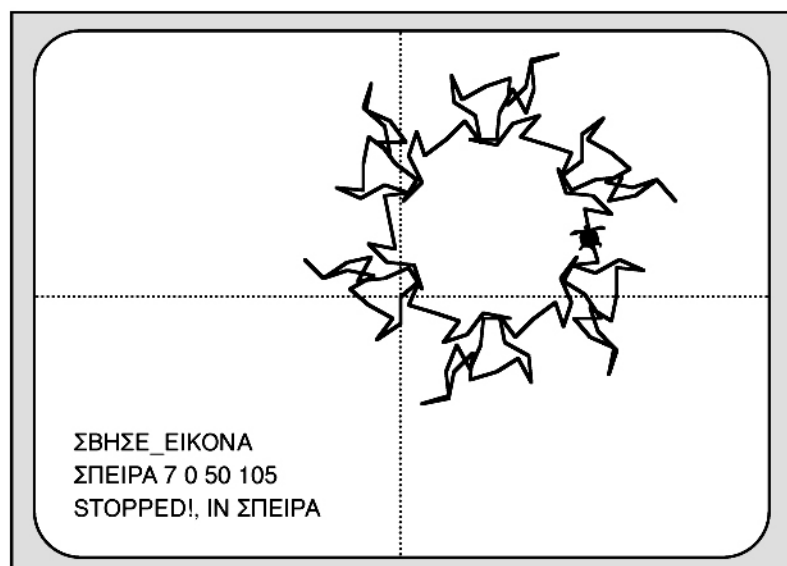
Ο χορός των μαγισών

Μπορούμε να τροποποιήσουμε την προηγούμενη αναδρομική διαδικασία δίνοντάς της τη δυνατότητα να μεταβάλλεται κάθε φορά η γωνία στροφής. Αυτό το πετυχαίνουμε προσθέτοντας μια ακόμη μεταβλητή, την D_GWNIA που θα καθορίζει το ποσό της μεταβολής της στροφής που θα κάνει η χελώνα. Έτσι, η διαδικασία ΣΠΕΙΡΑ διαμορφώνεται ως εξής:

```
TO ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ :D_GWNIA
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :ΜΗΚΟΣ
MAKE "ΜΗΚΟΣ :ΜΗΚΟΣ + :ΒΗΜΑ
MAKE "ΜΟΙΡΕΣ :ΜΟΙΡΕΣ + :D_GWNIA
ΣΠΕΙΡΑ :ΜΗΚΟΣ :ΒΗΜΑ :ΜΟΙΡΕΣ :D_GWNIA
END
```

Μια προσωπική εμπειρία.

Τα προγράμματα που φτιάχνουμε κατασκευάζονται αφού προσχεδιάσουμε τι ακριβώς θέλουμε να κάνουμε και πού να καταλήξουμε. Συχνά όμως δε μπορούμε να αντισταθούμε στον πειρασμό και να παίζουμε μ' αυτές τις διαδικασίες αλλάζοντας τις τιμές των μεταβλητών τους. Το αποτέλεσμα συνήθως μας αφήνει έκπληκτους. Ο υπολογιστής και αυτή η τόσο απλή γλώσσα -η logo- μας δίνουν την ευκαιρία να ξεπεράσουμε τη διαίσθηση και τη φαντασία μας. Βλέπουμε στην οθόνη πράγματα που μπορεί να τα ξέρουμε γιατί τα είχαμε προσεγγίσει μαθηματικά, αλλά ποτέ δεν τα είχαμε δει.

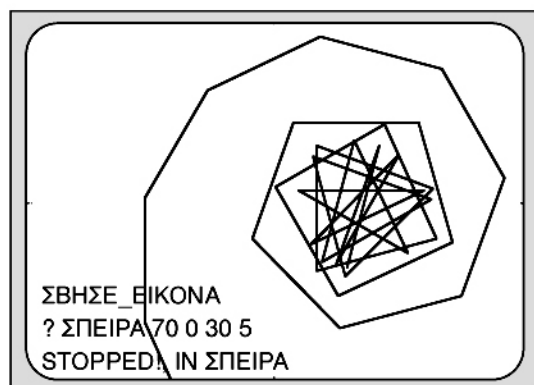


«"Θέλετε να πείτε» είπε το παιδί, «ότι πραγματικά δεν ξέρετε πως να το φτιάξετε;" Το παιδί δεν ήξερε ακόμα πως να το πει αλλά αυτό που είχε ανακαλύψει ήταν πως μαζί με τη δασκάλα είχαν ασχοληθεί με ερευνητικό έργο... Στο περιβάλλον της logo, τα παιδιά μαθαίνουν ότι και ο δάσκαλος είναι μαθητευόμενος και ότι όλοι μαθαίνουν από τα λάθη τους» Seymour Papert

«Ακόμα και η τάξη εμφανίζεται αυθόρμητα σε αυτά τα συστήματα - χάος και τάξη μαζί.» James Gleick

Το δένδρο και το δάσος

Μεταβάλλοντας μόνο μια παράμετρο όταν καλούμε τη διαδικασία μπορούμε να δούμε το δένδρο χωρίς να χάσουμε το δάσος.

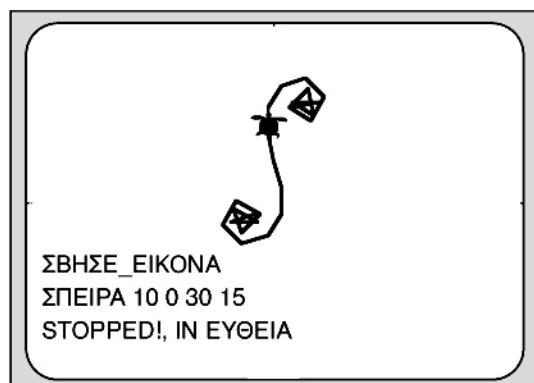


«Το απόλυτο μέγεθος των αντικειμένων ή η συγκεκριμένη χρονική διάρκεια της κίνησής τους δεν είναι αυτά που θα έπρεπε να μας ενδιαφέρουν. Η κεντρική ουσία της δομής και της δυναμικής πολλών αντικειμένων είναι ανεξάρτητη από την κλίμακα κάτω από την οποία τα παρατηρούμε. Η αποδέσμευσή μας από το ειδικό και τη λεπτομέρεια μας καθιστούν ικανούς να διακρίνουμε την παγκοσμιότητα των κοινών νόμων που διέπουν ένα μεγάλο πλήθος φαινομενικά διαφορετικών φυσικών συστημάτων.»

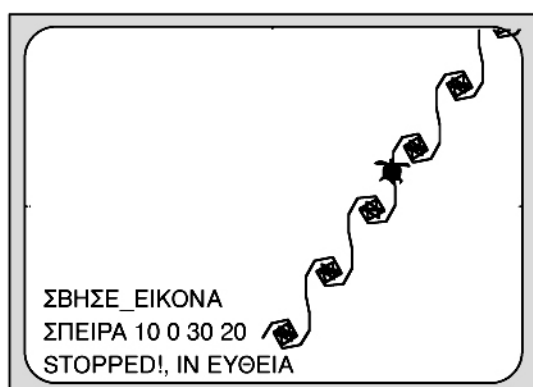
Τάσος Μπούνης,
από τον πρόλογο του "Χάος-Μιά νέα επιστήμη".

Μεταβάλλοντας τη μεταβολή της γωνίας

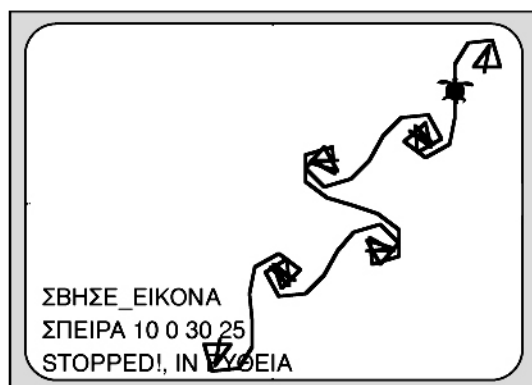
Μεταβάλλοντας την τιμή της μεταβολής της γωνίας παρατηρούμε τη χελώνα να διαγράφει διαφορετικές τροχιές κάθε φορά.



Εδώ η χελώνα επαναλαμβάνει αενάως τη διαδρομή που φαίνεται στο σχήμα, από τη μια άκρη στην άλλη.



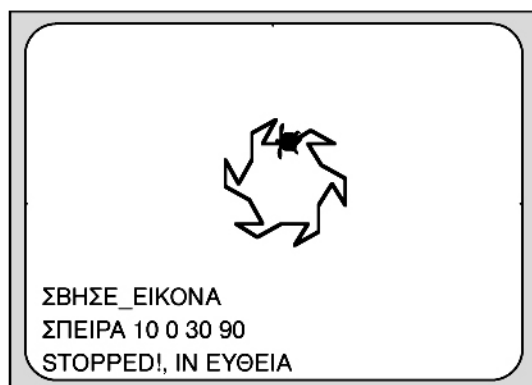
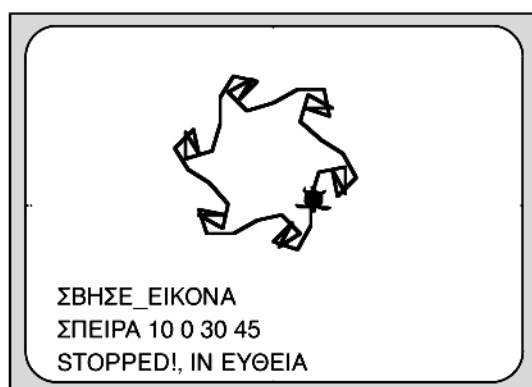
Μπορούμε να ομολογήσουμε στους μαθητές μας ότι δεν ξέρουμε πον θα καταλήξει η χελώνα. Μπορούμε όμως να επιχειρήσουμε να ψάξουμε μαζί τους μικραίνοντας την τιμή του μήκους ώστε η καμπύλη να χωράει στην οθόνη.



Η χελώνα επαναλαμβάνει αενάως την περισσότερο περίπλοκη διαδρομή που φαίνεται στο σχήμα, από την μια άκρη στην άλλη.

Μεταβάλλοντας τη μεταβολή της γωνίας

Συνεχίζουμε και εδώ να μεταβάλλουμε την τιμή της μεταβολής της γωνίας.



«Οι μελέτες του Μάντελμπροτ σχετικά με τις ακανόνιστες μορφές των φυσικών διαδικασιών και η εξερεύνηση των απεριόριστα πολυπλόκων σχημάτων βασίζονταν σε μια κοινή έννοια: στη σταθερότητα κλίμακας, στην ιδιότητα της αυτο-ομοιότητας».
James Gleick

Τα δομικά στοιχεία του σπιτιού: Ο όροφος

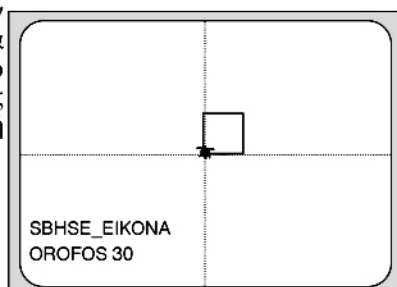
Ένα απλό σκίτσο σπιτιού αποτελείται τον όροφο, ένα τετράγωνο, και τη σκεπή, ένα τρίγωνο.

Ας δημιουργήσουμε μια διαδικασία που να σχεδιάζει έναν όροφο.

```
ΤΟ ΟΡΟΦΟΣ  
ΠΟΛΥΓΩΝΟ 30 4  
END
```

Μπορούμε να βελτιώσουμε την προηγούμενη διαδικασία αντικαθιστώντας το σταθερό αριθμό 30 που καθορίζει το ύψος του τετραγώνου με τη μεταβλητή YPSOS.

```
ΤΟ ΟΡΟΦΟΣ :YPSOS  
ΠΟΛΥΓΩΝΟ :YPSOS 4  
END
```



Η σκεπή, απεικονίζεται με ένα ισόπλευρο τρίγωνο.

Ας δημιουργήσουμε μια διαδικασία που να σχεδιάζει μια σκεπή.

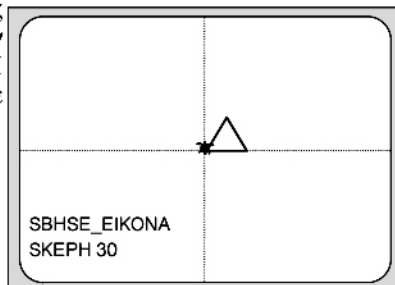
```
ΤΟ ΣΚΕΠΗ  
ΠΟΛΥΓΩΝΟ 30 3  
END
```

Παρόμοια με τη διαδικασία ΟΡΟΦΟΣ αντικαθιστούμε τον αριθμό 30 που καθορίζει το μήκος της πλευράς του τριγώνου με τη μεταβλητή PLEYRA.

```
ΤΟ ΣΚΕΠΗ :PLEYRA  
ΠΟΛΥΓΩΝΟ :PLEYRA 3  
END
```

Επειδή η διαδικασία ΟΡΟΦΟΣ θα συνδυαστεί με τη ΣΚΕΠΗ και επειδή η μεταβλητή YPSOS της πρώτης παίρνει την ίδια τιμή με τη μεταβλητή PLEYRA της δεύτερης, μετονομάζουμε στον ορισμό της διαδικασίας ΣΚΕΠΗ τη μεταβλητή PLEYRA σε YPSOS.

```
ΤΟ ΣΚΕΠΗ :YPSOS  
ΠΟΛΥΓΩΝΟ :YPSOS 3  
END
```



Συναρμολόγηση του όλου από τα μέρη

Ένας όροφος και μια σκεπή από μόνα τους δεν σχηματίζουν ένα σπίτι. Αφού δημιουργήσαμε τα τμήματα του σπιτιού θα πρέπει να τα συναρμολογήσουμε βάζοντάς τα στις σωστές τους θέσεις ώστε να σχηματιστεί το σπίτι.

Μπορούμε να σχεδιάσουμε πρώτα τον όροφο και μετά τη σκεπή.

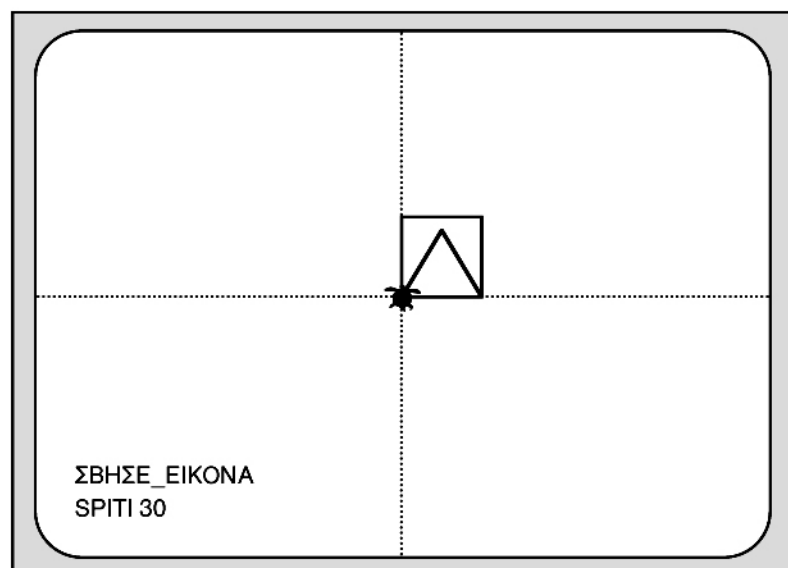
ΤΟ ΣΠΙΤΙ
ΟΡΟΦΟΣ
ΣΚΕΠΗ
END

Για τους λόγους που αναφέρθηκαν προηγουμένως η διαδικασία SPITI μπορεί να αντικατασταθεί με την παρακάτω:

ΤΟ ΣΠΙΤΙ :YPSOS
ΟΡΟΦΟΣ :YPSOS
ΣΚΕΠΗ :YPSOS
END

Όπως φαίνεται στο ίχνος που άφησε η χελώνα, τα πράγματα δεν ήρθαν όπως τα υπολογίσαμε.

Δεν πειράζει! Ας προσπαθήσουμε να καταλάβουμε **γιατί** έγινε το λάθος.



Μια διαδικασία μπορεί να περιέχει άλλες, οι οποίες εκτελούνται διαδοχικά με τη σειρά που εμφανίζονται.

Με την **ανάλυση** διακρίνουμε καταστάσεις που δε δηλώνονται άμεσα στην περιγραφή του αντικειμένου. Η ανάλυση μας επιτρέπει την επαναδόμηση του σχήματος με τα πραγματικά στοιχεία που δεν είναι τίποτε άλλο παρά μια **σύνθεση**.

«Να έχουν την ικανότητα οι μαθητές να κάνουν μαθηματικές ανακαλύψεις και γενικεύσεις»
B. Bloom

Βάζοντας τα πράγματα στη θέση τους

Για να εντοπίσουμε το λάθος στην προηγούμενη διαδικασία ας ακολουθήσουμε βήμα προς βήμα τις εντολές που εκτελεί η χελώνα.

Στην αρχή η πορεία της είναι ένα τετράγωνο, ο όροφος, με επιστροφή στην αφετηρία της. Στη συνέχεια από το σημείο που βρίσκεται σχεδιάζει ένα τρίγωνο, τη σκεπή.

Θα έπρεπε όμως πριν σχεδιάσει τη σκεπή να είχε μετακινηθεί στην πάνω αριστερή γωνία του ορόφου ώστε η σκεπή να τοποθετηθεί πάνω στον όροφο. Χρειάζεται λοιπόν μετά τον όροφο να γίνει αυτή η μετακίνηση με την εντολή ΕΥΘΕΙΑ και για μήκος όσο είναι το ύψος του ορόφου. Ας δούμε λοιπόν πως θα διορθώσουμε τη διαδικασία SPITI.

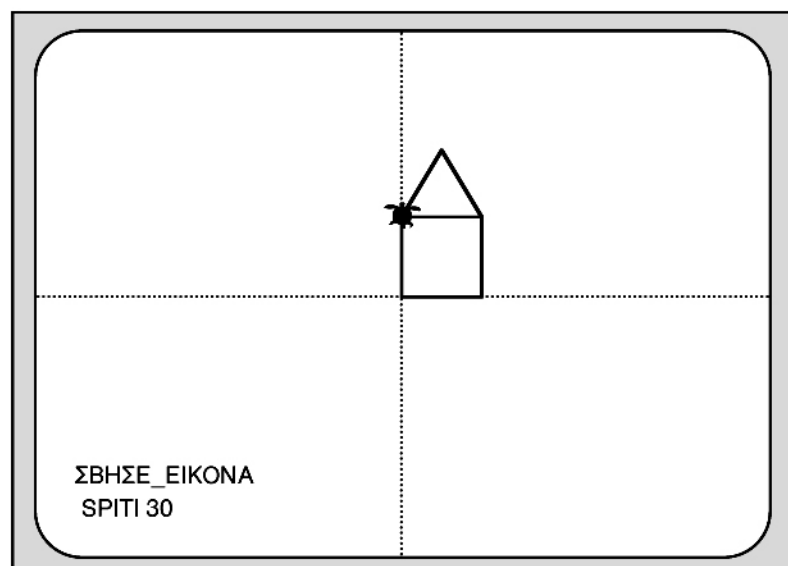
ΤΟ ΣΠΙΤΙ :YPSOS

ΟΡΟΦΟΣ :YPSOS

ΕΥΘΕΙΑ :YPSOS

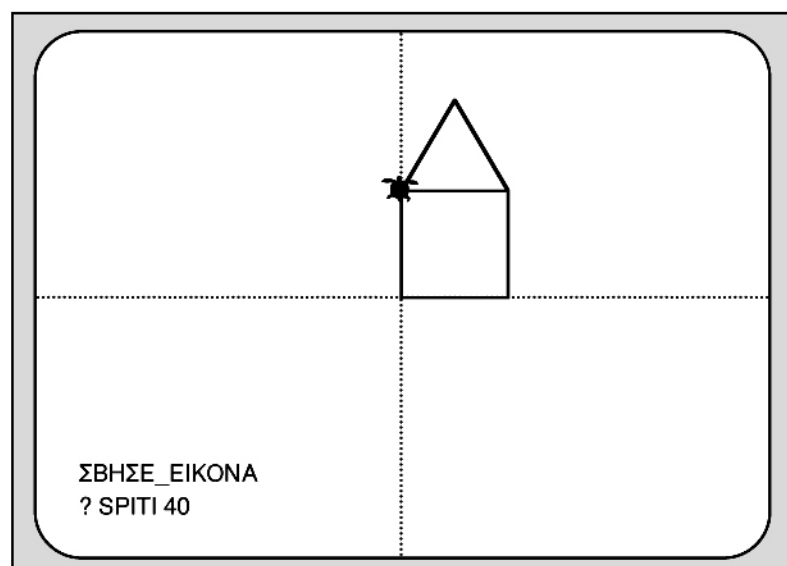
ΣΚΕΠΗ :YPSOS

END

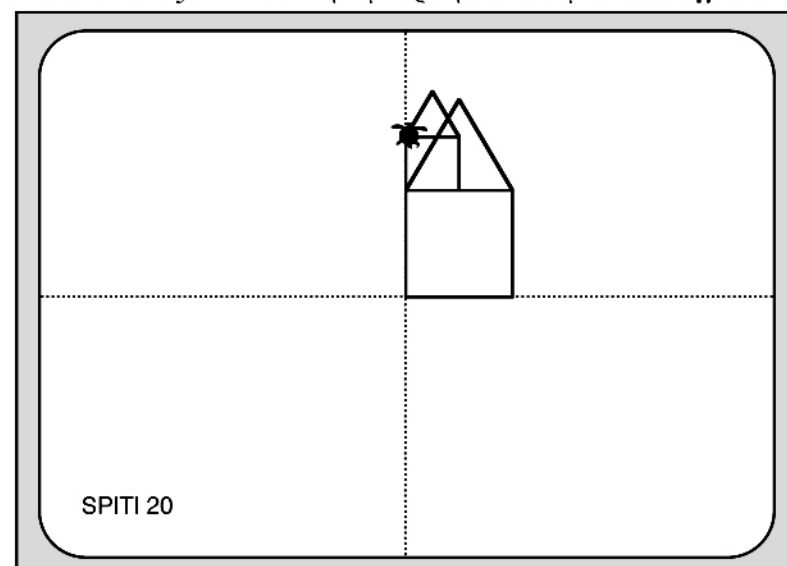


Διαπίστωση ότι κάτι δεν πάει καλά

Ας βάλουμε τη χελώνα να σχεδιάσει ένα σπίτι που ο όροφος να έχει ύψος 40.



Στη συνέχεια χωρίς να σβήσουμε το προηγούμενο σχήμα ας βάλουμε την χελώνα να σχεδιάσει ένα ακόμα σπίτι που ο όροφος να έχει ύψος 20. Βλέποντας το αποτέλεσμα μπορούμε να πούμε **τι δεν πήγε καλά;**



Πάμε πίσω...

Στην προηγούμενη σελίδα αυτό που δεν πήγε καλά ήταν το γεγονός ότι όταν τελείωσε η σχεδίαση ενός σπιτιού η χελώνα έμεινε στη στέγη κοιτάζοντας προς τα πάνω, αντί να κατεβεί στο έδαφος. Πρέπει λοιπόν να κάνουμε τη χελώνα να κινηθεί προς τα πίσω. Αυτό μπορούμε να το κάνουμε με δύο τρόπους.

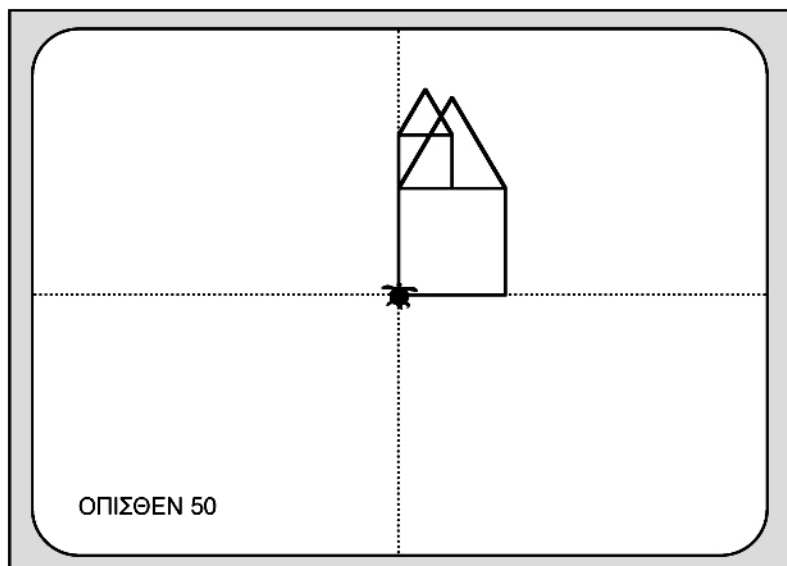
Ο πρώτος είναι να στρίψουμε τη χελώνα προς τα κάτω, να τη μετακινήσουμε ευθύγραμμα και μετά να την ξαναστρίψουμε ώστε να κοιτάζει προς τα πάνω. Αυτή η κίνηση γίνεται με την επόμενη διαδικασία που ονομάζουμε ΟΠΙΣΘΕΝ.

```
ΤΟ ΟΠΙΣΘΕΝ :ΜΗΚΟΣ  
ΔΕΞΙΑ 180  
ΕΥΘΕΙΑ :ΜΗΚΟΣ  
ΔΕΞΙΑ 180  
ΕΝΔ
```

Ο άλλος τρόπος για να κινηθεί η χελώνα με την όπισθεν είναι να χρησιμοποιήσουμε την εντολή BK της logo.

```
ΤΟ ΟΠΙΣΘΕΝ :ΜΗΚΟΣ  
BK :ΜΗΚΟΣ  
ΕΝΔ
```

Η εντολή BK της logo που υπάρχει στο προηγούμενο πρόγραμμα, προέρχεται από την ισοδύναμη εντολή **Back** που στα αγγλικά σημαίνει πίσω.



Λέγοντας στη χελώνα να κινηθεί προς τα πίσω θα πρέπει να της πούμε και πόσα βήματα να κάνει. Στο παράδειγμα που φαίνεται στην οθόνη είπαμε στη χελώνα να υποχωρήσει 50 βήματα και η χελώνα δεν κατάφερε να φτάσει στην κάτω αριστερή γωνία του σπιτιού. Μπορούμε να υπολογίσουμε πόσα βήματα πρέπει να υποχωρήσει ακόμα η χελώνα;

Με το παράδειγμα αυτό μπορούμε να κάνουμε αισθητή την αναγκαιότητα της λύσης της εξίσωσης πρώτου βαθμού.

Η μονοκατοικία

Ας ενσωματώσουμε τις προηγούμενες παρατηρήσεις στον ορισμό της διαδικασίας που σχεδιάζει ένα σπίτι και ας της αλλάξουμε το όνομα ονομάζοντάς τη ΜΟΝΟΚΑΤΟΙΚΙΑ.

```
ΤΟ ΜΟΝΟΚΑΤΟΙΚΙΑ :YPSOS
ΟΡΟΦΟΣ :YPSOS
ΕΥΘΕΙΑ :YPSOS
ΣΚΕΠΗ :YPSOS
ΟΠΙΣΘΕΝ :YPSOS
END
```

Για να καταλάβουμε τη δομή αυτής της διαδικασίας ας την «τρέξουμε με το χέρι».

Έστω ότι δίνουμε τη διαταγή ΜΟΝΟΚΑΤΟΙΚΙΑ 40.

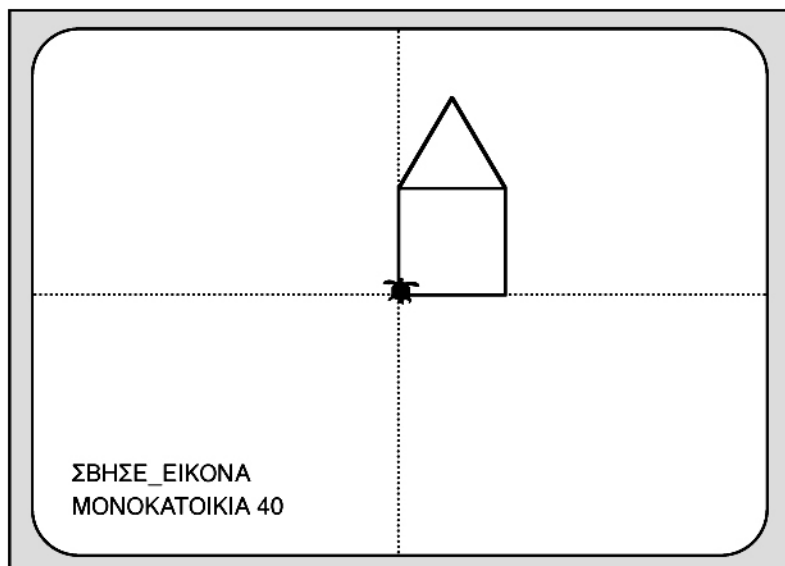
Ο αριθμός 40 μπαίνει ως τιμή στη μεταβλητή YPSOS.

Στη συνέχεια εκτελείται η πρώτη εντολή της διαδικασίας, η εντολή ΟΡΟΦΟΣ η οποία σχεδιάζει ένα τετράγωνο με πλευρά 40 και η χελώνα καταλήγει στην κάτω αριστερή γωνία του.

Με την εκτέλεση της επόμενης εντολής ΕΥΘΕΙΑ η χελώνα θα βρεθεί στην πάνω αριστερή γωνία.

Η εκτέλεση της εντολής ΣΚΕΠΗ θα σχεδιάσει ένα τρίγωνο και θα φέρει τη χελώνα πάλι στην επάνω αριστερή γωνία.

Τέλος η εντολή ΟΠΙΣΘΕΝ θα επιστρέψει τη χελώνα στην κάτω αριστερή γωνία από όπου είχε ξεκινήσει όταν άρχισε να εκτελεί τη διαδικασία ΜΟΝΟΚΑΤΟΙΚΙΑ 40.



«Μια πιο εμφανής διαφορά συνίσταται στο γεγονός ότι ορισμένες εντολές αφήνουν τη χελώνα στην αρχική της κατάσταση.

Προγράμματα γραμμένα με αυτό το καθαρό ύφος είναι πολύ πιο κατανοητά και μπορούν ευκολότερα να χρησιμοποιηθούν σε ποικίλες περιπτώσεις.

Και παρατηρώντας αυτές τις αρχές τα παιδιά παίρνουν δύο μαθήματα. Μαθαίνουν μια γενική "μαθητειακή αρχή", η οποία είναι να κάνουν τα συστατικά να διευκολύνουν την προσαρμογή. Και μαθαίνουν τη χρήση της πολύ δυναμικής ιδέας "κατάσταση»». Seymour Papert

Το διόροφο

Ας δούμε πως μπορούμε να μάθουμε τη χελώνα να σχεδιάζει ένα κτίριο με δύο ορόφους και σκεπή.

Δεν έχουμε παρά να προσθέσουμε έναν όροφο μετά το ισόγειο μαζί με την αντίστοιχη μετακίνηση προς τα πάνω. Φυσικά στο τέλος θα πρέπει να υπάρχει μια ακόμα ΟΠΙΣΘΕΝ η οποία θα αντιστοιχεί σε αυτή τη μετακίνηση του επιπλέον ορόφου.

ΤΟ ΔΙΟΡΟΦΟ :ΥPSOS

ΟΡΟΦΟΣ :ΥPSOS

ΕΥΘΕΙΑ :ΥPSOS

ΟΡΟΦΟΣ :ΥPSOS

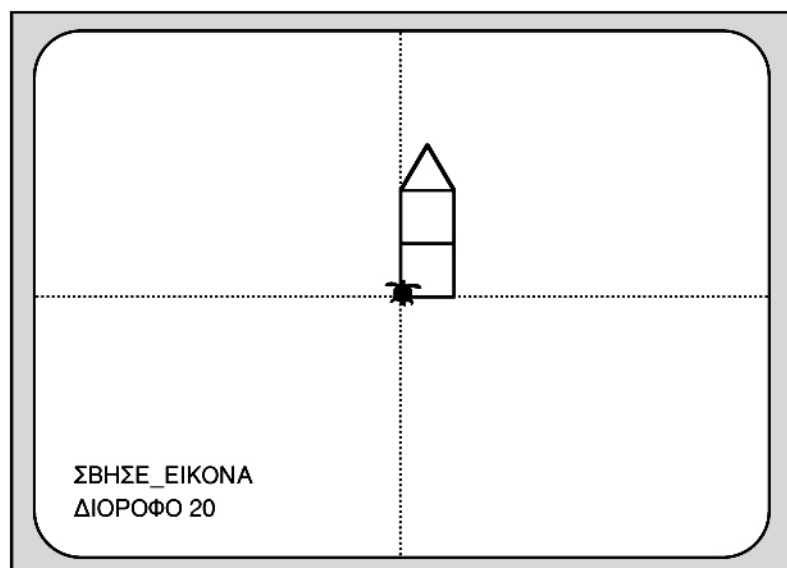
ΕΥΘΕΙΑ :ΥPSOS

ΣΚΕΠΗ :ΥPSOS

ΟΠΙΣΘΕΝ :ΥPSOS

ΟΠΙΣΘΕΝ :ΥPSOS

END



...και το τριόροφο

Ας βάλουμε έναν ακόμα όροφο στο προηγούμενο κτίριο.

Δεν έχουμε παρά να προσθέσουμε τον όροφο μετά τους άλλους δύο που υπάρχουν ήδη μαζί με την αντίστοιχη μετακίνηση προς τα πάνω. Φυσικά στο τέλος θα πρέπει να υπάρχει μια ακόμα ΟΠΙΣΘΕΝ η οποία θα αντιστοιχεί σ' αυτή τη μετακίνηση του επιπλέον ορόφου.

ΤΟ ΤΡΙΟΡΟΦΟ :YPSOS

ΟΡΟΦΟΣ :YPSOS

ΕΥΘΕΙΑ :YPSOS

ΟΡΟΦΟΣ :YPSOS

ΕΥΘΕΙΑ :YPSOS

ΟΡΟΦΟΣ :YPSOS

ΕΥΘΕΙΑ :YPSOS

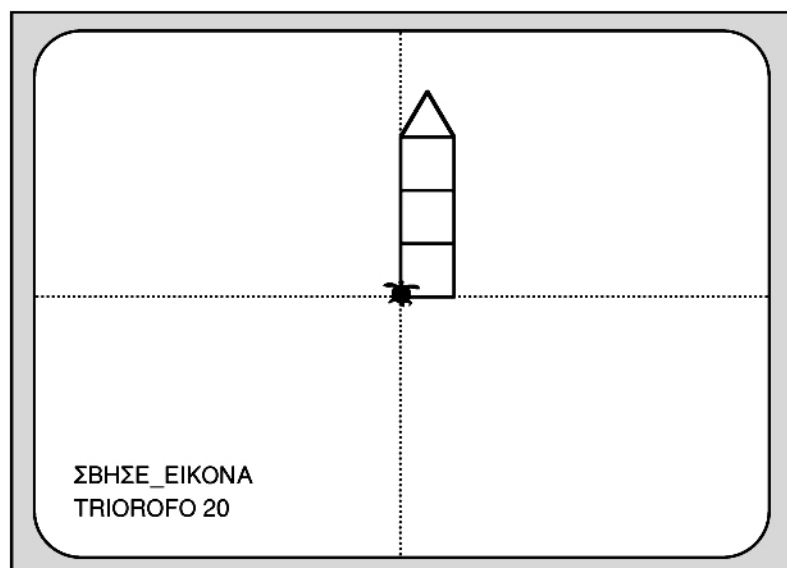
ΣΚΕΠΗ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS

ΟΠΙΣΘΕΝ :YPSOS

END



Ένα πολυόροφο κτίριο

Παρατηρούμε ότι στις προηγούμενες διαδικασίες το ζευγάρι των εντολών ΟΡΟΦΟΣ :YPSOS και ΕΥΘΕΙΑ :YPSOS επαναλαμβάνεται τόσες φορές όσοι είναι οι όροφοι του κτιρίου. Θα μπορούσαμε λοιπόν ολόκληρο το πακέτο αυτό με το παραπάνω ζευγάρι εντολών να το αντικαταστήσουμε με την εντολή:

REPEAT πλήθος ορόφων [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]

Παρόμοια η εντολή ΟΠΙΣΘΕΝ :YPSOS επαναλαμβάνεται τόσες φορές όσοι είναι οι όροφοι του κτιρίου. Θα μπορούσαμε λοιπόν να αντικαταστήσουμε αυτό το πακέτο εντολών με την εντολή:

REPEAT πλήθος ορόφων [ΟΠΙΣΘΕΝ :YPSOS]

Μπορούμε επομένως να αντικαταστήσουμε τις προηγούμενες διαδικασίες με μια, την ΠΟΛΥΟΡΟΦΟ, που είναι πιο γενική και χρησιμοποιεί μια ακόμα μεταβλητή, την PLHQOS_ΟΡΟΦΩΝ, που θα καθορίζει το πλήθος των ορόφων του κτιρίου.

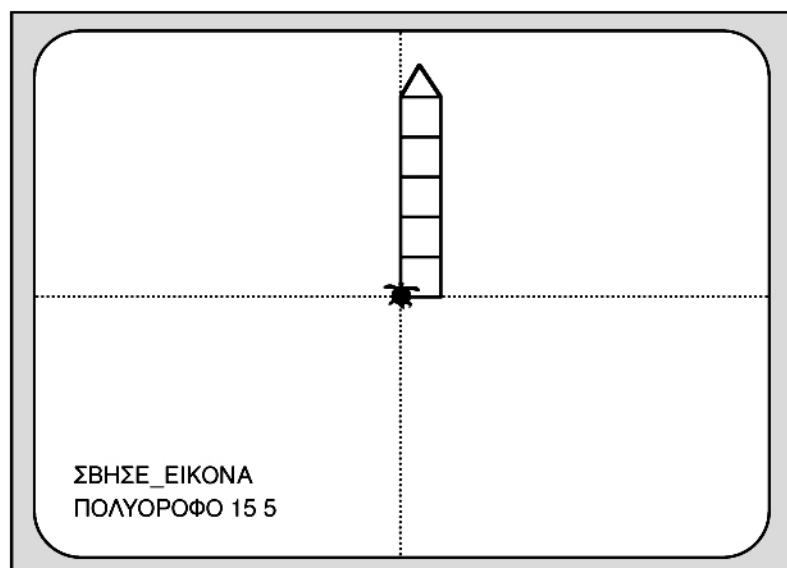
ΤΟ ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_ΟΡΟΦΩΝ

REPEAT :PLHQOS_ΟΡΟΦΩΝ [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]

ΣΚΕΠΗ :YPSOS

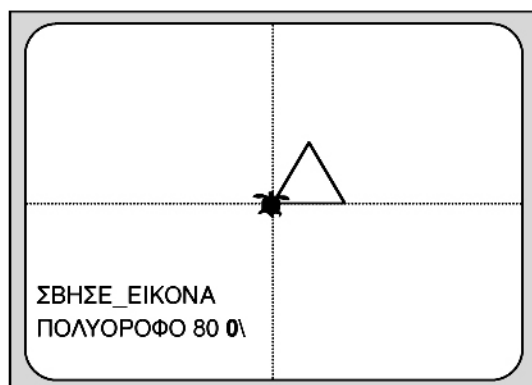
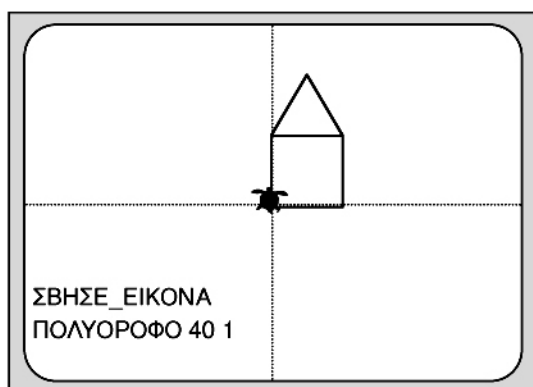
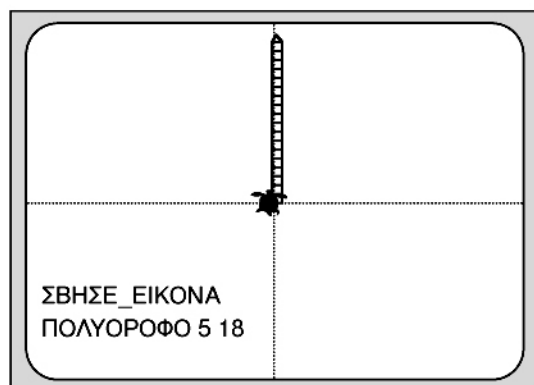
REPEAT :PLHQOS_ΟΡΟΦΩΝ [ΟΠΙΣΘΕΝ :YPSOS]

END



Ουρανοξύστης ή τσαντίρι;

Ας σχεδιάσουμε μερικά κτίρια δίνοντας διάφορες τιμές στο πλήθος των ορόφων και στο ύψος τους.



Ένα από τα σημεία στα οποία εντοπίζονται συχνά λάθη στον προγραμματισμό είναι οι περιπτώσεις που το πρόγραμμα παίρνει οριακές τιμές.

Ένα βελτιωμένο πολυόροφο κτίριο

Διαπιστώσαμε ότι η διαδικασία ΠΟΛΥΟΡΟΦΟ όταν λάβει το 0 ως τιμή της μεταβλητής PLHQOS_OROFWN σχεδιάζει μια σκεπή. Φυσικά δε νοείται ως κτίριο η σκεπή χωρίς έστω το ισόγειο.

Θα πρέπει λοιπόν το πρόγραμμα να λαμβάνει υπόψη του αυτόν τον περιορισμό. Έτσι,

αν το πλήθος των ορόφων είναι μεγαλύτερο του μηδενός

τότε σχεδιάζεται μόνο η σκεπή.

Στον προγραμματισμό αυτή την υπό συνθήκη ενέργεια την υλοποιούμε με την εντολή IF.

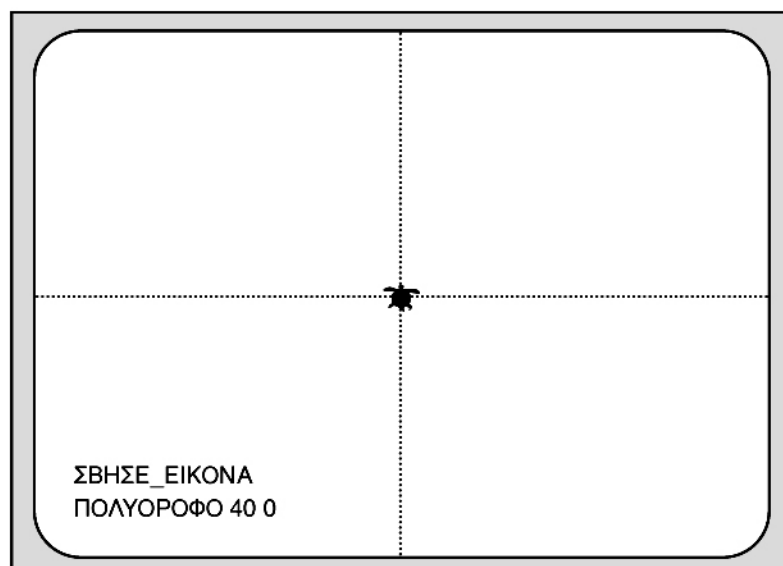
Η γενική σύνταξη της εντολής IF είναι:

IF ισχύει η λογική σχέση

THEN κάποιες ενέργειες

ELSE κάποιες άλλες ενέργειες.

```
ΤΟ ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
REPEAT :PLHQOS_OROFWN [ΟΡΟΦΟΣ :YPSOS ΕΥΘΕΙΑ :YPSOS]  
IF (:PLHQOS_OROFWN > 0) [ ΣΚΕΠΗ :YPSOS ]  
REPEAT :PLHQOS_OROFWN [ΟΠΙΣΘΕΝ :YPSOS]  
END
```



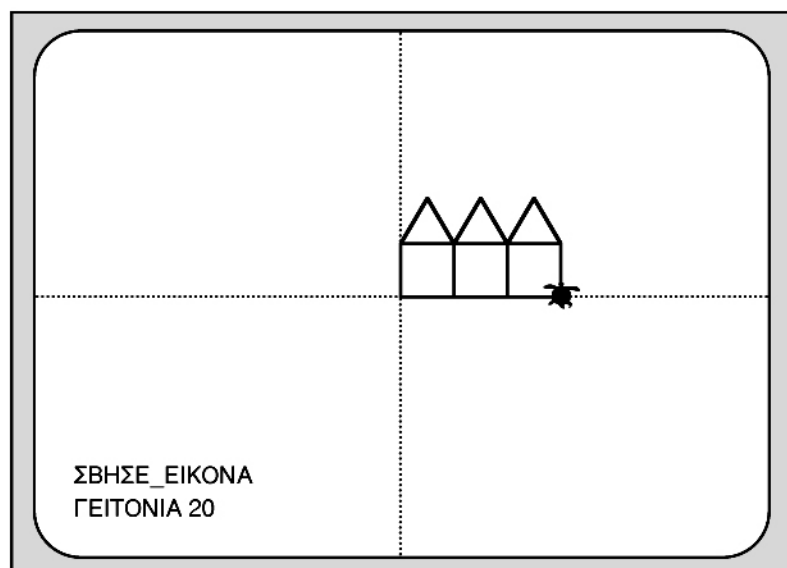
Μεταξύ διαφορετικών εκδόσεων της logo γενικά υπάρχουν διαφορετικοί τρόποι σύνταξης των εντολών. Στη LCSI-logo που χρησιμοποιείται εδώ, αν δεν υπάρχει κλάδος else, μετά τη λογική συνθήκη ακολουθούν μέσα σε αγκύλες οι εντολές που θα εκτελεσθούν αν ισχύει η συνθήκη.

Ας σχεδιάσουμε μια γειτονιά

Αφού μάθαμε τη χελώνα να σχεδιάζει μεμονωμένα σπίτια, μπορούμε τώρα να τη μάθουμε να σχεδιάζει περισσότερα έτσι ώστε να σχηματίζεται μια γειτονιά.

```
TO ΓΕΙΤΟΝΙΑ :YPSOS
MONOKATOΙΚΙΑ :YPSOS
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :YPSOS
ΑΡΙΣΤΕΡΑ 90
MONOKATOΙΚΙΑ :YPSOS
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :YPSOS
ΑΡΙΣΤΕΡΑ 90
MONOKATOΙΚΙΑ :YPSOS
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :YPSOS
ΑΡΙΣΤΕΡΑ 90
END
```

Σε τι χρησιμεύουν
οι εντολές
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :YPSOS
ΑΡΙΣΤΕΡΑ 90;



Μετατόπιση της χελώνας, απόπειρα πρώτη

Βλέπουμε στην προηγούμενη διαδικασία ότι το πακέτο των εντολών
ΔΕΞΙΑ 90

ΕΥΘΕΙΑ :ΥΡSOS

ΑΡΙΣΤΕΡΑ 90

επαναλαμβάνεται τρεις φορές και επιπλέον ότι χρησιμεύει για να μετακινεί τη χελώνα σε κάποιο διπλανό σημείο.

Μπορούμε με αυτές τις εντολές να ορίσουμε μια νέα διαδικασία την ΚΙΝΗΣΗ_ΔΙΠΛΑ, που θα μετακινεί τη χελώνα δίπλα.

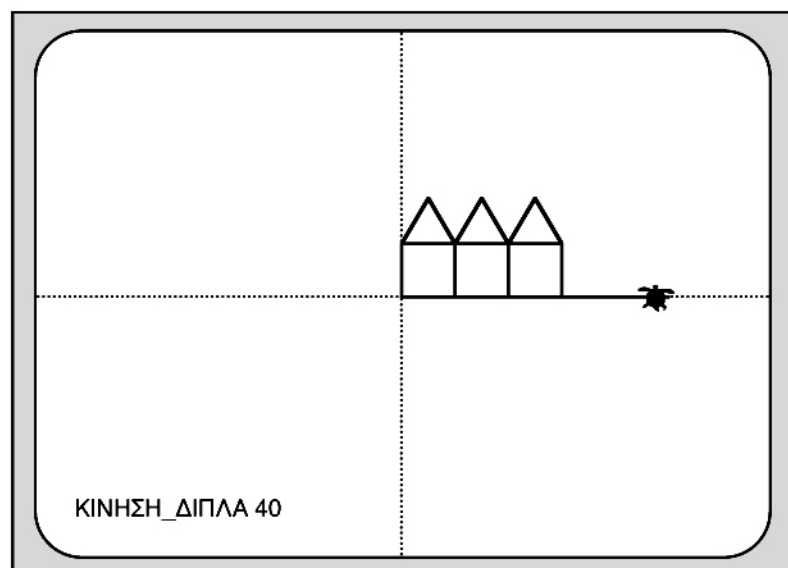
ΤΟ ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΑΡΟΣΤASH

ΔΕΞΙΑ 90

ΕΥΘΕΙΑ :ΑΡΟΣΤASH

ΑΡΙΣΤΕΡΑ 90

END



Μια πιο “κομψή” γειτονιά

Θα χρησιμοποιήσουμε τη διαδικασία ΚΙΝΗΣΗ_ΔΙΠΛΑ που ορίσαμε προηγουμένως μέσα στον ορισμό της διαδικασίας ΓΕΙΤΟΝΙΑ. Έτσι, η διαδικασία ΓΕΙΤΟΝΙΑ διαμορφώνεται ως εξής:

```
TO ΓΕΙΤΟΝΙΑ :ΥΡSOS  
  ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS
```

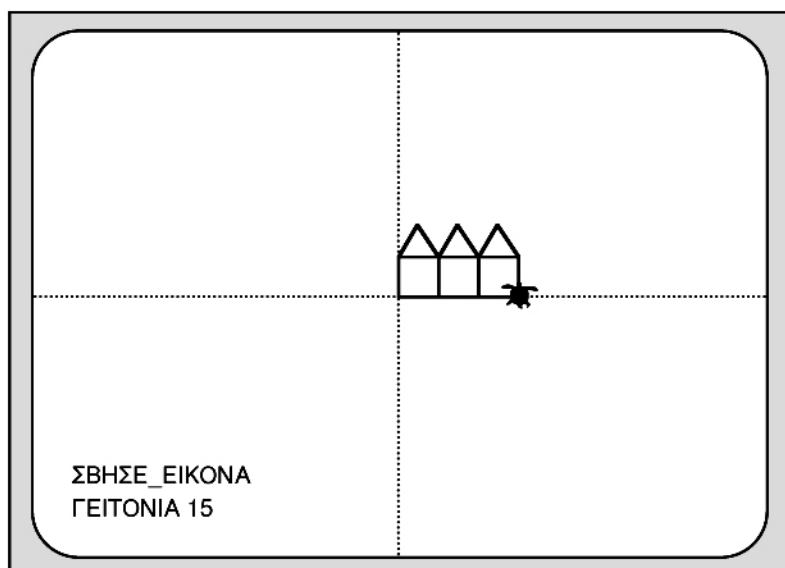
```
  ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS
```

```
  ΜΟΝΟΚΑΤΟΙΚΙΑ :ΥΡSOS  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΥΡSOS  
END
```

Περισσότερο ευανάγνωστο πρόγραμμα.

Συγκρίνετε τις δύο τελευταίες εκδόσεις της διαδικασίας ΓΕΙΤΟΝΙΑ. Η δεύτερη μπορεί να δώσει αρκετές πληροφορίες για το τι κάνει με μια απλή ανάγνωση έστω και σε κάποιον που δεν ξέρει από logo.

Τα προγράμματα πρέπει να είναι ευανάγνωστα και ευκολονόητα στον αναγνώστη-προγραμματιστή ώστε να μπορεί να αναζητήσει τα τυχόντα λάθη ή να κάνει βελτιώσεις πολύ εύκολα.



Μια γειτονιά αναπτύσσεται

Μπορούμε να κάνουμε περισσότερο ευέλικτη την προηγούμενη διαδικασία γενικεύοντάς την, έτσι ώστε το πλήθος των σπιτιών που θα σχεδιάζει η χελώνα να καθορίζεται όταν καλούμε τη διαδικασία να εκτελεστεί.

Αυτό γίνεται αν βάλουμε μέσα σε μια εντολή REPEAT το πακέτο των εντολών

MONOKATOΙΚΙΑ :YPSOS

ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS

που αντιπροσωπεύει κάθε σπίτι της γειτονιάς. Το πλήθος των επαναλήψεων είναι το πλήθος των σπιτιών που θα έχει η γειτονιά. Έτσι η διαδικασία ΓΕΙΤΟΝΙΑ διαμορφώνεται ως εξής:

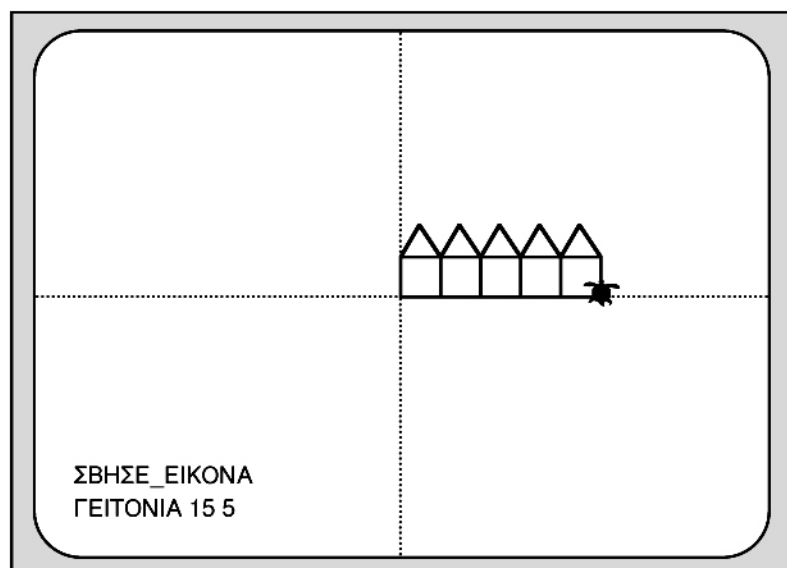
ΤΟ ΓΕΙΤΟΝΙΑ :YPSOS :PLHQOS_SPITIWN

REPEAT :PLHQOS_SPITIWN

[MONOKATOΙΚΙΑ :YPSOS

ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]

END



Από τη γειτονιά στη συνοικία

Στη γειτονιά ενός χωριού σχεδόν όλα τα σπίτια είναι μονοκατοικίες.

Πως θα βάλουμε τη χελώνα να σχεδιάσει τη συνοικία μιας πόλης όπου κατά κανόνα υπάρχουν πολυόροφα κτίρια;

Απλά στη διαδικασία ΓΕΙΤΟΝΙΑ θα αντικαταστήσουμε τη ΜΟΝΟΚΑΤΟΙΚΙΑ με το ΠΟΛΥΟΡΟΦΟ. Φυσικά επειδή η διαδικασία ΠΟΛΥΟΡΟΦΟ για να ξεκινήσει χρειάζεται να έχει και το πλήθος των ορόφων του κτιρίου με την εντολή MAKE θα δώσουμε στη μεταβλητή PLHQOS_OROFWN έστω την τιμή 4.

Ας δούμε πως ορίζεται η διαδικασία ΣΥΝΟΙΚΙΑ.

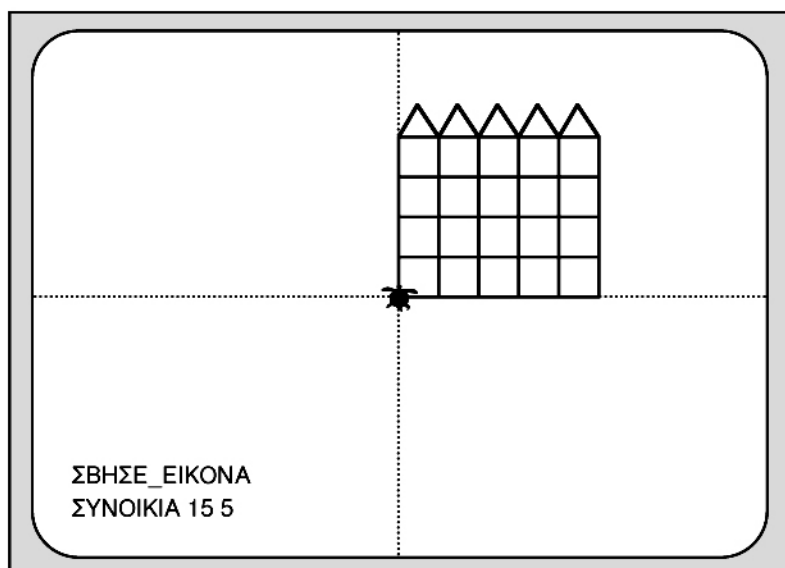
```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN
```

```
MAKE :PLHQOS_OROFWN 4
```

```
REPEAT :PLHQOS_SPITIWN
```

```
  [ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]
```

```
END
```



Μια πιο πραγματική εικόνα συνοικίας

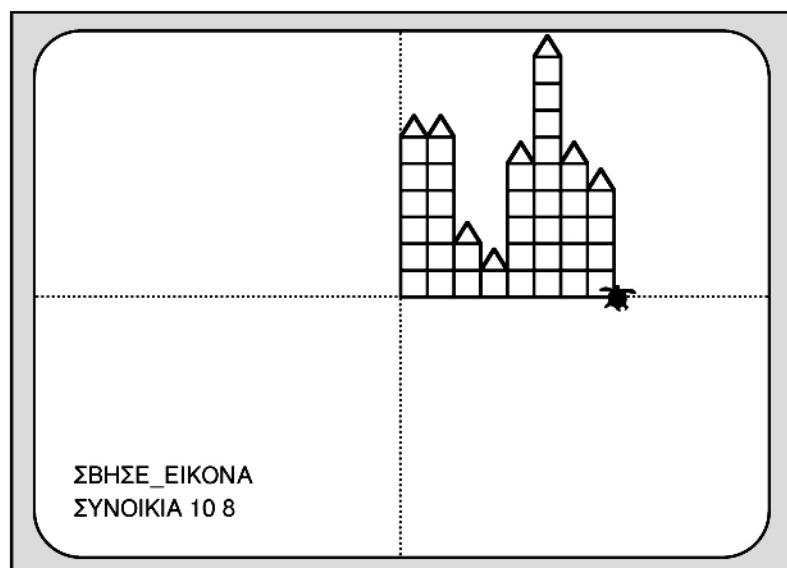
Στην προηγούμενη διαδικασία που σχεδιάζει μια συνοικία, όλα τα κτίρια είναι όμοια. Θα θέλαμε να αφήσουμε την τύχη να αποφασίζει πόσους ορόφους θα έχει κάθε κτίριο της συνοικίας. Αυτό γίνεται αν η μεταβλητή PLHQOS_OROFWN παίρνει τιμή με τη βοήθεια της συνάρτησης RANDOM. Ας δεχθούμε ότι δεν επιτρέπεται κανένα κτίριο να έχει πάνω από 10 ορόφους (δηλ. το μέγιστο επιτρεπόμενο όριο είναι οι 9 ορόφοι).

Ας δούμε πως θα βελτιωθεί η διαδικασία ΣΥΝΟΙΚΙΑ με τη χρήση της συνάρτησης RANDOM.

```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN  
REPEAT :PLHQOS_SPITIWN  
  [MAKE "PLHQOS_OROFWN RANDOM 10  
  ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]  
END
```

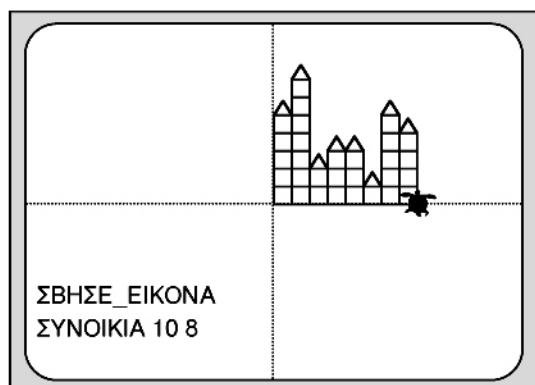
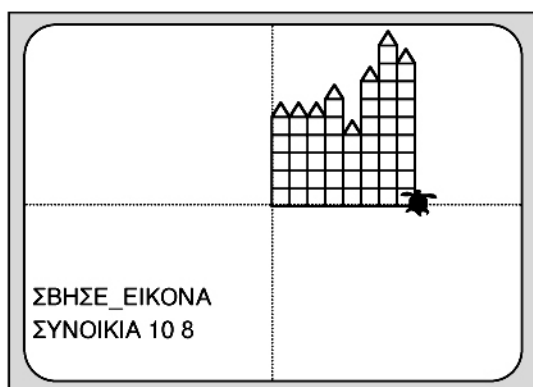
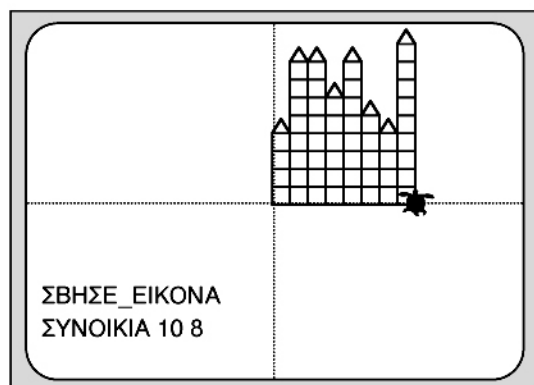
Η συνάρτηση RANDOM επιστρέφει στην τύχη έναν ακέραιο αριθμό που κυμαίνεται από το 0 μέχρι τον αμέσως μικρότερο ακέραιο του αριθμού με τον οποίο την τροφοδοτούμε.

Η συνάρτηση RANDOM 5, μπορεί να επιστρέψει έναν από τους παρακάτω αριθμούς: 0, 1, 2, 3, 4.



Το τυχαίο και η ομορφιά του (1)

Θα δούμε τώρα μερικές συνοικίες που σχεδιάζονται με την ίδια διαδικασία η οποία καλείται με τις ίδιες τιμές των παραμέτρων.



Μια γενίκευση της διαδικασίας

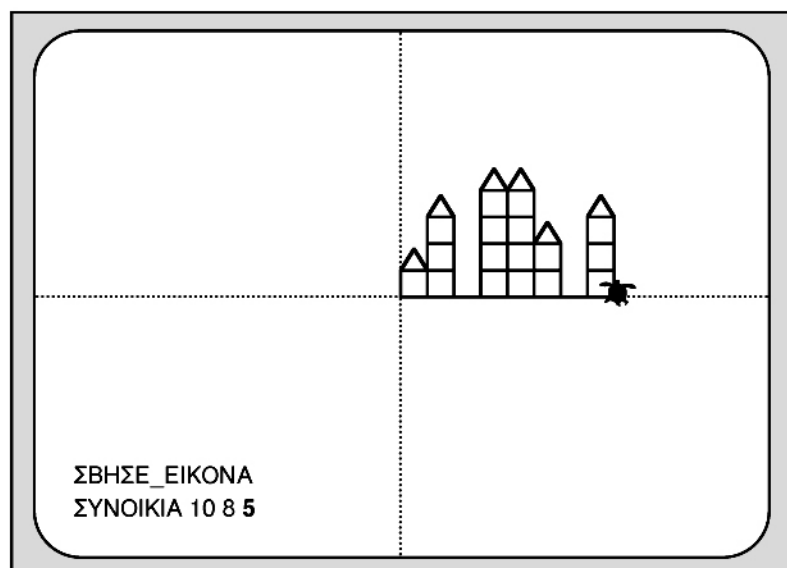
Στην προηγούμενη διαδικασία ως όριο των ορόφων χρησιμοποιείται στη συνάρτηση RANDOM το 10. Κάθε φορά που επιθυμούμε όμως να αλλάξουμε αυτό το όριο θα πρέπει να επεμβαίνουμε στο πρόγραμμα και να βάζουμε κάποια άλλη τιμή.

Για να γίνει περισσότερο ευέλικτη η διαδικασία, πρέπει να αντικαταστήσουμε τη σταθερή τιμή 10 με μια μεταβλητή την ORIO_OROF που θα δηλώνει το όριο του πλήθους των ορόφων.

Έτσι, η διαδικασία τροποποιείται ως εξής:

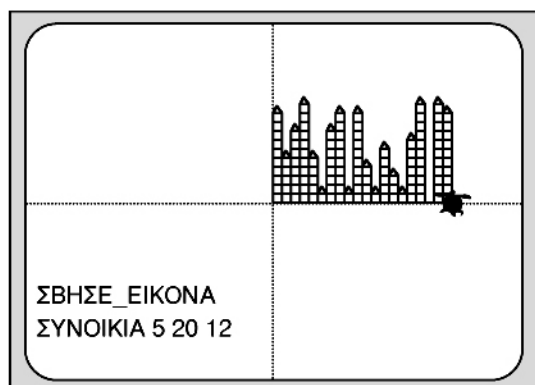
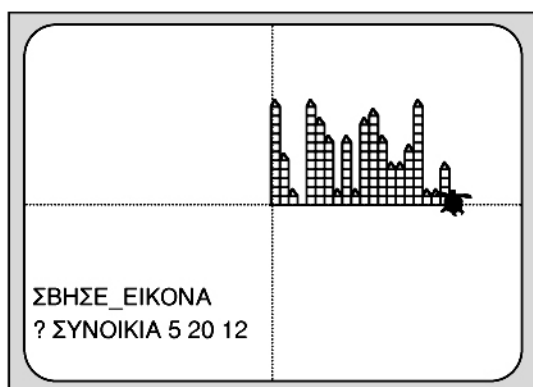
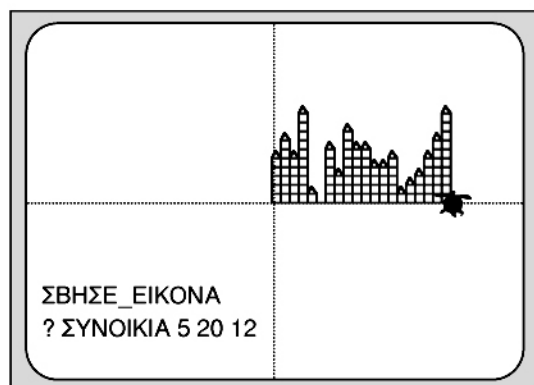
```
TO ΣΥΝΟΙΚΙΑ :YPSOS :PLHQOS_SPITIWN :ORIO_OROF
REPEAT :PLHQOS_SPITIWN
  [MAKE "PLHQOS_OROFWN RANDOM ORIO_OROF
  ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN
  ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]
END
```

Για να
αποφύγουμε
συνεχείς
επεμβάσεις στον
κώδικα του
προγράμματος,
εισάγουμε ως
μεταβλητές
εισόδου τις
επιθυμητές
παραμέτρους.



Το τυχαίο και η ομορφιά του (2)

Ας διασκεδάσουμε βάζοντας τη χελώνα να σχεδιάσει μερικές συνοικίες.



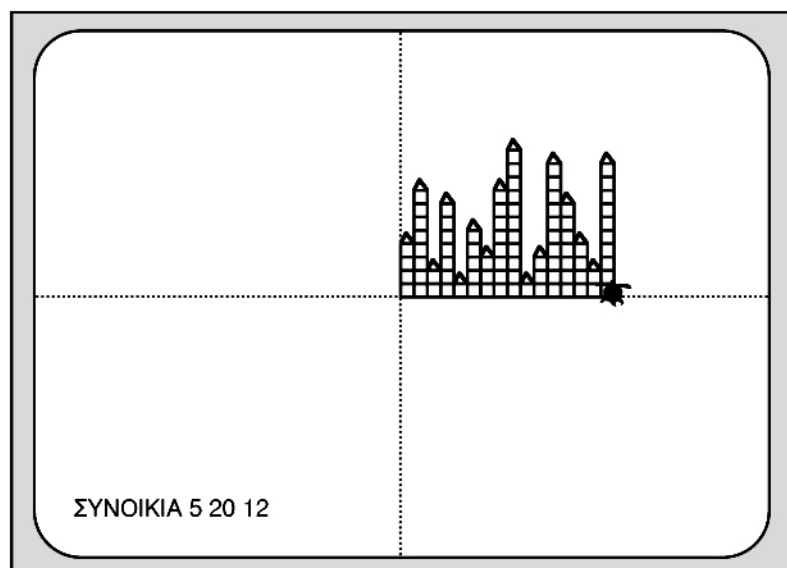
Μια ακόμα βελτίωση της συνοικίας

Οι προηγούμενες συνοικίες είχαν ένα εκ των προτέρων καθορισμένο πλήθος κτιρίων. Τι θα λέγατε αν θέλαμε να έχουμε τυχαίο αριθμό κτιρίων σε μια συνοικία;

Όταν θα καλούμε τη διαδικασία, θα της δίνουμε ένα μέγιστο πλήθος κτιρίων που θα μπορεί να έχει και το πλήθος των κτιρίων θα προκύπτει τυχαία με τη χρήση της συνάρτησης RANDOM.

Ας δούμε πως θα διαμορφωθεί η διαδικασία ΣΥΝΟΙΚΙΑ.

```
TO ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF
MAKE "PLHQOS_SPITIWN RANDOM :ORIO_SPITIWN
REPEAT :PLHQOS_SPITIWN
  [MAKE "PLHQOS_OROFWN RANDOM ORIO_OROF
   ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN
   ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]
END
```



Η πόλη

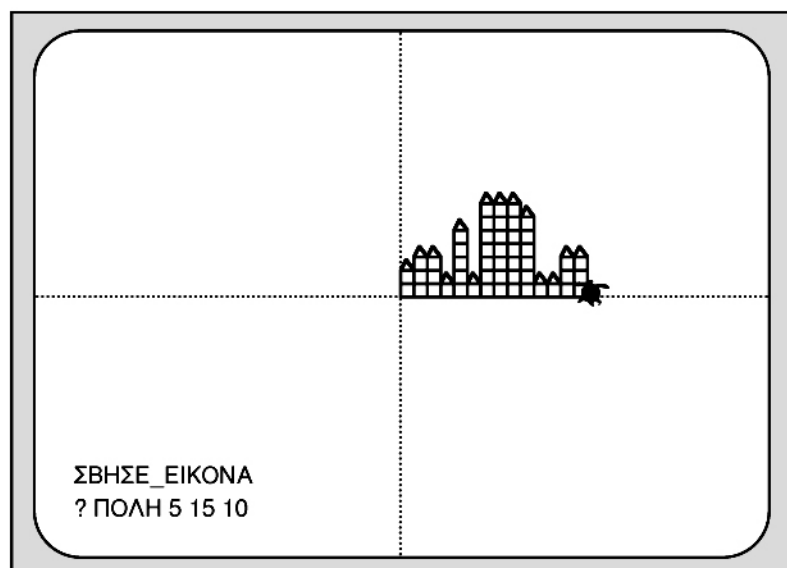
Μια συνοικία είναι ένα τμήμα μιας πόλης. Θα μπορούσαμε να διδάξουμε τη χελώνα να σχεδιάζει μια πόλη βάζοντας μια διαδικασία να σχεδιάζει πολλές συνοικίες.

Θα εκμεταλλευτούμε την εμπειρία που αποκτήσαμε προηγουμένως με τη συνάρτηση RANDOM και θα χρησιμοποιήσουμε τη μεταβλητή ORIO_SYNK ως την ανώτατη τιμή του πλήθους των συνοικιών που θα μπορεί να έχει η πόλη.

