Doamnelor si domnilor,

Buna ziua. Numele meu este Stefan Neacsu-Miclea, am 14 ani si impreuna cu alti colegi reprezint Colegiul “B.P. Hasdeu”, Buzau, Romania. Alaturi de coechipierul nostrum Aymeric de la ……………………. Va prezentam Subiectul 3 din acest program.

Suntem fericiti si onorati sa ne flam astazi aici, alaturi de dumneavoastra la acest minunat eveniment. Congresul ne-a oferit oportunitatea de a ne impartasi experiente din domeniul matematicii si informaticii, de a va cunoaste mai bine cultura, sistemul educational.

Asa cum arata si posterul nostrum de prezentare, subiectul nostru ne cerea sa desenam un copac prin realizarea unui program informatics care sa genereze o specie vegetala la alegere.

///////////////////////////////////////////////////////////

SUBIECTUL 3 –Deseneaza un copac-

**Puteti scrie un program care sa permita reprezentarea unei specii vegetale la alegere?**

/////////////////////////////////

Un copac poate fi reprezentat in multe moduri, pentru ca

exista extrem de multe moduri de a vedea natura. O prima solutie –poate naiva – a fost aceea sa realizam un program care “deseneaza” copacul, mai exact incarca o serie de imagini si le afiseaza la scurte intervale de timp, creand astfel impresia unei animatii.

/////////////////////////////////////

Prima data ne-am gandit la o astfel de reprezentare:

////////////////////////////////////

Ideea este relativ simpla:

-> Citim imaginile din fisiere text. Acestea vor fi afisate in consola

de la Windows, care in paleta presetata pe 16-bit nu permite decat 8 culori (negru, albastru, verde, rosu, mov, galben, gri, alb) avand doua intensitati: de background si de foreground. Pentru a desena un copac nu ne va trebui decat verde si maro. Metoda de citire si afisare este foarte simpla:

-> Se citeste din fisier caracter cu caracter si, in functie de acesta, se realizeaza modificari in fereastra consolei:

* Pentru “/” se trece pe urmatoarea linie;
* Pentru “u” se coloreaza casuta curenta in verde deschis;
* Pentru “v” se coloreaza casuta curenta in verde inchis;
* Pentru “m” se coloreaza casuta curenta in maro;
* Pentru “0” se sare peste casuta curenta.

->Dupa ce s-a terminat citirea din fisier, se asteapta

un timp (200 ms), se curata ecranul si se trece la fisierul

urmator.

/////////////////////////////////////////////////////////////

Codul realizat in C++ este foarte simplu:

Includem header-ul “fstream” pentru operatiile cu fisiere, “windows.h” si “stdlib” pentru a permite utilizarea culorilor in concola. Vom define cateva functii care ne vor usura munca: pentru setarea culorilor de fundal, pentru curatarea ecranului, pentru a seta cursorul la o anumita pozitie.

/////////////////////////////////////////////////////////////

Citim fisierele text character cu character si in acelasi timp desenam copacul dupa notatiile facute.

///////////////////////////////////////////////////////////

Intreg algoritmul este inclus intr-o bucla infinita pentru a crea impresia unei animatii.

Dupa cum s-a putut vedea, rezultatul este o simpla animatie, dar reprezentarea este una generala, copacul nu poate fi asemanat cu o specie anume, iar consola windows nu permite o calitate ridicata a imaginii, dat fiind ca unitatea de baza a imaginii nu a fost pixelul, ci casuta “buffer” a consolei.

///////////////////////////////////////////////////////

Despre Wingraph

Wingraph este o colectie de date implementata pentru Free Pascal 2.4.0 si a fost creata de Stefan Berinde, un profesor roman de la Universitatea Babes-Bolyai din Cluj, Romania.

Spre deosebire de unit-ul “Graph” care este pre-inclus in kit-ul de intalare Free Pascal, Wingraph mi s-a parut mai usor de folosit si contine multe alte functii noi. Are 255 de culori predefinite, se poate modifica dimeniunile ferestrei cu modul MCustom, are un tip structurat pentru animatii, iar functiile de UpdateGraph(UpdateOn/UpdateOff) permit memorarea si prelucrarea continutului grafic direct in memorie, pentru ca ulterior sa fie afisate pe ecran folosind accesul direct la pixeli.

///////////////////////////////////////////////////

Utilizand Wingraph, am putut folosi ideea Metodei 1 ajutandu-ne de grafice. Imaginile sunt memorate in fisiere Bitmap pe 24-bit, ceea ce ofera imaginii mai multa calitate grafica. Am realizat programul in Lazarus IDE, bazat pe Free Pascal 3.0.0.

