

# Κεφάλαιο 12: Επαναλήψεις II



Σε αυτό το κεφάλαιο:

- 12.1 Εντολή «για πάντα εάν...»
- 12.2 Εντολή «επανέλαβε ώσπου...»
- 12.3 Εντολή «περίμενε ώσπου...»
- 12.4 Συνδυαστικά παραδείγματα

«Ό, τι επαναλαμβάνουμε, μας καθορίζει»

(Αριστοτέλης)

## 12.1. Εντολή «για πάντα εάν...»

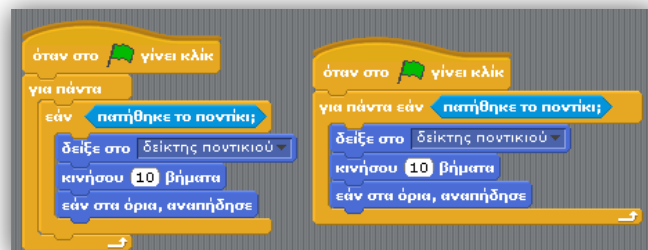
Όπως αναφέρθηκε και στο κεφάλαιο 10, η έννοια της επανάληψης παίζει κυρίαρχο ρόλο στη λογική της καθημερινότητάς μας. Πέρα όμως από τις δυο συγκεκριμένες μορφές επαναλήψεων που εξετάσαμε, δηλαδή τις επαναλήψεις που διαρκούν για πάντα (εντολή **για πάντα**) και τις επαναλήψεις που συμβαίνουν για γνωστό και συγκεκριμένο αριθμό φορών (εντολή **επανέλαβε X**), υπάρχουν και μορφές επαναλήψεων που διαρκούν όσο ισχύουν ορισμένες συνθήκες. Σε αυτές τις μορφές επαναλήψεων που γνωρίζουμε από την αρχή τον αριθμό των επαναλήψεων τους.

Πολλές φορές ξεκινάμε να βλέπουμε κάθε βδομάδα μια καινούρια τηλεοπτική σειρά που μας αρέσει εφόσον όμως δεν έχουμε κανονίσει να βγούμε με τους φίλους μας. Ή παίζουμε μπάλα συγκεκριμένη ημέρα, όταν όμως έχει καλό καιρό.



Υπάρχουν δηλαδή κάποιες ενέργειες που επαναλαμβάνονται μόνο εφόσον ισχύουν ορισμένες συνθήκες και για το λόγο αυτό δε γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων που θα συμβούν.

Πως θα μπορούσαμε να κάνουμε τον κ.Γατίδη να πηγαίνει προς το δείκτη του ποντικιού κάθε φορά που κρατάμε πατημένο το δεξί πλήκτρο του ποντικιού μας; Θέλουμε δηλαδή να κινείται για όσο όμως πατάμε το ποντίκι μας. Θα κυνηγάει δηλαδή το ποντίκι όσο του λέμε που βρίσκεται δηλαδή! Λογικό... Με βάση τις προηγούμενες γνώσεις μας κάτι τέτοιο δεν είναι δύσκολο. Χρειαζόμαστε μια εντολή **για πάντα** η οποία μέσα της θα περιέχει μια **εντολή εάν...** το ποντίκι πατήθηκε. Όταν θα είναι πατημένο το ποντίκι, ο γάτος μας θα δείχνει (κατεύθυνση) προς το δείκτη του ποντικιού και θα κινείται χρησιμοποιώντας την εντολή **κινήσου...βήματα**. Δείτε το αριστερό από τα δυο σενάρια της παρακάτω εικόνας:



Αυτή η υπό συνθήκη επανάληψη είναι πολύ συχρή στον προγραμματισμό και για αυτό το Scratch μας δίνει την εντολή για **παντα εάν...** που εμφανίζεται στο δεξί σενάριο. Η εντολή αυτή συμπεριφέρεται όπως περίπου μια εντολή **για πάντα** που περιέχει μέσα της μόνο μια **εάν...** και μας επιτρέπει να εκτελούμε ένα σύνολο εντολών για όσο ισχύει μια συνθήκη.



Ας δούμε όμως τη χρήση της εντολής σε ένα μεγαλύτερο παράδειγμα.

Όλοι κάποια στιγμή της ζωής μας έτυχε να παίξουμε με εφαρμογές κινητού. Μία αρκετά γνωστή εφαρμογή είναι η FaceWarp. Αυτή η εφαρμογή δίνει τη δυνατότητα στο χρήστη να παραμορφώνει είτε μία φωτογραφία που έχει στους φακέλους του είτε μία φωτογραφία που θα τραβήξει εκείνη τη στιγμή. Μεγαλώνει και μικραίνει τη μύτη και το χαμόγελο, αλλάζει το σχήμα του προσώπου και γενικά αλλάζει όλη τη φωτογραφία προκαλώντας πολύ γέλιο στην παρέα.

Τί σχέση όμως έχουν όλα αυτά με το Scratch;

Στο κεφάλαιο 7 μελετήσαμε τις εντολές της παλέτας **Όψεις** που μας έδιναν αντίστοιχα χαρακτηριστικά. Τί θα λέγατε αν συνδυάζαμε εκείνες τις εντολές, με την εντολή επανάληψης **για πάντα εάν...** για να φτιάξουμε μία εφαρμογή παρόμοια με αυτή των κινητών; Πιστεύετε ότι κάτι τέτοιο είναι δυνατό; Για πάμε να δούμε.

Σαν φωτογραφία που θα πειράζουμε, επιλέγουμε το αντικείμενο squareface από τη βιβλιοθήκη του Scratch. Αυτή η επιλογή

δεν είναι δεσμευτική, καθώς εσείς μπορείτε να βάλετε μία δική σας φωτογραφία ή μία φωτογραφία κάποιου φίλου σας.



Ορίζουμε τη συγκεκριμένη φωτογραφία ως υπόβαθρο στο σκηνικό μας, αφού πρώτα τη μεγεθύνουμε για να καλύπτει όλη την οθόνη. Στη συνέχεια δημιουργούμε 4 αντικείμενα με την ίδια μορφή και μικρότερο μέγεθος. Αυτά θα είναι τα αντικείμενα που θα τροποποιήσουμε.

Με την εντολή **πήγαινε στο x,y** τοποθετούμε τα 4 αντικείμενά μας στις 4 γωνίες του σκηνικού μας για την εκκίνηση του έργου και μέσω της εντολής **κινήσου ομαλά X δεύτ. στο x,y** τα μετακινούμε στο κέντρο του σκηνικού ώστε να ξεκινήσουμε με μία φαντασμαγορική αρχή! Αφού τα τοποθετήσαμε σωστά, πάμε να κάνουμε στο καθένα και μία μικρή αλλαγή!

Θέλουμε κάθε φορά που πατάμε το ποντίκι το κάθε αντικείμενο να κάνει μία αλλαγή στη μορφή του.

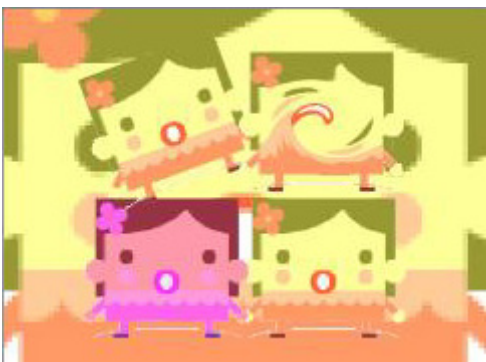
Έστω ότι το πρώτο αντικείμενο περιστρέφεται κατά 20 μοίρες, το δεύτερο χρησιμοποιεί το εφέ στροβίλιση, το τρίτο τα εφέ ψηφιδωτό και άλλαξε χρώμα και το τέταρτο αλλάζει το μέγεθός του.

Και τώρα αναρωτιόμαστε: Θέλουμε αυτή η αλλαγή να γίνει μία φορά; Όχι! Θέλουμε κάθε φορά που κάνουμε κλικ να γίνονται οι αλλαγές. Άρα, σίγουρα χρειαζόμαστε επανάληψη! Θέλουμε κάποια άλλη συνθήκη; Όχι. Μπορούμε λοιπόν να χρησιμοποιήσουμε την **Για πάντα-εάν**; Φυσικά.