Ας δούμε τη διαδικασία ΠΟΛΗ.

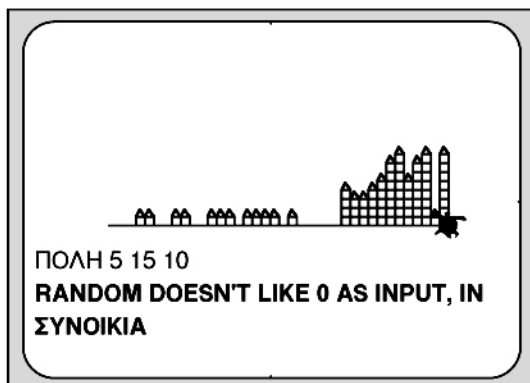
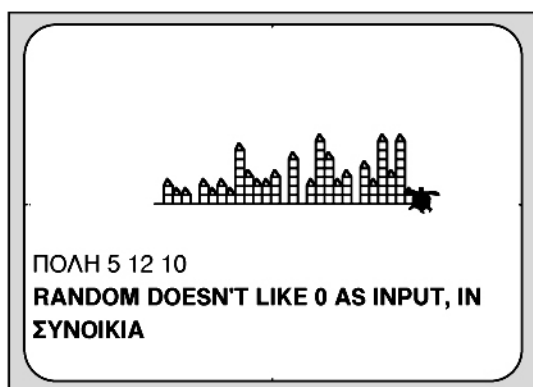
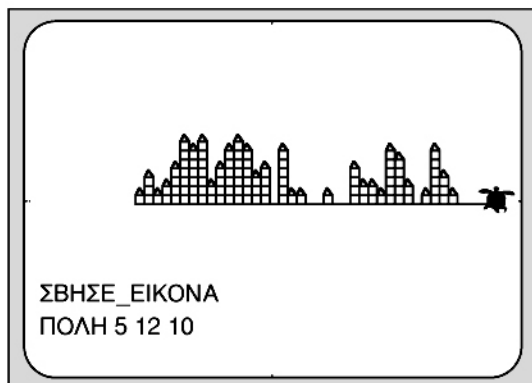
```
TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10  
    ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]  
END
```

Στη διαδικασία ΠΟΛΗ υπάρχουν τρεις απρόβλεπτοι παράγοντες: Το πόσες συνοικίες θα έχει η πόλη (RANDOM ORIO_SYNK), το πόσα κτίρια θα έχει κάθε συνοικία (RANDOM ORIO_SPT_SYNK) και το πόσους ορόφους θα έχει κάθε κτίριο (RANDOM 10).



Το τυχαίο και η ομορφιά του (3)

Ας βάλουμε τη χελώνα να σχεδιάσει μερικές πόλεις.



Εντοπισμός και διόρθωση του σφάλματος

Στη συγκεκριμένη περίπτωση το μήνυμα «RANDOM doesn't like 0 as input, in ΣΥΝΟΙΚΙΑ» μας πληροφορεί ότι δόθηκε το 0 ως στοιχείο εισόδου της συνάρτησης RANDOM της διαδικασίας ΣΥΝΟΙΚΙΑ.

Μάλιστα παρατηρώντας το σχέδιο που έφτιαξε η χελώνα μπορούμε να υποθέσουμε ότι πριν τύχει να δοθεί το 0, η διαδικασία συνοικία πόλης θα πρέπει να είχε εκτελεστεί αρκετές φορές.

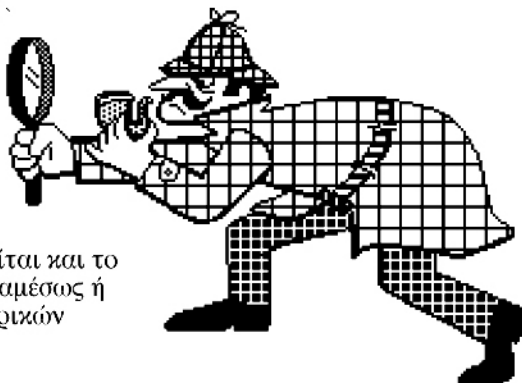
Ας δούμε λοιπόν πως συνδέονται οι διαδικασίες ΠΟΛΗ και ΣΥΝΟΙΚΙΑ.

```
TO ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10  
    ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]  
END  
↓  
TO ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF  
MAKE "PLHQOS_SPITIWN RANDOM :ORIO_SPITIWN  
REPEAT :PLHQOS_SPITIWN  
  [MAKE "PLHQOS_OROFWN RANDOM ORIO_OROF  
    ΠΟΛΥΟΡΟΦΟ :YPSOS :PLHQOS_OROFWN  
    ΚΙΝΗΣΗ_ΔΙΠΛΑ :YPSOS]  
END
```

Το πρόβλημα δηλαδή εμφανίζεται σε ένα από τα δύο RANDOM. Το πρώτο RANDOM έχει σχέση με τη μεταβλητή ORIO_SPT_SYNK η οποία διατηρεί την τιμή που πήρε από την αρχή όταν ο χρήστης διέταξε τη χελώνα να σχεδιάσει την ΠΟΛΗ. Άρα το λάθος πρέπει να είναι σε κάποιο 0 που μπαίνει ως τιμή στη μεταβλητή ORIO_OROF. Όμως η μεταβλητή ORIO_OROF της διαδικασίας ΣΥΝΟΙΚΙΑ παίρνει τιμή από τη μεταβλητή ORIO_OROF της διαδικασίας ΠΟΛΗ. Άρα το 0 πρέπει να προέρχεται από εκεί. Στη διαδικασία ΠΟΛΗ η μεταβλητή ORIO_OROF παίρνει τιμή στην εντολή

MAKE "ORIO_OROF RANDOM 10.

Πράγματι οι πιθανές τιμές που μπορεί να πάρει η μεταβλητή ORIO_OROF είναι 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Έτσι εξηγείται και το γεγονός ότι το σφάλμα δεν εμφανίζεται αμέσως ή και καθόλου αλλά μετά την σχεδίαση μερικών συνοικιών.



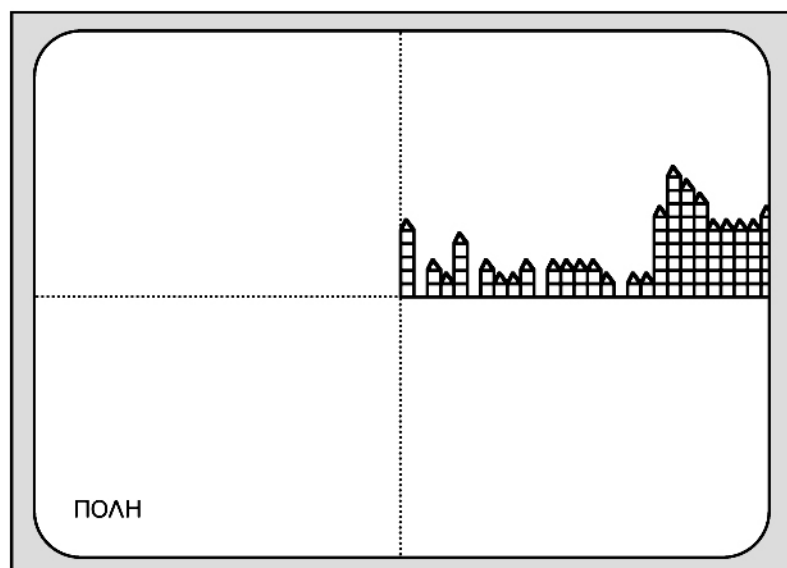
Η «διορθωμένη» πόλη

Όπως είδαμε θα πρέπει να κάνουμε τις απαραίτητες αλλαγές στο πρόγραμμα έτσι ώστε να αποκλείσουμε την περίπτωση η μεταβλητή ORIO_OROF να παίρνει την τιμή 0. Ένας απλός τρόπος είναι στην τιμή που θα προκύψει από τη συνάρτηση RANDOM να προσθέσουμε μια μονάδα μετασχηματίζοντας τις τιμές ως εξής:

0 + 1	μετασχηματίζεται σε	1
1 + 1	μετασχηματίζεται σε	2
2 + 1	μετασχηματίζεται σε	3
3 + 1	μετασχηματίζεται σε	4
4 + 1	μετασχηματίζεται σε	5
5 + 1	μετασχηματίζεται σε	6
6 + 1	μετασχηματίζεται σε	7
7 + 1	μετασχηματίζεται σε	8
8 + 1	μετασχηματίζεται σε	9
9 + 1	μετασχηματίζεται σε	10

```

ΤΟ ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN
REPEAT :PLHQOS_SYNOIKIWN
  [MAKE "ORIO_OROF RANDOM 10
  MAKE "ORIO_OROF :ORIO_OROF + 1
  ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF]
END
    
```



Ένα από τα περισσότερα σημαντικά καθήκοντα του προγραμματιστή είναι να εντοπίσει το λάθος και να το διορθώσει. Στον εντοπισμό του λάθους συνήθως μας βοηθάει ο υπολογιστής με διάφορα μηνύματα τα οποία μας πληροφορούν για το είδος του λάθους και το σημείο όπου κάτι δεν πήγε καλά. Η δουλειά του προγραμματιστή είναι να βρει την αιτία που δημιούργησε το λάθος και η οποία μπορεί να βρίσκεται όχι στο σημείο που παρουσιάστηκε η ανωμαλία αλλά κάπου αλλού.

Πόσες συνοικίες έχει αυτή η πόλη;

Κενές εκτάσεις μεταξύ συνοικιών

Στη διαδικασία της πόλης η μια συνοικία κολάει ακριβώς δίπλα στην άλλη με αποτέλεσμα να μην ξεχωρίζουν μεταξύ τους. Θα θέλαμε λοιπόν μεταξύ των συνοικιών να υπάρχει ένα κενός χώρος. Για το λόγο αυτό όταν τελειώνει μια συνοικία πρέπει η χελώνα να μετακινείται δεξιότερα χωρίς να αφήνει ίχνη. Αυτό γίνεται με την εντολή **PenUp** ή **PU** που σημαίνει σήκωσε την πένα (και μη γράφεις). Εμείς προσπαθώντας να εξελληνίσουμε το περιβάλλον της Logo θα ορίσουμε τη διαδικασία **ΜΗ_ΓΡΑΦΕΙΣ**. Η αντίθετη ενέργεια κατά την οποία η χελώνα κατεβάζει την πένα και μπορεί να αφήνει ίχνη γίνεται με την εντολή **Pen Down** ή **PD** και την εξελληνίζουμε με τη διαδικασία **ΓΡΑΨΕ**.

ΤΟ ΜΗ_ΓΡΑΦΕΙΣ

PU

END

ΤΟ ΓΡΑΨΕ

PD

END

Χρησιμοποιώντας αυτές τις δύο εντολές-διαδικασίες μαζί με τη διαδικασία **ΚΙΝΗΣΗ_ΔΙΠΛΑ**, μπορούμε να ορίσουμε τη διαδικασία **ΚΕΝΟ** που να μετακινεί δεξιότερα τη χελώνα χωρίς να αφήνει ίχνη στη διάρκεια της μετακίνησής.

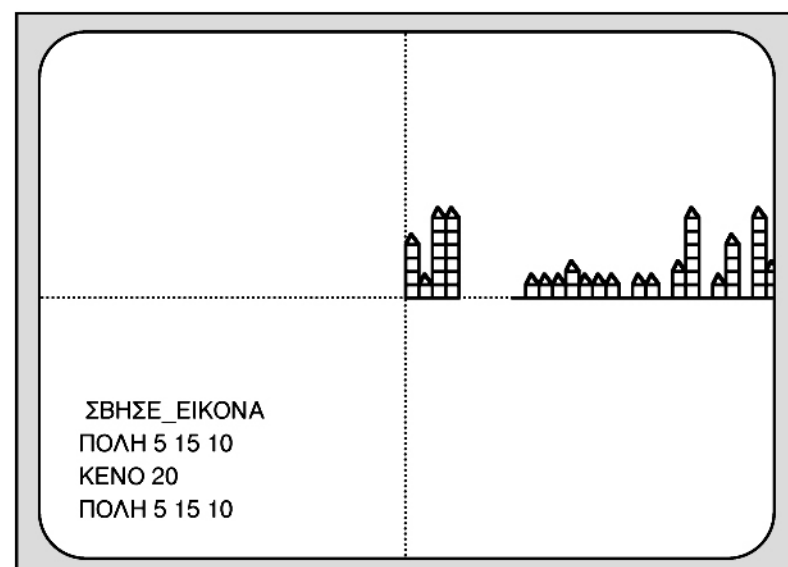
ΤΟ ΚΕΝΟ :ΑPOSTASH

ΜΗ_ΓΡΑΦΕΙΣ

ΚΙΝΗΣΗ_ΔΙΠΛΑ :ΑPOSTASH

ΓΡΑΨΕ

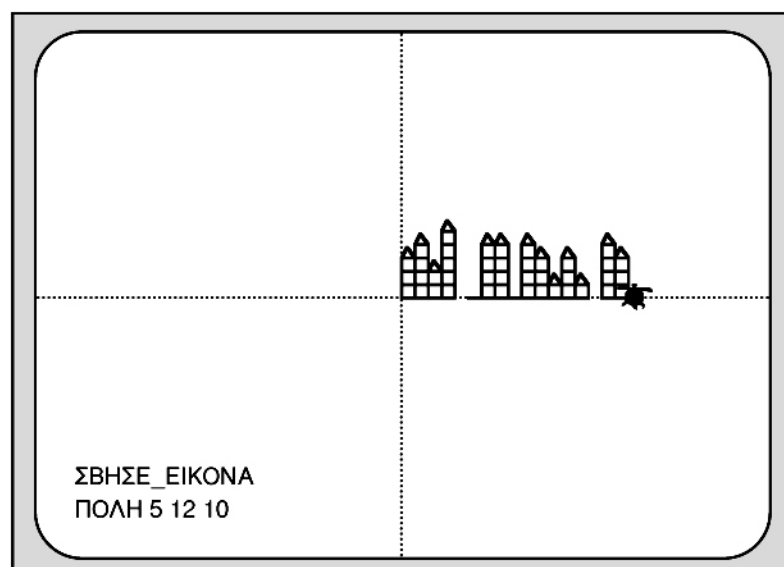
END



Εμπρός για μια καλύτερη πόλη

Στην οθόνη της προηγούμενης σελίδας ξεχωρίσαμε τις δύο πόλεις παρεμβάλλοντας ένα κενό χώρο μεταξύ τους. Το ίδιο μπορούμε να κάνουμε μεταξύ των συνοικιών μιας πόλης. Μπορούμε λοιπόν να ενσωματώσουμε στη διαδικασία ΠΟΛΗ τη διαδικασία ΚΕΝΟ μετά τη διαδικασία ΣΥΝΟΙΚΙΑ έτσι ώστε μετά το τέλος κάθε συνοικίας να αφήνεται λίγος κενός χώρος.

```
ΤΟ ΠΟΛΗ :YPSOS :ORIO_SPITIWN :ORIO_SYNOIKIWN  
MAKE "PLHQOS_SYNOIKIWN RANDOM :ORIO_SYNOIKIWN  
REPEAT :PLHQOS_SYNOIKIWN  
  [MAKE "ORIO_OROF RANDOM 10  
  MAKE "ORIO_OROF :ORIO_OROF + 1  
  ΣΥΝΟΙΚΙΑ :YPSOS :ORIO_SPITIWN :ORIO_OROF  
  ΚΕΝΟ :YPSOS]  
END
```



Μετά το τέταρτο κτίριο αφήνεται ένας κενός χώρος και ακολουθεί μια γραμμή πριν το επόμενο κτίριο. Που διαφέρει (στον τρόπο που κατασκευάστηκε από το πρόγραμμα) η γραμμή από το κενό;

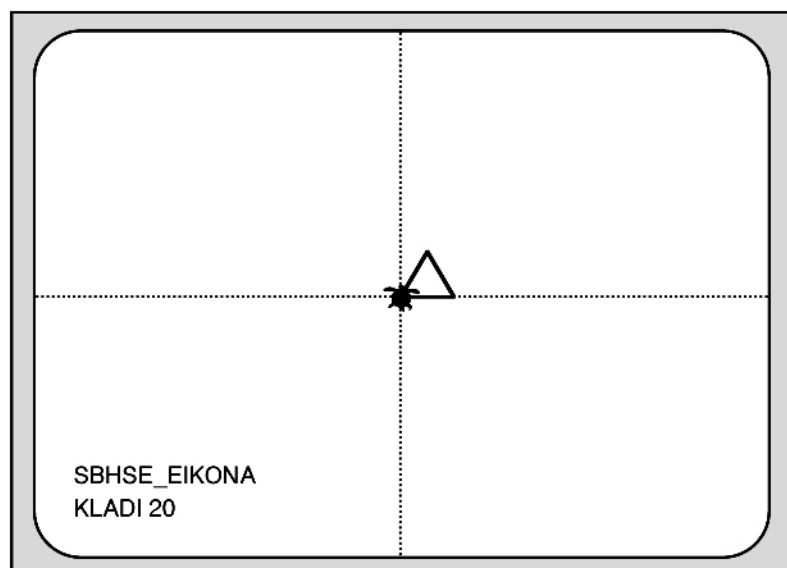
9 Εκδρομή στην Ελατού

Το κλαδί

Ας αρχίσουμε να σχεδιάζουμε ένα έλατο. Το έλατο μπορεί να αναπαρασταθεί από τον κορμό και τα κλαδιά του. Κάθε κλαδί είναι ένα τρίγωνο. Γνωρίζοντας ότι το τρίγωνο είναι ένα πολύγωνο με τρεις πλευρές, μπορούμε να γράψουμε τη διαδικασία KLADI ως εξής:

```
ΤΟ ΚΛΑΔΙ :MEGEQOS  
ΠΟΛΥΓΩΝΟ :MEGEQOS 3  
END
```

Ας διατάξουμε τη χελώνα να σχεδιάσει ένα κλαδί.
Αργότερα, κάτω από το τρίγωνο-κλαδί θα βάλουμε τον κορμό. Αυτός θα πρέπει να βρίσκεται στη μέση της βάσης του τριγώνου. Μας διευκολύνει λοιπόν το τρίγωνο-κλαδί να είναι "κεντραρισμένο". Δηλαδή η χελώνα όταν τελειώνει τη σχεδίαση του κλαδιού να βρίσκεται στη μέση της βάσης του τριγώνου.



Το κλαδί στη σωστή θέση

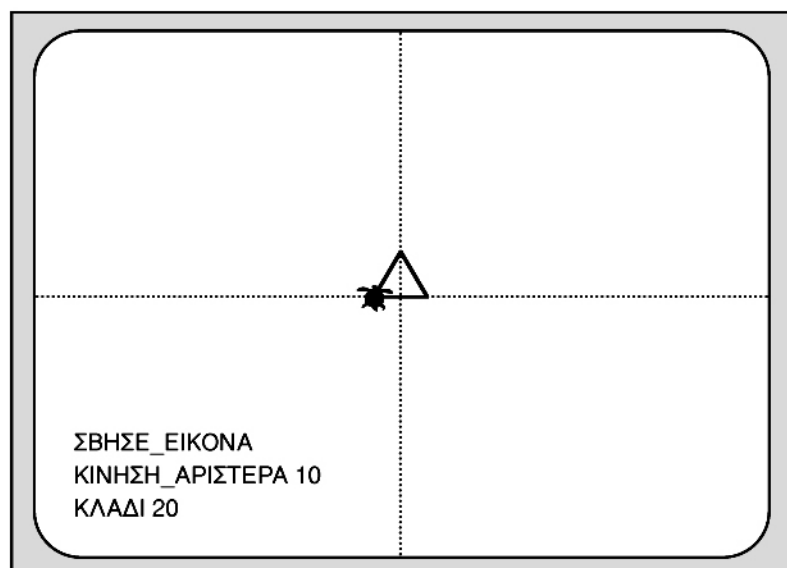
Για ευκολία ας προσπαθήσουμε να μεταφέρουμε το τρίγωνο έτσι ώστε η κορυφή του να περνάει από την ίδια κατακόρυφη η οποία θα περνάει από τη μέση της βάσης του.

Αυτό το πετυχαίνουμε αν πριν ξεκινήσει η σχεδίαση του τριγώνου έχουμε μετατοπίσει (οριζόντια) προς τα αριστερά τη χελώνα σε απόσταση ίση με το μισό της πλευράς του τριγώνου.

Αυτήν την κίνηση της χελώνας προς τα αριστερά θα την κάνουμε με τη διαδικασία ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ.

```
ΤΟ ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ :ΑΠΟΤΑΞΗ  
ΜΗ_ΓΡΑΦΕΙΣ  
ΑΡΙΣΤΕΡΑ 90  
ΕΥΘΕΙΑ :ΑΠΟΤΑΞΗ  
ΔΕΞΙΑ 90  
ΓΡΑΨΕ  
ΕΝΔ
```

Αφού ορίστηκε η διαδικασία ΚΙΝΗΣΗ_ΑΡΙΣΤΕΡΑ ας τη χρησιμοποιήσουμε για να πετύχουμε τη μετατόπιση του κλαδιού προς τα αριστερά. Δεν πρέπει να ξεχάσουμε ότι η μετακίνηση αυτή θα γίνει για απόσταση μισή από το μήκος της πλευράς του τριγώνου-κλαδιού.



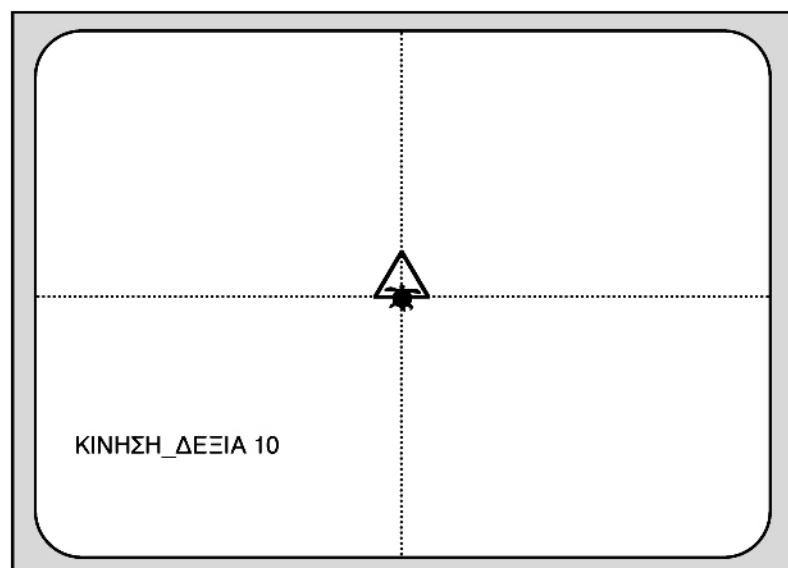
Επιστροφή στην αρχική κατάσταση

Από τα προηγούμενα διαπιστώνουμε ότι ναι μεν το κλαδί-τρίγωνο μετατοπίσθηκε στη θέση που θέλαμε, η χελώνα όμως δε βρίσκεται στο σημείο από το οποίο ξεκίνησε. Θα πρέπει λοιπόν η χελώνα να μετακινηθεί ξανά προς τα δεξιά.

Αυτό γίνεται με τη διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ.

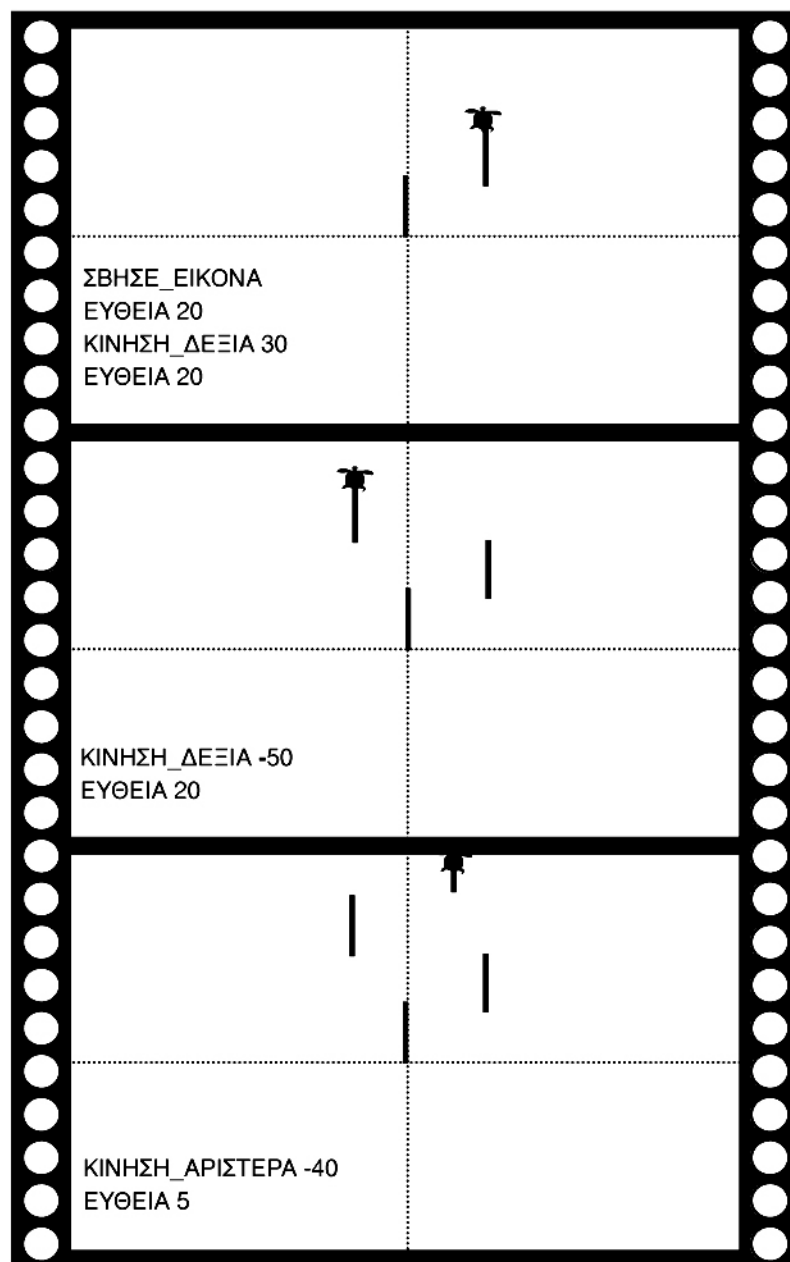
```
ΤΟ ΚΙΝΗΣΗ_ΔΕΞΙΑ :ΑΡΟΤΑΗ  
ΜΗ_ΓΡΑΦΕΙΣ  
ΔΕΞΙΑ 90  
ΕΥΘΕΙΑ :ΑΡΟΤΑΗ  
ΑΡΙΣΤΕΡΑ 90  
ΓΡΑΨΕ  
ΕΝΔ
```

Αφού ορίσθηκε η διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ ως τη χρησιμοποιήσουμε για να πετύχουμε τη μετατόπιση της χελώνας στην αφετηρία. Η μετατόπιση προς την αφετηρία είναι ίση με την αρχική απόσταση που διένυσε η χελώνα όταν μετατοπίσθηκε προς τα αριστερά.



Δεξιά μετακίνηση με αρνητική φορά

Μπορούμε με τη διαδικασία ΚΙΝΗΣΗ_ΔΕΞΙΑ να κινηθούμε προς τα αριστερά αν χρησιμοποιήσουμε αρνητική τιμή.



Οι αριθμοί που μετρούνε ύψος και βάθος, εμπρός και πίσω, δεξιά και αριστερά, ακόμα και όταν είναι ίδιοι, αναφέρονται σε διαφορετικά, αντίθετα μεταξύ τους πράγματα.

Τα μεγέθη δηλαδή μεταβάλλονται συνήθως κατά δύο αντίθετες φορές. Για την έκφραση τέτοιων εννοιών με αριθμούς ή άλλα σύμβολα και όχι με λέξεις, χρησιμοποιούνται τα γνωστά από τις πράξεις της πρόσθεσης και της αφαίρεσης σύμβολα συν (+) και πλην (-).

Το πλην λέγεται αρνητικό πρόσημο και το συν θετικό.

Τα πρόσημα μπαίνουν μπροστά από τους αριθμούς και συμβολίζουν τις λέξεις κάτω, πίσω, αριστερά κλπ. ή πάνω, εμπρός, δεξιά αντίστοιχα.

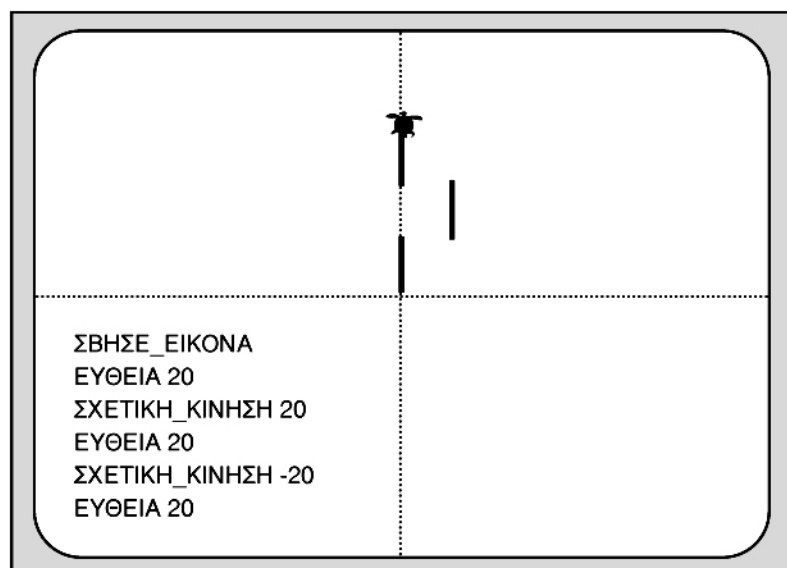
Σχετική οριζόντια μετακίνηση

Διαπιστώνουμε ότι δεν είναι απαραίτητο να έχουμε δύο διαδικασίες για αριστερά και δεξιά μετακίνηση. Αρχεί μια, η ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ, στην οποία αν η τιμή που εισάγεται στη μεταβλητή APOSTASH είναι θετική κινεί τη χελώνα προς τα δεξιά ενώ αν είναι αρνητική τη μετακινεί προς τα αριστερά.

Η διαδικασία αυτή είναι η:

```
ΤΟ ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :ΑPOSTASH  
ΜΗ_ΓΡΑΦΕΙΣ  
ΔΕΞΙΑ 90  
ΕΥΘΕΙΑ :ΑPOSTASH  
ΑΡΙΣΤΕΡΑ 90  
ΓΡΑΨΕ  
ΕΝΔ
```

Ας πειραματιστούμε με τη διαδικασία ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ για να ελέγξουμε τη σωστή λειτουργία της.



Το κλαδί ως δομικό στοιχείο του ελάτου

Έχοντας στη διαθεσή μας τη διαδικασία ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ μπορούμε να βελτιώσουμε τη διαδικασία ΚΛΑΔΙ έτσι ώστε το τρίγωνο που θα σχηματίζει να είναι "κεντραρισμένο".

Η διαδικασία ΚΛΑΔΙ θα πρέπει

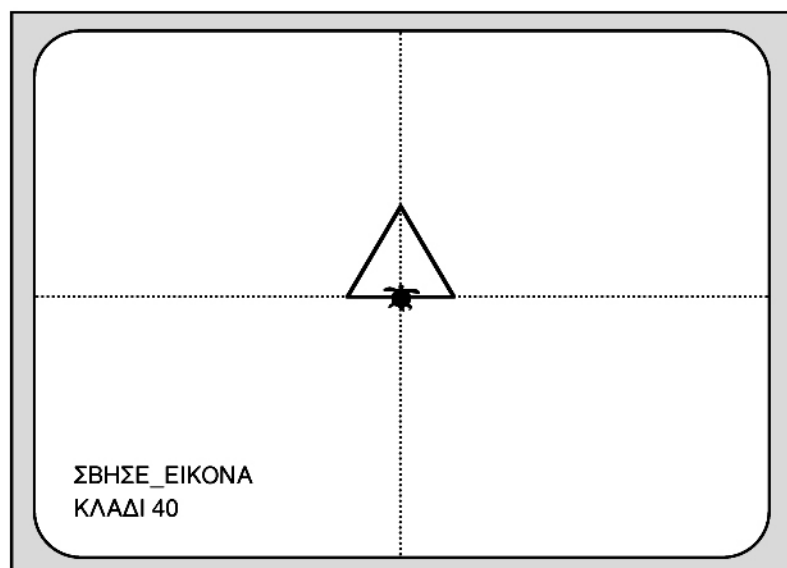
αρχικά να μετατοπίζει τη χελώνα προς τα αριστερά για απόσταση ίση με το μισό της βάσης του τριγώνου να σχηματίζει το κλαδί-τρίγωνο να επαναφέρει τη χελώνα στην αρχική της θέση (στο μέσο της βάσης του τριγώνου) μετακινώντας την προς τα δεξιά.

Φραστική
μέθοδος
περιγραφής του
αλγορίθμου

Ας δούμε πως θα διαμορφωθεί η διαδικασία ΚΛΑΔΙ.

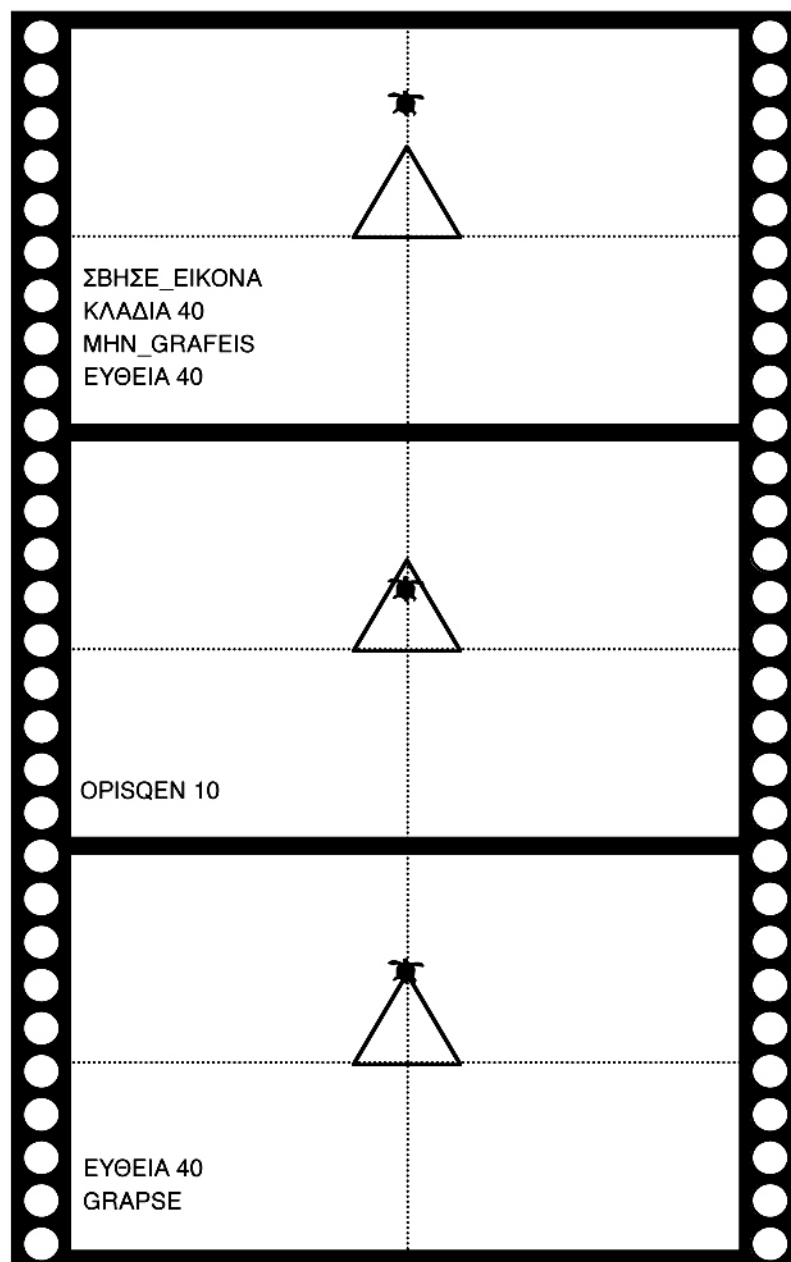
```
ΤΟ ΚΛΑΔΙ :MEGEQOS  
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:MEGEQOS / 2)  
ΠΟΛΥΓΩΝΟ :MEGEQOS 3  
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:MEGEQOS / 2)  
END
```

Η έκφραση **MINUS** (:MEGEQOS/2) αντιστοιχεί στην -:MEGEQOS/2



Από που θα ξεκινήσει το επόμενο κλαδί;

Για να σχεδιάσει το παραπάνω κλαδί η χελώνα θα πρέπει να ανέβει στην επάνω κορυφή του τριγώνου. Πόσο να μετακινηθεί;



Υπολογισμός με το Πυθαγόρειο θεώρημα

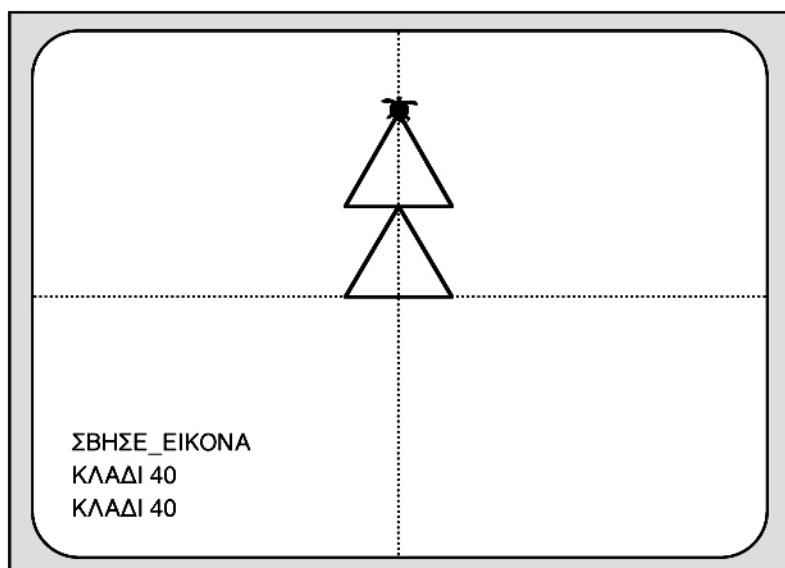
Κάναμε προηγουμένως μερικές προσπάθειες για να βρούμε πόσο διάστημα πρέπει να μετακινήσουμε τη χελώνα από τη βάση ως την κορυφή του κλαδιού. Το αποτέλεσμα όμως που βρήκαμε ισχύει μόνο για το συγκεκριμένο μέγεθος κλαδιού. Δεν είναι δυνατό να βρούμε εμπειρικά τρόπο που να προσδιορίζει το διάστημα αυτό ώστε να μπορούμε να γενικεύσουμε τον υπολογισμό.

Η γενίκευση μπορεί να γίνει αν εφαρμόσουμε το πυθαγόρειο θεώρημα στο ορθογώνιο τρίγωνο που σχηματίζεται από το ύψος του ισοπλεύρου τριγώνου που φέρεται από την επάνω κορυφή. Εφαρμόζοντας το πυθαγόρειο θεώρημα, καταλήγουμε στον τύπο που υψαο = $\frac{\sqrt{3} \times \text{ΜΗΚΟΣ } \Psi \text{ ΕΥΡΑ}}{2}$ γώνου:

Η τετραγωνική ρίζα υπολογίζεται από τη συνάρτηση SQRT.

```

ΤΟ ΚΛΑΔΙ :MEGEQOS
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:MEGEQOS / 2)
ΠΟΛΥΓΩΝΟ :MEGEQOS 3
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:MEGEQOS / 2)
ΜΗ_ΓΡΑΦΕΙΣ
MAKE "ΥPSOS ( (SQRT(3) * :MEGEQOS * ) / 2 )
ΕΥΘΕΙΑ :ΥPSOS
ΓΡΑΨΕ
END
    
```



Αναγκαιότητα για ακριβείς υπολογισμούς και γενίκευση του προγράμματος καθιστά απαραίτητο για το μαθητή να ξέρει να "χειρίζεται" το Πυθαγόρειο θεώρημα. Μια συνάρτηση, για όσους δεν ενδιαφέρονται για τα μαθηματικά, μπορεί να θεωρηθεί ως ένα μαύρο κουτί με δύο πόρτες. Από την πόρτα εισόδου μπαίνουν τα δεδομένα, επεξεργάζονται με τον κατάλληλο τρόπο που υπαγορεύει ο μηχανισμός της συνάρτησης και τα αποτελέσματα βγαίνουν από την πόρτα εξόδου.

Π.χ. η συνάρτηση SQRT παίρνει ως είσοδο το 9 και αφού υπολογίσει την τετραγωνική του ρίζα εξάγει ως αποτέλεσμα το 3.
9-> SQRT -> 3

Μια συνάρτηση είναι ένας νόμος που συνδέει δύο σύνολα X και Y και αντιστοιχίζει σε κάθε στοιχείο x του συνόλου X ένα και μόνο ένα στοιχείο y του συνόλου Y.

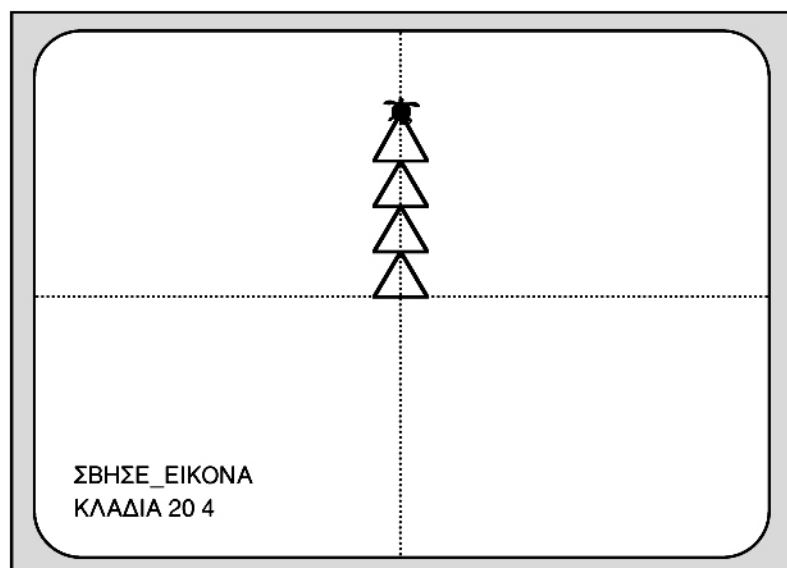
Όλα τα κλαδιά του ελάτου

Αφού μάθαμε τη χελώνα να σχεδιάζει ένα κλαδί και να πηγαίνει στην κορυφή του ώστε να είναι έτοιμη να σχεδιάσει το επόμενο πιο πάνω μπορούμε να επαναλάβουμε αυτήν τη διαδικασία ώστε να σχηματισθούν όλα τα κλαδιά του ελάτου. Αυτό γίνεται με τη διαδικασία ΚΛΑΔΙΑ.

```
TO ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ
REPEAT :ΠΛΗΘΟΣ
  [ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:ΜΕΓΕΘΟΣ / 2)
  ΠΟΛΥΓΩΝΟ :ΜΕΓΕΘΟΣ 3
  ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:ΜΕΓΕΘΟΣ / 2)
  ΜΗ_ΓΡΑΦΕΙΣ
  MAKE "ΥΡΣΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)
  ΕΥΘΕΙΑ :ΥΡΣΟΣ
  ΓΡΑΨΕ]
END
```

Παρατηρούμε ότι το πακέτο των εντολών που βρίσκονται μέσα στο REPEAT είναι το σύνολο των εντολών της διαδικασίας ΚΛΑΔΙ. Άρα μπορούμε να γράψουμε πιο κομψά τη διαδικασία ΚΛΑΔΙΑ χρησιμοποιώντας τη διαδικασία ΚΛΑΔΙ ως εξής:

```
TO ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ
REPEAT :ΠΛΗΘΟΣ
  [ΚΛΑΔΙ :ΜΕΓΕΘΟΣ]
END
```



Ο ΚΟΡΜΟΣ ΤΟΥ ΕΛΑΤΟΥ

Εκτός από τα κλαδιά για να σχεδιάσουμε το έλατο χρειάζεται και ο κορμός του που θα τον φτιάξουμε με ένα τετράγωνο.

Εκείνο που έχει σημασία είναι η θέση του τετραγώνου σε σχέση με τη θέση της χελώνας για να ταιριάζει ο κορμός με τα κλαδιά.

Θα εκμεταλλευθούμε την εμπειρία που αποκτήσαμε σχεδιάζοντας το κλαδί-τρίγωνο και θα τη χρησιμοποιήσουμε για να τοποθετήσουμε σωστά τον κορμό.

ΤΟ ΚΟΡΜΟΣ :Y

ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:Y / 2)

ΠΟΛΥΓΩΝΟ :Y 4

ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:Y / 2)

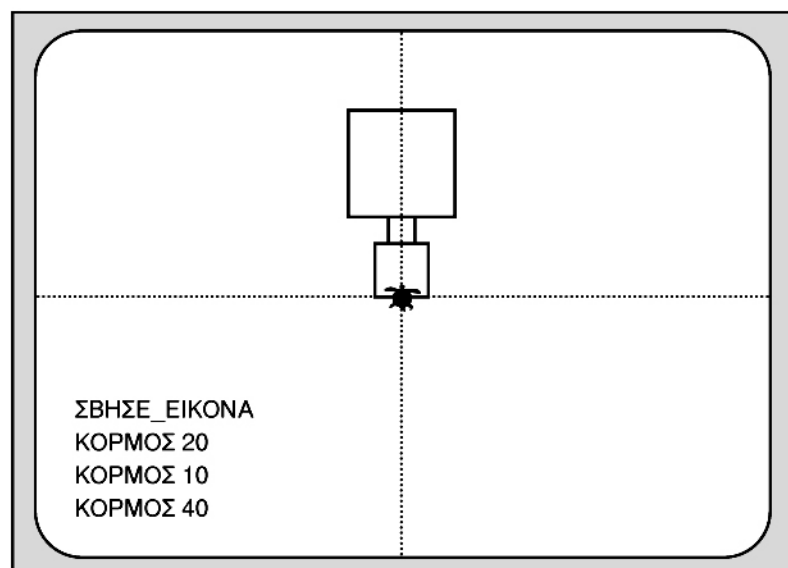
ΜΗ_ΓΡΑΦΕΙΣ

ΕΥΘΕΙΑ :Y

ΓΡΑΨΕ

END

Συγκρίνετε τις διαδικασίες ΚΟΡΜΟΣ και ΚΛΑΔΙ, αναφέρεται τις ομοιότητες και τις διαφορές. Που οφείλονται;



Το έλατο!

Διαθέτοντας τις διαδικασίες ΚΟΡΜΟΣ και ΚΛΑΔΙΑ η σχεδίαση του ελάτου είναι πλέον απλή υπόθεση.

Θα πρέπει μόνο να λάβουμε υπόψη μας ότι το μέγεθος του κορμού πρέπει να αναλογεί με τις διαστάσεις των κλαδιών. Κάνοντας μερικές δοκιμές καταλήγουμε ότι το έλατο θα φαίνεται φυσικότερο αν το μέγεθος του κορμού του είναι το $1/3$ του μεγέθους των πλευρών των κλαδιών-τριγώνων.

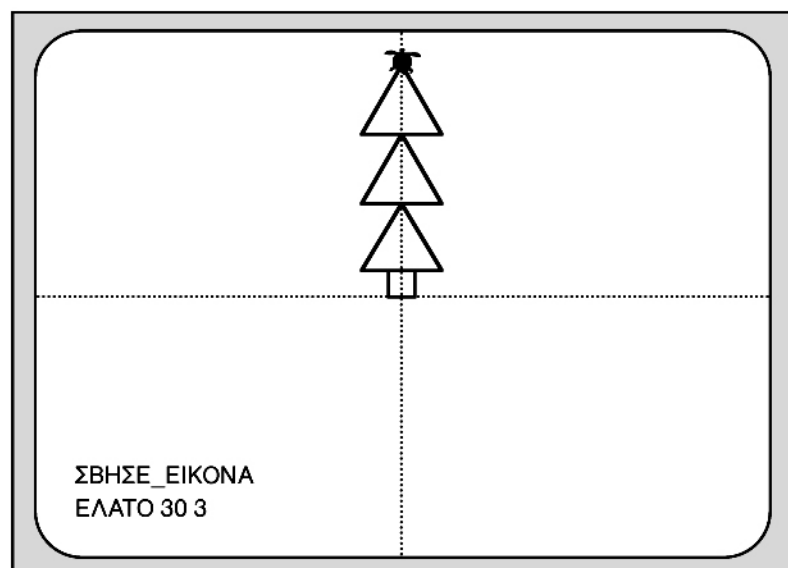
Ας δούμε τη διαδικασία ΕΛΑΤΟ:

ΤΟ ΕΛΑΤΟ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ

ΚΟΡΜΟΣ (:ΜΕΓΕΘΟΣ / 3)

ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ

END



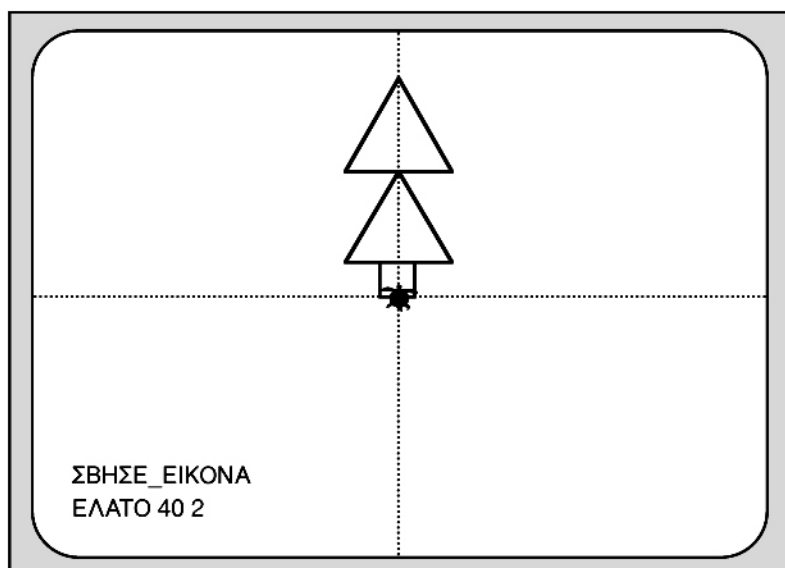
Επιστροφή στις ρίζες

Η διαδικασία ΕΛΑΤΟ όταν ολοκληρώσει τη σχεδίαση, αφήνει τη χελώνα στην κορυφή του. Είναι καλύτερα η χελώνα να επιστρέφει στη θέση από όπου ξεκίνησε.

Αυτό γίνεται βάζοντας τη χελώνα να οπισθοχωρήσει για απόσταση όση είναι το άθροισμα των υψών των κλαδιών συν το ύψος του κορμού.

Έτσι η διαδικασία ΕΛΑΤΟ βελτιώνεται ως εξής:

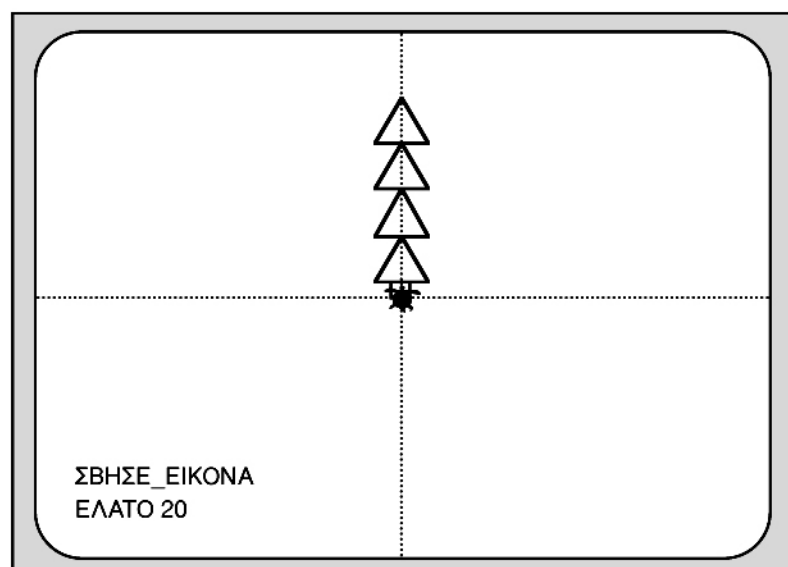
```
ΤΟ ΕΛΑΤΟ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ  
ΚΟΡΜΟΣ (:ΜΕΓΕΘΟΣ / 3)  
ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :ΠΛΗΘΟΣ  
ΜΗ_ΓΡΑΦΕΙΣ  
MAKE "ΥΨΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)  
REPEAT :ΠΛΗΘΟΣ  
  [ΟΠΙΣΘΕΝ :ΥΨΟΣ]  
ΟΠΙΣΘΕΝ (:ΜΕΓΕΘΟΣ / 3)  
ΓΡΑΨΕ  
ΕΝΔ
```



Η ηλικία του ελάτου

Το πλήθος των κλαδιών του ελάτου μας δείχνει την ηλικία του δένδρου. Θα χρησιμοποιήσουμε τη συνάρτηση RANDOM ώστε να έχουμε κάθε φορά τυχαίο πλήθος κλαδιών.

```
TO ΕΛΑΤΟ :ΜΕΓΕΘΟΣ
MAKE "PLHQOS RANDOM 5
ΚΟΡΜΟΣ (:ΜΕΓΕΘΟΣ / 3)
ΚΛΑΔΙΑ :ΜΕΓΕΘΟΣ :PLHQOS
ΜΗ_ΓΡΑΦΕΙΣ
MAKE "ΥΨΟΣ (:ΜΕΓΕΘΟΣ * (SQRT(3)) / 2)
REPEAT :PLHQOS
  [ΟΠΙΣΘΕΝ :ΥΨΟΣ]
ΟΠΙΣΘΕΝ (:ΜΕΓΕΘΟΣ / 3)
ΓΡΑΨΕ
END
```

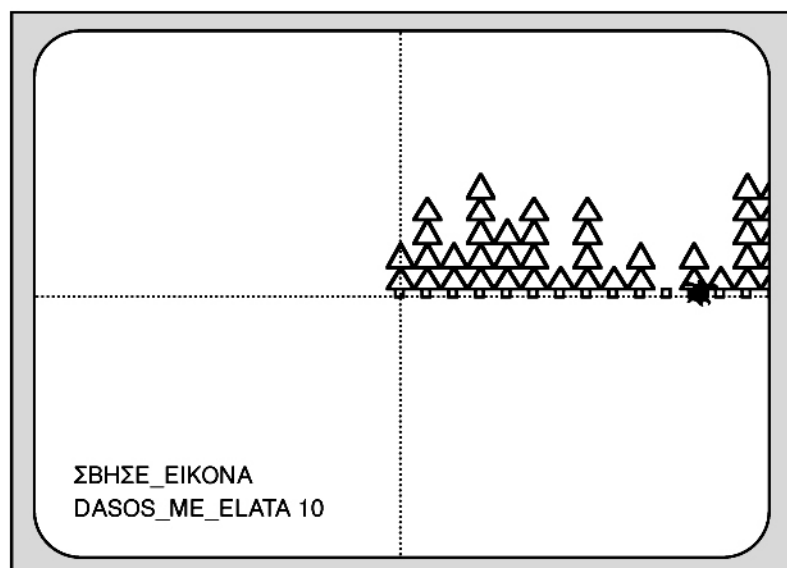


Ένα δάσος με έλατα

Πολύ εύκολα πλέον μπορούμε επαναλαμβάνοντας μερικές φορές τη διαδικασία ΕΛΑΤΟ να φτιάξουμε ένα δάσος με έλατα.

```
ΤΟ ΔΑΣΟΣ :MEGEQOS
MAKE "ΕΛΑΤΑ RANDOM 15
REPEAT :ΕΛΑΤΑ
  [ΕΛΑΤΟ :MEGEQOS
   ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :MEGEQOS]
END
```

Σε τι χρειάζεται
η διαδικασία
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ;
Γιατί βρίσκεται
μέσα στις
αγκύλες της
εντολής
REPEAT;



Τι συνέβη στο
11ο έλατο;
Ποιος το έκοψε;
Πως έγινε αυτό;

Το έλατο που δείχνει την ηλικία του

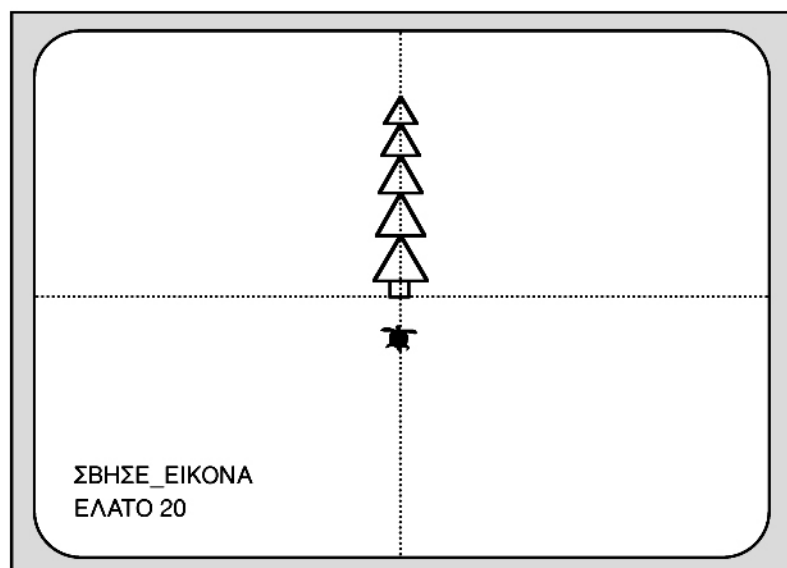
Τα κλαδιά που βρίσκονται προς τη βάση του ελάτου είναι τα πιο παλιά και συνεπώς τα πιο μεγάλα ενώ όσο ανεβαίνουμε προς την κορυφή τα κλαδιά να γίνονται μικρότερα.

Θα πρέπει λοιπόν το μέγεθος κάθε επόμενου κλαδιού να είναι μικρότερο και έστω ότι αυτή η μείωση είναι 2 βήματα χελώνας.

Η διαδικασία ΚΛΑΔΙΑ θα τροποποιηθεί ως εξής:

```
TO ΚΛΑΔΙΑ :MEGEQOS :PLHQOS
REPEAT :PLHQOS
  [ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ MINUS (:MEGEQOS / 2)
  ΠΟΛΥΓΩΝΟ :MEGEQOS 3
  ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ (:MEGEQOS / 2)
  ΜΗ_ΓΡΑΦΕΙΣ
  MAKE "ΥPSOS (:MEGEQOS * (SQRT(3)) / 2)
  ΕΥΘΕΙΑ :ΥPSOS
  ΓΡΑΨΕ
  MAKE "MEGEQOS :MEGEQOS - 2]
END
```

Τι θα συμβεί αν με τη μείωση αυτή η τιμή της μεταβλητής MEGEQOS γίνει αρνητική; Προτείνετε εναλλακτικούς τρόπους για να γίνεται η μείωση ώστε να μην υπάρξει περίπτωση αρνητικής τιμής για τη μεταβλητή MEGEQOS.



Γιατί η χελώνα
δεν σταμάτησε
στο έδαφος
αλλά προχώρησε
στις ρίζες;

Η χελώνα έπαψε να κυκλοφορεί υπόγεια

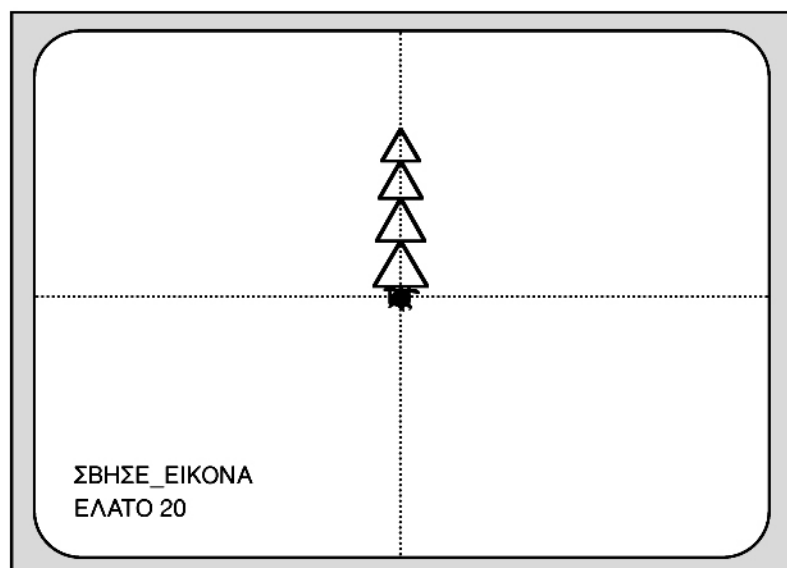
Η χελώνα προηγουμένως οπισθοχώρησε περισσότερο από όσο είχε προχωρήσει προς τα πάνω. Αυτό έγινε γιατί κατά την οπισθοχώρηση δε λάβαμε υπόψη ότι τα μεγέθη των κλαδιών μεταβάλλονταν.

Αν λοιπόν κατά την υποχώρηση μειώνουμε το μέγεθος όσο το αυξάναμε, τότε ισοφαρίζουμε τις δύο αποστάσεις.

```
ΤΟ ΕΛΑΤΟ :MEGEQOS
MAKE "PLHQOS RANDOM 6
ΚΟΡΜΟΣ (:MEGEQOS / 3)
MAKE "M1 :MEGEQOS
MAKE "M2 :MEGEQOS
ΚΛΑΔΙΑ :MEGEQOS :PLHQOS

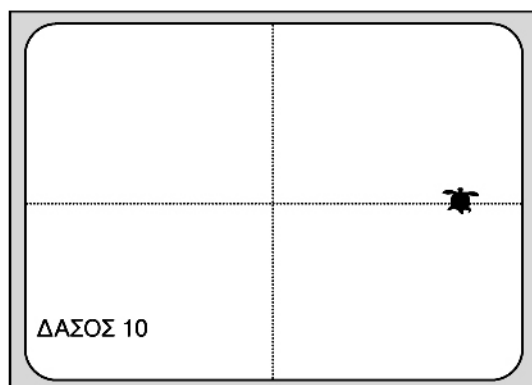
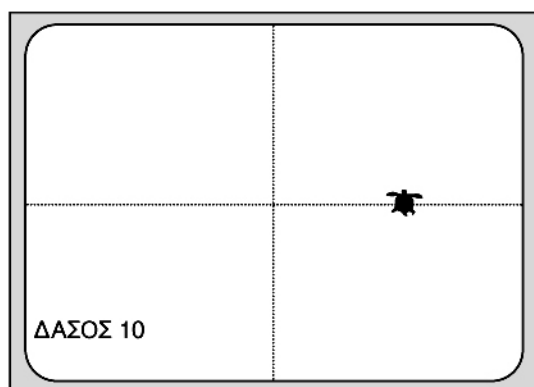
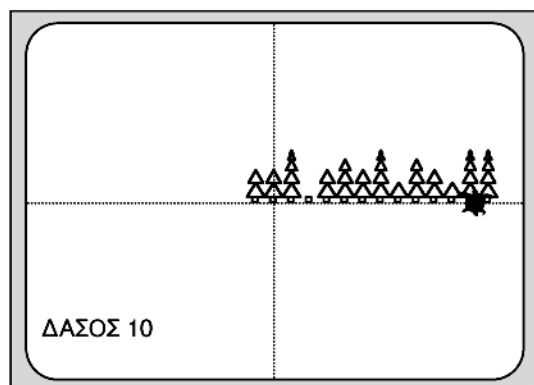
ΜΗ_ΓΡΑΦΕΙΣ
MAKE "MEGEQOS (:MEGEQOS + 2)
REPEAT :PLHQOS
  [MAKE "YPSOS (:M1 * (SQRT(3)) / 2)
   ΟΠΙΣΘΕΝ :YPSOS
   MAKE "M1 (:M1 - 2)]
ΟΠΙΣΘΕΝ (:M2 / 3)
ΓΡΑΨΕ
ΕΝΔ
```

Ποιά
σκοπιμότητα
εξυπηρετούν οι
εντολές της 4ης
και της 5ης
γραμμής της
διαδικασίας
ΕΛΑΤΟ;



Εκδρομή στην Ελατού*

Ας βάλουμε τη χελώνα να σχεδιάσει μερικά δάση με έλατα.



* Ελατού, χωριό της ορεινής Ναυπακτίας που δικαιολογεί πλήρως το όνομά του γιατί είναι κατάφυτο με έλατα.

10 Το μηδέν και το άπειρο

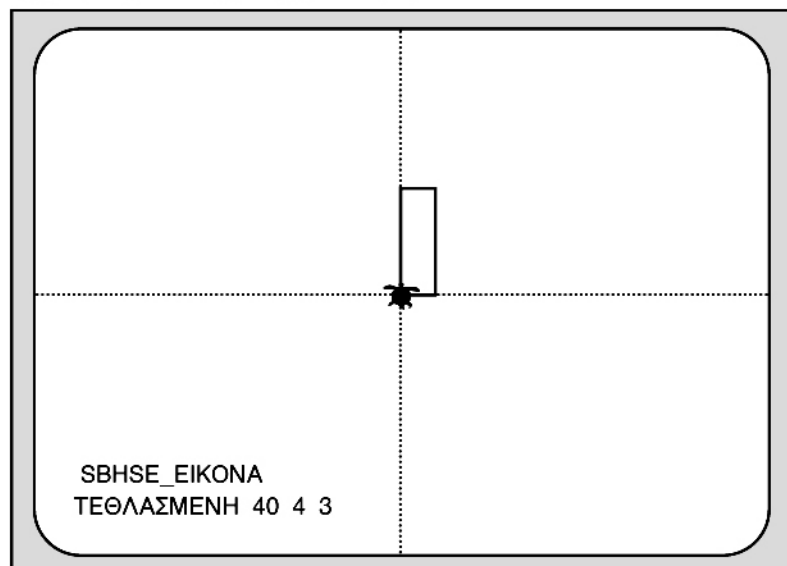
Τεθλασμένες γραμμές

Αναλύοντας ένα ορθογώνιο παραλληλόγραμμο μπορούμε να πούμε ότι αποτελείται από δύο ίδια κομμάτια. Το καθένα αποτελείται από ένα ευθύγραμμο τμήμα, μια στροφή 90 μοιρών και ένα μικρότερο ευθύγραμμο τμήμα.

Η δομή του σχήματος είναι ανάλογη με αυτήν του πολυγώνου και εκμεταλλευόμενοι την εμπειρία από τα πολύγωνα γράφουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ η οποία για την τιμή 4 στη μεταβλητή :PLHQOS, έχει ως αποτέλεσμα ένα ορθογώνιο παραλληλόγραμμο.

```
TO ΤΕΘΛΑΣΜΕΝΗ :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / 2)
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA
    ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS]
ΔΕΞΙΑ 90
END
```

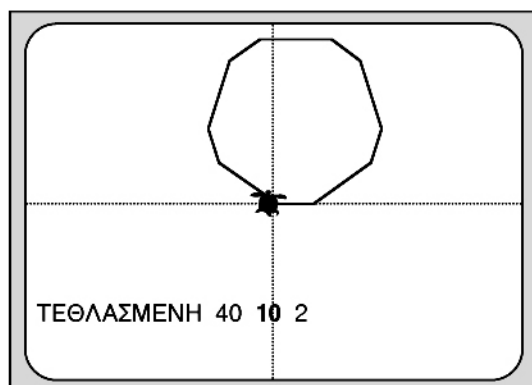
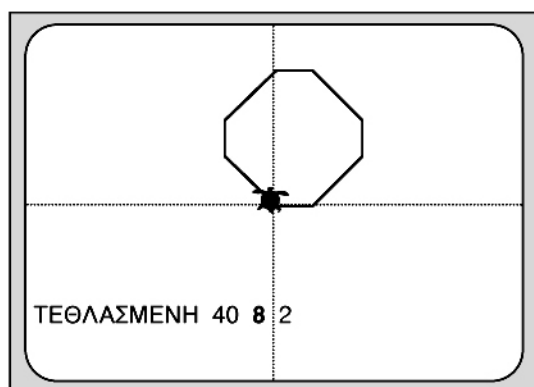
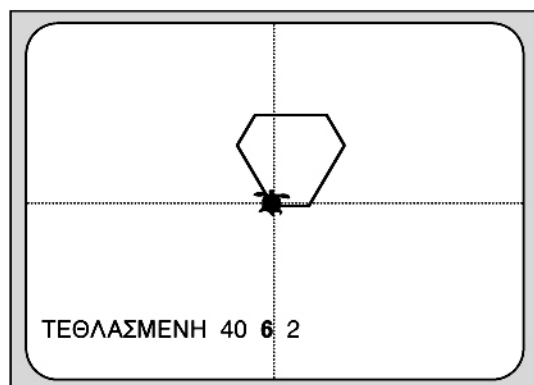
Η μεταβλητή LOGOS προσδιορίζει την αναλογία μεταξύ του μήκους της μεγάλης και της μικρής πλευράς.



«Για να διευκολυνθούμε, κατατάσσω τους συλλογισμούς με λόγους σε δύο επίπεδα ή στάδια:
(1) στο επίπεδο όπου ερμηνεύουμε περιφραστικά το αποτέλεσμα της διαίρεσης δύο αριθμών
(2) στο επίπεδο όπου χρησιμοποιούμε την προηγούμενη ερμηνεία για να υπολογίσουμε κάποια άλλη ποσότητα».
A. Arons

Κλειστές τεθλασμένες γραμμές

Στη διαδικασία ΤΕΘΛΑΣΜΕΝΗ διατηρούμε σταθερή την τιμή της μεταβλητής LOGOS και μεταβάλλουμε την τιμή της μεταβλητής ΡΛΗQΟΣ.



Ορθογώνιο παραλληλόγραμμο

Επειδή το ορθογώνιο παραλληλόγραμμο είναι ένα σχήμα που χρησιμοποιείται συχνά, το ορίζουμε ξεχωριστά.

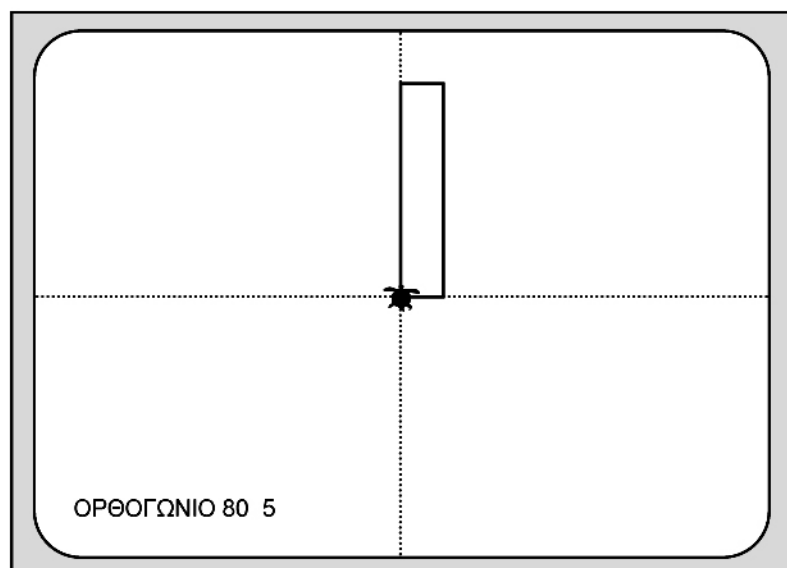
Για τον ορισμό μπορούμε ή να χρησιμοποιήσουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ

```
ΤΟ ΟΡΘΟΓΩΝΙΟ :PLEYRA :LOGOS  
ΤΕΘΛΑΣΜΕΝΗ :PLEYRA 4 :LOGOS  
END
```

ή να ορίσουμε αναλυτικά ξανά όλα τα βήματα.

