



Středoškolská technika 2017

Setkání a prezentace prací středoškolských studentů na ČVUT

PROGRAMY PRO PROSTOROVÉ ZOBRAZOVÁNÍ NĚKTERÝCH TĚLES

Vladimír Štěpnička

**Střední škola a vyšší odborná škola aplikované kybernetiky s. r. o.
Hradecká 1151/9, 500 01 Hradec Králové**

Úvod:

Dlouhodobě se zajímám o počítačovou grafiku a zobrazování spojené s matematikou. Vytvořit prostorové obrazce, které se budou lišit počtem vrcholů a barvami stěn, mě napadlo při hraní hry Minetest, kdy jsem si chtěl vytvořit vlastní mód do hry a potřeboval jsem si udělat textury přidaných itemů. Programy jsem vyvinul v prostředí mousepad, ke kompilaci byl vždy použitý program GCC. Program pro svůj běh potřebuje grafickou knihovnu Glut.

Jako první ze skupiny programů pro vykreslování těles představuji: Program pro zobrazování sférických obrazců

Jedná se o interaktivní program, který na monitoru vykresluje a zobrazuje prostorová tělesa. Funguje na principu umístění určitého počtu částic do prostoru. Na základě odpuzivých a přitažlivých sil program nasimuluje pohyb částic, které se vzájemně odpuzují a současně jsou přitahovány ke středu, tak, že se rozmístí na povrchu koule ve vrcholech tělesa a program vykreslí pravidelné, symetrické a někdy i nepravidelné těleso v závislosti na počtu navolených vrcholů, anebo program vykreslí těleso s vrcholy v těžištích trojúhelníků, tvořených spojnicemi částic.

Program umožňuje přidat nebo ubrat počet částic a s každou změnou vykreslí nové těleso. Program je zpracován jako animace tělesa při pohybu částic v důsledku přitažlivých a odpuzivých sil maximální rychlostí 100 snímků za sekundu, takže je možné zaregistrovat postupné uspořádávání částic. Vykreslené těleso je možné vertikálně i horizontálně otáčet.

Program pracuje ve třech módech

Ukázka zdrojového kódu pro krok simulace:

```
for (a=0; a<n; a++) { for (c=xd=yd=zd=0; c<n; c++) {  
x1=x[a]-x[c];  
y1=y[a]-y[c];  
z1=z[a]-z[c];  
r=x1*x1+y1*y1+z1*z1+0.1;  
xd+=x1/r;  
yd+=y1/r;  
zd+=z1/r;  
}  
x[a]+=xd/(n+3);  
y[a]+=yd/(n+3);  
z[a]+=zd/(n+3);  
x[a]/=1.5;  
y[a]/=1.5;  
z[a]/=1.5;
```

Ovládání programu společně pro všechny 3 módy:

Klávesou + je možné přidávat počet bodů, z nichž se stanou vrcholy tělesa.

