

ΑΝΑΦΟΡΑ PROJECT C++

Ντανάσης Περικλής 4231

Θυμανάκης Αλέξης 4452

Φάση 1^η

Σε αυτό το σημείο θα προσπαθήσουμε να εντοπίσουμε από την εκφώνηση της άσκησης τις κλάσεις που θα δημιουργήσουμε, τα χαρακτηριστικά τους και την συμπεριφορά τους.

Οι κλάσεις θα είναι οι εξής 5: Organisms, Animals, Plants, Herbivores και Carnivores. Είναι σαφές από την άσκηση ότι οι κλάσεις Herbivores και Carnivores θα είναι υποκλάσεις της Animals καθώς όλα τα φυτοφάγα και τα σαρκοφάγα είναι και ζώα ενώ οι κλάσεις Animals και Plants θα είναι με την σειρά τους υποκλάσεις της Organisms αφού τόσο τα ζώα όσο και τα φυτά είναι οργανισμοί. Ας δούμε τώρα κάθε κλάση ξεχωριστά ξεκινώντας από την Organisms.

ΚΛΑΣΣΗ ORGANISMS:

Τα βασικά χαρακτηριστικά που θα κρατάει η κλάση αυτή είναι αυτά που θα υπάρχουν για όλους οργανισμούς του οικοσυστήματος. Βάση του πρώτου προγραμματισμού αυτά τα χαρακτηριστικά φαίνεται να είναι τα εξής:

- Ηλικία (age): η οποία θα είναι τύπου int και θα μετριέται σε εβδομάδες δηλαδή η τιμή της θα αυξάνεται κατά 1 μετά το πέρας μιας εβδομάδας και αρχικά θα είναι 0.

- Μέγεθος (size): θα είναι και αυτό τύπου int, θα δίνεται σε αυτό μια αρχική τιμή ανάλογα με τον τύπο του οργανισμού και θα αλλάζει ανάλογα με το αν έφαγε ή όχι ο οργανισμός την απαιτούμενη ποσότητα τροφής.

- Βαθμός ανάπτυξης (growth rate): θα είναι τύπου int και θα είναι ένας σταθερός αριθμός που θα δίνεται κατά την δημιουργία του οργανισμού, δεν θα αλλάζει τιμή κατά την διάρκεια της προσομοίωσης και θα καθορίζει το πώς θα μεταβάλλεται το μέγεθος του οργανισμού.

- Κωδικός οργανισμού (orgID): θα είναι ένας int που θα είναι μοναδικός για κάθε οργανισμό. Δεν θα δίνεται από τον χρήστη αλλά θα δίνεται αυτόματα από το σύστημα και θα χρησιμοποιείται ως ταυτότητα για κάθε οργανισμό. Κυρίως θα χρησιμοποιείται για να προσδιορίζεται η θέση του οργανισμού στον χάρτη.

- Θέση του οργανισμού (posX, posY): θα αποτελείται από 2 συνιστώσες, την X και την Y, οι οποίες θα δηλώνουν το ύψος και το πλάτος που βρίσκεται ο οργανισμός στον χάρτη.

Όσο αναφορά τις μεθόδους που θα περιέχει η κλάση αυτή:

Μέθοδοι για την ανάκτηση πληροφοριών οργανισμού (get methods):

- int getAge(); επιστρέφει την ηλικία
- int getSize(); το μέγεθος
- int getGrowthRate(); τον βαθμό ανάπτυξης
- int getOrgID(); τον κωδικό οργανισμού
- int getPosX(); την συνιστώσα θέσης X
- int getPosY(); την συνιστώσα θέσης Y.

Επίσης θα υπάρχουν οι αντίστοιχες μέθοδοι για την αλλαγή των τιμών των παραπάνω μεταβλητών (set methods):

- void setAge(int); δίνει τιμή στην ηλικία
- void setSize (int) ; δίνει μια τιμή στο μέγεθος
- void setPosition(int , int); δίνει ταυτόχρονα τιμές στα posX και posY

Ο βαθμός ανάπτυξης και ο κωδικός οργανισμού δεν έχουν set μέθοδο γιατί πρέπει καθόλη την διάρκεια της προσομοίωσης να έχουν σταθερή τιμή.

