

Κεφ.15: Παιχνίδια

.....
Σε αυτό το κεφάλαιο:

15.1 Super Pong

15.1 Ναρκοπέδιο!

15.3 Σκοποβολή

15.4 Ο καρχαρίας

15.5 Παραλλαγή Pacman

15.6 Καροτοκυνηγός
.....



15.1 Super Pong

[15_π01.sb]

Στο παιχνίδι Super Pong που θα αναπτύξουμε, δύο μπάλες κινούνται ακατάπαυστα, αναπηδούν στα όρια της οθόνης του Scratch αλλά δεν επιτρέπεται να ακουμπήσουν στο κάτω μέρος της οθόνης. Για ακριβώς αυτό το λόγο, ο χρήστης αναλαμβάνει να αποκρούσει οποιαδήποτε μπάλα πλησιάζει προς το κάτω μέρος της οθόνης μέσω μιας ρακέτας που κινείται οριζόντια ανάλογα με τη θέση του δείκτη του ποντικιού. Το σκορ του χρήστη διαμορφώνεται από τις αποκρούσεις που κατάφερε να κάνει ενώ αν έστω και μια μπάλα φτάσει στο κάτω όριο, το παιχνίδι τερματίζεται.

Δεν πρέπει να ξεχνάμε ότι τα προβλήματα στο Scratch λύνονται πιο εύκολα αν αποκτήσουμε μια μεθοδική προσέγγιση στην ανάλυσή τους. Τα ερωτήματα που έχουμε να απαντήσουμε είναι τα ακόλουθα:

1) Ποια είναι τα διαφορετικά αντικείμενα που διακρίνουμε και ποια η γενική περιγραφή τους; Ποιο θα είναι το σκηνικό μας και πως θα συμμετάσχει στην επίλυση του προβλήματος;

2) Για κάθε αντικείμενο του προηγούμενου βήματος (αντικείμενα και σκηνικό):

- ✓ Χρειάζονται διαφορετικές **εμφανίσεις** και πολλαπλοί ήχοι για το αντικείμενο;
- ✓ Ποιες είναι οι **συμπεριφορές** του αντικειμένου (τι θα πρέπει να κάνει);
- ✓ Σε ποια **γεγονότα αντιδρά** το αντικείμενο; (γεγονότα και αισθητήρες)
- ✓ Χρειάζεται να **επικοινωνήσει** το συγκεκριμένο αντικείμενο με άλλα;
- ✓ Χρειάζεται το αντικείμενο να αποθηκεύει πληροφορίες σε **μεταβλητές ή λίστες**;
- ✓ Χρειαζόμαστε **τυχαιότητα** στη συμπεριφορά του αντικειμένου; Σε ποια σημεία;

Αντικείμενα & Σκηνικό

Από την περιγραφή του προβλήματος, είναι σαφές ότι τα αντικείμενά μας θα είναι οι δυο μπάλες που κινούνται αυτόνομα και η ρακέτα η οποία ελέγχεται από το χρήστη του παιχνιδιού. Το παιχνίδι μας θα περιλαμβάνει ως σκηνικό ένα τοπίο που θα δυσκολεύει οπτικά τη διάκριση κάθε μπάλας.

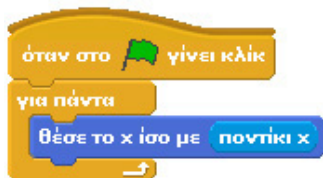


Αντικείμενο 1: Σκηνικό

Το σκηνικό μας δεν θα συμμετάσχει ενεργά στο έργο μας αφού δεν αλλάζει, ούτε εκτελεί κάποιες γενικές λειτουργίες (αν και θα μπορούσαμε να βάλουμε ένα τραγούδι στο παιχνίδι μας...). Έχουμε όμως μια απαίτηση από αυτό! Για να διευκολύνουμε τη δημιουργία σεναρίων για τον τερματισμό του παιχνιδιού, θα δημιουργήσουμε μια κόκκινη περιοχή στο κάτω μέρος του σκηνικού έτσι ώστε όταν οι μπάλες ακουμπούν αυτή τη γραμμή, το παιχνίδι να τερματίζεται. Το σκηνικό που φαίνεται στην προηγούμενη εικόνα έχει επιλεγεί από το φάκελο nature της βιβλιοθήκης υποβάθρων και έχει όνομα lake.

Αντικείμενο 2: Η ρακέτα

Θα σχεδιάσουμε τη ρακέτα στον επεξεργαστή ζωγραφικής τραβώντας μια μικρή οριζόντια μαύρη γραμμή. Θα τοποθετήσουμε τη ρακέτα κοντά στο κάτω όριο της οθόνης του Scratch. Θέλουμε ο παίχτης του παιχνιδιού να χειρίζεται τη ρακέτα, μέσω του δείκτη του ποντικιού του. Επομένως μεταφέροντας το ποντίκι προς τα δεξιά, θα πρέπει να μεταφέρεται και η ρακέτα προς τα δεξιά και, συνεπώς, πρέπει να συνδέσουμε την οριζόντια θέση της ρακέτας με την οριζόντια θέση του δείκτη του ποντικιού. Μελετήστε το παρακάτω σενάριο:



Με τη συγκεκριμένη εντολή (θέσε το x ίσο με...) και τη χρήση της μεταβλητής ποντίκι x, καταφέρνουμε η ρακέτα να ακολουθεί οριζόντια το δείκτη του ποντικιού.

Αντικείμενο 3: Οι μπάλες

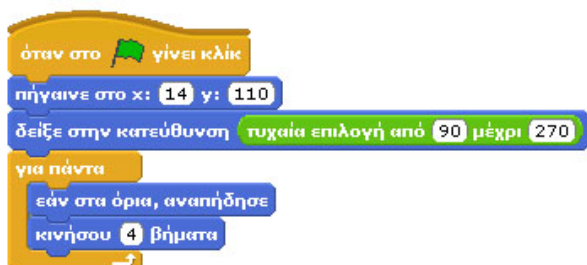
Ας συνοψίζουμε τη συμπεριφορά που θέλουμε να έχουν οι δύο μπάλες:

- ✓ θέλουμε να κινούνται συνεχώς και να αναπηδούν όταν ακουμπούν τα όρια της οθόνης,
- ✓ θέλουμε να τερματίζεται το παιχνίδι, αν οι μπάλες ακουμπήσουν την κόκκινη περιοχή,
- ✓ θέλουμε να αναπηδούν όταν ακουμπούν την ρακέτα. Κάθε φορά που αναπηδούν λόγω της ρακέτας το σκορ του χρήστη θα πρέπει να αυξάνεται.

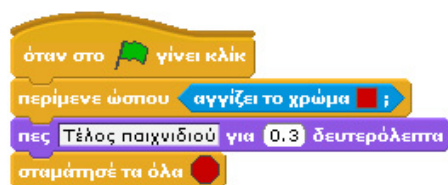
Ας ονομάσουμε τις δυο μπάλες ως Μπάλα1 και Μπάλα2. Εισάγουμε τις αντίστοιχες μορφές από το φάκελο things της βιβλιοθήκης αντικειμένων του Scratch. Τα σενάρια για τις δυο μπάλες είναι πανομοιότυπα και επομένως θα αναλύσουμε ένα από τα δυο.

Διακρίναμε παραπάνω τρεις διαφορετικές συμπεριφορές. Μέχρι τώρα προσπαθούσαμε να εντάξουμε τις διαφορετικές συμπεριφορές ενός αντικείμενου μέσα σε ένα και μόνο σενάριο. Μήπως όμως υπάρχει και άλλη λύση; Τι θα λέγατε αν, για αλλαγή, δημιουργούσαμε τρία διαφορετικά σενάρια που αφορούν τις τρεις διαφορετικές συμπεριφορές και τα οποία τρέχουν ταυτόχρονα όταν πατηθεί η πράσινη σημαία;

Το πρώτο σενάριο που αφορά τη συνεχή κίνηση της μπάλας και την αναπήδηση όταν ακουμπά τα όρια της οθόνης, είναι πολύ απλό. Πρέπει να προσδιορίσουμε την αρχική θέση της μπάλας, να της δώσουμε μια τυχαία κατεύθυνση προς τα κάτω (δηλαδή από 90 έως 270 μοίρες) και στη συνέχεια να εξασφαλίσουμε ότι η μπάλα θα κινείται για πάντα:



Το δεύτερο σενάριο αφορά τον τερματισμό του παιχνιδιού. Χρειαζόμαστε ένα σενάριο ώστε όταν η μπάλα αγγίξει την κόκκινη περιοχή να βγαίνει ένα σύντομο μήνυμα και να σταματούν όλα τα σενάρια:



Το τρίτο σενάριο αφορά την συμπεριφορά της μπάλας όταν έρχεστε σε επαφή με την ρακέτα. Καταρχάς, θα χρειαστούμε μια νέα μεταβλητή που θα μπορεί να χρησιμοποιηθεί και από τις δυο μπάλες και στην οποία θα αποθηκεύουμε το σκορ του χρήστη. Έστω ότι ονομάζουμε τη μεταβλητή αυτή σκορ. Κατά την έναρξη του προγράμματος θα πρέπει να μηδενίζουμε την τιμή της σε μια από τις δυο μπάλες. Από και και πέρα, όποτε η μπάλα αγγίζει τη ρακέτα (οι αισθητήρες θα μας βοηθήσουν να αναγνωρίσουμε το γεγονός αυτό), θα πρέπει να αυξάνεται το σκορ και η μπάλα να αναπηδά προς τα πάνω. Για να αναπηδά προς τα πάνω, θα χρησιμοποιήσουμε μια μαθηματική έκφραση για να αφαιρέσουμε την τρέχουσα κατεύθυνση από το 180 ώστε να δώσουμε συμπληρωματική κατεύθυνση στη μπάλα. Για να μην είναι όμως μονότονο το παιχνίδι, θα αλλάζουμε ελαφρώς την κατεύθυνση της μπάλας με τυχαίο τρόπο.



Υπάρχουν δυο ενδιαφέροντα σημεία στο συγκεκριμένο σενάριο για τους παρατηρητικούς:

- ✓ Έχουμε εισάγει μια εντολή κίνησης όταν το αντικείμενο αγγίζει τη ρακέτα. Γιατί δεν αφήνουμε στο πρώτο από τα τρία σενάρια την κίνηση; Και γιατί το κινούμε με μεγαλύτερο βήμα από το «κανονικό»; Η κίνηση με βήματα έχει ως αποτέλεσμα να είναι πιθανόν η επαφή της μπάλας με τη ρακέτα να γίνει σε διαφορετικό βάθος (από 1-4 βήματα), δηλαδή η μπάλα να «ακουμπάει» λιγότερο ή περισσότερο τη ρακέτα. Αυτό μπορεί να γίνει ακόμη πιο έντονο αν η ρακέτα αγγίξει την μπάλα από τις πλευρές της και όχι από πάνω. Σε όλες αυτές τις περιπτώσεις θα πρέπει να εξασφαλίσουμε ότι μπάλα θα φύγει μακριά από την ρακέτα μέσα στην ίδια επανάληψη ώστε να μην ξαναγίνει αληθής ο έλεγχος «εάν αγγίξει τη ρακέτα». Δοκιμάστε να αφαιρέσετε τις συγκεκριμένες γραμμές και συζητήστε τι συμβαίνει.
- ✓ γιατί αρχικά αλλάζουμε κατεύθυνση, στη συνέχεια κινούμε λίγο το αντικείμενο και μετά ξανα-αλλάζουμε κατεύθυνση; Μήπως είναι λίγο υπερβολικό αυτό; Πως θα μπορούσε να γίνει το σενάριο πιο συμπαγές και αποδοτικό;

Προτεινόμενες αλλαγές στο παιχνίδι

1) Δοκιμάστε να δημιουργήσετε ξανά το παιχνίδι χρησιμοποιώντας μόνο ένα και όχι τρία σενάρια.

2) Προσπαθήστε να δημιουργήσετε ένα τρόπο αξιολόγησης της απόδοσης του παιχνιδιού για τις δυο εναλλακτικές υλοποίησή του, χρησιμοποιώντας έναν μετρητή χρόνου και χρονομετρώντας συγκεκριμένες ενέργειες του παιχνιδιού. Ποιος από τους δυο τρόπους υλοποίησης του παιχνιδιού είναι γρηγορότερος; Αιτιολογήστε την απάντησή σας.