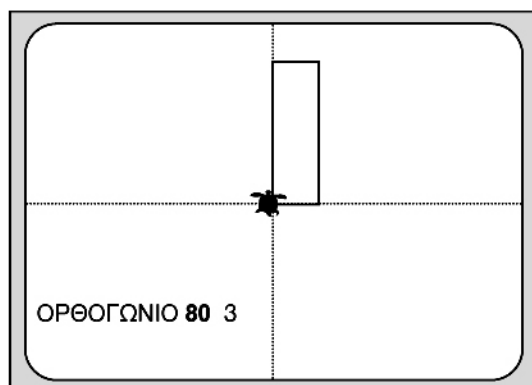
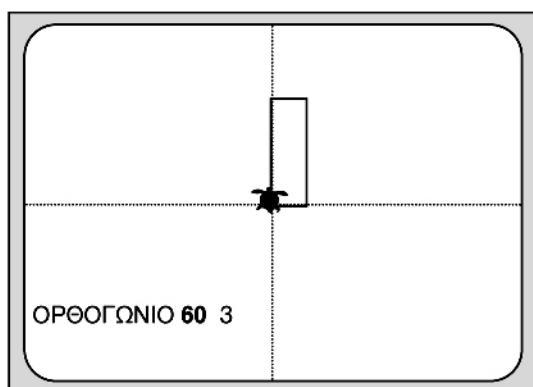
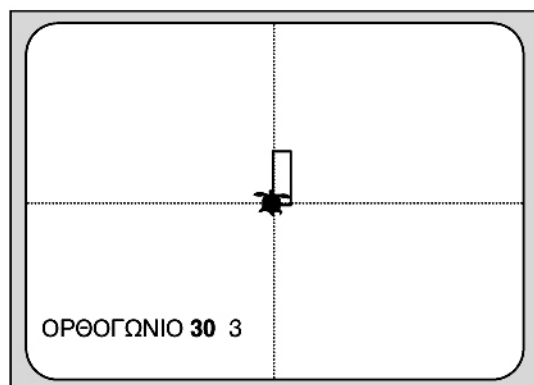
```
ΤΟ ΟΡΘΟΓΩΝΙΟ :PLEYRA :LOGOS  
ΑΡΙΣΤΕΡΑ 90  
MAKE "MOIRES (360 / 4)  
REPEAT 2  
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA  
    ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS ]  
ΔΕΞΙΑ 90  
END
```

Με τη δεύτερη λύση η διαδικασία ΟΡΘΟΓΩΝΙΟ γίνεται ανεξάρτητη από τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ.



Σταθερή αναλογία πλευρών

Ας σχεδιάσουμε μερικά ορθογώνια παραλληλόγραμμα με διαφορετικά μεγέθη αλλά με σταθερό λόγο μεταξύ των πλευρών τους.

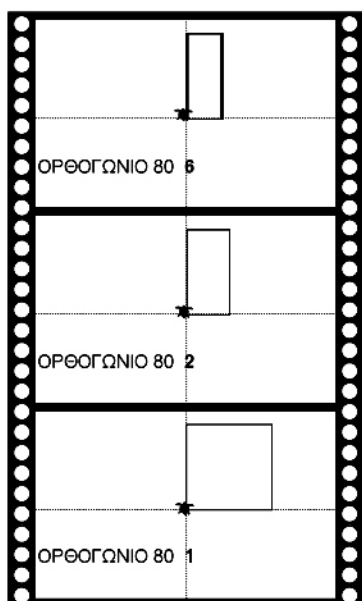


«...μελετάμε το
λόγο δύο ίδιων
φυσικών
μεγεθών...

Στη
συγκεκριμένη
περίπτωση η
αριθμητική τιμή
του λόγου
χρησιμεύει για
σύγκριση:
δείχνει πόσες
φορές το ένα
μήκος είναι
μεγαλύτερο (ή
μικρότερο) από
το άλλο».
Α. Arons

Διάφορες τιμές του λόγου των πλευρών

Ας σχεδιάσουμε ορθογώνια παραλληλόγραμμα με σταθερό μέγεθος πλευράς αλλάζοντας όμως το λόγο μεταξύ των πλευρών τους.

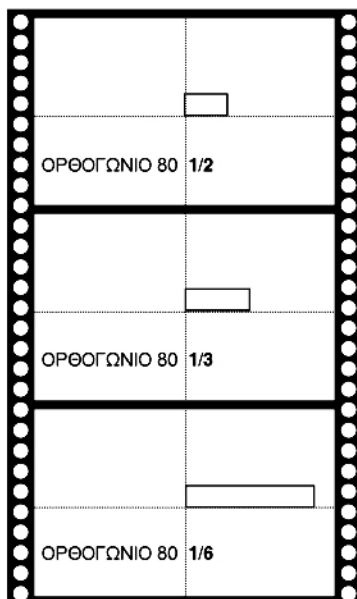


Λόγος 1 σημαίνει ισότητα

Το ορθογώνιο γίνεται τετράγωνο.

«...μελετάμε το λόγο δύο ίδιων φυσικών μεγεθών...
Στη συγκεκριμένη περίπτωση η αριθμητική τιμή του λόγου χρησιμεύει για σύγκριση: δείχνει πόσες φορές το ένα μήκος είναι μεγαλύτερο (ή μικρότερο) από το άλλο».
A. Arons

Διαίρεση δια (1/2) σημαίνει πολλαπλασιασμός επί 2.

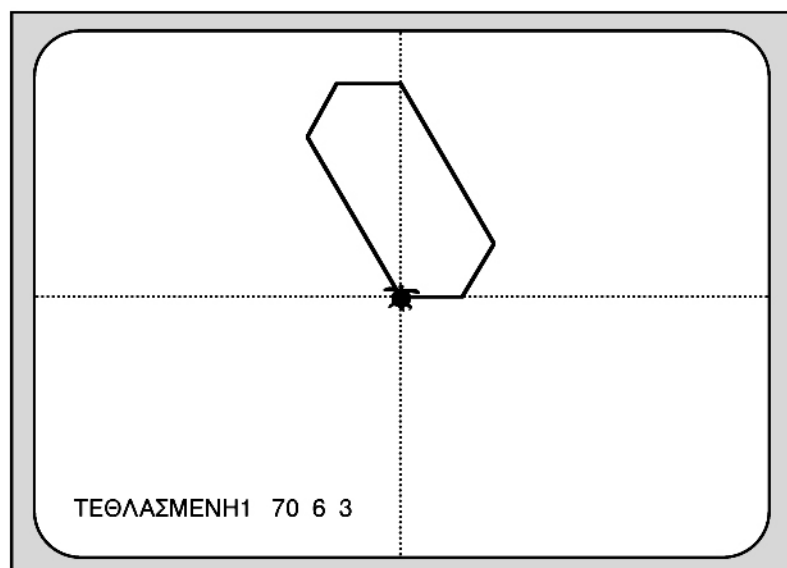


Μια πιο εξελιγμένη τεθλασμένη

Μπορούμε να εξελίξουμε τη διαδικασία ΤΕΘΛΑΣΜΕΝΗ έτσι ώστε να είναι τρεις οι πλευρές που θα επαναλαμβάνονται.

Ας δούμε πως θα είναι μια τέτοια διαδικασία.

```
ΤΟ ΤΕΘΛΑΣΜΕΝΗ1 :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "ΜΟΙΡΕΣ (360 / :PLHQOS)
REPEAT (:PLHQOS / 3)
  [ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA
    ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS
    ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS ]
ΔΕΞΙΑ 90
END
```



Πλάγιο παραλληλόγραμμο

Το άθροισμα των γωνιών κάθε κανονικού πολυγώνου είναι 360° .

Στο ορθογώνιο παραλληλόγραμμο όλες οι γωνίες είναι ορθές και κατά συνέπεια ίσες μεταξύ τους.

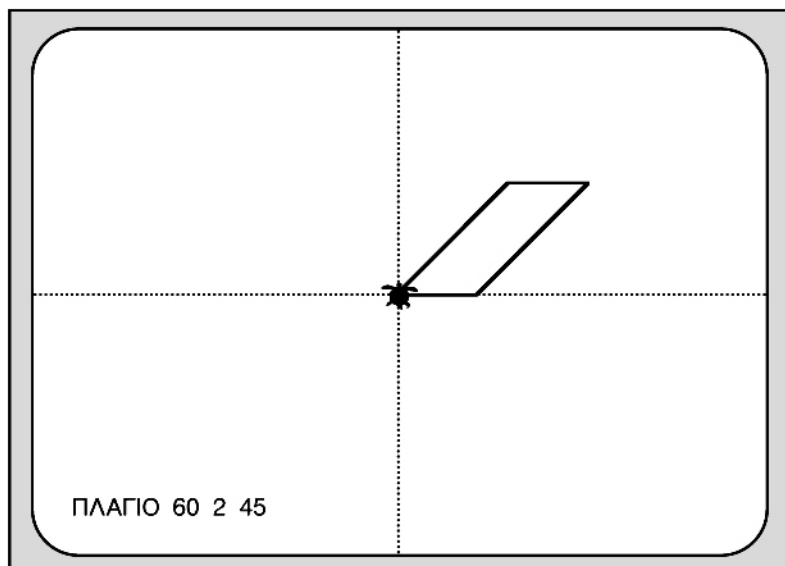
Τι θα συνέβαινε άραγε αν αυξάναμε δύο απέναντι γωνίες ενός ορθογώνιου παραλληλογράμμου και ταυτόχρονα μειώναμε τις άλλες δύο κατά ίση ποσότητα; Αυτό μπορεί να γίνει ως εξής:

Η γωνία του ορθογώνιου παραλληλογράμμου προκύπτει από τη διαίρεση $360/4$. Ας ονομάσουμε τις (απέναντι) μεγάλες γωνίες MOIRES_A και τις (απέναντι) μικρές γωνίες MOIRES_B.

Στις μεγάλες γωνίες προσθέτουμε την ποσότητα D δηλαδή έχουν τιμή $(360/4) + D$ ενώ στις μικρές την αφαιρούμε $(360/4) - D$.

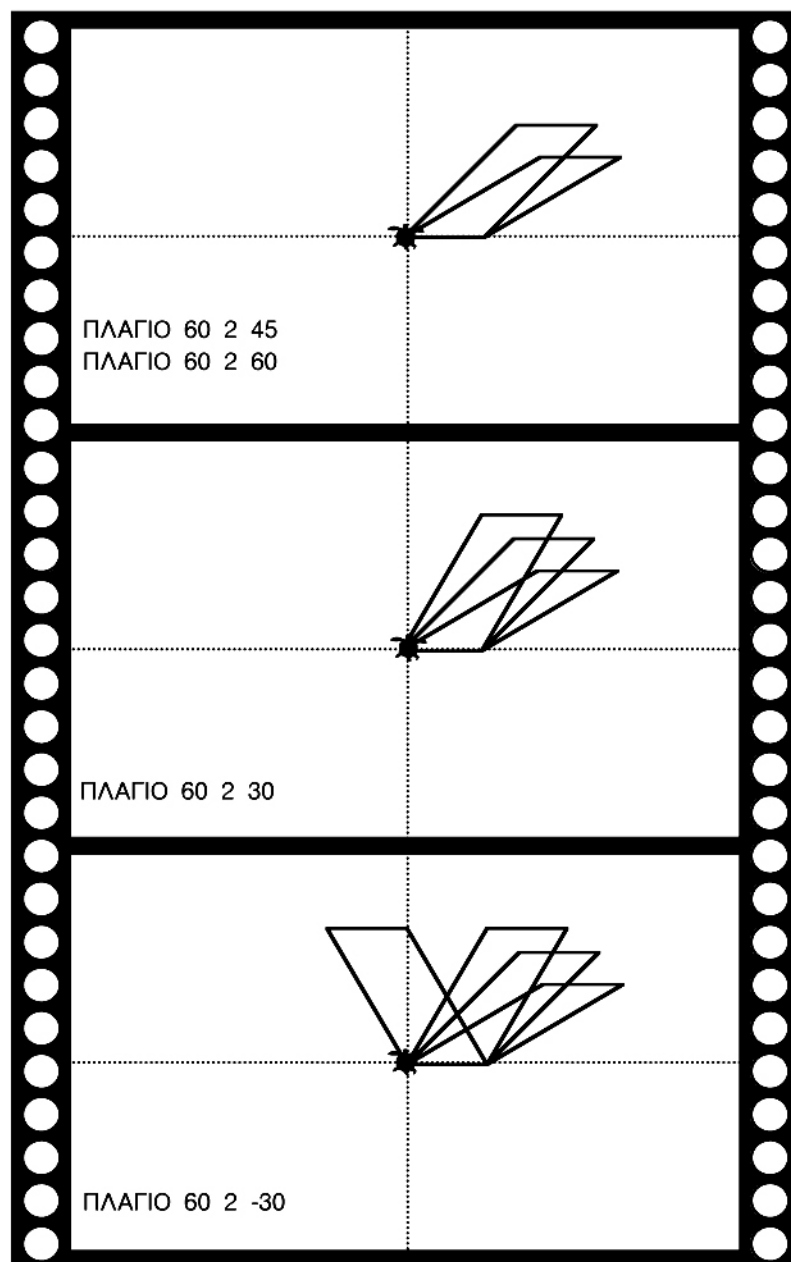
Ας δούμε πως διαμορφώνεται μια τέτοια διαδικασία, η PLAG_PARALL/MMO:

```
TO ΠΛΑΓΙΟ :PLEYRA :LOGOS :D
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRESA (360 / 4) + :D
MAKE "MOIRESEB (360 / 4) - :D
REPEAT 2
  [ ΓΡΑΜΜΗ :MOIRESA :PLEYRA
    ΓΡΑΜΜΗ :MOIRESEB :PLEYRA / :LOGOS]
ΔΕΞΙΑ 90
END
```



Πλάγια παραλληλόγραμμα

Για να καταλάβουμε καλύτερα το μηχανισμό της διαδικασίας ΠΛΑΓΙΟ ας τη βάλουμε να «τρέξει» μερικές φορές.



Η αίσθηση ότι η επιφάνεια πέφτει

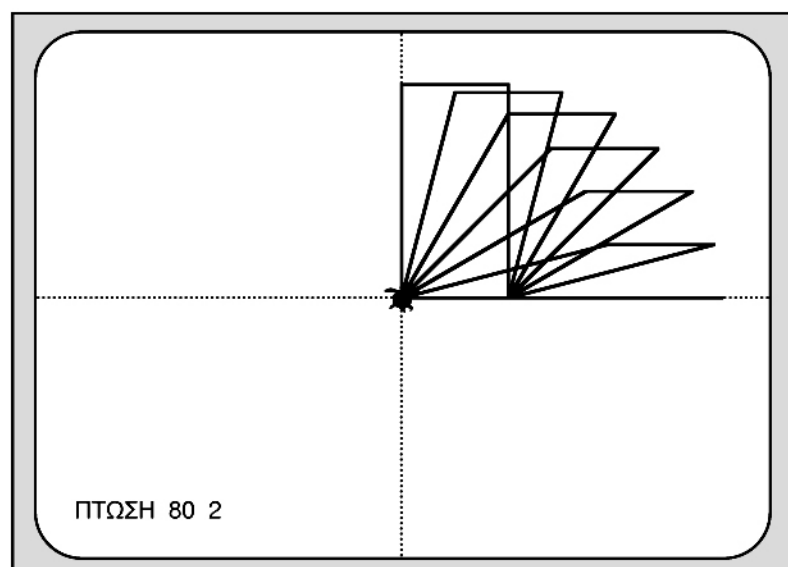
Οι εικόνες της προηγούμενης σελίδας δημιουργούν την αίσθηση ότι το παραλληλόγραμμο πέφτει.

Η αίσθηση δημιουργείται από τα διαφορετικά πλάγια παραλληλόγραμμα που σχεδιάζονται καθώς αυξάνει η τιμή της γωνίας D . Θα εκμεταλλευτούμε αυτήν την παρατήρηση για να φτιάξουμε μια διαδικασία που να σχεδιάζει διαδοχικές φάσεις της πτώσης του παραλληλογράμμου. Στην αρχή η D θα είναι μηδέν και στη συνέχεια θα αυξάνει κατά ένα σταθερό βήμα.

Ας δούμε τη διαδικασία PTWSH:

```
ΤΟ ΠΤΩΣΗ :PLEYRA :LOGOS  
MAKE "D 0  
REPEAT 7  
  [ ΠΛΑΓΙΟ :PLEYRA :LOGOS :D  
    MAKE "D (:D + 15) ]  
END
```

Παρατηρείστε ότι όταν η D είναι 0 σχηματίζεται ορθογώνιο παραλληλόγραμμο, ενώ και όταν γίνει 90° το πλάγιο παραλληλόγραμμο εκφυλίζεται σε μια (οριζόντια) γραμμή.

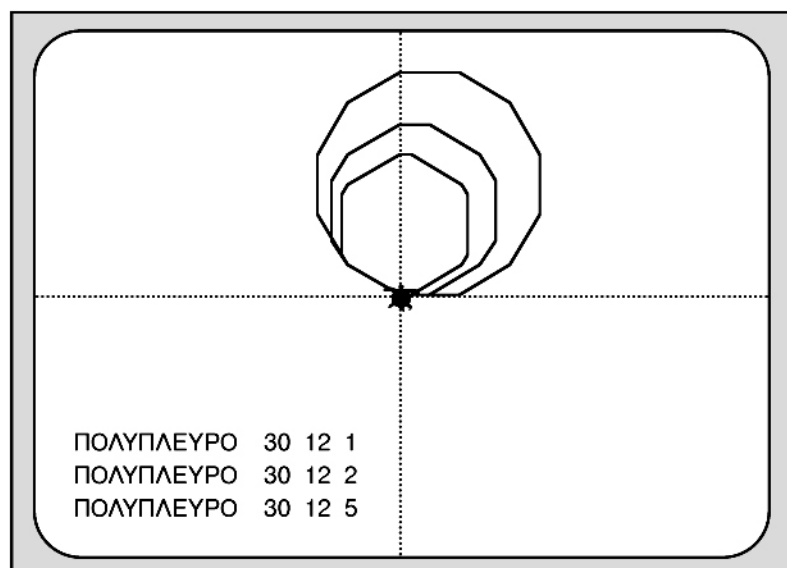


Ένα πολύπλευρο

Συνδυάζοντας την εμπειρία από τις διαδικασίες ΤΕΘΛΑΣΜΕΝΗ και ΠΟΛΥΓΩΝΟ μπορούμε να φτιάξουμε τη διαδικασία ΠΟΛΥΠΛΕΥΡΟ. Το πολύπλευρο είναι ένα σχήμα με σταθερό λόγο των μηκών δύο διαδοχικών πλευρών. Αυτός ο συνδυασμός των δύο πλευρών επαναλαμβάνεται μέχρι να σχηματισθεί μια κλειστή τεθλασμένη γραμμή. Στη διαδικασία θα πρέπει να δίνονται ως παράμετροι το πλήθος των πλευρών, το μέγεθος της μιας βασικής πλευράς και ο λόγος της δευτερεύουσας πλευράς ως προς τη βασική. Η γωνία μεταξύ των πλευρών θα υπολογίζεται ως το πηλίκο των 360 μοιρών δια το πλήθος των πλευρών.

Έτσι η διαδικασία ΠΟΛΥΠΛΕΥΡΟ θα είναι:

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :LOGOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / 2)
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA
    ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS ]
ΔΕΞΙΑ 90
END
```



Φωλιασμένες επαναλήψεις

Στον κώδικα της προηγούμενης διαδικασίας υπάρχουν δύο σχεδόν ίδιες εντολές που σχεδιάζουν την κύρια και τη δευτερεύουσα πλευρά του πολύπλευρου, οι

```
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA
```

```
ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA/:LOGOS
```

Μπορούμε και τις δύο να τις αντικαταστήσουμε με μια εντολή που επαναλαμβάνεται δυο φορές. Αυτό γίνεται αν χρησιμοποιήσουμε τη μεταβλητή ΒΗΜΑ_ΛΟΓΟΥ που θα μεταβάλλει κάθε φορά το λόγο της δευτερεύουσας πλευράς ως προς την κύρια:

```
REPEAT 2
```

```
_ [ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA/:LOGOS
```

```
_ MAKE "LOGOS :LOGOS*:ΒΗΜΑ_ΛΟΓΟΥ ]
```

Με αυτές τις τροποποιήσεις η διαδικασία ΠΟΛΥΠΛΕΥΡΟ γίνεται:

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :ΒΗΜΑ_ΛΟΓΟΥ
```

```
ΑΡΙΣΤΕΡΑ 90
```

```
MAKE "ΜΟΙΡΕΣ (360 / :PLHQOS)
```

```
REPEAT (:PLHQOS / 2)
```

```
  [ MAKE "LOGOS 1
```

```
    REPEAT 2
```

```
      [ ΓΡΑΜΜΗ :ΜΟΙΡΕΣ :PLEYRA / :LOGOS
```

```
        MAKE "LOGOS :LOGOS * :ΒΗΜΑ_ΛΟΓΟΥ
```

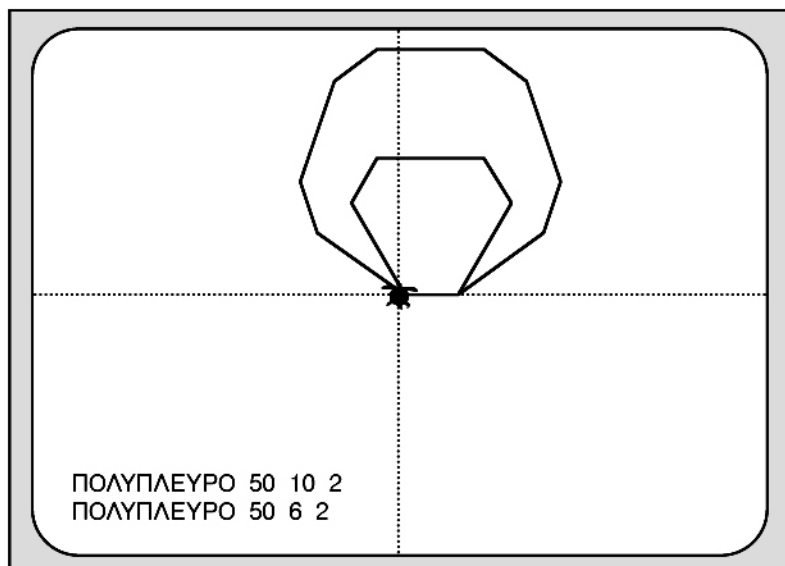
```
      ]
```

```
    ]
```

```
ΔΕΞΙΑ 90
```

```
END
```

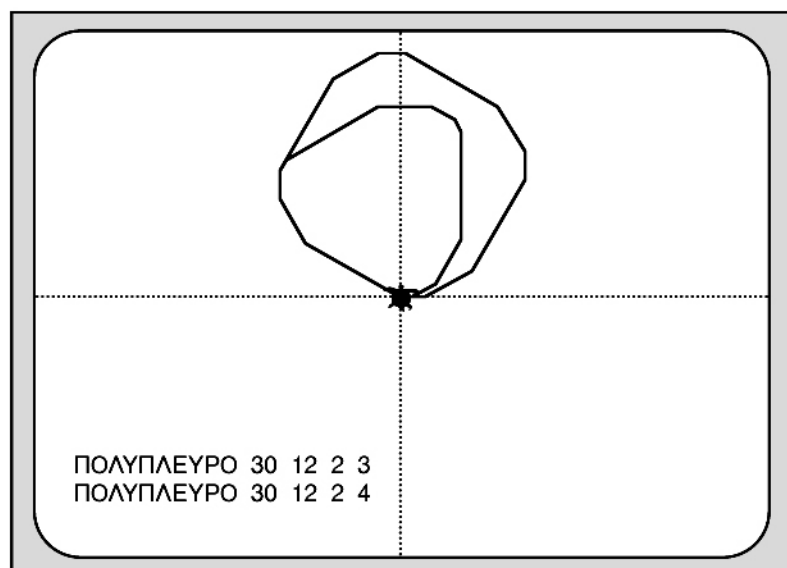
Παρατηρείστε την ύπαρξη μιας δομής επανάληψης μέσα σε μια άλλη. Είναι αξιοσημείωτο ότι η μια είναι φωλιασμένη στο εσωτερικό της άλλης και δεν υπάρχει περίπτωση διασταύρωσης των δύο αυτών δομών.



Ένα πιο ευέλικτο πολύπλευρο

Η προηγούμενη διαδικασία σχεδιάζει πολύπλευρα που έχουν προκύψει από την επανάληψη δύο πλευρών. Μπορούμε να βελτιώσουμε τη διαδικασία έτσι ώστε να δέχεται ως παράμετρο το πλήθος των διαδοχικών βασικών πλευρών που θα επαναλαμβάνονται. Αρχεί να χρησιμοποιήσουμε τη μεταβλητή DIADOXH που θα δίνει το πλήθος των διαδοχικών πλευρών που επαναλαμβάνονται και να κάνουμε τις σχετικές μικρές τροποποιήσεις στον κώδικα της προηγούμενης διαδικασίας.

```
ΤΟ ΠΟΛΥΠΛΕΥΡΟ :PLEYRA :PLHQOS :BHMA :DIADOXH
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS / :DIADOXH)
  [ MAKE "LOGOS 1
    REPEAT :DIADOXH
      [ ΓΡΑΜΜΗ :MOIRES :PLEYRA / :LOGOS
        MAKE "LOGOS :LOGOS * :BHMA_LOGOY
      ]
    ]
ΔΕΞΙΑ 90
END
```



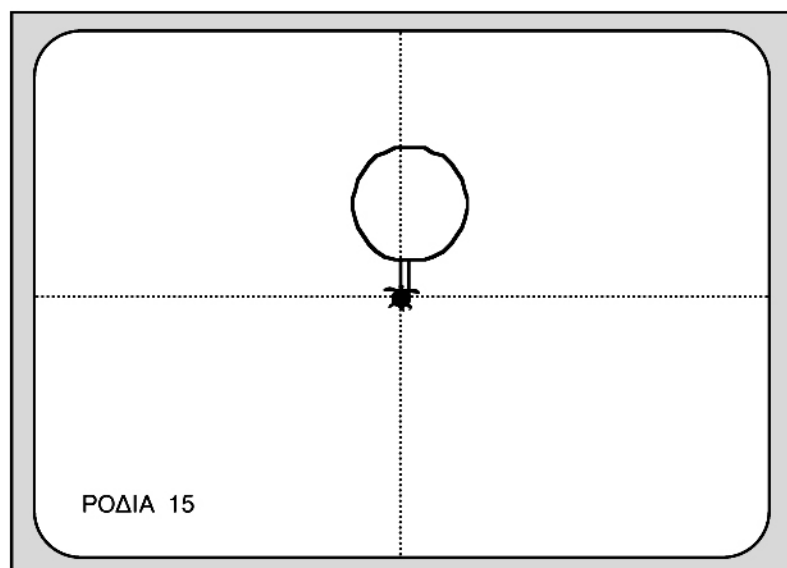
Σχεδιάζοντας τη ροδιά

Μια ροδιά μπορεί να σχεδιαστεί από τον κορμό και το φύλλωμα. Το φύλλωμα παριστάνεται ως κύκλος και ο κορμός σχεδιάζεται με τη βοήθεια του ορθογωνίου.

```
ΤΟ ΚΟΡΜΟΣ :YPS_KΟΡΜΟΥ  
MAKE "LOGOS 4  
ΚΙΝΗΣΗ_ΔΙΠΛΑ -(YPS_KΟΡΜΟΥ / (2*LOGOS))  
ΟΡΘΟΓΩΝΙΟ :YPS_KΟΡΜΟΥ :LOGOS  
ΚΙΝΗΣΗ_ΔΙΠΛΑ (:YPS_KΟΡΜΟΥ / (2*LOGOS))  
ΜΗ_ΓΡΑΦΕΙΣ  
ΕΥΘΕΙΑ :YPS_KΟΡΜΟΥ  
ΓΡΑΨΕ  
END
```

Η διαδικασία ΡΟΔΙΑ είναι:

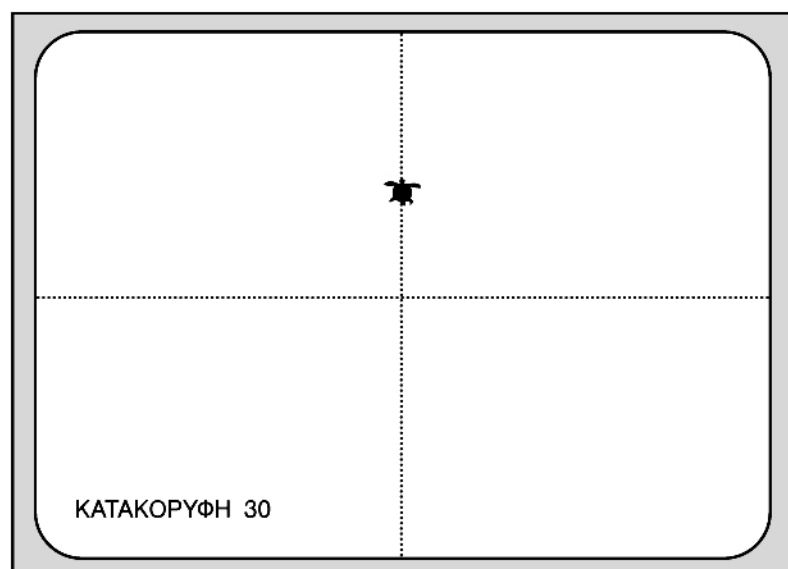
```
ΤΟ ΡΟΔΙΑ :ΜΕΓΕΘΟΣ  
ΚΟΡΜΟΣ :ΜΕΓΕΘΟΣ  
ΚΥΚΛΟΣ (:ΜΕΓΕΘΟΣ / 4)  
ΜΗ_ΓΡΑΦΕΙΣ  
ΟΠΙΣΘΕΝ :ΜΕΓΕΘΟΣ  
ΓΡΑΨΕ  
END
```



Κατακόρυφη μετατόπιση

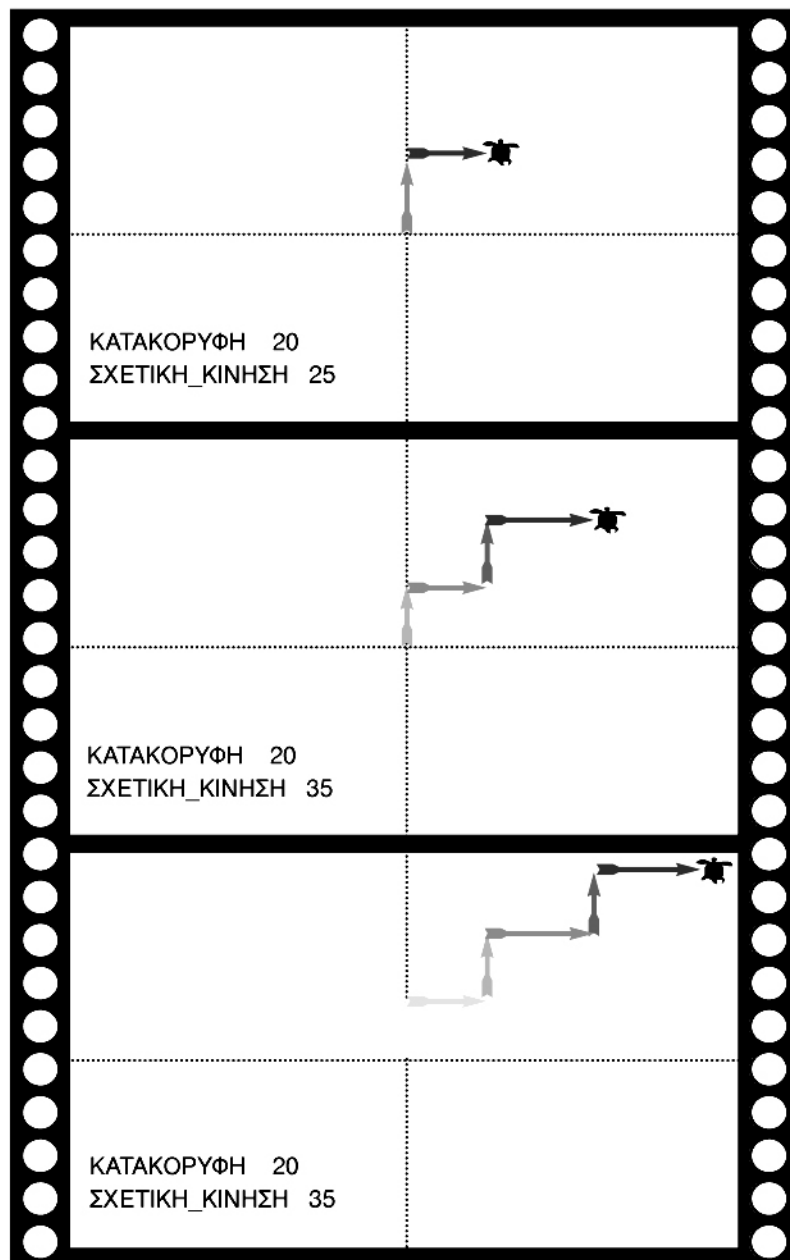
Εκτός από την οριζόντια μετατόπιση θα χρειαστεί και κατακόρυφη μετατόπιση της χελώνας. Η κατακόρυφη μετατόπιση είναι σχετικά εύκολη γιατί έχουμε συνηθίσει να αφήνουμε τη χελώνα με κατεύθυνση προς τα πάνω. Έτσι το μόνο που χρειάζεται είναι να πούμε στη χελώνα να μη γράφει, να κινηθεί ευθύγραμμα και τέλος να έρθει σε θέση για να γράφει πάλι.

```
ΤΟ ΚΑΤΑΚΟΡΥΦΗ :APOSTASH  
ΜΗ_ΓΡΑΦΕΙΣ  
ΕΥΘΕΙΑ :APOSTASH  
ΓΡΑΨΕ  
END
```



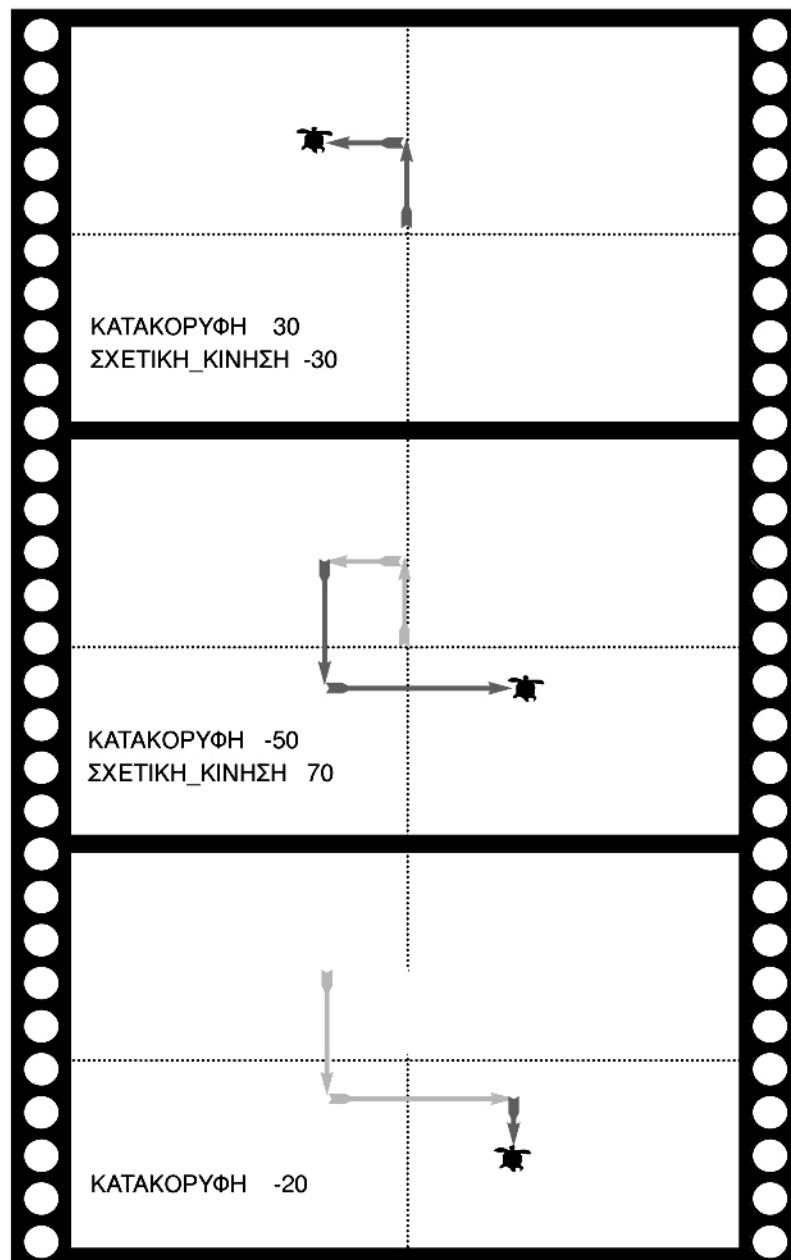
Η χελώνα μετατοπίζεται στο επίπεδο

Χρησιμοποιώντας τις ΚΑΤΑΚΟΡΥΦΗ και ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ μετακινούμε τη χελώνα σε όποιο σημείο της οθόνης θέλουμε.



Αρνητικές μετατοπίσεις

Δίνοντας αρνητικές τιμές στις προηγούμενες διαδικασίες μπορούμε να μετατοπίσουμε τη χελώνα προς όλες τις κατευθύνσεις.



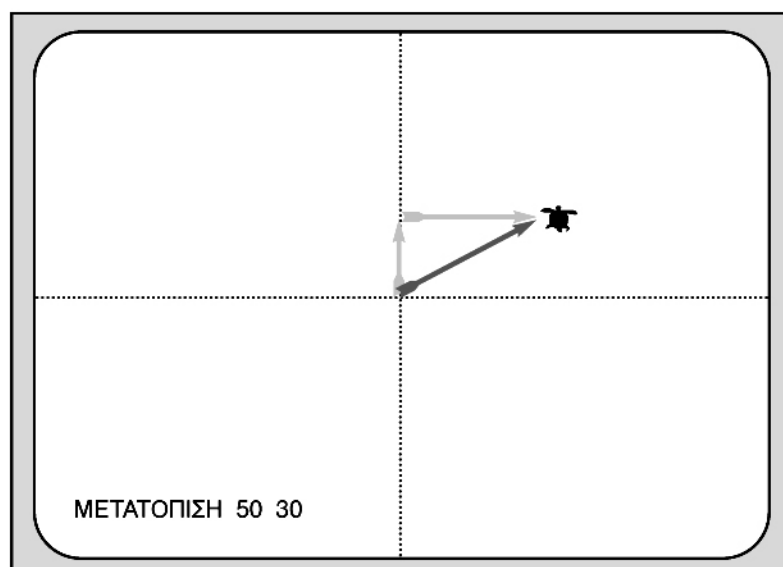
Μια διαδικασία μετατόπισης

Για την κίνηση στο επίπεδο οιδύο διαδικασίες, για την οριζόντια και για την κατακόρυφη μετατόπιση, μπορούν να αντικατασταθούν από μια. Αυτή η διαδικασία που θα την ονομάσουμε ΜΕΤΑΤΟΠΙΣΗ θα έχει δύο μεταβλητές που θα καθορίζουν την οριζόντια και την κάθετη συνιστώσα της μετατόπισης.

```
ΤΟ ΜΕΤΑΤΟΠΙΣΗ :Χ :Υ  
ΣΧΕΤΙΚΗ_ΚΙΝΗΣΗ :Χ  
ΚΑΤΑΚΟΡΥΦΗ : Υ  
END
```

Η πλάγια κίνηση θεωρείται ως **σύνθεση** μιας οριζόντιας και μιας κατακόρυφης.

Το διάνυσμα από μαθηματική άποψη, είναι ένα διατεταγμένο ζεύγος αριθμών.



Η τελική θέση της χελώνας στο σύστημα αναφοράς με αρχή το κέντρο της οθόνης, δίνεται από το διάνυσμα θέσης που σχεδιάζεται από το κέντρο του συστήματος μέχρι τη χελώνα.

Η χελώνα σκαφαλώνει σε μια σκάλα

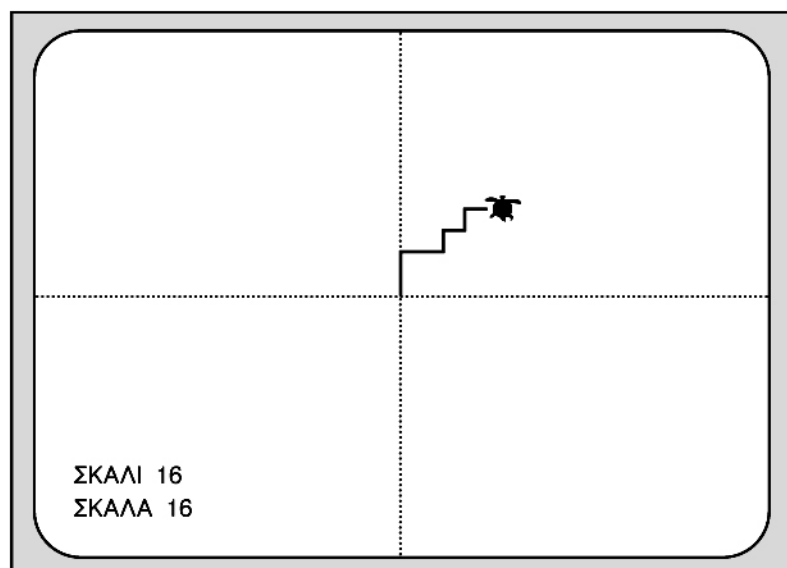
Χρησιμοποιώντας την εμπειρία από τις μετατοπίσεις, μπορούμε να φτιάξουμε μια διαδικασία που να διατάζει τη χελώνα να ακολουθήσει μια διαδρομή που να μοιάζει με σκάλα.

Το μόνο που χρειάζεται είναι η οριζόντια (ή η κατακόρυφη) συνιστώσα της απόστασης που πρέπει να διανύσει η χελώνα.

```
ΤΟ ΣΚΑΛΙ :APOSTASH
MAKE "ΥPSOS :APOSTASH
ΕΥΘΕΙΑ :ΥPSOS
MAKE "PLATOS :APOSTASH
ΔΕΞΙΑ 90
ΕΥΘΕΙΑ :PLATOS
ΑΡΙΣΤΕΡΑ 90
END
```

Έστω ότι θέλουμε να ανέβουμε στο ίδιο ύψος αλλά το σκαλί που σχεδιάσαμε προηγουμένως είναι πολύ ψηλό για τη χελώνα. Ας σχεδιάσουμε μια σκάλα που να αποτελείται από δύο σκαλιά που να έχουν το μισό ύψος.

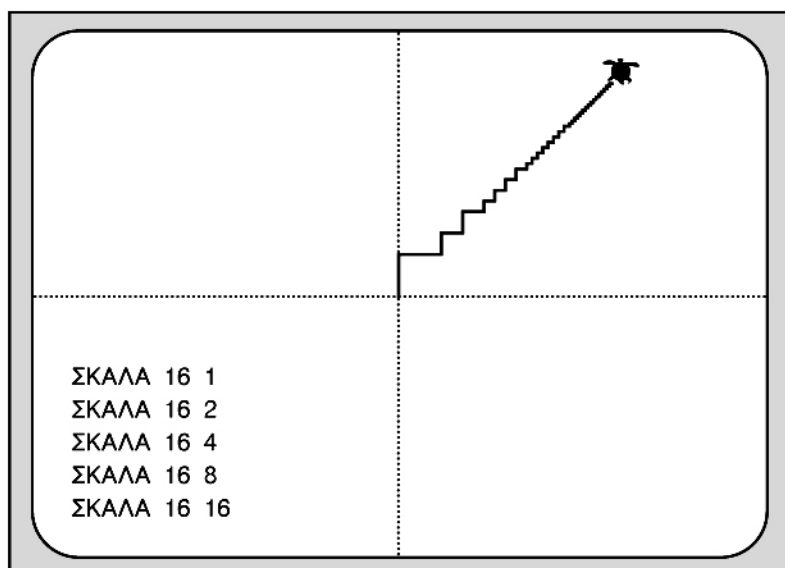
```
ΤΟ ΣΚΑΛΑ :APOSTASH
REPEAT 2
  [ ΣΚΑΛΙ (:APOSTASH / 2) ]
END
```



Επιλέγοντας το πλήθος των σκαλιών

Το επόμενο βήμα είναι μια σκάλα που θα καλύπτει την ίδια κατακόρυφη και οριζόντια απόσταση αλλά με περισσότερα σκαλιά το πλήθος των οποίων θα δίνεται ως παράμετρος της διαδικασίας.

```
TO ΣΚΑΛΑ :ΑPOSTASH :PLHQOS
MAKE "ΥPSOS (:ΑPOSTASH / PLHQOS)
MAKE "PLATOS (:ΑPOSTASH / PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :ΥPSOS
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90 ]
END
```



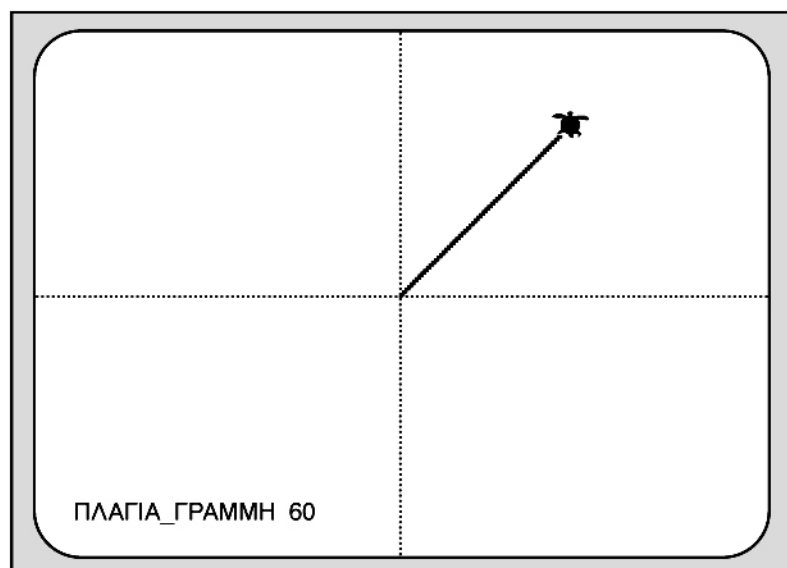
Πλάγια γραμμή

Η πραγματική απόσταση που διανύει σε ευθεία γραμμή η χελώνα από το σημείο που ξεκινάει μέχρι εκεί που καταλήγει, υπολογίζεται από το Πυθαγόρειο Θεώρημα ως η υποτείνουσα του ορθογώνιου τριγώνου με κάθετες πλευρές την κατακόρυφη και την οριζόντια απόσταση.

Αν το πλήθος των σκαλιών είναι ίσο με τα βήματα που πρέπει να γίνουν για να καλύψει η χελώνα την πραγματική απόσταση τότε το μέγεθος των σκαλιών-βημάτων θα προκύπτει από τη διαίρεση της υποτείνουσας δια το πλήθος των σκαλιών.

Έτσι μπορούμε να γράψουμε μια διαδικασία που να χαράζει μια πλάγια γραμμή.

```
TO ΠΛΑΓΙΑ_ΓΡΑΜΜΗ :ΑPOSTASH
MAKE "Y :ΑPOSTASH
MAKE "X :ΑPOSTASH
MAKE "ΥΡΟΤΕΙΝ SQRT((:Y * :Y) + (:X * :X))
MAKE "PLHQOS (:ΥΡΟΤΕΙΝ / 1)
MAKE "ΥPSOS (:Y / :PLHQOS)
MAKE "PLATOS (:X / :PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :ΥPSOS
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90 ]
END
```

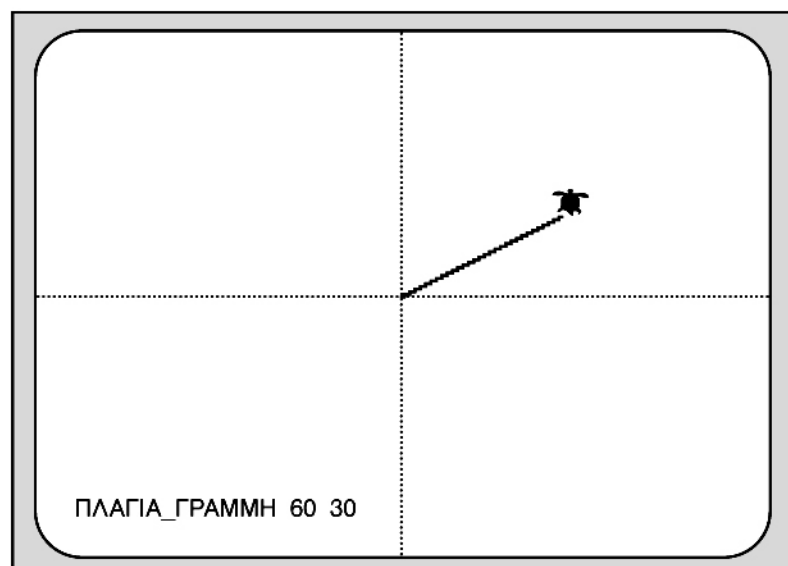


Διαισθητικός
ορισμός του
ορίου:
Αν υπάρχει μια
άπειρη
ακολουθία
διαδοχικών
προσεγγίσεων
ενός αριθμού,
τότε ονομάζουμε
αυτόν τον
αριθμό όριο της
ακολουθίας.
Παράδειγμα:
Η ακολουθία
0,3 0,33 0,333
0,3333... είναι
μια ακολουθία
διαδοχικών
προσεγγίσεων
του αριθμού 1/3.
Άρα το όριο της
είναι το 1/3.

Επιλογή της κλίσης της πλάγιας γραμμής

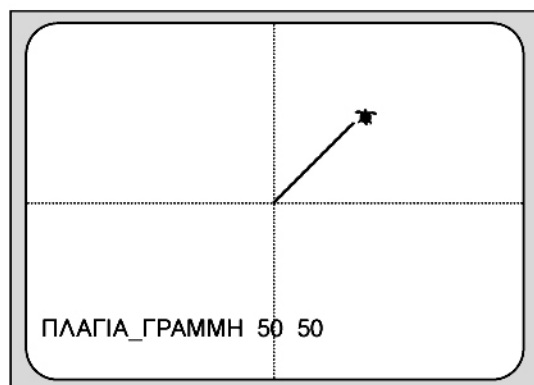
Αν η οριζόντια και η κατακόρυφη απόσταση δεν είναι ίδιες, τότε η κλίση της γωνίας δε θα είναι πάντοτε η ίδια. Έτσι αν τροποποιήσουμε τη διαδικασία ΠΛΑΓΙΑ_ΓΡΑΜΜΗ έτσι ώστε να δέχεται ως παραμέτρους την οριζόντια και την κατακόρυφη απόσταση τότε θα μπορούμε να αλλάζουμε και την κλίση της πλάγιας γραμμής.

```
TO ΠΛΑΓΙΑ_ΓΡΑΜΜΗ :X :Y
MAKE "ΥΡΟΤΕΙΝ SQRT((:Y * :Y) + (:X * :X))
MAKE "PLHQOS (:ΥΡΟΤΕΙΝ / 1)
MAKE "ΥPSOS (:Y / :PLHQOS)
MAKE "PLATOS (:X / :PLHQOS)
REPEAT :PLHQOS
  [ ΕΥΘΕΙΑ :ΥPSOS
    ΔΕΞΙΑ 90
    ΕΥΘΕΙΑ :PLATOS
    ΑΡΙΣΤΕΡΑ 90
  ]
END
```

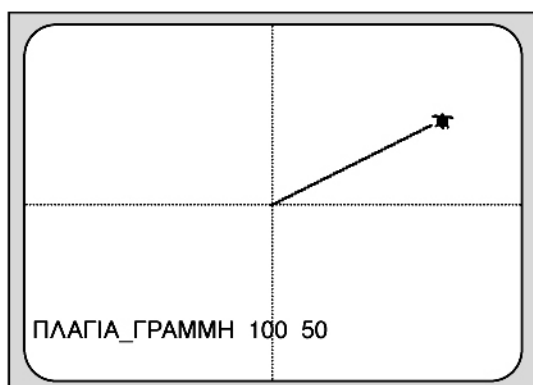


Πλάγιες γραμμές με διαφορετικές κλίσεις

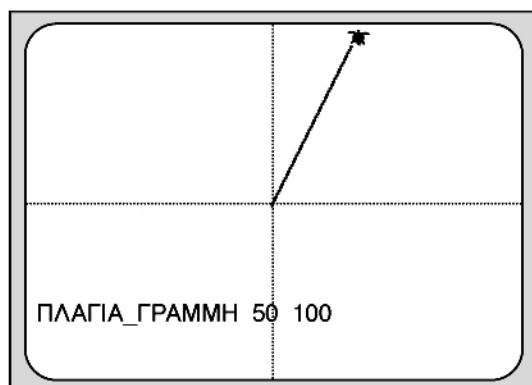
Ας πειραματιστούμε σχεδιάζοντας μερικές πλάγιες γραμμές που να έχουν μια ποικιλία οριζόντιων και κατακόρυφων αποστάσεων.



Η πλάγια ευθεία που έχει ίση την κατακόρυφη και την οριζόντια απόσταση έχει κλίση ίση με μισή ορθή γωνία.



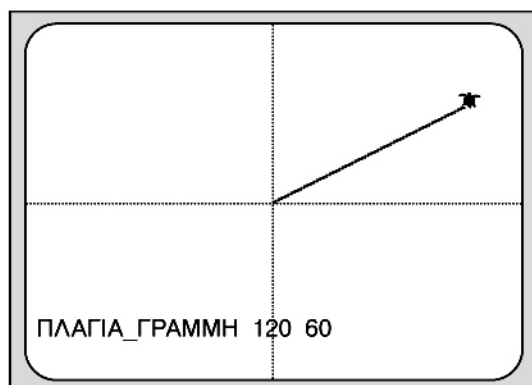
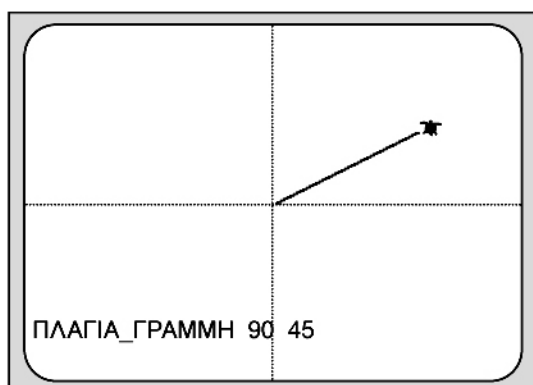
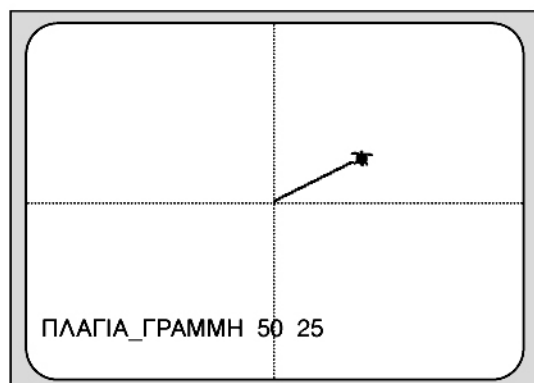
Οι πλάγιες ευθείες με παραμέτρους $(100, 50)$ και $(50, 100)$ ενώ έχουν το ίδιο μήκος, έχουν διαφορετική κλίση.



Γιατί αυτές οι δύο ευθείες έχουν το ίδιο μήκος;

Η αναλογία των αποστάσεων και η κλίση

Ας σχεδιάσουμε τρεις πλάγιες γραμμές με οριζόντιες και κατακόρυφες αποστάσεις (50 , 25) , (90 , 45) και (120 , 60) αντίστοιχα.



Οι τρεις πλάγιες γραμμές αν και έχουν διαφορετικά μήκη, έχουν την ίδια κλίση.

Πρέπει να αναζητήσουμε τι είναι αυτό από τις παραμέτρους εισόδου που μένει σταθερό και στις τρεις περιπτώσεις. Παρατηρούμε ότι οι λόγοι 25/50, 45/90 και 60/120 είναι ίσοι.

Άρα η κλίση μιας πλάγιας γραμμής εξαρτάται από το λόγο της κατακόρυφης ως προς την οριζόντια απόσταση.

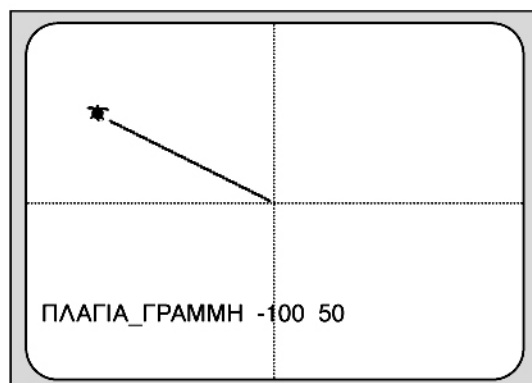
Αυτός ο λόγος εκφράζεται από τον

τριγωνομετρικό αριθμό που λέγεται εφαπτομένη της γωνίας που σχηματίζει η πλάγια ευθεία με την οριζόντια.

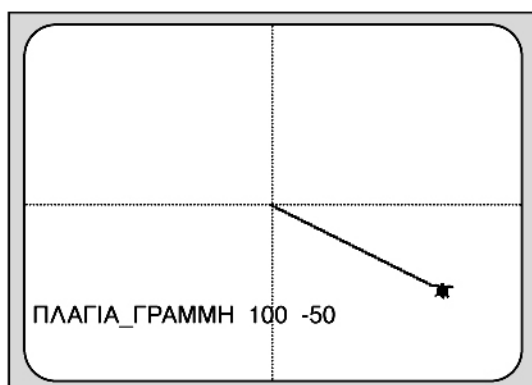
Η γωνία μπορεί να μετρηθεί (εκτός από μοίρες) και σε μια ακόμα μονάδα, το ακτίνιο. Η γωνία της συγκεκριμένης πλάγιας ευθείας είναι 0,5 ακτίνια.

Μετρήσεις της γωνίας σε ακτίνια

Θα χαράξουμε τρεις ακόμα πλάγιες ευθείες και θα υπολογίσουμε την γωνία της κλίσης τους σε ακτίνια.

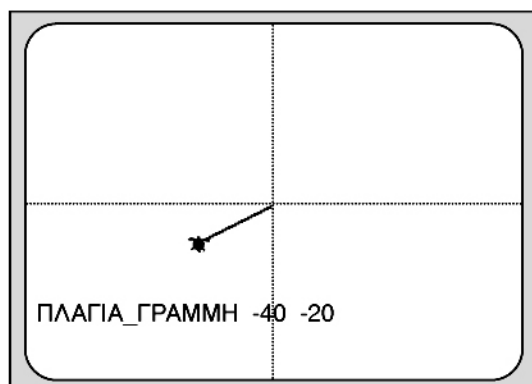


Η γωνία της κλίσης υπολογίζεται από το λόγο της κατακόρυφης προς την οριζόντια απόσταση. Έτσι η κλίση της πλάγιας ευθείας με παραμέτρους -100, 50 θα είναι $50/(-100) = -0,5$



Η κλίση της πλάγιας ευθείας με παραμέτρους 100, -50 είναι $(-50)/100 = -0,5$

Οι δύο προηγούμενες πλάγιες ευθείες έχουν την ίδια κλίση.



Η κλίση της πλάγιας ευθείας με παραμέτρους -40, -20 είναι $(-20)/(-40) = 0,5$

Μισό πολύγωνο

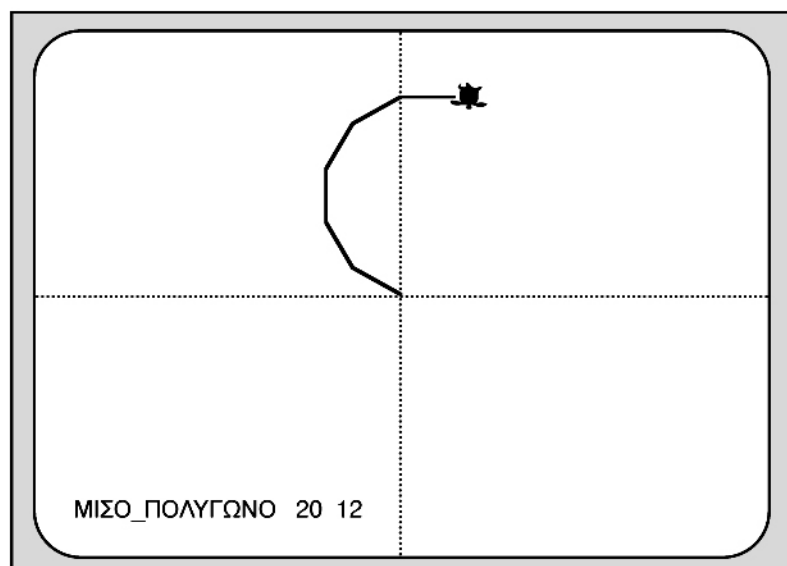
Η διαδικασία
ΤΟ ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT :PLHQOS
[ΓΡΑΜΜΗ :MOIRES :PLEYRA]
ΔΕΞΙΑ 90
END

σχεδιάζει ένα πολύγωνο.

Για να σχηματισθεί μόνο το μισό πολύγωνο θα πρέπει να σχεδιάσουμε μόνο το μισό πλήθος των πλευρών του. Δηλαδή αντί η επανάληψη να γίνει όλες φορές είναι το πλήθος των πλευρών θα πρέπει να γίνει τις μισές.

Ας δούμε πως θα γράψουμε τη διαδικασία που σχεδιάζει μισό πολύγωνο.

ΤΟ ΜΙΣΟ_ΠΟΛΥ ΓΩΝΟ :PLEYRA :PLHQOS
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS * (1 / 2))
[ΓΡΑΜΜΗ :MOIRES :PLEYRA]
ΔΕΞΙΑ 90
END

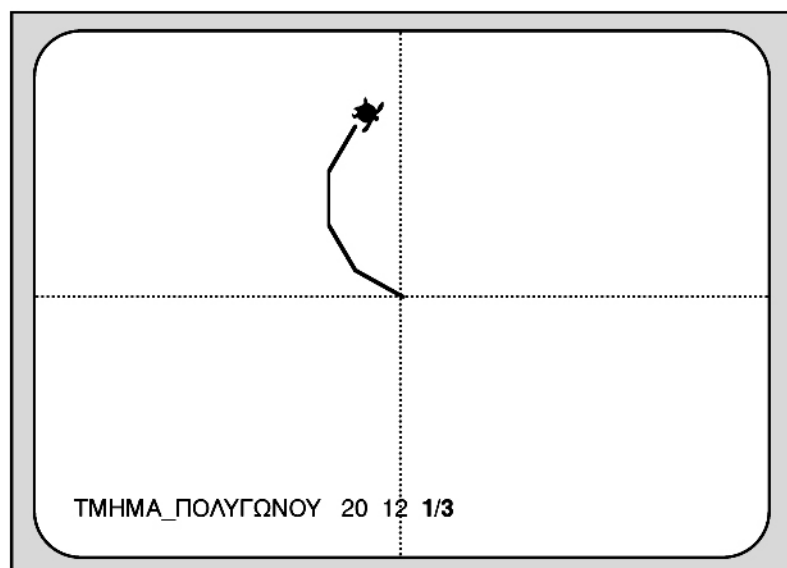


Ένα κομμάτι από ολόκληρο το πολύγωνο

Μπορούμε να σχεδιάζουμε κάποιο κλάσμα του πολυγώνου προχωρώντας στη γενίκευση με μια διαδικασία η οποία χρησιμοποιώντας μια παράμετρο επιπλέον, σχεδιάζει ένα τμήμα του πολυγώνου. Η επιπλέον αυτή παράμετρος θα είναι η μεταβλητή ΚΛΑΣΜΑ που θα αντιπροσωπεύει το κλάσμα του πολυγώνου που θέλουμε να σχεδιάσουμε.

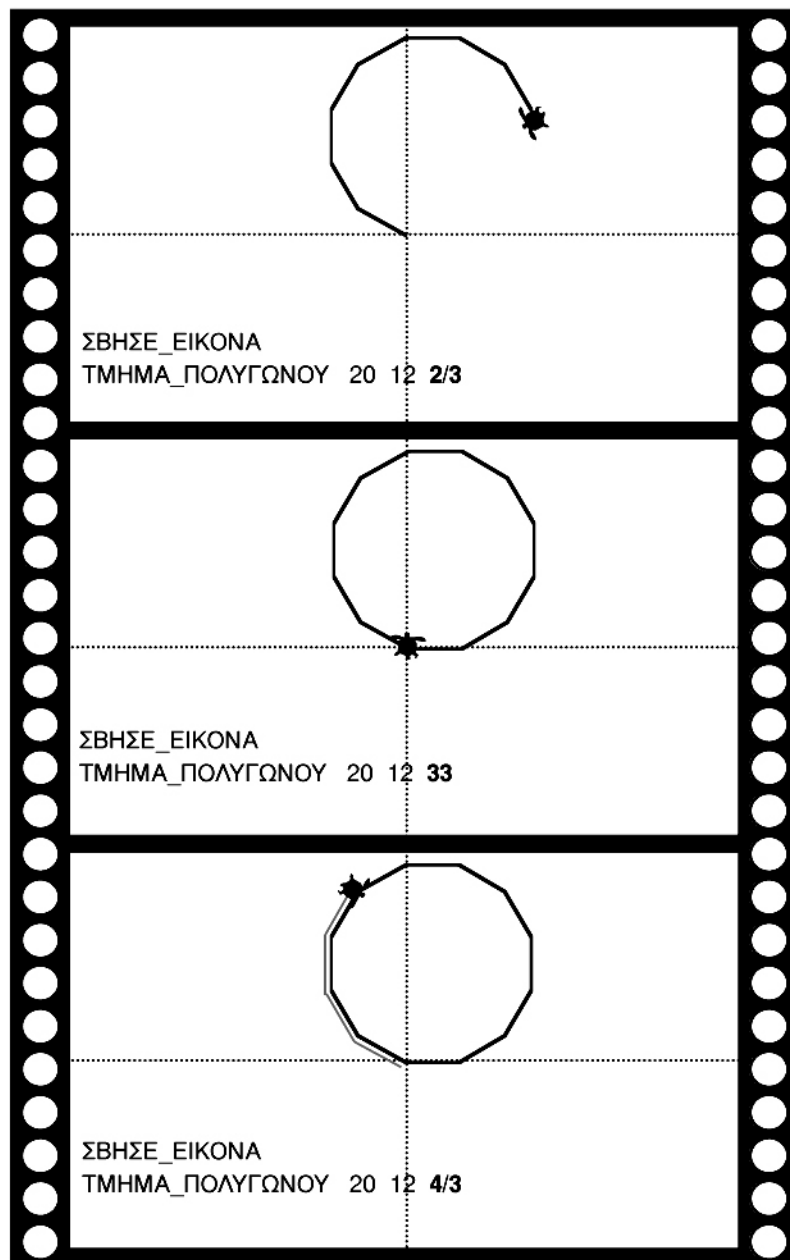
Να μια μέθοδος αναπαράστασης "γραμμικής" μορφής για τη διδασκαλία της έννοιας κλάσμα.

```
ΤΟ ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ :PLEYRA :PLHQOS :KLASMA
ΑΡΙΣΤΕΡΑ 90
MAKE "MOIRES (360 / :PLHQOS)
REPEAT (:PLHQOS * :KLASMA)
  [ ΓΡΑΜΜΗ :MOIRES :PLEYRA ]
ΔΕΞΙΑ 90
END
```



2/3, 3/3 και 4/3

Ας σχεδιάσουμε μερικά τμήματα πολυγώνων που αντιστοιχούν στα κλάσματα 2/3, 3/3 και 4/3.



Τα 6/6 είναι
μεγαλύτερα
από τα 2/2;

Ένα κομμάτι της περιφέρειας του κύκλου

Είχαμε δεχθεί ότι ο κύκλος θα σχεδιάζεται ως ένα πολύγωνο με 36 πλευρές. Αφού λοιπόν ο κύκλος "είναι" πολύγωνο, μπορούμε να εφαρμόσουμε και σ' αυτόν τα σχετικά με τα τμήματα των πολυγώνων. Έτσι θα μπορούμε να σχεδιάζουμε ημικύκλια, τεταρτοκύκλια και άλλα τμήματα του κύκλου.

Η τροποποίηση που πρέπει να γίνει στη διαδικασία ΚΥΚΛΟΣ

ΤΟ ΚΥΚΛΟΣ :PLEYRA

ΑΡΙΣΤΕΡΑ 90

MAKE "MOIRES (360/36)

REPEAT 36

— [ΓΡΑΜΜΗ :MOIRES :PLEYRA]

ΔΕΞΙΑ 90

END

είναι να συμπληρωθεί με τη μεταβλητή ΚΛΑΣΜΑ και να χρησιμοποιηθεί στο σημείο που καθορίζεται το πλήθος των επαναλήψεων.