Τέλος ο constructor της κλάσης θα είναι:

```
Organism::Organism(int sizeValue,int growthRateValue,int  
orgIDValue):size(sizeValue),growthRate(growthRateValue),orgID(orgID  
Value) {  
    age=0; }
```

δηλαδή θα δίνονται τιμές στο μέγεθος, τον βαθμό ανάπτυξης και τον κωδικό και η ηλικία θα αρχικοποιείται 0.

Ο destructor της κλάσης θα είναι:

```
Organism::~~Organism() { size=0; growthRate=0; orgID=0; age=0;  
posX=0; posY=0; }
```

δηλαδή θα μηδενίζονται όλα τα χαρακτηριστικά του αντικειμένου της κλάσης.

ΚΛΑΣΣΗ PLANTS:

Όπως ήδη αναφέραμε η κλάση αυτή θα κληρονομεί την Organisms.

Από την περιγραφή της εργασίας γίνεται σαφές ότι η κλάση αυτή δεν έχει επιπλέον χαρακτηριστικά εκτός από αυτά που κληρονομεί από την Organisms. Θα προσθέσουμε όμως ένα χαρακτηριστικό για να διακρίνουμε τα φυτά από τους υπόλοιπους οργανισμούς. Αυτή η διάκριση θα γίνεται με μια boolean μεταβλητή την isPlant που θα γίνεται true κατά την δημιουργία ενός αντικειμένου τύπου Plant. Έτσι ο constructor της Plant θα είναι ο εξής:

```
Plant::Plant(int sizeValue,int growthRateValue,int orgIDValue)
: Organism(sizeValue,growthRateValue,orgIDValue) {
    isPlant = true; }
```

δηλαδή θα παίρνει τιμές για το μέγεθος, τον βαθμό ανάπτυξης και τον κωδικό από τον constructor της Organisms και επιπρόσθετα θα δίνει την τιμή true στην isPlant.

Ενώ ο destructor θα είναι ο εξής:

```
Plant::~~Plant() { isPlant=false; }
```

Μία επιπλέον μέθοδος που θα έχει η κλάση αυτή είναι η

- bool getIsPlant(); η οποία θα επιστρέφει το αν ένα αντικείμενο είναι φυτό ή όχι.

ΚΛΑΣΣΗ ANIMLAS:

Θα είναι και αυτή υποκλάση της Organisms και έτσι θα κληρονομεί όλα τα χαρακτηριστικά της ενώ θα προσθέτει και τα εξής:

-Ποσότητα τροφής ανα εβδομάδα (foodPerWeek): θα είναι τύπου int θα δίνεται κατά την δημιουργία του ζώου ανάλογα με το τι ζώο είναι και θα παραμένει σταθερό κατά την διάρκεια της εξομοίωσης. Καθορίζει ποια είναι η απαραίτητη ποσότητα τροφής που πρέπει να καταναλώσει ένα ζώο κατά την διάρκεια της εβδομάδας για να αυξήσει το μέγεθος του (να επιβιώσει).

-Ποσότητα τροφής που έχει καταναλωθεί μέχρι στιγμής (eatenThisWeek) Θα είναι τύπου int, θα μηδενίζεται στην αρχή κάθε εβδομάδας και θα αυξάνεται μέσα στην εβδομάδα κάθε φορά που το ζώο τρέφεται. Στο τέλος της εβδομάδας συγκρίνεται με το foodPerWeek και ανάλογα με το αν είναι μικρότερο (ανεπαρκές) ή μεγαλύτερο (επαρκές) υπολογίζεται το νέο μέγεθος του ζώου.

-Αριθμός κινήσεων (moves): επίσης τύπου int, θα δίνεται κατά την δημιουργία του οργανισμού, θα διαφέρει από ζώο σε ζώο και θα καθορίζει τον αριθμό των κινήσεων που θα μπορεί να κάνει το ζώο κάθε μέρα.