15.2 Ναρκοπέδιο!

[15_π02.sb]

Όλοι μας έχουμε παίξει παιχνίδια όπου το χαρακτηριστικό που θα κρίνει το νικητή είναι το ποιος θα καταφέρει να πετύχει ένα συγκεκριμένο στόχο στο μικρότερο δυνατό χρόνο. Και στο Scratch μπορούμε να δημιουργήσουμε ενδιαφέροντα παιχνίδια έχοντας αντίπαλο το χρόνο.

Για να το πετύχουμε αυτό χρειαζόμαστε το **χρονόμετρο**, μια μεταβλητή κατάσταση η οποία βρίσκεται στην παλέτα **Αισθητήρες**. Για να εμφανιστεί το χρονόμετρο πρέπει να επιλέξουμε την προβολή του χρονομέτρου στη σκηνή τσεκάροντας την αντίστοιχη επιλογή.



Το **χρονόμετρο** ξεκινά να μετράει δευτερόλεπτα αυτόματα από τη στιγμή που εκκινούμε το Scratch! Αν τσεκάρουμε την αντίστοιχη επιλογή για να εμφανιστεί στην οθόνη, τότε θα παρατηρήσουμε ότι α) ο χρόνος κυλάει παρότι δεν εκτελούμε κάποιο έργο (!) β) η αρχική του τιμή δεν είναι 0 αλλά αντιστοιχεί στα δευτερόλεπτα που πέρασαν από τη στιγμή που ανοίξαμε το Scratch(!). Επειδή, ο χρόνος αυτός δεν μας ενδιαφέρει στα έργα μας, το Scratch μας παρέχει μια εντολή για να μηδενίζουμε την τιμή του, την εντολή **μηδένισε το χρονόμετρο**.

Αν λοιπόν θέλουμε να χρησιμοποιήσουμε το χρονόμετρο σε κάποιο πρόγραμμα, θα πρέπει απλά στην αρχή του έργου να το μηδενίσουμε.

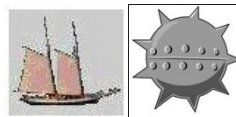


Πάμε τώρα στο νέο παιχνίδι που θα δημιουργήσουμε. Έστω ότι είμαστε καπετάνιοι σε ένα πλοίο που έχει μπλεχτεί σε ένα θαλάσσιο ναρκοπέδιο και προσπαθεί να φτάσει στη στεριά. Όποιος καπετάνιος κάνει το καλύτερο χρόνο, αυτός είναι και ο νικητής. Φυσικά χρειάζεται πολύ προσοχή στις νάρκες οι οποίες είναι έτοιμες να εκραγούν! Οι νάρκες δε μένουν σταθερές στη θέση τους αλλά κινούνται λόγω θαλασσιών ρευμάτων κάνοντας πιο δύσκολη την προσπάθεια του πλοίου να φτάσει στη στεριά. Αν το πλοίο ακουμπήσει σε μία από αυτές τις νάρκες, τότε αυτή θα εκραγεί και το πλοίο θα πάρει φωτιά και φυσικά ο παίκτης θα χάσει.



Αντικείμενα και Σκηνικό

Σίγουρα θα χρειαστούμε ένα αντικείμενο πλοίο και ένα αντικείμενο-νάρκη το οποίο θα αντιγράψουμε πολλές φορές ώστε να γεμίσουμε το ναρκοπέδιό μας. Ψάχνοντας στο διαδίκτυο αλλά και στη βιβλιοθήκη του Scratch επιλέξαμε τις εξής ενδυμασίες για τα αντικείμενα:



Τι σκηνικό θα θέλαμε να χρησιμοποιήσουμε για μια τέτοιου είδους εφαρμογή; Σκεφτόμαστε ένα φόντο θάλασσας με τη στεριά που να φαίνεται σε κάποια γωνία. Ενδεικτικά λοιπόν το σκηνικό θα μπορούσε να ήταν αυτό της επόμενης εικόνας.



Αντικείμενα νάρκες

Τώρα ας σκεφτούμε τα εμπόδια που θα υπάρχουν στο σκηνικό μας. Καλό θα ήταν να υπάρχουν αρκετές νάρκες έτσι ώστε να κάνουμε τη διαδρομή του πλοίου δυσκολότερη. Έστω, λοιπόν, ότι εισάγουμε πέντε νάρκες, οι οποίες θα επιπλέουν στη θάλασσά μας και θα είναι διασκορπισμένες σε όλη την επιφάνεια του νερού. Ποια θα είναι η συμπεριφορά τους; Απλά θα κινούνται ανάμεσα σε σταθερές θέσεις. Καλό θα ήταν κάποιες από αυτές να κινούνται κάθετα, άλλες οριζόντια και άλλες διαγώνια. Το ολοκληρωμένο σενάριο για μια από τις νάρκες:



Ορίζουμε μια αρχική θέση, και με την εντολή επανάληψης **για πάντα**, προκαλούμε τις νάρκες να μετακινούνται ομαλά μεταξύ δυο θέσεων. Ανάλογα με το χρόνο κίνησης που θα επιλέξετε στις εντολές **κινήσου ομαλά...δευτ**, θα προσδιορίσετε και τη δυσκολία του παιχνιδιού. Δημιουργήστε ένα μόνο αντικείμενο και στη συνέχεια πατήστε **δεξί κλικ** πάνω του (στη λίστα αντικειμένων) και επιλέξτε **διπλασίασε**. Εύκολα μπορείτε να δημιουργήσετε άλλα τέσσερα αντικείμενα που έχουν ενσωματωμένο μέσα τους το ίδιο σενάριο. Απλά αλλάξτε τις συντεταγμένες στις αντίστοιχες εντολές.

Αντικείμενο πλοίο

Το πλοίο μας χρειάζεται μια δεύτερη ενδυμασία (έκρηξη) έτσι ώστε σε περίπτωση που ακουμπήσει μια νάρκη να φανεί πιο ρεαλιστικά το αποτέλεσμα. Η ενδυμασία αυτή θα μπορούσε να μοιάζει με αυτήν που εμφανίζεται στην παρακάτω εικόνα.



Η συμπεριφορά του καραβιού είναι εύκολη. Ας διακρίνουμε τις διαστάσεις της:

- ✓ το πλοίο πρέπει να κινείται με τα βελάκια του πληκτρολογίου ώστε να μπορεί να μετακινηθεί μέχρι τη στεριά,
- ✓ αν το πλοίο καταφέρει να «αγγίξει» τη στεριά τότε το παιχνίδι μας θα έχει τερματιστεί με επιτυχία. Η άφιξη του πλοίου στη στεριά μπορεί να γίνει αντιληπτή μόλις το πλοίο ακουμπήσει το νησάκι που βρίσκεται στην κάτω αριστερή γωνία του σκηνικού.
- ✓ αν το πλοίο «αγγίξει» μια νάρκη, τότε φοράει την ενδυμασία «έκρηξη» και τερματίζεται το παιχνίδι.

Στην παρουσίαση του προηγούμενου παιχνιδιού, εισάγαμε τις τρεις διαφορετικές συμπεριφορές του αντικειμένου «ρακέτα» σε τρία διαφορετικά σενάρια. Το ίδιο θα μπορούσε να γίνει και σε αυτήν την περίπτωση οπότε θα είχαμε: 4 μικρά σενάρια για την κίνηση του πλοίου, ένα σενάριο που θα αφορούσε το τι θα συμβεί όταν το πλοίο αγγίζει τη νάρκη και ένα σενάριο που θα αφορούσε το τι θα συμβεί όταν το πλοίο αγγίζει τη στεριά. Θα προσπαθήσουμε όμως να μειώσουμε τον αριθμό των σεναρίων.

Όταν αρχίζει το πρόγραμμα πρέπει να φροντίσουμε το πλοίο να έχει την κατάλληλη ενδυμασία. Φυσικά, όπως και με τις νάρκες, τοποθετούμε το πλοίο στην αρχική του θέση και δεν ξεχνάμε να μηδενίσουμε το χρονόμετρο για ξεκινήσει η προσπάθεια του χρήστη. Το σενάριό μας προς το παρόν μοιάζει με το παρακάτω:

```
όταν στο [πλοίο] γίνει κλικ
αλλαγή σε ενδυμασία πλοίο
θέσε το x ίσο με 200
θέσε το y ίσο με 160
μηδένισε το χρονόμετρο
```

Χρειαζόμαστε να προσθέσουμε τους δυο ελέγχους για τις δυο διαφορετικές συνθήκες τερματισμού που περιγράφηκαν προηγουμένως. Προφανώς θα τις εντάξουμε μέσα σε μια εντολή **για πάντα** αφού θέλουμε να οι έλεγχοι να πραγματοποιούνται διαρκώς. Οι συνθήκες μας θα βασιστούν στο χρώμα των αντίστοιχων αντικειμένων:

- ✓ στην περίπτωση που το πλοίο μας αγγίζει το χρώμα της νάρκης θα προσδιορίζουμε την ενδυμασία του πλοίου σε «έκρηξη» και θα σταματά το σενάριο

- ✓ στην περίπτωση που αγγίζει το χρώμα της στεριάς, τότε το πλοίο θα μας ανακοινώσει για 2 δευτερόλεπτα το χρώμα μας και θα τερματίσει το έργο.

Το αρχικό σενάριο θα γίνει:

```
όταν στο [πλοίο] γίνει κλικ
αλλαγή σε ενδυμασία πλοίο
θέσε το x ίσο με 200
θέσε το y ίσο με 160
μηδένισε το χρονόμετρο
για πάντα
  εάν αγγίζει το χρώμα [νάρκη]
  αλλαγή σε ενδυμασία έκρηξη
  σταμάτησέ τα όλα
  εάν αγγίζει το χρώμα [στεριά]
  πες χρονόμετρο για 2 δευτερόλεπτα
  σταμάτησέ τα όλα
```

Αφού ορίσαμε τα αρχικά δεδομένα του πλοίου μας, και τις συνθήκες τερματισμού του παιχνιδιού, θέλουμε να το κινούμε με τα βελάκια του πληκτρολογίου. Τα γνωστά τέσσερα σενάρια:

```
όταν το πλήκτρο πάνω βέλος πατηθεί
  άλλαξε y κατά 2
όταν το πλήκτρο κάτω βέλος πατηθεί
  άλλαξε y κατά -2
όταν το πλήκτρο αριστερό βέλος πατηθεί
  άλλαξε x κατά -2
όταν το πλήκτρο δεξί βέλος πατηθεί
  άλλαξε x κατά 2
```

Και το παιχνίδι μας είναι έτοιμο. Πειραματιστείτε με τον αριθμό και τις θέσεις των ναρκών για να το κάνετε πιο ενδιαφέρον.

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Αλλάξτε τα σενάρια ώστε να αλλάζει ενδυμασία και η νάρκη κατά τη στιγμή της έκρηξης.
- 2) Μετασχηματίστε το παιχνίδι ώστε ο χρόνος να αποθηκεύεται σε μια μεταβλητή και στη συνέχεια μετατρέψτε το παιχνίδι ώστε να παίζεται στη σειρά από δυο παίκτες. Το παιχνίδι θα πρέπει να ανακοινώνει ποιος από τους δυο παίκτες ήταν ο τελικός νικητής.
- 3) Προσδιορίστε το χρόνο στις εντολές κίνησης των ναρκών (**κινήσου ομαλά...δευτ.**) με μια μεταβλητή και στη συνέχεια να εισάγετε διαφορετικά επίπεδα δυσκολίας στο παιχνίδι αλλάζοντας την τιμή αυτής της μεταβλητής. Ζητήστε από το χρήστη να επιλέξει βαθμό δυσκολίας στην αρχή του παιχνιδιού.