ΤΟ ΚΥΚΛΟΣ :PLEYRA:ΚΛΑΣΜΑ

ΑΡΙΣΤΕΡΑ 90

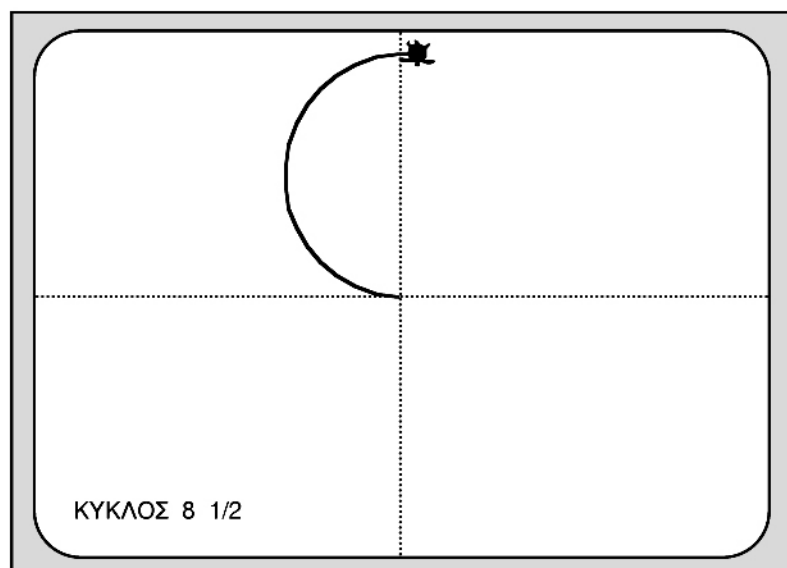
MAKE "MOIRES (360 / 36)

REPEAT (36 * :ΚΛΑΣΜΑ)

[ΓΡΑΜΜΗ :MOIRES :PLEYRA]

ΔΕΞΙΑ 90

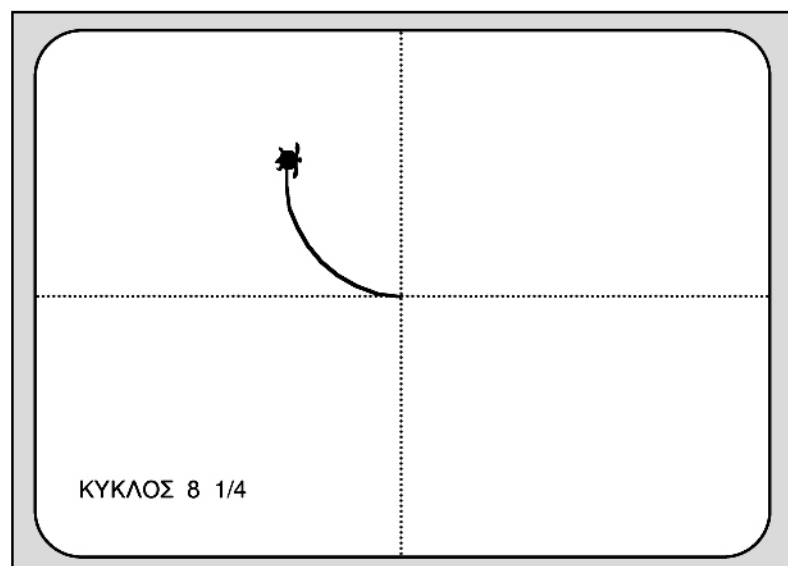
END



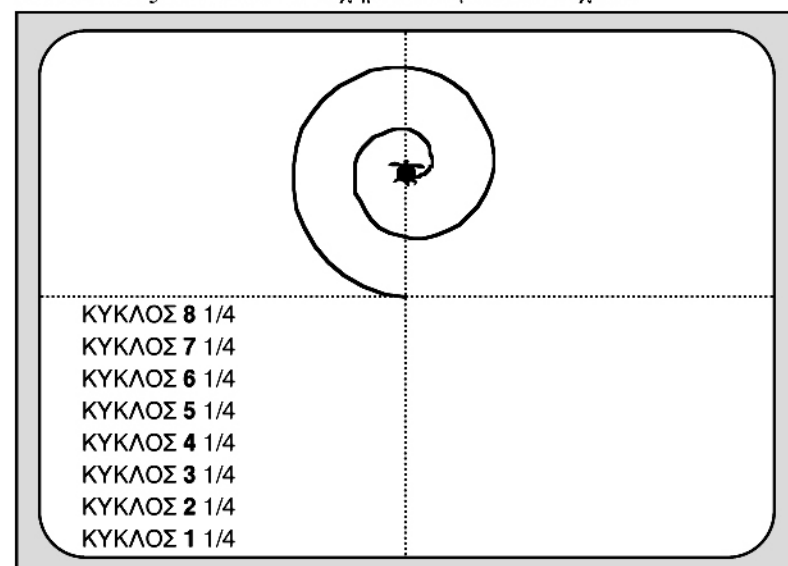
ημικύκλιο

Τεταρτοκύκλιο και κοχλίας

Χρησιμοποιώντας τη νέα έκδοση της διαδικασίας ΚΥΚΛΟΣ, μπορούμε να σχεδιάσουμε ένα τεταρτοκύκλιο δίνοντας απλά ως τιμή στην παράμετρο ΚΛΑΣΜΑ το $1/4$.

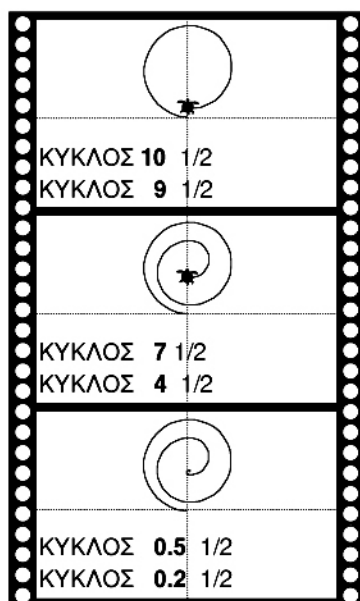


Κρατώντας σταθερό το κλάσμα και μειώνοντας την τιμή της μεταβλητής PLEYRA μπορούμε με διαδοχικές κλήσεις της διαδικασίας ΚΥΚΛΟΣ να σχηματίσουμε έναν κοχλία.

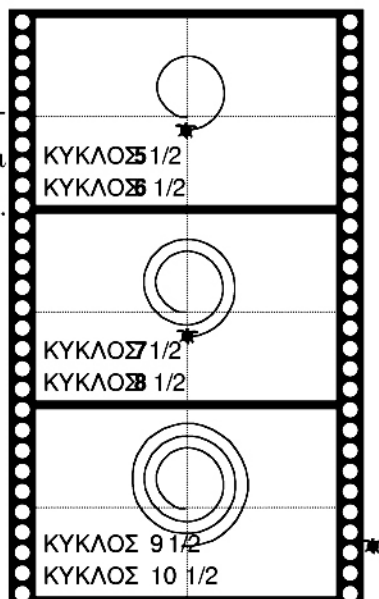


Κοχλίας από ημικύκλια

Ας χρησιμοποιήσουμε ημικύκλια αντί για τεταρτοκύκλια για τη δημιουργία ενός κοχλίας.



Με αύξηση του μεγέθους της πλευ-
ράς, ο “εσωστρεφής” κοχλίας γίνεται
“εξωστρεφής”.



Το πρόγραμμα του κοχλία

Ας γράψουμε μια διαδικασία που να σχεδιάζει έναν κοχλία. Ως παραμέτρους θα δέχεται το μέγεθος, το πλήθος των στροφών, το αν θα είναι εσωστρεφής ή εξωστρεφής και το κλάσμα του κύκλου που θα χρησιμοποιηθεί.

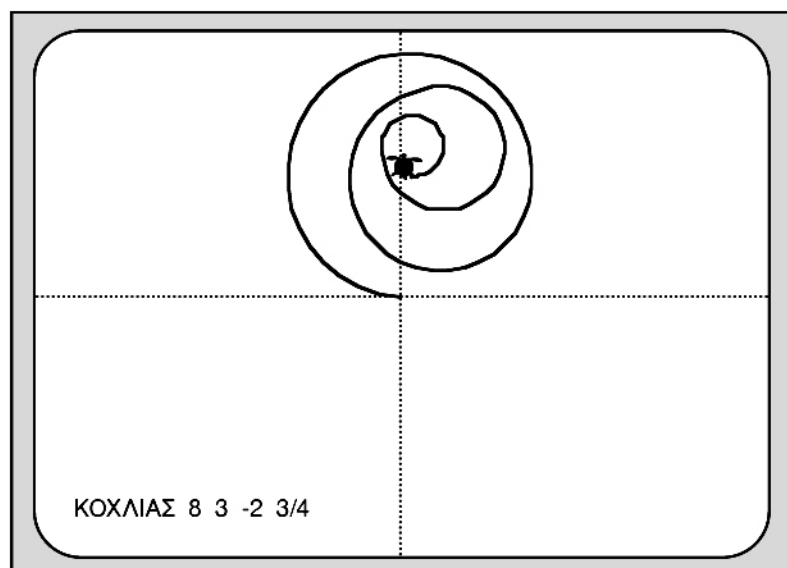
```
TO ΚΟΧΛΙΑΣ :MEG :STROFH :ESEX :KLASMA
MAKE "FORES (:STROFH / :KLASMA)
REPEAT :FORES
  [ ΚΥΚΛΟΣ :MEG :KLASMA
    MAKE "MEG :MEG + ESEX ]
END
```

Μπορείτε να μαντέψετε τι σχήματα θα προκύψουν από τις εξής κλήσεις της διαδικασίας ΚΟΧΛΙΑΣ;

ΚΟΧΛΙΑΣ 5 4 -1 1/2

ΚΟΧΛΙΑΣ -3 3 2 1/2

ΚΟΧΛΙΑΣ 3 3 2 -1/2



11 Animation / Εμ-ψυχο-ποίηση *

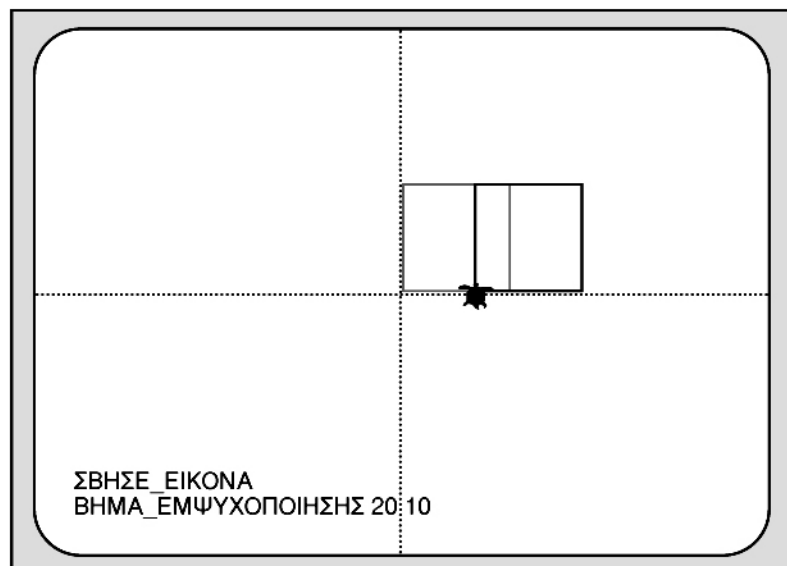
Η αίσθηση ότι το τετράγωνο ολισθαίνει

Ένα σχήμα μπορούμε να το σβήσουμε αν διατάξουμε τη χελώνα να το διατρέξει αλλά αν αντί να γράφει, να σβήνει. Η κατάσταση με την οποία η χελώνα σβήνει αντί να γράφει, υλοποιείται με την εντολή **Pen Erase** (που σημαίνει η πένα σβήνει) ή σε συντομογραφία **PE**. Μια "εξελληνισμένη" διαδικασία που θα πραγματοποιεί το ίδιο έργο με την εντολή **Pen Erase** είναι η διαδικασία **ΣΒΗΣΕ**.

```
ΤΟ ΣΒΗΣΕ  
PE  
END
```

Μπορούμε να δημιουργήσουμε την αίσθηση ότι ένα σχήμα που σχεδιάστηκε στην οθόνη κινείται. Αυτό το πετυχαίνουμε αν αφού σχεδιάσουμε το σχήμα το σβήσουμε, και στη συνέχεια το ξανασχεδιάσουμε μετατοπισμένο. Αυτό κάνει η διαδικασία **ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ**:

```
ΤΟ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ  
ΠΟΛΥΓΩΝΟ 40 4  
ΣΒΗΣΕ  
ΠΟΛΥΓΩΝΟ 40 4  
ΜΕΤΑΤΟΠΙΣΗ 10 0  
ΓΡΑΨΕ  
ΠΟΛΥΓΩΝΟ 40 4  
END
```



* Στην έβδομη τέχνη, τον κινηματογράφο, έχει καθιερωθεί ως **animation** να αναφέρονται οι ταινίες κινουμένων σχεδίων. Η αγγλική λέξη "**animation**" προέρχεται από τη λατινική "**anima**" που σημαίνει ψυχή. Στα ελληνικά ο όρος "**animation**" αποδίδεται ως "κινούμενο σχέδιο". Νομίζουμε ότι το "κινούμενο σχέδιο" δεν αποδίδει την έννοια του "**animation**" και γι' αυτό προτείνουμε τον όρο "**εμψυχοποίηση**". Το συνθετικό "**ποίηση**" αποδίδει την έννοια της δημιουργίας και ταυτόχρονα διαφοροποιεί τη λέξη από την έννοια του ρήματος "**εμψυχώνω**" που έχει διαφορετική σημασία.

Κίνηση τετραγώνου προς κάθε κατεύθυνση

Μπορούμε να βελτιώσουμε τη διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ βάζοντας της ως παραμέτρους την οριζόντια και την κατακόρυφη απόσταση που θα μετατοπισθεί το τετράγωνο.

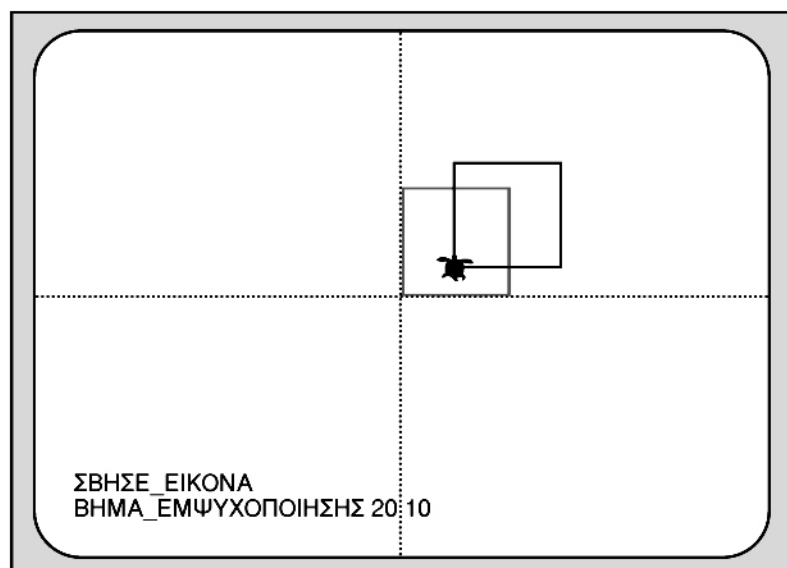
Η μεταβλητή που θα αντιπροσωπεύει την οριζόντια συνιστώσα της μετατόπισης ονομάζεται DX και η αντίστοιχη που θα αντιπροσωπεύει την κατακόρυφη DY.

Έτσι η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ γίνεται:

```
ΤΟ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY  
ΠΟΛΥΓΩΝΟ 40 4
```

```
ΣΒΗΣΕ  
ΠΟΛΥΓΩΝΟ 40 4
```

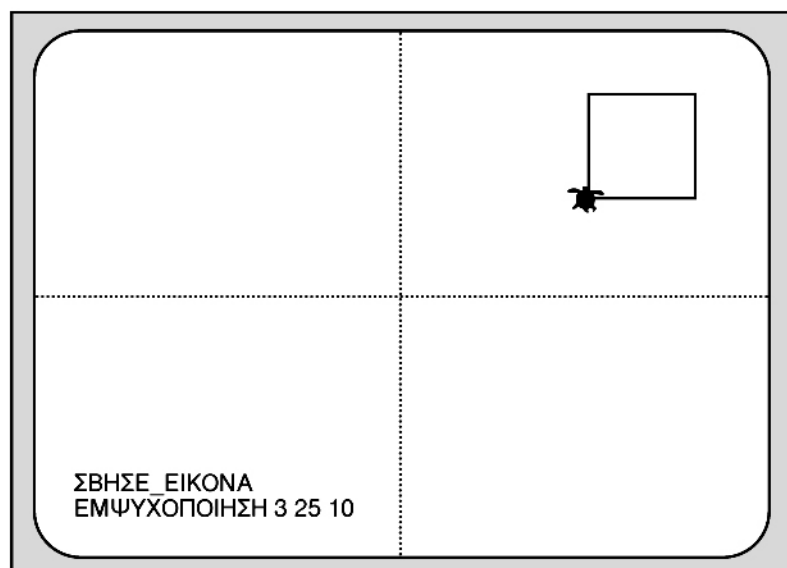
```
ΜΕΤΑΤΟΠΙΣΗ :DX :DY  
ΓΡΑΨΕ  
ΠΟΛΥΓΩΝΟ 40 4  
END
```



Ολοκληρωμένη κίνηση τετραγώνου

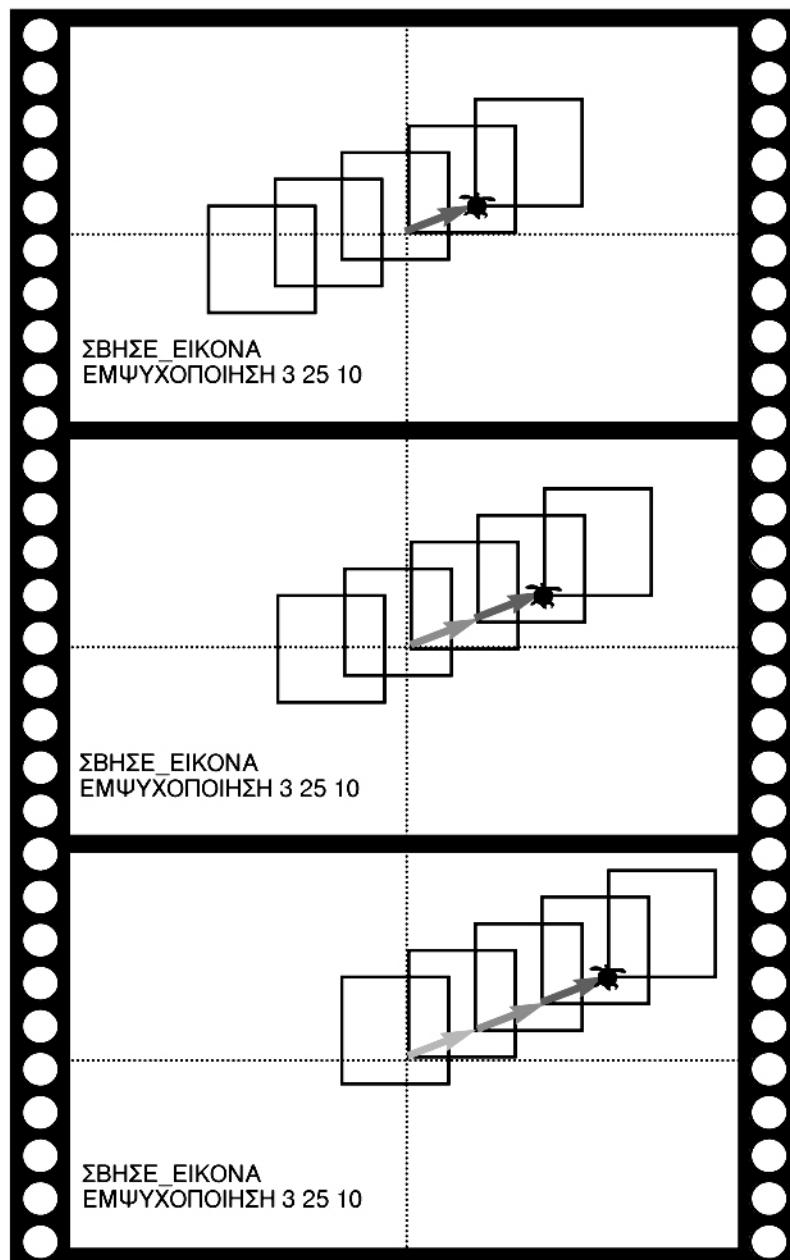
Η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ πραγματοποιεί μόνο ένα βήμα. Μια διαδικασία που επαναλαμβάνει πολλές φορές τη διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ θα δημιουργεί την αίσθηση μιας ολοκληρωμένης κίνησης. Την αίσθηση αυτή τη δημιουργεί η επόμενη διαδικασία, ΕΜΨΥΧΟΠΟΙΗΣΗ. Η μεταβλητή ΒΗΜΑΤΑ καθορίζει το πόσες φορές θα επαναληφθεί η διαδικασία ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ.

```
ΤΟ ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY  
REPEAT :ΒΗΜΑΤΑ  
  [ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY ]  
END
```



Τα διαδοχικά βήματα της κίνησης

Ας παρακολουθήσουμε τα τρία διαδοχικά βήματα που θα πραγματοποιήσει η κλήση της διαδικασίας ΕΜΨΥΧΟΠΟΙΗΣΗ 3 25 10.



Μια ταχύτερη κίνηση

Η εμφάνιση της χελώνας στην οθόνη καθώς κινείται για να σχηματίσει ή να σβήσει το τετράγωνο, κάνει την κίνηση αργή. Μπορούμε με την εντολή **Hide Turtle** (κρύψε τη χελώνα) ή σε συντομογραφία **HT** να εξαφανίσουμε την εικόνα της χελώνας από την οθόνη. Μια "εξελληνισμένη" διαδικασία που θα πραγματοποιεί το ίδιο έργο με την εντολή **Hide Turtle**, είναι η διαδικασία **ΚΡΥΨΟΥ** που ορίζεται ως:

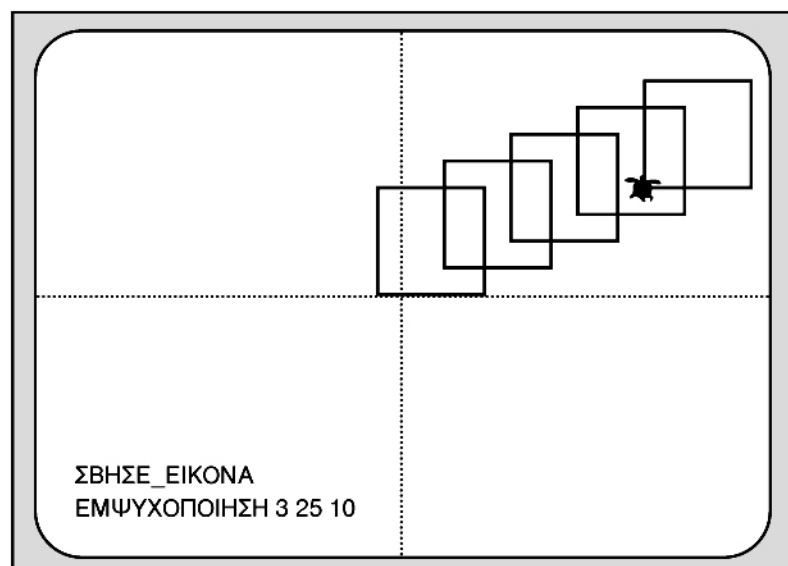
```
TO ΚΡΥΨΟΥ  
HT  
END
```

Η εντολή που επανεμφανίζει τη χελώνα στην οθόνη είναι η **Show Turtle** (εμφάνισε τη χελώνα) ή σε συντομογραφία **ST**. Μια "εξελληνισμένη" διαδικασία για το ίδιο έργο είναι η **ΒΓΕΣ**.

```
TO ΒΓΕΣ  
ST  
END
```

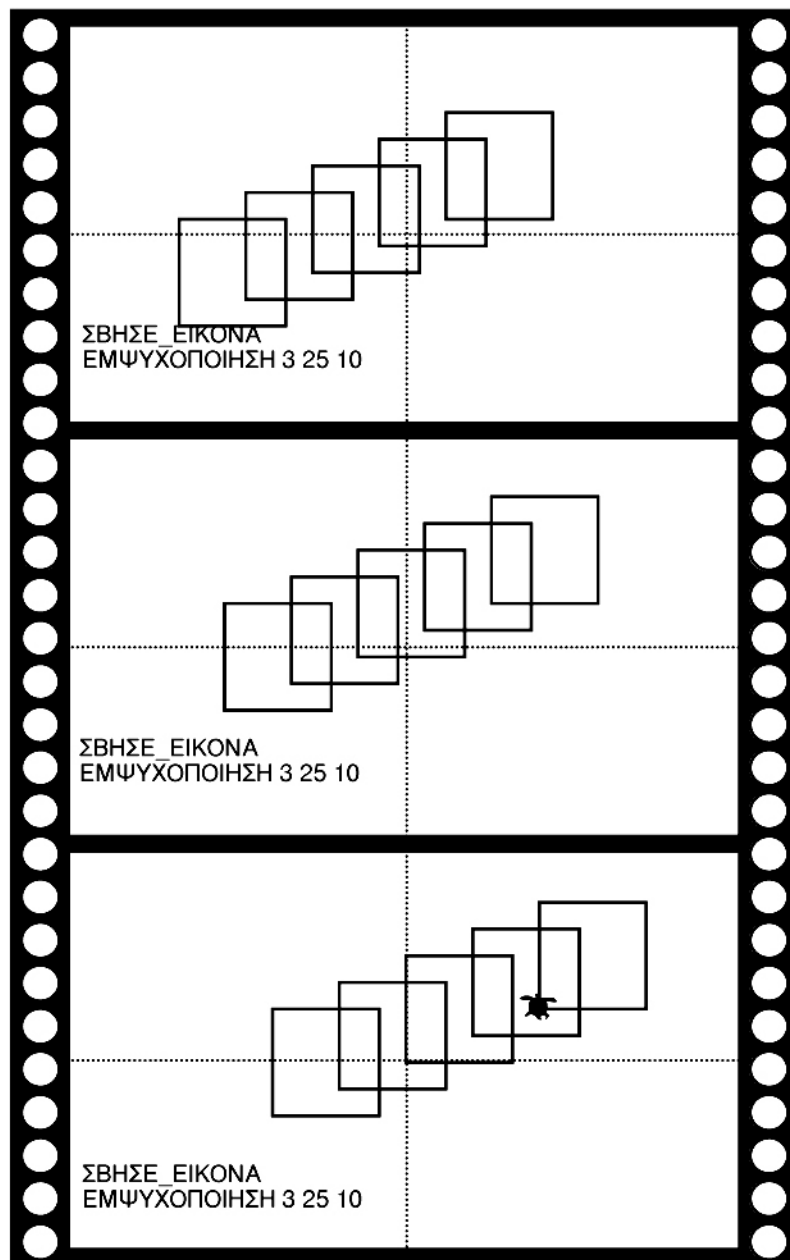
Με τη βοήθεια της διαδικασίας **ΚΡΥΨΟΥ**, η διαδικασία **ΕΜΨΥΧΟΠΟΙΗΣΗ** τροποποιείται και γίνεται ταχύτερη.

```
TO ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY  
ΚΡΥΨΟΥ  
REPEAT :ΒΗΜΑΤΑ  
[ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY]  
ΒΓΕΣ  
END
```



Διαδοχικά βήματα της ταχύτερης κίνησης

Η χελώνα εμφανίζεται μετά την ολοκλήρωση της κίνησης και έτσι η προσομοίωση της κίνησης είναι ταχύτερη και πιο ρεαλιστική.

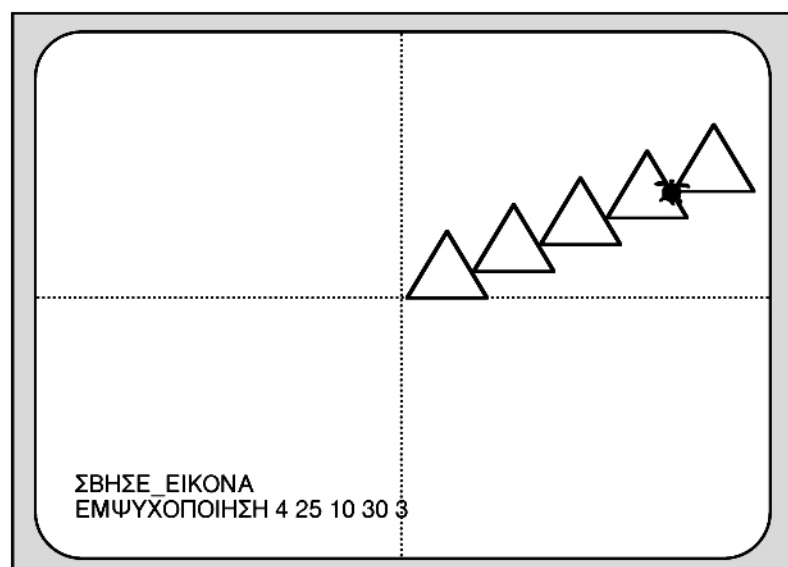


Ας επιλέξουμε το σχήμα που θα κινήσουμε

Μέχρι τώρα, το σχήμα που κινούσε η διαδικασία ΕΜΨΥΧΟΠΟΙΗΣΗ ήταν ένα τετράγωνο με πλευρά 40. Μπορούμε να προσθέσουμε δυο παραμέτρους στις διαδικασίες ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ και ΕΜΨΥΧΟΠΟΙΗΣΗ, που θα καθορίζουν το πλήθος των πλευρών του πολυγώνου (PLHQOS / PLHQ) καθώς και το μήκος της πλευράς του (PLEYRA / PLEYR).

```
ΤΟ ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS  
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS  
ΣΒΗΣΕ  
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS  
ΜΕΤΑΤΟΠΙΣΗ :DX :DY  
ΓΡΑΨΕ  
ΠΟΛΥΓΩΝΟ :PLEYRA :PLHQOS  
ΕΝΔ
```

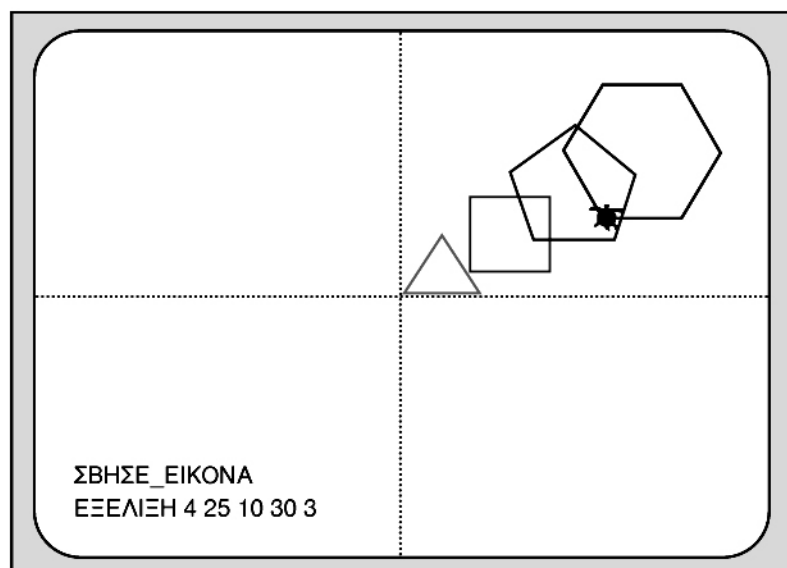
```
ΤΟ ΕΜΨΥΧΟΠΟΙΗΣΗ :ΒΗΜΑΤΑ :DX :DY :PLEYRA :PLHQOS  
ΚΡΥΨΟΥ  
REPEAT :ΒΗΜΑΤΑ  
  [ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS]  
ΒΓΕΣ  
ΕΝΔ
```



Οι γωνίες αυξάνονται

Μπορούμε να χρησιμοποιήσουμε τη δομή της διαδικασίας ΕΜΨΥΧΟΠΟΙΗΣΗ για να παρουσιάσουμε σε κάθε διαδοχικό βήμα και ένα πολύγωνο που να διαθέτει μια επιπλέον πλευρά από αυτές που είχε στο προηγούμενο βήμα. Αυτό γίνεται στην επόμενη διαδικασία ΕΞΕΛΙΞΗ όπου μέσα στην επανάληψη η τιμή της μεταβλητής PLHQ που αντιπροσωπεύει το πλήθος των πλευρών του πολυγώνου αυξάνει κατά μια μονάδα.

```
ΤΟ ΕΞΕΛΙΞΗ :ΒΗΜΑΤΑ :DX :DY :PLEYRA :PLHQOS  
ΚΡΥΨΟΥ  
REPEAT :ΒΗΜΑΤΑ  
  [ΒΗΜΑ_ΕΜΨΥΧΟΠΟΙΗΣΗΣ :DX :DY :PLEYRA :PLHQOS  
  MAKE "PLHQOS :PLHQOS + 1]  
ΒΓΕΣ  
END
```



Το διαστημικό πρόγραμμα της logo

Με την τεχνογνωσία που έχουμε αποκτήσει, μπορούμε να αναπτύξουμε το δικό μας διαστημικό πρόγραμμα. Πρώτα ας κατασκευάσουμε τον πύραυλο.

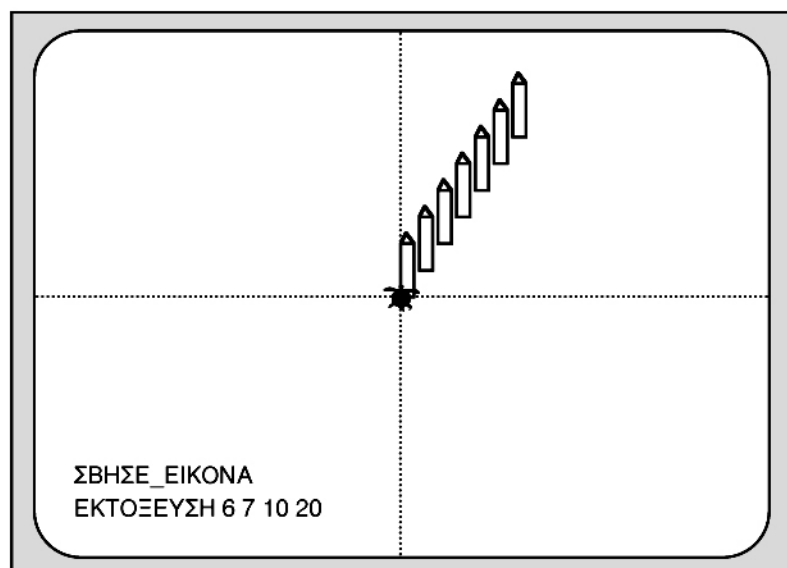
```
ΤΟ ΠΥΡΑΥΛΟΣ :MEGEQOS  
ΟΡΘΟΓΩΝΙΟ :MEGEQOS 4  
ΕΥΘΕΙΑ :MEGEQOS  
ΠΟΛΥΓΩΝΟ :MEGEQOS / 4 3  
ΟΠΙΣΘΕΝ :MEGEQOS  
END
```

Ας δοκιμάσουμε να μετακινήσουμε τον πύραυλο ένα βήμα.

```
ΤΟ ΒΗΜΑ_ΠΥΡΑΥΛΟΥ :DX :DY :MEGEQOS  
ΠΥΡΑΥΛΟΣ :MEGEQOS  
ΣΒΗΣΕ  
ΠΥΡΑΥΛΟΣ :MEGEQOS  
ΜΕΤΑΤΟΠΙΣΗ :DX :DY  
ΓΡΑΨΕ  
ΠΥΡΑΥΛΟΣ :MEGEQOS  
END
```

Τέλος, ας προγραμματίσουμε την εκτοξεύση του πυραύλου μας.

```
ΤΟ ΕΚΤΟΞΕΥΣΗ :ΒΗΜΑΤΑ :DX :DY :MEGEQOS  
ΚΡΥΨΟΥ  
REPEAT :ΒΗΜΑΤΑ [ΒΗΜΑ_ΠΥΡΑΥΛΟΥ :DX :DY :MEGEQOS]  
ΒΓΕΣ  
END
```



Ο χώρος του διαστημικού κέντρου

Ας σχεδιάσουμε τώρα το χώρο από όπου θα γίνεται η εκτόξευση. Ας ξεκινήσουμε από το έδαφος με την ομώνυμη διαδικασία.

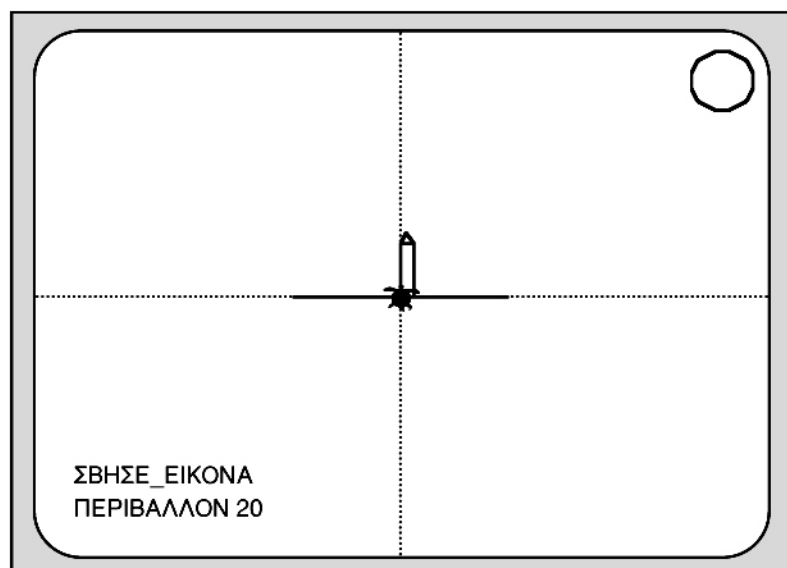
```
ΤΟ ΕΔΑΦΟΣ  
ΜΕΤΑΤΟΠΙΣΗ MINUS 40 0  
ΓΡΑΜΜΗ 90 80  
ΑΡΙΣΤΕΡΑ 90  
ΜΕΤΑΤΟΠΙΣΗ MINUS 40 0  
END
```

Ας σχεδιάσουμε τώρα τη σελήνη προς την οποία θα κατευθύνεται ο πυραύλος.

```
ΤΟ ΣΕΛΗΝΗ  
ΜΕΤΑΤΟΠΙΣΗ 120 70  
ΚΥΚΛΟΣ 2 1  
ΜΕΤΑΤΟΠΙΣΗ MINUS 120 MINUS 70  
END
```

Έχοντας ορίσει τις διαδικασίες ΕΔΑΦΟΣ και ΣΕΛΗΝΗ, τις συνδυάζουμε και σχεδιάζουμε το περιβάλλον όπου θα λάβει μέρος το διαστημικό πρόγραμμα.

```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS  
ΕΔΑΦΟΣ  
ΣΕΛΗΝΗ  
ΠΥΡΑΥΛΟΣ :MEGEQOS  
END
```

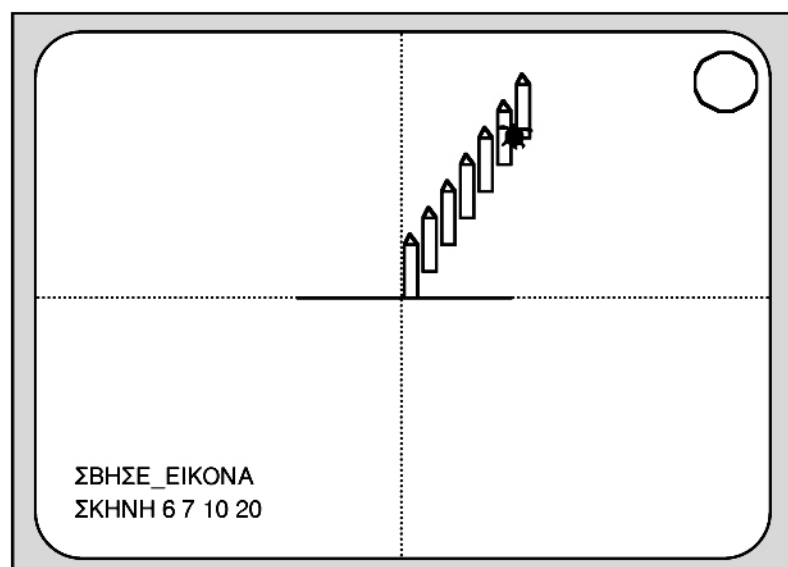


Δοκιμές της εκτόξευσης

Αφού έχουμε τη διαδικασία που περιγράφει την εκτόξευση του πυραύλου και αυτήν που σχεδιάζει το περιβάλλον από όπου θα γίνει η εκτόξευση, ας γράψουμε τη διαδικασία που θα παρουσιάζει τη σκηνή της εκτόξευσης.

```
ΤΟ ΣΚΗΝΗ :ΒΗΜΑΤΑ :DX :DY :MEGEQOS  
ΚΡΥΨΟΥ  
ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS  
ΕΚΤΟΞΕΥΣΗ :ΒΗΜΑΤΑ :DX :DY :MEGEQOS  
ΒΓΕΣ  
END
```

Η μεταβλητή MEGEQOS καθορίζει το μέγεθος του πυραύλου, οι μεταβλητές DX και DY καθορίζουν την οριζόντια και την κατακόρυφη μετατόπιση του πυραύλου σε κάθε του βήμα και η μεταβλητή ΒΗΜΑΤΑ το πλήθος των βημάτων (καρό) που θα κάνει ο πύραυλος μέχρι να εξαντληθούν τα καύσιμα του και να σταματήσει.



Ένα ρεαλιστικότερο περιβάλλον

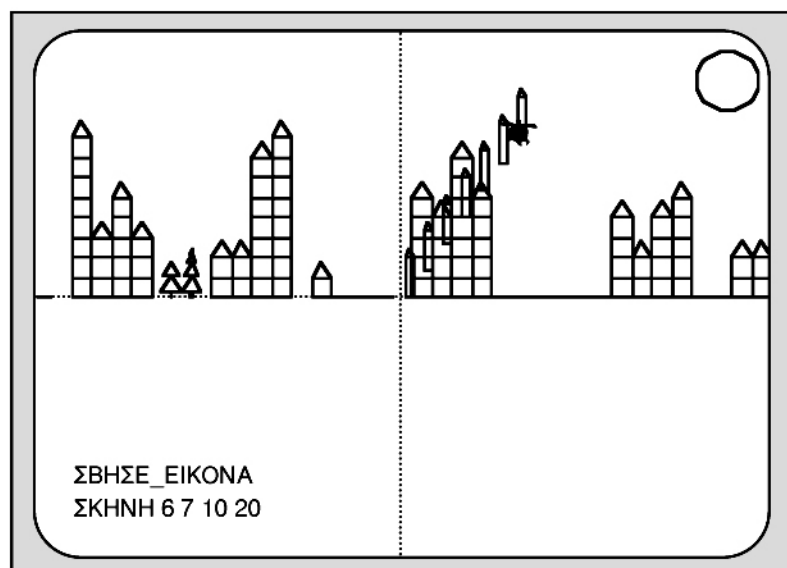
Ας σχεδιάσουμε ένα περιβάλλον με κτίρια και δένδρα.

```
ΤΟ ΕΔΑΦΟΣ :ΜΕΓΕΘΟΣ  
ΜΕΤΑΤΟΠΙΣΗ ΜΙΝΟΣ 200 0  
REPEAT 5  
  [MAKE "ΕΠΙΛΟΓΗ RANDOM 3  
  IFELSE :ΕΠΙΛΟΓΗ = 0  
    [MAKE "ΣΠΙΤΙΑ 4 MAKE "ΟΡΙΟ 4  
    [ΠΟΛΗ :ΜΕΓΕΘΟΣ / 2 :ΣΠΙΤΙΑ :ΟΡΙΟ]  
    [ΔΑΣΟΣ :ΜΕΓΕΘΟΣ / 2]  
  ]  
ΑΦΕΤΗΡΙΑ  
END
```

Μετά το σχεδιασμό του περιβάλλοντος η χελώνα θα πρέπει να επιστρέφει στην αφετηρία της. Αυτό γίνεται με την ΑΦΕΤΗΡΙΑ που χρησιμοποιεί την εντολή HOME (σπίτι). Η εντολή αυτή επιστρέφει τη χελώνα στο σημείο αφετηρίας, στο κέντρο της οθόνης.

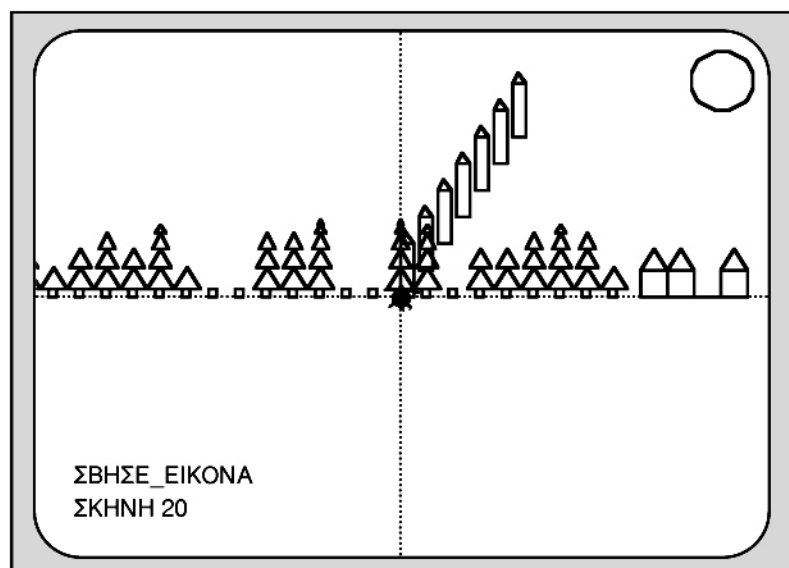
```
ΤΟ ΑΦΕΤΗΡΙΑ  
ΜΗ ΓΡΑΦΕΙΣ  
HOME  
ΓΡΑΨΕ  
END
```

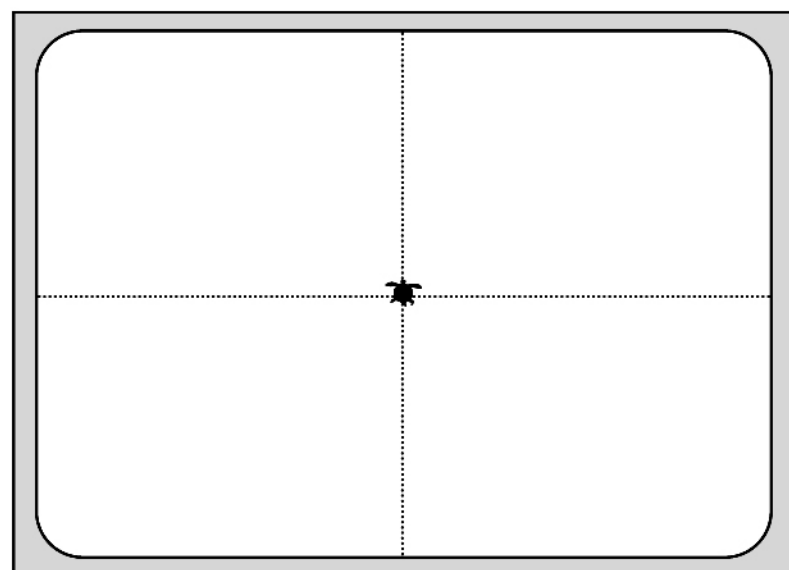
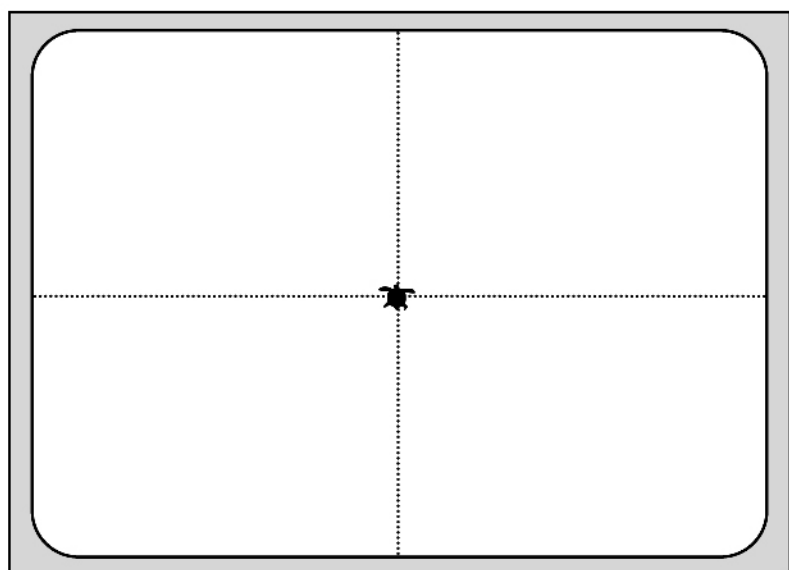
```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ  
ΕΔΑΦΟΣ :ΜΕΓΕΘΟΣ  
ΣΕΛΗΝΗ  
ΠΥΡΑΥΛΟΣ :ΜΕΓΕΘΟΣ  
END
```



Το διαστημικό πρόγραμμα πέτυχε!

Ας φτιάξουμε μια διαδικασία που να επιτρέπει να δοκιμάσουμε να εκτοξεύσουμε τον πύραυλο τρεις φορές, δίνοντας από το πληκτρολόγιο τιμές για το μέγεθος του βήματος, καθώς και το πλήθος των βημάτων.

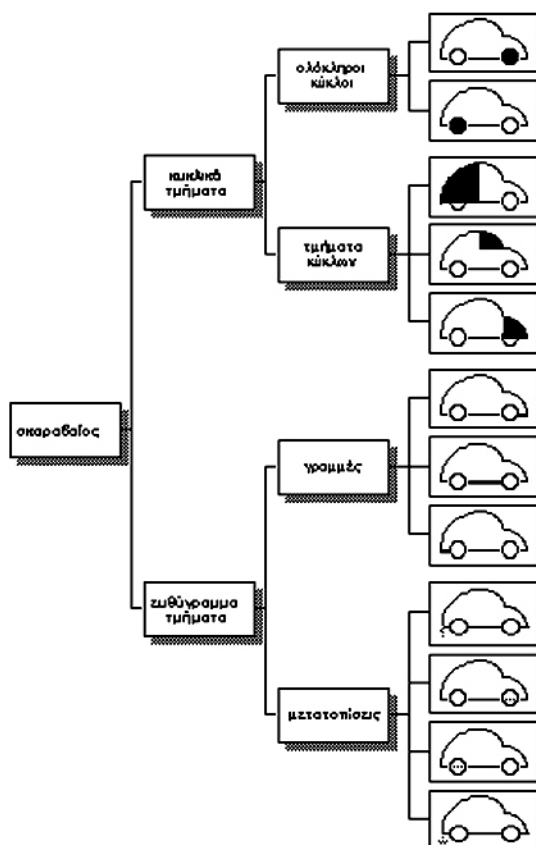




11 Ιεραρχικός σχεδιασμός

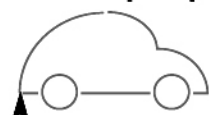
Οπτικός πίνακας περιεχομένων (Ο.Πι.Πε.)

Σε έναν οπτικό πίνακα περιεχομένων (Ο.Πι.Πε.) προσπαθούμε να απεικονίσουμε ομαδοποιημένα τα στοιχειώδη δομικά υλικά από τα οποία αποτελείται το σχήμα που θέλουμε να υλοποιήσουμε. Η σχεδίαση ενός Ο.Πι.Πε. είναι μια καθαρά αναλυτική νοητική διεργασία.



Στο διπλανό Ο.Πι.Πε. απεικονίζονται τα δομικά υλικά από τα οποία αποτελείται ο σκαρφαβίος. Αναλύοντας το σχήμα του σκαρφαβιού σε ένα πρώτο επίπεδο, διαπιστώνουμε ότι αποτελείται από κυκλικά και ευθύγραμμα τμήματα. Έτσι ξέροντας να φτιάχνουμε κύκλους και ευθύγραμμα τμήματα, έχουμε τα απαραίτητα υλικά για να δομήσουμε το σκαρφαβίο. Προχωρώντας σε δεύτερο επίπεδο, παρατηρούμε ότι τα κυκλικά τμήματα είναι είτε ολόκληροι κύκλοι είτε τεταρτοκύκλια. Ανάλογα, τα ευθύγραμμα τμήματα είναι είτε γραμμές είτε μετατοπίσεις. Άρα "παίζοντας" με τις παραμέτρους του κλάσματος του κύκλου και με το γράψε-μη γράφεις μπορούμε να έχουμε τα επιθυμητά σχήματα. Σε ένα τρίτο επίπεδο π.χ. μεταξύ των τεταρτοκυκλίων, διαπιστώνουμε ότι οι διαφορές εντοπίζονται στο μέγεθος και τον προσανατολισμό τους.

Η σωστή σειρά των πραγμάτων



Είδαμε στον Ο.Πι.Πε. τα τμήματα από τα οποία αποτελείται ο σκαραβαίος. Για να σχεδιαστεί όμως θα πρέπει αυτά να γίνουν με μια συγκεκριμένη σειρά. Ας ξεκινήσουμε με τη **SXET_METATOPISH 0 5**.



Αφού η χελώνα έφτασε στο κάτω αριστερά σημείο θα πρέπει να σχεδιάσουμε το πίσω τεταρτοκύκλιο. Αυτό γίνεται με τις εντολές **ΔΕΞΙΑ 90** και **ΚΥΚΛΟΣ 10 1/4**.



Στη συνέχεια πρέπει να σχεδιαστεί το τεταρτοκύκλιο που αντιπροσωπεύει το μπροστινό μέρος της σκεπής του σκαραβαίου. Αυτό θα γίνει με την εντολή **ΚΥΚΛΟΣ 4 1/4**.



Το εμπρός μέρος του αυτοκινήτου είναι και αυτό ένα τεταρτοκύκλιο που θα σχεδιαστεί αφού προσανατολιστεί η χελώνα με τις εντολές **ΑΡΙΣΤΕΡΑ 90** και **ΚΥΚΛΟΣ 4 1/4**.



Το επόμενο βήμα είναι να σχεδιάσουμε το ευθύγραμμο κάτω μέρος μέχρι τη μπροστινή πόρτα. Αυτό θα γίνει με την εντολή **ΕΥΘΕΙΑ 15**.



Η μπροστινή ρόδα είναι ένας πλήρης κύκλος και η δημιουργία της είναι εύκολη με την εντολή **ΚΥΚΛΟΣ 1.5 1**.



Με τον πλήρη κύκλο η χελώνα έχει επιστρέψει στο δεξιά σημείο από όπου ξεκίνησε. Θα πρέπει να μετατοπισθεί παρακάμπτοντας τη ρόδα. Γίνεται με την εντολή **SXET_METATOPISH 0 18**.



Το κάτω μέρος του σκαραβαίου μεταξύ των δύο τροχών γίνεται με την εντολή **ΕΥΘΕΙΑ 42**.



Ο πίσω τροχός είναι ένας πλήρης κύκλος και η δημιουργία του είναι εύκολη με την εντολή **ΚΥΚΛΟΣ 1.5 1**.



Με τον πλήρη κύκλο η χελώνα έχει επιστρέψει στο δεξιά σημείο από όπου ξεκίνησε. Θα πρέπει να μετατοπισθεί παρακάμπτοντας τη ρόδα. Γίνεται με την εντολή **SXET_METATOPISH 0 18**.



Απομένει το κάτω μέρος του αυτοκινήτου, από τον τροχό μέχρι το πίσω μέρος. Αυτό γίνεται με την εντολή **10**.

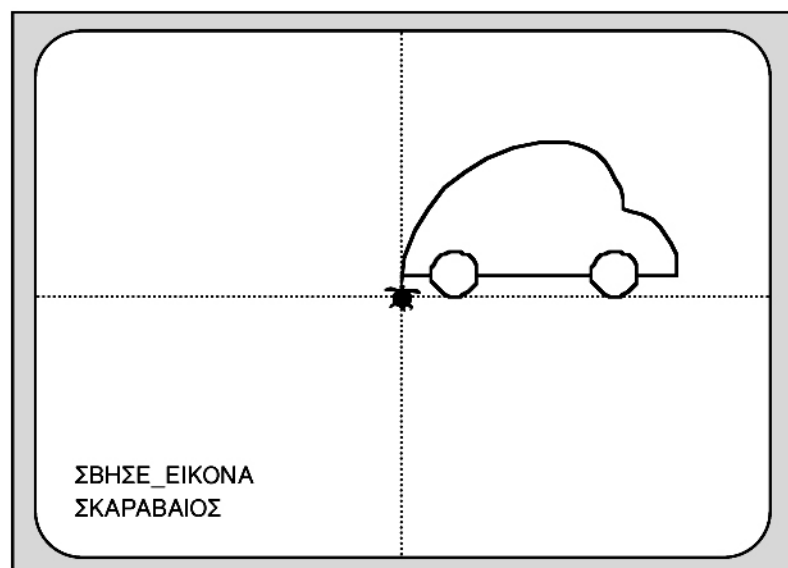


Τέλος πρέπει η χελώνα αφού προσανατολιστεί με την εντολή **ΔΕΞΙΑ 90** να επιστρέψει στην αφετηρία της με την εντολή **SXET_METATOPISH 0 -7**.

Ο σκαρabaίος ως διαδικασία

Οι προηγούμενες εντολές δημιουργούν τη διαδικασία ΣΚΑΡΑΒΑΙΟΣ.

```
ΤΟ ΣΚΑΡΑΒΑΙΟΣ  
ΜΕΤΑΤΟΠΙΣΗ 0 5  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ 10 1 / 4  
ΚΥΚΛΟΣ 4 1 / 4  
ΑΡΙΣΤΕΡΑ 90  
ΚΥΚΛΟΣ 4 1 / 4  
ΕΥΘΕΙΑ 15  
ΚΥΚΛΟΣ 1.5 1  
ΜΕΤΑΤΟΠΙΣΗ 0 18  
ΕΥΘΕΙΑ 42  
ΚΥΚΛΟΣ 1.5 1  
ΜΕΤΑΤΟΠΙΣΗ 0 18  
ΕΥΘΕΙΑ 10  
ΔΕΞΙΑ 90  
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS 7  
END
```



Παραμετροποιημένος σκαραβαίος

Η προηγούμενη διαδικασία σχεδιάζει ένα σκαραβαίο σταθερού μεγέθους. Μπορούμε χρησιμοποιώντας την παράμετρο YPSOS να μεταβάλουμε το μέγεθος του σκαραβαίου.

Προϋπόθεση είναι να διατηρήσουμε στη νέα διαδικασία τις αναλογίες μεταξύ των τιμών με τις οποίες καλούνται να εκτελεστούν όλες οι διαδικασίες που χρησιμοποιούνται από την κύρια διαδικασία.

ΤΟ ΣΚΑΡΑΒΑΙΟΣ :YPSOS

ΜΕΤΑΤΟΠΙΣΗ :YPSOS 5

ΔΕΞΙΑ 90

ΚΥΚΛΟΣ :YPSOS 1 / 4

ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4

ΑΡΙΣΤΕΡΑ 90

ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4

ΕΥΘΕΙΑ :YPSOS * 1.5

ΚΥΚΛΟΣ :YPSOS * 0.15 1

ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8

ΕΥΘΕΙΑ :YPSOS * 4.2

ΚΥΚΛΟΣ :YPSOS * 0.15 1

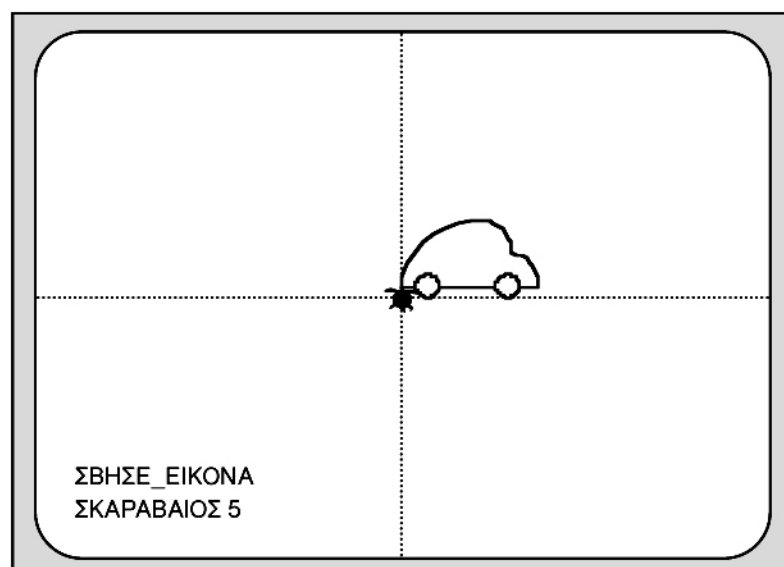
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8

ΕΥΘΕΙΑ :YPSOS * 1.0

ΔΕΞΙΑ 90

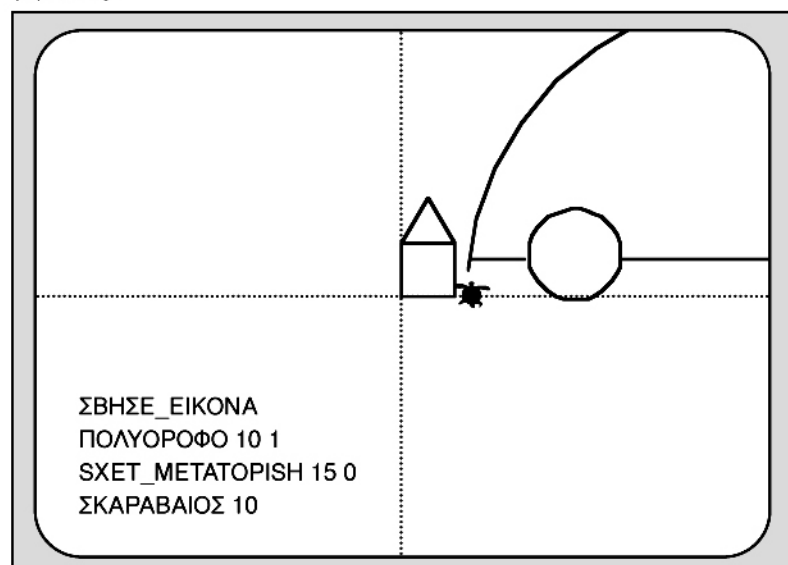
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:YPSOS * 0.7)

END

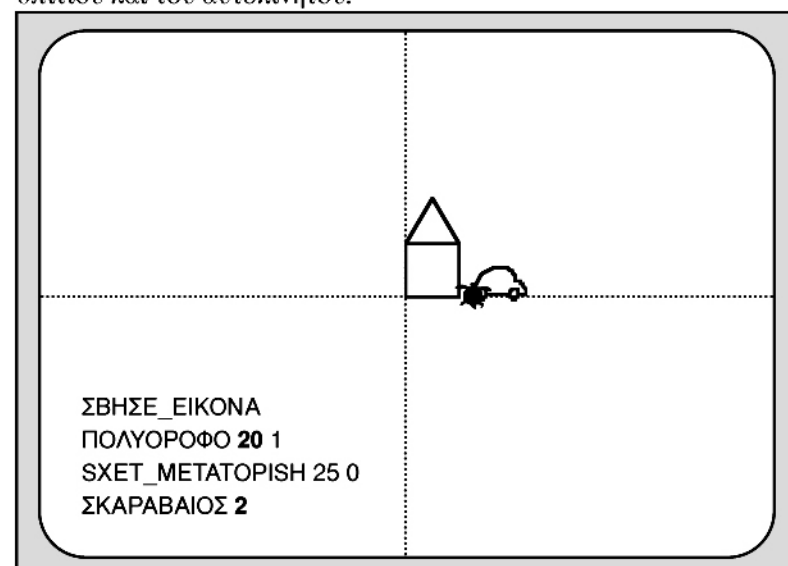


Ένας σκαραβαίος γίγαντας

Σχεδιάζοντας ένα σπίτι και ένα σκαραβαίο δίνοντας την ίδια τιμή στην παράμετρο που καθορίζει το μέγεθός τους, θα πάρουμε το παρακάτω σχήμα. Προφανώς το σκαραβαίο τον οδηγεί κάποιος γίγαντας.



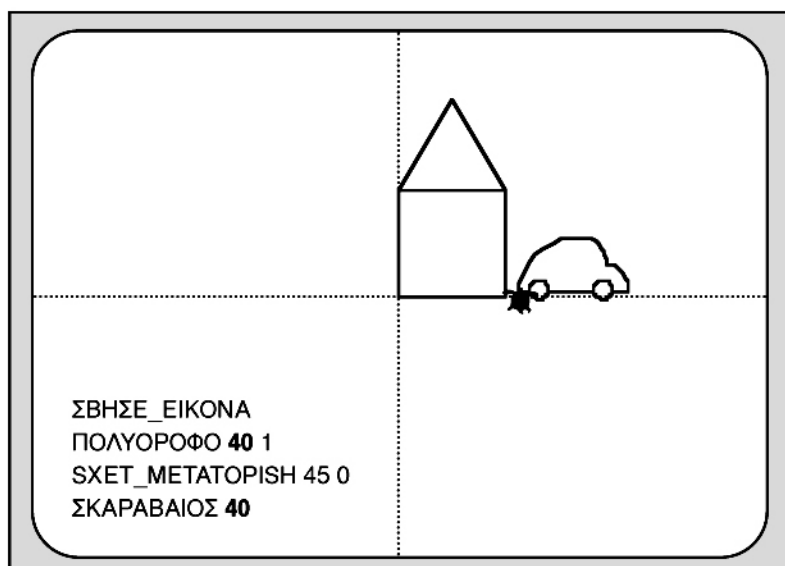
Μπορούμε με μερικές προσπάθειες κρατώντας σταθερό το μέγεθος του ενός σχήματος και μεταβάλλοντας την τιμή του μεγέθους του άλλου, να πετύχουμε μια παραδεκτή αναλογία μεγέθους μεταξύ του σπιτιού και του αυτοκινήτου.



“Κανονικοποίηση” του σκαραβαίου

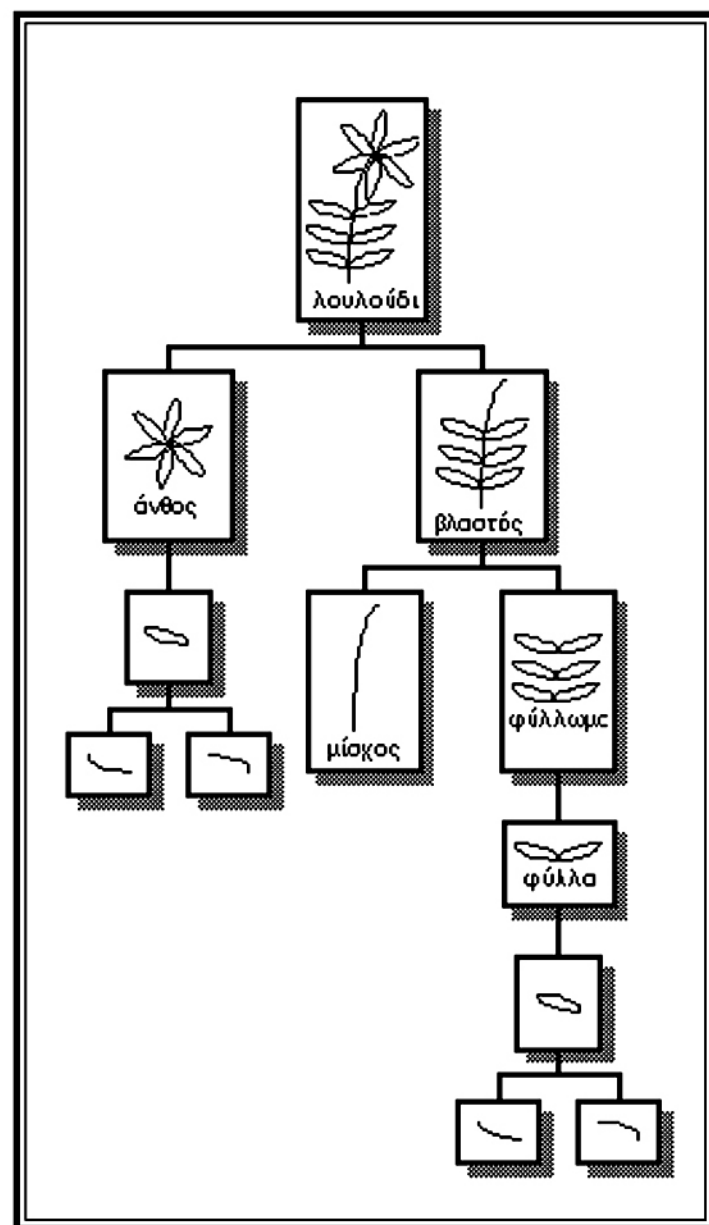
Θα πρέπει την αναλογία μεταξύ σπιτιού-περιβάλλοντος και αυτοκινήτου να την ενσωματώσουμε στον κώδικα της διαδικασίας του σκαραβαίου. Είδαμε ότι το μέγεθος του σκαραβαίου αντιστοιχεί στο 1/10 της τιμής του μεγέθους του σπιτιού. Αυτήν την τιμή που κανονικοποιεί τις δύο διαδικασίες θα εισάγουμε στον κώδικα της διαδικασίας ΣΚΑΡΑΒΑΙΟΣ.

```
ΤΟ ΣΚΑΡΑΒΑΙΟΣ :ΜΕΓΕΘΟΣ  
MAKE "YPSOS (:ΜΕΓΕΘΟΣ / 10)  
ΜΕΤΑΤΟΠΙΣΗ :YPSOS 5  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ :YPSOS 1 / 4  
ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4  
ΑΡΙΣΤΕΡΑ 90  
ΚΥΚΛΟΣ :YPSOS * 0.4 1 / 4  
ΕΥΘΕΙΑ :YPSOS * 1.5  
ΚΥΚΛΟΣ :YPSOS * 0.15 1  
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8  
ΕΥΘΕΙΑ :YPSOS * 4.2  
ΚΥΚΛΟΣ :YPSOS * 0.15 1  
ΜΕΤΑΤΟΠΙΣΗ 0 :YPSOS * 1.8  
ΕΥΘΕΙΑ :YPSOS * 1.0  
ΔΕΞΙΑ 90  
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:YPSOS * 0.7)  
END
```



Αναλύοντας τα μέρη ενός λουλουδιού

Για να σχεδιάσουμε το λουλούδι θα ακολουθήσουμε την ίδια μέθοδο που ακολουθήσαμε και στο σκαραβαίο. Θα αναλύσουμε το λουλούδι στα βασικά του τμήματα και στη συνέχεια θα τα αναλύσουμε περαιτέρω. Αυτό θα συνεχιστεί μέχρι να φτάσουμε σε στοιχειώδη μέρη που μπορούν να σχεδιαστούν εύκολα.



Στο διπλανό σχήμα φαίνεται ο Οπτικός Πίνακας Περιεχομένων όπου απεικονίζονται τα δομικά υλικά από τα οποία αποτελείται το λουλούδι.

Σε ένα πρώτο επίπεδο το λουλούδι αποτελείται από το άνθος και το βλαστό.

Αναλύοντας το βλαστό σε ένα δεύτερο επίπεδο τον χωρίζουμε στο μίσχο και το φύλλωμα. Ο μίσχος είναι ένα τόξο (τμήμα περιφέρειας κύκλου).

