

”Учиться, учиться и учиться!”

Владимир Ильич Ульянов (Ленин)

Számítógépes grafika

Adminisztratív információk

Szirmay-Kalos László

email: szirmay@iit.bme.hu

<http://cg.iit.bme.hu/>

Moodle

- Powerpoint
- Mintaprogramok
- Ellenőrző kérdések
- Kvíz
- Könyvek, jegyzet
- Videók: kiegészítés!
- Kommunikáció

Tárgykövetelmények: TAD

3 előadás/hét
Kéthetenként feladat
(nem kötelező labor)
Hetenként kvíz
Vizsga jegy, 5 kredit

	Feladatok:				
	I	II	III	IV	V
	1	1	1	1	1
Bónusz kérdés:	2	2	2	2	2

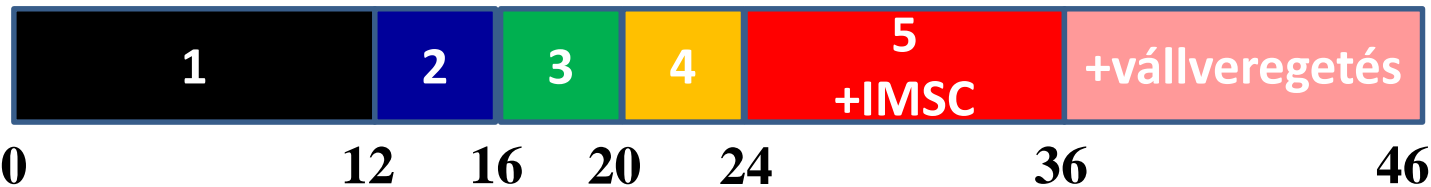
Önkéntes
nagyházi:

N
6



0

Vizsgára bocsátás feltétele: min 3 feladat pont, >40% helyes kvíz kitöltés!



Labor

- Félév során 5 programozási feladat
- Részletes instrukciók a Moodle-en
- Kéthetenkénti laboralkalmak konzultációk (konkrét kérdések)
- Személyes megjelenés nem kötelező/nem szükséges
- Megoldások a Jportára feltöltendőek (bináris értékelés)
- Első két hét kimarad

Miért éppen OpenGL/GLSL?

- Operációs rendszer független, mindenre fut
- Stabil kódbázis, biztos jövőkép
- Minden feladathoz jó (közös alap, nem céleszköz)
- Absztrakciós szint = GPU működését láttatja
- Egyszerű, kezdőknek is jó

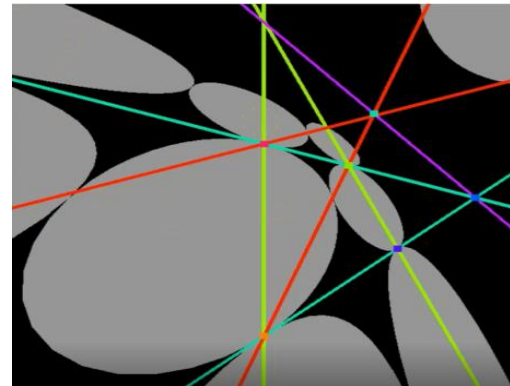
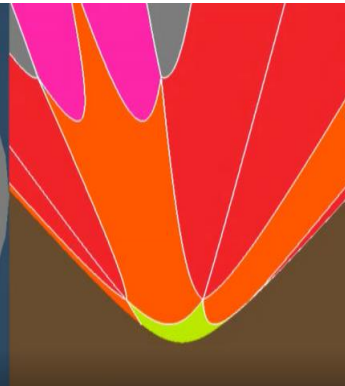
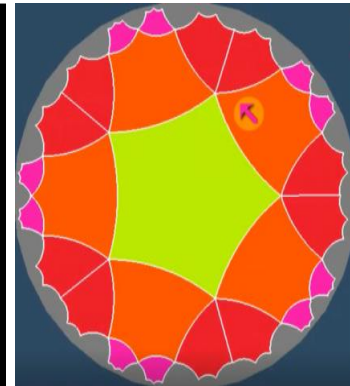
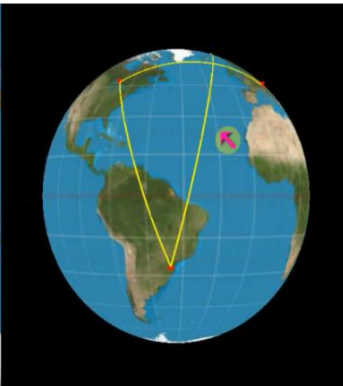
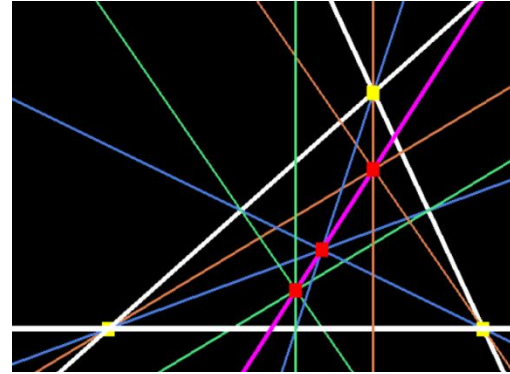
What is the best way to learn computer graphics?

