

ΕΡΓΑΣΙΑ #5 (Project Μαθήματος)

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

A.M. : 1115201200106

Όνομα : Βασίλειος

Επίθετο : Μαυρομμάτης

ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Η εργασία περιέχει τα **14** παρακάτω **αρχεία** :

1 αρχείο πηγαίου κώδικα **EcoSim.cpp** το οποίο περιέχει την **main** του προγράμματος και υλοποιεί την προσομοίωση ενός οικοσυστήματος στο **επίπεδο 0**.

1 αρχείο πηγαίου κώδικα **Map.cpp** και το αντίστοιχο **1** αρχείο επικεφαλίδας

Map.h , τα οποία υλοποιούν τις απαραίτητες κλάσεις καθώς και τις

αντίστοιχες συναρτήσεις για την δημιουργία του “χάρτη” του οικοσυστήματος την τοποθέτηση των ζώων και φυτών , και αποτελεί ουσιαστικά το interface **1ου επιπέδου** μεταξύ των παρακάτω αρχείων, για να προσομοιωθεί η λειτουργία του.

1 αρχείο πηγαίου κώδικα **Date.cpp** και το αντίστοιχο **1** αρχείο επικεφαλίδας **Date.h** , τα οποία στο **επίπεδο 2**, υλοποιούν τα Data members που παρέχουν την έννοια του χρόνου στην προσομοίωση.

1 αρχείο πηγαίου κώδικα **Tile.cpp** και το αντίστοιχο **1** αρχείο επικεφαλίδας **Tile.h** , τα οποία στο **επίπεδο 2**, διαχωρίζουν τον χάρτη σε οντότητες Tiles τα οποία προσομοιώνουν περιοχές του οικοσυστήματος, διαθέτουν ένα μόνο κυρίαρχο φυτό και ομαδοποιούν τα ζώα σε λίστες.

1 αρχείο πηγαίου κώδικα **Plant.cpp** και το αντίστοιχο αρχείο επικεφαλίδας **Plant.h** , τα οποία υλοποιούνται στο **επίπεδο 3** και ορίζουν τα φυτά τις συναρτήσεις και την συμπεριφορά τους, με γονεική κλάση την Plant και τις αντίστοιχες θυγατρικές Seeded και Seedless , που αποτελούν data members των Tiles **επιπέδου 2**.

1 αρχείο πηγαίου κώδικα **List.cpp** και το αντίστοιχο **1** αρχείο επικεφαλίδας **List.h** , τα οποία στο **επίπεδο 3**, υλοποιούν το interface μεταξύ των animals και των Tiles **επιπέδου 2** ως μονά συνδεδεμένες με κεφαλή λίστες σε στύλ στοίβας και τις απαραίτητες λειτουργίες προσθήκης, διαγραφής, διαφορετικών αναζητήσεων , και ενημέρωσης κόμβων, όπου ο κάθε κόμβος αντιπροσωπεύει ένα animal.

1 αρχείο πηγαίου κώδικα **Animal.cpp** και το αντίστοιχο αρχείο επικεφαλίδας **Animal.h** , τα οποία υλοποιούνται στο χαμηλότερο **επίπεδο 4** και ορίζουν τα ζώα τις συναρτήσεις και την συμπεριφορά τους, με γονεική κλάση την Animal και τις αντίστοιχες θυγατρικές Herbivore και Carnivore , και αποτελούν τα data members των κόμβων του **επιπέδου 3**.

1 αρχείο **Makefile** για την δημιουργία του εκτελέσιμου “projectEco”.

Αρχικά η **main** αρχικοποιεί, μέσω των 3 ορισμάτων της γραμμής εντολών τις διαστάσεις του χάρτη , την αρχική εποχή και τις συνολικές μέρες για τις οποίες θα τρέξει η προσομοίωση.

Στην συνέχεια καλεί την generateMap() για να δημιουργήσει έναν τυχαίο τετράγωνο χάρτη μέσω των generateRiver(), generateLake(), generateHills() και τέλος generateMeadow() οι οποίες δημιουργούν ένα ποτάμι τυχαίας ολίσθησης, μια τετράγωνη λίμνη πεπερασμένων τυχαίου εύρους διαστάσεων, έναν τυχαίο αριθμό λόφων που καλύπτουν από 2 έως 4 tiles, και να γεμίσουν με πεδιάδα ότι απέμεινε.

Έπειτα γίνεται τοποθέτηση των φυτών με την placePlants() το πλήθος των οποίων καθορίζεται μόνο ως προς το πάνω όριο ως ποσοστό των υπάρχοντων τύπων tiles, σε τυχαίες αλλά επιτρεπτές θέσεις με βάση τον τύπο του φυτού και ένα φυτό ανα tile. Αντίστοιχα γίνεται η τοποθέτηση των ζώων με τις διαφορές ότι το μέγιστο πλήθος τους καθορίζεται συναρτήσει και του μεγέθους του χάρτη ενώ μπορούν να τοποθετηθούν περισσότερα από ένα animals στην ίδια λίστα του tile.

Μετά απο την αρχικοποίηση του χάρτη τυπώνονται μέσω των `printSystem()`, `printPlantStatistics()` και `printAnimalStatistics()` η μορφολογία του χάρτη και το αντίστοιχο φυτό του πλέγματος καθώς και το πλήθος αλλά και το συνολικό πλήθος των διαφορετικών ειδών φυτών και ζώων.