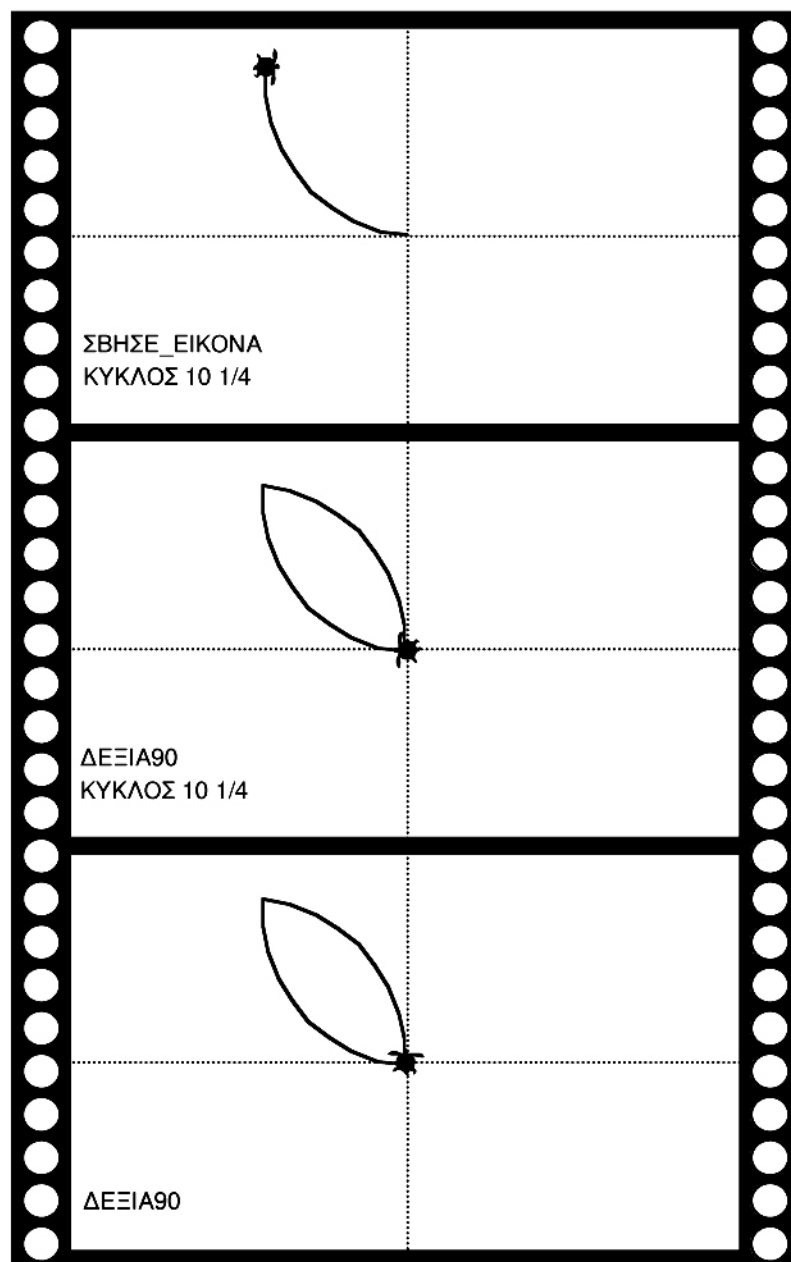
Σε τρίτο επίπεδο το φύλλωμα αποτελείται από τα φύλλα.

Στο προτελευταίο επίπεδο τα φύλλα αλλά και το άνθος αποτελούνται από το "φύλλο" ή "πέταλο".

Τέλος το "φύλλο" ή "πέταλο" σχηματίζεται από δύο τμήματα κύκλου.

Δημιουργώντας το φύλλο ή πέταλο

Σχεδιάζουμε το φύλλο, ένα από τα στοιχειώδη σχήματα του λουλουδιού.



Σχεδιάζουμε ένα τεταρτοκύκλιο.

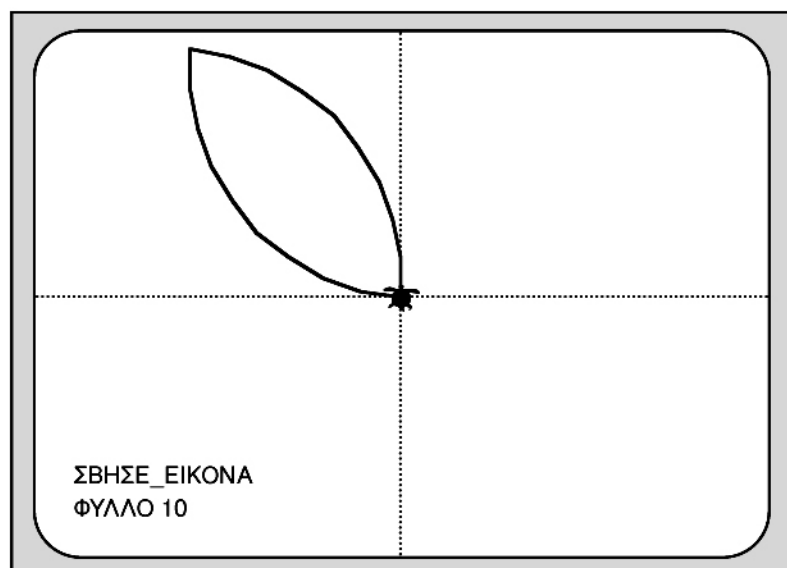
Στη συνέχεια στρίβοντας τη χελώνα 90 μοίρες δεξιά σχεδιάζουμε ένα ακόμα τεταρτοκύκλιο. Η χελώνα έχει επιστρέψει στην αφετηρία.

Τέλος, επαναφέρουμε τη χελώνα στην αρχική της κατάσταση.

Το φύλλο ως διαδικασία

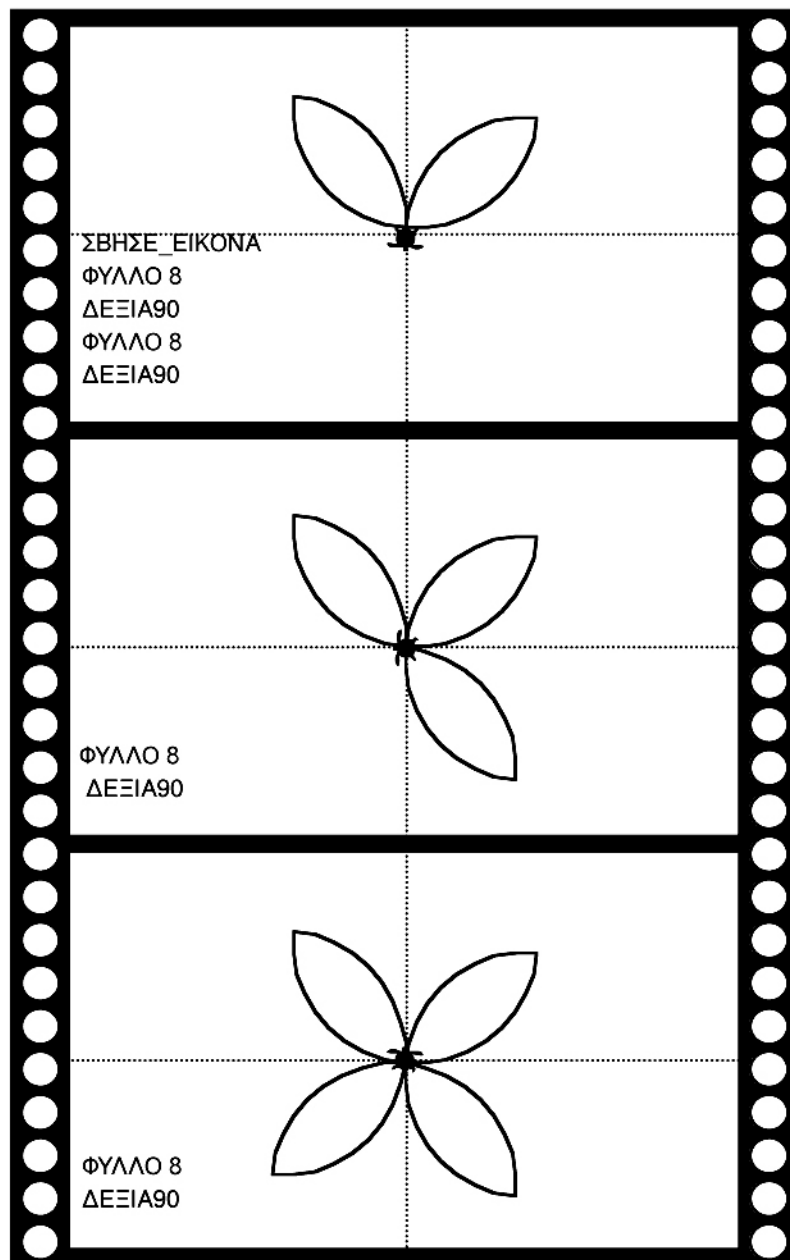
Οι προηγούμενες εντολές ολοκληρώνονται στη διαδικασία ΦΥΛΛΟ που θα σχεδιάζει το φύλλο του βλαστού και το πέταλο του άνθους του λουλουδιού.

```
ΤΟ ΦΥΛΛΟ :MEGEQOS  
ΚΥΚΛΟΣ :MEGEQOS 1 / 4  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ :MEGEQOS 1 / 4  
ΔΕΞΙΑ 90  
END
```



Ένα άνθος με τέσσερα πέταλα

Περιστρέφοντας το φύλλο/πέταλο 90 μοίρες κάθε φορά, σχηματίζουμε ένα άνθος.



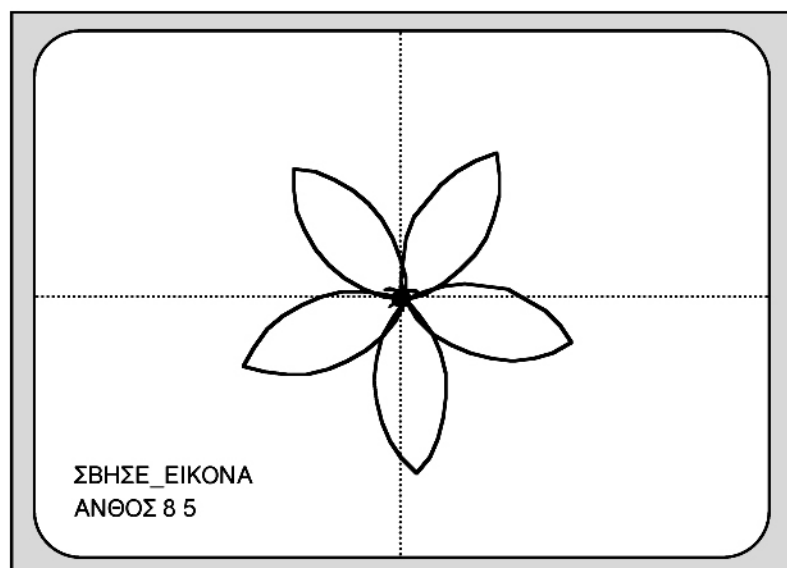
Σχεδιάζοντας το άνθος

Η διαδικασία ΑΝΘΟΣ ολοκληρώνεται ως εξής:

```
TO ΑΝΘΟΣ :ΜΕΓΕΘΟΣ  
REPEAT 4  
  [ΦΥΛΛΟ :ΜΕΓΕΘΟΣ ΔΕΞΙΑ 90]  
END
```

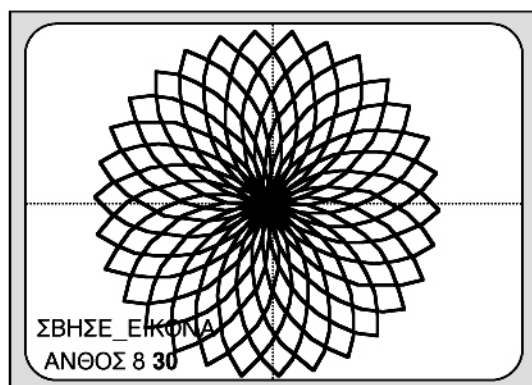
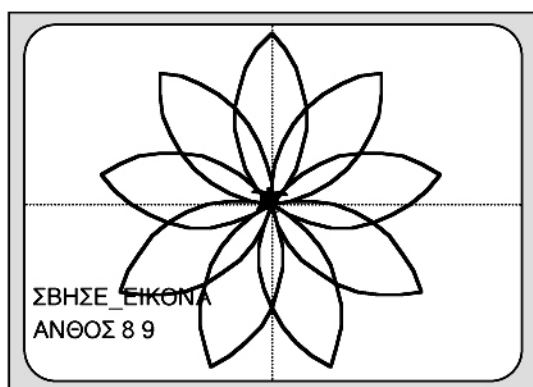
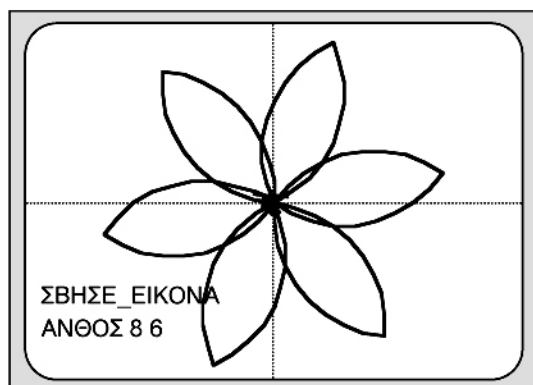
Μπορούμε να γενικεύσουμε την διαδικασία ΑΝΘΟΣ έτσι ώστε να καθορίζεται έξω από αυτήν το πλήθος των πετάλων που θα έχει το άνθος. Η στροφή που θα κάνει το φύλλο κάθε φορά δε θα είναι πλέον 90 μοίρες αλλά υπολογίζεται από τη διαίρεση του 360 διά το πλήθος των πετάλων.

```
TO ΑΝΘΟΣ :ΜΕΓΕΘΟΣ :ΡΕΤΑΛΑ  
MAKE "ΓΩΝΙΑ (360 / :ΡΕΤΑΛΑ)  
REPEAT :ΡΕΤΑΛΑ  
  [ΦΥΛΛΟ :ΜΕΓΕΘΟΣ  
    ΔΕΞΙΑ :ΓΩΝΙΑ]  
END
```



Σχεδιάζοντας άνθη

Ας βάλουμε τη χελώνα να σχεδιάσει μερικά άνθη με διαφορετικό πλήθος πετάλων.



Ένα άνθος με 30
ροδοπέταλα θα
μπορούσαμε να
το ονομάζουμε
τριαντάφυλλο.

Τμήμα μίσχου του βλαστού του λουλουδιού

Θα μπορούσαμε να σχεδιάσουμε ολόκληρο το μίσχο αλλά μετά θα έπρεπε να οπισθοχωρούμε και να "κολλάμε" σε διάφορα σημεία του τα φύλλα. Γι' αυτόν το λόγο θα ακολουθήσουμε μια διαφορετική προσέγγιση. Κάθε φορά, σχεδιάζουμε ένα τμήμα του μίσχου μέχρι το σημείο στο οποίο θα πρέπει να προστεθούν τα φύλλα.

Ας γράψουμε τη διαδικασία που θα σχεδιάζει το τμήμα του μίσχου. Βασικό στοιχείο του θα είναι το τόξο της περιφέρειας κύκλου που θα αντιστοιχεί σε μια μικρή γωνία και θα πρέπει να είναι σωστά προσανατολισμένο.

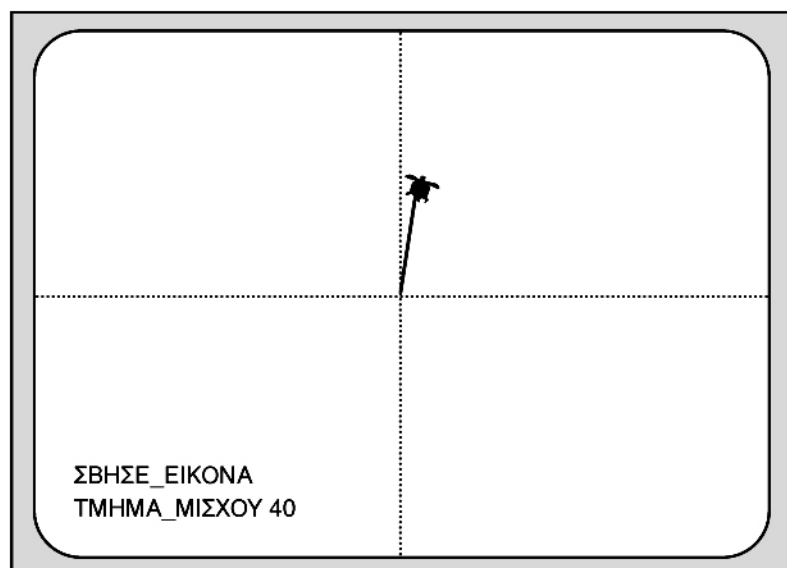
ΤΟ ΤΜΗΜΑ_ΜΙΣΧΟΥ :MEGEQOS

ΔΕΞΙΑ 90

ΚΥΚΛΟΣ :MEGEQOS 1 / 24

ΑΡΙΣΤΕΡΑ 90

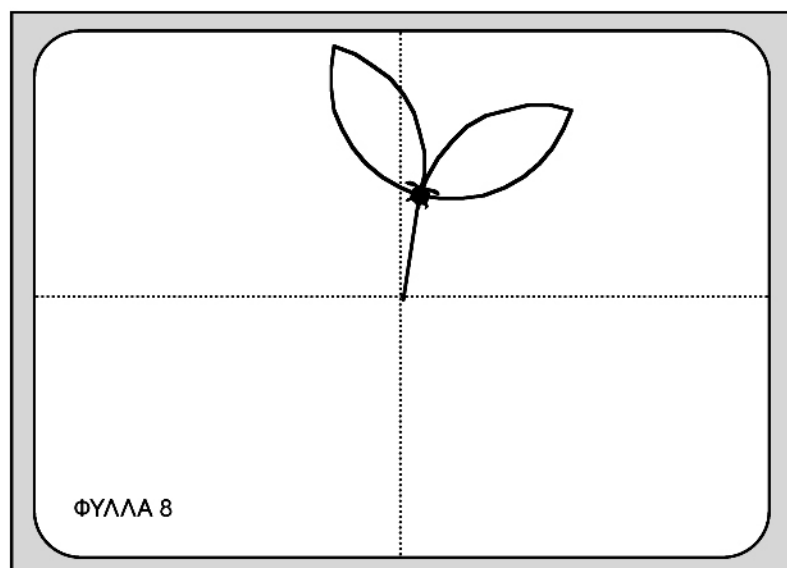
END



Τα φύλλα του βλαστού του λουλουδιού

Μετά το τμήμα του μίσχου προσθέτουμε τα δύο φύλλα. Ας γράψουμε τη διαδικασία που θα σχεδιάζει αυτά τα δύο φύλλα.

```
ΤΟ ΦΥΛΛΑ :MEGEQOS  
ΦΥΛΛΟ :MEGEQOS  
ΔΕΞΙΑ 90  
ΦΥΛΛΟ :MEGEQOS  
ΑΡΙΣΤΕΡΑ 90  
END
```



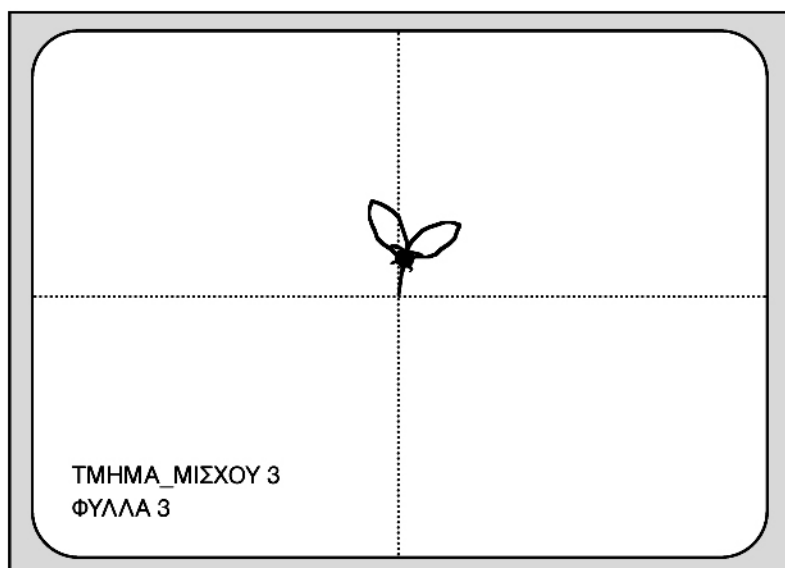
“Κανονικοποίηση” του τμήματος του μίσχου

Τα φύλλα σχεδιάζονται με τη διαδικασία ΦΥΛΛΑ 8, ενώ το τμήμα του μίσχου με τη ΤΜΗΜΑ_ΜΙΣΧΟΥ 40.

Θα πρέπει το μέγεθος του μίσχου να προσαρμοστεί έτσι ώστε να ταιριάζει με το μέγεθος των φύλλων.

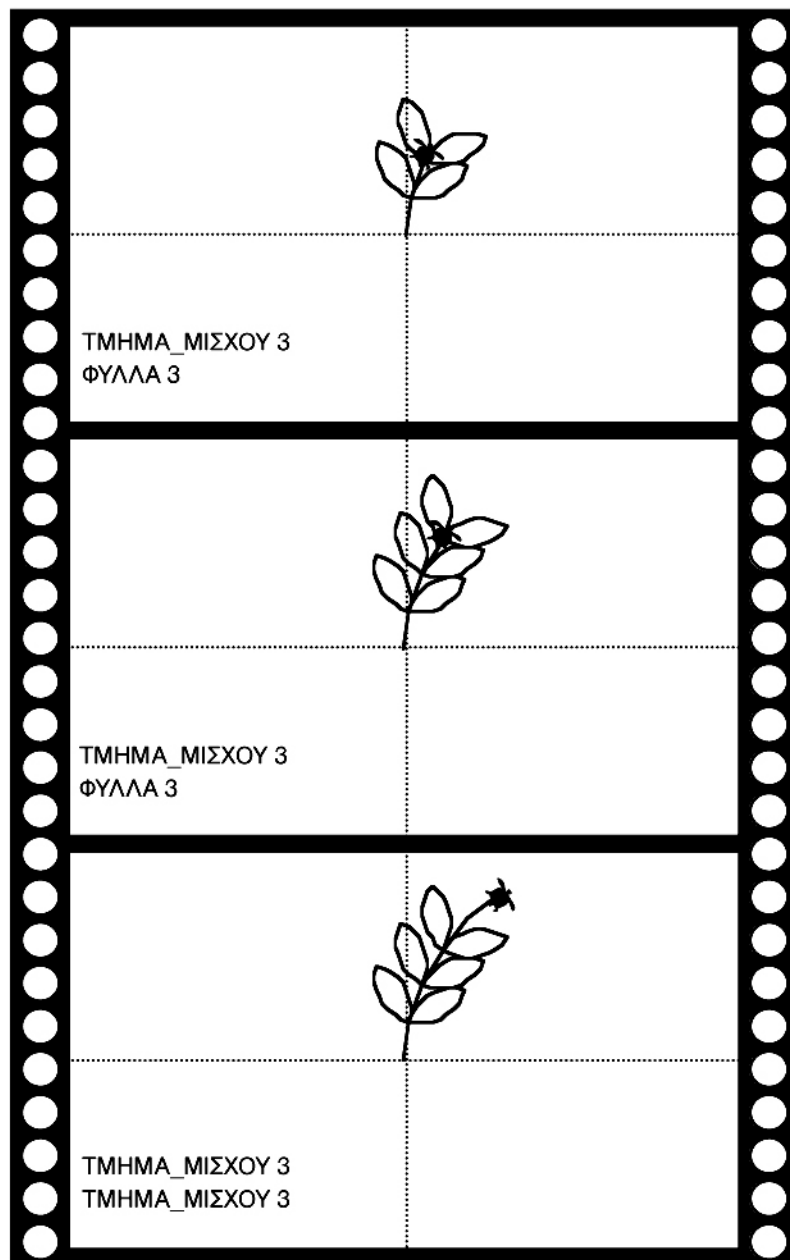
Συγκρίνοντας τα αντίστοιχα μεγέθη (8 και 40) διαπιστώνουμε ότι είτε θα πρέπει να πενταπλασιάσουμε το μέγεθος του τμήματος του μίσχου ή να υποπενταπλασιάσουμε το μέγεθος των φύλλων. Ας διαλέξουμε τη πρώτη επιλογή και ας την ενσωματώσουμε στη διαδικασία ΤΜΗΜΑ_ΜΙΣΧΟΥ.

```
TO ΤΜΗΜΑ_ΜΙΣΧΟΥ :ΜΕΓΕΘΟΣ  
MAKE "ΜΕΓΕΘΟΣ (:ΜΕΓΕΘΟΣ * 5)  
ΔΕΞΙΑ 90  
ΚΥΚΛΟΣ :ΜΕΓΕΘΟΣ 1 / 24  
ΑΡΙΣΤΕΡΑ 90  
END
```



Σχηματίζοντας το θλαστό

Σχεδιάζοντας βήμα προς βήμα τα τμήματα του μίσχου και τα φύλλα, σχηματίζεται ο βλαστός.



Η διαδικασία που σχεδιάζει το βλαστό

Η διαδικασία ΒΛΑΣΤΟΣ ενσωματώνει όλες τις προηγούμενες ενέργειες για να σχεδιασθεί ο βλαστός του λουλουδιού.

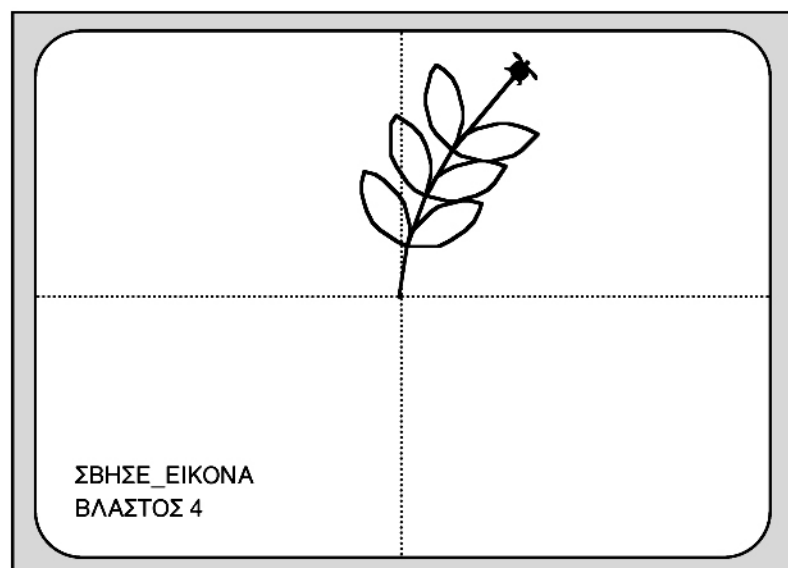
Στο επάνω μέρος του μίσχου βάλαμε δύο επιπλέον τμήματα μίσχου ώστε στο τέλος του να υπάρχει χώρος για να σχεδιαστεί το άνθος.

ΤΟ ΒΛΑΣΤΟΣ :MEGEQOS

REPEAT 3 [ΤΜΗΜΑ_ΜΙΣΧΟΥ :MEGEQOS ΦΥΛΛΑ :MEGEQOS]

ΤΜΗΜΑ_ΜΙΣΧΟΥ (:MEGEQOS + :MEGEQOS)

END



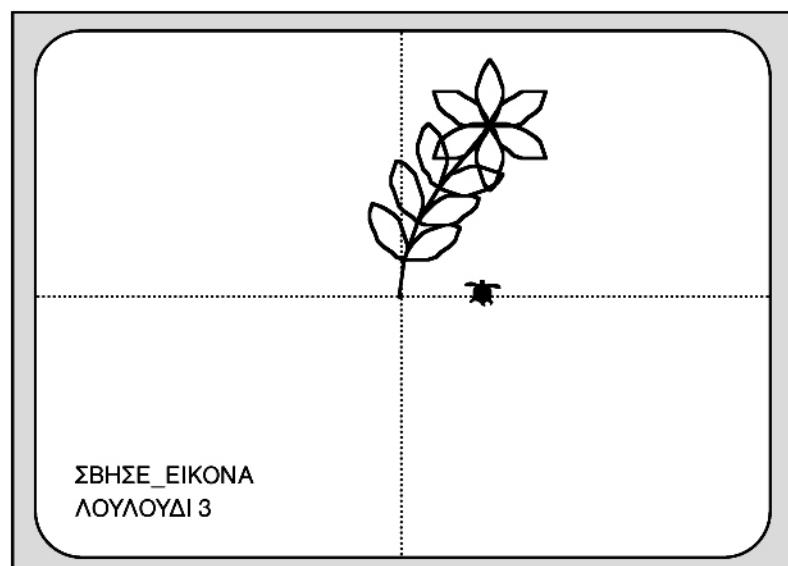
Ολόκληρο το λουλούδι

Αφού έχουμε δημιουργήσει ξεχωριστά όλες τις επιμέρους διαδικασίες από τις οποίες αποτελείται το λουλούδι, τις συναρμολογούμε σε μια που το σχεδιάζει.

Το πλήθος των πετάλων του άνθους θα παίρνει τυχαία τιμή κάθε φορά, με τη βοήθεια της συνάρτησης RANDOM. Εμείς όμως δε θέλουμε ένα μαδημένο άνθος με κανένα ή ένα-δύο πέταλα. Γι' αυτό θα προσθέτουμε τον αριθμό 3 στην τιμή που θα προκύπτει από τη συνάρτηση RANDOM έτσι ώστε η τιμή του πλήθους των πετάλων να μη μπορεί να είναι μικρότερη του 3 (περίπτωση που η συνάρτηση RANDOM επιστρέφει την τιμή 0).

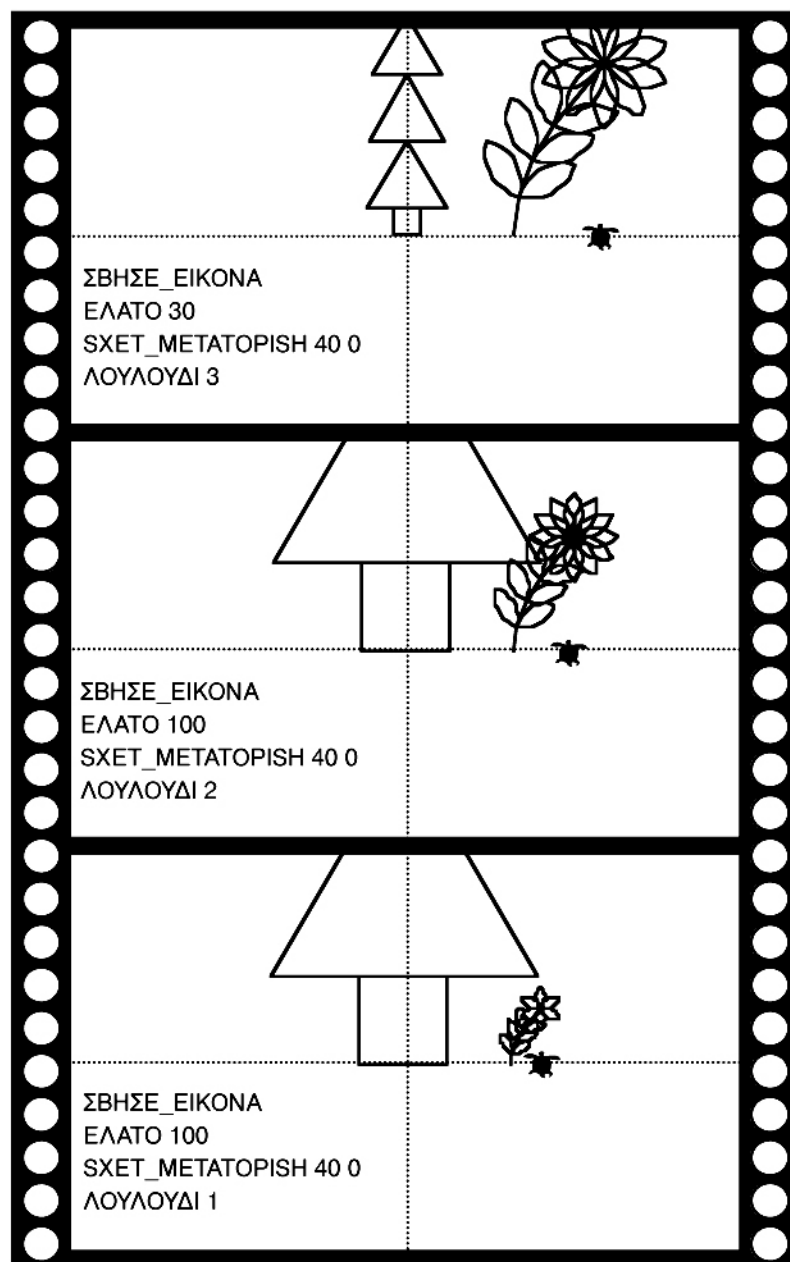
Θα πρέπει να λάβουμε πρόνοια ώστε η χελώνα μετά τη σχεδίαση του λουλουδιού, να παίρνει θέση ώστε να μπορεί στη συνέχεια να σχεδιαστεί κάτι άλλο στην κατάλληλη θέση.

```
ΤΟ ΛΟΥΛΟΥΔΙ :MEGEQOS  
MAKE "PETALA (3 + RANDOM 10)  
ΒΛΑΣΤΟΣ :MEGEQOS  
ΑΝΘΟΣ :MEGEQOS :PETALA  
ΜΗ_ΓΡΑΦΕΙΣ  
ΑΡΙΣΤΕΡΑ 40  
ΟΠΙΣΘΕΝ (21.5 * :MEGEQOS)  
ΓΡΑΨΕ  
END
```



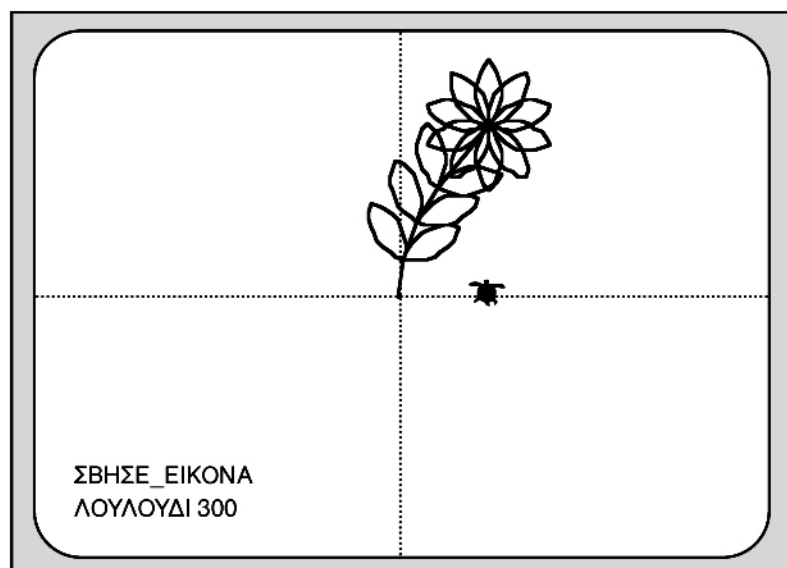
“Κανονικοποίηση” του λουλουδιού

Θα προσπαθήσουμε να βρούμε τη σωστή αναλογία μεταξύ του μεγέθους του λουλουδιού και ενός δένδρου.



Η διαδικασία που σχεδιάζει το λουλούδι

Πειραματιζόμενοι διαπιστώνουμε ότι το μέγεθος του λουλουδιού θα πρέπει να είναι το 1/100 του μεγέθους του δένδρου. Αυτή τη διαπίστωση θα την ενσωματώσουμε στη διαδικασία ΛΟΥΛΟΥΔΙ.



Σχετική και απόλυτη θέση στην οθόνη

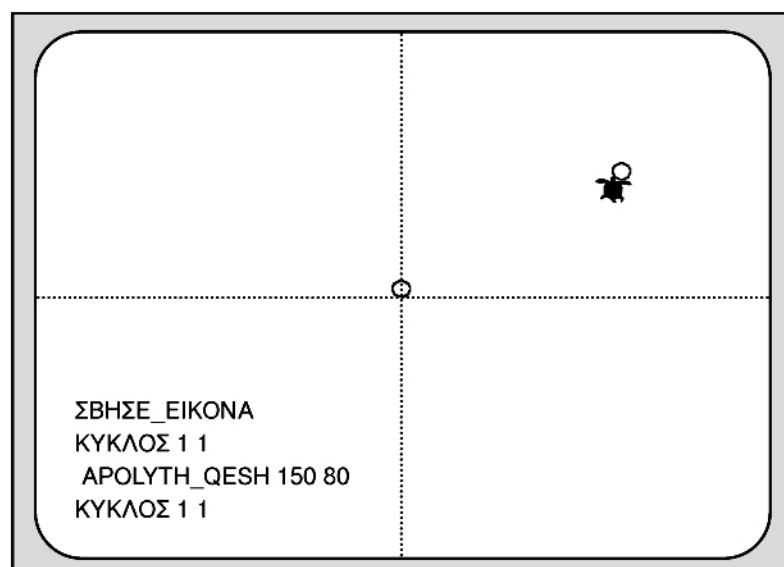
Για να μετακινηθεί η χελώνα από το σημείο που βρίσκεται σε κάποιο άλλο, πρέπει να ξέρουμε την απόσταση των δύο σημείων. Το σημείο που θα φτάσει η χελώνα καθορίζεται με βάση το σημείο από το οποίο ξεκινάει, δηλαδή η θέση του είναι σχετική προς το αρχικό σημείο.

Μπορούμε να θεωρήσουμε αυθαίρετα κάποιο συγκεκριμένο σημείο της οθόνης ως απόλυτο αρχικό σημείο και όλα τα σημεία να καθορίζονται με βάση αυτό. Έτσι το σημείο που βρίσκεται κάποια στιγμή η χελώνα αλλά και το σημείο που θα βρεθεί μια επόμενη στιγμή, εξαρτώνται από το απόλυτο αρχικό σημείο.

Με τη διαδικασία APOLYTH_QESH μετατοπίζουμε κατευθείαν τη χελώνα στο σημείο που θέλουμε δηλώνοντας μόνο τις συντεταγμένες της θέσης του.

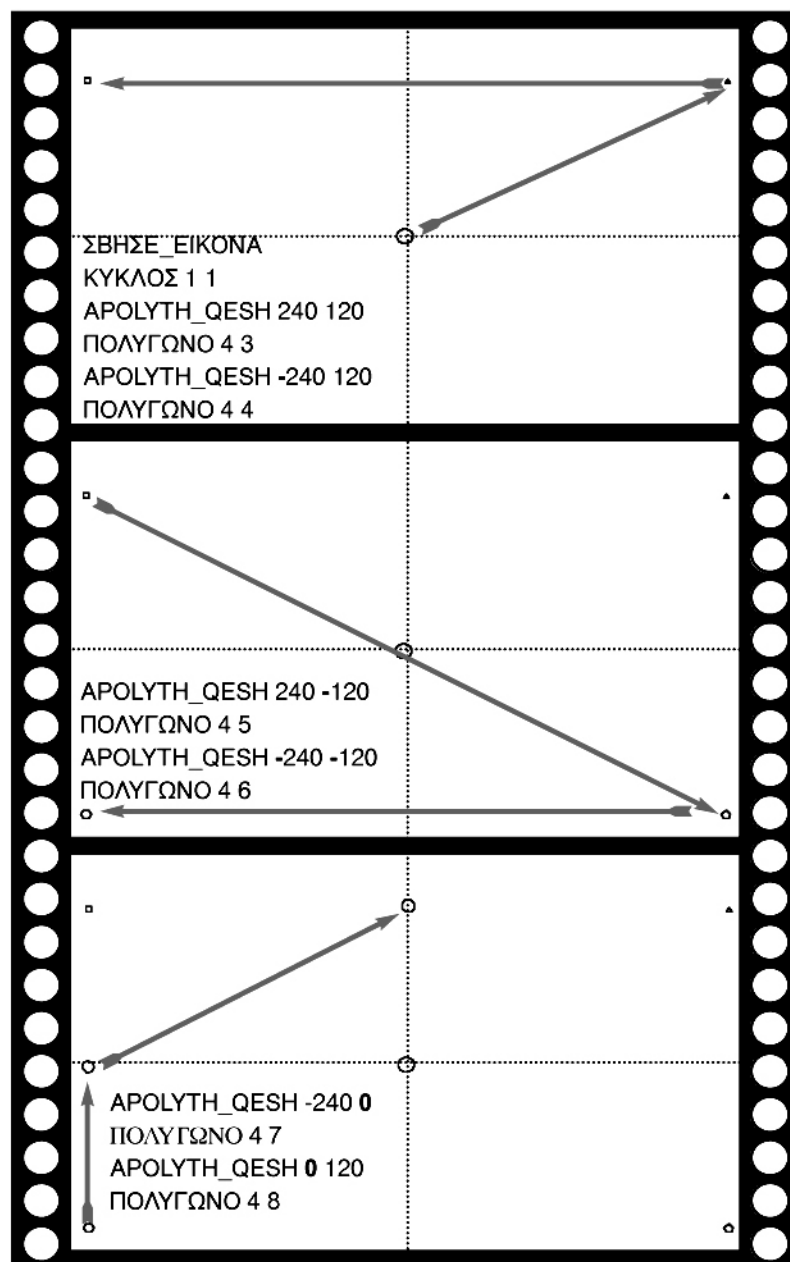
```
TO ΑΠΟΛΥΤΗ_ΘΕΣΗ :X :Y  
MH_ΓΡΑΦΕΙΣ  
SETPOS [:X :Y]  
ΓΡΑΨΕ  
END
```

Η διαδικασία APOLYTH_QESH χρησιμοποιεί την εντολή SETPOS που δέχεται δύο αριθμούς (την οριζόντια και τη κάθετη συντεταγμένη) και μετακινεί τη χελώνα στη θέση που έχει ως συντεταγμένες αυτούς τους αριθμούς.



Εξερευνώντας το όρια της οθόνης

Με περαματισμούς όπως αυτοί που ακολουθούν μπορούμε να εξερευνήσουμε τα όρια της οθόνης του υπολογιστή για τη χελώνα.



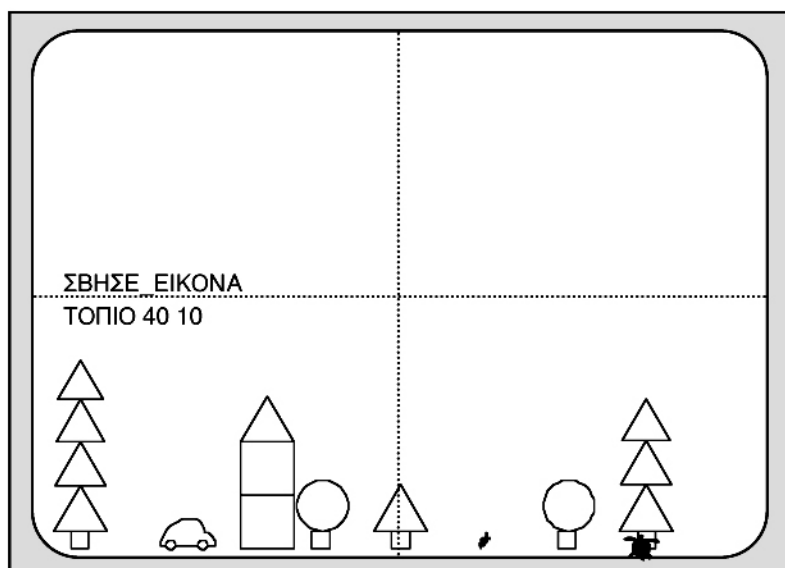
Πολλαπλές επιλογές

Ας φτιάξουμε μια διαδικασία που να σχεδιάζει ένα τοπίο με αυτοκίνητα, κτίρια, λουλούδια, ροδιές και έλατα επιλέγοντας στην τύχη τον εκάστοτε συνδυασμό τους. Η δομή IFELSE μας επιτρέπει να επιλέξουμε μεταξύ δύο δυνατοτήτων. Αν και η δεύτερη είναι μια εντολή IFELSE, τότε συνολικά οι επιλογές μας γίνονται τρεις. Αυτό μπορεί να συνεχιστεί και σε επόμενα στάδια.

```

ΤΟ ΤΟΠΙΟ :ΜΕΓΕΘΟΣ :ΦΟΡΕΣ
MAKE "ΑΡΧΗΧ MINUS 270
MAKE "ΑΡΧΗΥ MINUS 140
ΑΠΟΛΥΤΗ_ΘΕΣΗ :ΑΡΧΗΧ :ΑΡΧΗΥ
REPEAT :ΦΟΡΕΣ
  [ ΜΕΤΑΤΟΠΙΣΗ (:ΜΕΓΕΘΟΣ * 1.5) 0
    MAKE "ΜΕΓ :ΜΕΓΕΘΟΣ
    MAKE "ΕΠΙΛΟΓΗ RANDOM 6
    IFELSE :ΕΠΙΛΟΓΗ = 0
      [ ΣΚΑΡΑΒΑΙΟΣ :ΜΕΓ ]
      [ IFELSE :ΕΠΙΛΟΓΗ = 1
        [ MAKE "ΟΡΟΦΟΙ RANDOM 3
          ΠΟΛΥΟΡΟΦΟ :ΜΕΓ :ΟΡΟΦΟΙ + 1 ]
        [ IFELSE :ΕΠΙΛΟΓΗ = 2
          [ ΛΟΥΛΟΥΔΙ :ΜΕΓ ]
          [ IFELSE :ΕΠΙΛΟΓΗ = 3
            [ ΡΟΔΙΑ :ΜΕΓ / 3 ]
            [ ΕΛΑΤΟ :ΜΕΓ ]
          ]
        ]
      ]
    ]
  ]
END
  
```

Η μεταβλητή
ΕΠΙΛΟΓΗ
παίρνει τυχαία
μια τιμή από 0
μέχρι 5.
Αν η τιμή είναι 0
τότε σχεδιάζεται
ένα αυτοκίνητο,
αλλιώς αν η τιμή
είναι 1 τότε
σχεδιάζεται ένα
κτίριο, αλλιώς
αν η τιμή είναι 2
τότε σχεδιάζεται
ένα λουλούδι,
αλλιώς αν η τιμή
είναι 3 τότε
σχεδιάζεται μια
ροδιά αλλιώς
(τιμή 4 ή 5)
σχεδιάζεται ένα
έλατο.



Βάλε το τοπίο όπου θέλεις

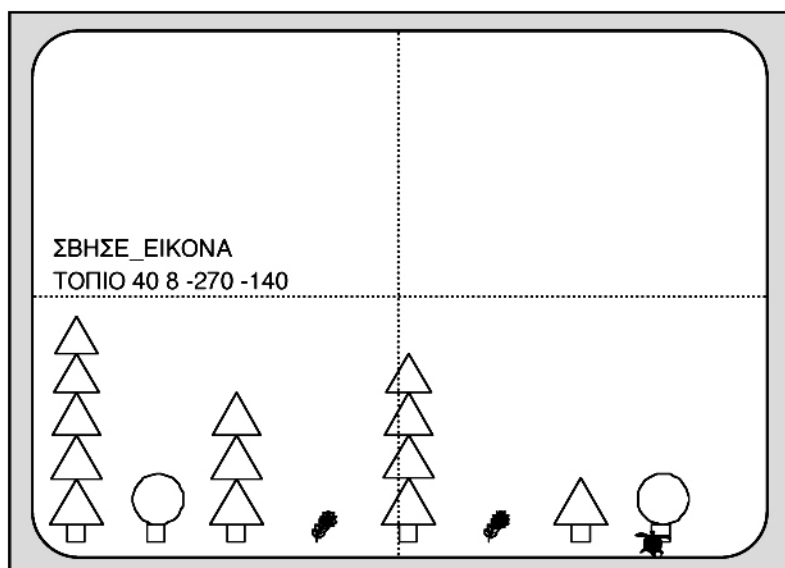
Μπορούμε να παραμετροποιήσουμε την προηγούμενη διαδικασία δίνοντας κατά την κλήση της τις απόλυτες συντεταγμένες από τις οποίες θα αρχίσει να σχεδιάζεται το τοπίο.

```

ΤΟ ΤΟΠΙΟ :ΜΕΓΕΘΟΣ :FORES :ΑΡΧΗΧ :ΑΡΧΗΥ
ΑΠΟΛΥΤΗ_ΘΕΣΗ :ΑΡΧΗΧ :ΑΡΧΗΥ
REPEAT :FORES
  [ ΜΕΤΑΤΟΠΙΣΗ (:ΜΕΓΕΘΟΣ * 1.5) 0
  MAKE "MEG :MEG
  MAKE "ΕΠΙΛΟΓΗ RANDOM 8
  IFELSE :ΕΠΙΛΟΓΗ = 0
    [ ΣΚΑΡΑΒΑΙΟΣ :ΜΕΓ * 1.2 ]
    [ IFELSE :ΕΠΙΛΟΓΗ = 1
      [ MAKE "ΟΡΟΦΟΙ RANDOM 3
        ΠΟΛΥΟΡΟΦΟ :ΜΕΓ :ΟΡΟΦΟΙ + 1 ]
      [ IFELSE :ΕΠΙΛΟΓΗ = 2
        [ ΛΟΥΛΟΥΔΙ :ΜΕΓ * 2 ]
        [ IFELSE :ΕΠΙΛΟΓΗ = 3
          [ ΡΟΔΙΑ :ΜΕΓ / 3 ]
          [ ΕΛΑΤΟ :ΜΕΓ ]
        ]
      ]
    ]
  ]
END

```

Το κείμενο του προγράμματος πρέπει να είναι ευανάγνωστο για να διευκολύνει τον προγραμματιστή να το κατανοεί εύκολα. Για το λόγο αυτό δε βλέπουμε να σπαταλώνται μερικές γραμμές (όπως γίνεται με τις αγκύλες που κλείνουν στη διπλανή διαδικασία) προκειμένου ο κώδικας του προγράμματος να είναι περισσότερο ευανάγνωστος.



Μια ολοκληρωμένη ζωγραφιά

Μπορούμε να γεμίσουμε την οθόνη με τοπία, που το ένα να βρίσκεται πίσω από το άλλο και να έχει μικρότερο μέγεθος ώστε να δημιουργείται η αίσθηση του βάθους.

Η διαδικασία ΖΩΓΡΑΦΙΑ, σχεδιάζει τα τοπία με μια επανάληψη που καθορίζεται από την παράμετρο BAQ. Σε κάθε επανάληψη οι συντεταγμένες από τις οποίες αρχίζει η σχεδίαση του τοπίου μεταβάλλονται, καθώς επίσης και το μέγεθος των αντικειμένων του τοπίου.

```
TO ΖΩΓΡΑΦΙΑ :MEG :FORA :ARXHX :ARXHY :BAQ
```

```
REPEAT :BAQ
```

```
  [ ΤΟΠΙΟ :MEG :FORA :ARXHX :ARXHY
```

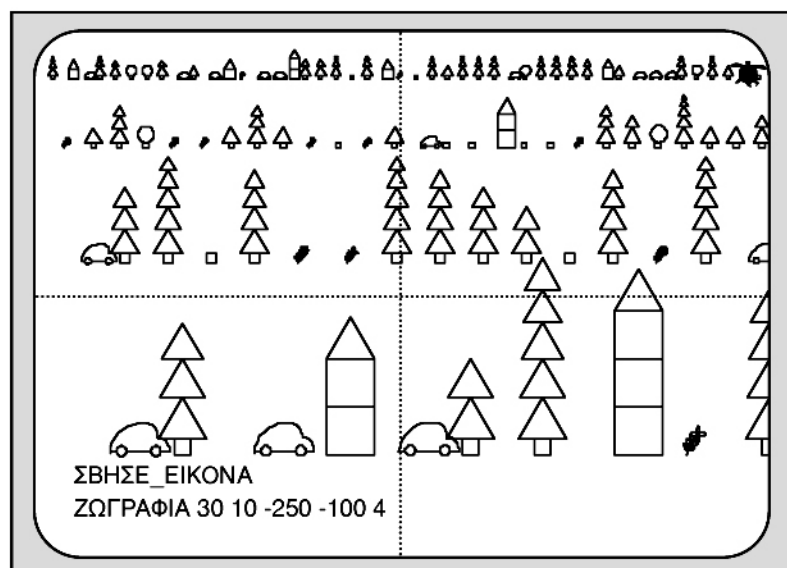
```
    MAKE "ARXHX :ARXHX
```

```
    MAKE "ARXHY :ARXHY + (4 * :MEG)
```

```
    MAKE "FORA "FORA * 1.7
```

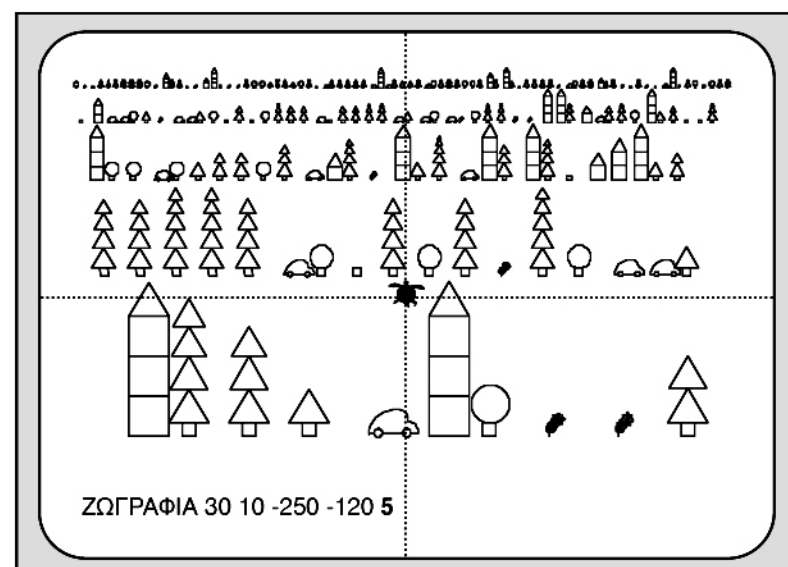
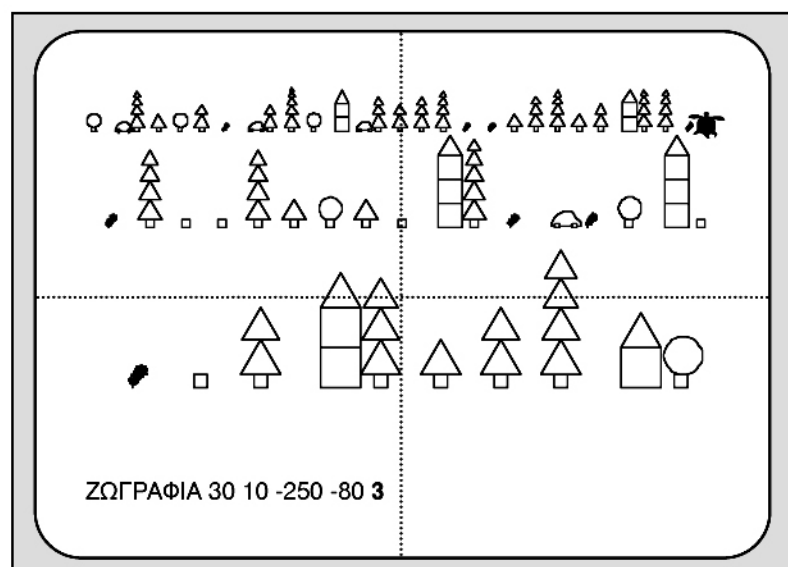
```
    MAKE "MEG :MEG * 0.6 ]
```

```
END
```



Η δημιουργικότητα της χελώνας

Ας αφήσουμε τη χελώνα να δημιουργήσει τη δική της πανασοθήκη, σχεδιάζοντας με τη βοήθεια της τύχης μερικές ζωγραφιές.



13 Αναδρομικές διαδικασίες

Ένα δυαδικό δένδρο

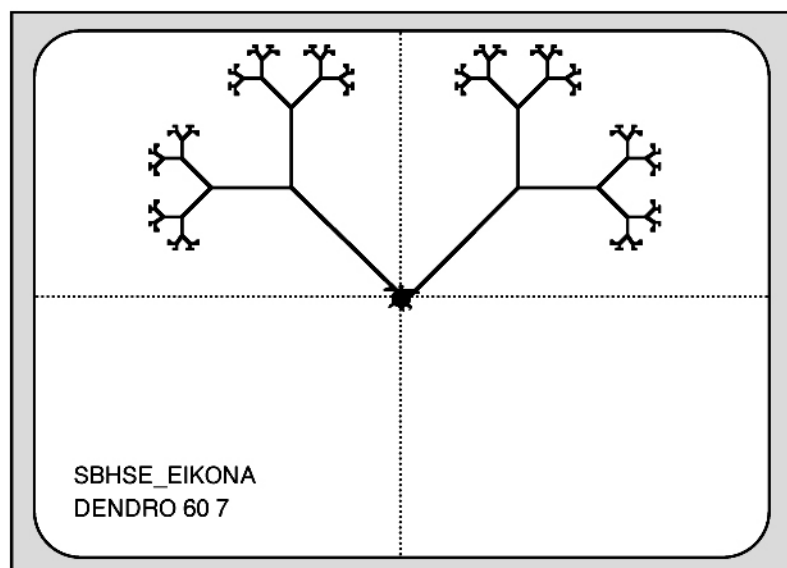
Η διαδικασία DENDRO που περιγράφεται παρακάτω σχηματίζει την εικόνα ενός δυαδικού δένδρου.

Η διαδικασία λειτουργεί αναδρομικά καλώντας τον εαυτό της δύο φορές, μια για τα αριστερά κλαδιά και μια για τα δεξιά.

Το «βάθος» που αναπαράγονται τα κλαδιά του δένδρου καθορίζεται από τη μεταβλητή GENIA, που ελέγχει πότε θα σταματήσει η αναδρομική κλήση της διαδικασίας.

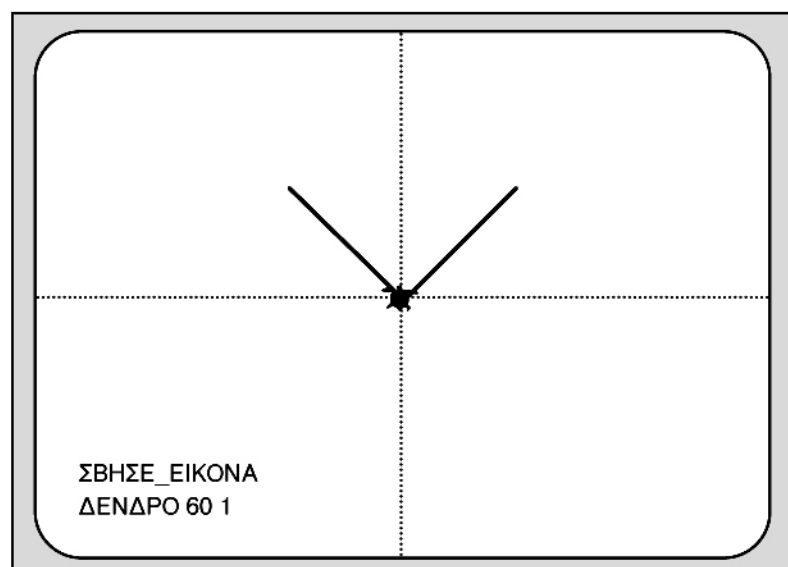
```
TO ΔΕΝΔΡΟ :KLADI :GENIA
IF :GENIA > 0
  [ ΓΡΑΜΜΗ 315 :KLADI
    ΔΕΝΔΡΟ :KLADI * (1 / 2) :GENIA - 1
    ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
    ΓΡΑΜΜΗ 90 :KLADI
    ΔΕΝΔΡΟ :KLADI * (1 / 2) :GENIA - 1
    ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
    ΑΡΙΣΤΕΡΑ 45 ]
END
```

Το μέγεθος του δένδρου καθορίζεται από την τιμή της μεταβλητής KLADI. Σε κάθε επόμενη κλήση της διαδικασίας DENDRO το μήκος του κλαδιού μειώνεται στο μισό.

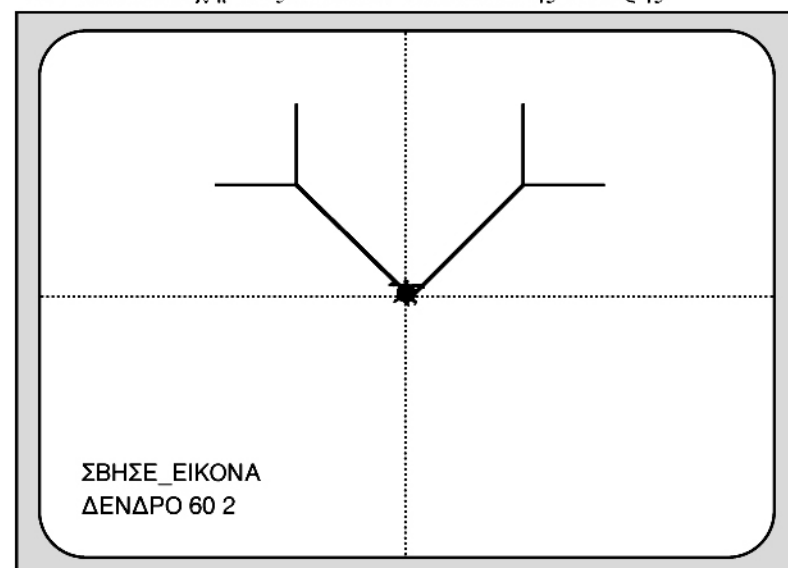


Διαφορετικές γενιές δυαδικών δένδρων

Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 1 , σχηματίζεται η πρώτη γενιά των κλαδιών.

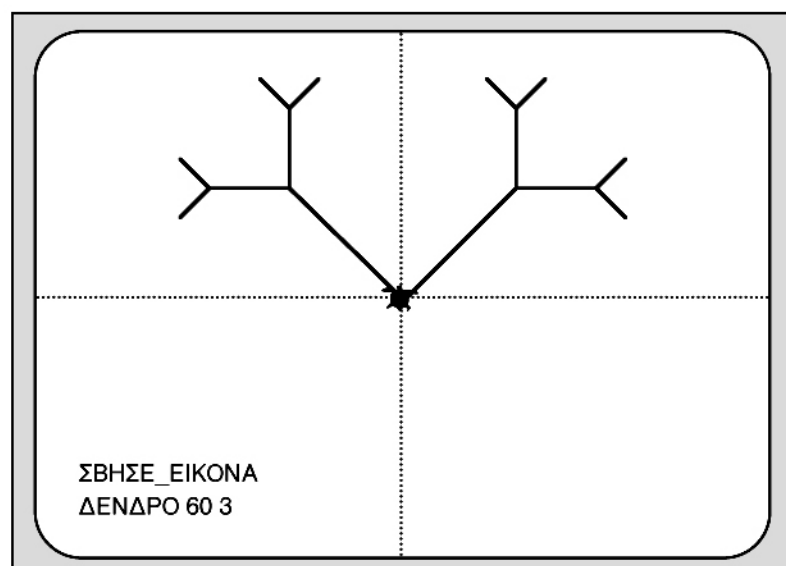


Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 2 , εκτός από την πρώτη γενιά των κλαδιών σχηματίζονται και τα κλαδιά της δεύτερης.

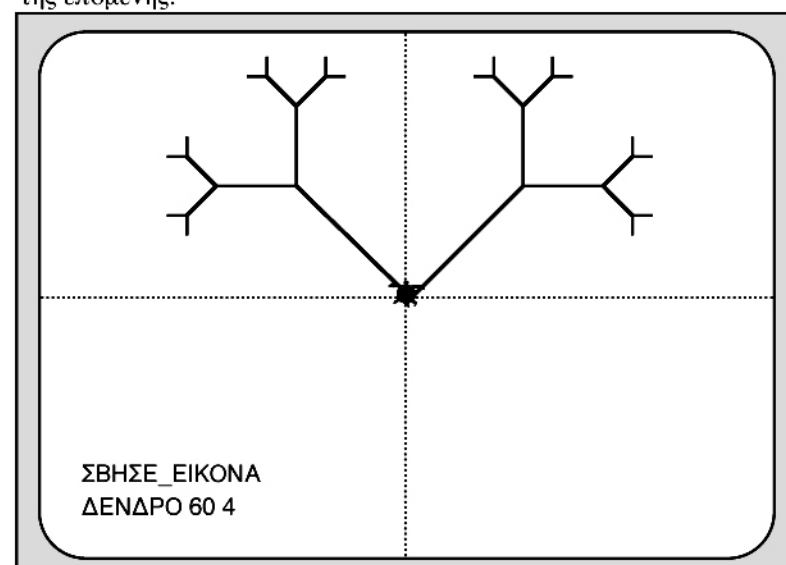


Διαφορετικές γενιές δυαδικών δένδρων

Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 3, εκτός από την πρώτη και δεύτερη γενιά των κλαδιών σχηματίζονται και τα κλαδιά της τρίτης.



Αν κληθεί η διαδικασία ΔΕΝΔΡΟ 60 4, εκτός από την πρώτη, δεύτερη και τρίτη γενιά των κλαδιών, σχηματίζονται και τα κλαδιά της επόμενης.



Πως δουλεύει η αναδρομή

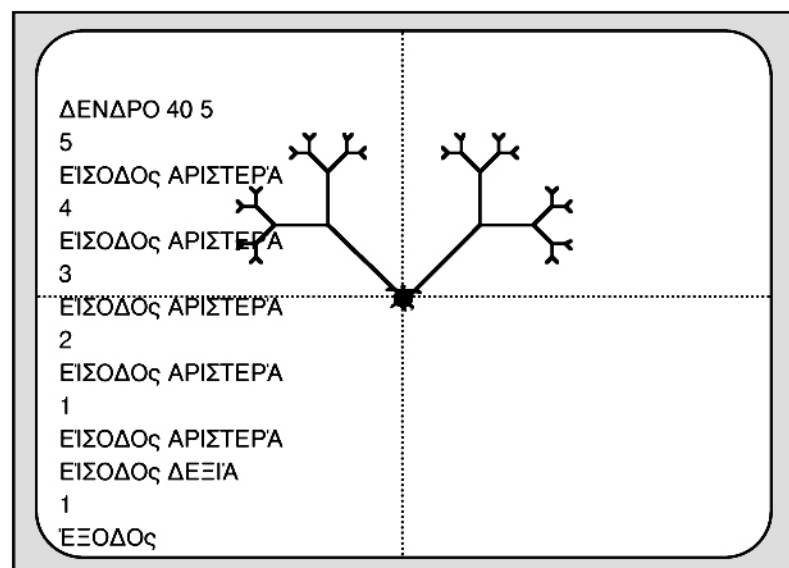
Μια τεχνική για να κατανοήσουμε τον τρόπο λειτουργίας ενός μηχανισμού, είναι να τον παρακολουθήσουμε βήμα προς βήμα όταν δουλεύει. Αυτό θα κάνουμε για τη διαδικασία ΔΕΝΔΡΟ. Για να μπορέσουμε να παρακολουθήσουμε την εξέλιξη του προγράμματος, θα αριθμήσουμε τις γραμμές του και θα προσθέσουμε μερικές εντολές που θα μας δείχνουν το εκάστοτε σημείο της ροής.

```

1  ΤΟ ΔΕΝΔΡΟ :ΚΛΑΔΙ :ΓΕΝΙΑ
2  IF :ΓΕΝΙΑ > 0
3      [PRINT :ΓΕΝΙΑ
4      PRINT "είσοδος αριστερά
5      ΓΡΑΜΜΗ 315 :ΚΛΑΔΙ
6      MAKE "Q READCHAR
7      ΔΕΝΔΡΟ :ΚΛΑΔΙ * (1 / 2) :ΓΕΝΙΑ - 1
8      ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ
9
10     PRINT "είσοδος δεξιά
11     ΓΡΑΜΜΗ 90 :ΚΛΑΔΙ
12     MAKE "Q READCHAR
13     ΔΕΝΔΡΟ :ΚΛΑΔΙ * (1 / 2) :ΓΕΝΙΑ - 1
14     ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :ΚΛΑΔΙ
15     ΑΡΙΣΤΕΡΑ 45
16     PRINT :ΓΕΝΙΑ
17     PRINT "έξοδος
18     MAKE "Q READCHAR]
19  END
    
```

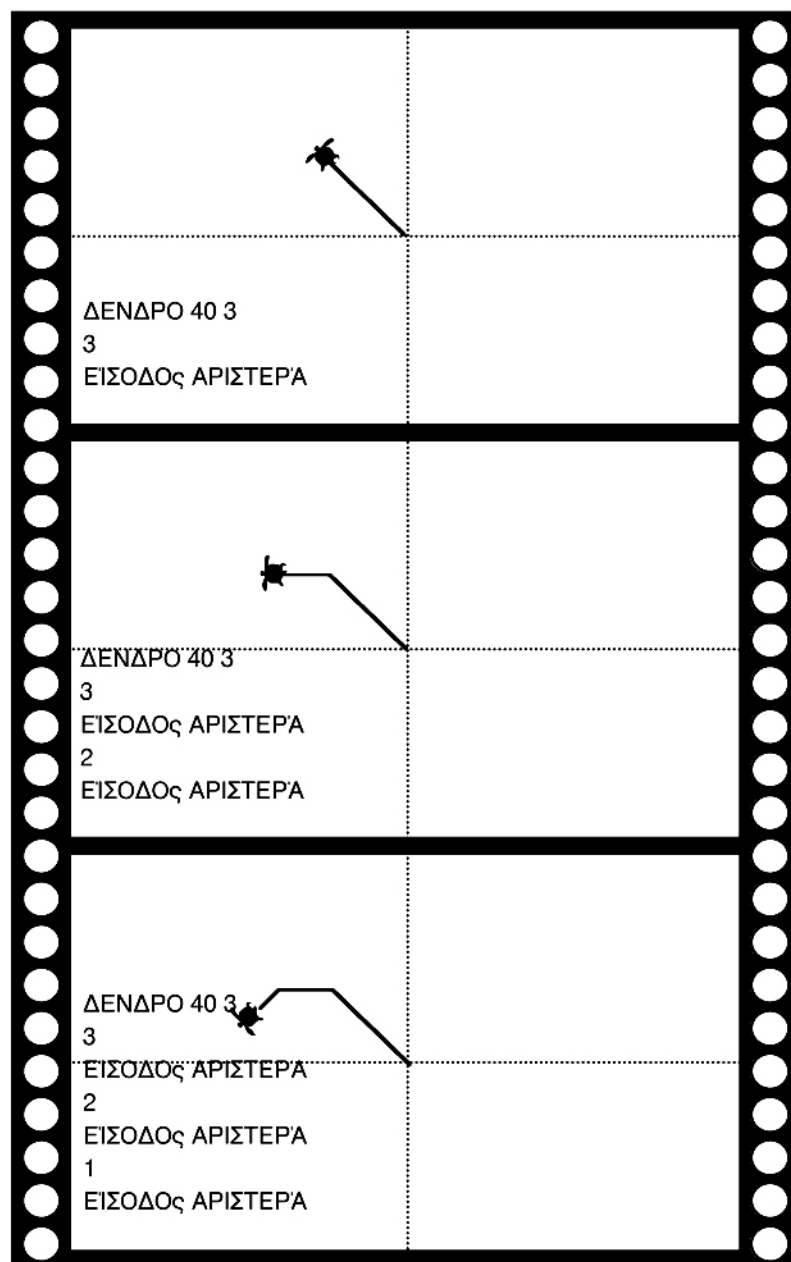
Οι εντολές των γραμμών 3,4,10,16,17 εμφανίζουν στην οθόνη μηνύματα που μας βοηθούν να καταλάβουμε ποιό σημείο του προγράμματος εκτελείται την αντίστοιχη στιγμή.

Οι εντολές των γραμμών 6,12,18 σταματούν τη ροή του προγράμματος μέχρι να δώσει ο χρήστης κάποιο δεδομένο από το πληκτρολόγιο. Αυτό το τέχνασμα χρησιμοποιείται όταν θέλουμε να τρέξουμε κομμάτι-κομμάτι το πρόγραμμα για να το παρακολουθήσουμε.



Αναρρίχηση σε δυαδικό δένδρο (1)

Ας παρακολουθήσουμε την εκτέλεση της διαδικασίας ΔΕΝΔΡΟ στην οθόνη (σχέδιο και μηνύματα) και στη μνήμη του υπολογιστή.



Το ΔΕΝΔΡΟ 40 3
βάζει στη μεταβλητή
ΚΛΑΔΟ το 40, στη
ΓΕΝΙΑ το 3 και
εκτελεί τις εντολές
μέχρι τη γραμμή 6.
Εκεί σταματάει την
εκτέλεση και
αναμένει πάτημα
πληκτρού από το
πληκτρολόγιο για να
συνεχίσει.