Chat GPT: Learning computer graphics can be a challenging yet rewarding experience.

- **Start with the fundamentals:** Before diving into advanced techniques, it's important to have a solid understanding of the fundamentals of computer graphics. This includes 2D and 3D coordinate systems, transformations, and basic rendering algorithms.
- **Practice with code:** One of the best ways to learn computer graphics is to get your hands dirty by writing code.
- **Work on personal projects:** To reinforce your learning, work on personal projects that interest you.
- **Remember that learning computer graphics takes time and effort, but with dedication and persistence, you can become proficient in this exciting field.**

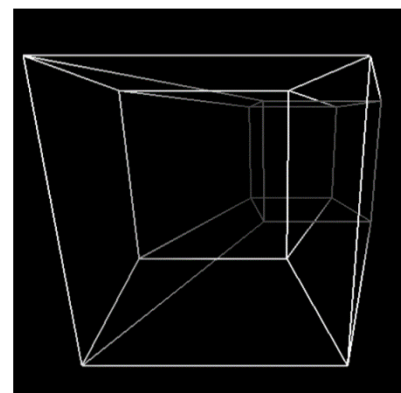
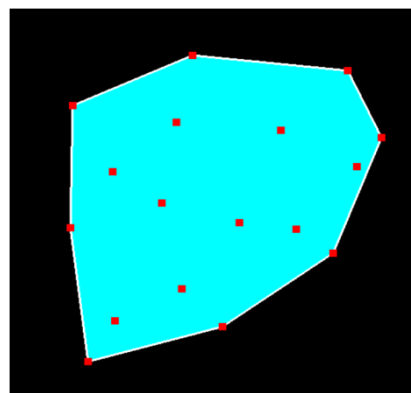
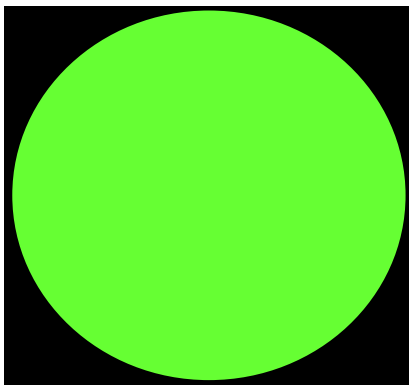
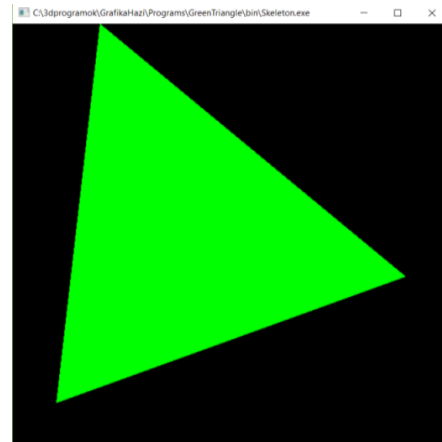
Geometriák és algebrák

- Euklideszi és nem-euklideszi geometriák
- Analitikus geometria **külső nézőpontból**
- Pont, vektor, sík, kvaternió = **vec4**