15.3 Σκοποβολή



[15_π03.sb]

Ας δούμε πως θα μπορέσετε να φτιάξετε ένα παιχνίδι σκοποβολής, δηλαδή ένα παιχνίδι αντανάκλαστικών και ακρίβειας! Στο παιχνίδι θα υπάρχουν στόχοι οι οποίοι θα κινούνται από το ένα άκρο της οθόνης μέχρι το άλλο με διαφορετικές ταχύτητες και διαφορετικές κατευθύνσεις κάθε φορά και ο χρήστης θα προσπαθεί να τους πετύχει κάνοντας κλικ πάνω τους. Αν ευστοχήσει στο κλικ του, ο στόχος εκρήγνυται, ακούγεται σχετικός ήχος και το σκορ αυξάνει. Σε διαφορετική περίπτωση απλά ακούγεται ήχος αστοχίας! Επιπλέον, πριν ξεκινήσει το παιχνίδι, ο παίκτης θα πρέπει να καθορίσει το βαθμό δυσκολίας του. Μεγαλύτερη δυσκολία θα πρέπει να ισοδυναμεί με γρηγορότερη κίνηση των στόχων. Η διάρκεια του παιχνιδιού περιορίζεται σε 60" με το χρονόμετρο. Δείτε ένα στιγμιότυπο του παιχνιδιού στην παρακάτω εικόνα:



Αντικείμενα και Σκηνικό

Πρώτη μας δουλειά για να σχεδιάσουμε το παιχνίδι μας είναι να διακρίνουμε τα κατάλληλα αντικείμενα που θα χρειαστούμε. Είναι σαφές ότι θα έχουμε πολλά πανομοιότυπα αντικείμενα-στόχους, δηλαδή πολλές μπάλες. Στο συγκεκριμένο όμως παράδειγμα δεν διαφαίνεται να έχουμε κάποιο αντικείμενο-πρωταγωνιστή που να επιδεικνύει συγκεκριμένες συμπεριφορές. Ο χρήστης είτε κάνει κλικ στις μπάλες είτε στο σκηνικό. Μπορούμε επομένως να χρησιμοποιήσουμε το σκηνικό μας ως βασικό μας αντικείμενο για τις εντολές έναρξης του έργου.

Σκηνικό

Το σκηνικό μας θα πρέπει να έχει δύο υπόβαθρα. Το πρώτο αφορά την αρχική οθόνη υποδοχής του παίκτη, στην οποία θα ορίζεται ο βαθμός δυσκολίας του παιχνιδιού. Η αρχική οθόνη θα καλωσορίζει τον παίκτη και θα του ζητάει να εισάγει ένα γράμμα e (easy), m (medium), h (hard), για να ορίσει το βαθμό δυσκολίας του παιχνιδιού. Μπορούμε εύκολα να ζωγραφίσουμε το συγκεκριμένο υπόβαθρο στη ζωγραφική. Το αποτέλεσμα θα πρέπει να είναι κάπως έτσι:

Το δεύτερο υπόβαθρο θα αφορά την οθόνη στην οποία θα εμφανίζονται οι στόχοι και η οποία θα είναι μονόχρωμη για να είναι ευδιάκριτοι οι στόχοι.

Καλωσορίσατε

Επιλέξτε δυσκολία παιχνιδιού:

Εύκολο: e

Μέτριο: m

Δύσκολο: h

Χρειάζεται επίσης να ενσωματώσουμε στο σκηνικό μας έναν ήχο που θα ακούγεται όταν ο παίκτης κάνει κλικ πάνω στο υπόβαθρο, δηλαδή όταν αποτύχει να χτυπήσει κάποιο στόχο. Θα πρέπει να είναι ένας σύντομος ήχος ώστε να μπορεί να γίνει αντιληπτός όταν ο χρήστης επαναλαμβανόμενα αστοχήσει. Στο παράδειγμά μας θα χρησιμοποιήσουμε τον ήχο WaterDrop από το φάκελο effects της βιβλιοθήκης ήχων του Scratch.

Το σκηνικό μας χρειάζεται δυο σενάρια, ένα για την περίπτωση που ο χρήστης κάνει κλικ πάνω στο σκηνικό και ένα για την παρουσίαση της οθόνης εισαγωγής αλλά και τη διαχείριση του χρόνου και του σκορ.

Το πρώτο σενάριο που αφορά την αναπαραγωγή ενός ήχου όταν ο χρήστης κάνει κλικ στο υπόβαθρο και όχι στους στόχους:



Το δεύτερο σενάριο θα πρέπει να έχει την εξής δομή:

- ✓ επιλογή επιπέδου δυσκολίας από το χρήστη,
- ✓ εκκίνηση παιχνιδιού,
- ✓ τερματισμός παιχνιδιού.

Πριν ξεκινήσουμε να βλέπουμε το σενάριο, ας σκεφτούμε τι μεταβλητές χρειαζόμαστε καθώς έχουμε να διαχειριστούμε το σκορ, τη δυσκολία και το χρονόμετρο. Επιλέγουμε το χρονόμετρο από την παλέτα **Αισθητήρες** ενώ για το σκορ δημιουργούμε μια μεταβλητή που πρέπει να είναι ορατή από όλα τα αντικείμενα. Πως όμως θα ελέγξουμε τη δυσκολία του παιχνιδιού; Όπως είδαμε και από προηγούμενα παραδείγματα, η δυσκολία ουσιαστικά αντιστοιχεί στην ταχύτητα κίνησης των στόχων, δηλαδή στο χρόνο που προσδιορίζουμε για να μετακινηθούν σε ένα σημείο της οθόνης. Όσο μικρότερος ο χρόνος, τόσο πιο γρήγορα κινούνται οι στόχοι. Πως θα ελέγξουμε τους χρόνους όλων των αντικειμένων; Χρησιμοποιώντας μια μεταβλητή, που θα ονομάσουμε «χρόνος» και η οποία θα εισαχθεί σε όλες τις εντολές **κινήσου ομαλά...δευτερόλεπτα στο x...y...**

Άρα στο δεύτερο σενάριο του σκηνικού, αρχικά θα επιλέξουμε να εμφανιστεί το υπόβαθρο με τις οδηγίες και στη συνέχεια με τη χρήση της εντολής **ρώτησε...και περίμενε** θα ζητήσουμε από το χρήστη να επιλέξει το επίπεδο δυσκολίας προσδιορίζοντας το αντίστοιχο επίπεδο με ένα γράμμα. Ανάλογα με την τιμή της μεταβλητής **απάντηση** θα μπορέσουμε στη συνέχεια

να προσδιορίσουμε την τιμή της μεταβλητής «χρόνος». Υπάρχει όμως ένα πρόβλημα. Τι θα συμβεί αν ο χρήστης δώσει ως είσοδο ένα διαφορετικό χαρακτήρα; Πως θα γνωρίζουμε το επιθυμητό επίπεδο δυσκολίας; Το έργο μας δεν πρέπει να προχωρά αν ο χρήστης δεν εισάγει ένα από τα τρία γράμματα. Αυτό μπορούμε να το επιτύχουμε αν δημιουργήσουμε μια επανάληψη η οποία σταματά μόνο εφόσον το γράμμα εισόδου είναι ένα από τα επιθυμητά. Μελετήστε την επόμενη εντολή:

```

επανάλαβε ώσπου <επάντηση = h ή απάντηση = m ή απάντηση = e>
  ρώτησε <δυσκολία παιχνιδιού: και περιμένε
  
```

Η συγκεκριμένη τεχνική χρησιμοποιείται ευρέως. Δηλαδή αν θέλετε η είσοδος του χρήστη να περιοριστεί σε ορισμένες μόνο επιλογές, δεν έχετε παρά να συνδυάσετε μια εντολή επανάληψης με μια εντολή ερωτήματος που συνεχίζει να ζητάει από το χρήστη μια τιμή έως ότου αυτός να δώσει μια από τις επιθυμητές.

Για να εξετάσουμε τη συγκεκριμένη συνθήκη χρειαστήκαμε 2 συνθήκες ...ή... αφού χρειαζόμαστε τον έλεγχο τριών ισοτήτων.

Τώρα, αφού έχουμε καταφέρει να πάρουμε μια ικανοποιητική απάντηση, θα χρησιμοποιήσουμε τρεις φορές την εντολή **εάν...** για να ορίσουμε την κατάλληλη τιμή της μεταβλητής «χρόνος».

```

εάν <απάντηση = h>
  όρισε το <χρόνος> σε 1.5
εάν <απάντηση = m>
  όρισε το <χρόνος> σε 2
εάν <απάντηση = e>
  όρισε το <χρόνος> σε 2.5
  
```

Σε αυτή τη φάση του προγράμματος συνεπώς ο χρήστης έχει επιλέξει επίπεδο και είναι έτοιμος να ξεκινήσει το παιχνίδι. Άρα, πρέπει να ξεκινήσουμε το χρονόμετρο, να μηδενίσουμε το σκορ και να εμφανίσουμε το μονόχρωμο υπόβαθρο, πάνω στο οποίο θα εμφανιστούν οι στόχοι.

```

όρισε το <Σκορ> σε 0
μηδένισε το <χρονόμετρο>
αλλαγή σε υπόβαθρο <οthoni παιχνιδιού>
  
```

Και μας μένει μια ακόμη μόνο δουλειά: να εμφανιστούν και να ξεκινήσουν την κίνησή τους οι στόχοι και να τερματιστεί το παιχνίδι μετά από 60 δευτερόλεπτα. Παρατήστε ότι τα αντικείμενα στόχοι δεν πρέπει να κάνουν τίποτε μέχρι να φτάσουμε σε αυτό το σημείο και πρέπει να σταματήσουν όταν ο χρόνος ξεπεράσει τα 60 δευτερόλεπτα. Πως θα μπορούσαμε να πετύχουμε κάτι τέτοιο στα σενάριά τους; Ο πιο απλός τρόπος είναι να τους στείλει το σκηνικό ένα μήνυμα τη στιγμή που ξεκινά το παιχνίδι και ένα δεύτερο μήνυμα τη στιγμή που τελειώνει, δηλαδή μετά από 60 δευτερόλεπτα. Οι στόχοι δε θα έχουν παρά

να χειριστούν αυτά τα δυο μηνύματα. Δείτε το τελευταίο τμήμα του δεύτερου σεναρίου για το σκηνικό:

```

μετάδωσε start
περίμενε ώσπου <60 < χρονόμετρο>
μετάδωσε end
σταμάτησε τα όλα
  
```

Όπως βλέπετε, η εντολή **περίμενε ώσπου...** με συνθήκη το χρονόμετρο να μη ξεπεράσει τα 60 δευτερόλεπτα, μπορεί να χρησιμοποιηθεί ως μηχανισμός για να σταλεί το δεύτερο σήμα «end» την κατάλληλη στιγμή. Ολοκληρωμένο πλέον το σενάριο του σκηνικού:

```

όταν στο <σκηνικό> γίνει κλικ
  αλλαγή σε υπόβαθρο <arxiki othoni>
  επανάλαβε ώσπου <απάντηση = h ή απάντηση = m ή απάντηση = e>
    ρώτησε <δυσκολία παιχνιδιού: και περιμένε
  εάν <απάντηση = h>
    όρισε το <χρόνος> σε 1.5
  εάν <απάντηση = m>
    όρισε το <χρόνος> σε 2
  εάν <απάντηση = e>
    όρισε το <χρόνος> σε 2.5
  όρισε το <Σκορ> σε 0
  μηδένισε το <χρονόμετρο>
  αλλαγή σε υπόβαθρο <οthoni παιχνιδιού>
  μετάδωσε start
  περίμενε ώσπου <60 < χρονόμετρο>
  μετάδωσε end
  σταμάτησε τα όλα
  
```

Αντικείμενα-στόχοι

Οι στόχοι είναι τα αντικείμενα πάνω στα οποία θα επιδιώκει ο χρήστης να κάνει κλικ. Στο παράδειγμά μας μπορούμε να εισάγουμε μια σειρά από διαφορετικές μπάλες που βρίσκονται μέσα στο φάκελο **things** της βιβλιοθήκης αντικειμένων του Scratch! Θα χρειαστείτε αρκετές μπάλες γι' αυτό μη ξεχνάτε ότι μπορείτε να διπλασιάστε τα αντικείμενά σας (δεξιά κλικ πάνω στο αντικείμενο και επιλέγουμε «διπλασίασε»!)



Οι στόχοι θα πρέπει να έχουν δυο ενδυμασίες: την αρχική ενδυμασία και την ενδυμασία έκρηξης «mroum» την οποία θα δημιουργήσετε στην ζωγραφική! Δεν είναι δύσκολο! Το αποτέλεσμα θα πρέπει να είναι κάτι τέτοιο:



Ποιες είναι οι συμπεριφορές των συγκεκριμένων αντικειμένων;

- ✓ όταν λάβουν το μήνυμα «start» θα πρέπει να ξεκινά η αδιάκοπη και τυχαία κίνησή τους,
- ✓ όταν ο χρήστης πατήσει κλικ πάνω στους στόχους, οι στόχοι θα πρέπει να αλλάζουν εμφάνιση,
- ✓ όταν ξεκινά το έργο θέλουμε τα αντικείμενα-στόχοι να μην φαίνονται,
- ✓ όταν τελειώνει το έργο θέλουμε τα αντικείμενα-στόχοι να εξαφανίζονται και να σταματά η κίνησή τους.