εντολές κλαδι γενιά		
	40	3

Η γραμμή 7
διακόπτει τη ροή της
ΔΕΝΔΡΟ 40 3 και
αφού φυλάξει τον
αριθμό γραμμής 8,
καλεί το ΔΕΝΔΡΟ
202 του οποίου
εκτελεί τις εντολές
μέχρι τη γραμμή
6.

εντολές κλαδι γενιά		
8	10	1
8	20	2
	40	3

Η γραμμή 7
διακόπτει τη ροή της
ΔΕΝΔΡΟ 202 και
αφού φυλάξει τον
αριθμό γραμμής 8
καλεί το ΔΕΝΔΡΟ
101 του οποίου
εκτελεί τις εντολές
μέχρι την γραμμή 6.

Αναρρίχηση σε δυαδικό δένδρο (2)

Η χελώνα συνεχίζει τη σχεδίαση του αριστερού κλαδιού του δένδρου με όλα τα παρακάδια του.

Όμοια καλείται η ΔΕΝΔΡΟ 05 και εκτελεί τις γραμμές 1, 2, 19.

εντολές κλαδι		γενιά
8	5	0
8	10	1
8	20	2
8	40	3

Το ΔΕΝΔΡΟ 05 πετάει τις τιμές 5, 0 και παίρνει από τη στοίβα την πιο πάνω εντολή (8). Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 101.

εντολές κλαδι		γενιά
8	10	1
8	20	2
8	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 50 και αφού φυλάξει το 14 εκτελεί τις γραμμές 1, 2, 19.

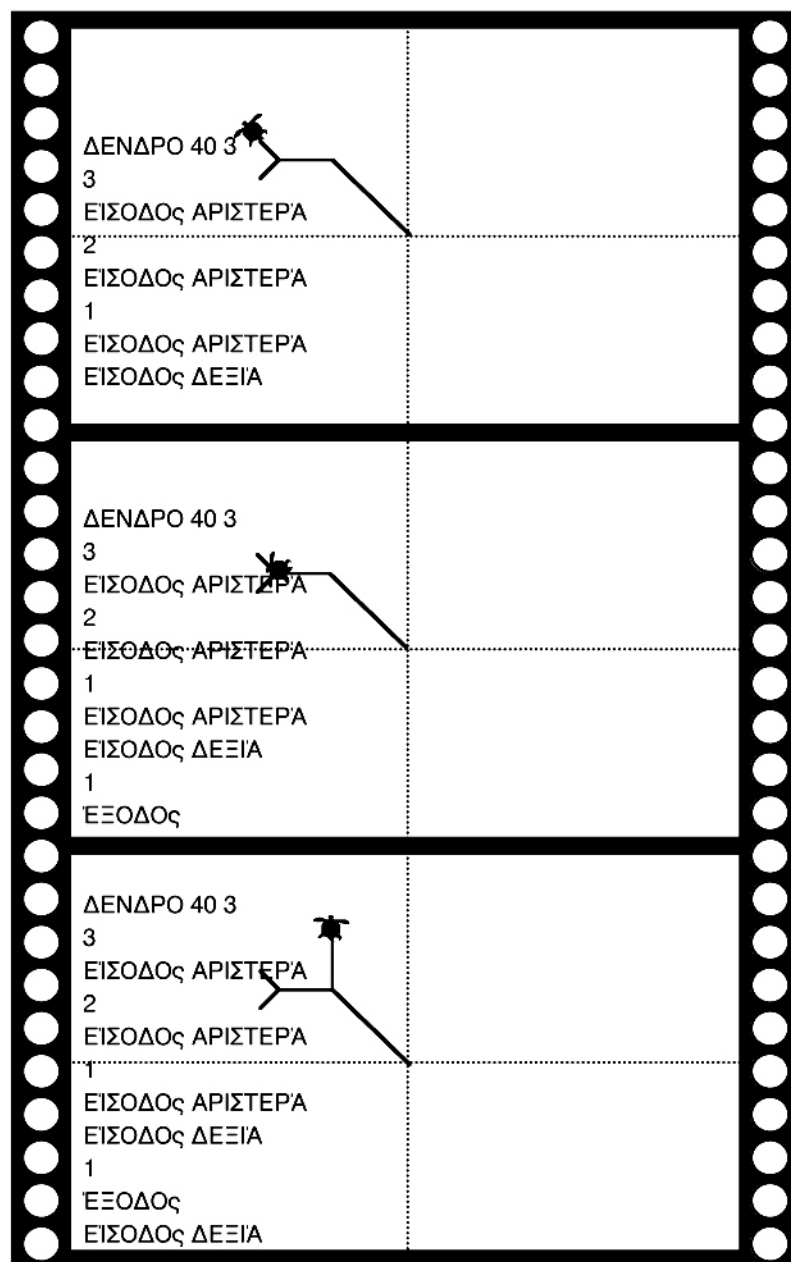
εντολές κλαδι		γενιά
14	5	0
8	10	1
8	20	2
8	40	3

Το ΔΕΝΔΡΟ 50 πετάει τις τιμές 5, 0 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 101.

εντολές κλαδι		γενιά
8	10	1
8	20	2
8	40	3

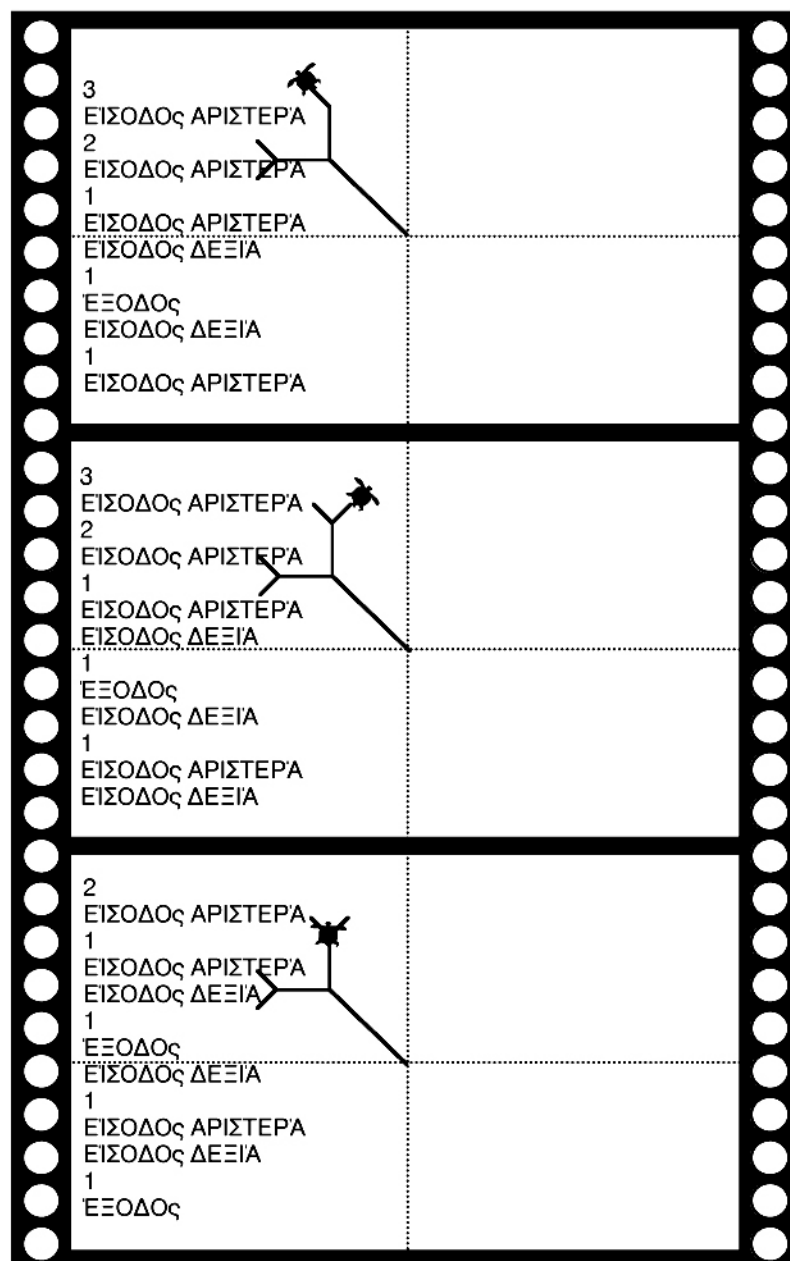
Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 101 πετάει τις τιμές 10, 1 και παίρνει από τη στοίβα τα 8. Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 202.

εντολές κλαδι		γενιά
8	20	2
8	40	3



Αναρρίχηση σε δυαδικό δένδρο (3)

Η χελώνα έχει ολοκληρώσει τη σχεδίαση του αριστερού κλαδιού του δένδρου και επιστρέφει στη ρίζα.



Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 101 και αφού φυλάξει τον αριθμό γραμμής 14, εκτελεί τις εντολές μέχρι τη γραμμή 6 του ΔΕΝΔΡΟ 101.

εντολές κλαδι γενιά

14	10	1
8	20	2
8	40	3

Η γραμμή 7 καλεί τη ΔΕΝΔΡΟ 5 και εκτελεί τις γραμμές 1, 2, 19.

εντολές κλαδι γενιά

8	5	0
14	10	1
8	20	2
8	40	3

Το ΔΕΝΔΡΟ 5 ορίζεται τις τιμές 5, 0 και παίρνει την πιο πάνω εντολή από τη στοίβα και εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 101.

εντολές κλαδι γενιά

14	10	1
8	20	2
8	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 50 και αφού φυλάξει το 14 εκτελεί τις γραμμές 1, 2, 19.

εντολές κλαδι γενιά

14	5	0
14	10	1
8	20	2
8	40	3

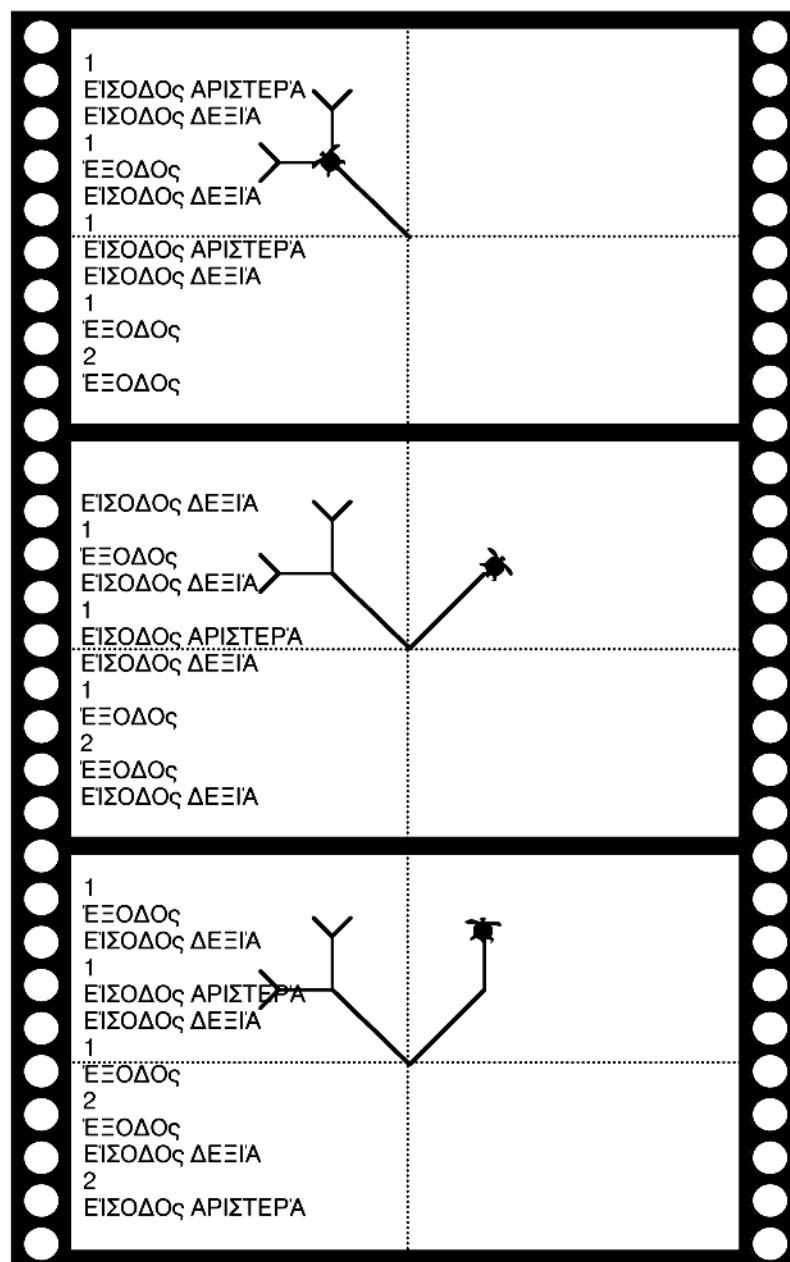
Το ΔΕΝΔΡΟ 50 ορίζεται τις τιμές 5, 0 και παίρνει από τη στοίβα το 14 και εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 101.

εντολές κλαδι γενιά

14	10	1
8	20	2
8	40	3

Αναρρίχηση σε δυαδικό δένδρο (4)

Η χελώνα επέστρεψε στη ρίζα και από εκεί ξεκινά να σχεδιάζει το δεξί κλαδί του δένδρου.



Η εντολή γραμμής
19(ΕΝΔ) της
ΔΕΝΔΡΟ 101 πετάει
τις τιμές 10, 1 και παίρνει
από τη στοίβα το 14.
Έτσι εκτελεί τις γραμμές
15-18 της
ΔΕΝΔΡΟ 202.

εντολές κλαδί γενιά

8	20 40	2 3
---	----------	--------

Η εντολή γραμμής
19(ΕΝΔ) της
ΔΕΝΔΡΟ 202 πετάει
τις τιμές 20, 2 και παίρνει
από τη στοίβα τα 8. Έ
εκτελεί τις γραμμές 8-12
της ΔΕΝΔΡΟ 403.

εντολές κλαδί γενιά

	40	3
--	----	---

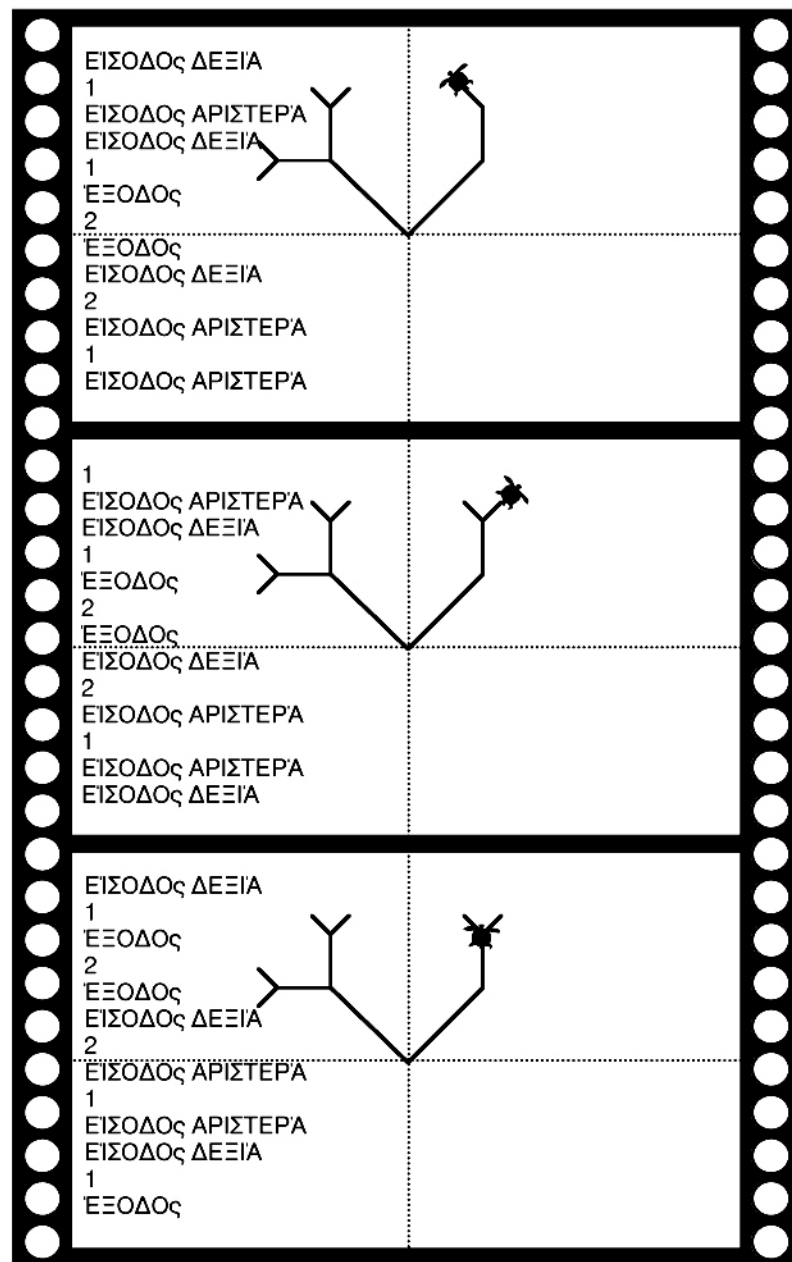
Η γραμμή 13 καλεί τη
ΔΕΝΔΡΟ 202 και αφού
φυλάξει τον αριθμό
γραμμής 14 εκτελεί τις
εντολές μέχρι τη γραμμή 6
του ΔΕΝΔΡΟ 202.

εντολές κλαδί γενιά

14	20 40	2 3
----	----------	--------

Αναρρίχηση σε δυαδικό δένδρο (5)

Η χελώνα συνεχίζει τη σχεδίαση του δεξιού κλαδιού του δένδρου με όλα τα παρακλάδια του.



Η γραμμή 7 διακόπτει τη ροή της ΔΕΝΔΡΟ 20 2 και αφού φυλάξει τον αριθμό γραμμής 8 καλεί το ΔΕΝΔΡΟ 101 του οποίου εκτελεί τις εντολές μέχρι τη γραμμή 6.

εντολές	κλαδί	γενιά
	10	1
8	20	2
14	40	3

Όμοια καλείται η
ΔΕΝΔΡΟ και εκτελεί
τις γασιμές 1, 2, 19.

εντολές	κλαδί	γενιά
	5	0
8	10	1
8	20	2
14	40	3

Το **END** δίνει τις τιμές 5,
Ο και παίρνει την πιο πάνω
εντολή από τη στοίβα.
Έτσι εκτελεί τις γραμμές 8-
12 της ΔΕΝΔΡΟ 101.

εντολές	κλαδί	γενιά
	10	1
8	20	2
14	40	3

Η γραμμή 13 καλεί τη
ΔΕΝΔΡΟ50 και αφού
φυλάξει το 14 εκτελεί τις
γραμμές 1, 2, 19.

εντολές	κλαδί	γενιό
14	5	0
8	10	1
14	20	2
14	40	3

Το ~~END~~ δέχεται τις τιμές 5,0 και παίρνει από τη στοίβα το ~~14~~ εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 101.

εντολές	κλαδί	γενιά
	5	0
14	10	1
8	20	2
14	40	3

Αναρρίχηση σε δυαδικό δένδρο (6)

Η χελώνα ολοκληρώνει τη σχεδίαση του δεξιού κλαδιού του δένδρου με όλα τα παρακάτω.

<p>1 ΕΞΟΔΟΣ 2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ</p>	
<p>1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p>	
<p>2 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ</p>	
<p>ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 2 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΞΟΔΟΣ ΕΙΣΟΔΟΣ ΔΕΞΙΑ 1 ΕΙΣΟΔΟΣ ΑΡΙΣΤΕΡΑ ΕΙΣΟΔΟΣ ΔΕΞΙΑ</p>	

Η γραμμή 19(ΕΝΔ) της ΔΕΝΔΡΟ 101 πετάει τις τιμές 10, 1 και παίρνει από τη στοίβα τα 8. Εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 202.

εντολές κλαδι γενιά		
14	20	2
14	40	3

Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ 101 και αφού φυλάξει τον αριθμό γραμμής 14 εκτελεί τις εντολές μέχρι τη γραμμή 6 του ΔΕΝΔΡΟ 101.

εντολές κλαδι γενιά		
14	10	1
14	20	2
14	40	3

Η γραμμή 7 καλεί τη ΔΕΝΔΡΟ 05 και εκτελεί τις γραμμές 1, 2, 19.

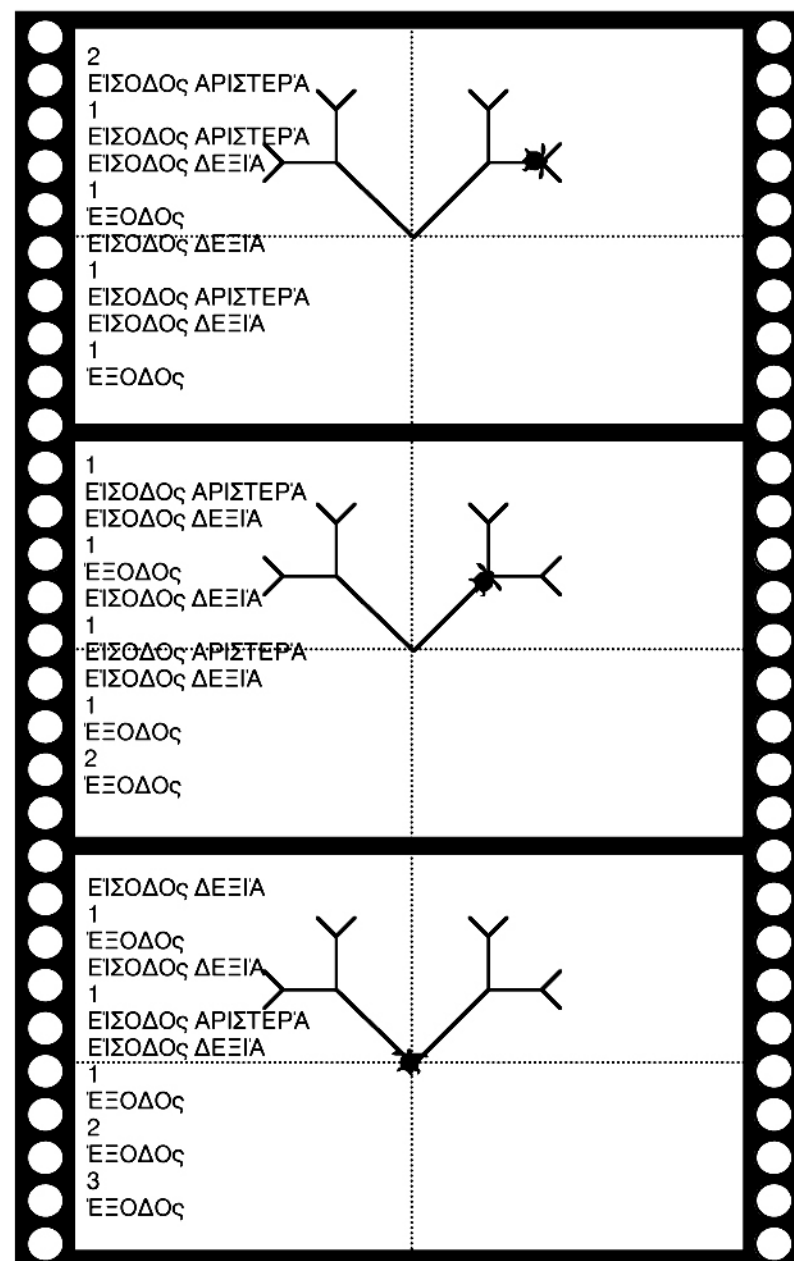
εντολές κλαδι γενιά		
8	5	0
14	10	1
14	20	2
14	40	3

Το ΕΝΔ πετάει τις τιμές 5, 0 και παίρνει την πτω πάνω εντολή από τη στοίβα. Εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 101.

εντολές κλαδι γενιά		
14	10	1
14	20	2
14	40	3

Αναρρίχηση σε δυαδικό δένδρο (7)

Η χελώνα έχει ολοκληρώσει τη σχεδίαση και του δεξιού κλαδιού του δένδρου και επιστρέφει στη ρίζα, όπου τερματίζει.



Η γραμμή 13 καλεί τη ΔΕΝΔΡΟ50 και αφού φυλάξει το 14 εκτελεί τις γραμμές 1, 2, 19.

εντολές κλαδί γενιά		
14	5	0
14	10	1
14	20	2
14	40	3

Το END πετάει τις τιμές 5, 0 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 14-18 της ΔΕΝΔΡΟ 101.

εντολές κλαδί γενιά		
14	10	1
14	20	2
14	40	3

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 101 πετάει τις τιμές 10, 1 και παίρνει από τη στοίβα το 14. Έτσι εκτελεί τις γραμμές 15-18 της ΔΕΝΔΡΟ 202.

εντολές κλαδί γενιά		
14	20	2
	40	3

Η εντολή της γραμμής 19 (END) της ΔΕΝΔΡΟ 202 πετάει τις τιμές 20, 2 και παίρνει από τη στοίβα τα 8. Έτσι εκτελεί τις γραμμές 8-12 της ΔΕΝΔΡΟ 403.

εντολές κλαδί γενιά		
	40	3

Το END της ΔΕΝΔΡΟ 403 τελειώνει την ολόκληρη διαδικασία.

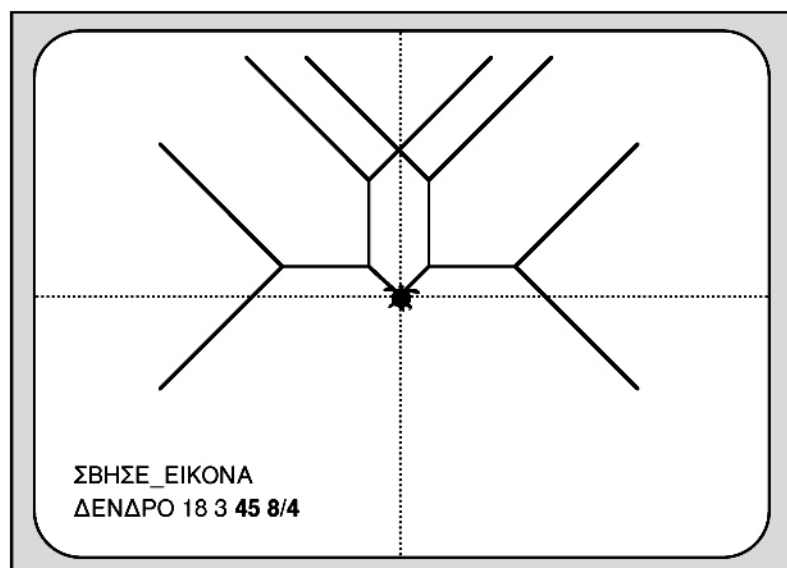
Ποικιλία δυαδικών δένδρων

Αφού κατανοήσαμε τον τρόπο λειτουργίας της αναδρομικής διαδικασίας ΔΕΝΔΡΟ, ας εισάγουμε δύο νέες μεταβλητές που θα την κάνουν περισσότερο ευέλικτη.

Η μεταβλητή FOYNT που οι τιμές της θα είναι σε μοίρες, θα καθορίζει το πόσο φουντωτό θα είναι το δένδρο επηρεάζοντας τη γωνία μεταξύ δύο κλαδιών της ίδιας γενιάς.

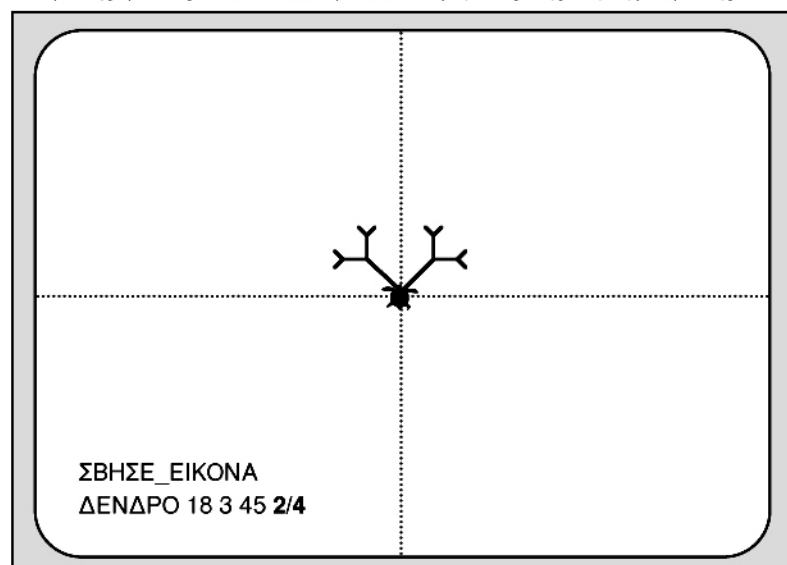
Η άλλη παράμετρος, η KLASMA, θα καθορίζει το ποσοστό της μεταβολής του μήκους των κλαδιών από γενιά σε γενιά.

```
ΤΟ ΔΕΝΔΡΟ :KLADI :GENIA :FOYNTA :KLASMA
IF :GENIA > 0
  [ΓΡΑΜΜΗ (360 - :FOYNTA) :KLADI
  ΔΕΝΔΡΟ :KLADI * :KLASMA :GENIA - 1 :FOYNTA :KLASMA
  ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
  ΓΡΑΜΜΗ (2 * :FOYNTA) :KLADI
  ΔΕΝΔΡΟ :KLADI * :KLASMA :GENIA - 1 :FOYNTA :KLASMA
  ΜΕΤΑΤΟΠΙΣΗ 0 MINUS :KLADI
  ΑΡΙΣΤΕΡΑ :FOYNTA]
END
```

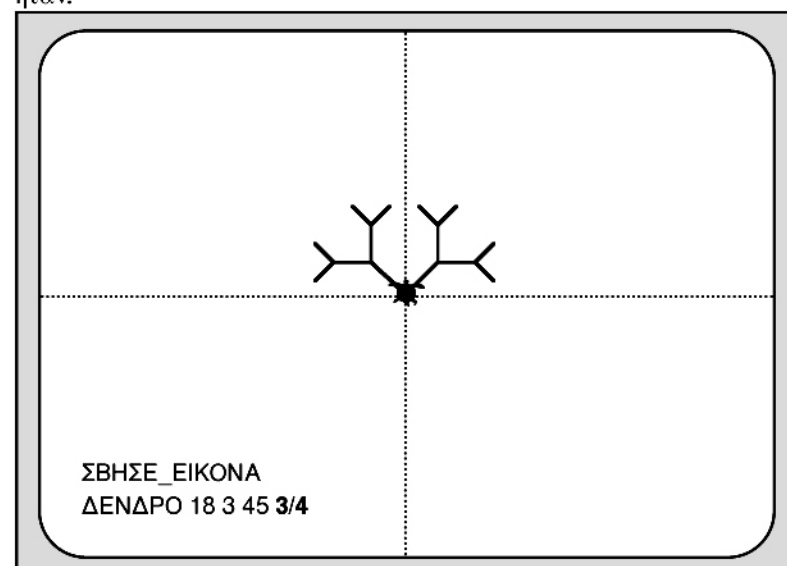


Παίζοντας με τις παραμέτρους (1)

Για να έχουμε ένα μέτρο σύγκρισης, θα ζητήσουμε από τη χελώνα να σχηματίσει ένα δένδρο που η γωνία μεταξύ των κλαδιών της ίδιας γενιάς να είναι 45 μοίρες και που το μήκος του κλαδιού κάθε επόμενης γενιάς να είναι το μισό του μήκους της προηγούμενης.

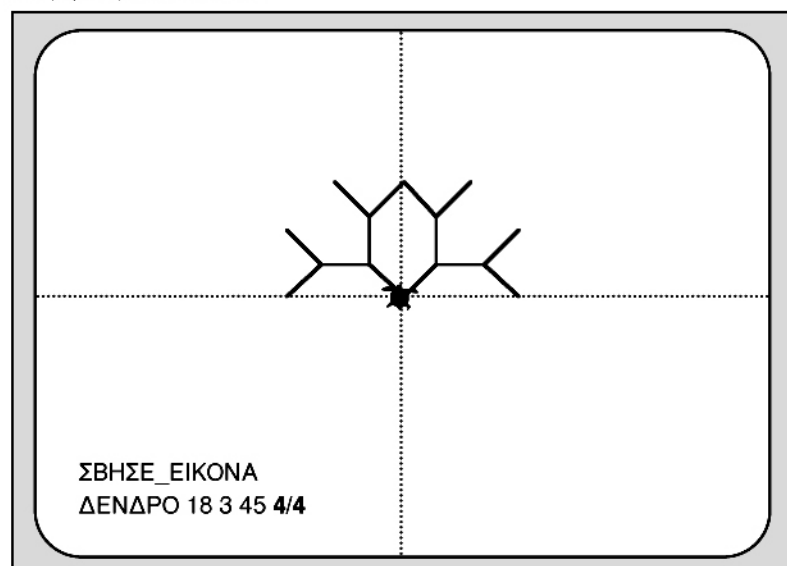


Στο επόμενο δένδρο θα διατηρήσουμε σταθερή τη γωνία μεταξύ των κλαδιών (45 μοίρες) και θα αλλάξουμε το ποσοστό της μεταβολής του μήκους του κλαδιού από γενιά σε γενιά κάνοντάς το $\frac{3}{4}$, από $\frac{2}{4}$ που ήταν.

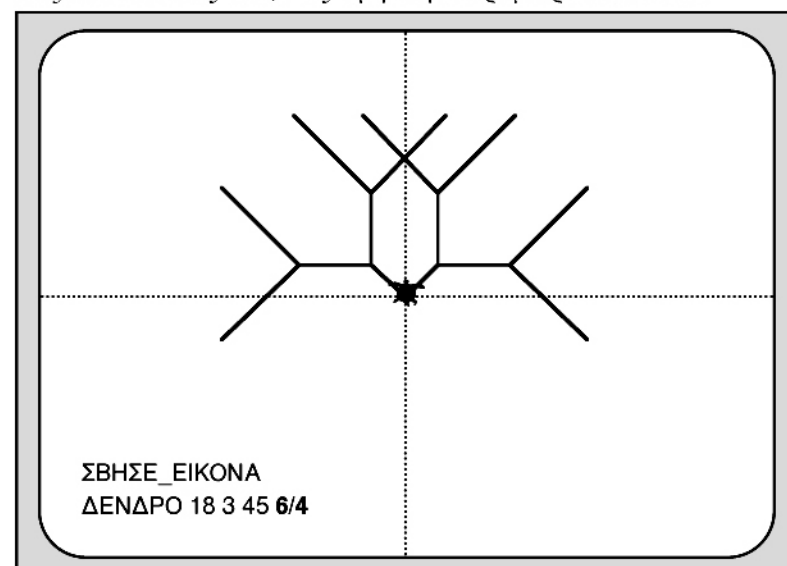


Παίζοντας με τις παραμέτρους (2)

Στο δένδρο που θα σχηματιστεί θα διατηρήσουμε σταθερό το μήκος των κλαδιών από γενιά σε γενιά, δίνοντας τα 4/4 ως τιμή στην παράμετρο KLASMA.

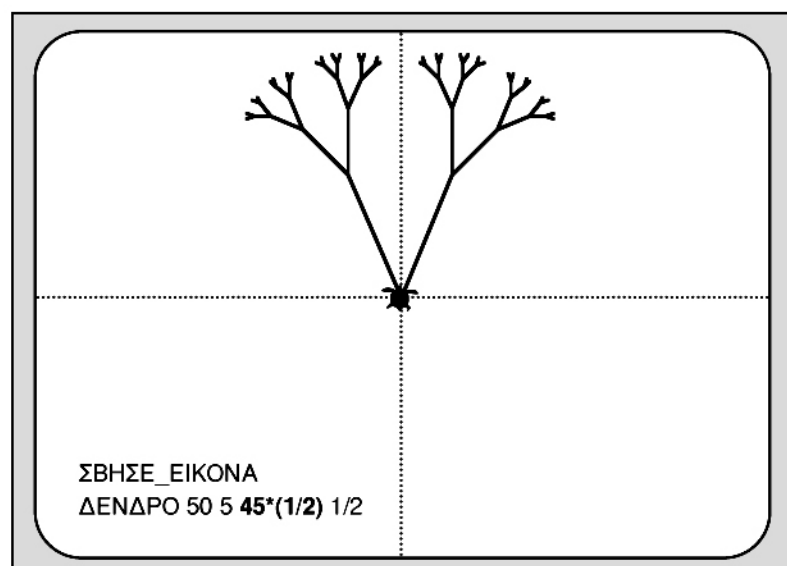


Στο επόμενο δένδρο, το μήκος των κλαδιών από γενιά σε γενιά θα αυξάνει δίνοντας τα 6/4 ως τιμή στην παράμετρο KLASMA.

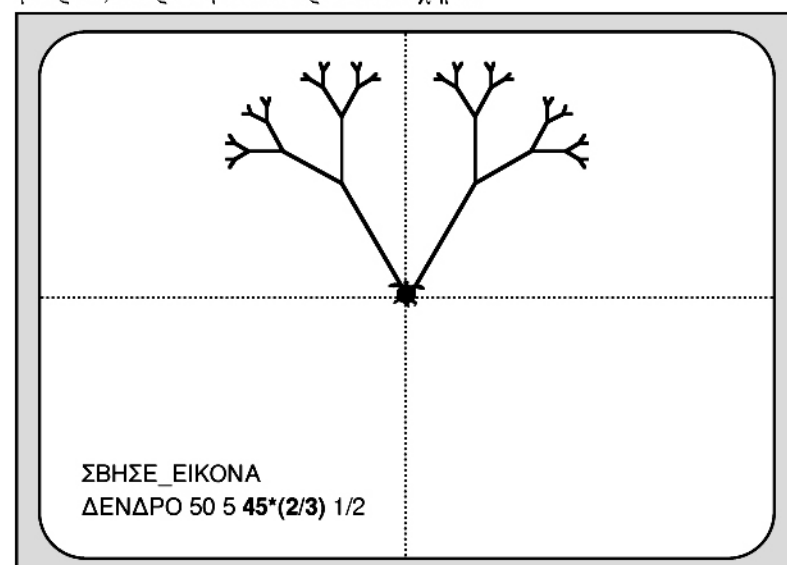


Παίζοντας με τις παραμέτρους (3)

Στη συνέχεια θα δώσουμε διάφορες τιμές στην παραμέτρο **ΦΟΥΝΤ** της διαδικασίας **ΔΕΝΔΡΟ**. Ας δούμε το δένδρο που η γωνία μεταξύ των κλαδίων θα είναι το $1/2$ των 45 μοιρών.

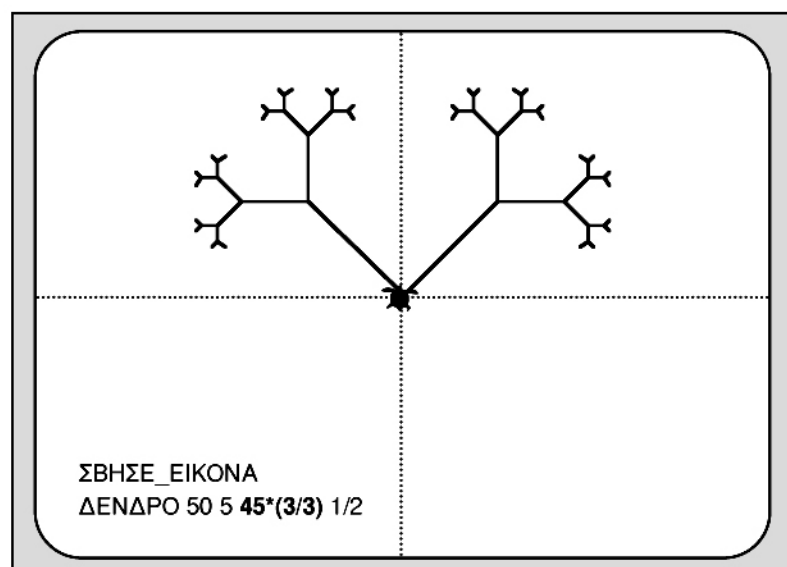


Αλλάζοντας τη γωνία μεταξύ των κλαδίων από το $1/2$ στα $2/3$ των 45 μοιρών, παίρνουμε το παρακάτω σχήμα.

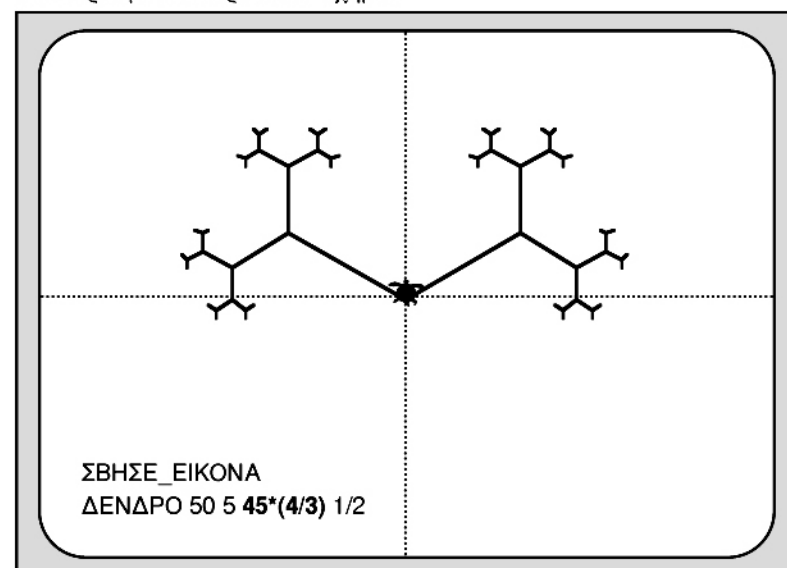


Παίζοντας με τις παραμέτρους (4)

Το επόμενο σχήμα θα σχηματιστεί αν αναφέρουμε τη γωνία μεταξύ των κλαδιών στις 45 μοίρες.



Αυξάνοντας τη γωνία μεταξύ των κλαδιών στα $\frac{4}{3}$ των 45 μοιρών, θα πάρουμε το παρακάτω σχήμα.



Τα διαδικα
δένδρα δίνουν
μια άλλη οπτική
στις
αναπαραστάσεις
των κλασμάτων.
Η έννοια του
κλάσματος ως
τμήμα κάποιου
διάστατου
(επιπέδου)
σχήματος
μεταφέρεται στη
μια διάσταση,
ως το μήκος
ενός
ευθύγραμμου
τμήματος.

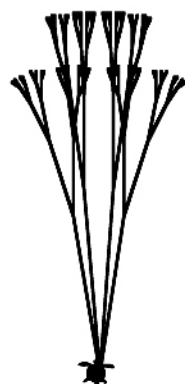
Κατ' εικόνα και ομοίωση

Δίνοντας διάφορες τιμές στις παραμέτρους, θα προσπαθήσουμε να δημιουργήσουμε σχήματα που να μοιάζουν με πραγματικά φυτά.

ΣΒΗΣΕ_ΕΙΚΟΝΑ

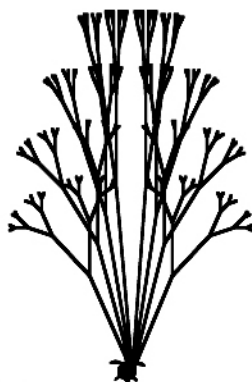
ΔΕΝΔΡΟ 70 5 (45/10) 1/2

ΔΕΝΔΡΟ 60 5 (45/5) 1/2



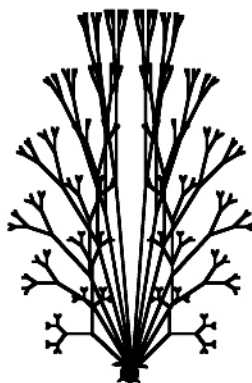
ΔΕΝΔΡΟ 50 5 (45/3) 1/2

ΔΕΝΔΡΟ 40 5 (45/2) 1/2



ΔΕΝΔΡΟ 50 5 (45/(3/2)) 1/2

ΔΕΝΔΡΟ 40 5 (45/1) 1/2



«Αυτά τα σχήματα έχουν πολλές ενδιαφέρουσες αναφορές στη βιολογία. Είναι εύκολο να δούμε πως μπορούμε με τη βοήθειά τους να ερευνήσουμε πολύ απλά τα κελύφη των σαλιγκαριών, τις κορυφές των σπόρων, τα κέρατα των ζώων και άλλα παρόμοια φαινόμενα. Η εργασία αυτή συνδέεται και με εκπαιδευτικά θέματα υψηλότερου επιπέδου...»
Boris Allan.

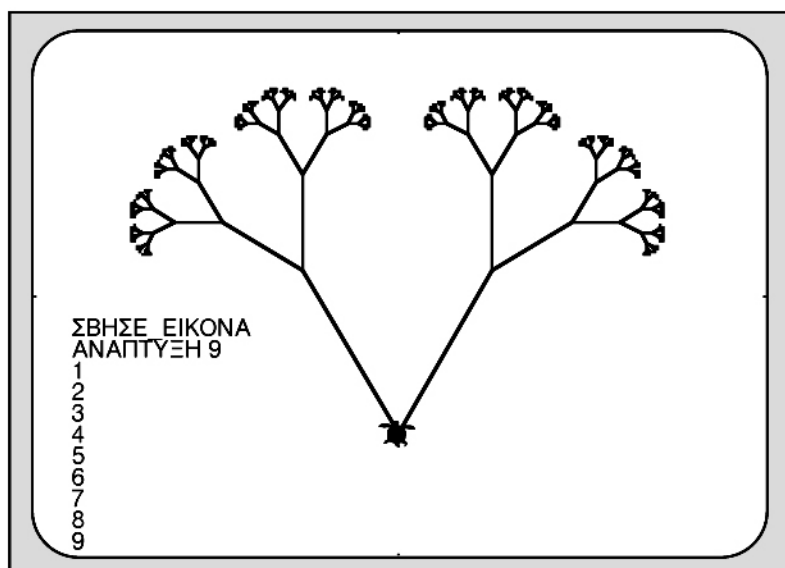
Η ανάπτυξη ενός φυτού

Μπορούμε να προσομοιώσουμε την ανάπτυξη ενός φυτού με μια διαδικασία που θα σχηματίζει διαδοχικά τις εικόνες του φυτού με το πέρασμα του χρόνου. Μια διαδικασία θα επαναλαμβάνει μερικές φορές (που θα αντιστοιχούν σε χρονικές στιγμές) τη διαδικασία ΔΕΝΔΡΟ και θα μεταβάλλει κάθε φορά τις τιμές των παραμέτρων της, έτσι ώστε να φαίνεται ότι το φυτό αναπτύσσεται.

Η επιτυχία της προσομοίωσης θα εξαρτηθεί από το βαθμό μεταβολής της κάθε παραμέτρου της διαδικασίας ΔΕΝΔΡΟ.

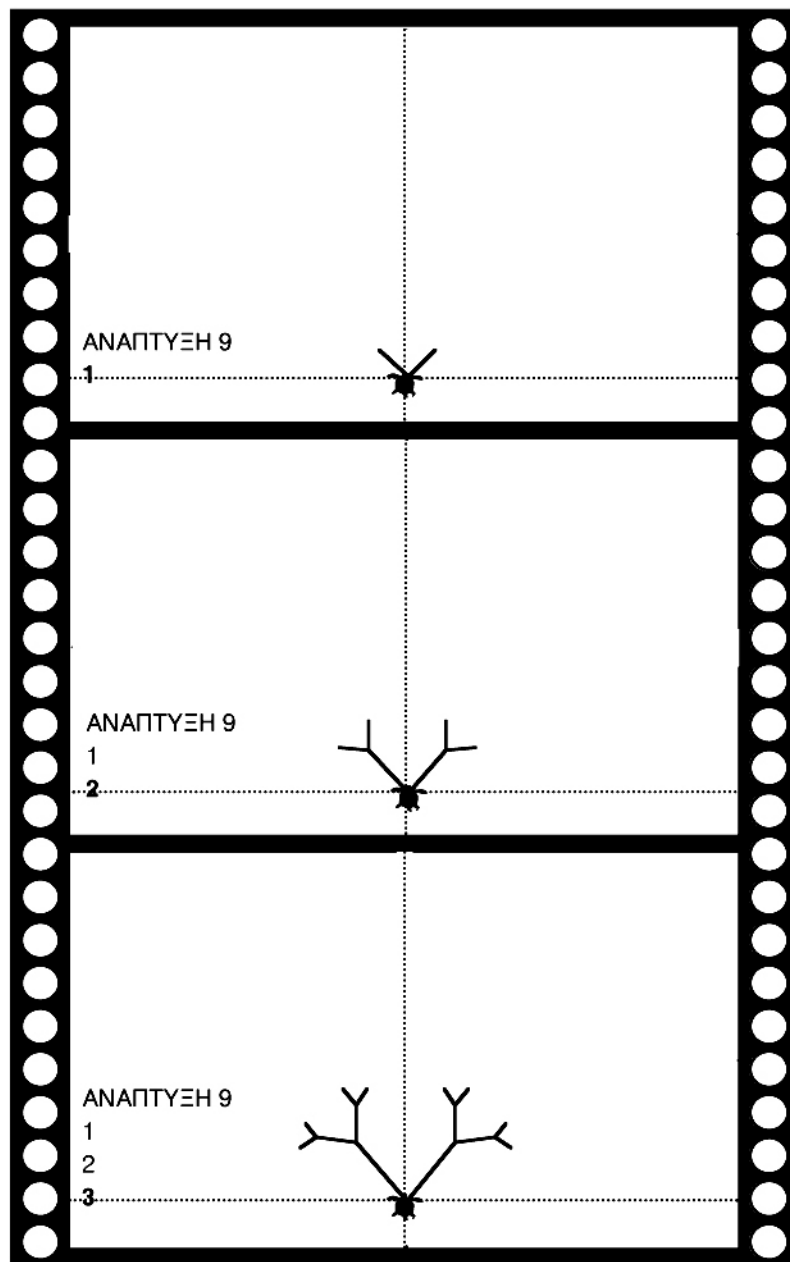
Ας δούμε μια τέτοια διαδικασία, την ΑΝΑΠΤΥΞΗ.

```
ΤΟ ΑΝΑΠΤΥΞΗ :ΧΡΟΝΙΑ
ΚΡΥΨΟΥ
MAKE "KLADI 15
MAKE "GENIA 1
MAKE "FOYNTA 45
MAKE "KLASMA 1 / 2
REPEAT :ΧΡΟΝΙΑ
  [ΣΒΗΣΕ_ΕΙΚΟΝΑ
  PRINT :GENIA
  ΔΕΝΔΡΟ :KLADI :GENIA :FOYNTA :KLASMA
  MAKE "KLADI :KLADI + 7
  MAKE "GENIA :GENIA + 1
  MAKE "FOYNTA :FOYNTA * 0.95]
ΕΜΦΑΝΙΣΕ_ΧΕΛΩΝΑ
END
```



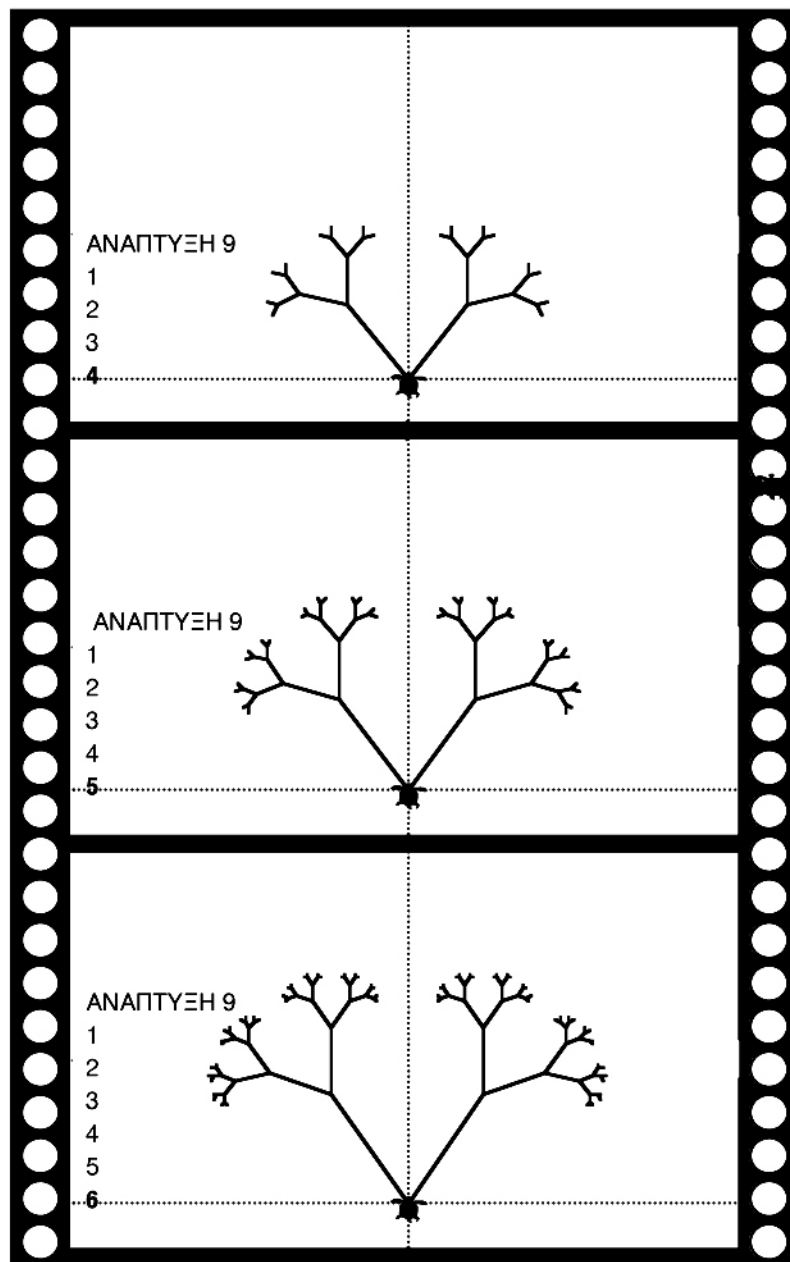
Το φυτό αναπτύσσεται (1)

Στα επόμενα σχήματα παρακολουθούμε στιγμιότυπα από την ανάπτυξη του φυτού κατά τα 3 πρώτα χρόνια-στάδια της ζωής του.



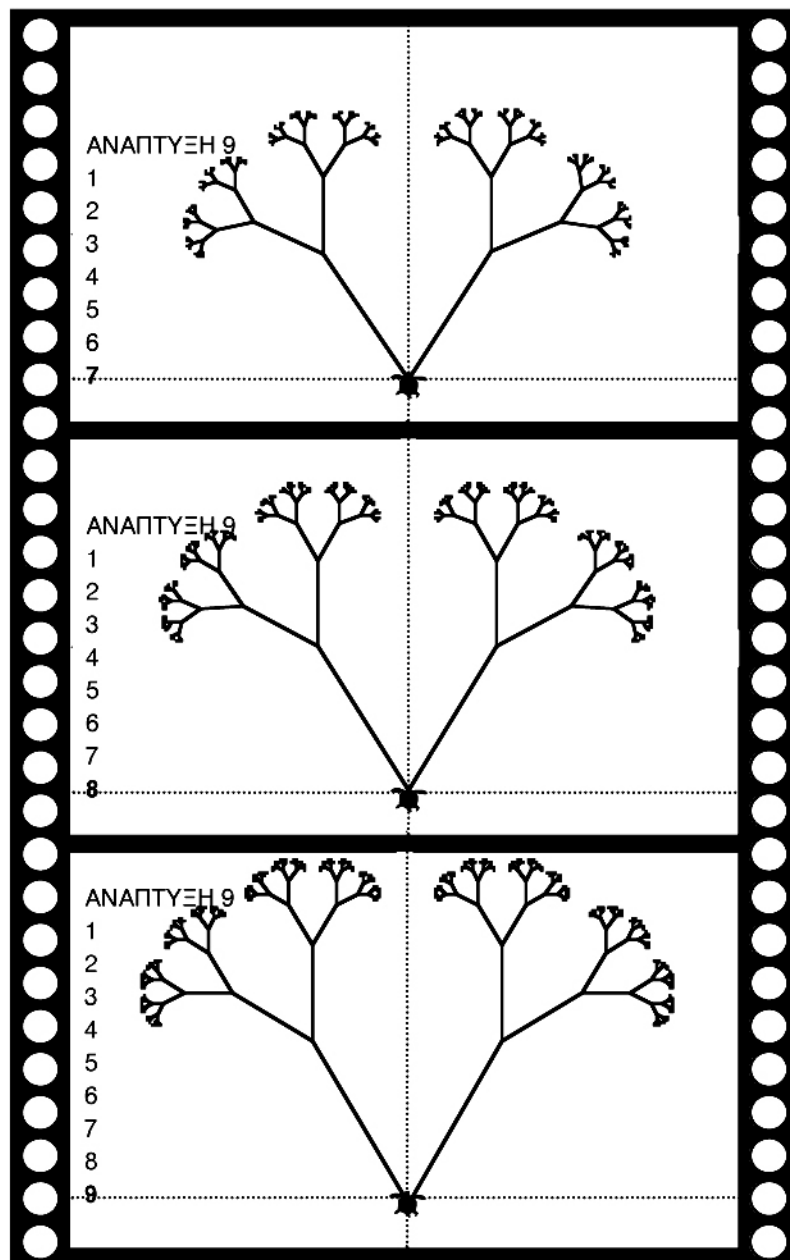
Το φυτό αναπτύσσεται (2)

Στα επόμενα στιγμιότυπα παρακολουθούμε την ανάπτυξη του φυτού από τον 4ο μέχρι τον 6ο χρόνο της ζωής του.



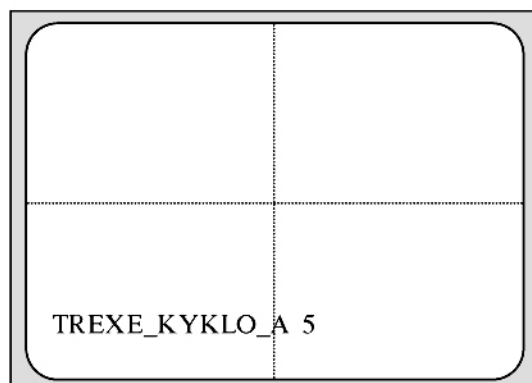
Το φυτό αναπτύσσεται (3)

Στα επόμενα στιγμιότυπα παρακολουθούμε την ανάπτυξη του φυτού από τον 7ο μέχρι τον 9ο χρόνο της ζωής του.



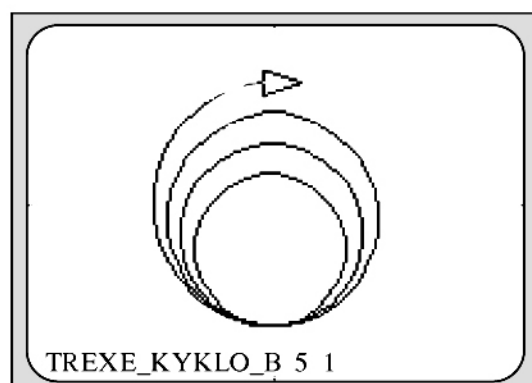
Κίνηση με άμεση αναδρομή

Ξέρουμε ήδη να σχεδιάζουμε την περιφέρεια ενός κύκλου (ως ένα τριανταεξάγωνο) με τη διαδικασία KYKLOS.



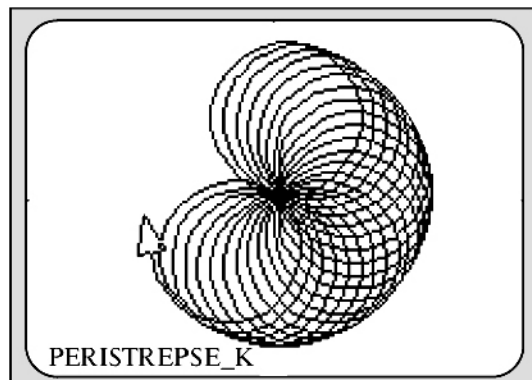
Μπορούμε με την διαδικασία TREXE_KKL_A να κάνουμε τη χελώνα να τρέχει αριστερόστροφα τον κύκλο αυτό αενάως (χρησιμοποιώντας άμεση αναδρομή).

```
TO TREXE_KYKLO_A :PLEYRA
  KYKLOS :PLEYRA 1
  TREXE_KYKLO_A :PLEYRA
END
```



Επίσης μπορούμε αυξάνοντας το μέγεθος της πλευράς του τριανταεξαγώνου-κύκλου με την διαδικασία TREXE_KYKLO_B να έχουμε το διπλανό σχήμα.

```
TO TREXE_KYKLO_B :PL :BHMA
  KYKLOS :PL 1
  MAKE "PL :PL + :BHMA
  TREXE_KYKLO_B :PL
: BHMA
END
```



Ανάλογα μπορούμε με τη διαδικασία PERISTREPSE_K να περιστρέψουμε αυτούς τους κύκλους.

```
TO PERISTREPSE_K :PLEYRA
  KYKLOS :PLEYRA 1
  MAKE "MOIRES (360/36)
  STROFH_DEXIA :MOIRES
  PERISTREPSE_K :PLEYRA
END
```

Εξετάσεις για δίπλωμα μοτοσυκλέτας

Ας γράψουμε μια διαδικασία η οποία με άμεσο αναδρομικό τρόπο να κινείται αενάως σχεδιάζοντας ένα οκτάρι.

Η διαδικασία MISO_OKTW

TO MISO_OKTW :PLEYRA

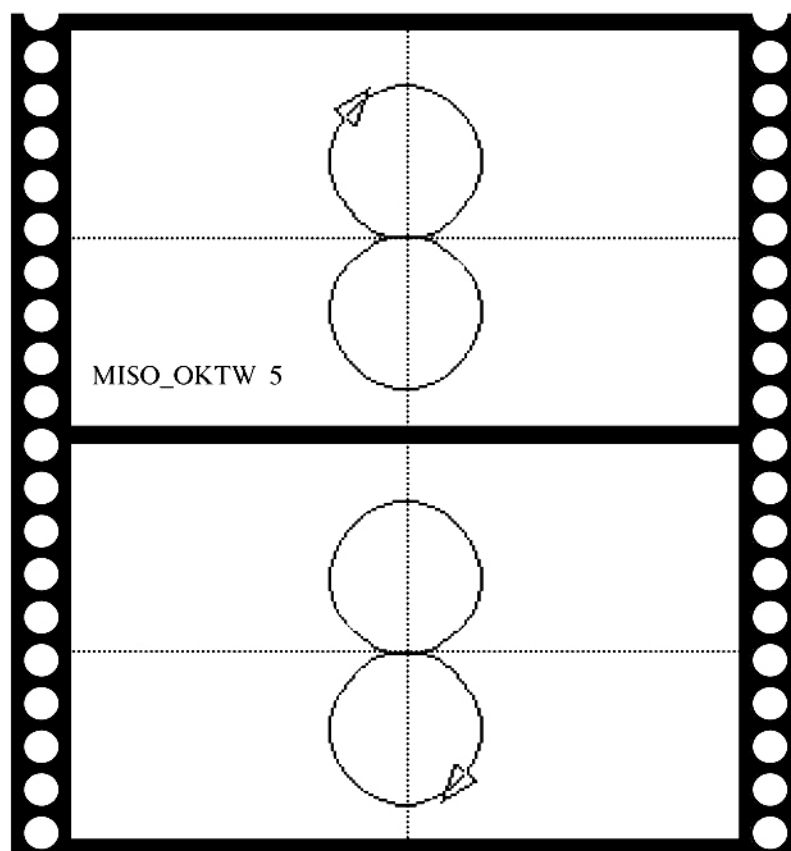
KYKLOS :PLEYRA 1

STROFH_DEXIA 180

MISO_OKTW :PLEYRA

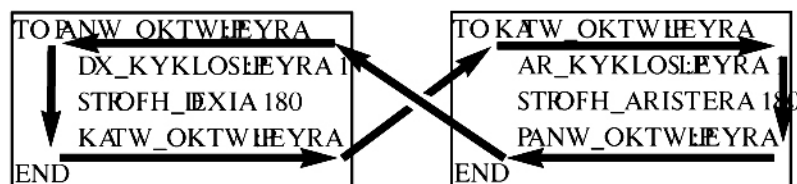
END

μας δίνει τα επόμενα στιγμιότυπα όπου όμως το 8 σχηματίζεται από δύο αριστερόστροφους κύκλους.



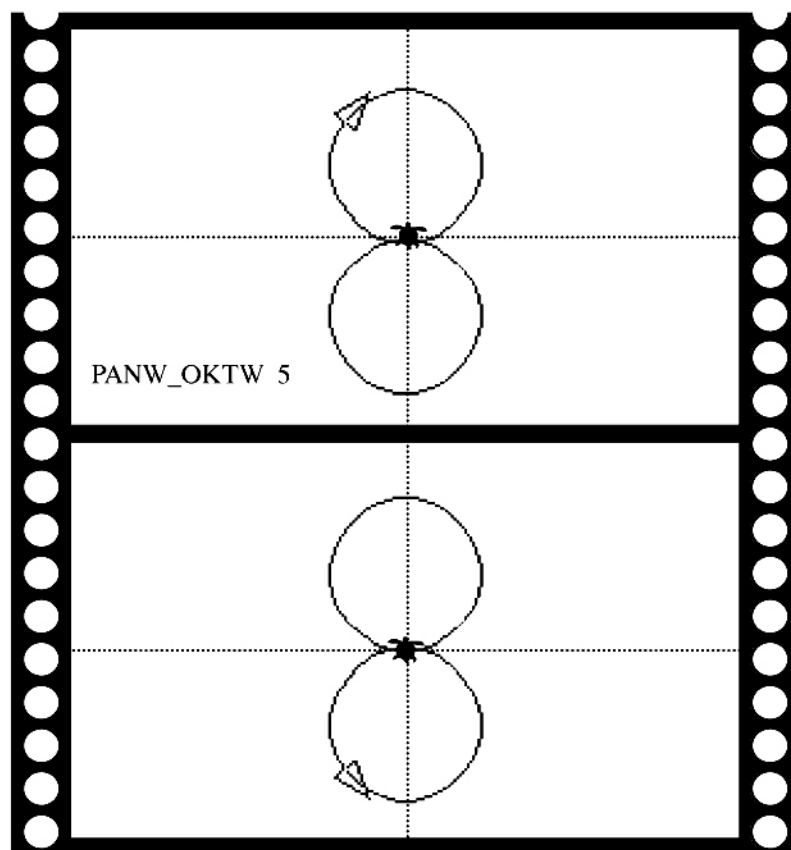
Κίνηση με έμμεσο αναδρομικό τρόπο

Για να πετύχουμε να σχηματίζεται το 8 με τον τρόπο που γράφουμε δηλ. με ένα αριστερόστροφο πάνω κύκλο και έναν δεξιόστροφο κάτω κύκλο θα χρειαστούμε δύο διαδικασίες, την PANW_OKTW και την KATW_OKTW,



οι οποίες κατορθώνουν να κινούν τη χελώνα καλώντας η μία την άλλη με έμμεσο αναδρομικό τρόπο.

Οι παραπάνω διαδικασίες χρησιμοποιούν τις διαδικασίες DX_KYKLOS και AR_KYKLOS που δίνονται παρακάτω:



```

TO DX_KYKLOS :PL :KLASMA
  STROFH_ARISTERA 90
  MAKE "MOIRES (360/36)
  REPEAT (:KLASMA*36) [DX_GRAMMH :MOIRES :PL]
  STROFH_DEXIA 90
END

```

```

TO AR_KYKLOS :PL :KLASMA
  STROFH_DEXIA 90
  MAKE "MOIRES (360/36)
  REPEAT (:KLASMA*36) [AR_GRAMMH :MOIRES :PL]
  STROFH_ARISTERA 90
END

```

οι οποίες με τη σειρά τους χρησιμοποιούν τις διαδικασίες
DX_GRAMMH και AR_GRAMMH:

```

TO DX_GRAMMH :MOIRES :MHKOS
  STROFH_DEXIA :MOIRES
  EYQEIA :MHKOS
END

```

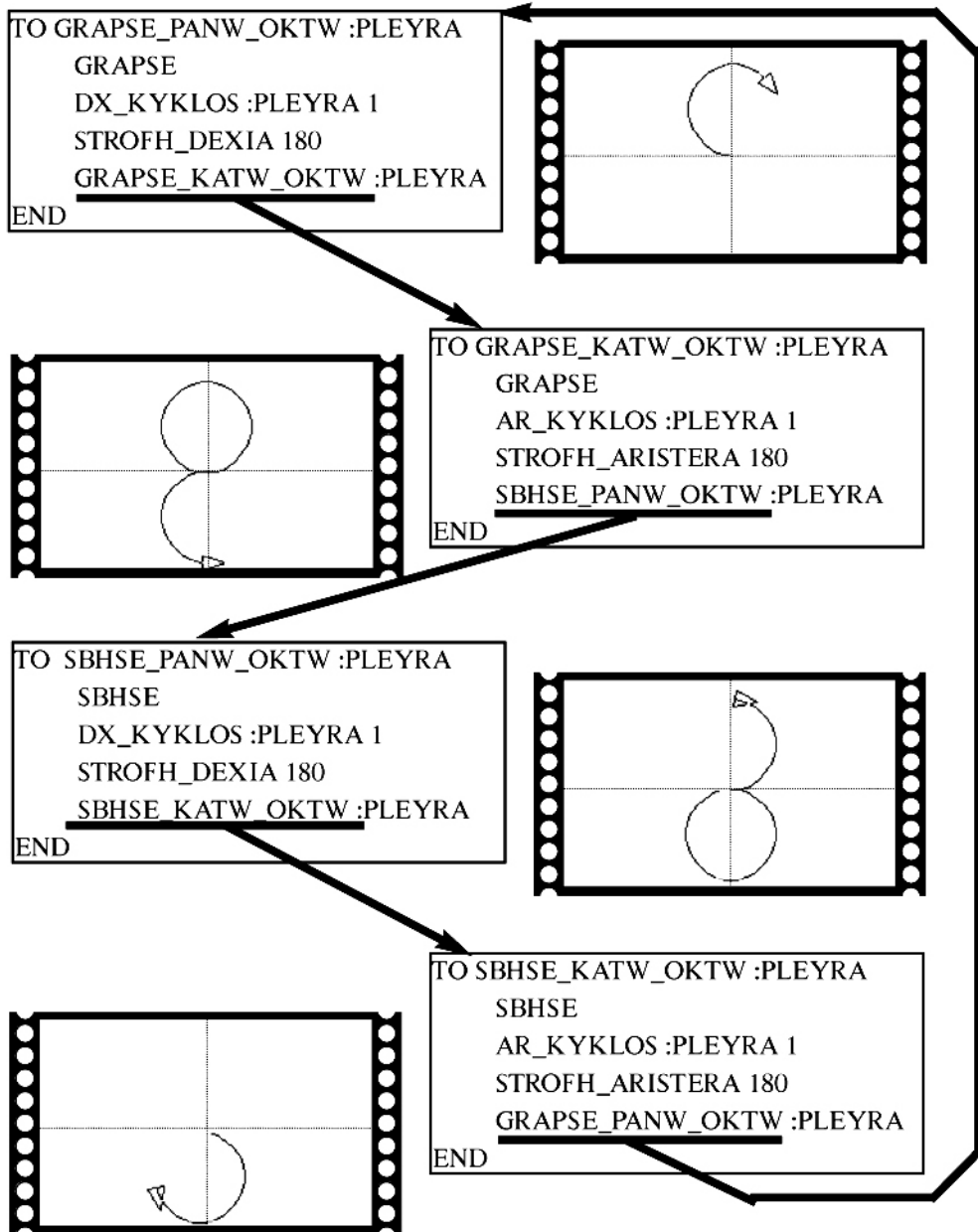
```

TO AR_GRAMMH :MOIRES :MHKOS
  STROFH_ARISTERA :MOIRES
  EYQEIA :MHKOS
END

```

Τετραπλή έμμεση αναδρομή

Ας φτιάξουμε τώρα μια σειρά τεσσάρων διαδικασιών οι οποίες καλώντας η μία την άλλη με έμμεσο αναδρομικό τρόπο να σχηματίζουν ένα οκτάρι και να το σβήνουν όπως φαίνεται στα επόμενα στιγμιότυπα.