Klávesou - je možné ubírat počet bodů, z nichž se stanou vrcholy tělesa

Klávesami 2 a 8 je možné tělesa otáčet horizontálně

Klávesami 4 a 6 je možné tělesa otáčet vertikálně

Mód 0

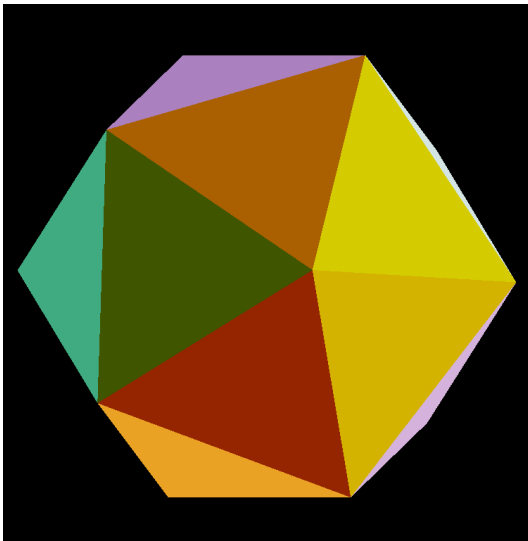
Klávesa D přepne program do módu 0

0. mód

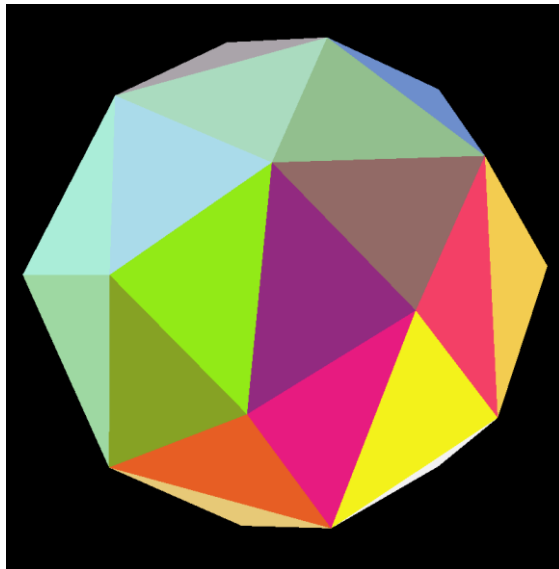
Zobrazuje těleso s vrcholy v bodech, které tvoří částice rozmístěné na povrchu koule na základě odpuzivých a přitažlivých sil
 Vybarví každou plochu jednotnou barvou, rozdílnou od barvy jiných ploch téhož tělesa.
 Barva každé stěny je dána pořadím částic tvořícím vrcholy. Podle pořadí částic uložených v paměti počítače je zobrazována světlost barev. První částice uložená do paměti je nejtmavší a poslední přidaná částice je nejsvětější.

Ukázka zdrojového kódu pro vykreslení tělesa

```
if(!mod){glColor3f(a/(float)n,b/(float)a,c/(float)b);
glVertex3f(x[a],y[a],z[a]);
glVertex3f(x[b],y[b],z[b]);
glVertex3f(x[c],y[c],z[c]);}
```