Ελπίζουμε να έχει γίνει φανερό πλέον, ότι αν πριν ξεκινήσουμε να δημιουργούμε το σενάριο ενός αντικειμένου, έχουν αναλύσει προσεκτικά τις συμπεριφορές του, η κατασκευή του διευκολύνεται κατά πολύ.

Επειδή όλες οι συμπεριφορές βασίζονται σε γεγονότα που προκαλούνται είτε από το χρήστη (κλικ), είτε από άλλα αντικείμενα (μηνύματα start, end), είτε από το έργο (πάτημα στην πράσινη σημαία), θα χρειαστεί να δημιουργήσουμε 4 αντίστοιχα σενάρια. Τα δυο τελευταία είναι εύκολα και δεν χρειάζονται επεξήγηση:



Όταν ο παίκτης κάνει κλικ πάνω στο στόχο, δηλαδή πετυχαίνει μια βολή, τότε θα πρέπει

- ✓ να αλλάζει ενδυμασία και να φοράει την ενδυμασία «μπουμ»,
- ✓ να παίζει τον αντίστοιχο ήχο (εισάγουμε τον ήχο Pop από τον φάκελο effects των έτοιμων ήχων στους στόχους μας),
- ✓ να αυξάνει το σκορ,
- ✓ να κρύβεται.

Αυτά ακριβώς εκτελεί το παρακάτω σενάριο



Η εντολή **περίμενε...δευτερόλεπτα**, επιτρέπει στο χρήστη να δει έστω και για λίγο την εμφάνιση έκρηξης του στόχου.

Τέλος, το σενάριο της κίνησης πρέπει να ξεκινά όταν το σκηνικό μεταδώσει το σήμα «start». Γι' αυτό, το σενάριο θα ξεκινάει με την εντολή **όταν λάβω....** Το αντικείμενο θα ξεκινάει πάντα από το αριστερό ή το δεξί άκρο του σκηνικού και θα καταλήγει στο απέναντι. Επίσης, για λόγους συμμετρίας, τα μισά αντικεί-

μενα θα ξεκινούν από τα αριστερά και τα άλλα μισά από τα δεξιά. Για να είναι τυχαία η κίνησή τους θα πρέπει να αλλάζουμε διαρκώς το ύψος, τη συντεταγμένη y δηλαδή, από την οποία ξεκινούν και τελειώνουν κάθε φορά που διανύουν την οθόνη του Scratch.

Επιπλέον για να ενισχύσουμε το τυχαίο χαρακτήρα της εμφάνισης των στόχων, μπορούμε να προσθέσουμε ένα τυχαίο χρονικό διάστημα πριν εμφανιστεί κάθε αντικείμενο, με τη χρήση της εντολής **περίμενε... δευτερόλεπτα**. Και φυσικά δεν πρέπει να ξεχάσουμε ότι στην εντολή **κινήσου ομαλά... δεύ. στο x, y** εισάγουμε ως χρόνο τη μεταβλητή «χρονός» ώστε οι μπάλες να κινούνται με διαφορετική ταχύτητα στα διαφορετικά επίπεδα δυσκολίας του παιχνιδιού. Μελετήστε το παρακάτω σενάριο για ένα από τα αντικείμενα-στόχους:



Γιατί στο σενάριο υπάρχει η εντολή **αλλαγή σε ενδυμασία...**; Μη ξεχνάτε ότι μπορεί το αντικείμενό μας να έχει «σκάσει» από το χτύπημα του χρήστη και να φορά συνεπώς την ενδυμασία «έκρηξη». Άρα με την εντολή **αλλαγή σε ενδυμασία... α...** εξασφαλίζουμε ότι θα φορά την αρχική του ενδυμασία. Μήπως όμως υπάρχει καλύτερο μέρος για να εισάγουμε την εντολή αυτή; Πως θα μπορούσαμε να κάνουμε πιο «αποδοτικό» το σενάριό μας;

Για να αποκτήσουν όλα τα αντικείμενα την ίδια ενδυμασία σύρετε το εικονίδιο της από τη **λίστα ενδυμασιών**, πάνω στα αντικείμενα που βρίσκονται στη **λίστα αντικειμένων**. Με πολύ γρήγορο τρόπο μπορείτε να εισάγετε μια κοινή ενδυμασία σε πολλαπλά αντικείμενα. Το ίδιο ισχύει και για τους ήχους. Αλλά ακριβώς το ίδιο μπορείτε να κάνετε και όταν θέλετε να αντιγράψετε ένα σενάριο σε ένα άλλο αντικείμενο. Δεν έχετε παρά να το σύρετε πάνω στο αντικείμενο που βρίσκεται στη λίστα αντικειμένων

Επιπλέον, όμως, όταν θεωρούμε ότι μια σειρά από αντικείμενα θα έχουν στο πρόγραμμά μας πανομοιότυπη συμπεριφορά, μπορούμε να δημιουργήσουμε μόνο ένα αρχικά, και στη συνέχεια τα άλλα να προκύψουν ως αντίγραφα του πρώτου.

Με όλες τις προηγούμενες οδηγίες μπορείτε να κερδίσετε πολύτιμο χρόνο.

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Δημιουργήστε διαφορετικούς μεγέθους στόχους που δίνουν και διαφορετική βαθμολογία.
- 2) Δημιουργήστε ένα τέταρτο επίπεδο δυσκολίας στην αρχική οθόνη του παιχνιδιού σύμφωνα με το οποίο η δυσκολία του παιχνιδιού θα αυξάνεται με το πέρασμα του χρόνου (π.χ. κάθε 20 δευτερόλεπτα).



15.4 Ο καρχαρίας

[15_π04.sb]

Το επόμενο παιχνίδι που θα αναλύσουμε λαμβάνει χώρο σε υδάτινο περιβάλλον και βασικός ήρωας είναι ένας πεινασμένος καρχαρίας. Ο χρήστης ελέγχει τον καρχαρία με τα βελάκια του πληκτρολογίου με στόχο να τον ταΐσει με τα καημένα ψαράκια που περιπλανώνται στο βυθό της θάλασσας. Πρέπει όμως να αποφύγει αυτά που είναι δηλητηριώδη. Ο χρήστης κερδίζει πόντους ανάλογα με το είδος των ψαριών που τρώει ο καρχαρίας, ενώ αν φάει δηλητηριώδες ψάρι, τότε χάνει μια ζωή.

Ποιο συγκεκριμένα, τα επόμενα ψάρια δίνουν τους εξής βαθμούς:

- ✓ Κίτρινα ψάρια (1 βαθμός).
- ✓ Μοβ ψάρια (3 βαθμοί).
- ✓ Χταπόδι (5 βαθμοί).

Το κόκκινο ψάρι είναι δηλητηριώδες και όταν ακουμπά τον καρχαρία, αφαιρείται μια ζωή από το χρήστη. Κατά την εκκίνηση του παιχνιδιού ο χρήστης διαθέτει 3 ζωές.

Στο παιχνίδι συμμετέχουν και μικρά ψαράκια που δεν αποτελούν τροφή για τον καρχαρία και κάνουν τη ζωή του παίχτη δυσκολότερη. Παρακάτω φαίνεται μια εικόνα του παιχνιδιού



Τα αντικείμενα

Τα αντικείμενα ανιχνεύονται σχετικά εύκολα αφού για κάθε μορφή ζωής μέσα στον υδάτινο κόσμο μας χρειαζόμαστε ένα αντικείμενο. Το αντικείμενο καρχαρίας θα ελέγχεται από το χρήστη ενώ όλα τα υπόλοιπα θα κινούνται από μόνα τους.

Το σκηνικό

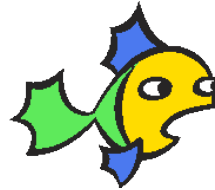
Το σκηνικό μας είναι απλό και η μόνη ζητούμενη συμπεριφορά είναι η αναπαραγωγή του ήχου bubbles (από τη βιβλιοθήκη ήχων του Scratch) για να δώσουμε την εντύπωση στο χρήστη ότι βρίσκεται πραγματικά μέσα σε ένα υδάτινο περιβάλλον. Το σκηνικό που έχει επιλεγεί ονομάζεται underwater και βρίσκεται στο φάκελο nature της βιβλιοθήκης υποβάθρων του Scratch. Στο σενάριο του σκηνικού, είναι σημαντικό να θυμηθούμε ότι μέσα στην εντολή επανάληψης για τον ήχο πρέπει να επιλέξουμε την εντολή αναπαραγωγής παιξε ήχο...μέχρι τέλους και όχι την παιξε ήχο... Δοκιμάστε το αντίστροφο και εξηγήστε τι συμβαίνει.



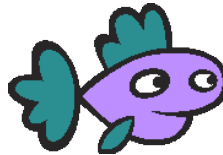
Ψάρια και χταπόδι

Ας μελετήσουμε την εμφάνιση και τη συμπεριφορά των ψαριών και του χταποδιού.

Για το κίτρινο ψάρι θα επιλέξουμε ως βασική ενδυμασία την εικόνα fish3 από τον κατάλογο animals της βιβλιοθήκης αντικειμένων του Scratch. Το κίτρινο ψάρι εμφανίζεται σε τυχαία θέση στο σκηνικό στην αρχή του παιχνιδιού αλλά και αφού φαγωθεί (!). Κινείται συνεχώς με σταθερή ταχύτητα και σε κατεύθυνση που μπορεί να αλλάζει λίγο με τυχαίο τρόπο, ενώ στην περίπτωση που φτάσει στα όρια πρέπει να αναπηδά. Το κίτρινο ψάρι δίνει 1 βαθμό στον καρχαρία όταν το φάει. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα.



Για το μοβ ψάρι επιλέγουμε ως ενδυμασία την εικόνα fish2 από τον κατάλογο animals. Και το μοβ ψάρι εμφανίζεται σε τυχαία θέση στο σκηνικό και κινείται συνεχώς από μόνο του με σταθερή ταχύτητα και σε τυχαίες κατευθύνσεις χώρο. Το μοβ ψάρι δίνει 3 βαθμούς στον καρχαρία. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα.



Για το κόκκινο ψάρι επιλέγουμε ως ενδυμασία την εικόνα fish4 από τον κατάλογο animals. Το κόκκινο ψάρι θα πρέπει να εμφανίζεται αρχικά σε τυχαία θέση στο σκηνικό μας, εκτός όμως από την περιοχή στην οποία βρίσκεται ο καρχαρίας (π.χ. στην αρχή του παιχνιδιού τοποθετείται στο πάνω αριστερό άκρο της οθόνης). Αν ακουμπήσει το κόκκινο ψάρι τον καρχαρία, ο χρήστης χάνει μια ζωή και θα πρέπει να αποφεύγουμε να συμβαίνει κάτι τέτοιο στην αρχή του παιχνιδιού και μετά από κάθε «χάσιμο» ζωής.



Το χταπόδι εμφανίζεται αρχικά στα κάτω όρια της οθόνης σε τυχαία οριζοντίως θέση, ενώ κινείται διαρκώς κατακόρυφα μέχρι τα πάνω όρια. Για να δείξουμε την κίνηση του χταποδιού όταν ανεβαίνει από τον βυθό προς τα πάνω, θα προσπαθήσο-

υμε να φαίνεται ότι σπρώχνει το νερό προς τα κάτω οπότε και δημιουργείται η απαιτούμενη ώθηση. Για το λόγο αυτό θα χρησιμοποιήσουμε ως ενδυμασίες, τις εικόνες octopus1-a και octopus1-b από το φάκελο animals και θα τις εμφανίζουμε εναλλάξ.



Το χταπόδι δίνει 5 βαθμούς στον καρχαρία. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα σε άλλη θέση στο κάτω μέρος της οθόνης. Όπως φαίνεται από τις προηγούμενες εικόνες, οι ενδυμασίες δεν έχουν κατεύθυνση ακριβώς προς τα πάνω αλλά προς πάνω-δεξιά και για αυτό θα πρέπει να αλλάξουμε λίγο την κατεύθυνση τους.