14 fractals

Η έννοια του μετασχηματισμού

Με απλές αναδρομικές διαδικασίες μπορούν να δημιουργηθούν εντυπωσιακές εικόνες.

Η διαδικασία ΒΗΜΑ μετασχηματίζει ένα ευθύγραμμο τμήμα (η κατακόρυφη γραμμή στα αριστερά της οθόνης) σε μια τεθλασμένη γραμμή όπως η δεύτερη από αριστερά γραμμή της οθόνης. Ο μετασχηματισμός αυτός θα γίνεται και στα επιμέρους ευθύγραμμα τμήματα της δεύτερης γραμμής με αποτέλεσμα να έχουμε το σχήμα της τρίτης γραμμής κ.ο.κ. Η αναδρομική κλίση θα σταματάει όταν η τιμή της μεταβλητής GENIA πάρει την τιμή μηδέν.

TO ΒΗΜΑ :MEGEQOS :GENIA

IFELSE :GENIA = 0

[ΕΥΘΕΙΑ :MEGEQOS]

[ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)

ΑΡΙΣΤΕΡΑ 60

ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)

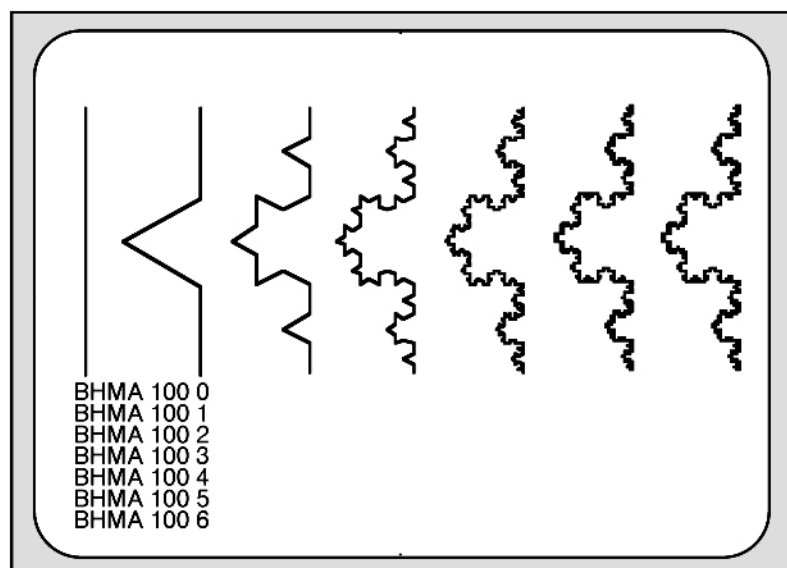
ΔΕΞΙΑ 120

ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)

ΑΡΙΣΤΕΡΑ 60

ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)]

END



«Το σύνολο Μάντελμπροτ είναι το πιο πολύπλοκο πράγμα στα μαθηματικά... Η καταγραφή των διαφορετικών σχημάτων που υπάρχουν σ' αυτό ή η αριθμητική περιγραφή της περιμετρικής του γραμμής φαίνεται πως θα απαιτούσε μια απειρία πληροφοριών. Αλλά εδώ βρίσκεται το παράδοξο: για να σταλεί μια πλήρης περιγραφή του συνόλου μέσα από μια γραμμή επικοινωνίας χρειάζονται λίγες μόνο δεκάδες χαρακτήρων κώδικα. Ένα μικρό πρόγραμμα υπολογιστή περιέχει αρκετές πληροφορίες για να αναπαράγει ολόκληρο το σύνολο...»
James Gleick

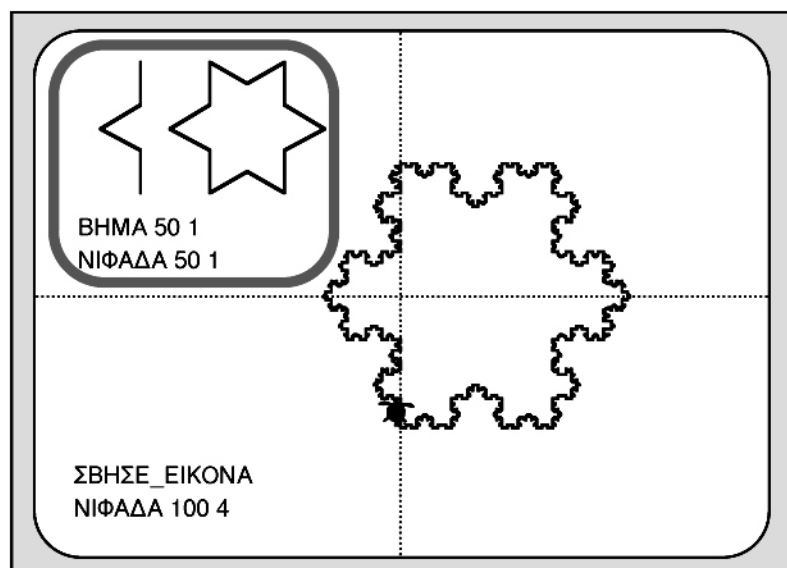
Μετασχηματίζοντας τις πλευρές τριγώνου

Ας θεωρήσουμε το τμήμα της τεθλασμένης γραμμής που φτιάξαμε προηγουμένως ως την πλευρά ενός ισόπλευρου τριγώνου και ας φτιάξουμε τη διαδικασία ΝΙΦΑΔΑ.

```
ΤΟ ΝΙΦΑΔΑ :MEGEQOS :GENIA  
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:MEGEQOS / 2)  
REPEAT 3  
  [ΒΗΜΑ :MEGEQOS :GENIA ΔΕΞΙΑ 120 ]  
END
```

Το σχήμα που σχηματίζεται ονομάζεται καμπύλη του Koch και είναι fractal.

Ο όρος fractal αποδίδεται στα ελληνικά ως "κλασματομορφικός" (Τάσος Μπουντής "Εισαγωγή στα Δυναμικά Συστήματα").

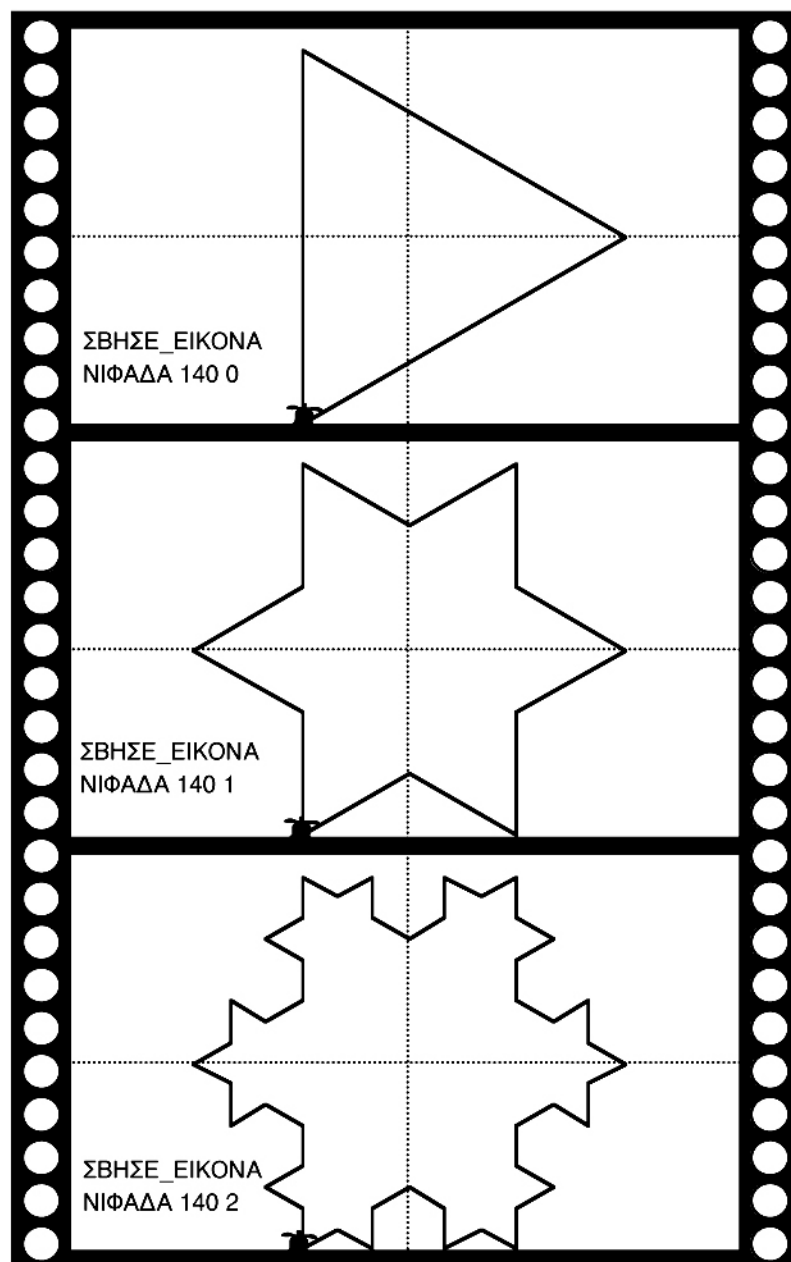


«Ένα χειμωνιάτικο απόγευμα του 1975, ο Μάντελμπροτ (Mandelbrot) κατέληξε ότι χρειαζόταν ένα όνομα για τα σχήματα, τις διαστάσεις και τη γεωμετρία του. Στο λατινικό λεξικό του γιού του συνάντησε το επίθετο fractus, από το ρήμα frangere: σπάω. Η ακουστική ομοιότητα με τις ομόρριζες αγγλικές λέξεις fracture και fraction (σπάω και τεμάχιο/κλάσμα) φαινόταν κατάλληλη. Ο Μάντελμπροτ επινόησε τη λέξη fractal, που είναι ουσιαστικό και επίθετο στην αγγλική και τη γαλλική γλώσσα».

James Gleick

Ο κ. Koch και μια ιδανική νιφάδα χιονιού (1)

Ας «τρέξουμε» μερικές φορές τη διαδικασία ΝΙΦΑΔΑ για διάφορες τιμές της παραμέτρου GENIA.



Έστω ένα
ισόπλευρο
τρίγωνο.
Διαιρέστε μια
πλευρά του στα
τρία. Με το
μεσαίο ένα τρίτο
της πλευράς και
προς το εξωτερικό
του τριγώνου,
σηματίστε ένα
άλλο ισόπλευρο
τρίγωνο.
Επαναλάβετε αυτή
την εργασία και
για τις άλλες δύο
πλευρές του
αρχικού τριγώνου.

Εφαρμόστε την
παραπάνω
διαδικασία σε
κάθε πλευρά του
σχήματος που
δημιουργήθηκε.

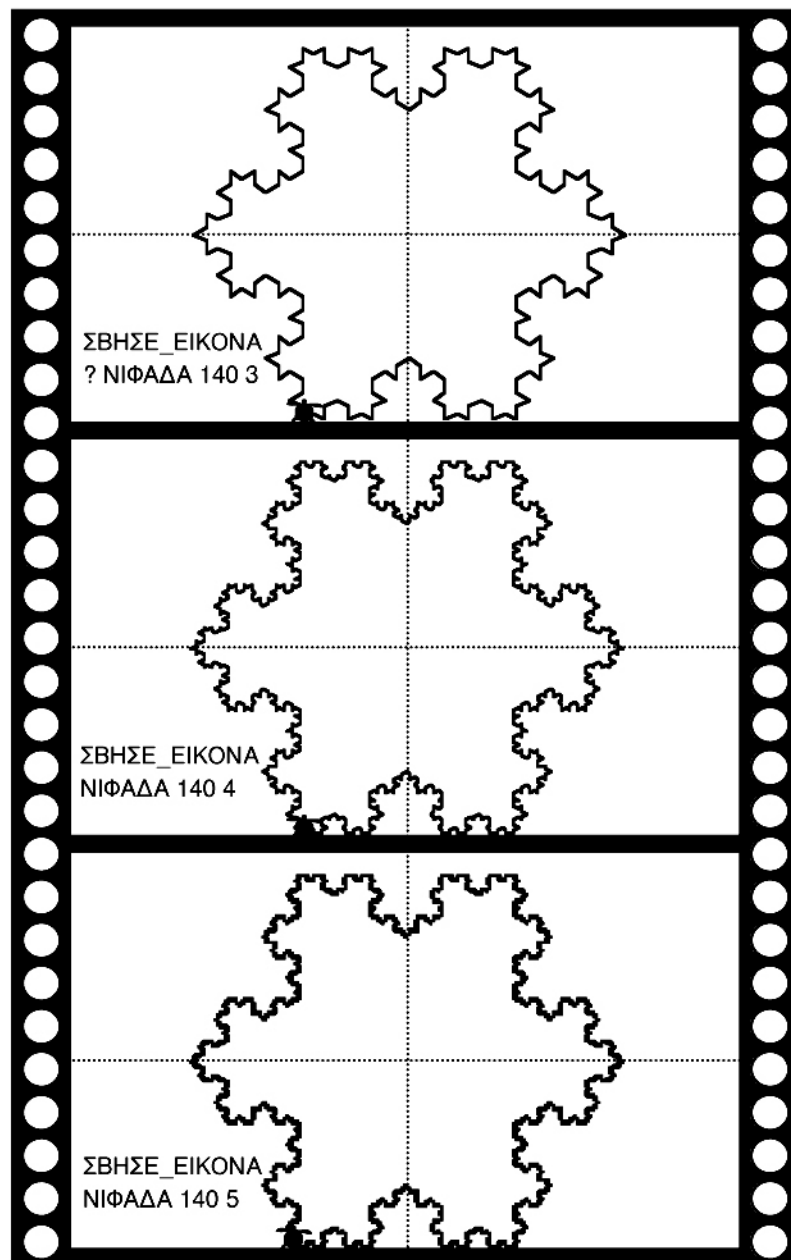
Το αποτέλεσμα
γίνεται όλο και
πιο λεπτομερές.

Εφαρμόστε την
παραπάνω
διαδικασία σε
κάθε πλευρά του
σχήματος που
δημιουργήθηκε.

Το αποτέλεσμα
γίνεται ακόμη πιο
λεπτομερές.

Ο κ. Koch και μια ιδανική νιφάδα χιονιού (2)

Συνεχίζοντας το τρέξιμο της διαδικασίας ΝΙΦΑΔΑ για μεγαλύτερες τιμές της GENIAS αυξάνει η πολυπλοκότητα των σχημάτων.



Συνεχίζοντας την προηγούμενη διαδικασία μέχρι το άπειρο παίρνουμε την καμπύλη Κοχ (από τον Σουηδό μαθηματικό Helge Von Koch). Η καμπύλη Κοχ έχει μερικά ενδιαφέροντα χαρακτηριστικά. Είναι μια συνεχής κλειστή γραμμή που δεν τέρνει ποτέ τον εαυτό της. Ενώ το μήκος της με κάθε μετασχηματισμό αυξάνει τείνοντας στο άπειρο, το εμβαδόν της αυξάνει ελάχιστα, αλλά το ολικό εμβαδόν παραμένει πάντα μικρότερο από το εμβαδόν του περιγεγραμμένου κύκλου στο αρχικό τρίγωνο. Έτσι έχουμε μια άπειρη σε μήκος γραμμή να περιβάλλει ένα πεπερασμένο εμβαδόν. Είναι κάτι περισσότερο από γραμμή αλλά και κάτι λιγότερο από επίπεδο. Έχει κάτι περισσότερο από μια διάσταση και κάτι λιγότερο από δύο διαστάσεις. Έχει κλασματική διάσταση ίση με 1,2618.

Ο κ. Peano και ο πονοκέφαλος των μαθηματικών

“... Άλλη μια βόμβα έσκαψε γύρω στο 1890 όταν ο Giuseppe Peano ανακάλυψε την «καμπύλη που γεμίζει το χώρο». Μια καμπύλη δεν είναι τίποτε περισσότερο από μια γραμμή που κάμπτεται και παραμορφώνεται, όπως ξέρει κάθε μαθητής η γραμμή είναι μονοδιάστατη. Οι μαθητικοί θεώρησαν δεδομένο από την κοινή λογική ότι κάθε καμπύλη, ανεξαρτήτως του πόσο κάμπτεται, πρέπει να είναι μονοδιάστατη.

Το επίπεδο (ένα κομμάτι χαρτί για παράδειγμα) είναι διδιάστατο. Το επίπεδο και η καμπύλη διακρίνονται πανεύκολα όσον αφορά τις διαστάσεις τους.

Εντούτοις, ο Peano κατασκεύασε μια καμπύλη που συστρεφόταν με τέτοιο πολύπλοκο τρόπο

ώστε γέμιζε

πράγματι ολόκληρο

το επίπεδο του

χαρτιού όπου

σχεδιάζόταν. Δεν

υπήρχε κανένα

σημείο πάνω στο

επίπεδο από το

οποίο να μην περνά

η γραμμή του Peano.

Αυτό οδήγησε μια καθόλου ενχάριστη

κατάσταση για τους μαθητικούς. Η

διδιάστατη φύση του επιπέδου

συνίσταται στο σύνολο των σημείων του.

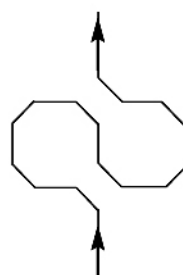
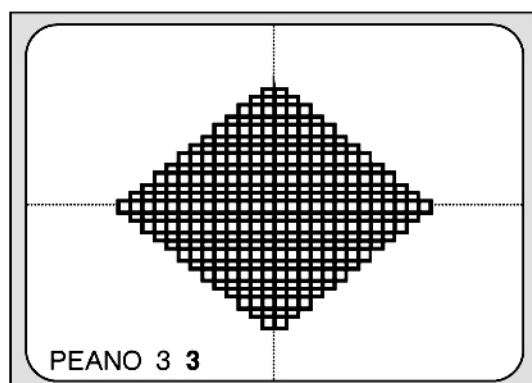
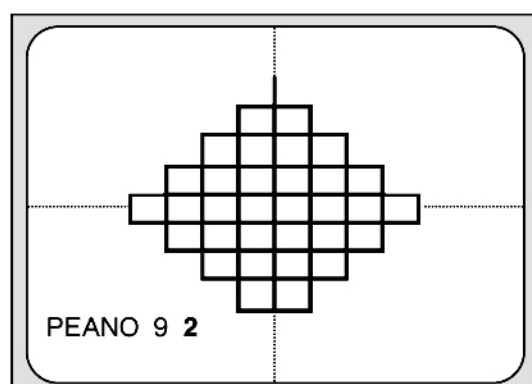
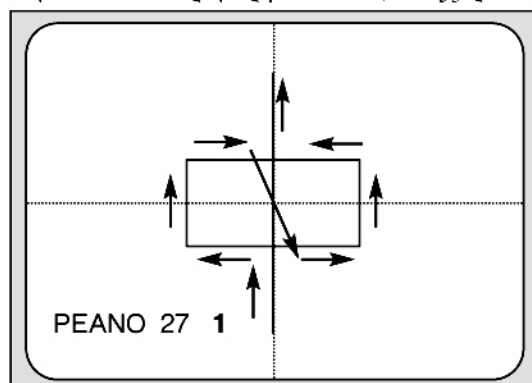
Τι θα σήμανε το να βρίσκονται επιπλέον

όλα αυτά τα σημεία πάνω σε μια

μονοδιάστατη γραμμή; Πως θα μπορούσε

ένα αντικείμενο να είναι μονοδιάστατο

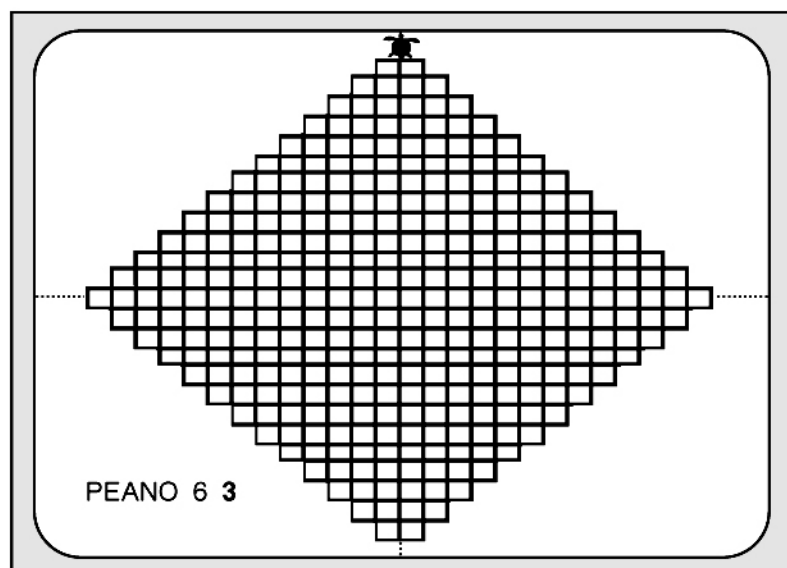
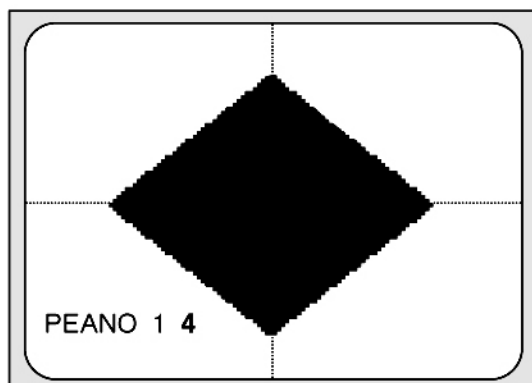
και επίσης διδιάστατο;” (Briggs & B).



Ο κ.Ρεανο και η καμπύλη που γεμίζει το χώρο

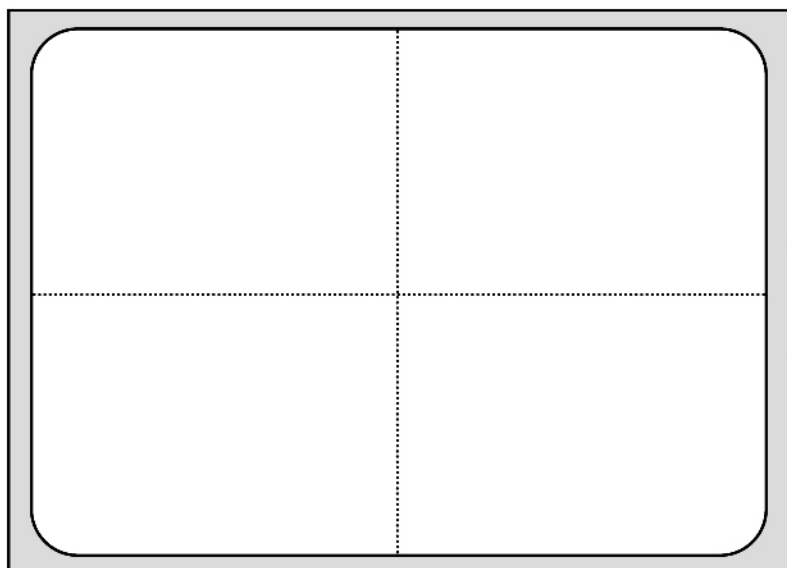
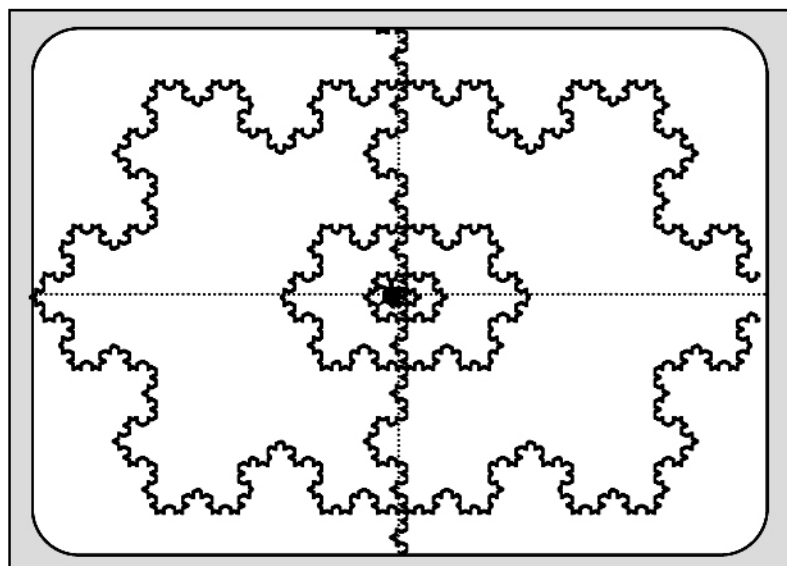
Ας καθοδηγήσουμε τη χελώνα ώστε να σχηματίσει την καμπύλη Peano χρησιμοποιώντας την τεχνική της άμεσης αναδρομής.

```
TO PEANO :BHMA :GENIA
IFELSE (:GENIA>0)
  [ PEANO (:BHMA) (:GENIA-1)
    GRAMMH 270 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 90 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 90 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 90 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 270 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 270 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 270 :BHMA
    PEANO (:BHMA) (:GENIA-1)
    GRAMMH 90 :BHMA
    PEANO (:BHMA) (:GENIA-1)
  ]
[ STOP ]
END
```



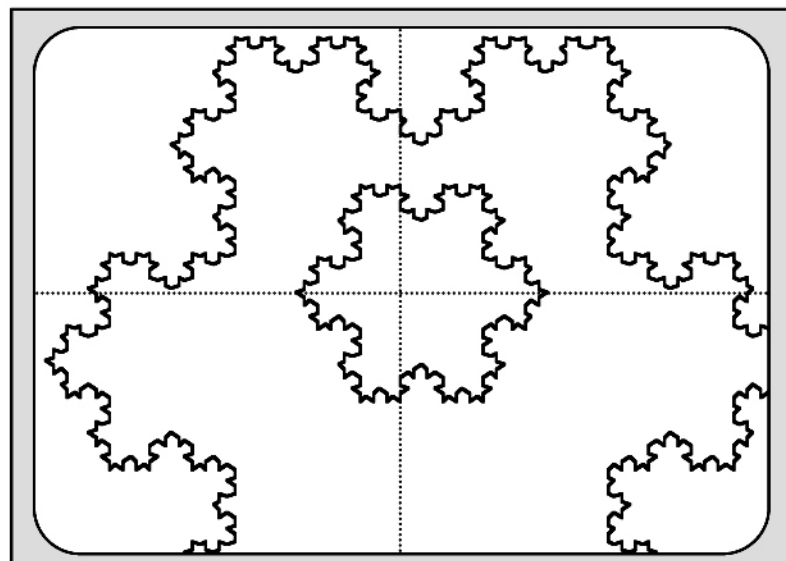
Η αυτο-ομοιότητα της καμπύλης Koch

Η ομοιότητα μεταξύ των επομένων πέντε καμπυλών Koch είναι προφανής και ανεξάρτητη της κλίμακας σχεδίασης.

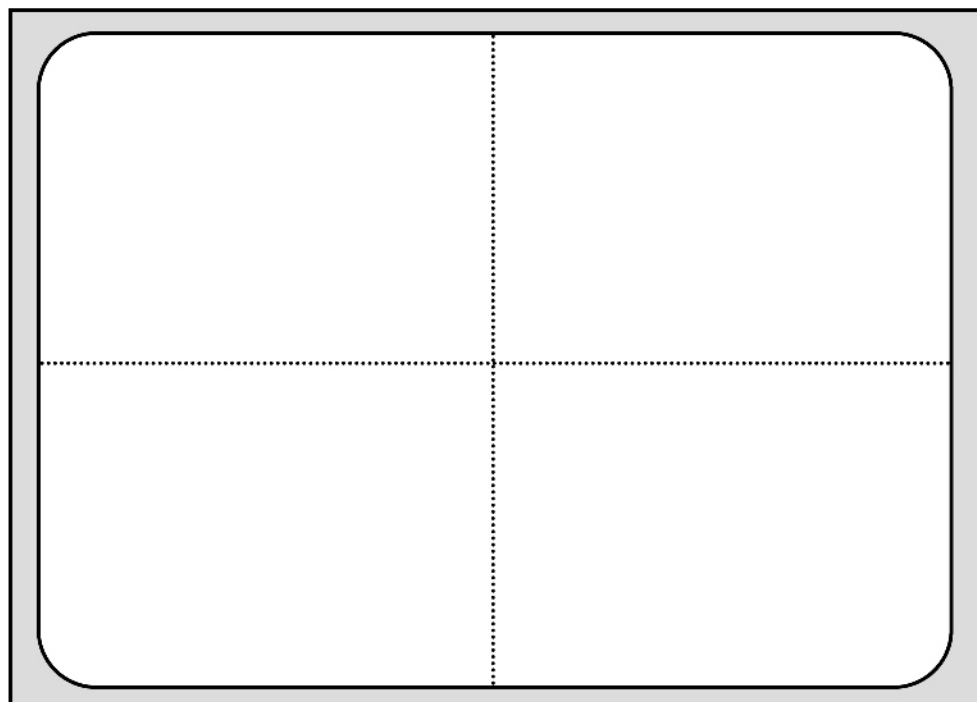


Ανακαλύπτοντας τη δομή (1)

Βασιζόμενοι στην ιδιότητα της αυτο-ομοιότητας μπορούμε να ανακαλύψουμε τους κανόνες που κρύβονται πίσω από ένα φράκταλ.

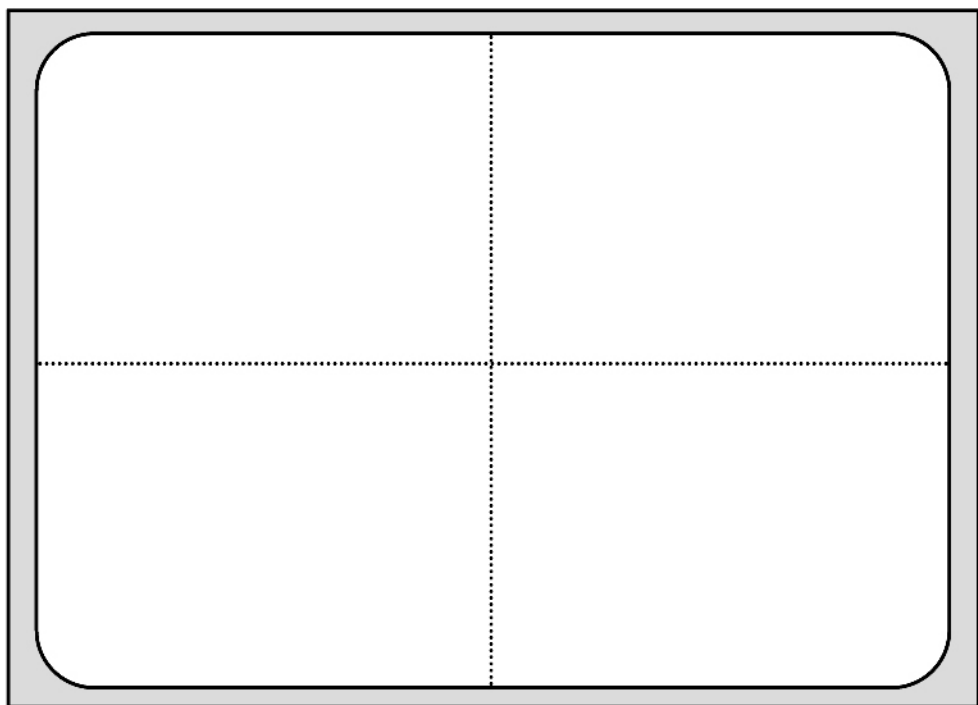
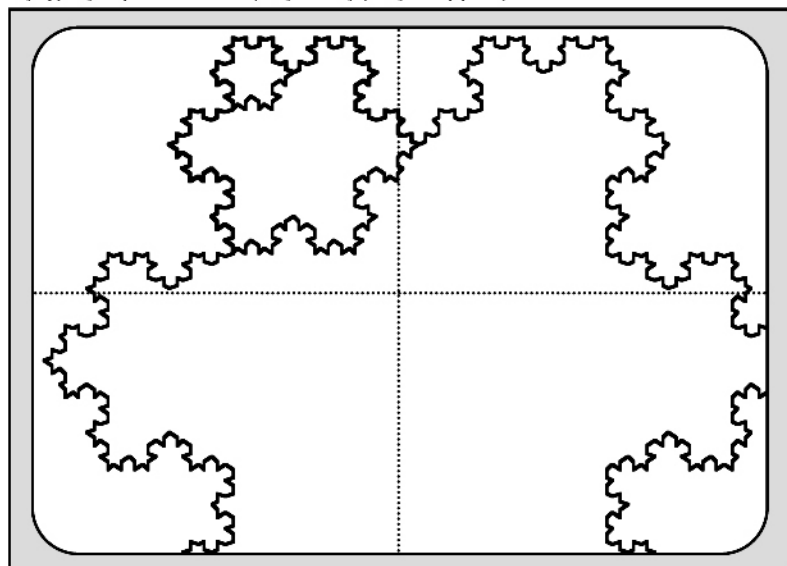


Με τη βοήθεια ενός σχεδιαστικού προγράμματος (του SuperPaint), μπορούμε να μετατοπίσουμε, αναπαράγουμε και εφαρμόσουμε τα φράκταλς που παίρνουμε από τη διαδικασία ΝΙΦΑΔΑ. Η τοποθέτηση των έξι μικρών φράκταλς στις άκρες του μεγάλου, μας βοηθούν να κατανοήσουμε τον τρόπο που δημιουργήθηκε και να ανακαλύψουμε τη δομή του.



Ανακαλύπτοντας τη δομή (2)

Λογω της αυτο-ομοιότητας η δομή του φράκταλ επιβεβαιώνεται όσο προχωρούμε σε λεπτομερέστερη προσέγγιση.



Μια εσωστρεφής καμπύλη Koch

Στη διαδικασία ΝΙΦΑΔΑ το μικρότερο τρίγωνο σχηματίζεται προς το εξωτερικό του μεγαλύτερου τριγώνου. Ας πειραματιστούμε κάνοντας το μικρότερο τρίγωνο να σχηματίζεται προς το εσωτερικό του μεγαλύτερου.

Αυτό μπορούμε να το πετύχουμε αντιμεταθέτοντας τις εντολές ΑΡΙΣΤΕΡΑ και ΔΕΞΙΑ στη διαδικασία ΒΗΜΑ, η οποία πλέον ξαναορίζεται.

ΤΟ ΒΗΜΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ

IFELSE :ΓΕΝΙΑ = 0

[ΕΥΘΕΙΑ :ΜΕΓΕΘΟΣ]

[ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)

ΔΕΞΙΑ 60

ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)

ΑΡΙΣΤΕΡΑ 120

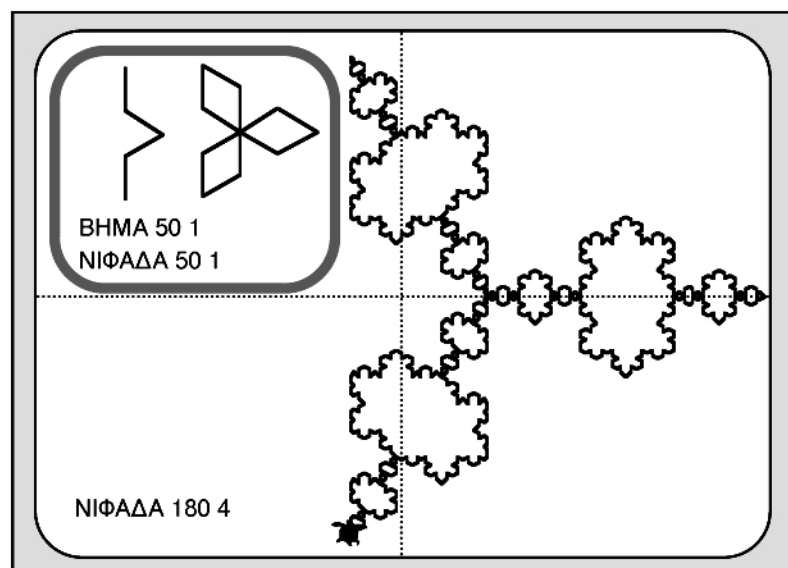
ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)

ΔΕΞΙΑ 60

ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)

]

END



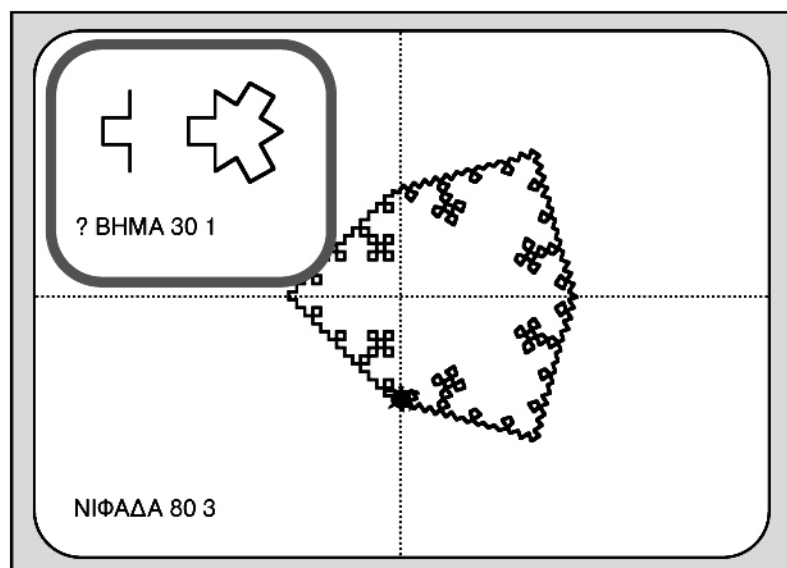
«Ανοίγοντας τις
πλούσιες
ευκαιρίες του
παιχνιδιού με το
άπειρο, η ομάδα
των ιδεών που
αντιπροσωπεύει
αυτή η
διαδικασία
φέρνει το παιδί
σε επαφή με κάτι
από το πως
είναι να είναι
κανείς
μαθηματικός...
Το εκπληκτικό
φαινόμενο που
παράγεται
συχνά δίνει
κίνητρα για
μακρινές
εξευνήσεις,
στις οποίες η
μαθηματική και
γεωμετρική
σκέψη
περιπλέκεται με
την αισθητική».

Seymour Papert

Ένας άλλος μετασχηματισμός

Αν αντί για τρίγωνο στο ένα τρίτο κάθε πλευράς σχηματίζεται ένα τετράγωνο, τότε το σχήμα που θα παραχθεί θα είναι διαφορετικό. Μερικές από τις ιδιότητες που είχαν τα προηγούμενα φράκταλς όπως η περιμετρική γραμμή τους να μην τέμνει τον εαυτό της, εδώ δεν ισχύουν.

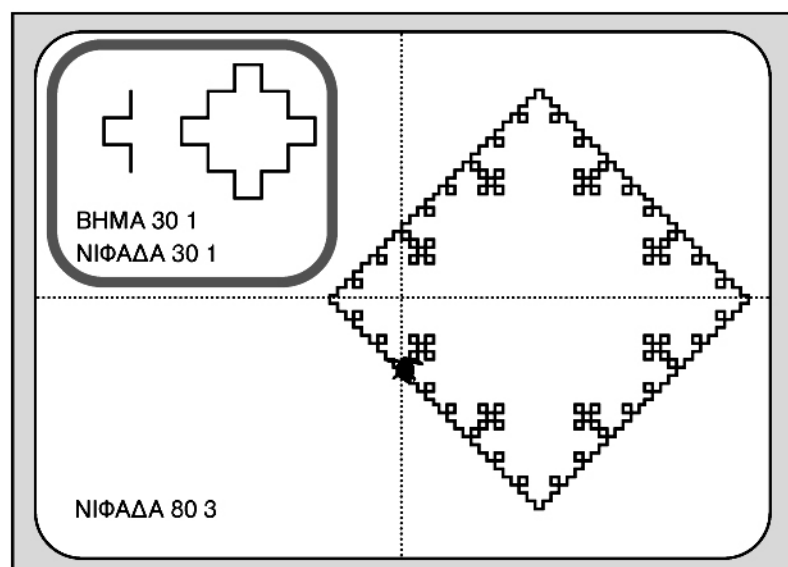
```
TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ]
END
```



Εξελίσσοντας τη νιφάδα

Αν στη διαδικασία ΝΙΦΑΔΑ βάλουμε την επανάληψη να γίνεται 4 αντί για 3 φορές και αντικαταστήσουμε τη γωνία της στροφής με 90 μοίρες, τότε το φράκταλ αντί να έχει τριγωνική θα έχει τετραγωνική μορφή.

```
ΤΟ ΝΙΦΑΔΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:ΜΕΓΕΘΟΣ / 2)
REPEAT 4
  [ ΒΗΜΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
    ΔΕΞΙΑ 90 ]
END
```

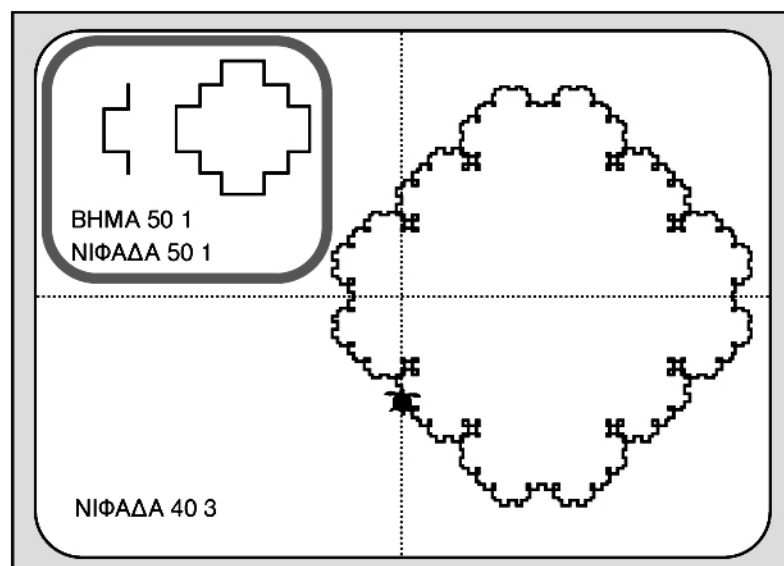


Το φράκταλ - δαντέλα

Για να μετασχηματίσουμε το αρχικό μήκος με την ακόλουθη τεθλασμένη γραμμή, θα χρειαστεί να χρησιμοποιήσουμε την παρακάτω διαδικασία. Η τεθλασμένη γραμμή ονομάζεται γεννήτορας του φράκταλ.

```

TO ΒΗΜΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
IFELSE :ΓΕΝΙΑ = 0
  [ ΕΥΘΕΙΑ :ΜΕΓΕΘΟΣ ]
  [ ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
    ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:ΜΕΓΕΘΟΣ / 3.0) (:ΓΕΝΙΑ - 1)
  ]
END
    
```



«Αν υπάρχει ένα μέτρο για την "αισθητική πραγματικότητα" ενός καλλιτεχνικού αντικειμένου, τότε το μέτρο αυτό εξαρτάται τόσο από την "πολυπλοκότητα" του υλικού που αναλώθηκε, όσο και από τη διάρθρωσή του, δηλ. την "τάξη"... Ο Birkhoff ερμήνευσε έτσι το "αισθητικό μέτρο" της μορφοποίησης σε συνάρτηση δύο μεταβλητών της "πολυπλοκότητας" και της "τάξης"... Αυτό που ο Birkhoff ονομάζει «πολυπλοκότητα» δεν είναι τίποτα άλλο παρά η στατιστική πληροφοριακή ποσότητα και αυτό που ονομάζει "τάξη" ανήκει στα πλεονασματικά χαρακτηριστικά που είναι απαραίτητα για να αναγνωρισθεί και κατανοηθεί μια "πληροφορία" σαν τέτοια».

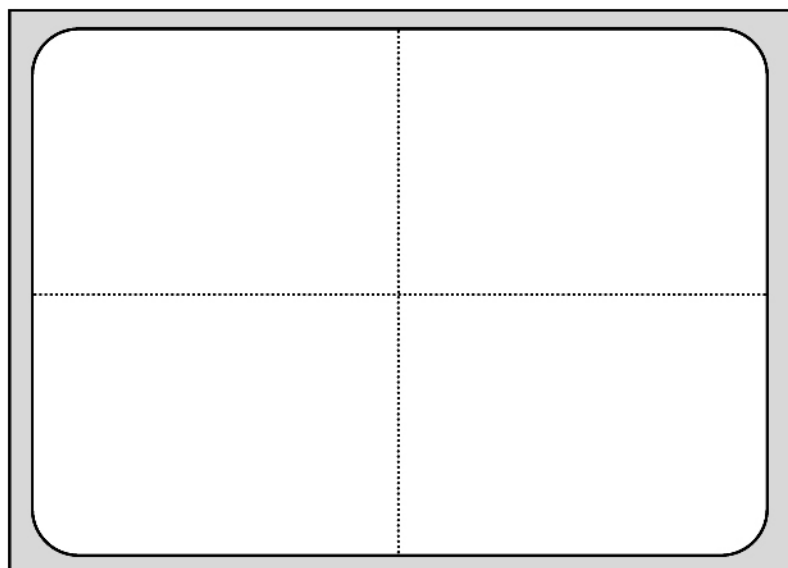
Max Bense

Ένα ακόμα φράκταλ

Ας χρησιμοποιήσουμε έναν ακόμα γεννήτορα για να φτιάξουμε το επόμενο φράκταλ. Ο γεννήτορας που θα χρησιμοποιήσουμε παρουσιάζεται στο διπλανό σχήμα.



```
TO ΒΗΜΑ :MEGEQOS :GENIA
IFELSE :GENIA = 0
  [ ΕΥΘΕΙΑ :MEGEQOS ]
  [ ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΑΡΙΣΤΕΡΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
    ΔΕΞΙΑ 90
    ΒΗΜΑ (:MEGEQOS / 3.0) (:GENIA - 1)
  ]
END
```



Η κότα κάνει το αυγό...

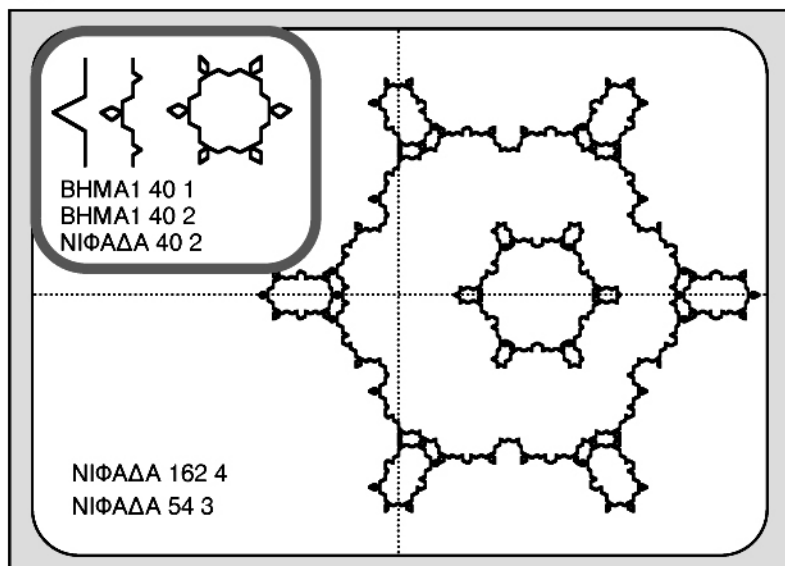
Μέχρι τώρα στις αναδρομικές διαδικασίες που χρησιμοποιήσαμε, μια διαδικασία καλεί τον εαυτό της (άμεση αναδρομή). Ας δοκιμάσουμε την τεχνική της έμμεσης αναδρομής όπου μια διαδικασία καλεί μια άλλη και αυτή με την σειρά της καλεί αυτή που την κάλεσε. Έτσι η διαδικασία ΒΗΜΑ1 καλεί τη διαδικασία ΒΗΜΑ2 και αυτή με τη σειρά της καλεί τη ΒΗΜΑ1.

Έστω ότι η ΒΗΜΑ1 χρησιμοποιεί ως γεννήτορα το εξωστρεφές τρίγωνο, ενώ η ΒΗΜΑ2 το εσωστρεφές.

TO ΒΗΜΑ1 :M :GN	TO ΒΗΜΑ2 :M :GN
IFELSE :GN = 0	IFELSE :GN = 0
[EYΘΕΙΑ :M]	[EYΘΕΙΑ :M]
[ΒΗΜΑ2 (:M / 3.0) (:GN - 1)	[ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
ΑΡΙΣΤΕΡΑ 60	ΑΡΙΣΤΕΡΑ 60
ΒΗΜΑ2 (:M / 3.0) (:GN - 1)	ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
ΔΕΞΙΑ 120	ΔΕΞΙΑ 120
ΒΗΜΑ2 (:M / 3.0) (:GN - 1)	ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
ΑΡΙΣΤΕΡΑ 60	ΑΡΙΣΤΕΡΑ 60
ΒΗΜΑ2 (:M / 3.0) (:GN - 1)	ΒΗΜΑ1 (:M / 3.0) (:GN - 1)
]]
END	END

```

TO ΝΙΦΑΔΑ :MEGEQOS :GENIA
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:MEGEQOS / 2)
REPEAT 3
  [ΒΗΜΑ1 :MEGEQOS :GENIA
  ΔΕΞΙΑ 120
  ]
END
  
```



... ή το αυγό την κότα;

Αν η διαδικασία ΝΙΦΑΔΑ χρησιμοποιεί τη διαδικασία ΒΗΜΑ2 αντί τη διαδικασία ΒΗΜΑ1, τότε το φράκταλ που θα προκύψει θα διαφέρει σημαντικά από το προηγούμενο.

```

ΤΟ ΝΙΦΑΔΑ :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
ΜΕΤΑΤΟΠΙΣΗ 0 MINUS (:ΜΕΓΕΘΟΣ / 2)
ΡΕΠΕΑΤ 3
  [ ΒΗΜΑ2 :ΜΕΓΕΘΟΣ :ΓΕΝΙΑ
    ΔΕΞΙΑ 120
  ]
ΕΝΔ

```

Το σχήμα που προκύπτει από τη συγκεκριμένη διαδικασία είναι ένα καλό παράδειγμα για την αυτο-ομοιότητα.

«...Πάνω από όλα, φράκταλ σημαίνει κάτι όμοιο με τον εαυτό του. Η αυτο-ομοιότητα είναι συμμετρία που συναντάμε σε όλες τις κλίμακες.

Σημαίνει αναδρομή, μορφή μέσα στη μορφή.

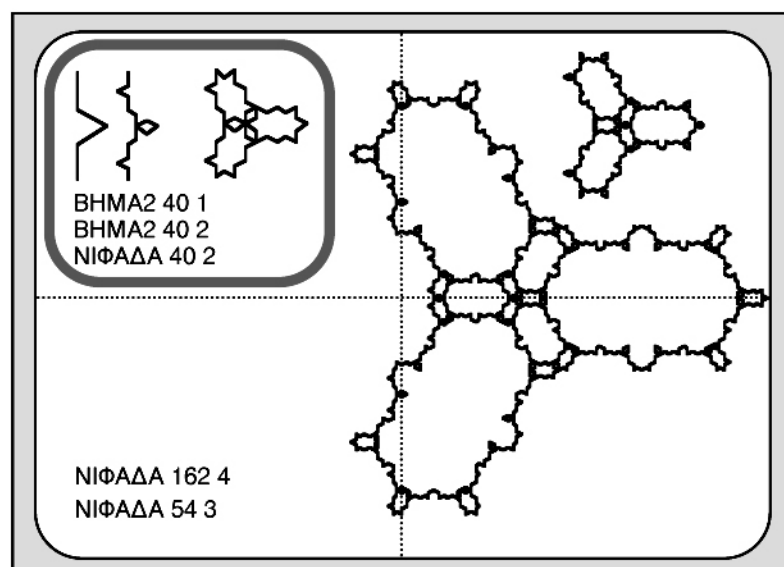
...σχήματα σαν την καμπύλη του Κοχ εμφανίζουν ομοιότητα με τον εαυτό τους γιατί

φαίνονται ακριβώς ίδια, ακόμα και σε μεγάλη

μεγέθυνση. Η αυτο-ομοιότητα βρίσκεται μέσα στην τεχνική της κατασκευής των καμπυλών - ο

ίδιος μετασχηματισμός επαναλαμβάνεται σε όλο και μικρότερες κλίμακες...»

James Gleick



15 Παράλληλος προγραμματισμός

Ψευδο-παράλληλος προγραμματισμός

Όταν μιλάμε για παράλληλο προγραμματισμό ενός υπολογιστή, προϋποτίθεται ότι ο υπολογιστής διαθέτει περισσότερους του ενός επεξεργαστές οι οποίοι θα εκτελούν συγχρόνως (παράλληλα) ο καθένας κάποιο δικό του τμήμα προγράμματος. Ένας υπολογιστής που διαθέτει μόνο ένα επεξεργαστή είναι δυνατό να δημιουργήσει στο χρήστη την αίσθηση ότι «τρέχει» αυτόχρονα περισσότερα από ένα προγράμματα, εκτελεί ένα λίγο το πρώτο πρόγραμμα, στη συνέχεια εκτελεί το δεύτερο... και μετά πάλι το πρώτο κοκ. Η εναλλαγή μεταξύ των προγραμμάτων πρέπει να είναι τόσο γρήγορη ώστε ο χρήστης να μην την αντιλαμβάνεται. Τότε λέμε ότι έχουμε ψευδο-παράλληλο προγραμματισμό. Ένα παράδειγμα ψευδο-παράλληλου προγραμματισμού στο περιβάλλον της `Logo` δούμε παρακάτω.

Στη `Logo` εκτός από τη βασική χελώνα είναι δυνατό να έχουμε και άλλες. Η βασική χελώνα αριθμείται ως 0 ενώ οι άλλες είναι οι 1, 2, 3 κλπ. με το πλήθος τους να εξαρτάται από την έκδοση της `Logo`.

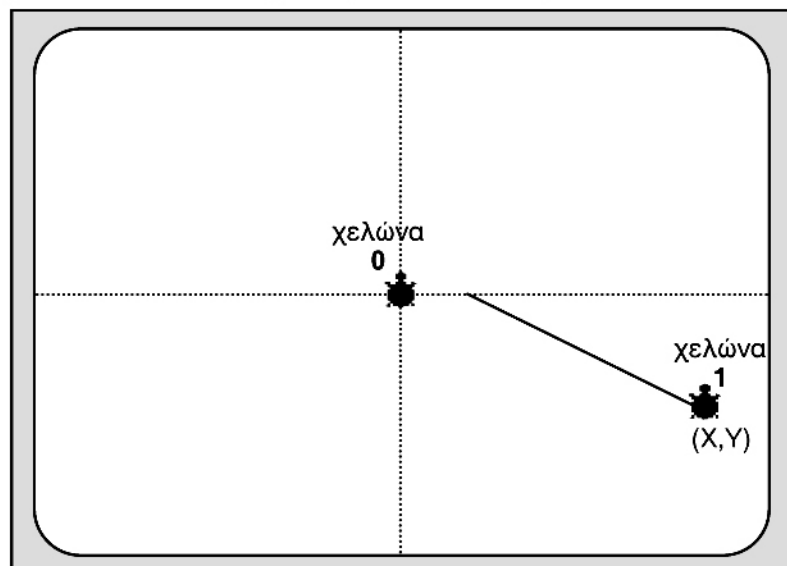
Με την εντολή `TELL` απευθυνόμαστε στη συγκεκριμένη χελώνα που θέλουμε. Για παράδειγμα αν θέλουμε να απευθυνθούμε στη χελώνα 1 και στη συνέχεια να της ζητήσουμε να πάει στις συντεταγμένες (X, Y) θα πρέπει να κάνουμε τα εξής:

```
TELL 1
```

```
SETX :X
```

```
SETY :Y
```

Αυτό έχει ως αποτέλεσμα εμφανισθεί μια επιπλέον χελώνα (δίπλα στην χελώνα 0) και στη συνέχεια να μετατοπισθεί στις θέσης (X, Y).



Η χελώνα σε ρόλο χαμαιλέοντα

Η χελώνα της logo (ο δείκτης της θέσης στην οθόνη δηλαδή) μπορεί να πάρει διάφορες μορφές εκτός από το εξ ορισμού (που είναι η χελώνα). Αυτά τα σχήματα ποικίλουν από έκδοση σε έκδοση της logo. Στον επόμενο πίνακα παρουσιάζονται οι μορφές που μπορεί να πάρει ο δείκτης της οθόνης της logoWriter.

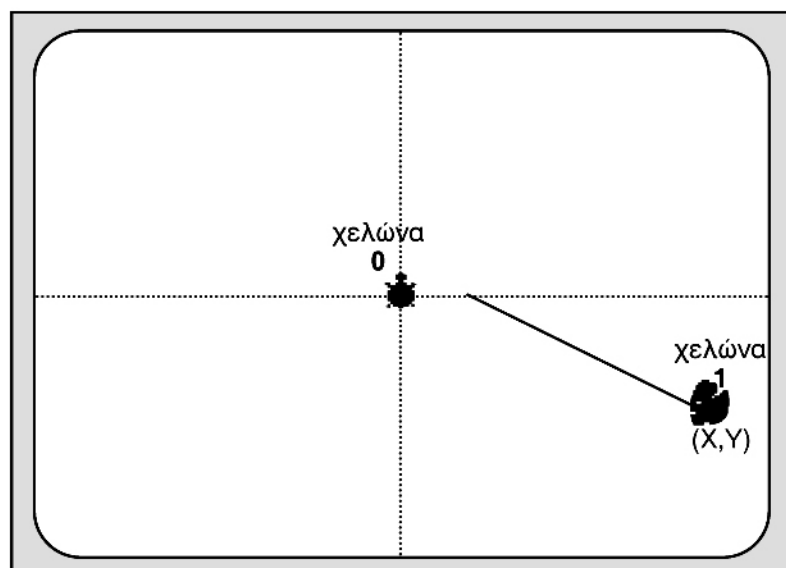
1	1	2	2	3	3	4	4	5	5	6	6
7	7	8	8	9	9	10	0	11	■	12	●
13	←	14	♥	15	😊	16	♂	17	♀	18	↗
19	↖	20	🏠	21	🐱	22	🐶	23	🌳	24	🍀
25	🚲	26	🚗	27	🚗	28	🚗	29	👥	30	

Αν θα θέλαμε η χελώνα 1 να πάρει άλλη μορφή τότε θα πρέπει να χρησιμοποιήσουμε την εντολή SETSH. Για παράδειγμα αν θέλαμε να πάρει το σχήμα του λαγού (στη συγκεκριμένη έκδοση της logoWriter) θα χρησιμοποιήσουμε τις εντολές:

TELL 1

SETSH 22

Το αποτέλεσμα φαίνεται παρακάτω:



Λάβετε θέσεις, έτοιμοι...

Ας φτιάξουμε μια διαδικασία που θα βάζει το λαγό ή τη χελώνα στη θέση εκκίνησης για τον αγώνα.

Η διαδικασία EKKIN :ANTK :X :Y :SXHM μετατοπίζει, χωρίς να αφήνει ίχνη, την υπ' αριθμό :ANTK χελώνα στη θέση με συντεταγμένες X και Y, τη στρέφει προς τα αριστερά και της δίνει το σχήμα SXHM.

TO EKKIN :ANTK :X :Y :SXHM

TELL :ANTK

PU

SETX :X

SETY :Y

PD

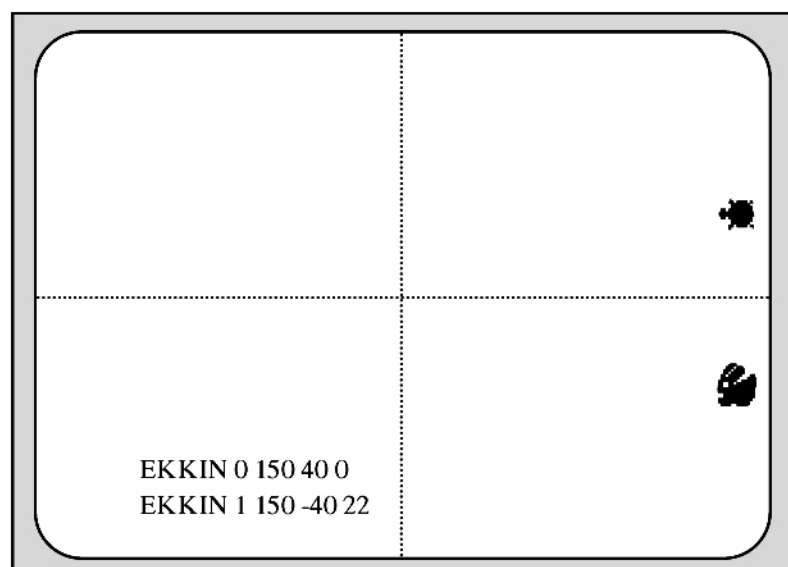
LT 90

SETSH :SXHM

ST

END

Αν τη καλέσουμε δύο φορές με τις αντίστοιχες τιμές θα βάλει το λαγό και τη χελώνα στην εκκίνηση.



Φύγαμε...

Αν θέλουμε να φτιάξουμε ένα πρόγραμμα που να προσομοιώνει τον αγώνα του λαγού με τη χελώνα θα πρέπει να κινούμε το λαγό αλλά και τη χελώνα ταυτόχρονα. Όπως αναφέρθηκε παραπάνω η αίσθηση του «ταυτόχρονα» επιτυγχάνεται προγραμματίζοντας τη χελώνα να κάνει μια κίνηση, το λαγό να κάνει και αυτός μια κίνηση και επαναλαμβάνοντας αυτή τη διαδοχή των κινήσεων μέχρι κάποιος από τους δύο να φτάσει στο τέλος.

Ας δούμε λοιπόν τη διαδικασία KINSHS που θα κινεί διαδοχικά το λαγό και τη χελώνα.

Ας φτιάξουμε το πρόγραμμα που θα προσομοιώνει το τρέξιμο του λαγού και της χελώνας κάνοντάς τα να προχωρούν κάθε φορά κάποια “τυχαία” απόσταση.

ΤΟ KINSHS

TELL 0

MAKE “BX RANDOM 4

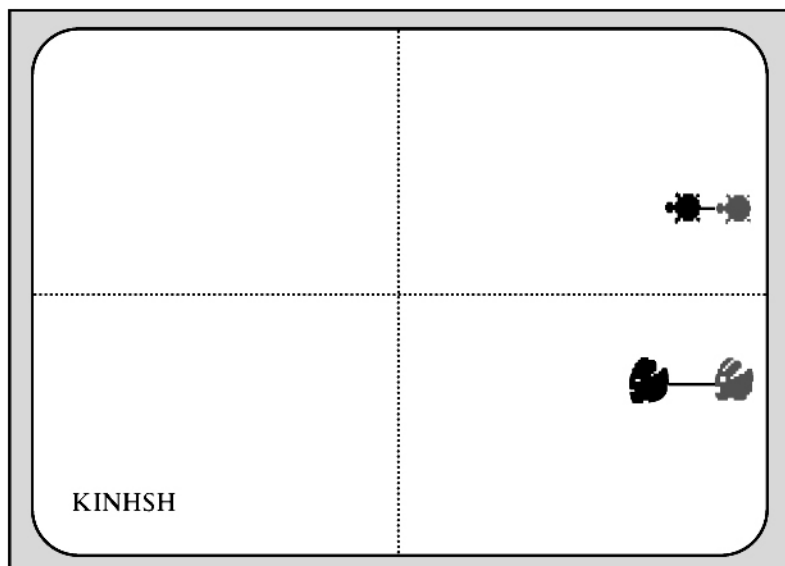
FD :BX

TELL 1

MAKE “BL RANDOM 10

FD :BL

END

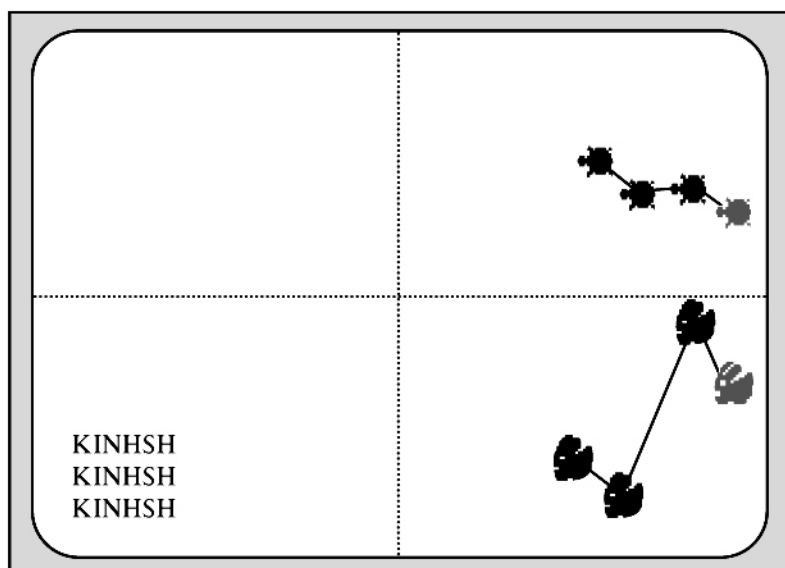


«Κίνηση Brawn»

Για να γίνει πιο ενδιαφέρων ο αγώνας ας κάνουμε το κάθε δρομέα να αποκλίνει (όπως θα έκανε ένα ζώο) κάθε φορά από τη σωστή κατεύθυνση κατά μια “τυχαία” γωνία. Η διαδικασία KINHSH γίνεται:

```
TO KINHSH
  TELL 0
  MAKE "GX RANDOM 10
  IFELSE :PKX = 1
    [RT :GX MAKE "PKX 0]
    [LT :GX MAKE "PKX 1]
  MAKE "BX RANDOM 4
  FD :BX
  TELL 1
  MAKE "GL RANDOM 30
  IFELSE :PKL = 1
    [RT :GL MAKE "PKL 0]
    [LT :GL MAKE "PKL 1]
  MAKE "BL RANDOM 10
  FD :BL
END
```

Η μεταβλητές PKX και PKL χρησιμοποιούνται για να κάνουν τους αγωνιζόμενους τη μια φορά να αποκλίνουν προς τα δεξιά και την άλλη προς τα αριστερά.



Το τρέξιμο

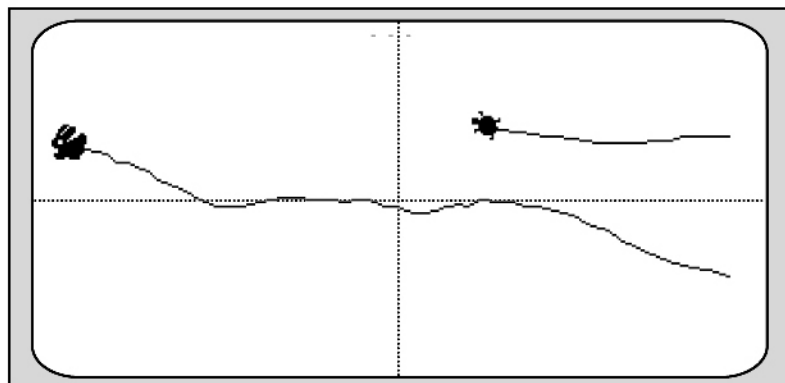
Η διαδικασία KINSHH κινεί τους αγωνιζόμενους μόνο κατά ένα βήμα. Αν θέλουμε να έχουμε μια διαδικασία που να περιγράφει όλο το τρέξιμο του αγώνα θα πρέπει να τροποποιήσουμε τη διαδικασία KINSHH έτσι ώστε με αναδρομικό τρόπο να “επαναλαμβάνεται” μέχρι ο λαγός ή η χελώνα να φτάσει στο τέλος. Έτσι προκύπτει η διαδικασία TREXIMO.

ΤΟ TREXIMO

```
TELL 0
MAKE "GX RANDOM 10
IFELSE :PKX = 1
  [RT :GX MAKE "PKX 0]
  [LT :GX MAKE "PKX 1]
MAKE "BX RANDOM 4
FD :BX
TELL 1
MAKE "GL RANDOM 30
IFELSE :PKL = 1
  [RT :GL MAKE "PKL 0]
  [LT :GL MAKE "PKL 1]
MAKE "BL RANDOM 10
FD :BL
```

```
TELL 0
MAKE "XXEL XCOR
TELL 1
MAKE "XLAG XCOR
MAKE "TELOS OR :XXEL < -150 :XLAG < -150
IF NOT :TELOS [TREXIMO]
```

END



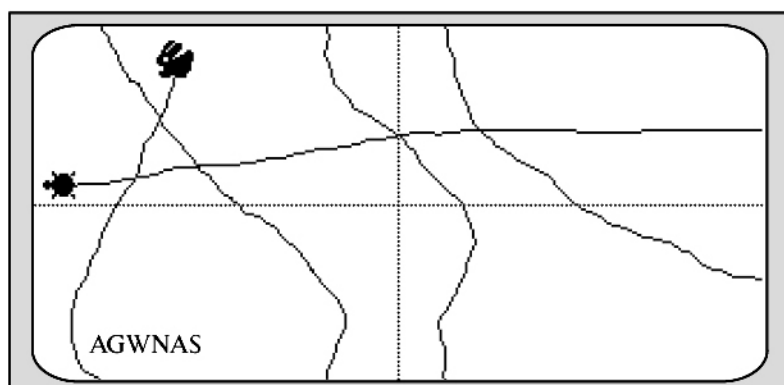
Νυν υπερ πάντων αγών...

Το κύριο πρόγραμμα που θα βάζει το λαγό και τη χελώνα στην αφετηρία και θα καλεί την διαδικασία TREXIMO είναι το παρακάτω:

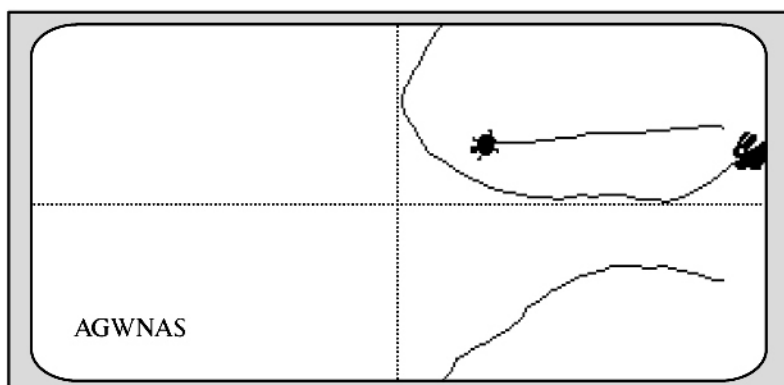
```
TO AGWNAS
  EKKIN 0 150 40 0
  EKKIN 1 150 -40 22
  MAKE "PKX 1
  MAKE "PKL 1
  TREXIMO
```

END

Η έξαψη του αγώνα καθορίζεται από πολλούς τυχαίους παράγοντες και έτσι άλλες φορές βλέπουμε το λαγό να νικά (όπως προηγουμένως) ...