Obr. 1: Dvacetistěn vykreslený podle módu 0 z různých pohledů



Obr 2: Mnohostěn vytvořený podle módu 0

Mód 1

Klávesa F přepne program do módu 1

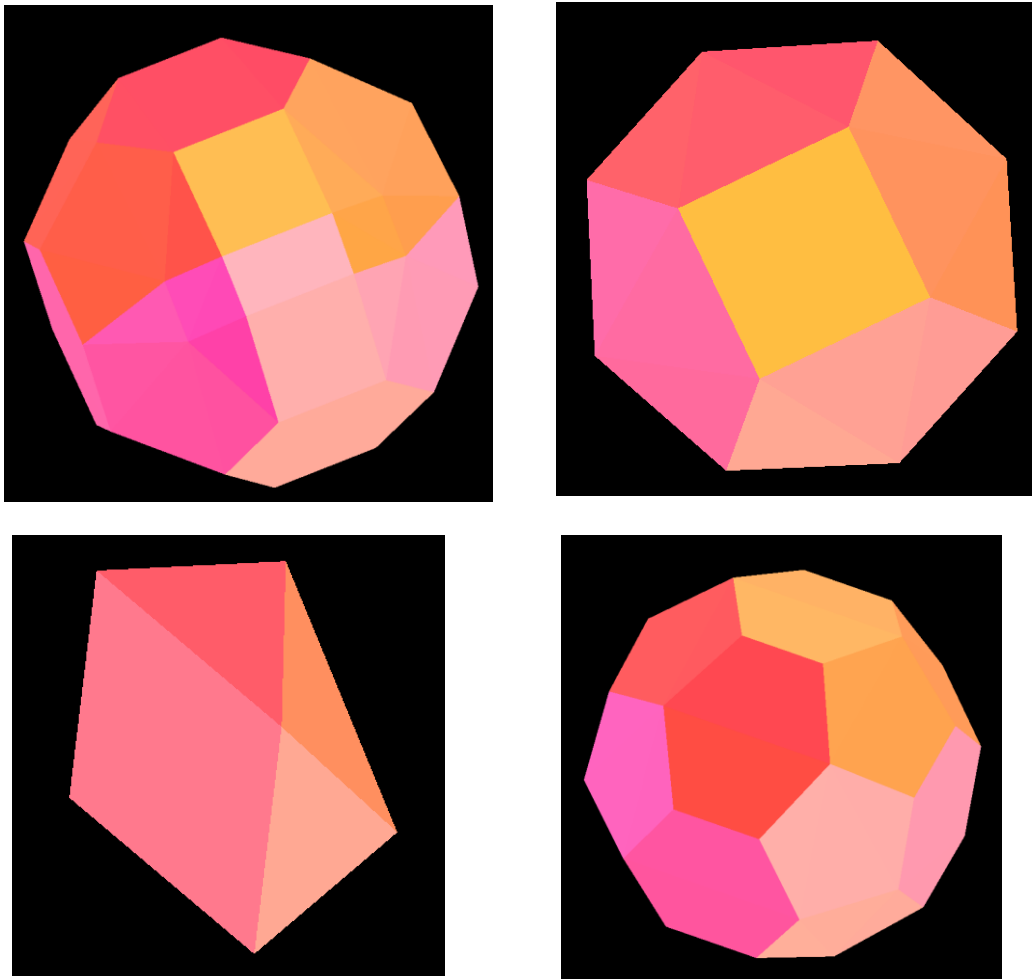
1. mód

Zobrazuje těleso s vrcholy v těžištích trojúhelníků tvořených spojnici částic.

Barva každé stěny je dána tím v jakém bodě rovina, v níž se stěna nachází, protíná souřadnicovou osu x a y.

Ukázka kódu pro výpočet barev

```
double dx, dy, ex, ey, x1, y1;  
dx=(x[a]*z[b]-x[b]*z[a])/(z[a]-z[b]);  
dy=(y[a]*z[b]-y[b]*z[a])/(z[a]-z[b]);  
ex=(x[a]*z[c]-x[c]*z[a])/(z[a]-z[c]);  
ey=(y[a]*z[c]-y[c]*z[a])/(z[a]-z[c]);  
x1=dx*ey-ex*dy;  
y1=x1/(dx-ex);  
x1/= (dy-ey);  
glColor3f(1,atan(x1)/6.28+0.5,atan(y1)/6.28+0.5);
```