Τέλος ξεκινάει η προσομοίωση για το απαραίτητο πλήθος ημερών που θέλει ο χρήστης με την κλήση της `runEcosystem()` ενώ οι ίδιες πληροφορίες εκτυπώνονται στο τέλος της προσομοίωσης αλλά και κατά την αλλαγή εποχής κάθε 90 μέρες αν προκύψει. Στο σώμα της `runEcosystem()` πραγματοποιούνται οι εξής λειτουργίες με την ακόλουθη σειρά (συνοπτικά γιατί εδώ καλούνται πάνω από 50 διαφορετικές συναρτήσεις για πρόσβαση στο κάθε επίπεδο) :

1) Ανανέωση της μέρας , αρχικοποίηση πείνας και ανάπτυξη φυτών και ζώων αν είναι μέρα ανάπτυξης. Τα φυτά είτε αναπτύσσονται είτε εξαιτίας αρρώστειας εξασθενούν και χάνουν μέγεθος φύλλα και καρπούς αν έχουν.

2)Κάθε ώρα της ημέρας , τα ζώα ψάχνουν για τροφή στα cords που βρίσκονται στο ίδιο tile με αυτά και αν δεν βρουν μετακινούνται. Για κάθε επιτυχημένο γεύμα είτε φυτού είτε ζώου σαρώνεται όλος ο χάρτης και αφαιρούνται νεκρές οντότητες.

3)Στο τέλος της ημέρας αν κάποιο ζώο δεν έχει τραφεί αρκετά αυξάνει τον μετρητή της πείνας του και αν ξεπεράσει τις 10 μέρες νηστικό πεθαίνει ενώ αν είναι σαρκοφάγο και ξεπεράσει τις 7 μέρες, την 8η και μέχρι να πεθάνει αν δεν τα καταφέρει κυνηγάει και σαρκοφάγα.

4)Ζώα και φυτά αναπαράγονται συγκεκριμένες μέρες και με βάση τα attributes που αλλάζουν κάθε 90 ημέρες με την αλλαγή εποχής αν προκύψει.

Ο κύκλος αυτός επαναλαμβάνεται μέχρι να διανυθούν όλες οι μέρες προσομοίωσης.

ΠΑΡΑΤΗΡΗΣΕΙΣ - ΠΑΡΑΔΟΧΕΣ

1) Ο καννιβαλισμός δεν επιτρέπεται στα σαρκοφάγα που έχουν να φάνε μέρες.

2) Σε περίπτωση ισοδυναμίας μεταξύ 2 σαρκοφάγων δεν γίνεται τίποτα.

3) Φυτά που μόλις έχουν γεννηθεί δεν επιτρέπεται να αναπαραχθούν ξανά μέχρι και την επόμενη μέρα αναπαραγωγής.

4) Δεν επιτρέπεται τοποθέτηση άλλου τύπου tile επάνω από ήδη δημιουργημένο tile απο προηγούμενη `generate()`

5) Ζώα που πέφτουν σε χειμερία νάρκη τον Χειμώνα δεν μετέχουν σε καμία διαδικασία ανάπτυξης , αναπαραγωγής , κίνησης ή τροφής, δεν μπορούν να κηρυγθουν από άλλα, δεν πεινάνε , δεν αυξάνεται ο μετρητής πείνας τους, γενικά όλα τους τα στοιχεία μένουν στάσιμα στην κατάσταση πριν και την τελευταία μέρα του φθινοπώρου (θεωρώ οτι κρύβονται-κινούνται), και επανακκινούνται με τον ερχομό της Άνοιξης.

6)Την πρώτη ώρα της πρώτης μέρας τα ζώα πρώτα θα ψάξουν τροφή στο tile που είχαν το αρχικό spawn και έπειτα θα κινηθούν, και εαν είναι φυτοφάγα δεν φεύγουν από το tile που βρίσκονται όσο έχει φαγητό ανεξάρτητα απ'το αν πεινάνε ή όχι, για να δώσω μια πιο ρεαλιστική συμπεριφορά.

7)Αρκούδα = instant kill ανεξάρτητου speed όταν το θύμα είναι herbivore δηλαδή ελέγχω μόνο αν `victimSize <= bearSize` και αν ναι η αρκούδα τρώει, ενώ για τα carnivores κανονικά πρώτα με βάση size και μετά στην ισότητα τα stats έτσι ώστε μια ενήλικη ναι μεν μπορεί να σκοτώσει οποιοδήποτε ζώο αλλά μία ανήλικη δεν θα τα καταφέρει ενάντια σε έναν ενήλικο λύκο πχ.

ΔΗΜΙΟΥΡΓΕΙΑ ΕΚΤΕΛΕΣΙΜΟΥ ΚΑΙ ΕΚΤΕΛΕΣΗ

\$ make

\$./projectEco <(int) terrainSize> <(string) startingEpoch> <(int) totalDays>

(3 args διαχωρισμένα με κενό)

\$ make clean

ΥΛΟΠΟΙΗΘΗΚΕ ΣΕ:

Language: C++

OS: Ubuntu 18.04

Text Editor: Sublime Text 3

Debugger: Valgrind 3.13.0