Τέλος, υπάρχει και το μικρό ψάρι που δεν αποτελεί τροφή για το καρχαρία. Για τα ψάρια αυτά επιλέγουμε την εικόνα fish1-a. Η συμπεριφορά τους είναι απλή καθώς τριγυρίζουν στον βυθό με σταθερή οριζόντια κατεύθυνση και κινούνται με χαμηλή σταθερή ταχύτητα.

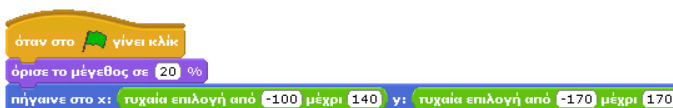
Θα έχετε διαπιστώσει ότι οι εικόνες που τοποθετήσαμε στο παιχνίδι μας για τους διάφορους ήρωες είναι πολύ μεγάλες σε σχέση με μέγεθος της οθόνης του Scratch. Σε αυτό το παράδειγμα θα τις μικρύνουμε προγραμματιστικά χρησιμοποιώντας την εντολή όρισε το μέγεθος σε...% από την παλέτα Όψεις.

- Για τα κίτρινα ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 20%.
- Για τα κόκκινα ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 15%.
- Για τα μοβ ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 20%.
- Για τα μικρά ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 10%.

Ποια είναι τα σενάρια των προηγούμενων αντικειμένων; Ποιες συμπεριφορές επιδεικνύουν; Θα μπορούσαμε να διακρίνουμε δυο είδη συμπεριφορών:

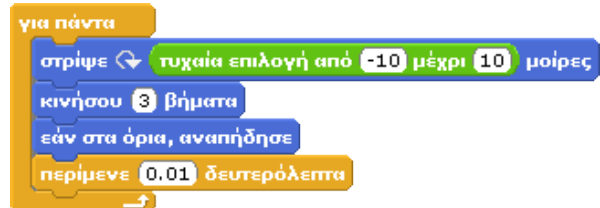
- A) Αρχικοποίηση θέσης και εμφάνιση.
- B) Συνεχής κίνηση.
- Γ) Αντίδραση όταν έρχονται σε επαφή με τον καρχαρία.

Βήμα πρώτο: Όλα τα ψάρια εμφανίζονται σε τυχαία θέση στο σκηνικό και συνεπώς θα χρησιμοποιήσουμε την εντολή τυχαία επιλογή από...μέχρι.... Ταυτόχρονα, πρέπει να ορίσουμε και το κατάλληλο μέγεθος. Άρα το σενάριό τους ξεκινά κάπως έτσι:



Βήμα δεύτερο: Όλα τα ψάρια κινούνται διαρκώς. Συνεπώς οι εντολές κίνησης θα πρέπει να βρίσκονται μέσα σε μια εντολή

επανάληψης για πάντα. Λόγω του ότι το ψάρι καθώς κινείται θα πρέπει να αλλάζει κατευθύνσεις, καλό θα ήταν να χρησιμοποιήσουμε συνδυαστικά τις εντολές κινήσου. . . βήματα και στρίψε δεξιόστροφα. . . μοίρες. Θέλουμε το ψάρι να κινείται τυχαία, χωρίς όμως απότομες αλλαγές κατεύθυνσης. Έτσι θα χρησιμοποιήσουμε την εντολή τυχαία επιλογή από...μέχρι... προσδιορίζοντας ένα μικρό εύρος μοιρών (π.χ. -10 μέχρι 10). Για να μειώσουμε την ταχύτητα με την οποία κινείται το ψάρι θα βάλουμε επίσης την εντολή περίμενε. . . δευτερόλεπτα και να δώσουμε μια πολύ μικρή τιμή καθυστέρησης (π.χ. 0.01). Το σενάριο του ψαριού θα πρέπει να περιέχει τον επόμενο συνδυασμό εντολών:



Ανάλογα με τις ιδιότητες που θέλουμε να έχει κάθε είδος ψαριού μπορούμε να χρησιμοποιήσουμε διαφορετικές τιμές στα πιο πάνω πεδία.

Βήμα τρίτο: Θέλουμε να ελέγχουμε διαρκώς το κατά πόσο το αντικείμενο άγγιξε τον καρχαρία. Για να το καταφέρουμε αυτό θα βάλουμε μέσα στην προηγούμενη εντολή επανάληψης μια εντολή εάν... που θα ελέγχει ακριβώς για αυτή τη συνθήκη. Συγκεκριμένα, συνδυάζουμε την εντολή εάν... με τον αισθητήρα αγγίζει το...



Ποια είναι η συμπεριφορά του ψαριού όταν αγγίζει τον καρχαρία; Καταρχάς, πρέπει να αυξηθεί η τιμή της μεταβλητής σκορ, ανάλογα με το ψάρι για το οποίο δημιουργούμε το σενάριο (π.χ. κίτρινο ψάρι 1 βαθμός). Αμέσως μετά, το αντικείμενο πρέπει να κρυφτεί καθώς θα βρίσκεται στη κοιλιά του καρχαρία. Στη συνέχεια πρέπει να ξαναεμφανιστεί μετά από κάποιο τυχαίο χρονικό διάστημα και, τέλος, να βρεθεί σε μια τυχαία θέση του υδατινού κόσμου. Για να γίνει όμως πιο πειστικό το παιχνίδι, το ψάρι θα ξανα-εμφανίζεται από τα άκρα της οθόνης ως νέο ψάρι (!) για το χρήστη, αφού το προηγούμενο το έχει φάει ο καρχαρίας. Θα προσπαθήσουμε το ψάρι να ξεκινά είτε από το αριστερό είτε από το δεξί άκρο της οθόνης τυχαία. Πως θα προγραμματίσουμε αυτές τις συμπεριφορές;

Για την απόκρυψη του ψαριού θα χρησιμοποιήσουμε την εντολή απόκρυψη. Για την τυχαία καθυστέρηση μέχρι την επανεμφάνιση θα χρησιμοποιούμε το συνδυασμό εντολών τυχαία επιλογή από...μέχρι... και περίμενε...δευτερόλεπτα, όπως κάναμε και στο προηγούμενο παιχνίδι.



Για την επιλογή τυχαίας θέσης επανεμφάνισης, θα εφαρμόσουμε ένα τρικ. Θα δημιουργήσουμε μια συνθήκη που θα ελέγχει αν το αποτέλεσμα του τελεστή τυχαία επιλογή από...μέχρι... είναι 1 ή 2 και ανάλογα θα προσδιορίζουμε συντε-

ταγμένες εμφάνισης για το ψάρι από το αριστερό ή το δεξί άκρο της οθόνης. Μελετήστε το επόμενο σενάριο:

```

εάν <τυχαία επιλογή από 1 μέχρι 2 = 1>
  αλλιώς

```

Στις δυο περιπτώσεις που προκύπτουν από την εντολή εάν...αλλιώς θα προσδιορίσουμε τη θέση x (240 ή -240), τη κατεύθυνση (-90 ή 90 αντίστοιχα) και θα θέσουμε τυχαίο y για τις συντεταγμένες του ψαριού. Έτσι το ολοκληρωμένο σενάριο για το κίτρινο ψάρι είναι το ακόλουθο:

```

όταν στο <πλήκτρο> γίνει κλικ
  όρισε το μέγεθος σε 20 %
  πήγαινε στο x: <τυχαία επιλογή από -100 μέχρι 140> y: <τυχαία επιλογή από -170 μέχρι 170>
  για πάντα
    εάν <αγγίζει το <καρχαρίας>>
      περίμενε 0.5 δευτερόλεπτα
      παίξε ήχο <Pop>
      άλλαξε <score> κατά 1
      απόκρυψη
      περίμενε <τυχαία επιλογή από 1 μέχρι 5> δευτερόλεπτα
      εάν <τυχαία επιλογή από 1 μέχρι 2 = 1>
        θέσε το x ίσο με -240
        δείξε στην κατεύθυνση 90
      αλλιώς
        θέσε το x ίσο με 240
        δείξε στην κατεύθυνση -90
      θέσε το y ίσο με <τυχαία επιλογή από -180 μέχρι 180>
      εμφάνισε
      στρίψε <τυχαία επιλογή από -10 μέχρι 10> μοίρες
      κινήσου 3 βήματα
      εάν στα όρια, αναπήδησε
      περίμενε 0.01 δευτερόλεπτα

```

Προσέξτε ότι έχει προστεθεί η εντολή περίμενε 0.5 δευτερόλεπτα αμέσως μετά τη συνθήκη που αφορά το άγγιγμα του καρχαρία. Γιατί; Μήπως για να προλάβει ο καρχαρίας να αλλάξει ενδυμασία, δηλαδή να ανοίξει το στόμα του και στη συνέχεια να φάει το ψάρι;

Αυτό το σενάριο είναι πανομοιότυπο για τα μοβ και τα κίτρινα ψάρια. Τα μικρά ψάρια απλά κινούνται και δεν περιέχουν το μπλοκ εντολών που βρίσκεται μέσα στην πρώτη εντολή εάν... ενώ για τα εχθρικά ψάρια, υπάρχουν δυο αλλαγές: α) στο άγγιγμα του καρχαρία αντί να αυξάνουν τη μεταβλητή «σκορ», μειώνουν τη μεταβλητή «ζωές», και β) προσδιορίζουμε ως συντεταγμένες νέας εμφάνισης, θέσεις εκτός της γωνίας στην οποία θα επανεμφανιστεί ο καρχαρίας. Μια λίγο μεγαλύτερη διαφοροποίηση έχει το σενάριο του χταποδιού που κινείται κατακόρυφα:

```

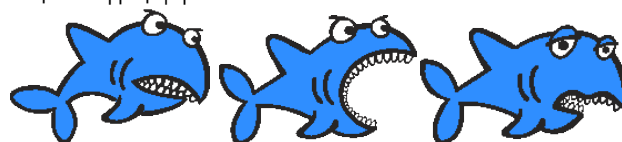
όταν στο <πλήκτρο> γίνει κλικ
  όρισε το μέγεθος σε 20 %
  για πάντα
    θέσε το x ίσο με <τυχαία επιλογή από -235 μέχρι 235>
    θέσε το y ίσο με -180
    δείξε στην κατεύθυνση 65
    εμφάνισε
    επανάλαβε ώσπου <θέση y > 160>
      θέσε το y ίσο με <θέση y + 25>
      αλλαγή σε ενδυμασία <octopus1-a>
      περίμενε 0.15 δευτερόλεπτα
      αλλαγή σε ενδυμασία <octopus1-b>
      περίμενε 0.15 δευτερόλεπτα
      εάν <αγγίζει το <καρχαρίας>>
        περίμενε 0.5 δευτερόλεπτα
        παίξε ήχο <Pop>
        άλλαξε <score> κατά 5
        απόκρυψη
      απόκρυψη
      περίμενε <τυχαία επιλογή από 1 μέχρι 5> δευτερόλεπτα

```

Μπορείτε να διακρίνετε ότι η κίνηση του χταποδιού γίνεται ουσιαστικά μέσα στην εντολή επανάλαβε ώσπου... και διαρκεί μέχρι το χταπόδι να φτάσει στο πάνω όριο της οθόνης. Κατά τη διάρκεια της κίνησης, γίνεται ο γνωστός έλεγχος για το κατά πόσο το χταπόδι άγγιξε το καρχαρία. Ταυτόχρονα, όμως, το αντικείμενο αλλάζει ενδυμασίες και αυξάνει μόνο τη συντεταγμένη y.

Ο καρχαρίας

Ο καρχαρίας εμφανίζεται αρχικά πάνω αριστερά στην οθόνη, ελέγχουμε τη θέση του με τα βελιάκια του πληκτρολογίου και μπορεί να φάει διαφορετικά είδη ψαριών ενώ όταν ακουμπήσει κόκκινο ψάρι, τότε του αφαιρείται μια από τις διαθέσιμες ζωές του. Η συμπεριφορά του, όταν ακουμπήσει κάποιο ψάρι, είναι να στρέφεται προς το μέρος του θύματος, να ανοίγει το στόμα του και να το τρώει. Πιο συγκεκριμένα, θα προσδιορίσουμε ως αρχική μορφή του καρχαρία την ενδυμασία shark-b από τη βιβλιοθήκη αντικειμένων του Scratch, ενώ όταν ο καρχαρίας ακουμπήσει ένα ψάρι, στρέφεται προς το μέρος του και παίρνει τη μορφή shark-b.