Obr. 3: Tělesa vykreslená podle módu 1

Mód 2

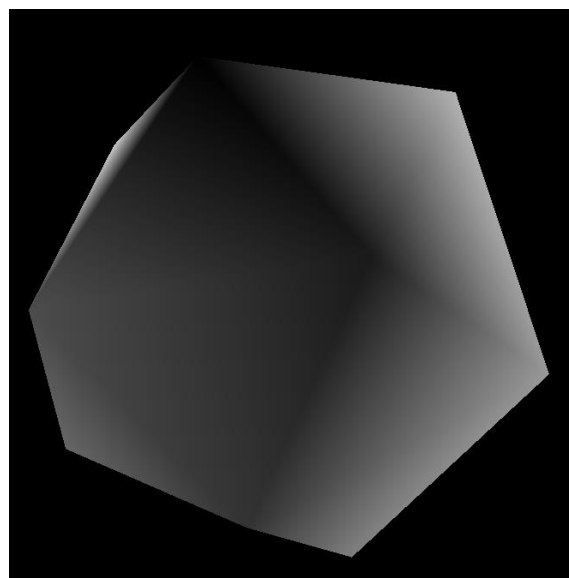
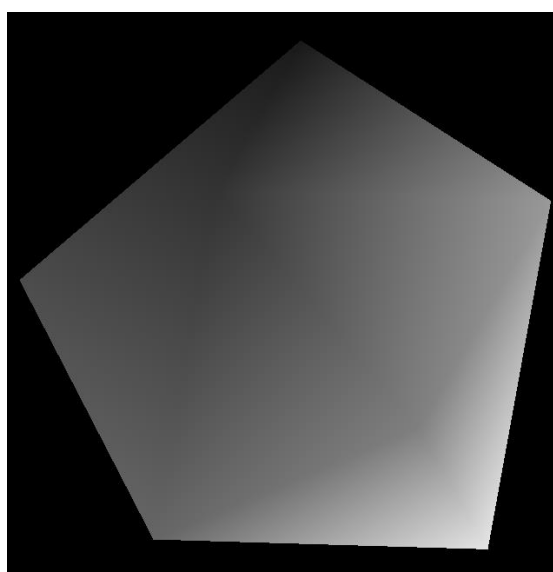
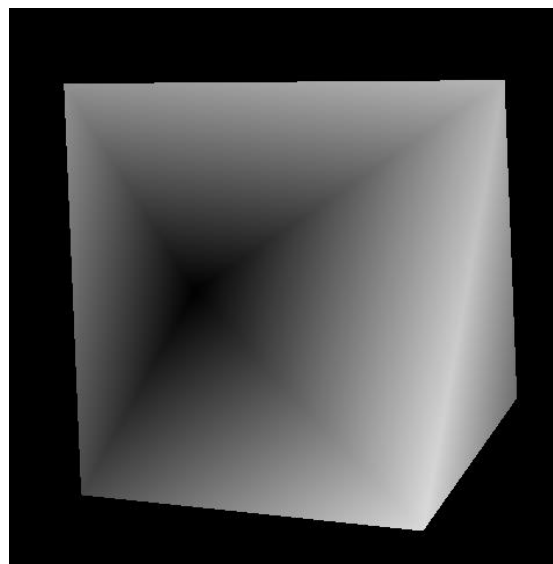
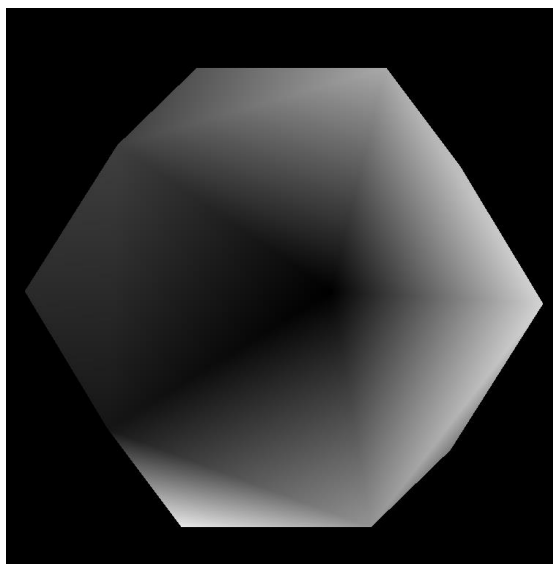
Klávesa S přepne program do módu 2

2. mód

Zobrazuje těleso stejně jako mód 0, liší se vybarvením stěn v odstínech šedi tak, že u každého vrcholu je světlost odpovídající pořadí příslušné částice v paměti.

Ukázka zdrojového kódu pro výpočet vybarvování stěn v módu 2

```
if(mod==2){  
glColor3f(a/(float)n,a/(float)n,a/(float)n);  
glVertex3f(x[a],y[a],z[a]);  
glColor3f(b/(float)n,b/(float)n,b/(float)n);  
glVertex3f(x[b],y[b],z[b]);  
glColor3f(c/(float)n,c/(float)n,c/(float)n);  
glVertex3f(x[c],y[c],z[c]);}
```