Επίσης η κλάση αυτή θα περιλαμβάνει τις εξής μεθόδους:

-int getFoodPerWeek(); που θα επιστρέφει το foodPerWeek

-int getEatenThisWeek (); που θα επιστρέφει το eatenThisWeek

-int getMoves(); που θα επιστρέφει το moves

-void setEatenThisWeek (int); που θα δίνει μία τιμή στο eatenThisWeek

Ο constructor θα είναι ο εξής:

```
Animal::Animal(int sizeValue,int growthRateValue,int orgIDValue,int
foodPerWeekValue,int movesValue)
```

```
:Organism(sizeValue,growthRateValue,orgIDValue) {  
    foodPerWeek=foodPerWeekValue;  
    moves=movesValue;  
    eatenThisWeek=0; }
```

και ο destructor:

```
Animal::~~Animal() { foodPerWeek=0; eatenThisWeek=0; moves=0; }
```

ΚΛΑΣΣΗ CARNIVORES:

Η κλάση αυτή θα κληρονομεί την κλάση Animals και θα αναφέρεται στα σαρκοφάγα ζώα. Δεν έχει επιπρόσθετα χαρακτηριστικά εκτός από μία boolean μεταβλητή που θα καθορίζει το αν είναι σαρκοφάγο ή όχι. Αυτή θα είναι η: bool isCarn.

Έτσι θα προσθέσουμε και την ανάλογη get μέθοδο που θα επιστρέφει αν το αν το ζώο είναι σαρκοφάγο ή όχι:

```
-bool getIsCarn( );
```

Οι constructor και destructor αντίστοιχα:

```
Carnivore::Carnivore(int sizeValue,int growthRateValue,int  
orgIDValue,int foodPerWeekValue,int movesValue)
```

```
:Animal(sizeValue,growthRateValue,orgIDValue,foodPerWeekValue,mo  
vesValue) {  
    carn=true; }
```

και:

```
Carnivore::~~Carnivore() { carn=false; }
```

ΚΛΑΣΣΗ HERBIVORES:

Όμοια με την Carnivores, η κλάση αυτή θα κληρονομεί την Animals και θα έχει το χαρακτηριστικό bool isHerb και την μέθοδο bool getIsHerb(); για να διακρίνουμε τα φυτοφάγα από τα υπόλοιπα ζώα.

Ο constructor και destructor αντίστοιχα:

```
Herbivore::Herbivore(int sizeValue,int growthRateValue,int  
orgIDValue,int foodPerWeekValue,int movesValue)
```

```
:Animal(sizeValue,growthRateValue,orgIDValue,foodPerWeekValue,mo  
vesValue) {  
    herb=true; }
```

και:

```
Herbivore::~~Herbivore() { herb=false; }
```

Λίγα λόγια για την υλοποίηση:

Θα αναφερθούμε εδώ σε κάποιες σκέψεις σχετικά με το πώς θα υλοποιήσουμε τα βασικά στοιχεία της εφαρμογής. Φυσικά η τελική

υλοποίηση πιθανότατα θα διαφέρει από αυτόν τον αρχικό σχεδιασμό ανάλογα με τα δεδομένα που θα προκύψουν στην πορεία της υλοποίησης. Ο χάρτης: Θα είναι δισδιάστατος πίνακας όπου κάθε διάσταση θα δίνει και μια γεωγραφική παράμετρο. Σε κάθε κελί του πίνακα θα υπάρχει μια τιμή που θα καθορίζει το τι υπάρχει στην θέση αυτή. Αν η θέση είναι κενή τότε η τιμή του αντίστοιχου κελιού του πίνακα θα είναι 0. Αν υπάρχει κάποιος οργανισμός στην θέση αυτή τότε ο πίνακας θα έχει ως τιμή τον κωδικό του εν λόγω οργανισμού. Προφανώς αυτό σημαίνει ότι κανένας οργανισμός δεν θα πρέπει να έχει κωδικό 0, το οποίο φυσικά θα προβλέπεται από την εφαρμογή.