Στη συνέχεια, το ψάρι πρέπει να εξαφανίζεται και ο καρχαρίας να επανέρχεται στην αρχική του μορφή. Επιπλέον, όταν ο καρχαρίας ακουμπήσει ένα κόκκινο ψάρι τότε θα πρέπει να φορέσει την ενδυμασία απογοήτευσης shark-c, να χάνει το εφέ του χρώματός του κατά 20 (να φαίνεται πιο αδύναμος) και να επιστρέφει στην αρχική του θέση.

Άρα ο καρχαρίας:

Α) Πρέπει να αρχικοποιεί τις μεταβλητές που αφορούν το σκορ και τις ζωές, να αλλάζει μέγεθος και να πηγαινει σε μια αρχική θέση,

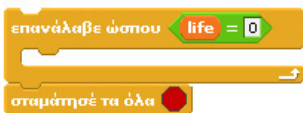
Β) πρέπει διαρκώς να ελέγχει για το αν άγγιξε κάποιο από τα αντικείμενα που το επηρεάζουν, και

Γ) πρέπει να μετακινείται με τα βελάκια του πληκτρολογίου.

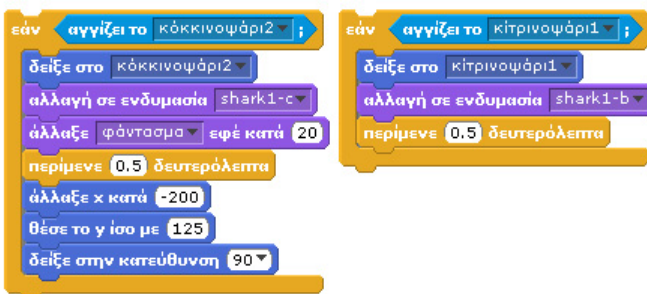
Το πρώτο κομμάτι της συμπεριφοράς του μπορούμε να το δημιουργήσουμε εύκολα:



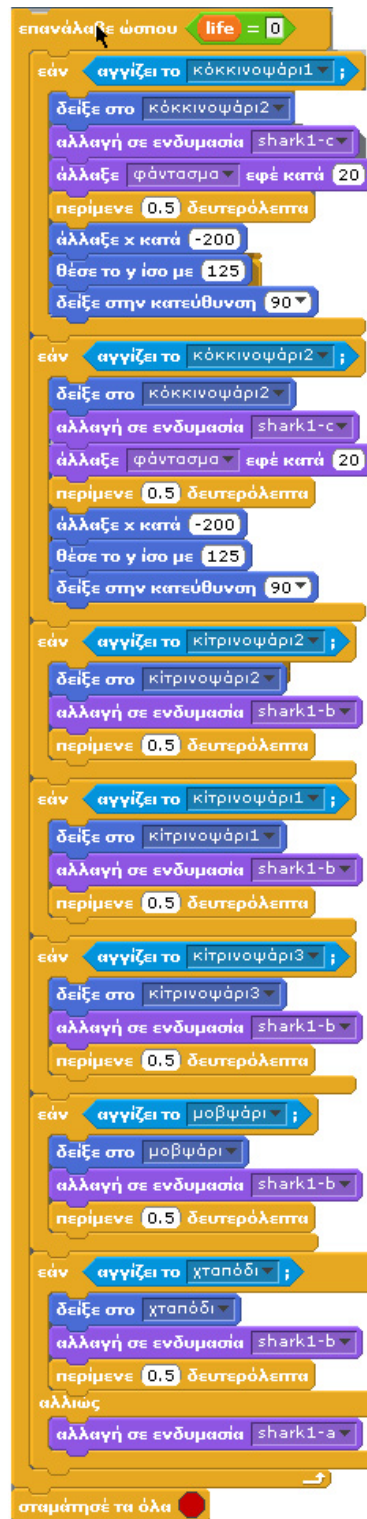
Για το δεύτερη συμπεριφορά του, πρέπει να δημιουργήσουμε μια επανάληψη μέσα στην οποία θα ελέγχουμε το αν άγγιξε άλλα αντικείμενα. Αλλα πόσο θα διαρκεί αυτή η επανάληψη; Μέχρι η μεταβλητή ζωής να πάρει την τιμή 0. Για το λόγο αυτό θα χρησιμοποιήσουμε την εντολή **επανάλαβε ώσπου...** και μετά από αυτή θα χρησιμοποιήσουμε την εντολή **σταμάτησέ τα όλα** ώστε να τερματίζεται το παιχνίδι όταν δεν υπάρχουν άλλες διαθέσιμες ζωές. Δηλαδή πρέπει να προσθέσουμε τις εντολές:



Επιπλέον, χρειαζόμαστε τις συνθήκες ελέγχου για το αν άγγιξε άλλο αντικείμενο. Αν άγγιξει άλλο αντικείμενο, ο καρχαρίας πρέπει αλλάξει την κατεύθυνσή του για να δείξει προς αυτό (ώστε όταν ανοίξει το στόμα του να είναι κοντά το ψάρι!), να ανοίξει το στόμα του και να περιμένει λίγο ώστε να προλάβει ο χρήστης να παρακολουθήσει το ανοιγμένο στόμα του. Στην περίπτωση που ακουμπήσει κόκκινο ψάρι, τότε φοράει την ενδυμασία του απογοητευμένου καρχαρία, γίνεται πιο αδύναμος με το εφέ «φάντασμα» και στη συνέχεια μεταφέρεται στην αρχική του θέση. Παρακάτω παρουσιάζονται τα δυο είδη συνθηκών που θα περιληφθούν στην εντολή **επανάλαβε ώσπου...**



Μια από τις δυο συνθήκες χρειαζόμαστε για όλα τα αντικείμενα. Το δεύτερο τμήμα του σεναρίου εμφανίζεται στην παρακάτω εικόνα:



Τέλος, για το κομμάτι της κίνησης του καρχαρία, δεν μπορούμε να χρησιμοποιήσουμε τις εντολές κίνησης **άλλαξε x κατά...** και **άλλαξε y κατά...** καθώς ο καρχαρίας θα φαινόταν να πηγαίνει ακόμη και με την όπισθεν! Χρειαζόμαστε εντολές κίνησης που ενσωματώνουν κατεύθυνση. Επομένως, θα χρησιμοποιήσουμε τις εντολές **δείξε στην κατεύθυνση...** και **κινήσου...βήματα**:

όταν το πλήκτρο κάτω βέλος πατηθεί
δείξε στην κατεύθυνση 180
κινήσου 5 βήματα

όταν το πλήκτρο πάνω βέλος πατηθεί
δείξε στην κατεύθυνση 0
κινήσου 5 βήματα

όταν το πλήκτρο δεξί βέλος πατηθεί
δείξε στην κατεύθυνση 90
κινήσου 5 βήματα

όταν το πλήκτρο αριστερό βέλος πατηθεί
δείξε στην κατεύθυνση -90
κινήσου 5 βήματα

Το παιχνίδι μας είναι έτοιμο και είναι πραγματικά απολαυστικό. Ως προγραμματιστές το δοκιμάσαμε για πολύ ώρα ☺

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Δημιουργήστε στο σκηνικό φυσικά εμπόδια με χρώμα τα οποία περιορίζουν τις δυνατότητες κίνησης του καρχαρία. Δεν είναι και μικρός!
- 2) Ζητήστε από το χρήστη να προσδιορίσει στην αρχή του παιχνιδιού το επιθυμητό επίπεδο δυσκολίας, όπως στο προηγούμενο παιχνίδι, και εμφανίστε διαφορετικό αριθμό κόκκινων ψαριών για τα διαφορετικά επίπεδα.
- 3) Δημιουργείστε ένα turbo-boost για τον καρχαρία. Δώστε τη δυνατότητα στο χρήστη με ένα πλήκτρο να μετακινεί το καρχαρία πολύ γρηγορότερα για κάποια δευτερόλεπτα. Το turbo-boost, θα μπορεί να χρησιμοποιηθεί μόνο 2 φορές κατά τη διάρκεια του παιχνιδιού.



15.5 Καροτοκυνηγός

[15_π05.sb]

Το παιχνίδι λαμβάνει χώρα σε ένα κτήμα, όπου στη δεξιά του πλευρά του υπάρχει ένα χωράφι με καρότα τα οποία οριοθετούνται από μια λευκή ευθεία γραμμή αριστερά τους (βλ. επόμενη εικόνα). Το πρόβλημα όμως είναι ότι διάφοροι επιτήδειοι (!) λαγοί θέλουν να φάνε τα καρότα και ο κτηματίας είναι αναγκασμένος να πάρει σκληρά μέτρα. Για αυτό και δεξιά από το χωράφι τοποθετεί ένα κυνηγό (!), που ελέγχεται από το χρήστη και ο οποίος μετακινείται προς τα πάνω ή προς τα κάτω με τα δυο αντίστοιχα βελάκια του πληκτρολογίου. Ο κυνηγός πυροβολεί όταν ο χρήστης πατάει το πλήκτρο κενό (space) με στόχο να σκοτώσει τους λαγούς. Όταν πυροβολεί μία σφαίρα, αυτή πρέπει να εξέρχεται από το όπλο του και να κινείται μέχρι τα αριστερά όρια του σκηνικού ή μέχρι να πετύχει το λαγό ενώ στη συνέχεια θα πρέπει να εξαφανίζεται.



Ο λαγός εμφανίζεται σε κάποια τυχαία θέση στα αριστερά όρια του σκηνικού και κινείται προς τα καρότα φυσικά.

Αν ο κυνηγός πετύχει το λαγό, πριν αυτός αγγίξει τη λευκή γραμμή του χωραφιού, τότε ο λαγός εξαφανίζεται και επανεμφανίζεται στα αριστερά όρια του σκηνικού, συνεχίζοντας την κίνησή του προς το χωράφι, ενώ ταυτόχρονα πρέπει να αυξάνεται η βαθμολογία του χρήστη. Όταν ο χρήστης πετύχει το λαγό αριθμό φορές πολλαπλάσιο του δέκα, η δυσκολία του παιχνιδιού πρέπει να μεγαλώνει και το μέγεθος του λαγού να μειώνεται.

Αν ο λαγός καταφέρει να φτάσει στην άσπρη γραμμή δίπλα από τα καρότα, τότε ο χρήστης χάνει μια ζωή και ο λαγός αρχίζει πάλι την κίνησή του από τα αριστερά όρια του σκηνικού μετά από ένα δευτερόλεπτο. Οι ζωές του χρήστη εμφανίζονται αρχικά με τρεις καρδιές στο πάνω δεξιά μέρος του σκηνικού, και πρέπει αντίστοιχα να μειώνονται κάθε φορά που ο χρήστης χάνει μια ζωή.

Στόχος του παιχνιδιού είναι ο κυνηγός να πετύχει όσο το δυνατόν μεγαλύτερη βαθμολογία, ενώ το παιχνίδι τελειώνει όταν ο κυνηγός χάσει και τις τρεις ζωές του. Στο τέλος του παιχνιδιού πρέπει να εμφανίζεται το μήνυμα «Έχασες...!».

Αντικείμενα

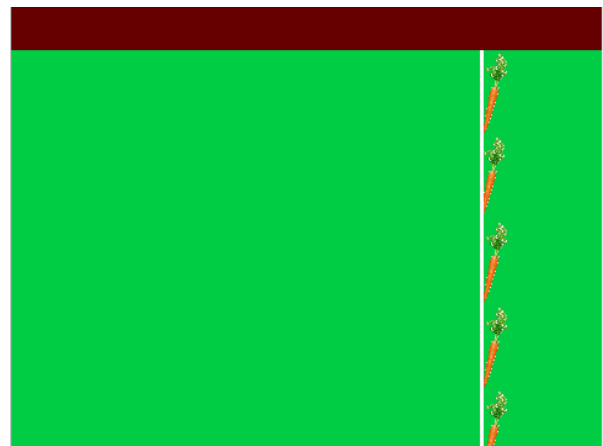
Ποια θα είναι τα αντικείμενα του έργου μας; Ο λαγός και ο κυνηγός είναι τα δυο προφανή. Επιπλέον, για να κινείται η σφαίρα θα πρέπει να δημιουργήσουμε ένα αντίστοιχο αντικείμενο ενώ και οι καρδιές που πρέπει να μειώνονται υποδηλώνουν ένα αντικείμενο με διαφορετικές ενδυμασίες. Τέλος, χρειαζόμαστε ένα αντικείμενο το οποίο θα παρουσιάσει το μήνυμα ολοκλήρωσης του παιχνιδιού.