Obr. 4: Tělesa vykreslená podle módu 2

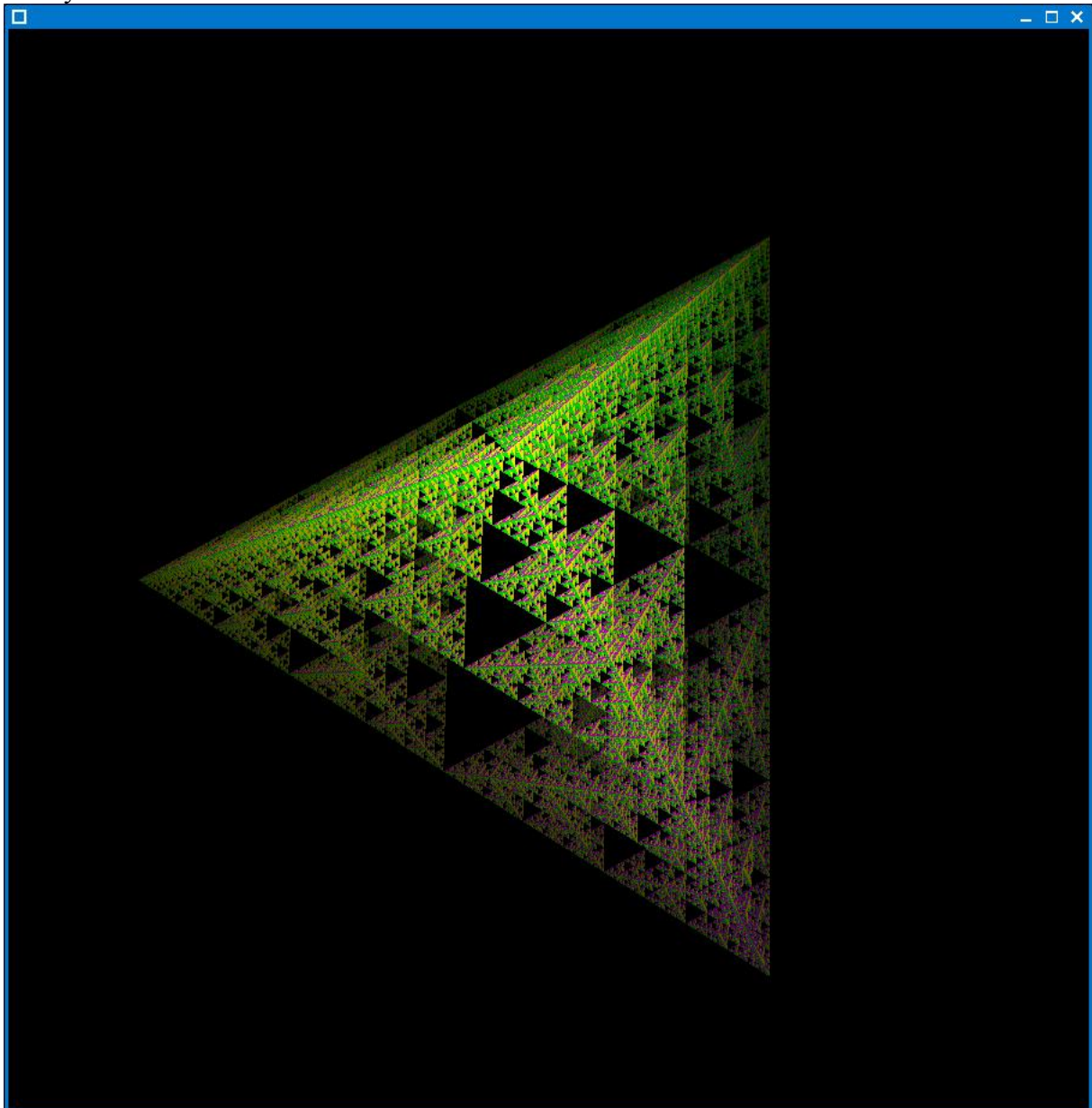
Baví mě matematika a zobrazování fraktálů a proto jsem zkusil napsat program pro vykreslování fraktálů ve 3D. Stejně jako u sférických obrazců, jsem do programu zapracoval obě možnosti otáčení, přidal jsem také možnost otáčení vnitřních trojúhelníků a také postupné překreslování obrazců podle navoleného počtu iterací.