Κινήσεις: Ένας οργανισμός θα μπορεί να κινείται σε μία από τις 8 γειτονικές θέσεις. Θα ελέγχεται ποιες από αυτές τις θέσεις είναι κενές (οι αντίστοιχες τιμές του πίνακα-χάρτη θα είναι 0) και θα διαλέγεται μια τυχαία. Η τιμή του χάρτη από 0 παίρνει τον κωδικό του οργανισμού ενώ η τιμή της θέσης που ήταν αρχικά ο οργανισμός γίνεται 0 (δηλώνεται δηλαδή ως κενή).

Επιθέσεις: Όταν ένα ζώο κάνει μία κίνηση τότε ελέγχονται όλες οι γειτονικές θέσεις της νέας του θέσης και αν βρεθεί σε κάποια από αυτές άλλος οργανισμός τότε ελέγχεται 'αν και ποιος τρώει ποιόν'.

Φάση 2

Οι κλάσεις και ο τελικός σχεδιασμός τους έγιναν σύμφωνα με τον σχεδιασμό της φάσης 1 χωρίς αποκλίσεις.

Εδώ θα αναλύσουμε την εφαρμογή λεπτομερειακά. Θα ακολουθήσει και οδηγός χρήστη (user's manual) με λεπτομέρειες για την λειτουργία της εφαρμογής.

Αρχικά θα αναφέρουμε τα στοιχεία τα οποία χρησιμοποιούμε για την αποθήκευση των δεδομένων της εφαρμογής. Όπως είχε προβλεφθεί στην φάση 1 ο χάρτης αποτελεί έναν δισδιάστατο πίνακα `Map[10][10]` με τιμές σε κάθε κελί τον κωδικό του οργανισμού (`orgID`) που βρίσκεται στην αντίστοιχη θέση ή το 0 αν η θέση αυτή είναι κενή. Οι διαστάσεις του χάρτη είναι 10x10.

Οι οργανισμοί αποθηκεύονται σε 3 διαφορετικούς πίνακες τύπου `Plant`, `Carnivore` και `Herbivore` αντίστοιχα. Ο `A[100]` χρησιμοποιείται για την αποθήκευση των φυτών, ο `B[100]` για την αποθήκευση των σαρκοφάγων και ο `C[100]` για την αποθήκευση των φυτοφάγων. Δύο ακόμα πίνακες, ο `carnMovesLeft[100]` και ο `herbMovesLeft[100]`, χρησιμοποιούνται για την αποθήκευση των κινήσεων των ζώων. Η χρήση τους θα γίνει κατανοητή αργότερα κατά την περιγραφή της λειτουργίας προσομοίωσης.

Τώρα θα αναφερθούμε στις λειτουργίες της εφαρμογής μας που είναι οι εξής:

Λειτουργία δημιουργίας οργανισμού

Σε αυτή την λειτουργία ο χρήστης μπορεί να επιλέξει έναν οργανισμό από την λίστα και να τον βάλει στον χάρτη. Αρχικά καλείται η συνάρτηση `printList()` που εκτυπώνει την λίστα με τους οργανισμούς και τα χαρακτηριστικά τους (9 οργανισμοί σύνολο). Στην συνέχεια ζητείται από τον χρήστη να δώσει μία επιλογή. Αν η επιλογή ήταν έγκυρη ζητείται από τον χρήστη να δώσει μία θέση για τον οργανισμό στον χάρτη ή να δοθεί αν το επιθυμεί τυχαία θέση. Αν ο χρήστης δώσει θέση στην οποία υπάρχει κάποιος άλλος οργανισμός ενημερώνεται γι' αυτό και του ζητείται να δώσει άλλη θέση. Η επιλογή τυχαίας θέσης γίνεται με την `rand()%10`. Αν η `rand` δώσει κατηλημμένη θέση την ξανακαλούμε μέχρι να έχουμε έγκυρες τιμές. Αφού λοιπόν έχει προσδιοριστεί η θέση καλούμε τον κατάλληλο `constructor` ανάλογα με την επιλογή του χρήστη και δημιουργούμε τον οργανισμό βάζοντας τον στον κατάλληλο πίνακα (A , B ή C). Επίσης αυξάνουμε κατά 1 την τιμή ορσιμένων μετρητών που μετρούν το πλήθος των φυτών, των φυτοφάγων, των σαρκοφάγων αλλά και του συνόλου των οργανισμών.