Μεταβλητές

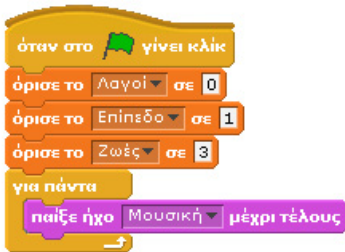
Πρέπει να αποθηκεύουμε κάπου τη βαθμολογία του παίχτη, τις ζωές του και το επίπεδο του παιχνιδιού. Για αυτό και θα χρειαστούμε τρεις μεταβλητές, με ονόματα «Λαγοί», «Ζωές» και «Επίπεδο», και αρχικές τιμές «0», «3» και «1» αντίστοιχα. Οι μεταβλητές μπορούν να αρχικοποιούνται από το σκηνικό με το κλικ στην πράσινη σημαία.

Σκηνικό

Σημαντικό ρόλο στο πρόβλημά μας διαδραματίζει η λευκή ευθεία γραμμή που οριοθετεί το χωράφι με τα καρότα και την οποία προσπαθεί να αγγίξει ο λαγός. Το σκηνικό μας πρέπει να την περιέχει, οπότε θα χρειαστεί να σχεδιάσουμε εμείς ένα σκηνικό με τη βοήθεια του επεξεργαστή ζωγραφικής του Scratch. Στο σκηνικό μπορούμε να βάλουμε εικόνες από καρότα για το χωράφι, ενώ καλό θα ήταν να υπάρχει χώρος στο πάνω μέρος για να φαίνονται καθαρά οι ζωές, το επίπεδο του παιχνιδιού και η βαθμολογία του κυνηγού. Δε χρειάζεται να προσθέσουμε διαφορετικά υπόβαθρα στο σκηνικό μας, ενώ το μοναδικό σενάριο που θα περιέχει το σκηνικό, είναι η αρχικοποίηση των μεταβλητών του προβλήματος και μια επανάληψη για πάντα, μέσα στην οποία θα αναπαράγεται ένα μουσικό κομμάτι. Προσπαθήστε να δημιουργήσετε ένα σκηνικό παρόμοιο με αυτό που παρουσιάζεται παρακάτω.



Το σενάριο του σκηνικού είναι απλό:



Κυνηγός

Για το αντικείμενο του κυνηγού μπορούμε να βρούμε μια εικόνα από το διαδίκτυο, ενώ δεν είναι απαραίτητο να χρησιμοποιήσουμε επιπρόσθετες ενδυμασίες ή κάποιον ήχο. Ο κυνηγός κινείται από το χρήστη προς τα πάνω και προς τα κάτω, με το πάτημα των πλήκτρων «πάνω βέλος» και «κάτω βέλος».



Άρα, η συμπεριφορά του κυνηγού περιλαμβάνει απλά την κάθετη κίνησή του στην περιοχή που υπάρχουν τα καρότα. Στην έναρξη του παιχνιδιού, ο κυνηγός μας πρέπει να πληροφορήσει το χρήστη ότι πυροβολεί όταν πατηθεί το πλήκτρο «κενό», οπότε χρησιμοποιούμε την εντολή πες...για...δευτερόλεπτα, σε συνδυασμό με τη φράση «Πάτα κενό για να πυροβολήσω...!». Θα χρησιμοποιούμε την εντολή όταν το πλήκτρο πατηθεί για να αλλάξουμε τη κάθετη θέση του κυνηγού, δηλαδή τη συντεταγμένη y . Όμως θα πρέπει να περιορίσουμε την κίνηση του κυνηγού μόνο στην περιοχή του χωραφιού που υπάρχουν τα καρότα. Για αυτό θα πρέπει να κάνουμε χρήση μιας εντολής εάν..., η οποία δε θα επιτρέπει την κίνησή του όταν αυτός φτάσει στα άκρα της επιτρεπόμενης περιοχής κίνησης. Μελετήστε το σενάριο του κυνηγού.



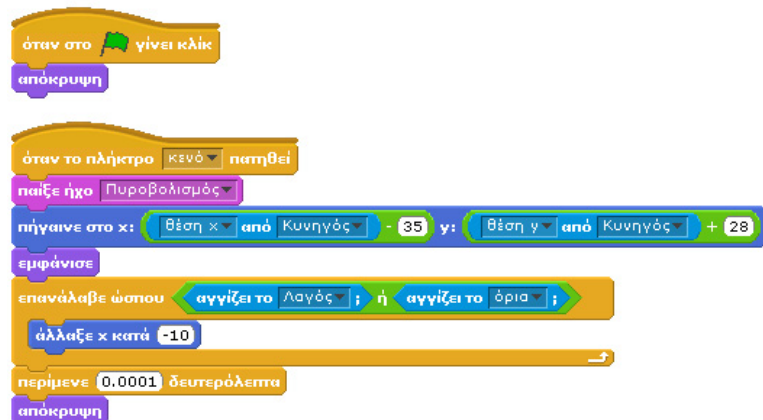
Προκύπτει μια εύλογη ερώτηση: γιατί δε βάζουμε στο αντικείμενο-κυνηγός, τη διαχείριση των ζωών του κυνηγού ή της βαθμολογίας; Η απάντηση είναι γιατί απλά δεν εξαρτώνται από αυτόν και τη συμπεριφορά του. Από τη στιγμή που ο κυνηγός χάνει μια ζωή κάθε φορά που ο λαγός ακουμπά την άσπρη γραμμή, ως αντικείμενο (ο κυνηγός) δεν

εμπλέκεται στο συγκεκριμένο συμβάν. Το ίδιο ισχύει και για κάθε φορά που μια σφαίρα βρίσκει το λαγό και κερδίζεται ένας πόντος. Άρα τη διαχείριση των ζωών και τη βαθμολογία, θα πρέπει να την ανιχνεύσει ο λαγός που με τη σειρά του θα στέλνει τα κατάλληλα μηνύματα.

Σφαίρα

Μπορούμε να σχεδιάσουμε εμείς το αντικείμενο της σφαίρας με μια μικρή ευθεία μαύρη γραμμή, ενώ πρέπει να βρούμε και να εισάγουμε κάποιον ήχο πυροβολισμού. Η σφαίρα εμφανίζεται στο σκηνικό όταν πατηθεί το πλήκτρο «κενό» από το χρήστη και κινείται προς τα αριστερά μέχρι να πετύχει το λαγό ή να φτάσει τα αριστερά όρια του σκηνικού. Όταν μια από τις δυο συνθήκες ικανοποιηθεί, η σφαίρα θα πρέπει να εξαφανιστεί. Σε ποιο σημείο του σκηνικού πρέπει όμως να εμφανιστεί; Στην περιγραφή του προβλήματός μας είπαμε ότι η σφαίρα εξέρχεται από το όπλο του κυνηγού, δηλαδή από το πάνω αριστερό άκρο του αντικειμένου «Κυνηγός». Άρα το αντικείμενο «Σφαίρα» πρέπει να εμφανιστεί αριστερότερα από το «Κυνηγός» (δηλαδή με x μειωμένο σε σχέση με τη θέση x του «Κυνηγός») και λίγο πιο πάνω (δηλαδή y μεγαλύτερο σε σχέση με τη θέση y του «Κυνηγός»). Για να πετύχουμε τη συγκεκριμένη θέση, θα πρέπει να χρησιμοποιήσουμε τον αισθητήρα θέση x/y από μορφή..., ο οποίος δίνει τη δυνατότητα σε ένα αντικείμενο να αξιοποιεί τις συντεταγμένες ενός άλλου αντικειμένου. Στη συγκεκριμένη περίπτωση η σφαίρα εξέρχεται σε θέση σχετική με τις συντεταγμένες του κυνηγού. Ίσως χρειαστεί να δοκιμάσουμε ορισμένες τιμές για να καταφέρουμε να βρούμε την κατάλληλη θέση εμφάνισης της σφαίρας.

Είναι επίσης σαφές ότι πρέπει να χρησιμοποιήσουμε μια εντολή επανάληψης (όπως π.χ. την εντολή επανάλαβε ώσπου), ώστε η κίνηση της σφαίρας να συνεχίζεται μέχρι να αγγίξει το αντικείμενο Λαγός ή τα όρια του σκηνικού. Οπότε, για τη συνθήκη τερματισμού θα χρειαστούμε και τον τελεστή ...ή.... Η σφαίρα θα πρέπει να είναι αρχικά κρυμμένη ενώ θα πρέπει να εμφανίζεται όταν πατηθεί το κενό πλήκτρο.



Λαγός

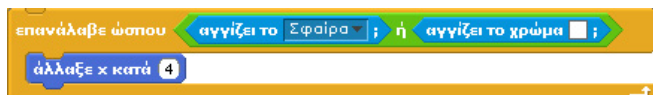
Σε ό,τι αφορά το αντικείμενο του λαγού, επιλέγουμε να εισάγουμε ως ενδυμασία το αρχείο rabbit1 από το φάκελο Animals, της βιβλιοθήκης ενδυμασιών του Scratch. Επίσης, μπορούμε να εισάγουμε κάποιους ήχους για να κάνουμε πιο

διασκεδαστικό το παιχνίδι και συγκεκριμένα τον ήχο Pop, που θα αναπαράγεται όταν η σφαίρα χτυπήσει το λαγό, τον ήχο "WaterDrop" για την αλλαγή επιπέδου και, τέλος, έναν ήχο «σατανικού γέλιου», ο οποίος θα παίζει όταν ο κυνηγός χάνει μία ζωή του. Αντίστοιχους ήχους μπορείτε να βρείτε από τη διεύθυνση <http://www.freesound.org>

Ο λαγός θα εμφανίζεται στα αριστερά όρια του σκηνικού (έστω σε κάποιο συγκεκριμένο σημείο στη έναρξη του παιχνιδιού) και μέχρι να μηδενιστούν οι ζωές του κυνηγού θα κινείται οριζόντια προς τα δεξιά. Η κίνηση του λαγού σταματάει όταν υπάρξει επαφή με τη σφαίρα ή με τη λευκή γραμμή του χωραφιού. Στην πρώτη περίπτωση, θα αυξάνεται η βαθμολογία του χρήστη (η τιμή της μεταβλητής «Λαγοί» αυξάνεται κατά ένα) και θα παίζει ο ήχος Pop. Στη δεύτερη περίπτωση, θα πρέπει να μειώνονται οι ζωές του χρήστη (μειώνεται η τιμή της μεταβλητής «Ζωές» κατά ένα και αλλάζει ενδυμασία το αντικείμενο «Ζωές»), θα παίζει ο τρίτος ήχος του λαγού, ενώ ο λαγός θα αρχίζει πάλι την κίνησή του από τα αριστερά όρια. Όλα αυτά όμως μόνο εφόσον απομένουν ζωές στο χρήστη, γιατί αλλιώς το παιχνίδι θα τερματιστεί.

Όπως είπαμε και στην αρχή του παιχνιδιού, αν ο αριθμός των λαγών που χτύπησε ο κυνηγός γίνει πολλαπλάσιος του 10, τότε θα πρέπει να αλλάξει η δυσκολία του παιχνιδιού και να μειώνεται το μέγεθος του λαγού. Ας προσπαθήσουμε να δημιουργήσουμε το σενάριο που θα περιλαμβάνει όλες τις προηγούμενες συμπεριφορές.

Η βασική κίνηση του λαγού είναι να κινείται οριζόντια προς τα δεξιά, και η κίνησή του θα πρέπει να συνεχίζεται μέχρι είτε να αγγίξει την άσπρη γραμμή του σκηνικού είτε να αγγίξει τη σφαίρα:



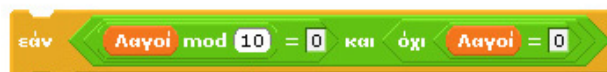
Όταν ο λαγός σταματήσει να κινείται (δηλαδή όταν τερματιστεί η επανάληψη), θα έχει αγγίξει τη σφαίρα ή θα έχει αγγίξει τη λευκή γραμμή. Αν τον έχει πετύχει η σφαίρα, θα πρέπει να εξαφανιστεί αφού «χτυπήθηκε», και να αυξηθεί κατά ένα η τιμή της μεταβλητής «Λαγοί», πριν ξανά-εμφανιστεί το αντικείμενο στα αριστερά του σκηνικού. Η νέα θέση του λαγού θα πρέπει φυσικά να είναι μια τυχαία θέση. Τυχαία σε ό,τι αφορά τη κάθετη συντεταγμένη y του αντικειμένου, αφού το x πρέπει να είναι σταθερό και με τιμή μικρότερη από -200, για να ξεκινάει το αντικείμενο από τα αριστερά όρια του σκηνικού (-217 στην υλοποίηση που παρουσιάζουμε). Για να κάνουμε το y να δέχεται τυχαίες τιμές πρέπει να χρησιμοποιήσουμε την εντολή **τυχαία επιλογή από...μέχρι...**, με τιμές τα κάθετα όρια του κτήματος (πράσινη περιοχή του σκηνικού).