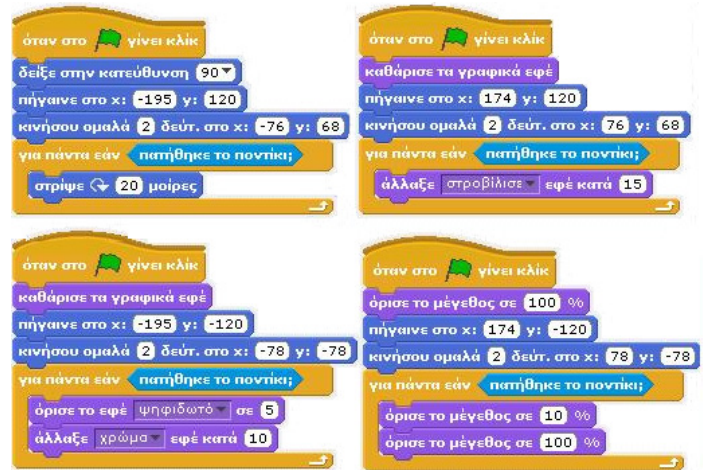
Το σκηνικό μας αρχικά έχει την εξής μορφή:



Ένα στιγμιότυπο αυτού είναι όταν έχουμε κάνει κλικ στο ποντίκι μας:



ενώ τα σενάρια για κάθε αντικείμενο είναι:



[12\_π01.sb]



Υπάρχουν πολλά περιθώρια βελτίωσης των σεναρίων. Μπορείτε να εντοπίσετε τα προβλήματά τους ή ευκαιρίες για να γίνουν πιο αποδοτικά:

Είναι σημαντικό να παρατηρήσουμε ότι και η εντολή **για πάντα-εάν...**, όπως η εντολή **για πάντα**, δεν δέχεται άλλες εντολές από κάτω της, δεν έχει την χαρακτηριστική εξοχή στην οποία μπορούμε να «κολλήσουμε» μια εντολή. Αυτό συμβαίνει γιατί όταν ο έλεγχος του έργου μας φτάσει στη συγκεκριμένη εντολή, τότε μέχρι το τέλος του έργου το Scratch θα ελέγχει τη συνθήκη και εφόσον ικανοποιείται θα εκτελεί τις εντολές που περιέχονται μέσα σε αυτήν.

Ας θεωρήσουμε ότι πρέπει να υλοποιήσουμε στο Scratch την εξής περιγραφή «*Αν το ελεφαντάκι ακουμπήσει το έντομο, τότε το έντομο πρέπει να πάει πίσω στην αρχική του θέση και να πει ότι έχασε*». Σε ποιου αντικείμενου το σενάριο, θα εισάγουμε την εντολή **εάν...**; Αρχικά θα σκεφτόμασταν να εισάγουμε μια εντολή **εάν...** στο αντικείμενο ελεφαντάκι. Αλλά αυτό θα μας οδηγούσε σε ένα αδιέξοδο, καθώς από το σενάριο που αφορά το ελεφαντάκι δεν μπορούμε να κινήσουμε το έντομο αλλά μόνο το ελεφαντάκι.

Το σημείο αυτό θέλει προσοχή. Οι συνθήκες των εντολών **εάν...** μπορούν να περιέχουν αντικείμενα με οποιαδήποτε σειρά αλλά εμείς ενδιαφερόμαστε να προγραμματίσουμε το αποτέλεσμα, το τι θα συμβεί αν ισχύσει η συνθήκη. Αυτά που θα συμβούν ποιο αντικείμενο αφορούν; Το έντομο. Σε αυτό το αντικείμενο θα εισάγουμε την αντίστοιχη εντολή.

## 12.2 Εντολή «επανάλαβε ώσπου»

Είναι 11 η ώρα το πρωί και δεν έχετε σχολείο. Αποφασίζετε λοιπόν να κοιμηθείτε λίγο παραπάνω! Ξαφνικά ακούτε από μέσα μία φωνή πολύ γνωστή που να λέει: «ΰπνα! Πρέπει να σηκωθείς για να συμμαζέψεις το δωμάτιο σου!». Και να! Είναι η φωνή της μητέρας. Τί σχέση όμως μπορεί να έχει αυτό με το μάθημά μας;

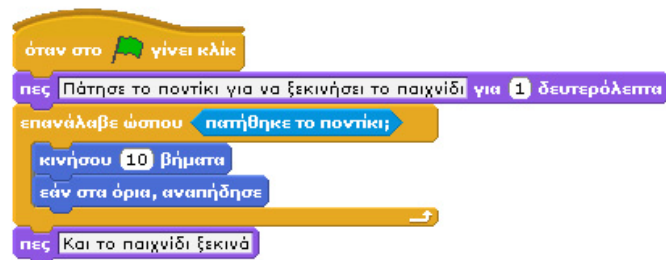
Επανάληψη: Η μαμά θα συνεχίσει να φωνάζει ότι πρέπει να σηκωθείτε για να συμμαζέψετε το δωμάτιο.

Συνθήκη: Μέχρι να σηκωθείτε από το κρεβάτι.

Το σίγουρο λοιπόν είναι ότι θα συνεχίσει να φωνάζει μέχρι εσείς να ξυπνήσετε και να σηκωθείτε από το κρεβάτι. Δηλαδή μέχρι η συνθήκη (αν σηκωθήκατε από το κρεβάτι) να γίνει αληθής!!!

Στην παλέτα **Έλεγχος** υπάρχει η εντολή της μαμάς ☺ **επανάλαβε ώσπου**. Μέσω αυτής της επιλογής μπορούμε να εξασφαλίσουμε τη συνεχή επανάληψη εντολών ώσπου η συνθήκη ελέγχου να γίνει αληθής. Για παράδειγμα, μπορούμε να σκεφτούμε την πρώτη οθόνη ενός παιχνιδιού, στην οποία ο χαρακτήρας κινείται ασταμάτητα μέχρι να πατήσουμε το ποντίκι και να ξεκινήσει το παιχνίδι. Πως θα μπορούσαμε να υλοποιήσουμε κάτι τέτοιο:

Μελετήστε το παρακάτω πολύ απλό σενάριο:



Η **επανάλαβε ώσπου** κάνει τον πρωταγωνιστή μας να κινείται μεταξύ των ορίων της οθόνης **μέχρι** να πατήσουμε το ποντίκι.

Η εντολή **επανάλαβε ώσπου** δεν επαναλαμβάνει τις εντολές που περιέχει για συγκεκριμένο αριθμό φορών όπως η **επανάλαβε Χ**. Αντίθετα, στην **επανάλαβε ώσπου**:

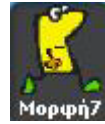
- ✓ αν η συνθήκη είναι **αληθής** από την αρχή, τότε οι περιεχόμενες εντολές **δεν** θα εκτελεστούν καμία φορά!
- ✓ αν η συνθήκη είναι **ψευδής**, θα επαναλαμβάνει τις εντολές **μέχρι** η συνθήκη να γίνει αληθής.

Μοιάζει επίσης με την εντολή **για πάντα εάν...** με τη διαφορά ότι

- ✓ οι συνθήκες είναι αντιστροφές, αφού στην **για πάντα εάν...** αν η συνθήκη είναι αληθής, πραγματοποιούνται οι επαναλήψεις, ενώ στην εντολή **επανάλαβε ώσπου** οι επαναλήψεις πραγματοποιούνται όσο η συνθήκη είναι ψευδής
- ✓ ο έλεγχος της συνθήκης στην **επανάλαβε ώσπου** δεν γίνεται καθ' όλη τη διάρκεια του έργου αλλά μέχρι να ικανοποιηθεί η συνθήκη. Αν ικανοποιηθεί η συνθήκη το πρόγραμμα εκτελεί την επόμενη εντολή και για αυτό επιτρέπεται να προστεθούν άλλες εντολές από κάτω της. Αντίθετα, όταν ξεκινήσει μια **για πάντα εάν...** ο έλεγχος δεν σταματά παρά μόνο όταν τερματιστεί ο έλεγχος.

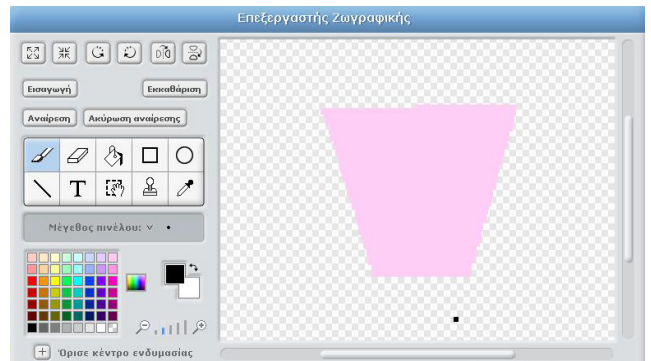
Ας δούμε τη χρήση της μέσα σε ένα παράδειγμα. Θέλουμε ο ήρωας μας να μαζεύει όλα τα επικίνδυνα κοχύλια που δεν σταματούν να κουνιούνται μέσα σε έναν κουβά για να καθαρήσει την παραλία!

Τί χρειαζόμαστε; Χρειαζόμαστε σίγουρα κάποιον να κάνει τη βρώμικη(?) δουλειά! Έστω ότι επιλέγουμε το αντικείμενο **fantasy1-b** από τον κατάλογο **fantasy** της βιβλιοθήκης αντικειμένων του Scratch.

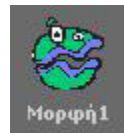


Επίσης πρέπει να έχουμε έναν κουβά για να μαζεύουμε τα κοχύλια. Ας τον δημιουργήσουμε εμείς!