Aby byl dosažen lepší prostorový efekt použil jsem postupné ztmavení vykreslovaných stěn.

Jako druhý ze skupiny programů pro vykreslování těles představuji:

L fraktály 3D

Jedná se o dva interaktivní programy, které pomocí OpenGL a rekurze vykresluji fraktální útvary



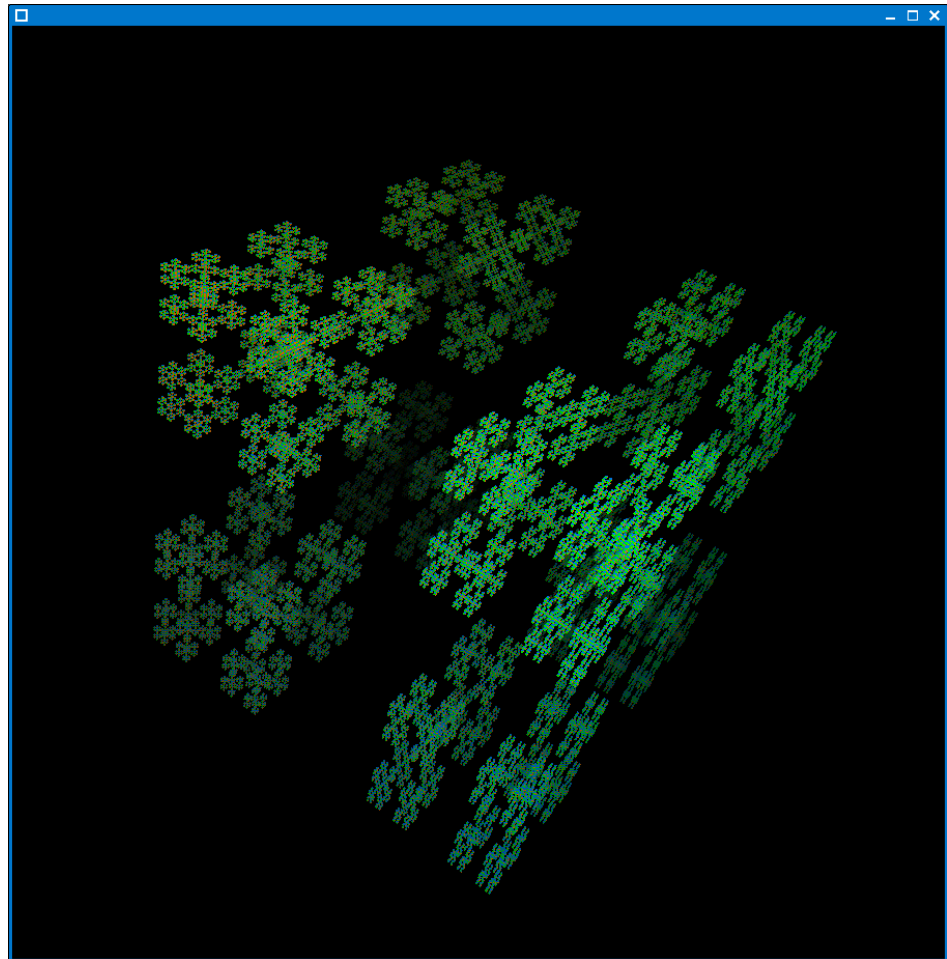
oba fungují na principu umístění částic do prostoru a následné simulaci přitažlivých a odpuzivých sil, tím se získají polohy vrcholů, které se následně použijí jako parametry funkce pro vykreslení tělesa (jejím posledním parametrem je počet iterací) a ta zkontroluje jestli je počet iterací 0 a pokud ano vykreslí těleso pomocí několika trojúhelníků a pokud ne, provede

rozdělení na dílčí části a ty pak vykreslí tou samou funkcí (rekurze)