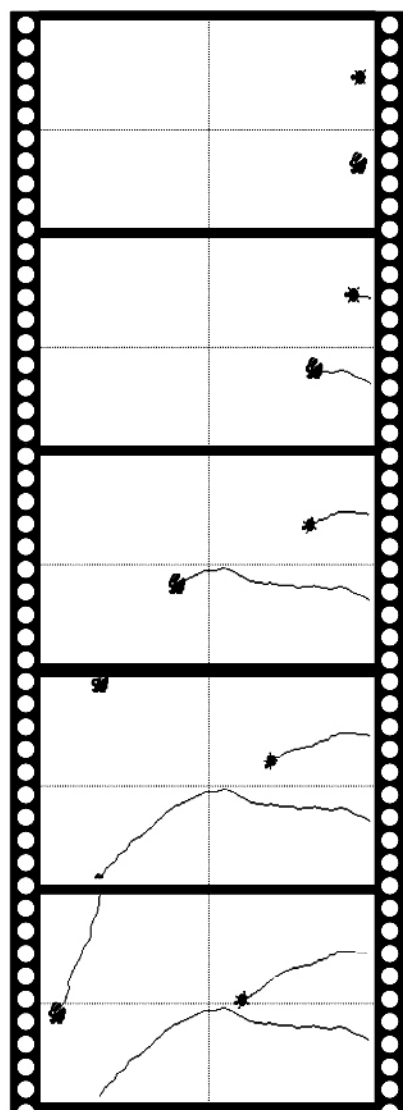
...και άλλες φορές νικά η χελώνα, αφού ο λαγός αν και ταχύτερος αποπροσανατολίζεται δίνοντας στην χελώνα την ευκαιρία να τερματίσει ή χάνει τον αγώνα επιστρέφοντας στην αφετηρία.



«Οι μαθητές δεν μπορούν να κάνουν κτήμα τους κάτι που δεν το γνωρίζουν. Γι' αυτό είναι προαπαιτούμενο να βρούμε τρόπους να διευκολύνουμε την προσωπική ιδιοποίηση όχι μόνο της νευτώνειας κίνησης και των νόμων που την περιγράφουν, αλλά επίσης της γενικής έννοιας των νόμων.

Αυτό το κάνουν σχεδιάζοντας μια σειρά από μικρόκοσμους. Ο σχεδιασμός του μικρόκοσμου τον κάνει "φυτώριο" ενός συγκεκριμένου είδους δυναμικών ιδεών και διανοητικών δομών».

Seymour Papert



16 Λίστες, μια δυναμική δομή της logο

Υλοποίηση δομής στοίβας με λίστες

Το φυσικό πρόβλημα:

Θέλουμε να προγραμματίσουμε έναν υπολογιστή ώστε να αυτοματοποιεί τη διαδικασία εισόδου - εξόδου (παρκάρισμα - ξεπαρκάρισμα) σε ένα ιδιόμορφο γκαράζ. Το γκαράζ είναι στενόμακρο, με μια λωρίδα και μια πόρτα (είσοδο / έξοδο). Επίσης δίπλα από το γκαράζ υπάρχει ένα όμοιο πάρκινγκ για να διευκολύνει τις манοῦβρες των αυτοκινήτων. Όταν έρχεται ένα αυτοκίνητο απλά χρειάζεται να το βάλουμε στο γκαράζ πίσω από το τελευταίο αυτοκίνητο που είναι παρκαρισμένο. Όταν έρχεται κάποιος και ζητάει το αυτοκίνητό του τότε θα πρέπει να βγάλουμε ένα ένα τα αυτοκίνητα (βάζοντάς τα προσωρινά στο πάρκινγκ) μέχρι να βγάλουμε το σωστό. Αφού παραδόσουμε το ζητούμενο αυτοκίνητο πρέπει να μεταφέρουμε τα αυτοκίνητα που απέμειναν στο πάρκινγκ ξανά στο γκαράζ.

Το γκαράζ και το πάρκινγκ μπορούμε να τα προσομοιώσουμε με δομές στοίβας που θα τις υλοποιήσουμε με λίστες. Μια στοίβα (stack) είναι μια δομή δεδομένων όπου ό,τι μπαίνει τελευταίο βγαίνει πρώτο (Last In First Out, LIFO).

Κατ' αρχήν ας γράψουμε τη διαδικασία INITG που δημιουργεί μια άδεια λίστα με όνομα GARAGE και τη διαδικασία INITP που δημιουργεί μια άδεια λίστα με όνομα PARKING.

TO INITG

MAKE "GARAGE []

END

TO INITP

MAKE "PARKING []

END



το parking

Η διαδικασία **BALEG** βάζει στη στοίβα **GARAGE** ως πρώτο στοιχείο (**FPUT** = **FirstPUT**) ό,τι υπάρχει στη μεταβλητή **IMPG**. Όμοια η διαδικασία **BALEP** βάζει στη στοίβα **PARKING** ως πρώτο στοιχείο ό,τι υπάρχει στη μεταβλητή **IMPP**.

TO **BALEG** :**IMPG**

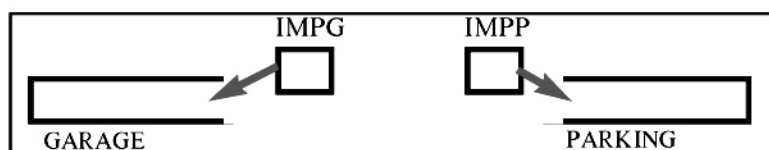
MAKE "GARAGE **FPUT** :**IMPG** :GARAGE

END

TO **BALEP** :**IMPP**

MAKE "PARKING **FPUT** :**IMPP** :PARKING

END



Η διαδικασία **BGALEG** βγάζει από τη στοίβα **GARAGE** το στοιχείο που μπήκε τελευταίο και το φυλάει στη μεταβλητή **EXPG** αφού εξετάσει (με τη συνθήκη **EMPTY?**) αν η λίστα :**GARAGE** είναι άδεια.

Η εντολή **MAKE "EXPG FIRST :GARAGE**

βάζει στη μεταβλητή **EXPG** ένα αντίγραφο του πρώτου στοιχείου που υπάρχει στη λίστα :**GARAGE**

ενώ η εντολή **MAKE "GARAGE BF :GARAGE**

βάζει στη λίστα "GARAGE όλα τα στοιχεία της λίστας :**GARAGE** εκτός από το πρώτο (**BF** = **ButFirst**).

Όμοια η διαδικασία **BGALEP** βγάζει από τη στοίβα **PARKING** το στοιχείο που μπήκε τελευταίο και το φυλάει στη μεταβλητή **EXPP**.

TO **BGALEG**

IFELSE **EMPTY?** :**GARAGE**

[**PR** [το γκαράζ είναι άδειο]]

[**MAKE "EXPG FIRST :GARAGE**

MAKE "GARAGE BF :GARAGE]

END

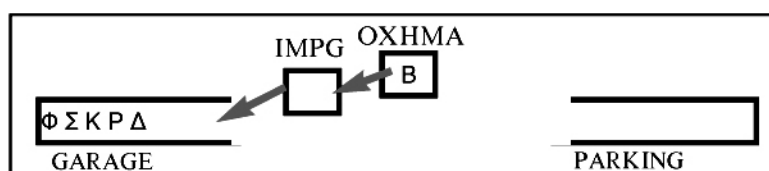
to parking

```
TO BGALEP
  IFELSE EMPTY? :PARKING
    [PR [ADEIO PARKING]]
    [MAKE "EXPP LAST :PARKING
      MAKE "PARKING BL :PARKING]
END
```



Ας δούμε πως θα φτιάξουμε τη διαδικασία EISHGAGE που θα πάρει από το πληκτρολόγιο τον κωδικό ενός αυτοκινήτου και θα τον βάλει στο γκαράζ.

```
TO EISHGAGE
  PR [Δώσε τον κωδικό του οχήματος (χαρακτήρα)]
  MAKE "ΟΧΗΜΑ READCHAR
  BALEG :ΟΧΗΜΑ
END
```



Η διαδικασία TYPWSEG εμφανίζει στην οθόνη όλα τα στοιχεία της λίστας GARAGE. Όμοια η διαδικασία εμφανίζει όλα τα στοιχεία της λίστας PARKING.

```
TO TYPWSEG
  SHOW :GARAGE
END
```

```
TO TYPWSEP
  SHOW :PARKING
END
```

to parking

Η διαδικασία TYPWSE εμφανίζει στην οθόνη τα περιεχόμενα και των δύο λιστών

```
TO TYPWSE
    SHOW [περιεχόμενο γκαράζ]
    TYPWSEG
    SHOW [περιεχόμενο πάρκινγκ]
    TYPWSEP
END
```

Ας δούμε πως μπορούμε να βγάλουμε από το γκαράζ το αυτοκίνητο που ζητάει κάποιος. Η διαδικασία EXHGAGE αφού ζητήσει να μάθει ποιο όχημα ζητείται, ελέγχει αν αυτό υπάρχει στο γκαράζ με τη συνθήκη **MEMBER?** :OXHMA :GARAGE και αν υπάρχει τότε καλεί την διαδικασία PARE που αναλαμβάνει να παραδώσει το όχημα.

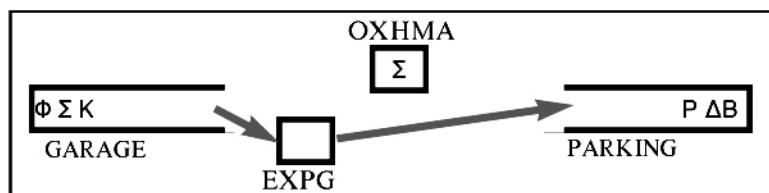
```
TO EXHGAGE
    PR [Δώσε τον κωδικό του οχήματος (χαρακτήρα)]
    MAKE "OXHMA READCHAR
    IFELSE MEMBER? :OXHMA :GARAGE
        [IF NOT EMPTY? :GARAGE
            [ PARE :OXHMA
              TYPWSE ]
          [PR [Δεν υπάρχει τέτοιο όχημα στο γκαράζ] ]
        ]
END
```

Ας δούμε πως η διαδικασία PARE διεκπεραιώνει το καθήκον της που είναι να βρεί το ζητούμενο όχημα και να το παραδώσει.

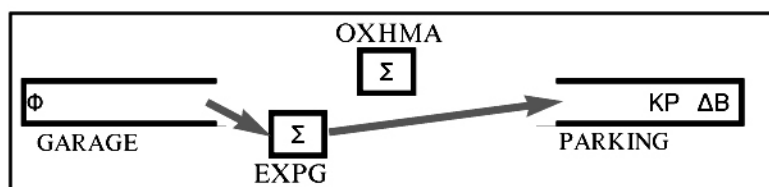
```
TO PARE :OXHMA
    BGALEG
    IFELSE :EXPG = :OXHMA
        [ PR [πάρε το όχημα ]
          ADEIASEP ]
        [ BALEP :EXPG
          PARE :OXHMA ]
END
```

το parking

Αυτό το πετυχαίνει (με αναδρομικό τρόπο) βγάζοντας ένα-ένα τα οχήματα από το γκαράζ και βάζοντας τα στο πάρκινγκ μέχρι να βρεθεί το ζητούμενο.



Όταν βρεί το ζητούμενο όχημα, το παραδίδει στον κάτοχο



και επαναφέρει στο γκαράζ τα οχήματα που έχουν μεταφερθεί προσωρινά στο παρκινγκ με τη βοήθεια της διαδικασίας ADEIASEP.

Η διαδικασία ADEIASEP (με αναδρομικό τρόπο) βγάζει ένα-ένα τα οχήματα από το πάρκινγκ και τα βάζει στο γκαράζ μέχρι να αδειάσει το πάρκινγκ.

ΤΟ ADEIASEP

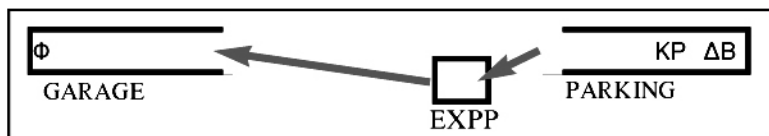
IF NOT EMPTY? :PARKING

[BGALEP

BALEG :EXPP

ADEIASEP]

END



το parking

Η διαδικασία EXYPHRETHSH επιτρέπει να επικοινωνήσει το πρόγραμμα με τον άνθρωπο που θα ζητήσει να παρκάρει ή να ξεπαρκάρει το όχημά του. Η διαδικασία EXYPHRETHSH ζητάει από τον άνθρωπό να την ενημερώσει για το αν θα γίνει εισαγωγή οχήματος, εξαγωγή ή τέλος του προγράμματος. Σε περίπτωση που δεν επιλέξει μιλια από τις τρεις περιπτώσεις η διαδικασία καλεί τον εαυτό της ζητώντας νέα επιλογή.

TO EXYPHRETHSH

```
PR [Να κάνω εισαγωγή, εξαγωγή ή τέλος]
MAKE "ENERGEIA READLIST
IFELSE :ENERGEIA = [ εισαγωγή ]
    [ EISHGAGE ]
    [IFELSE :ENERGEIA = [ εξαγωγή ]
        [ EXHGAGE ]
        [ IF :ENERGEIA = [ τέλος ]
            [ STOPALL ]
        ]
    ]
EXYPHRETHSH
END
```

Αφού φτιάξαμε όλες τις απαραίτητες διαδικασίες που θα διαχειριστούν το πρόβλημα του ιδιόμορφου γκαράζ ας φτιάξουμε και το τελικό πρόγραμμα PARKADOROS.

TO PARKADOROS

```
INITG
INITP
EXYPHRETHSH
END
```

Μια κερκόπορτα για την τεχνητή ευφυΐα

Στη logo αντικείμενα μιας λίστας μπορούν να είναι και εντολές της. Έτσι παρέχεται η δυνατότητα στον προγραμματιστή να βάλει σε μια λίστα κάποιες εντολές

π.χ. MAKE "ENTOLAS [REPEAT 4 [RT 90 FD 10]]

και στη συνέχεια με την εντολή RUN να ζητήσει να εκτελεστούν.

RUN :ENTOLAS

Ένα παράδειγμα δίνεται με το παρακάτω πρόγραμμα που με αναδρομικό τρόπο ανεβοκατεβάζει τη χελώνα μεταξύ του άνω και του κάτω ορίου της οθόνης εκμεταλλευόμενο την προαναφερθείσα δυνατότητα της logo.

TO ANWKATW

MAKE "QESH POS

MAKE "Y LAST :QESH

IFELSE AND :Y < 80 :Y > -80

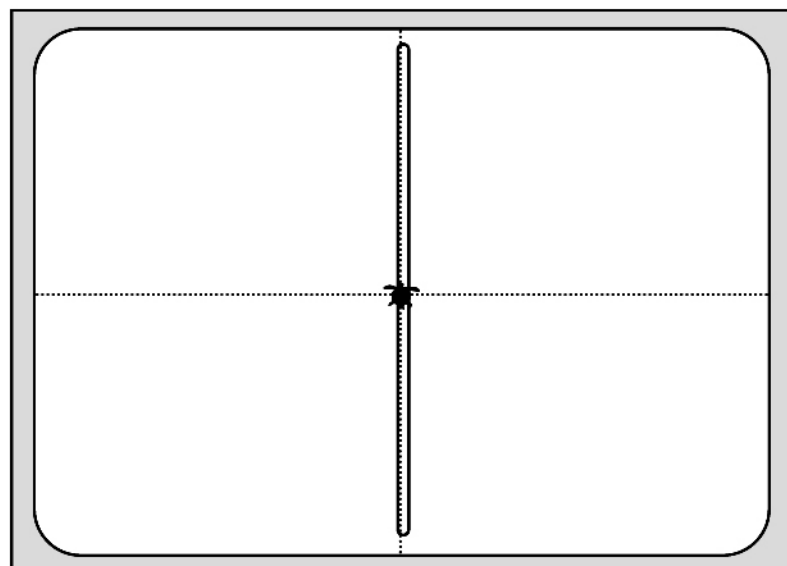
[MAKE "ENTOLH [FD 1]]

[MAKE "ENTOLH [RT 180 FD 1]]

RUN :ENTOLH

ANWKATW

END



Στη δυνατότητα της logo να αλλάζει τον κώδικα του προγράμματος ενώ αυτό εκτελείται, υπάρχει το σπέρμα της τεχνητής ευφυΐας. Το πρόγραμμα κατά τη διάρκεια της εκτέλεσής του μπορεί από τα αποτελέσματα που προκύπτουν να αποφασίζει να αλλάξει την δομή του, να αυτοβελτιωθεί και συνεπώς να γίνει ευφρέστερο από μόνο του.

Το πρόγραμμα “αλλάζει” τον εαυτό του

Ας χρησιμοποιήσουμε τη δυνατότητα της Logo να αλλάζει το πρόγραμμα ενώ αυτό εκτελείται για να μετασχηματίσουμε το πρόγραμμα που προσομοιώνει τον αγώνα του λαγού με τη χελώνα.

Ας φτιάξουμε μια διαδικασία, την INITKNS, η οποία να δημιουργεί μια λίστα KNS η οποία να περιέχει ως δεδομένα τις εντολές που περιγράφουν την κίνηση ενός αντικειμένου (του λαγού ή της χελώνας).

TO INITKNS

```
MAKE "KNS [TELL :ANTK  
MAKE "GA RANDOM :EGA  
RT :GA  
MAKE "PKA :PERA  
MAKE "BA RANDOM :EBA  
FD :BA ]
```

END

όπου η μεταβλητή ANTK μπορεί να περιέχει την τιμή 0 ή 1 που να αντιστοιχεί στη χελώνα ή το λαγό, η μεταβλητή EGA την μέγιστη γωνία που μπορεί να αποκλίνει από την πορεία του το αντικείμενο, η μεταβλητή GA την γωνία που θα αποκλίνει από την πορεία του το αντικείμενο, η μεταβλητή EBA το μέγιστο βήμα που μπορεί να κάνει το αντικείμενο, η μεταβλητή BA το βήμα που θα κάνει το αντικείμενο και τέλος η μεταβλητή PKA χρησιμεύει για να σφίβει εναλλάξ το αντικείμενο αριστερά και δεξιά.

Η επόμενη διαδικασία INITLX καλεί την διαδικασία και δίνει αρχικές τιμές στις μεταβλητές ANTK, EGA, PERA, EBA.

TO INITLX

```
INITKNS  
MAKE "ANTK 0  
MAKE "EGA 10  
MAKE "PERA 1  
MAKE "EBA 5
```

END

Η διαδικασία EKKIN είναι η ίδια που χρησιμοποιήθηκε στην προηγούμενη έκδοση του προγράμματος και σκοπό έχει να βάλει το αντικείμενο στην θέση εκκίνησης.

```
TO EKKIN :ANTK :X :Y :SXHM
  TELL :ANTK
  PU
  SETX :X
  SETY :Y
  PD
  LT 90
  SETSH :SXHM
  ST
END
```

Ας δούμε τώρα τη διαδικασία KINLX που είναι αυτή που με αναδρομικό τρόπο περιγράφει την κίνηση του λαγού και της χελώνας μέχρι κάποιο από τα δύο να τερματίσει.

```
TO KINLX
  IFELSE :PERA = 1
    [ALLAXE 7 "LT
      KINHSELX
      MAKE "PERA 0
    ]
    [ALLAXE 7 "RT
      KINHSELX
      MAKE "PERA 1
    ]
  TELL 0
  MAKE "XXEL XCOR
  TELL 1
  MAKE "XLAG XCOR
  MAKE "TELOS OR :XXEL < -150 :XLAG < -150
  IF NOT :TELOS [KINLX]
END
```

Αναλόγως με την τιμή της μεταβλητής PERA η διαδικασία ALLAXE (θα δούμε παρακάτω πως) αλλάζει στη λίστα KNS το 7^ο στοιχείο τη μια φορά σε LT και την άλλη φορά σε RT.

Η διαδικασία KINHSELX θέτει (με την βοήθεια της διαδικασίας ALLAXE) τις τιμές εκείνες στη λίστα KNS που αντιστοιχούν στα δεδομένα της χελώνας και μετά με την εντολή RUN :KNS πραγματοποιεί το “στιγμιαίο” βήμα της χελώνας. Στη συνέχεια κάνει την αντίστοιχη δουλειά και κινεί στιγμιαία και το λαγό.

TO KINHSELX

ALLAXE 2 0

ALLAXE 6 15

ALLAXE 15 5

RUN :KNS

ALLAXE 2 1

ALLAXE 6 30

ALLAXE 15 10

RUN :KNS

END

Ας δούμε τώρα πως η διαδικασία ALLAXE αλλάζει στο νιοστό στοιχείο μιας λίστας (το AAST) με την τιμή της μεταβλητής NEO.

TO ALLAXE :AAST :NEO

MAKE “PRS []

REPEAT :AAST [BGALEKNS BALEPRS]

MAKE “PRS BF :PRS

MAKE “KNS FPUT :NEO :KNS

MAKE “AAST :AAST - 1

REPEAT :AAST [BGALEPRS BALEKNS]

END

Με την εντολή MAKE “PRS []

φτιάχνει μια προσωρινή άδεια λίστα. Στη συνέχεια με την εντολή REPEAT βγάζει τα ν πρώτα στοιχεία της λίστας KNS (με τη διαδικασία BGALEKNS) και για να μην χαθούν τα φυλάει στη λίστα PRS (με τη διαδικασία BALEPRS). Μετά σβήνει το τελευταίο

στοιχείο που μπήκε στη λίστα PRS και βάζει στη λίστα KNS το στοιχείο που υπάρχει στη μεταβλητή NEO. Τέλος μεταφέρει ξανά όλα τα στοιχεία της λίστας PRS στη λίστα KNS.

Οι λίστες KNS και PRS δουλεύονται ως στοίβες και έτσι χρησιμοποιούνται οι διαδικασίες BALEKNS, BGALEKNS, BALEPRS και BGALEPRS που βάζουν και βγάζουν ένα στοιχείο αντίστοιχα.

TO BALEKNS

MAKE "KNS FPUT :STOIXEIO :KNS
END

TO BGALEKNS

IF NOT EMPTY? :KNS
[MAKE "STOIXEIO FIRST :KNS
MAKE "KNS BF :KNS]
END

TO BALEPRS

MAKE "PRS FPUT :STOIXEIO :PRS
END

TO BGALEPRS

IF NOT EMPTY? :PRS
[MAKE "STOIXEIO FIRST :PRS
MAKE "PRS BF :PRS]
END

Ας δούμε τώρα και το βασικό πρόγραμμα που θα φτιάξει τον αγώνα του λαγού με τη χελώνα.

TO AGWN

EKKIN 0 150 40 0
EKKIN 1 150 -40 22
INITLX
KINLX
END

Ανάπτυξη εργαλείων λογισμικού

Συχνά υπάρχει η ανάγκη να κάνουμε κάποιο χειρισμό αλλά η γλώσσα προγραμματισμού στην οποία δουλεύουμε δεν διαθέτει την αντίστοιχη εντολή. Τότε μπορούμε να φτιάξουμε μια διαδικασία που να χρησιμοποιεί τις υπάρχουσες εντολές της γλώσσας και να κάνει τη δουλειά που θέλουμε. Για παράδειγμα θέλουμε να φτιάξουμε την “εντολή-φίλτρο” `ΠΑΡΕ_ΑΡΙΘΜΟ` που να παίρνει από το πληκτρολόγιο έναν πολυψήφιο ακέραιο αριθμό αγνοώντας τυχόν άλλους χαρακτήρες που μπορεί να πληκτρολογηθούν κατά λάθος.

Η διαδικασία `ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ` χρησιμοποιεί την εντολή `READCHAR` και παίρνει από το πληκτρολόγιο ένα χαρακτήρα. Αν είναι χαρακτήρας που αντιστοιχεί σε αριθμητικό ψηφίο ή ο χαρακτήρας που αντιστοιχεί στο `RETURN` τον φυλάει στη μεταβλητή `MONOCHAR` αλλιώς τον αγνοεί και με αναδρομικό τρόπο παίρνει από το πληκτρολόγιο τον επόμενο χαρακτήρα.

ΤΟ `ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ`

MAKE “`ΑΡΙΘΜΟΙ [0 1 2 3 4 5 6 7 8 9]`”

MAKE “`ENTER CHAR 13`”

MAKE “`MONOCHAR READCHAR`”

MAKE “`TELOS :MONOCHAR = :ENTER`”

IF NOT MEMBER? :MONOCHAR :ΑΡΙΘΜΟΙ

[IF NOT :TELOS [`ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ`]]

END

Ο χαρακτήρα που υπάρχει στη μεταβλητή `MONOCHAR` μπορεί να μετασχηματιστεί στον αντίστοιχο του μονοψήφιο αριθμό `PSHFIO` με τις εντολές:

MAKE “`MONONUM ASCII :MONOCHAR`”

MAKE “`PSHFIO :MONONUM - 48`”

όπου 48 είναι ο αριθμός του ASCII κώδικα που αντιστοιχεί στο χαρακτήρα του αριθμού μηδέν.

Η διαδικασία `ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ` με αναδρομικό τρόπο παίρνει πολλούς χαρακτήρες και αφού τους φιλτράρει τους μετατρέπει σε πολυψήφιο ακέραιο που τον βάζει στη μεταβλητή `NUM`.

```

ΤΟ ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
  ΠΑΡΕ_ΧΑΡΑΚΤΗΡΑ_ΑΡΙΘΜΟΥ
  IFELSE NOT :ΤΕΛΟΣ
    [ MAKE "ΜΟΝΟΝΟΜ ASCII :ΜΟΝΟΧΑΡ
      MAKE "ΨΗΦΙΟ :ΜΟΝΟΝΟΜ - 48
      MAKE "ΝΟΜ :ΝΟΜ * 10
      MAKE "ΝΟΜ :ΝΟΜ + :ΨΗΦΙΟ
      ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
    ]
  [STOP]
END

```

Τέλος η διαδικασία ΠΑΡΕ_ΑΡΙΘΜΟ αφού μηδενίσει αρχικά τον αριθμό ΝΟΜ καλεί την διαδικασία ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ και πετυχαίνει το επιθυμητό αποτέλεσμα.

```

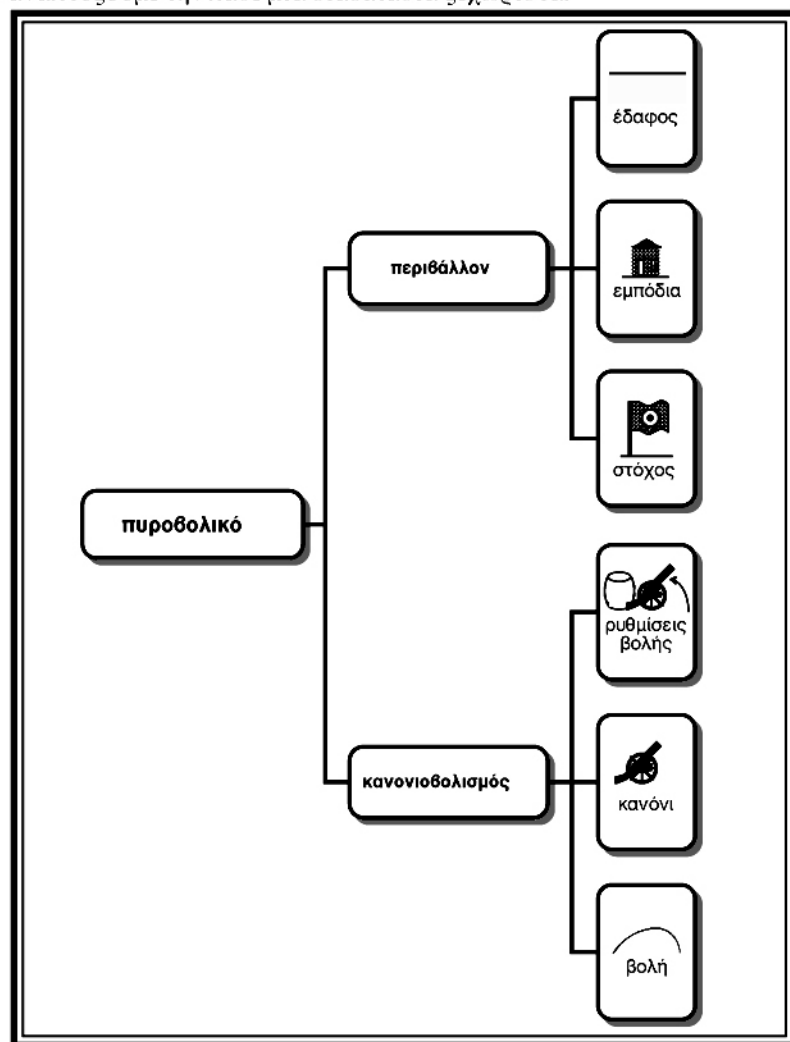
ΤΟ ΠΑΡΕ_ΑΡΙΘΜΟ
  MAKE "ΝΟΜ 0
  ΣΧΗΜΑΤΙΣΕ_ΑΚΕΡΑΙΟ
END

```


17 Μικρόκοσμοι

Ο οπτικός πίνακας περιεχομένων

Το «CannonFodder» είναι ένα παιχνίδι για υπολογιστή που προσομοιάνει μια μάχη πυροβολικού. Θα προσπαθήσουμε να το υλοποιήσουμε με logo. Στο πρώτο επίπεδο του οπτικού πίνακα περιεχομένου εργασιών που πρέπει να γίνουν είναι να ζωγραφιστεί το περιβάλλον και να ρυθμιστεί υπολογιστικά και σχεδιαστεί η διαδικασία του κανονιοβολισμού. Στις επόμενες σελίδες θα αναπτύξουμε την κάθε μια διαδικασία ξεχωριστά.



Το έδαφος

Ξεκινώντας να σχεδιάσουμε το περιβάλλον, ας αρχίσουμε με το έδαφος. Για να δημιουργήσουμε την αίσθηση του συμπαγούς που πρέπει να έχει το έδαφος, σχεδιάζουμε πολλές (20) παράλληλες γραμμές από την επιφάνεια το εδάφους μέχρι κάποιο βάθος. Για να μη σχηματισθεί μια μαύρη επιφάνεια, μεταξύ δύο διαδοχικών γραμμών αφήνονται δύο κενές.

Η γραμμή που θα αντιστοιχεί στην επιφάνεια του εδάφους θα ξεκινά από το σημείο (-190, -100) και θα εκτείνεται μέχρι το (240, -100).

ΤΟ ΕΔΑΦΟΣ

```
MAKE "X MINUS 190
```

```
MAKE "Y MINUS 100
```

```
REPEAT 20
```

```
  [ ΜΗ_ΓΡΑΦΕΙΣ
```

```
    SETX :X
```

```
    SETY :Y
```

```
    ΓΡΑΨΕ
```

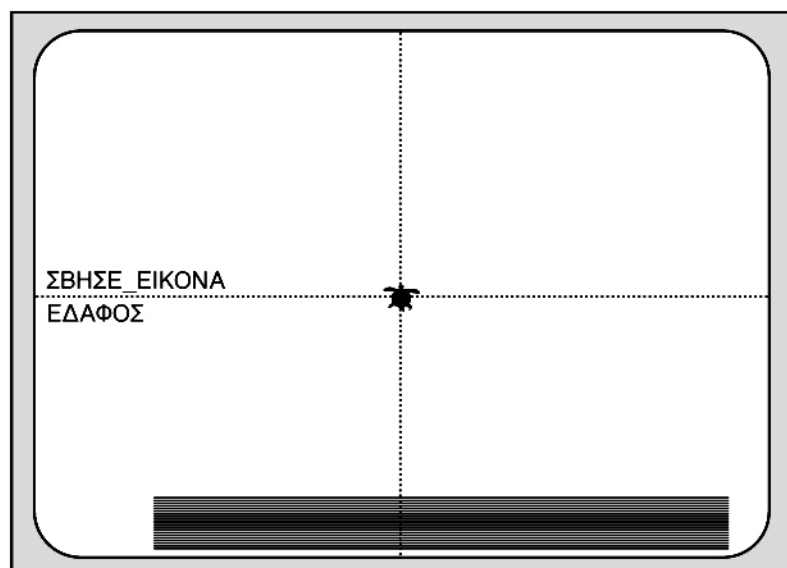
```
    ΓΡΑΜΜΗ 90 430
```

```
    MAKE "X MINUS 190
```

```
    MAKE "Y :Y - 2
```

```
    ΑΦΕΤΗΡΙΑ ]
```

```
END
```



Τα εμπόδια

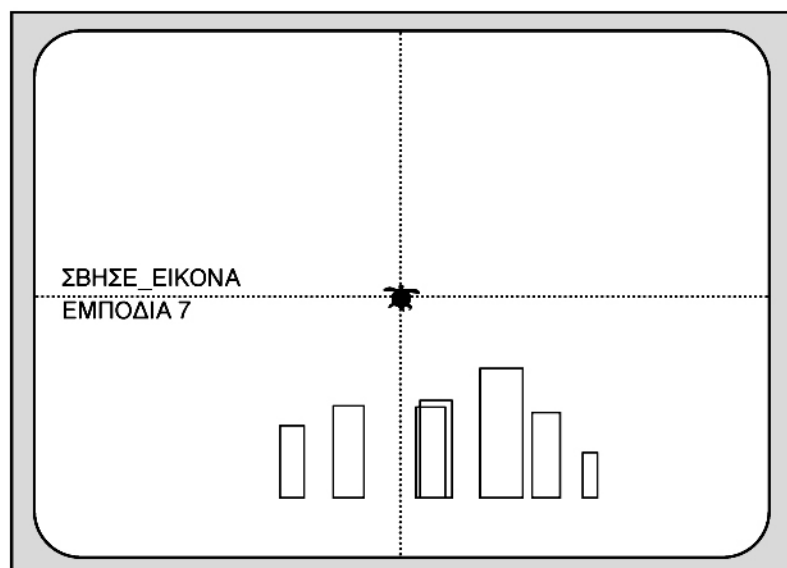
Στο παιχνίδι μεταξύ του κανονιού και του στόχου παρεμβάλλονται μερικά κτίρια που εμποδίζουν το κανόνη να βάλει σε ευθεία βολή κατά του στόχου.

Στο πρόγραμμά μας, θα προσομοιάσουμε τα κτίρια με ορθογώνια παραλληλόγραμμα. Η θέση των κτιρίων στο έδαφος θα βρίσκεται μεταξύ -125 και 124 και θα προκύπτει τυχαία. Ακόμα το ύψος του κτιρίου θα κυμαίνεται από 0 μέχρι 99 βήματα χελώνας.

```
ΤΟ ΕΜΠΟΔΙΟ
ΜΗ ΓΡΑΦΕΙΣ
MAKE "X (RANDOM 250) - 125
MAKE "Y MINUS 100
SETX :X
SETY :Y
ΓΡΑΨΕ
MAKE "YPSOS RANDOM 100
ΟΡΘΟΓΩΝΙΟ :YPSOS 3
ΑΦΕΤΗΡΙΑ
END
```

Τα πολλά εμπόδια που χρειάζονται, θα δημιουργούνται από τη διαδικασία ΕΜΠΟΔΙΑ και το πλήθος τους θα καθορίζεται από την τιμή της μεταβλητής POSA.

```
ΤΟ ΕΜΠΟΔΙΑ :POSA
REPEAT :POSA
  [ΕΜΠΟΔΙΟ]
END
```



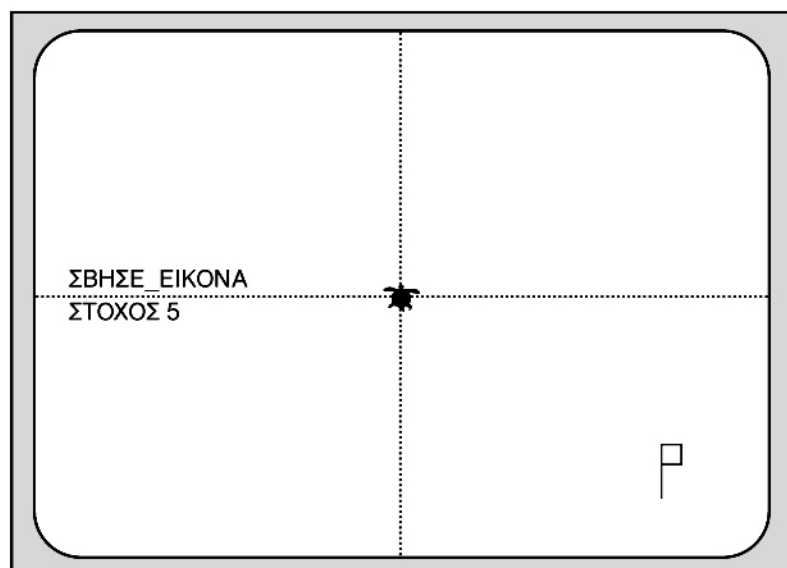
Ο στόχος

Ο στόχος εναντίον του οποίου θα βάλλει το κανόνι, είναι μια σημαία. Η θέση της σημαίας θα είναι επί του εδάφους, στο δεξιό άκρο της οθόνης στο σημείο (200, -100).

Η σημαία αποτελείται από τον ιστό και το πανί της. Ο ιστός θα σχεδιαστεί ως μια ευθεία γραμμή και το πανί ως ένα τετράγωνο.

Τέλος για να μη χάσει η χελώνα τον μπούσουλα της (προσανατολισμό), ώστε να μπορεί να χρησιμοποιηθεί από επόμενη διαδικασία, την επαναφέρουμε στο σημείο αφετηρίας της.

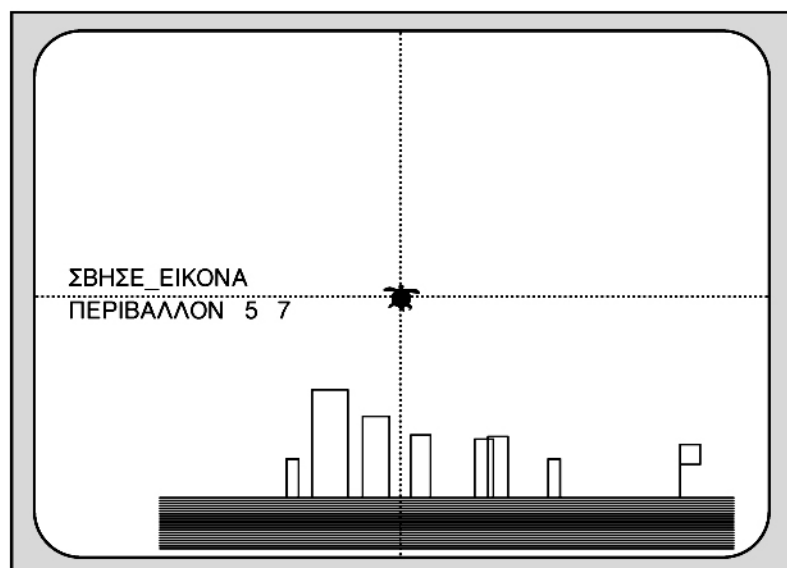
```
ΤΟ ΣΤΟΧΟΣ :MEGEQOS
ΜΗ_ΓΡΑΦΕΙΣ
MAKE "X 200
MAKE "Y MINUS 100
SETX :X
SETY :Y
ΓΡΑΨΕ
ΕΥΘΕΙΑ :MEGEQOS * 5
ΠΟΛΥΓΩΝΟ :MEGEQOS * 3 4
ΑΦΕΤΗΡΙΑ
END
```



Το περιβάλλον

Τα τμήματα του περιβάλλοντος στο οποίο θα πραγματοποιηθεί ο κανονιοβολισμός είναι πλέον έτοιμα. Απομένει να συνδέσουμε τις διαδικασίες που σχεδιάζουν το έδαφος, τα εμπόδια και το στόχο. Αυτό γίνεται με τη διαδικασία ΠΕΡΙΒΑΛΛΟΝ.

```
ΤΟ ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ :ΡΟΔΑ  
ΕΔΑΦΟΣ  
ΕΜΠΟΔΙΑ :ΡΟΔΑ  
ΣΤΟΧΟΣ :ΜΕΓΕΘΟΣ  
ΕΝΔ
```



Οι ρυθμίσεις της βολής

Στο παιχνίδι «Cannon Fodder» δύο παράμετροι καθορίζουν την τροχιά του βλήματος. Η γωνία κλίσης της κάνης του πυροβόλου (συμβολίζεται ως φ ή F στο πρόγραμμα) και η ποσότητα του μπαρουτιού που υπάρχει στο κανόνι. Αυτές οι δύο παράμετροι θα ζητούνται από το το χρήστη κατά την εκτέλεσή του προγράμματος.

Για λόγους απλούστευσης του προβλήματος, ταυτίζουμε την ποσότητα του μπαρουτιού με την αρχική ταχύτητα του βλήματος, που συμβολίζεται ως u_0 ή U_0 στο πρόγραμμα.

ΤΟ ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ

PR "Δώσε κλίση κανονιού

ΠΑΡΕ_ΑΡΙΘΜΟ

MAKE "F :NUM

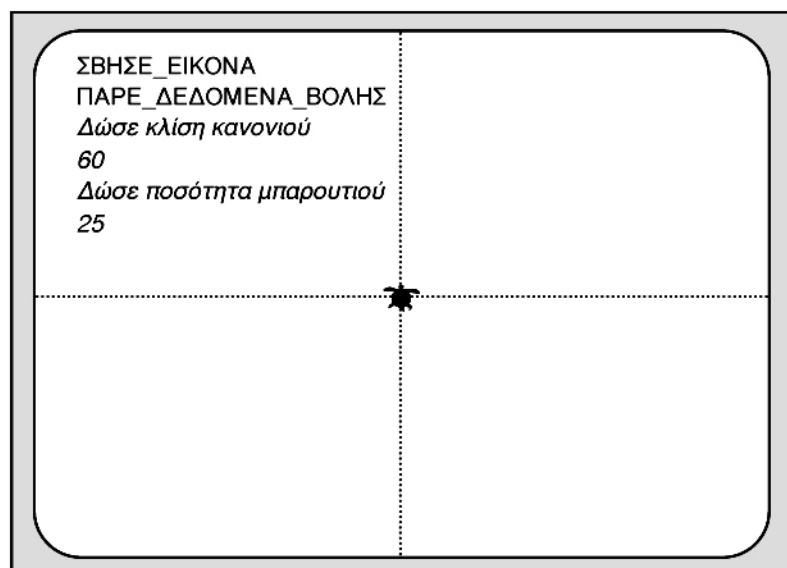
PR "Δώσε ποσότητα μπαρουτιού

ΠΑΡΕ_ΑΡΙΘΜΟ

MAKE "U0 :NUM

JEND

Μπορούμε να θέσουμε και άλλες παραμέτρους από τις οποίες να εξαρτάται η βολή. Τη φορά και την ένταση του πνέοντος ανέμου, από το αν η βολή εκτελείται στον Ισημερινό ή στους πόλους, στην επιφάνεια της θάλασσας ή σε βουνό, στη Σελήνη ή στον Άρη (μεταβάλλεται η τιμή της επιτάχυνσης της βαρύτητας που συμβολίζεται με g ή G στο πρόγραμμα).

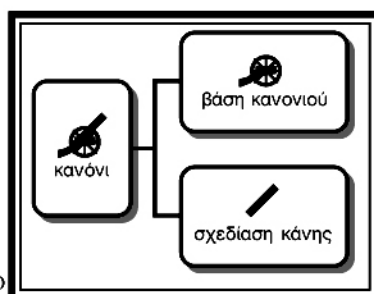


«Όλοι
μαθαίνουμε
κατασκευάζοντα
ς, εξερευνώντας
και κτίζοντας
θεωρίες, αλλά το
μεγαλύτερο
μέρος του
κτισίματος που
μας
ταλαιπώρησε
κατέληξε σε
θεωρίες τις
οποίες θα
χρειαστεί
αργότερα να
παρατήσουμε...
Τα παιδιά δεν
ακολουθούν ένα
μονοπάτι
μάθησης που
οδηγείται από
μια "αληθινή
θέση" σε άλλη,
πιο
προχωρημένη
"αληθινή θέση".
Τα φυσικά τους
μονοπάτια
μάθησης
περιέχουν
"ψευδείς
θεωρίες" που
διδάσκουν
σχετικά με το
κτίσιμο θεωριών
όσα και οι
αληθινές».
Seymour Papert

Η βάση και η κάνη του κανονιού

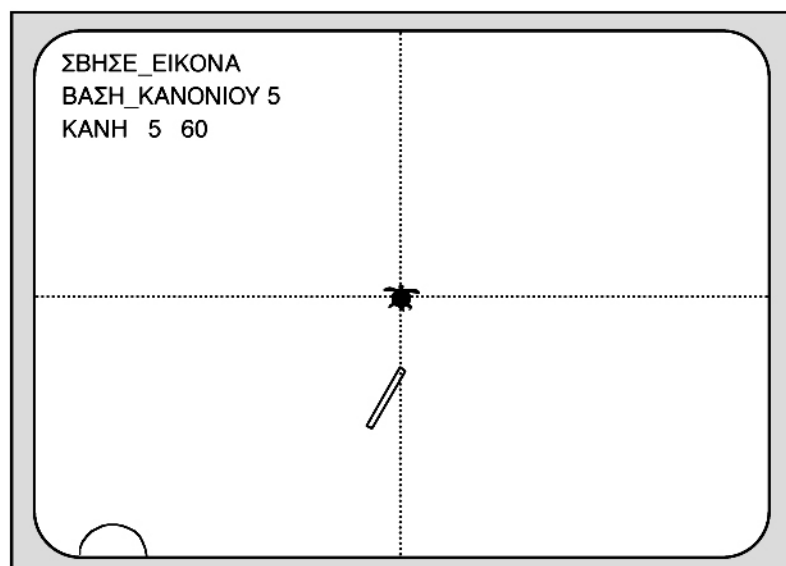
Το κανόνι αποτελείται από δύο μέρη, τη βάση και την κάνη. Είναι απαραίτητο να σχεδιαστούν αυτά τα μέρη ξεχωριστά, γιατί ενώ η βάση παραμένει ακίνητη, η κάνη πρέπει να στρέφεται ανάλογα με την κλίση. Η βάση θα σχεδιαστεί ως ένα ημικύκλιο,

```
ΤΟ ΒΑΣΗ ΚΑΝΟΝΙΟΥ :MG  
ΜΗ_ΓΡΑΦΕΙΣ  
MAKE "X MINUS 240  
MAKE "Y MINUS 140  
SETX :X  
SETY :Y  
ΓΡΑΨΕ  
ΔΕΞΙΑ 90  
ΤΜΗΜΑ_ΠΟΛΥΓΩΝΟΥ :MG 32 1 / 2  
ΑΦΕΤΗΡΙΑ  
END
```



ενώ η κάνη ως ένα ορθογώνιο παραλληλόγραμμο. Η στροφή της κάνης θα γίνεται ως προς το άνω δεξιό άκρο που θα παραμένει ακίνητο και από το οποίο θα αρχίζει η κίνηση του βλήματος.

```
ΤΟ ΚΑΝΗ :MEGEQOS :KLISH  
ΑΡΙΣΤΕΡΑ :KLISH + 180  
MAKE "LOGOS 1 / 10  
ΟΡΘΟΓΩΝΙΟ :MEGEQOS :LOGOS  
ΔΕΞΙΑ :KLISH + 180  
END
```



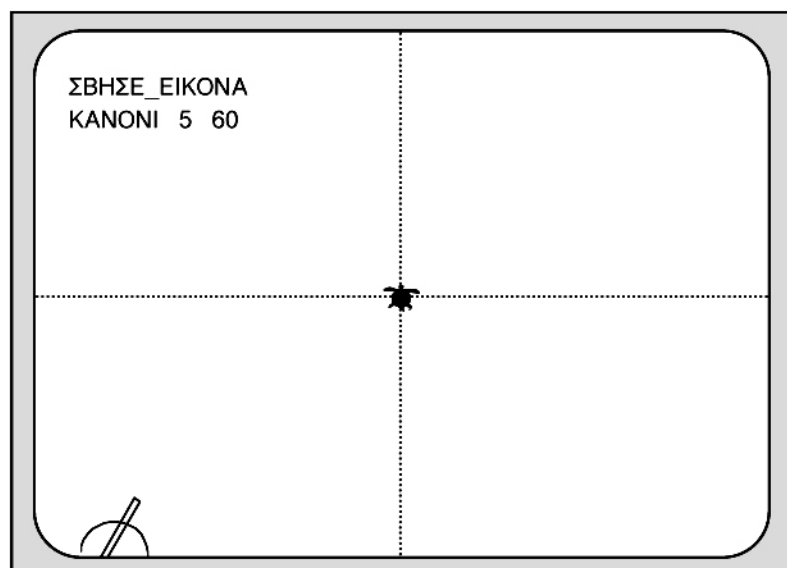
Το κανόνι

Η κνήμη θα πρέπει να τοποθετηθεί στο σωστό σημείο και όχι στο κέντρο της οθόνης. Αυτό γίνεται με τη διαδικασία ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ που μεταφέρει τη χελώνα στο σημείο (-200, -100) που είναι η θέση το πυροβόλου.

```
ΤΟ ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΜΗ_ΓΡΑΦΕΙΣ
MAKE "X MINUS 200
MAKE "Y MINUS 100
SETX :X
SETY :Y
ΓΡΑΨΕ
ΕΝΔ
```

Η διαδικασία που σχεδιάζει το κανόνι, δεν έχει παρά να συνδέσει τις διαδικασίες ΒΑΣΗ_ΚΑΝΟΝΙΟΥ, ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ, ΚΑΝΗ και ΑΦΕΤΗΡΙΑ.

```
ΤΟ ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :ΚΛΙΣΗ
ΒΑΣΗ :ΜΕΓΕΘΟΣ
ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΚΑΝΗ :ΜΕΓΕΘΟΣ :ΚΛΙΣΗ
ΑΦΕΤΗΡΙΑ
ΕΝΔ
```



«Όλοι
μαθαίνουμε
κατασκευάζοντας,
εξερευνώντας και
κτίζοντας
θεωρίες, αλλά το
μεγαλύτερο μέρος
του κτισίματος
που μας
ταλαιπώρησε
κατέληξε σε
θεωρίες τις
οποίες θα
χρειαστεί
αργότερα να
παρατήσουμε...
Τα παιδιά δεν
ακολουθούν ένα
μονοπάτι
μάθησης που
οδηγείται από μια
"αληθινή θέση" σε
άλλη, πιο
προχωρημένη
"αληθινή θέση".
Τα φυσικά τους
μονοπάτια
μάθησης
περιέχουν
"ψεύτικες
θεωρίες" που
διδάσκουν
σχετικά με το
κτίσιμο θεωριών
όσα και οι
αληθινές».
Seymour Papert

Υπολογισμοί της τροχιάς του βλήματος

Η βολή προϋποθέτει κάποιους υπολογισμούς, τη σχεδίαση του βλήματος και τη διαδικασία του "animation" που θα σχεδιάσει την τροχιά του βλήματος.

Ο καθορισμός της τροχιάς που θα ακολουθήσει το βλήμα, γίνεται με τη βοήθεια της φυσικής. Δε θα μας απασχολήσει ο τρόπος που θα καταλήξουμε στους απαραίτητους τύπους. Αλλά θα δανειστούμε από την φυσική τους εξής τύπους που σχετίζονται με την πλάγια βολή:

$x = u_0 \sin \varphi t$ x , είναι η οριζόντια απόσταση του βλήματος από την αρχή κάποια χρονική στιγμή t .

$y = x \tan \varphi - \frac{g t^2}{2 \cos^2 \varphi}$ y , είναι η κατακόρυφη απόσταση του βλήματος από την αρχή την χρονική στιγμή t .

$t_{\text{ολ}} = \frac{2 u_0 \sin \varphi}{g}$ $t_{\text{ολ}}$, ο χρόνος που διαρκεί η βολή.

Κάποια τμήματα αυτών των υπολογισμών είναι ανεξάρτητα του χρόνου και έτσι μας συμφέρει (από άποψη ταχύτητας εκτέλεσης του προγράμματος) να υπολογιστούν στην αρχή. Αυτά είναι τα εξής:

$x = x_{\text{χωρίς}} t$

$\text{παρονομαστής} = 2 u_0 \sin \varphi$

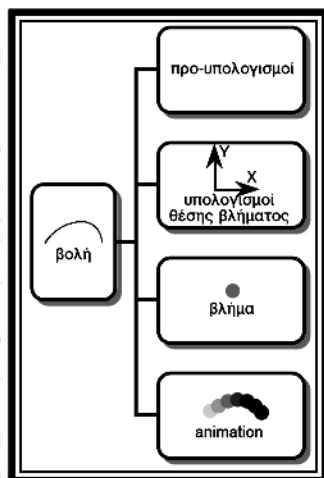
καθώς και ο υπολογισμός της συνάρτησης της εφα.

Επίσης θα πρέπει να δοθεί η σταθερή τιμή στην επιτάχυνση της βαρύτητας g , καθώς και οι συντεταγμένες $X_0 = -195, Y_0 = -100$ από όπου θα ξεκινάει το βλήμα.

Έτσι έχουμε τη διαδικασία ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ.

```

ΤΟ ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ :F :U0
MAKE "G 9.81
MAKE "X0 MINUS 195
MAKE "Y0 MINUS 100
MAKE "U0 :U0 * 4
MAKE "TOL (2 * :U0 / :G) * SIN(:F)
MAKE "T 0
MAKE "X_XWRIS_T (:U0 * COS(:F))
MAKE "EFF (SIN(:F) / COS(:F))
MAKE "PAR (2 * :U0 * COS(:F))
END
    
```



«... μαθητές οι οποίοι προσπαθούν να αναπτύξουν νευτώνεια σχέση σχετικά με την κίνηση αντιμετωπίζουν προβλήματα... Πρέπει πρώτα να μάθουν πως να δουλεύουν με τις εξισώσεις πριν τις χρησιμοποιήσουν για να σχεδιάσουν ένα νευτώνειο κόσμο... Σε ένα προσομοιωμένο κόσμο, όπου οι μαθητές θα έχουν άμεση πρόσβαση στη νευτώνεια κίνηση δε χρειάζεται κανείς να μάθει καλά τις εξισώσεις. Ακριβώς το αντίθετο: αντί να κάνουμε τους μαθητές να περιμένουν τις εξισώσεις, μπορούμε να προκαλέσουμε και να διευκολύνουμε την απόκτηση εξισωτικών δεξιοτήτων παρέχοντας ένα διασθετικά πολύ καλά κατανοητό πλαίσιο για τη χρήση τους».

Seymour Papert

Η θέση του βλήματος κάποια στιγμή

Με βάση τους αρχικούς υπολογισμούς, οι εξισώσεις που καθορίζουν τις συντεταγμένες X, Y κάποια χρονική στιγμή είναι:

$$\begin{aligned} x &= x_0 + v_{x0} \cdot t \\ y &= y_0 + v_{y0} \cdot t - \frac{g \cdot t^2}{2} \end{aligned}$$

Έτσι, υπολογίζεται η θέση του βλήματος με τη διαδικασία:

```
TO ΘΕΣΗ_ΒΛΗΜΑΤΟΣ
MAKE "X :X_XWRIS_T*:T
MAKE "Y (:X*EFF) - (:G*:T*:X/:PAR)
END
```

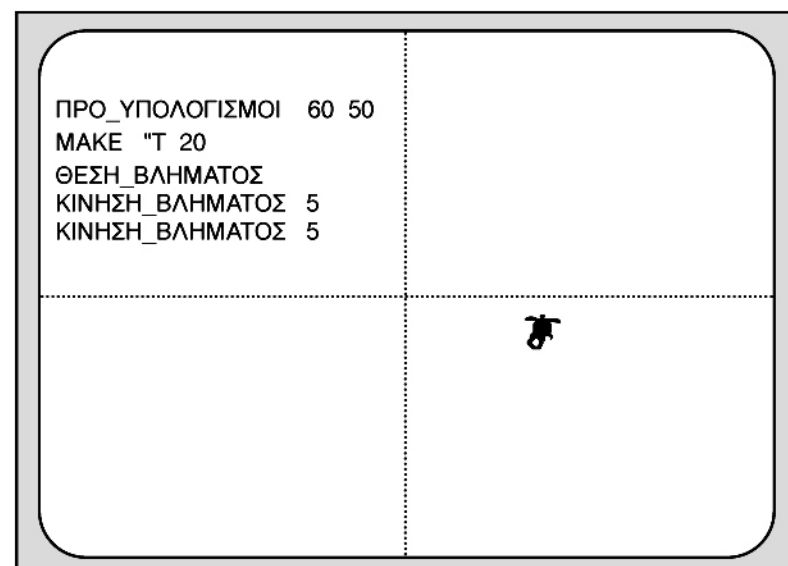
Το βλήμα και η κίνησή του, περιγράφονται από τις ακόλουθες διαδικασίες:

```
TO ΒΛΗΜΑ :MEGEQOS
ΠΟΛΥΓΩΝΟ :MEGEQOS/2 6
END

TO ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ :MEGEQOS
ΜΗ_ΓΡΑΦΕΙΣ
SETPOS [(:X0+:X/10)(:Y0+:Y/10)]
ΓΡΑΨΕ
ΒΛΗΜΑ :MEGEQOS
ΣΒΗΣΕ
ΒΛΗΜΑ :MEGEQOS
END
```

Για λόγους που έχουν σχέση με την ταχύτητα εκτέλεσης του προγράμματος το σχήμα του βλήματος δεν έγινε κυκλικό (36-πλευρο) αλλά κανονικό εξάγωνο.

Για να ταιριάζει με το μέγεθος της οθόνης, το βλήμα X και Y διαίρεται δια 10.

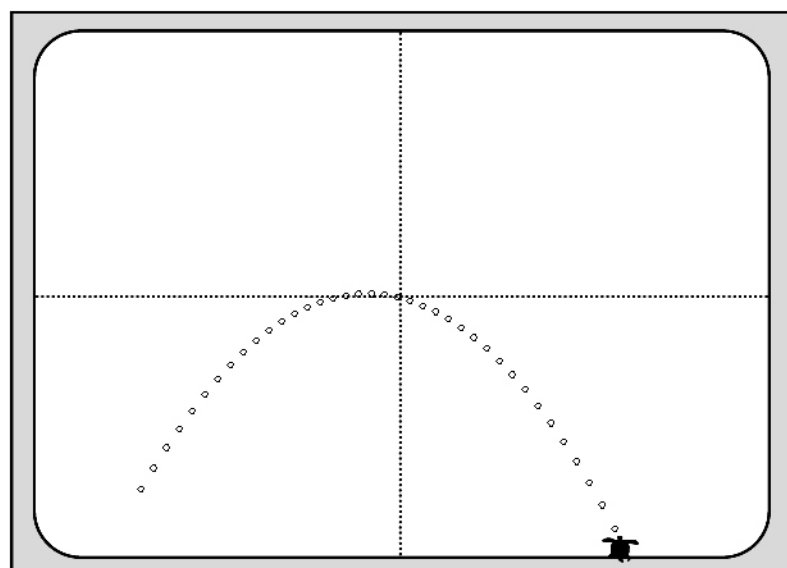


Η τροχιά του βλήματος

Η χάραξη της τροχιάς του βλήματος είναι πλέον απλή υπόθεση. Η διαδικασία που θα την υλοποιεί αφού κάνει τους απαραίτητους αρχικούς υπολογισμούς, θα επαναλάβει τη διαδικασία ANIMATION_BLHMATOS τόσες φορές όση είναι η τιμή του χρόνου $t_{ολ}$ που διαρκεί η βολή.

Σε κάθε επανάληψη θα πρέπει να αυξάνει η τιμή της μεταβλητής T που εκφράζει το χρόνο που ρέει και με βάση την οποία γίνονται κάθε φορά οι υπολογισμοί που καθορίζουν τις συντεταγμένες X,Y του βλήματος.

```
TO ΤΡΟΧΙΑ :F :U0 :MEGEQOS
  ΠΡΟ_ΥΠΟΛΟΓΙΣΜΟΙ :F :U0
  REPEAT :TOL
    [ ΘΕΣΗ_ΒΛΗΜΑΤΟΣ
      ΚΙΝΗΣΗ_ΒΛΗΜΑΤΟΣ :MEGEQOS
      MAKE "T :T + 1
      ΓΡΑΨΕ ]
  END
```



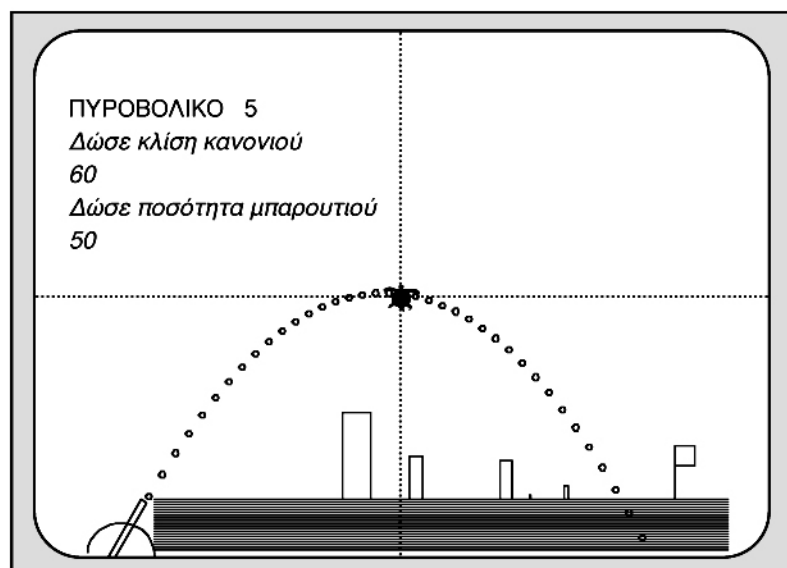
«Οι μαθητές δεν μπορούν να κάνουν κτήμα τους κάτι που δεν το γνωρίζουν. Γι' αυτό είναι προαπαιτούμενο να βρούμε τρόπους να διευκολύνουμε την προσωπική ιδιοποίηση όχι μόνο της νευτώνειας κίνησης και των νόμων που την περιγράφουν, αλλά επίσης της γενικής έννοιας των νόμων. Αυτό το κάνουν σχεδιάζοντας μια σειρά από μικρόκοσμους. Ο σχεδιασμός του μικρόκοσμου τον κάνει "φυτώριο" ενός συγκεκριμένου είδους δυναμικών ιδεών και διανοητικών δομών».

Seymour Papert

Το κανόνι βάλλει κατά του στόχου

Ας συνοψίσουμε σε μια διαδικασία, την ΠΥΡΟΒΟΛΙΚΟ, τη σχεδίαση του περιβάλλοντος, την εισαγωγή των δεδομένων της ρύθμισης της βολής, τη σωστή τοποθέτηση του κανονιού και τέλος τη χάραξη της τροχιάς του βλήματος.

```
ΤΟ ΠΥΡΟΒΟΛΙΚΟ :MEGEQOS
ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΡΥΨΟΥ
ΠΕΡΙΒΑΛΛΟΝ :MEGEQOS 5
ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
ΚΑΝΟΝΙ :MEGEQOS :F
ΤΡΟΧΙΑ :F :U0 :MEGEQOS
ΒΓΕΣ
ΕΝΔ
```



«Σ' ένα καλό μάθημα τέχνης, το παιδί αποκτά **τεχνική γνώση ως μέσο** για να φτάσει στο δημιουργικό και προσωπικά καθορισμένο τέλος. Θα υπάρξει προϊόν. Ο δάσκαλος όπως και το παιδί μπορεί να ενθουσιαστεί γνήσια απ' αυτό. Στο περιβάλλον της logo αυτό συμβαίνει συχνά. Ο γνήσιος ενθουσιασμός του δασκάλου για το προϊόν μεταδίδεται στα παιδιά τα οποία γνωρίζουν ότι κάνουν κάτι σημαντικό. Διαφορετικά από το μάθημα αριθμητικής, όπου γνωρίζουν ότι οι αθροίσεις που κάνουν είναι μόνον ασκήσεις, εδώ αντιλαμβάνονται ισοβαρά τη δουλειά τους».

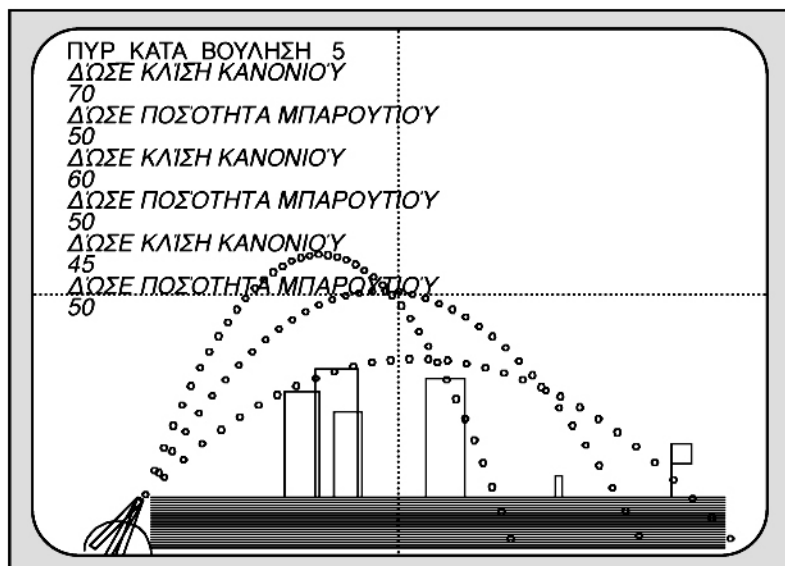
Seymour Papert

Δικαίωμα για επαναληπτικές βολές

Μπορούμε να επαναλαμβάνουμε τη βολή όσες φορές θέλουμε αλλά κάθε φορά θα αλλάζει και το τοπίο με τα εμπόδια. Για να αποφύγουμε αυτή τη συνεχή αλλαγή, θα βάλουμε μέσα σε μια εντολή repeat το τμήμα της διαδικασίας ΠΥΡΟΒΟΛΙΚΟ που υπάρχει μετά τη γραμμή που σχεδιάζει το περιβάλλον.

Καθορίζουμε σε 3 το δικαίωμα των βολών που δικαιούται κάποιος για να πετύχει το στόχο. Έτσι θα έχουμε την διαδικασία ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ.

```
ΤΟ ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ :ΜΕΓΕΘΟΣ  
ΣΒΗΣΕ_ΕΙΚΟΝΑ  
ΚΡΥΨΟΥ  
ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ 5  
REPEAT 3  
  [ ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ  
    ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :F  
    ΤΡΟΧΙΑ :F :U0 :ΜΕΓΕΘΟΣ  
    MAKE "Q READCHAR ]  
ΒΓΕΣ  
END
```



Παρατηρούμε ότι σε κάθε βολή, η κώνη του κανονιού στρέφεται για να πάρει τη νέα γωνία. Στην οθόνη παραμένουν τα σχέδια και από τις τρεις διαδοχικές κλίσεις.

Κάλυψη και απόκρυψη του κανονιού

Διαπιστώνουμε την ανάγκη να σβήνει κανή του κανονιού μετά από κάθε βολή. Έτσι φτιάχνεται η διαδικασία

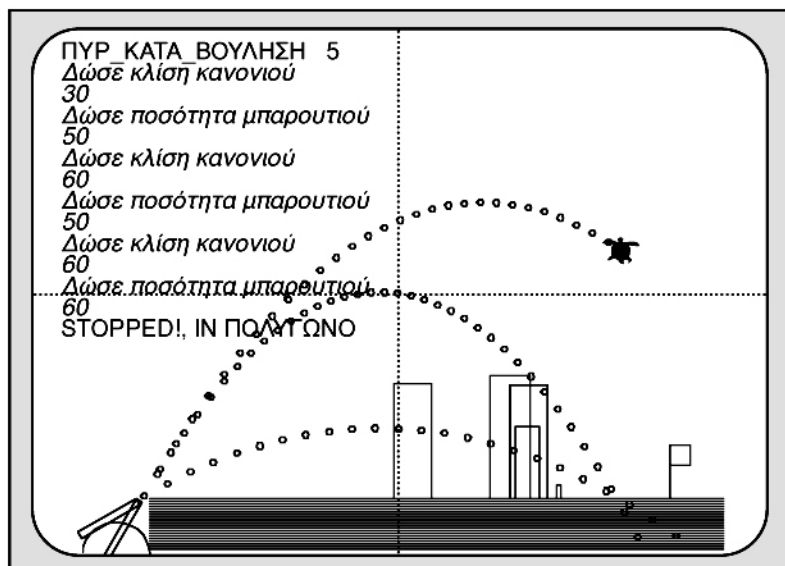
```

ΤΟ ΣΒΗΣΕ_ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :F
ΠΗΓΑΙΝΕ_ΣΤΟ_ΚΑΝΟΝΙ
ΑΡΙΣΤΕΡΑ :F + 180
ΜΑΚΕ "LOGOS 1 / 10
ΣΒΗΣΕ
ΟΡΘΟΓΩΝΙΟ :ΜΕΓΕΘΟΣ :LOGOS
ΓΡΑΨΕ
ΔΕΞΙΑ :F + 180
ΑΦΕΤΗΡΙΑ
ΕΝΔ
    
```

Η διαδικασία ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ τροποποιείται:

```

ΤΟ ΠΥΡ_ΚΑΤΑ_ΒΟΥΛΗΣΗ :ΜΕΓΕΘΟΣ
ΣΒΗΣΕ_ΕΙΚΟΝΑ
ΚΡΥΨΟΥ
ΠΕΡΙΒΑΛΛΟΝ :ΜΕΓΕΘΟΣ 5
ΡΕΠΕΑΤ 3
    [ ΠΑΡΕ_ΔΕΔΟΜΕΝΑ_ΒΟΛΗΣ
      ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :F
      ΤΡΟΧΙΑ :F :U0 :ΜΕΓΕΘΟΣ
      ΜΑΚΕ "Q READCHAR
      ΣΒΗΣΕ_ΚΑΝΟΝΙ :ΜΕΓΕΘΟΣ :F ]
ΒΓΕΣ
ΕΝΔ
    
```



«Ο Piaget έδειξε ότι τα παιδιά μαθαίνουν θεμελιώδεις μαθηματικές ιδέες, κτιζοντας πρώτα τα δικά τους, πολύ διαφορετικά μαθηματικά. Και τα παιδιά μαθαίνουν μια γλώσσα μαθαίνοντας πρώτα τις δικές τους ("παιδιάστικες") διαλέκτους. Έτσι, όταν σκεπτόμαστε τους μικρόκοσμους ως εκκολαπτήρια δυναμικών ιδεών, προσπαθούμε να σχεδιάσουμε πάνω σ' αυτή την αποτελεσματική στρατηγική: επιτρέπουμε στους μαθητές να μάθουν την "επίσημη" φυσική, δίνοντας τους την ελευθερία να εφεύρουν πολλά που μπορούν να λειτουργήσουν σε ισάριθμους κόσμους που θα επινοήσουν». Seymour Papert

Πειραματιζόμενοι με διάφορες γωνίες και κρατώντας σταθερό το μπαρούτι μπορούμε να βγάλουμε χρήσιμα συμπεράσματα για τη σχέση της γωνίας με το βεληνεκές.