Με τον επεξεργαστή ζωγραφικής φτιάχνουμε ένα απλό τραπέζιο σε μορφή κουβά και το χρωματίζουμε στο χρώμα της αρεσκείας μας. Εμείς θα επιλέξουμε το ροζ κι έτσι θα έχουμε:



Τέλος χρειαζόμαστε τα κοχύλια τα οποία επιλέγουμε να είναι τα (κατάλογος **animals**, μορφή **clam1**):



Φυσικά για σκηνικό επιλέγουμε μία εξωτική παραλία! Το παιχνίδι μας θα ξεκινήσει από την εξής κατάσταση:



Πάμε όμως να δούμε τι θέλουμε να κάνει το κάθε αντικείμενο ξεχωριστά.

**Ο... συλλέκτης:** Σαν χρήστες της εφαρμογής, θέλουμε να μπορούμε να μετακινούμε το πρωταγωνιστή μας με τα βελάκια προς όποια κατεύθυνση επιθυμούμε. Επιπρόσθετα, ο πρωταγωνιστής μας θα πρέπει να βρίσκεται σε συγκεκριμένη θέση στην αρχή του παιχνιδιού. Το σενάριο του λοιπόν είναι το παρακάτω:



```

όταν στο  γίνει κλικ
πήγαινε στο x: -200 y: 44

```

```

όταν το πλήκτρο  πάω βέλος  πατηθεί
άλλαξε y κατά 5

```

```

όταν το πλήκτρο  κάτω βέλος  πατηθεί
άλλαξε y κατά -5

```

```

όταν το πλήκτρο  δεξί βέλος  πατηθεί
άλλαξε x κατά 5

```

```

όταν το πλήκτρο  αριστερό βέλος  πατηθεί
άλλαξε x κατά -5

```

**Κοχύλια:** Θέλουμε να δώσουμε στα κοχύλια μία συνεχόμενη κίνηση μπρος-πίσω για να τα κάνουμε να μοιάζουν λίγο επικίνδυνα! Αυτή η κίνηση σταματάει μόλις ο συλλέκτης τα πιάσει (στη γλώσσα μας = ακουμπήσει) και αυτά αμέσως μετακινούνται στον ροζ κουβά μας! Θέλουμε δηλαδή να κινούνται συνέχεια, ώσπου ο συλλέκτης να τα πιάσει. Αυτή είναι δουλειά της **επανάλαβε ώσπου...** Επομένως, αφού δημιουργήσουμε 5 κοχύλια, και τα τοποθετούμε στην αρχή του έργου σε συγκεκριμένες θέσεις, χρησιμοποιώντας την εντολή **επανάλαβε ώσπου** τα κάνουμε να κινούνται δεξιά και αριστερά με τη χρήση της εντολής **κινήσου ομαλά X δεύτ. στο x,y**. Η κίνηση θα σταματά όταν θα τα αγγίζει ο συλλέκτης, δηλαδή θα κινούνται «**ώσπου να τα αγγίξει ο συλλέκτης**». Όταν τα πιάνει, θα μετακινούνται στον κουβά. Τα σενάρια όλων των κοχυλιών είναι πανομοιότυπα και οι μόνες τιμές που αλλάζουν είναι κάθε φορά οι αρχικές και τελικές θέσεις τους. Ένα από αυτά τα σενάρια:

```

όταν στο  γίνει κλικ
πήγαινε στο x: -90 y: -146
επανάλαβε ώσπου  αγγίζει το  ;
κινήσου ομαλά 1 δεύτ. στο x: -70 y: -146
κινήσου ομαλά 1 δεύτ. στο x: -90 y: -146
πήγαινε στο x: 148 y: -114
σταμάτησε το σενάριο

```



[12\_π02.sb]

Μέσω της εντολής **επανάλαβε ώσπου** έχουμε τη δυνατότητα να εκτελούμε επαναληπτικά κάποιες εντολές χωρίς προκαθορισμένο αριθμό επαναλήψεων και μετά το πέρας αυτής της επανάληψης να βάζουμε επιπλέον εντολές! Αυτό είναι πολύ σημαντικό γιατί μέχρι τώρα με την **για πάντα** και την **για πάντα εάν...** δε μας δινόταν αυτή η δυνατότητα.

Για να καταλάβετε καλύτερα τις διαφορές τους όμως, δοκιμάστε να αλλάξετε τις εντολές κίνησης του συλλέκτη χρησιμοποιώντας αρχικά την εντολή **για πάντα εάν...** και μετά την **επανάλαβε ώσπου**. Να είστε σίγουροι ότι με αυτόν τον τρόπο θα καταλάβετε όλες τις διαφορές τους!

### 12.3 Εντολή "περίμενε ώσπου..."

Χτυπάει το κουδούνι για το μεγάλο διάλειμμα! Η μυρωδιά του κρουασάν που ψήθηκε είναι ακαταμάχητη! Η μόνη λύση είναι να πάτε να το αγοράσετε από το κυλικείο του σχολείου σας. Με ένα μεγάλο όμως τίμημα: θα πρέπει να περιμένετε ώσπου να έρθει η σειρά σας...

Η έννοια της αναμονής είναι ένα επίσης αναπόσπαστο κομμάτι της καθημερινότητας μας: η αναμονή στο super-market μέχρι να πληρώσετε, η αναμονή σε ένα τηλεφώνημα που κάνετε μέχρι ο συνομιλητής σας να σηκώσει το τηλέφωνο, αναμονή όταν περιμένετε να ξεκινήσετε να παίζετε ένα παιχνίδι μέχρι να επιλέξετε τον ήρωά σας, αναμονή για την εμφάνιση ενός κακού χαρακτήρα στο παιχνίδι που θέλει να σας εξοντώσει.

Υπάρχουν φορές στον προγραμματισμό που θέλουμε να σταματήσει η εκτέλεση ενός σεναρίου μέχρι να συμβεί κάτι; Πολλές. Για το λόγο αυτό, το Scratch μας προσφέρει την εντολή **περίμενε ώσπου...** η οποία σταματά την εκτέλεση της ροής



του σεναρίου μέχρι να γίνει η συνθήκη που περιέχει η εντολή αληθής. Η εντολή **περίμενε ώσπου X** δεν περιέχει στο εσωτερικό της άλλες εντολές. Όλες οι εντολές που την ακολουθούν, θα εκτελεστούν εφόσον η συνθήκη της γίνει αληθής

Έστω για παράδειγμα ότι ο κ.Γατίδης θέλουμε να δημιουργεί ένα σχέδιο, π.χ. δυο τετράγωνα, με μια συγκεκριμένη σειρά αλλά μόνο αφού του δώσουμε το OK. Πως θα μπορούσαμε να το κάνουμε χρησιμοποιώντας την εντολή **περίμενε ώσπου X**; Μελετήστε το παρακάτω σενάριο:

```

όταν στο  γίνει κλικ
καθάρισε
πήγαινε στο x: -75 y: 0
κατέβασε πένα
περίμενε ώσπου  πατήθηκε πλήκτρο 1 ;
επανάλαβε 4
κινήσου 100 βήματα
στρίψε 90 μοίρες
περίμενε ώσπου  πατήθηκε πλήκτρο 2 ;
σήκωσε πένα
πήγαινε στο x: 75 y: 0
κατέβασε πένα
επανάλαβε 4
κινήσου 100 βήματα
στρίψε 90 μοίρες
σήκωσε πένα

```

[12\_π03.sb]

Όταν τρέξουμε το έργο μας, αρχικά θα εκτελεστούν οι τρεις πρώτες εντολές και το σενάριο θα σταματήσει στην πρώτη εντολή **περίμενε ώσπου...** Η εντολή θα επιτρέψει την εκτέλεση των επόμενων εντολών μόνο εφόσον ικανοποιηθεί η συνθήκη της, δηλαδή όταν πατηθεί το πλήκτρο 1. Αν πατήσουμε το αντίστοιχο πλήκτρο, μέσω της εντολής **επανάλαβε...** ο



κ.Γατίδης θα σχεδιάσει ένα τετράγωνο και στη συνέχεια η εκτέλεση των εντολών θα σταματήσει πάλι στην επόμενη εντολή περίμενε ώσπου.... Πρέπει να πατήσουμε 2 ώστε να σχεδιαστεί και το δεύτερο τετράγωνο. Η χρησιμότητα της περίμενε ώσπου... είναι ολοφάνερη. Δεν θα μπορούσαμε με καμία άλλη εντολή να καταφέρουμε το ίδιο αποτέλεσμα μέσα σε ένα μόνο σενάριο (ίσως μόνο με ορισμένα τρικ). Συνεπώς, με την περίμενε ώσπου... μπορούμε να δημιουργούμε σενάρια που τα διαδοχικά τους βήματα θα εκτελούνται σε διαφορετικές χρονικές στιγμές ανάλογα με τις συνθήκες που θα έχουμε ορίσει.