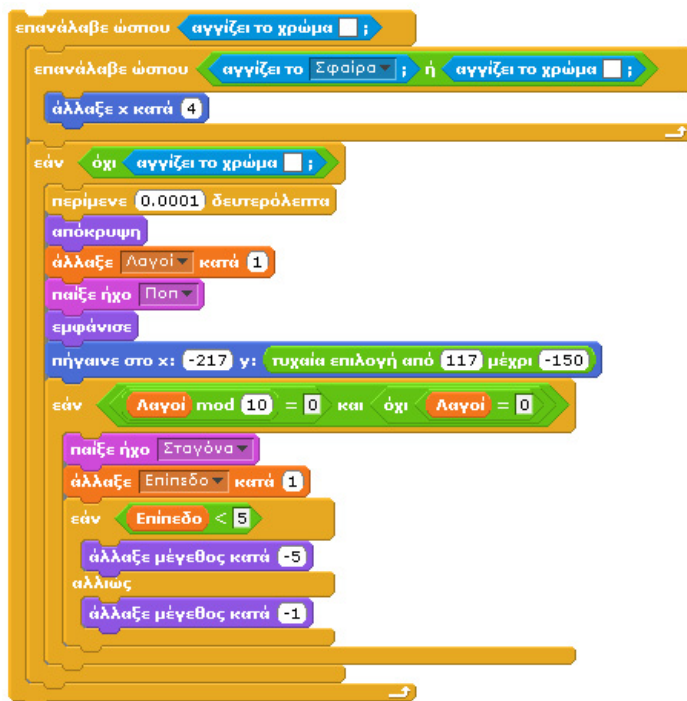
Στη συνέχεια, πρέπει να ελέγξουμε αν πρέπει να αυξηθεί το επίπεδο, δηλαδή αν ο συνολικός αριθμός των λαγών που έχει σκοτώσει ο χρήστης είναι πολλαπλάσιος του 10. Ουσιαστικά, το επίπεδο αυξάνεται κάθε φορά που ο χρήστης πυροβολεί άλλους 10 λαγούς. Ένας αριθμός είναι πολλαπλάσιος του 10, αν το υπόλοιπο της διαίρεσής του με το 10 έχει αποτέλεσμα 0! Σε αυτή την περίπτωση, θα πρέπει να παίζει ο ήχος WaterDrop, να αυξάνεται η τιμή της μεταβλητής «Επίπεδο» κατά ένα και ανάλογα με το επίπεδο, το αντικείμενο «Λαγός» να μειώνει το μέγεθός του. Όσο το επίπεδο του παιχνιδιού είναι μικρότερο από 5, το μέγεθος του λαγού μειώνεται κατά -5, αλλιώς μειώνεται κατά -1.



Αν συνεχίζαμε την υλοποίηση του παιχνιδιού, αφήνοντας το προηγούμενο σενάριο και παίζαμε το παιχνίδι, πιθανόν να μην αντιλαμβανόμασταν ποτέ ένα σημαντικό λάθος. Η διαίρεση mod του 0 με το 10 έχει και αυτή αποτέλεσμα 0. Συνεπώς, αν δεν πετύχουμε τον πρώτο λαγό με σφαίρα και αυτός έφτανε στο χωράφι, η τιμή της μεταβλητής «Λαγοί» θα παρέμενε 0 και θα άλλαζε το επίπεδο χωρίς να έχουμε πετύχει κάποιον λαγό! Οπότε, πρέπει να τροποποιήσουμε τη συνθήκη ελέγχου, προσθέτοντας άλλη μια πρόταση που πρέπει να είναι αληθής για να εκτελεστεί το μπλοκ εντολών.



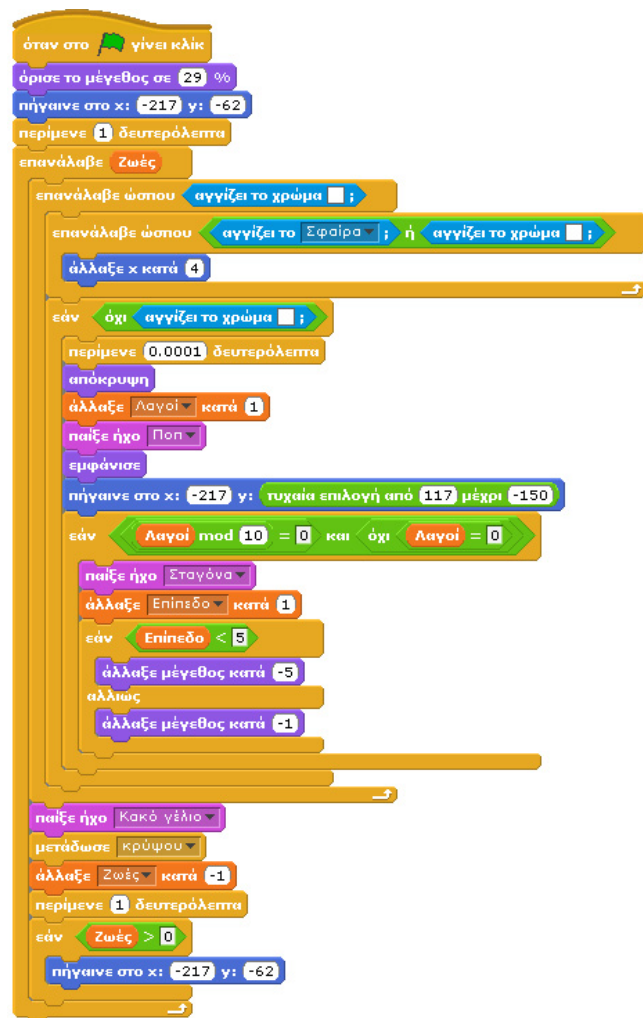
Όλα τα προηγούμενα πρέπει να επαναλαμβάνονται, όσο ο κυνηγός δε καταφέρνει να σκοτώνει το λαγό και όσο ο λαγός δε καταφέρνει να φτάσει στο χωράφι. Αν ο λαγός φτάσει στο χωράφι και αγγίξει το λευκό χρώμα της γραμμής, η επανάληψη θα πρέπει να σταματήσει (προσωρινά). Συμπεριλαμβανόμε, λοιπόν, όλα τα παραπάνω σε μία εντολή επανάλαβε ώσπου.



Η έξοδος από την παραπάνω επανάληψη, δείχνει ότι ο λαγός κατάφερε να φτάσει στο χωράφι. Τότε, σύμφωνα με την περιγραφή του προβήματος, ο χρήστης χάνει μία από τις ζωές του. Έτσι, μειώνεται η τιμή της μεταβλητής «Ζωές» κατά 1, παίζει ο ήχος «σατανικού γέλιου» και μεταδίδουμε μήνυμα «κρύψου», ώστε να το λάβει το αντικείμενο «Ζωές» και να αλλάξει ενδυμασία.

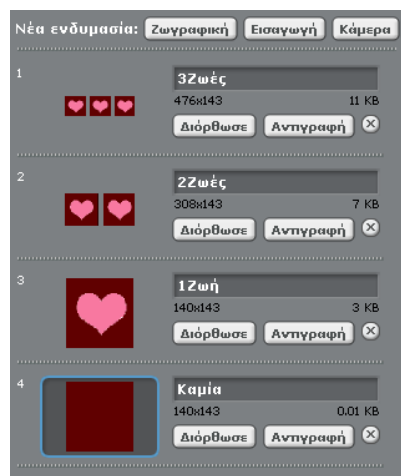


Τελειώνοντας, θέλουμε να έχει τρεις ζωές ο χρήστης, οπότε εσωκλείουμε τον κώδικα που έχουμε περιγράψει ως τώρα σε μια επανάλαβε, όπου στη θέση του αριθμού των επαναλήψεων βάζουμε τη μεταβλητή «Ζωές».



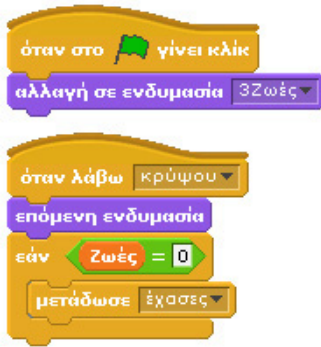
Ζωές

Οι ζωές του παίχτη αναπαρίστανται με καρδιές, οπότε κατεβάζουμε από το διαδίκτυο μια εικόνα καρδιάς και δημιουργούμε τέσσερις ενδυμασίες για το αντικείμενό μας με τον επεξεργαστή ζωγραφικής (τρεις καρδιές, δύο καρδιές, μία καρδιά, καμία καρδιά), προσέχοντας όμως να έχουμε σε όλες τις ενδυμασίες μας χρώμα φόντου ίδιο με το χρώμα του πάνω μέρους του σκηνικού.



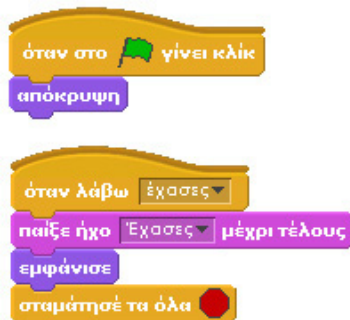
Στην έναρξη του παιχνιδιού θα πρέπει να φαίνονται τρεις καρδιές (όσες και οι ζωές του παίχτη) και θα πρέπει να μειώνονται μέσω αλλαγής ενδυμασίας όταν ο λαγός καταφέρει να αγγίξει τη λευκή γραμμή. Και πάλι η πληροφορία θα έρχεται μέσω μηνύματος από άλλο αντικείμενο («Λαγός»). Επιπλέον,

θα πρέπει το αντικείμενό μας να στέλνει μήνυμα στο αντικείμενο «Τέλος» για να το ενημερώνει ότι οι ζωές τελείωσαν και ο παίχτης έχασε. Το σενάριο είναι ιδιαίτερα απλό.



Αντικείμενο Τέλος

Το πιο εύκολο προς υλοποίηση αντικείμενο είναι το αντικείμενο «Τέλος». Αυτό είναι κρυμμένο στην έναρξη του παιχνιδιού και εμφανίζεται, παίζοντας τον ήχο «Έχασες», όταν ο κυνηγός χάσει και τις τρεις ζωές του. Για να είναι σε θέση να αντιδρά όταν τελειώσουν οι ζωές του κυνηγού, θα πρέπει να λαμβάνει το μήνυμα «έχασες» που στέλνει το αντικείμενο «Ζωές». Θα χρησιμοποιήσουμε συνεπώς την εντολή όταν λάβω... συμπληρώνοντας όνομα μηνύματος ίδιο με αυτό που θα συμπληρώσουμε στην εντολή μετάδωσε στο σενάριο του αντικειμένου «Ζωές». Ας δώσουμε το όνομα «έχασες» στο μήνυμα.



Στην επόμενη εικόνα φαίνεται η οθόνη του παιχνιδιού όταν εμφανιστεί το αντικείμενο «τέλος».



Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Αλλάξτε το έργο ώστε ο κυνηγός να μπορεί να ρίχνει δυο σφαίρες και ταυτόχρονα να εμφανίζονται 2 λαγόι.
- 2) Αλλάξτε το έργο ώστε να παίζεται ταυτόχρονα από 2 χρήστης που θα ελέγχουν 2 διαφορετικούς κυνηγούς.
- 3) Δώστε επιπλέον όπλα στους κυνηγούς (π.χ. μια πέτρα που πέφτει σε συγκεκριμένη θέση - αν τη στιγμή που πέφτει αγγίζει

το λαγό τότε τον σκοτώνει). Επιπλέον προσδιορίστε πιο σύνθετη κίνηση στους λαγούς (π.χ. με δυνατότητα τυχαίας τροχιάς +/- 10 μοίρες από την οριζόντια κίνηση) ώστε να κάνετε το παιχνίδι πιο ενδιαφέρον και δύσκολο.