Incarcam imaginile in memorie prin tehnica precalcularii pentru a le desena ulterior pe rand tot in ciclul unei bucle infinite.

/////////////////////////////////////////////////////////

Am memorat copacul ca un tip obiect, care contine coordonatele,

adresa imaginii si functii pentru citire si afisare:

///////////////////////////////////////////////////////

Blocul central are la baza aceeasi idee aplicata la metoda 1:

Animatia se va termina cand fereastra grafica este inchisa.

//////////////////////////////////////////////////////

Rezultatul este tot o animatie, dar obtinuta la nivel de pixel. In functie de talentul artistic al fiecaruia, poate fi reprezentat orice tip de copac prin simpla desenare a lui. Aici avem doar un exemplu general:

/////////////////////////////////////////////////////////////////

Metoda 2

Modelele anterioare de areprezentare sunt doar o solutie grafica, dar nu una matematica. Iar cea mai fidela interpretare a realitatii in mod matematic sunt fractalii.

**Fractalul** este "o figură geometrică fragmentată sau frântă care poate fi divizată în părți, astfel încât fiecare dintre acestea să fie (cel puțin aproximativ) o copie miniaturală a întregului". Termenul de fractal a fost creat de matematicianul  Benoît Mandelbrot. Setul Mandelbrot si curba lui Koch sunt cateva exemple renumite de fractali.

Fractalii se gasesc peste tot in natura: in fulgii de nea, in formele de relief, in fulgere si in formarea galaxiilor. Bineinteles, si copacul poate fi considerat un fractal.

////////////////////////////////////////////////////////////////////

Inainte de a trece la algoritmul informatic, a trebuit sa rezolvam cateva probleme de matematica care au intervenit in aceasta rezolvare.

1: Avem de construit un segment (AB) cu A fxat la coordonatele (x1,y1) de lungime l si care formeaza ununghi u cu orizontala. Sau,pe scurt, va trebui sa gasim pozitia punctului B al segmentului. Incadram segmentul (AB) intr-un system de axe ortogonale. Am incercat sa ne adaptam in acelasi timp la configuratia unei ferestre grafice, care are originea O in stanga-sus la coordonatele (0px,0px).

////////////////////////////////////////////////////////////////////////

Comsideram punctul C(X2,Y1). Calculam lungimile laturilor AC si BC in triunghiul dreptunghic ABC:

AC = AB \* cos(u) => AC = l \* cos(u)

BC = AB \* sin(u ) => BC = l \* sin(u )

Din grafic observam ca X1-X2=AC si Y1-Y2=BC, de unde:

X2 = X1 – l \* cos(u) Y2= Y1 – l \* sin(u)

//////////////////////////////////////////////////////////////////

**2. Dat un segment (AB), care face un unghi a cu orizontala, trebuiesa construim un segment (BC) care face un unghi u cu segmentul (AB).**

Altfel spus, trebuie sa determinam unghiul b pe care il face segmentul (BC) cu orizontala pentru a putea aplica formulele obtinute la problema 1.

Rezultatul se obtine simplu, folosind congruenta unghiurilor corespondente determinate de doua paralele si o secanta

(vom alege ca paralele axele orizontale ce trec prin A si prin B,

Iar secanta va fi AB):

In functie de orientarea segmentului (BC) fata de (AB), desprindem doua cazuri:

//////////////////////////////////////////////////////////////////////////////////

Cand (BC) este in stanga lui (AB), unghiul facut de (BC) cu orizontala este diferenta dintre unghiul a si unghiul dintre cele doua drepte.

Cand (BC) este in dreapta lui (AB), unghiul facut de (BC) cu orizontala este suma dintre unghiul a si unghiul dintre cele doua drepte.

////////////////////////////////////////////////////////////////////////////////

Raportul de aur

Este primul numar rational definit si descoperit in instorie. Avand

o valoare aprozimativa de 1,618033, pare sa stea la baza formarii

tuturor lucrurilor din jurul nostru. Il putem gasi in Univers, in monumente, in formarea norilor, la om, in natura. Are o legatura

cu sirul lui Fibonacci, deoarece daca facem raportul a doua numere

consecutive din acest sir obtinem o valoare apropiata de cea a

sectiunii de aur si cu cat numerele sunt mai mari, cu atat valoarea

Raportului tinde spre acest numa irational.