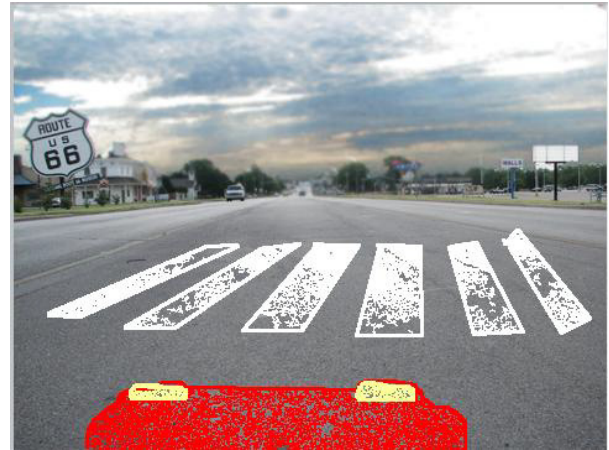
Θυμηθείτε ότι έχουμε χρησιμοποιήσει αρκετές φορές και την εντολή περίμενε...δευτερόλεπτα, η οποία λειτουργεί κατά αντίστοιχο τρόπο, με τη διαφορά όμως ότι απαιτεί να γνωρίζουμε εκ των προτέρων σε πόσο χρόνο θα αρχίσει να εκτελείται το επόμενο κομμάτι εντολών που την ακολουθεί. Είναι εξαιρετικά δύσκολο σε ένα παιχνίδι να γνωρίζουμε τη συμπεριφορά των αντικειμένων σε επίπεδο δευτερολέπτων από πριν και εκεί η εντολή περίμενε ώσπου... είναι ακόμη πιο καθοριστική.

Ένα μεγαλύτερο παράδειγμα; Πώς θα σας φαινόταν η ιδέα να ζούσαμε σε έναν ιδανικό κόσμο όπου τα φανάρια θα κοκκίνιζαν για τους οδηγούς και θα πρασίνιζαν για τους πεζούς με το πάτημα ενός μόνο κουμπιού; Πάμε να το φτιάξουμε στο Scratch!

Αρχικά χρειαζόμαστε το σκηνικό να είναι κάποιος δρόμος. Έστω ότι επιλέγουμε την εικόνα «rout66» που βρίσκεται στο φάκελο έτοιμων υποβάθρων «outdoors»:



Μέσω της ζωγραφικής, φτιάχνουμε διάβαση για τους πεζούς και την μπροστινή πλευρά ενός αμαξίου. Το αποτέλεσμα είναι κάπως έτσι (μάλλον εσείς θα τα καταφέρετε καλύτερα ☺):



Στη συνέχεια και πάλι με τη βοήθεια του επεξεργαστή ζωγραφικής δημιουργούμε ως ξεχωριστά αντικείμενα το φανάρι, το κόκκινο και το πράσινο, όπως φαίνεται στις 3 εικόνες που ακολουθούν:



Τέλος, μένει να επιλέξουμε το χαρακτήρα που θα περάσει το δρόμο καθώς και να δημιουργήσουμε το κουμπί.

Όπως σε προηγούμενο παράδειγμα, επιλέξαμε από τη βιβλιοθήκη του Scratch και το φάκελο People, το αντικείμενο boy4-walking και τις τέσσερις σχετικές ενδυμασίες για να του δώσουμε την αίσθηση της κίνησης.



Το κουμπί το φτιάχνουμε πολύ απλά και πάλι μέσω της ζωγραφικής, φτιάχνοντας ένα μήνυμα και από κάτω ένα κουμπί.

Το αρχικό σκηνικό μας θα είναι:



Το πράσινο φανάρι για τα αυτοκίνητα είναι πάντα αναμμένο πράσινο εκτός από τη στιγμή που κάποιος πεζός πατήσει το κουμπί. Τότε αυτόματα κοκκινίζει το φανάρι και ο πεζός έχει 10 δευτερόλεπτα για να περάσει απέναντι. Σημειώνουμε ότι ο πεζός πρέπει να περιμένει μέχρι το φανάρι των αυτοκινήτων να γίνει κόκκινο και δεν μπορεί να κάνει καμία κίνηση.

Πάμε να το δούμε αναλυτικότερα:

**Κόκκινο και Πράσινο:** Το πράσινο είναι αναμμένο και σταματάει για 10 δευτερόλεπτα αν ο πεζός πατήσει το κουμπί και το κόκκινο το αντίθετο. Τα σενάρια τους παρουσιάζονται στις παρακάτω εικόνες:

```

όταν στο [πράσινο] γίνει κλικ
για πάντα
  απόκρυψη
  εάν πατήθηκε το ποντίκι;
    πήγαινε στο x: 82 y: 132
    εμφάνισε
    περίμενε 10 δευτερόλεπτα
  →

```

```

όταν στο [κόκκινο] γίνει κλικ
για πάντα
  εμφάνισε
  πήγαινε στο x: 82 y: 76
  εάν πατήθηκε το ποντίκι;
    απόκρυψη
    περίμενε 10 δευτερόλεπτα
  →

```

**Ο πεζός:** Ο πεζός θα προχωράει με την εναλλαγή ενδυμασιών κάθε φορά που θα πατάμε το δεξί βελάκι, αλλά πρώτα θα πρέπει να περιμένει να πατηθεί το ποντίκι, γιατί αλλιώς δεν μπορεί να περάσει.

Πώς όμως θα το πετύχουμε αυτό; Με την εντολή **περίμενε ώσπου...** μπορούμε να αναγκάσουμε το αντικείμενο να περιμένει, χωρίς να εκτελεί καμία εντολή μέχρι η συνθήκη να γίνει αληθής. Στο συγκεκριμένο παράδειγμα, η συνθήκη είναι αν πατήθηκε το ποντίκι οπότε το σενάριο θα είναι το παρακάτω:

```

όταν στο [πράσινο] γίνει κλικ
πήγαινε στο x: -198 y: -13
περίμενε ώσπου πατήθηκε το ποντίκι;
για πάντα εάν πατήθηκε πλήκτρο [δεξί βέλος];
  κινήσου 7 βήματα
  αλλαγή σε ενδυμασία boy4-walking-a
  περίμενε 0.2 δευτερόλεπτα
  κινήσου 7 βήματα
  αλλαγή σε ενδυμασία boy4-walking-b
  περίμενε 0.2 δευτερόλεπτα
  κινήσου 7 βήματα
  αλλαγή σε ενδυμασία boy4-walking-c
  περίμενε 0.2 δευτερόλεπτα
  κινήσου 7 βήματα
  αλλαγή σε ενδυμασία boy4-walking-d
  →

```

[12\_π04.sb]

Σύμφωνα με το παραπάνω σενάριο, θα πρέπει να πατήσουμε πρώτα το ποντίκι ώστε να αλλάξει το χρώμα του φαναριού και στη συνέχεια, θα μπορέσουμε με το δεξί βέλος να κινήσουμε προς τα δεξιά τον πρωταγωνιστή μας. Στο σενάριο έχουμε ενσωματώσει μια σειρά από λαθάκια τα οποία μπορούν να

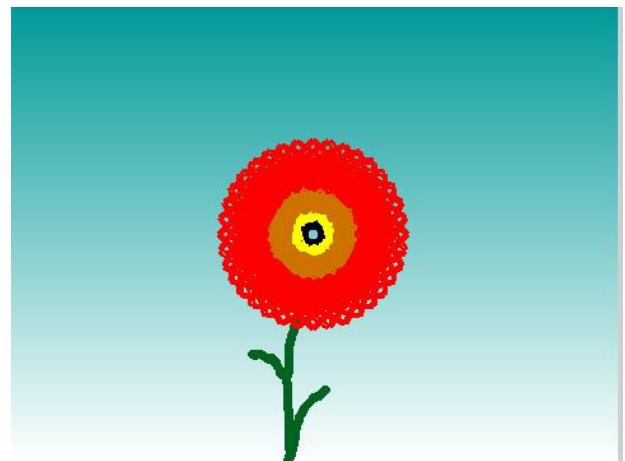
αποφευχθούν. Προσπαθήστε να απαντήσετε στα επόμενα ερωτήματα:

- A) Ποιος είναι ο ελάχιστος αριθμός αντικειμένων με τα οποία θα μπορούσαμε να είχαμε το ίδιο αποτέλεσμα;
- B) Πως θα μπορούσε να γίνει πιο «αποδοτικό» το σενάριο των δυο φαναριών;
- Γ) Πως θα μπορούσε να γίνει πιο «αποδοτικό» το σενάριο του χαρακτήρα που περνά από το δρόμο;
- Δ) Υπάρχει δυνατότητα να ξαναγραφτεί το σενάριο του χαρακτήρα ώστε αφού περάσει το δρόμο, να μπορεί να γίνει και η αντίστροφη κίνηση με το πάτημα του ποντικιού;
- Ε) Πως θα μπορούσαμε να ελέγξουμε το κατά πόσο ο χαρακτήρας μας πρόλαβε να περάσει απέναντι όσο το φανάρι ήταν κόκκινο για τα αυτοκίνητα;

### 12.4 Συνδυασμός εντολών επανάληψης

Στην ενότητα αυτή θα προσπαθήσουμε να κατανοήσουμε καλύτερα τις διαφορετικές εντολές επανάληψης που έχουμε παρουσιάσει μέσα από διαφορετικά παραδείγματα.