Λειτουργία προσομοίωσης

Αρχικά ο χρήστης καλείται να επιλέξει εάν θέλει να αρχίσει/συνεχίσει την εξομοίωση ή αν θέλει να την αρχίσει από την αρχή. Η 2^η επιλογή προχωρά στην κλήση του `destructor` για κάθε οργανισμό που βρίσκεται αυτή την στιγμή στον χάρτη, μηδενίζει όλους τους μετρητές και αδειάζει τον χάρτη δηλαδή βάζει 0 σε όλα τα κελιά του. Στην συνέχεια επιστρέφει στο `menu` επιλογής των λειτουργιών. Αν επιλεγεί αρχή/συνέχιση της προσομοίωσης μπαίνουμε σε ένα `endless loop` έτσι ότι γίνεται να συνεχίζεται επ'άπειρον, εκτός αν ο χρήστης επιλέξει διαφορετικά. Αρχικά ελέγχεται αν πρόκειται για συνέχιση ή αρχή της προσομοίωσης με την μεταβλητή `notStarted`. Αν πρόκειται για αρχή τότε μηδενίζεται το `eatenThisWeek` κάθε οργανισμού και αρχικοποιούνται οι πίνακες `carnMovesLeft` και `herbMovesLeft` με τον αριθμό κινήσεων κάθε ζώου. Μετά ξεκινά ένα `loop` που μετρά τις ημέρες δηλαδή κάνει συνολικά 7 κύκλους. Μέσα σε αυτό το `loop` ξεκινά ένα άλλο που κάνει τόσες επαναλήψεις όσες ο μέγιστος αροθμός κινήσεων ανάμεσα στα ζώα δηλαδή στην περίπτωση μας 5. Στο σημείο αυτό ξεκινά μια `if` η οποία επαληθεύεται όταν έχει πατηθεί κάποιο κουμπί από τον χρήστη, μέσω της `kbhit()`, και αποτελεί τον μόνο τρόπο εξόδου από την λειτουργία προσομοίωσης (σπάει το `endless loop`). Μέσα σε αυτήν την `if` ο χρήστης καλείται να επιλέξει αν θα επιστρέψει στο `menu` ή θα συνεχίσει την προσομοίωση. Μετά για κάθε σαρκοφάγο υπολογίζεται η επόμενη κίνηση του (εκτός αν η αντίστοιχη τιμή του πίνακα `carnMovesLeft` είναι 0 που σημαίνει ότι δεν του έχουν μείνει άλλες κινήσεις). Η επόμενη θέση

υπολογίζεται με την `rand()%3-1` η οποία υπολογίζει τυχαία έναν αριθμό από το σύνολο `-1,0,1` και τον προσθέτει στην `posX` του ζώου. Έτσι το νέο `posX` θα είναι ίδιο με το προηγούμενο ή κατά 1 μεγαλύτερο ή κατά 1 μικρότερο. Το ίδιο γίνεται και με το `posY`. Έτσι υπολογίζεται ως νέα θέση μία από τις 8 γειτονικές θέσεις. Μόλις πάει στην νέα θέση ελέγχεται αν υπάρχει κάποιος άλλος οργανισμός στις γειτονικές θέσεις και αν πληρούνται οι προϋποθέσεις για να γίνει κάποια επίθεση. Αν κάποιος οργανισμός πεθάνει καλείται ο κατάλληλος destructor και ενημερώνονται κατάλληλα οι μετρητές. Όλα τα παραπάνω γίνονται στην συνέχεια και για κάθε φυτοφάγο ζώο. Τα παραπάνω επαναλαμβάνονται μέχρι να τελειώσουν όλες οι κινήσεις των ζώων και έχει δηλαδή περάσει μια ημέρα. Τότε ξανααρχικοποιούνται οι πίνακες `carnMovesLeft` και `herbMovesLeft` και οι γίνονται οι κινήσεις της επόμενης ημέρας. Αυτό γίνεται μέχρι να τελειώσει μια βδομάδα. Τότε υπολογίζεται το αν μεγαλώνει η μικραίνει το μέγεθος κάθε οργανισμού ανάλογα με το πόσο έφαγε την εβδομάδα αυτή, αυξάνεται η ηλικία του και τέλος πριν αρχίσει η επόμενη εβδομάδα μηδενίζει το `eatenThisWeek` κάθε ζώου.