ukázka kódu:

```
#define VERT(A)
glVertex3f((A).x/(1+(A).z*mod2),(A).y/(1+(A).z*mod2),(A).z);
#define PLUS(A,B,C)
C.x=((A).x+(B).x)/2;C.y=((A).y+(B).y)/2;C.z=((A).z+(B).z)/2;
#define TRH(A,B,C) VERT(A);VERT(B);VERT(C);
int vk5(bod_t a,bod_t b,bod_t c,bod_t d,int n){
bod_t a1,b1,c1;
if(!n){float jas;
jas=0.6-a.z*0.4;
jas*=jas;
glColor3f(jas,jas,jas);
TRH(a,b,c);
glColor3f(jas,0,jas);
TRH(a,b,d);
glColor3f(jas,jas,0);
TRH(b,c,d);
glColor3f(0,jas,0);
TRH(a,c,d);
return 0;
}
PLUS(a,d,a1);
PLUS(b,d,b1);
PLUS(c,d,c1);
if(mod1&2)vk5(a1,c1,b1,d,n-1);else vk5(a1,b1,c1,d,n-1);//
PLUS(a,c,a1);
PLUS(b,c,b1);
PLUS(c,d,c1);
if(mod1&1)vk5(a1,b1,c1,c,n-1);else vk5(a1,b1,c,c1,n-1);//
PLUS(a,b,a1);
PLUS(b,c,b1);
PLUS(b,d,c1);
if(mod1&4)vk5(b,a1,b1,c1,n-1);else vk5(a1,b,b1,c1,n-1);//
PLUS(a,b,a1);
PLUS(a,c,b1);
PLUS(a,d,c1);
if(mod1&8)vk5(c1,a1,b1,a,n-1);else vk5(a,a1,b1,c1,n-1);//
}
```

Druhý program je o trochu složitější, protože simulace částic nemůže udělat vrcholy krychle a navíc ne ve správném pořadí, takže se simulace provádí ze 6 částic a následně se vyberou tři a jejich polohy se použijí jako vektory podle kterých budou hrany



ukázka kódu:

```
f=0;
for(e=2;e< 6;e++)
for(d=1;d< e;d++)
for(c=0;c< d;c++){
TEZI(bo[e],bo[d],bo[c],xy[f]);
if(RANG(xy[f])>0.2){e1=e;d1=d;c1=c;c=d=e=10;break;}
}
f=0;i=0.5;
SUM(bo[c1],i,bo[d1],i,bo[e1],i,xy[0]);f++;
SUM(bo[c1],-i,bo[d1],i,bo[e1],i,xy[1]);f++;
SUM(bo[c1],-i,bo[d1],-i,bo[e1],i,xy[2]);f++;
SUM(bo[c1],i,bo[d1],-i,bo[e1],i,xy[3]);f++;
SUM(bo[c1],i,bo[d1],i,bo[e1],-i,xy[4]);f++;
SUM(bo[c1],-i,bo[d1],i,bo[e1],-i,xy[5]);f++;
SUM(bo[c1],-i,bo[d1],-i,bo[e1],-i,xy[6]);f++;
SUM(bo[c1],i,bo[d1],-i,bo[e1],-i,xy[7]);f++;
vk6(xy[0],xy[1],xy[2],xy[3],xy[4],xy[5],xy[6],xy[7],mod);
```


Ovládání programů pro vykreslování fraktálů:

ovládání je vlastně pro oba programy společné

- otáčení
 - 4 -> vlevo
 - 6 -> vpravo
 - 8 -> nahoru
 - 2 -> dolu
- perspektiva
 - a -> přidat
 - b -> ubrat
- počet iterací
 - + -> přidat
 - - ->ubrat
- NULL (crtl 2 nabo ctrl mezera) -> ukončit program

Závěr:

Programy jsem tvořil sobě i druhým pro radost.

Programy jsou zkompileovány pro prostředí LINUX s použitím X11, nicméně je možné zdrojový kód překompilovat do jiných systémů, které obsahují knihovnu Glut.