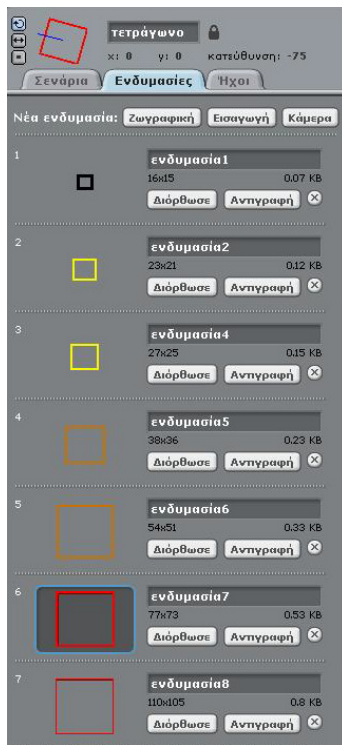
**Παράδειγμα «Ένα λουλούδι»:** Αυτό το παράδειγμα έχει ως στόχο τη δημιουργία του άνθους ενός λουλουδιού σε ένα μπλε φόντο όπου υπάρχει ήδη ο μίσχος του, ενώ παράλληλα ακούγεται το τραγούδι Suspense. Ο μίσχος θέλουμε να είναι τοποθετημένος στο κέντρο του παραθύρου ενώ το άνθος θα δημιουργηθεί στο πάνω άκρο του όπως φαίνεται στην επόμενη εικόνα:



Όπως είδαμε και σε προηγούμενα κεφάλαια, η δημιουργία τέτοιων πολύπλοκων σχημάτων βασίζεται συνήθως σε σχήματα που αφήνουν επαναλαμβανόμενα τη σφραγίδα τους σε διαφορετικά σημεία ή με διαφορετική κατεύθυνση. Κάτι αντίστοιχο θα κάνουμε και στο συγκεκριμένο παράδειγμα.

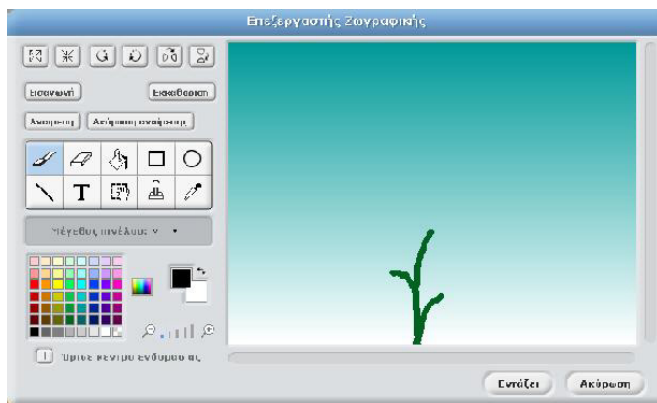
Η δημιουργία του άνθους θα βασιστεί σε ένα αντικείμενο με επτά διαφορετικές ενδυμασίες. Όλες οι ενδυμασίες θα είναι περιγράμματα τετραγώνων διαφορετικού μεγέθους και διαφορετικών χρωμάτων. Πιο συγκεκριμένα χρειαζόμαστε τα περιγράμματα ενός μαύρου τετραγώνου 16x15, δύο κίτρινων τετραγώνων 23x21 και 27x25, δύο πορτοκαλί τετραγώνων 38x36 και 54x51 και δύο κόκκινων διαστάσεων 77x73 και 110x105. Οι διαφορετικές ενδυμασίες φαίνονται στην επόμενη εικόνα.





Η δημιουργία του άνθους θα ξεκινάει από το μαύρο τετράγωνο, την αρχική δηλαδή ενδυμασία, το οποίο θα αφήνει μια σφραγίδα ενώ στη συνέχεια θα επιλέγεται η επόμενη ενδυμασία, το αντικείμενό μας θα στρίβει δεξιά κατά 10 μοίρες και θα αφήνει το αποτύπωμα της νέας ενδυμασίας. Η διαδικασία αυτή θέλουμε να επαναλαμβάνεται έως ότου σχηματιστεί το άνθος που φαίνεται στην προηγούμενη εικόνα.

Δημιουργούμε το φόντο χρησιμοποιώντας τον επεξεργαστή ζωγραφικής.



Για να ξεκινάει η δημιουργία του άνθους από το μαύρο τετράγωνο θα χρησιμοποιήσουμε την εντολή αλλαγή σε ενδυμασία (ενδυμασία1). Για να δοκιμάσουμε την προσοχή σας στις προηγούμενες εικόνες, για την επανάληψη των βημάτων επιλέγουμε την εντολή επανάλαβε ώσπου με συνθήκη «ενδυμασία # =8» (τι σημαίνει αυτό;). Στο σώμα της επανάληψης θα περιλάβουμε τη δημιουργία αποτυπώματος με την εντολή σφραγίδα, την περιστροφή του αντικειμένου κατά 10 μοίρες και μια μικρή αναμονή πριν προχωρήσει το έργο μας στην επόμενη σφραγίδα για να προλάβουμε να παρατηρήσουμε πως σχηματίζεται το άνθος. Το σενάριο παρουσιάζεται παρακάτω:



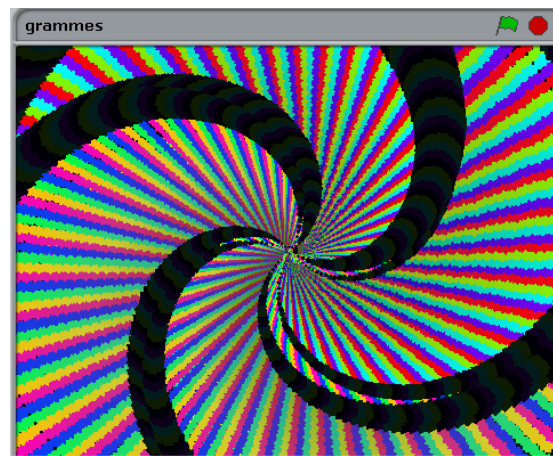
[12\_π05.sb]



Η επανάληψη σε συνδυασμό με την εντολή επόμενη ενδυμασία επιτυγχάνει το στόχο μας. Ή μήπως όχι;

Παρατηρήστε ότι παρότι το όνομα της τελευταίας ενδυμασίας είναι το «ενδυμασία 8», οι ενδυμασίες είναι 7 και η τελευταία ενδυμασία προσδιορίζεται με τον ίδιο αριθμό (λείπει η ενδυμασία 3!). Άρα η συνθήκη δε θα γίνει ποτέ αληθής και η επανάληψη θα συνεχίσει μέχρι να τερματίσει ο χρήστης το έργο. Άρα η συγκεκριμένη μορφή της εντολής επανάλαβε ώσπου... λειτουργεί όπως μια εντολή για πάντα! Τι θα συνέβαινε αν στη συνθήκη αλλάζαμε το 8 σε 7; Θα τυπωνόταν το άνθος; Ποια θεωρείτε ότι είναι η καταλληλότερη εντολή για το συγκεκριμένο στόχο (για πάντα, επανάλαβε ώσπου..., επανάλαβε Χ). Αν απαντήσατε η Τρίτη, μαντέψατε σωστά. Μπορείτε να μας εξηγήσετε και το γιατί;

Μελετήστε από μόνοι σας ένα ανάλογο παράδειγμα που έχει ως αποτέλεσμα το σχήμα της επόμενης εικόνας:



[12\_π06.sb]

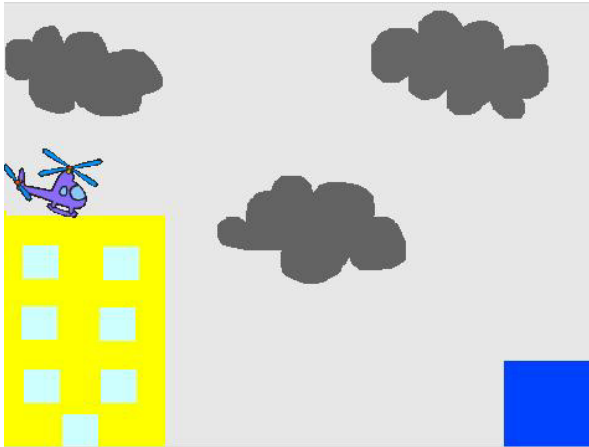


Παράδειγμα «Το ελικόπτερο πέφτει...»: Έχουμε εξετάσει σε προηγούμενο κεφάλαιο αντίστοιχο παράδειγμα. Ζητούμενό μας είναι ο χρήστης να μπορεί να ελέγχει ένα ελικόπτερο που πρέπει να απογειώνεται από μια συγκεκριμένη θέση στην ταράτσα ενός κίτρινου ουρανοξύστη με στόχο να το προσγειώσει σε ένα μπλε ελικοδρόμιο, οπότε και τερματίζει το παιχνίδι.