Pentru a reduce timpul de calcul efectuat de calculator, vom alege

primele numere din sirul lui Fibonacci, 3 si 5, cu 5/3 ≈Φ. In rezolvarea

noastra vom avea nevoie de dispunerea crengilor de jos in sus,

dinspre cele mai mari spre cele mai mici, deci vom avea nevoie de

inversul raportului de aur, 3/5.

/////////////////////////////////////////////////////////////

Cu ajutorul acestor formule, putem realiza diverse tipuri de fractali. Initial ne-am ghidat dupa reprezentarea grafica a unui arbore si ne-am imaginat un tip lista. Din fiecare ramura principala a copacului se duc doua sub-ramuri, care in pasii urmatori ai algoritmului devin ramuri principale pentru care se executa aceeasi operatie pana cand lungimea unei ramuri devine 1px. (fractalul se desfasoara la infinit, dar calculatorul are o limita de afisare, deci nu poate fi reprezentat in totalitate).

////////////////////////////////////////////

Codul are la baza un algoritm recursiv. Initial definim un tip de date ce memoreaza coordonatele unui punct si o functie care returneaza coordonatele capatului unui segment de lungime l cu celalalt capat la coordonatele (x,y) si care face un unghi u cu orizontala, aplicand rezultatul problemei 1.

//////////////////////////////////////////////

In Algoritmul prezentat vom randomiza unghiurile, pentru a conferi cat mai mult impresia realitatii. Vom lucre cu pointeri care memoreaza coordonatele punctelor in care se gasesc varfurile ramurilor principale la un moment dat.

//////////////////////////////////////////////

Copacul obtinut este doar o idee a ceea ce ne dorim sa realizam, deoarece vom avea nevoie de o formula mai complexa de a genera arborele.

//////////////////////////////////////////////

Bazandu-ne pe aceasta idee putem realiza un copac de tip feriga. Putem renunta la tipul lista (l-am utilizat pentru a ne crea o impresie despre constructia copacului), dar vom adauga la structura “coordinates”, pe langa coordonatele capatului unui segment, si unghiul pe care il face acesta cu orizontala pentru a putea aplica rezultatul problemei 2 🡪 . Metoda de baza va fi tot recursivitatea.

///////////////////////////////////////////////////

Diferenta intre algoritmul de mai devreme si cel de aici este aceea ca ramurile se vor genera cand in stanga, cand in dreapta, pe anumite noduri ale tulpinii stabilite de formula l/(i%2+1). Putem genera n subramuri pe o ramura pincipala, repetand algoritmul pe fiecare dintre crengile obtinute.

///////////////////////////////////////////////////////

In mod asemanator putem obtine si arborele lui Pitagora, dar care va avea cate doua generari pe fiecare ramura.

//////////////////////////////////////////////////////

Observam ca, daca punem raportul de aur intre lungimile nodurilor unei ramuri, obtinem o structura ce seamana cu stuful.

/////////////////////////////////////////////////////////////////

Pe aceasta idee am incercat sa reprezentam un copac mai special, dupa codul urmator. Daca dublam structura de stuf pe ambele parti ale unei ramuri principale obtinem o specie vegetala care seaman cu stejarul.

/////////////////////////////////////////////////////////////////////

Rezultatul este mult mai satisfacator. Prima structura poate fi asemenate cu un prun, iar cea de-a doua cu un stejar.

////////////////////////////////////////////////////////////////////

Eroarea de calcul a unghiului pentru feriga ne-a condus la reprezentarea grafica a unui brad, pe care il putem decora cum vrem.

////////////////////////////////////////////////////////////////

Prezentam in continuare un program care contine tipurile de copaci realizate cu ajutorul fractalilor si cu functii de culoare si de decorare.

Din partea dreapta putem selecta tipurile copacilor. Primul reprezinta varianta de la care am plecat. Activam culorile si ne putem bucura de privelisti frumoase in fiecare anotimp. [ ]Cel de-al doilea este arborele lui Pitagora.[ ] La copacul feriga putem selecta lungimea si unghiul pentru generare.Un unghi mai mare de 35 de grade va deforma copacul, iar pentru a se incadra in ecran lungimea trebuie sa fie de cel mult 200px. Ultimul copac ar putea fi un mar. Algoritmul de generare se bazeaza pe cel al copacului stuf cu unghiurile usor modificate pentru a coferi impresia realitatii. Dezavantajul acestui algoritm este faptul ca structura seaman cu un arbore natural doar pentru anumite unghiuri (10,15,25), deformandu-se la unghiuri cum ar fi 17, 30. Pentru acest tip de copac am conceput o animatie prin incrementarea si decrementarea unghiului.