Λειτουργία πληροφορίες ανα πλυθησμό οργανισμών

Ο χρήστης καλείται να διαλέξει μια από τις 3 κατηγορίες: `Plants`, `Carnivores`, `Herbivores`.

Ανάλογα με την επιλογή υπολογίζονται και εμφανίζονται τα ζητούμενα στοιχεία. Ο υπολογισμός τους είναι απλός και δεν χρειάζεται να εξηγηθεί.

Λειτουργία πληροφορίες ανα οργανισμό

Εκτυπώνονται οι κωδικοί οργανισμού (`orgID`) όλων των ζωντανών αυτή τη στιγμή οργανισμών. Ο χρήστης επιλέγει έναν κωδικό και εμφανίζονται τα στοιχεία του οργανισμού αυτού. Συγκεκριμένα μόλις δοθεί το ζητούμενο `orgID` ψάχνονται όλα τα φυτά και εάν ο οργανισμός βρεθεί σε αυτά εμφανίζονται τα στοιχεία του και μια μεταβλητή `found` γίνεται 1. Αν `found=0` δηλαδή δεν βρέθηκε στα φυτά, ψάχνονται τα φυτοφάγα, και ομοίως αν δεν βρεθεί σε αυτά ψάχνονται τα σαρκοφάγα.

ΔΥΣΚΟΛΙΕΣ ΩΣ ΠΡΟΣ ΤΟΝ ΧΡΗΣΤΗ

Μία δυσκολία ως προς τον χρήστη είναι ότι κατά την διάρκεια της εξομοίωσης δεν μπορεί να ξέρει που βρίσκεται το κάθε ζώο και τι επιθέσεις συμβαίνουν και αν κάποιο ζώο πέθανε και πότε. Φυσικά αντίστοιχα μηνύματα θα μπορούσαν να ενημερώνουν τον χρήστη αλλά επιπλέον θα ήταν κουραστικό να τα παρακολουθεί σε περίπτωση μάλιστα που συνέβεναν πολλά μαζί αποφασίστηκε να αποφευχθούν. Προκειμένου να έχει όμως ο χρήστης την αίσθηση του χρόνου ώστε να αποφασίσει πότε πρέπει να κάνει `pause` ή όχι, εμφανίζονται μηνύματα στην αρχή

κάθε μέρας που μας ενημερώνουν για την χρονική στιγμή (εβδομάδα και ημέρα).

Επίσης μια ακόμη δυσλειτουργία είναι το γεγονός ότι η αναφορά σε κάθε οργανισμό γίνεται με τον κωδικό του πράγμα που εμποδίζει την επαφή του χρήστη με το πρόγραμμα.

ΕΠΕΚΤΑΣΕΙΣ

Μία πιθανή επέκταση θα ήταν να δοθεί κάποια νοημοσύνη στα ζώα με την έννοια οι κινήσεις τους να μην είναι τελείως τυχαίες. Δηλαδή αφού κινηθεί κάποιο ζώο να ελέγχει αν υπάρχει κάποιος άλλος οργανισμός γύρω του π.χ σε ακτίνα 2 τετραγώνων-θέσεων του χάρτη και η επόμενη κίνηση να είναι προς την άλλη κατεύθυνση σε περίπτωση κινδύνου ή προς την ίδια σε περίπτωση τροφής.