Πιο συγκεκριμένα, το ελικόπτερο θα πρέπει να κινείται προς τα πάνω όταν πατάμε το πάνω βελάκι του πληκτρολογίου, ενώ αν δεν το πατάμε (και το ελικόπτερο δεν βρίσκεται πάνω στον κίτρινο ουρανοξύστη) θα πρέπει να χάνει ύψος. Όταν πατάμε το δεξί βελάκι θα μετακινείται προς τα δεξιά, ενώ διόρθωση με το αριστερό βελάκι δεν θα υπάρχει! Όσο κινείται στον αέρα θα

πρέπει να ακούγεται αντίστοιχος ήχος για τον έλικα, ενώ όταν ακουμπήσει την μπλε πλατφόρμα θα πρέπει να εμφανίζει το μήνυμα «Τέλος παιχνιδιού» για 1 δευτερόλεπτο και να τελειώνει η εκτέλεση του έργου μας. Αν ακουμπήσει σύννεφο, τότε θα πρέπει να γυρίζει στο σημείο εκκίνησης εμφανίζοντας το μήνυμα «Προσπάθησε ξανά!» για 1 δευτερόλεπτο.

Για να δημιουργήσουμε το σκηνικό θα χρησιμοποιήσουμε τον επεξεργαστή ζωγραφικής. Χρειαζόμαστε ένα γκρι ανοιχτό φόντο, στην κάτω αριστερή γωνία ένα κίτρινο ουρανοξύστη, στην κάτω δεξιά ένα μπλε ελικοδρόμιο και τρία σύννεφα χρώματος γκρι σκούρο διάσπαρτα στον ουρανό. Το ολοκληρωμένο σκηνικό μαζί με το ελικόπτερο φαίνονται στην επόμενη εικόνα.



Για να δημιουργήσουμε το ελικόπτερο, επιλέγουμε τη μορφή helicopter1 από τον κατάλογο transportation της βιβλιοθήκης αντικειμένων του Scratch. Η συμπεριφορά του ελικοπτερού είναι σχετικά απλή. Για την ανοδική κίνηση θα πρέπει να αυξάνουμε τη συντεταγμένη y της θέσης του, ενώ για την κίνηση προς τα δεξιά θα πρέπει να αυξάνουμε την συντεταγμένη x της θέσης του. Για να πέφτει το ελικόπτερο θα μειώνουμε την συντεταγμένη y της θέσης του. Για να ακούγεται ο ήχος του έλικα, εισάγουμε στους ήχους τους αντικειμένου τον ήχο Peculiar από το φάκελο Electronic του Scratch.

Θα βάλουμε το ελικόπτερο αρχικά πάνω στην κίτρινη πολυκατοικία και στη συνέχεια με το συνδυασμό των εντολών **για πάντα** και **εάν...** θα ελέγχουμε για όλες τις παραπάνω συνθήκες κίνησης.

Οι συνθήκες που πρέπει να εξεταστούν είναι:

- A) *αν αγγίζει το κίτρινο χρώμα*, δηλαδή αν βρίσκεται πάνω στην πολυκατοικία, όπου δεν πρέπει να ακούγεται ο ήχος του έλικα,
- B) *αν δεν αγγίζει το κίτρινο χρώμα*, δηλαδή αν βρίσκεται στον αέρα, οπότε πρέπει να χάνει λίγο ύψος και να ακούγεται ο ήχος του έλικα,
- Γ) *αν πατήθηκε το πάνω βέλος*, οπότε πρέπει να αυξάνουμε το ύψος του,
- Δ) *αν πατήθηκε το δεξί βέλος*, οπότε πρέπει να μετακινείται προς τα δεξιά,
- Ε) *αν αγγίζει το μπλε χρώμα*, οπότε πρέπει να βγει το αντίστοιχο μήνυμα τέλους και να ολοκληρωθεί το έργο μας.

```

όταν στο [ ] γίνει κλικ
  πήγαινε στο x: [-198] y: [33]
  για πάντα
    εάν αγγίζει το χρώμα [κίτρινο] ;
      σταμάτησε όλους τους ήχους
    εάν όχι αγγίζει το χρώμα [κίτρινο] ;
      άλλαξε y κατά [-1]
      παιξε ήχο Peculiar
    εάν πατήθηκε πλήκτρο [πάνω βέλος] ;
      άλλαξε y κατά [3]
    εάν πατήθηκε πλήκτρο [δεξί βέλος] ;
      άλλαξε x κατά [3]
    εάν αγγίζει το χρώμα [μπλε] ;
      πες [Τέλος παιχνιδιού!] για [1] δευτερόλεπτα
      σταμάτησε τα όλα
  
```

Επιπλέον, για την περίπτωση που το ελικοπτεράκι πέσει πάνω στα σύννεφα, θα χρησιμοποιήσουμε την εντολή **για πάντα εάν** ... με συνθήκη το αν αγγίζει το χρώμα γκρι.

```

όταν στο [ ] γίνει κλικ
  για πάντα εάν αγγίζει το χρώμα [γκρι] ;
    πήγαινε στο x: [-198] y: [33]
    πες [Προσπάθησε ξανά!] για [2] δευτερόλεπτα
  
```

[12\_π07.sb]



Φαντάζεστε κάποιο για ποιο λόγο εισάγαμε το συγκεκριμένο σενάριο ξεχωριστά από τους υπόλοιπους ελέγχους; Θα μπορούσαμε να είχαμε εισάγει τις αντίστοιχες εντολές στο πρώτο σενάριο χωρίς καμία αλλαγή; Αν όχι, με ποιες αλλαγές θα μπορούσαν να εισαχθούν οι συγκεκριμένες εντολές στο πρώτο σενάριο;

Θα μπορούσαμε να έχουμε μόνο μια **εάν...αλλιώς** στο πρώτο σενάριο για να ελέγξουμε την κίνηση στον κάθετο άξονα (όταν είναι στον αέρα και όταν πατιέται το πάνω βελάκι); Επίσης, ο έλεγχος της κίνησης θα μπορούσε να γίνει με τη χρήση της εντολής **όταν πατηθεί το πλήκτρο...**;

Όπως μπορείτε να παρατηρήσετε, οι εναλλακτικοί δρόμοι που μας παρέχει το Scratch για να λύσουμε τα προβλήματά μας είναι πολλοί και κάποιες φορές έντονα διαφορετικοί. Δεν υπάρχει μια λύση, δεν υπάρχει μια μόνο εντολή επανάληψης που μπορεί χρησιμοποιηθεί.

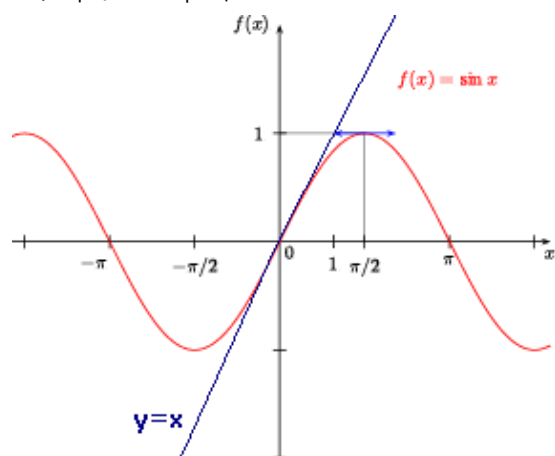
### 12.3 Τελεστές, συνθήκες και άλλα!

Όπως έχει διαφανεί από όλα τα προηγούμενα παραδείγματα, η κατασκευή κατάλληλων συνθηκών στις λέξεις **εάν...**, **εάν...αλλιώς**, **για πάντα εάν...**, **περίμενε ώσπου...**, **επανάλαβε ώσπου...**, είναι εξαιρετικά σημαντική. Συζητήσαμε στο προηγούμενο κεφάλαιο για τους τελεστές συσχέτισης συνθηκών (...ή..., ...και..., όχι...) αλλά βλέπουμε στα παραδείγματά μας ότι υπάρχουν πολλά διαφορετικά στοιχεία και



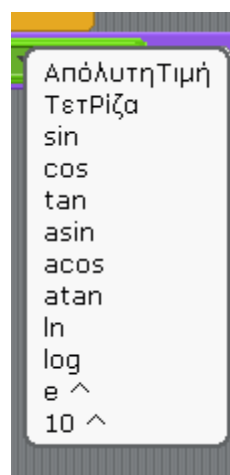
επιπλέον τελεστές που μπορούμε να αξιοποιήσουμε στη διαμόρφωση συνθηκών και όχι μόνο.

**Παράδειγμα «Ημίτονο και ζωγραφική!»:** Έχει δει στα μαθηματικά τη γραφική παράσταση της συνάρτησης του ημιτόνου; Σας θυμίζει κάτι η παρακάτω εικόνα;

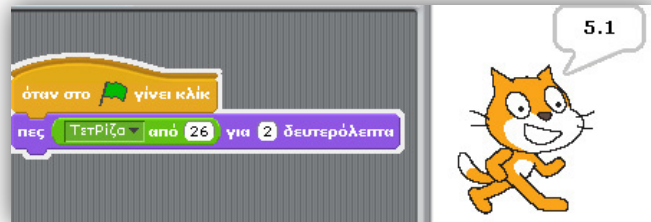


Θα προσπαθήσουμε να την αναπαράγουμε στο Scratch. Ας ξεκινήσουμε όμως από τα βασικά. Το Scratch εμπεριέχει τον αυτόματο υπολογισμό πληθώρας μαθηματικών συναρτήσεων. Η αντίστοιχη εντολή που μας επιστρέφει το αποτέλεσμα των μαθηματικών συναρτήσεων βρίσκεται στην παλέτα **Τελεστές** και έχει ως προεπιλεγμένη συνάρτηση την τετραγωνική ρίζα.

Όλες οι μαθηματικές συναρτήσεις απαιτούν μια τιμή εισόδου (το λευκό κουτάκι στην εντολή) και παράγουν μια τιμή εξόδου. Οι διαθέσιμες συναρτήσεις φαίνονται στην επόμενη εικόνα (απόλυτη τιμή, τετραγωνική ρίζα, ημίτονο, συνημίτονο, εφαπτομένη, αντίστροφο ημίτονο, αντίστροφο συνημίτονο, αντίστροφη εφαπτομένη, λογάριθμος, δεκαδικός λογάριθμος, εκθετική συνάρτηση και δύναμη του 10)



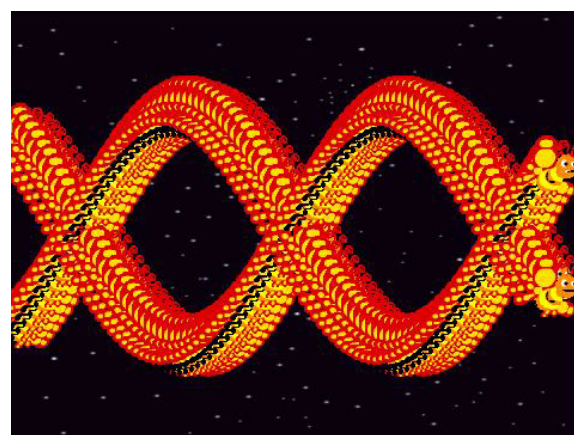
Το αποτέλεσμα της μαθηματικής συνάρτησης μπορεί να χρησιμοποιηθεί είτε ως είσοδος σε οποιαδήποτε εντολή περιέχει λευκό κουτάκι είτε για κάποιον έλεγχο σε μια συνθήκη. Μπορείτε συνεπώς να βάλετε για παράδειγμα το χαρακτήρα σας να σας λείει την τετραγωνική ρίζα του 26:



ή να κινηθεί κατά εκατό βήματα δηλώνοντάς το με έναν άλλο τρόπο.



Ας επιστρέψουμε όμως στο παράδειγμα του ημιτόνου. Θέλουμε δυο πανομοιότυπα αντικείμενα να δημιουργήσουν το σχήμα της παρακάτω εικόνας.



Συμμηθείτε ότι η θέση κάθε αντικείμενου στο Scratch προσδιορίζεται βάση του καρτεσιανού συστήματος συντεταγμένων και η διάσταση στον άξονα X παίρνει νούμερα από -240 έως 240. Τι θα συνέβαινε αν ζητούσαμε επαναλαμβανόμενα να υπολογιστεί το ημίτονο του -235, -230, -230,...0, 5, 10, 15,...210; Θα παράγονταν οι τιμές του ημιτόνου που θα θέλαμε να αναπαραστήσουμε στον άξονα Y. Το ημίτονο παίρνει τιμές από -1 έως 1. Αν όμως πολλαπλασιάσαμε με το 100 αυτές τις τιμές, δε θα δημιουργούνταν συντεταγμένες Y και για την οθόνη του Scratch;

Βάσει αυτής ακριβώς της λογικής θα δημιουργήσουμε την προηγούμενη εικόνα. Αρχικά, θα εισάγουμε 2 αντικείμενα με την ίδια μορφή από τη βιβλιοθήκη αντικειμένων του Scratch (butterfly2 από τον φάκελο Animals), και θα τα τοποθετήσουμε στις αντισυμμετρικές θέσεις (235,0) και (-235,0) αντίστοιχα.



Ο στόχος μας είναι να ρυθμίσουμε την κίνησή τους έτσι ώστε, το ένα αντικείμενο να σχεδιάζει τη συνάρτηση του ημιτόνου της θέσης του x, ενώ το άλλο να σχεδιάζει την αντισυμμετρική της προηγούμενης γραφική παράσταση. Όσον αφορά το σκη-

νικό, επιλέγουμε το stars από τον φάκελο Nature των υποβάθρων του Scratch.



Ο σχεδιασμός των γραφημάτων θέλουμε να εκτελείται για τα δύο αντικείμενα, όσο η τιμή του  $x$  είναι μικρότερη του 210, μέχρι δηλαδή το δεξί άκρο της οθόνης. Αφού η συμπεριφορά του αντικειμένου θα εκτελείται επαναληπτικά όσο η συντεταγμένη  $X$  των αντικειμένων είναι μικρότερη του 210, θα χρησιμοποιήσουμε τη δομή επανάληψης **για πάντα εάν  $X$**  με την αντίστοιχη συνθήκη. Στην παλέτα Τελεστές, το Scratch μας προσφέρει όλους τους κλασσικούς αριθμητικούς τελεστές σύγκρισης:



Το σχήμα τους είναι εξαγωνικό, γεγονός που σημαίνει ότι οι τελεστές αυτοί μπορούν να χρησιμοποιηθούν ως συνθήκες. Επιλέγουμε τον κατάλληλο τελεστή και στη συνέχεια μεταφέρουμε μέσα στα λευκά κουτάκια τα στοιχεία της σύγκρισης.

Τέλος, για την αναπαράσταση της συνάρτησης, θα αφήσουμε τη σφραγίδα των αντικειμένων σε κάθε σημείο υπολογισμού του ημιτόνου. Όπως προαναφέρθηκε θα υπολογίζουμε τη νέα θέση  $Y$  του αντικειμένου πολλαπλασιάζοντας το αποτέλεσμα της συνάρτησης του ημιτόνου του  $x$  με το 100. Πως μπορούμε να κάνουμε τον πολλαπλασιασμό; Το Scratch παρέχει τη δυνατότητα υπολογισμού όλων των τυπικών μαθηματικών πράξεων:



Παρατηρήστε ότι το σχήμα των εντολών αυτών είναι στρογγυλοποιημένο στα άκρα τους γεγονός που σημαίνει ότι μπορούν να χρησιμοποιούν σε οποιοδήποτε λευκό κουτάκι αλλά όχι ως συνθήκες, όπως είναι αυτονόητο.

Μελετήστε και τρέξτε τα σενάρια των δυο αντικειμένων:



[12\_π08.sb]



Το αποτέλεσμα είναι πραγματικά εντυπωσιακό. Θα μπορούσαμε να χρησιμοποιήσουμε την εντολή **επανάλαβε ώσπου** για την υλοποίηση της συγκεκριμένης επανάληψης; Ποια θα ήταν η κατάλληλη συνθήκη;

Συνοψίζοντας τα συμπεράσματα του παραδείγματος:

A) Όλες οι μεταβλητές κατάστασης που έχουμε δει μέχρι αυτό το κεφάλαιο μπορούν να χρησιμοποιηθούν είτε ως είσοδοι στα λευκά κουτάκια των εντολών είτε ως στοιχεία σύγκρισης στις συνθήκες.

B) στα λευκά κουτάκια μπορούμε να εισάγουμε υπολογιζόμενες τιμές από σύνθετες αριθμητικές αναπαραστάσεις.

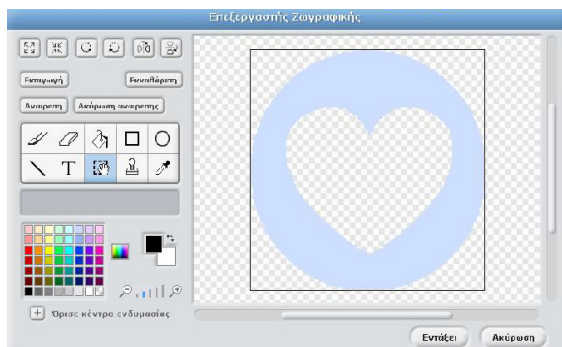
Γ) οι μαθηματικές συναρτήσεις μας επιτρέπουν να φέρουμε άλλα μαθήματα μέσα στο Scratch! Φανταστείτε και προτείνονται προβλήματα που μπορούν να σας ζητήσουν να υλοποιήσετε στο Scratch οι καθηγητές των Μαθηματικών και της Φυσικής!