Φυσικά επεκτάσεις μπορούν να γίνουν σε σχέση με τους διαθέσιμους οργανισμούς προς δημιουργία και τα στατιστικά στοιχεία που θα μπορούν να εμφανίζονται.

USER'S MANUAL

Με την εκκίνηση του προγράμματος βρίσκεστε στο αρχικό menu με τις εξής επιλογές:

- 1.Λειτουργία προσομοίωσης
- 2.Δημιουργία καινούριου οργανισμού
- 3.Πληροφορίες ανα πληθυσμό οργανισμών
- 4.Πλξροφορίες ανα οργανισμό
- 5.Έξοδος

Διαλέκστε πατώντας 1 έως 5.

Σας προτείνουμε να ξεκινήσετε με την δημιουργία νέου οργανισμού.

- 2.Δημιουργία καινούριου οργανισμού:

Θα εμφανιστεί μπροστά σας η λίστα με τους διαθέσιμους 9 οργανισμούς. Πατείστε 1 έως 9 για τον αντίστοιχο οργανισμό ή 10 για επιστροφή στο menu. Αν επιλέξατε 1-9 θα σας ζητηθεί να δώσετε θέση στον οργανισμό ή να δοθεί τυχαία θέση. Πιέστε 1 για να δώσετε θέση, 2 για τυχαία θέση. Αν πίεσατε 1 το σύστημα θα περιμένει δύο αριθμούς από 0 έως 9 που θα είναι η θέση του οργανισμού. Δώστε τους 2 αυτούς αριθμούς. Αν η θέση είναι κατηλλημένη θα ενημερωθείτε κατάλληλα και θα σας ζητηθεί να ξαναδώσετε θέση. Μόλις δοθεί επιτυχώς θέση, επιστρέφετε στο αρχικό μενού.

- 1.Λειτουργία προσομοίωσης:

Πιέστε 1 για Αρχή/Συνέχεια ή 2 για Επανεκκίνηση.

Επανεκίνηση: Διαγράφει όλους τους οργανισμούς από το χάρτη και επιστρέφει στο menu.

Αρχή/Συνέχεια: Ξεκινά την προσομοίωση. Σας ενημερώνει με το πέρασμα κάθε μέρα για να έχετε την αίσθηση του χρόνου. Μπορείτε να διακόψετε την ροή πατώντας οποιοδήποτε κουμπί. Μόλις το κάνετε σας ρώτα αν θέλετα να συνεχίσετε την προσομοίωση ή να κάνετε πάυση και να επιστρέψετε στο μενού. Πατήστε n για το 1^ο ή y για το 2^ο.

ΠΡΟΣΟΧΗ: Το κουπί που πατήσατε για να σταματήσετε την ροή της προσομοίωσης συμμετέχει στην επιλογή του y ή n. Για το λόγο αυτό πατήστε backspace πριν το y ή n.

3.Πληροφορίες ανα πληθυσμό οργανισμών:

Θα σας ζητηθεί να επιλέξετε ανάμεσα στους 3 τύπους οργανισμών που υπάρχουν δηλαδή 1.Herbivores , 2.Carnivores , 3.Plants.

Πιέστε 1-3 για επιλογή και θα εμφανιστούν τα στατιστικά στοιχεία για την κατηγορία που επιλέξατε. Στη συνέχεια θα μεταφερθείτε στο αρχικό μενού.

4.Πληροφορίες ανα οργανισμό:

Εκτυπώνει τους κωδικούς όλων των εν ζωή οργανισμών ταξινομημένους κατά τύπο (Plants, Herbivores, Carnivores). Διαλέξτε ποιόν οργανισμό θέλετε και δώστε τον κωδικό του. Θα εμφανιστούν τα στοιχεία του οργανισμού και θα οδηγηθείτε στο κεντρικό μενού.

5.Έξοδος:

Τερματίζει την εφαρμογή.