**Παράδειγμα «Κρυμμένο σχήμα»:** Στο παράδειγμα που ακολουθεί θα προσκαλέσουμε τους χρήστες μας να αποκαλύψουν ένα κρυμμένο σχήμα πατώντας επαναλαμβανόμενα κλικ με το ποντίκι τους σε ένα φαινομενικά κενό σκηνικό. Τα κλικ μας θα δημιουργούν το αποτύπωμα μιας κουκίδας που ακολουθεί το δείκτη του ποντικιού μόνο όμως όταν η κουκίδα δεν βρίσκεται πάνω από το κρυμμένο αντικείμενο.

Το έργο μας θα περιλαμβάνει δύο αντικείμενα. Το πρώτο θα είναι το κρυμμένο σχήμα το οποίο θα έχει το ίδιο χρώμα με το σκηνικό του έργου μας ώστε να μη φαίνεται. Το δεύτερο θα είναι μία κουκίδα η οποία θα μετακινείται μαζί με το δείκτη του ποντικιού και θα αφήνει το αποτύπωμά της με διαφορετικό χρώμα κάθε φορά που θα πατάμε στο ποντίκι μας (εφόσον δεν βρισκόμαστε πάνω από το κρυμμένο αντικείμενο). Για να είναι λίγο δυσκολότερο το έργο μας, η κουκίδα δεν θα μετακινείται ακριβώς κάτω από το δείκτη του ποντικιού, αλλά ανάλογα με το πόσο μακριά βρίσκεται ο δείκτης του ποντικιού στον άξονα  $X$  από το κέντρο της οθόνης του Scratch, η κουκίδα θα απομακρύνεται αναλόγως από το δείκτη του ποντικιού! Τέλος, θέλουμε όταν ο χρήστης πατά το πλήκτρο  $c$  να καθαρίζει η οθόνη ώστε να μπορεί να ξεκινά ξανά την προσπάθειά του.



Το κρυφό μας σχήμα:



Στην επόμενη εικόνα βλέπετε την οθόνη του έργου μας μετά από πολλά κλικ...



Ποιες θα είναι οι συμπεριφορές των αντικειμένων μας;

**Κουκίδα:** Καταρχάς, η κουκίδα θα πρέπει να μετακινείται μαζί με το δείκτη του ποντικιού. Η συντεταγμένη Y θα είναι ίδια με τη θέση Y του δείκτη του ποντικιού. Για να καταφέρουμε να μετακινούμε την κουκίδα στον άξονα X ανάλογα με την απόστασή της από το κέντρο θα πρέπει να δημιουργήσουμε μια μαθηματική έκφραση. Για παράδειγμα,

**Νέα θέση = θέση X του δείκτη του ποντικιού + α\* θέση X αντικειμένου**

Η θέση X της κουκίδας αποτελεί την απόστασή της από το κέντρο. Άρα, αν στην τρέχουσα θέση του δείκτη του ποντικιού προσθέσουμε μια «ποσότητα» ανάλογη της θέσης X του αντικειμένου, τότε η κουκίδα θα απομακρύνεται από το δείκτη του ποντικιού όσο ο δείκτης απομακρύνεται από το κέντρο της οθόνης του Scratch. Για να δημιουργήσουμε την παραπάνω έκφραση, χρειαζόμαστε μια πρόσθεση και έναν πολλαπλασιασμό, και συνεπώς θα χρησιμοποιήσουμε τους τελεστές εισάγοντας το δεύτερο μέσα στον πρώτο.

Η κουκίδα θα πρέπει να αφήνει το αποτύπωμά της μόνο εφόσον δεν ακουμπά το κρυμμένο αντικείμενο και μόνο όταν ταυτόχρονα πατιέται το πλήκτρο του ποντικιού. Αυτή την απαίτηση μπορούμε να την ικανοποιήσουμε με την εντολή για **πάντα εαν...** και τον τελεστή **...ΚΑΙ...** ο οποίος απαιτεί και οι δυο συνθήκες να είναι αληθείς ώστε να αποτιμηθεί ως αληθής και η έκφραση. Μελετήστε τα παρακάτω 2 σενάρια που αφορούν την κουκίδα

```

όταν στο  γίνει κλικ
για πάντα
  πήγαινε στο x:  $\text{ποντίκι } x + 0.2 * \text{ποντίκι } x$  y:  $\text{ποντίκι } y$ 

```

```

όταν στο  γίνει κλικ
καθάρισε
για πάντα εάν  $\text{πατήθηκε το ποντίκι;}$  και όχι  $\text{αγγίζει το κρυμμένο σχήμα ;}$ 
  σφραγίδα

```

Η τελευταία απαίτηση του προβλήματος είναι να καθαρίζουμε την οθόνη όποτε επιθυμούμε, με το πάτημα του πλήκτρου c. Για το σκοπό αυτό αρκεί το σενάριο της επόμενης εικόνας.

```

όταν το πλήκτρο  πατηθεί
καθάρισε

```

[12\_π09.sb]



Είναι σημαντικό να διακρίνουμε ότι έχουμε δει τρία διαφορετικά είδη συνθηκών:

A) συνθήκες που αφορούν την αλληλεπίδραση ενός χαρακτήρα με το χρήστη (π.χ. αν πατήθηκε το ποντίκι ή κάποιο άλλο πλήκτρο),

B) συνθήκες που αφορούν τη σχέση του αντικειμένου με άλλα αντικείμενα και το σκηνικό (π.χ. αν ακουμπά άλλο αντικείμενο, ή αν αγγίζει άλλο χρώμα),

Γ) συνθήκες που αφορούν τη συσχέτιση μιας ιδιότητας του αντικειμένου (π.χ. θέση, ενδυμασία κτλ.) με μια μαθηματική έκφραση.

Η ποικιλία αυτή συνθηκών μας δίνει μια εξαιρετική ευελιξία για να δημιουργήσουμε ενδιαφέροντα και απρόβλεπτα παιχνίδια στη συνέχεια.

### Περίληψη

Ανακεφαλαιώνοντας, στην παρούσα ενότητα ασχοληθήκαμε και πάλι με την έννοια της επανάληψης που γνωρίσαμε αρχικά στο κεφάλαιο 10, ενώ εμβαθύνσαμε περισσότερο στις έννοιες των τελεστών και των αισθητήρων, που γνωρίσαμε στο κεφάλαιο 11. Συζητήσαμε για τρεις νέες εντολές επανάληψης, την **για πάντα εάν X**, με την οποία ένα σώμα εντολών εκτελείται για πάντα στο έργο μας όσο ικανοποιείται μια συνθήκη, την **επανάλαβε ώσπου**, στην οποία οι περιεχόμενες εντολές εκτελούνται όσο η συνθήκη είναι ψευδής και την **περίμενε ώσπου X**, η οποία διακόπτει την εκτέλεση ενός σεναρίου μέχρι να ικανοποιηθεί μια συνθήκη. Συζητήσαμε και είδαμε ότι πολλά προγραμματιστικά προβλήματα μπορούν να λυθούν με διαφορετικούς συνδυασμούς των συγκεκριμένων εντολών. Είστε έτοιμοι για να μπείτε στα πιο βαθιά; Πως επικοινωνούν τότε τα αντικείμενα μεταξύ τους;

### Ερωτήσεις

1) Διακρίνεται αν οι παρακάτω δηλώσεις είναι σωστές ή λανθασμένες:

Σ-Λ : Μετά την ολοκλήρωση της δομής **για πάντα εάν...**, μπορούμε να προσθέσουμε κι άλλες εντολές.





δι, τα αντικείμενα θα κινούνται διαρκώς, όταν ακουμπούν στα όρια του παραθύρου θα αναπηδούν και θα στρίβουν δεξιόστροφα κατά 1 μοίρα και η κίνησή τους θα επαναλαμβάνεται μέχρι να κάνουμε κλικ με το ποντίκι πάνω τους.

2) Θέλουμε να κατασκευάσουμε ένα παιχνίδι με δύο αντικείμενα, έναν λαβύρινθο και ένα ανθρωπάκι που θα κινείται μέσα σ' αυτόν. Το παιχνίδι μας θα έχει 3 διαφορετικά επίπεδα λαβυρίνθων και με την ολοκλήρωση του προηγούμενου αυτομάτως θα μεταφέρει το χρήστη στο επόμενο επίπεδο. Το φόντο σε κάθε λαβύρινθο θα είναι κίτρινο, ενώ τα εμπόδια θα έχουν διαφορετικό χρώμα. Το σημείο τερματισμού θα είναι μπλε στα 2 πρώτα επίπεδα, ενώ στο τελευταίο θα είναι μωβ. Οι λαβύρινθοι θα είναι κλιμακούμενης δυσκολίας της επιλογής σας.

Το ανθρωπάκι θα κινείται με τα αντίστοιχα βέλη πάνω, κάτω, δεξιά, αριστερά κατά 5 βήματα κάθε φορά. Όμως κάθε στιγμή το ανθρωπάκι θα κινείται κατά τουλάχιστον 3 βήματα σε οποιαδήποτε κατεύθυνση είναι στραμμένο. Όταν πέφτει σε εμπόδιο ή τοίχο θα πρέπει να επιστρέφει στο σημείο εκκίνησης του εκάστοτε λαβυρίνθου που ορίζετε.