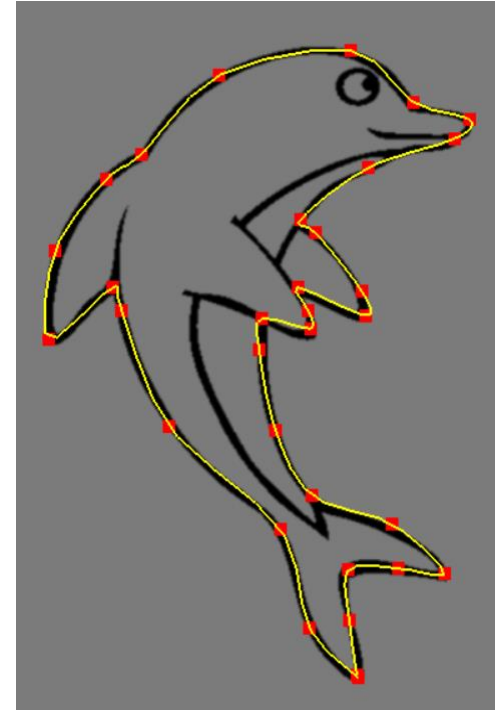
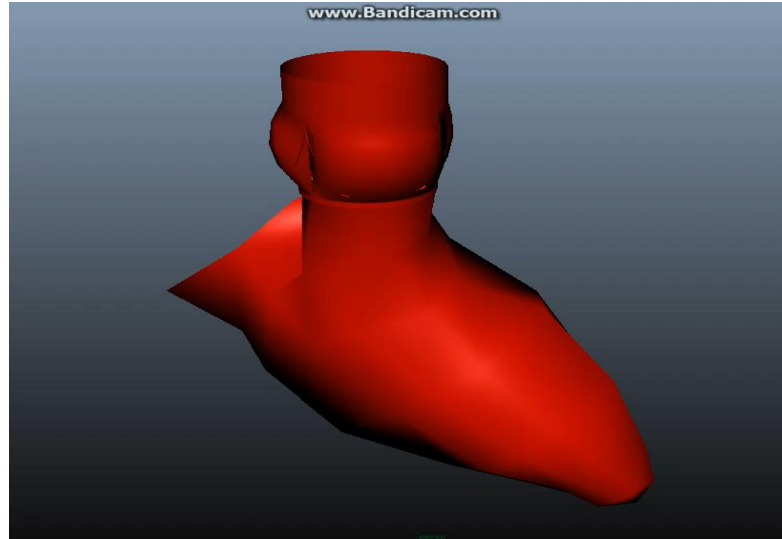
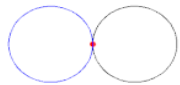
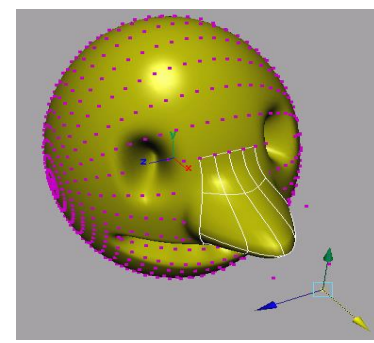
Grafikus hw/sw

- Interaktív grafikus alkalmazások
- OpenGL 3+, GLUT, Glew,
- GPU, GLSL



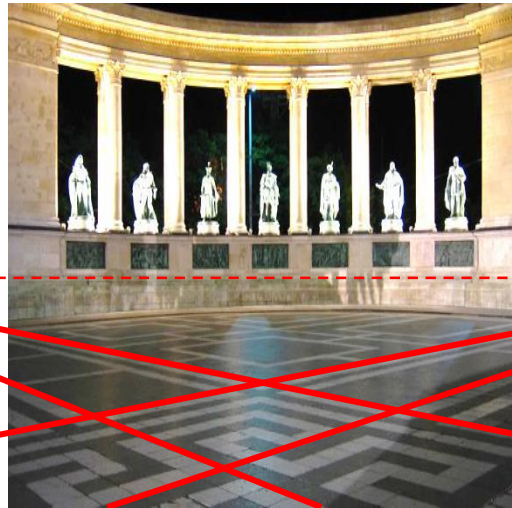
Modellezés

- Descartes és baricentrikus koordinátarendszer
- Implicit és parametrikus görbék/felületek
- Szabadformájú görbék/felületek



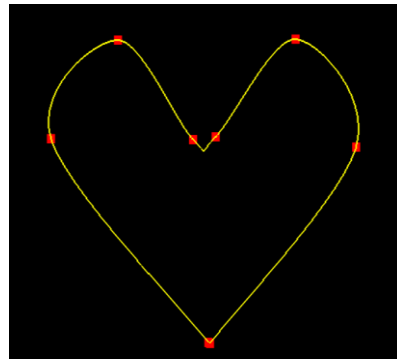
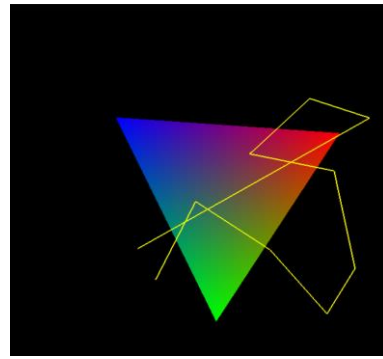
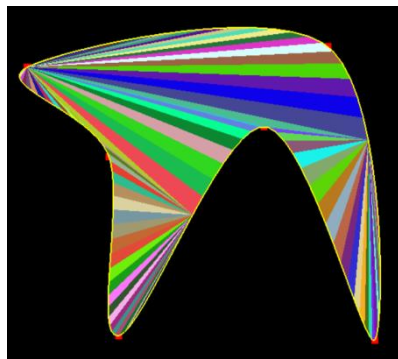
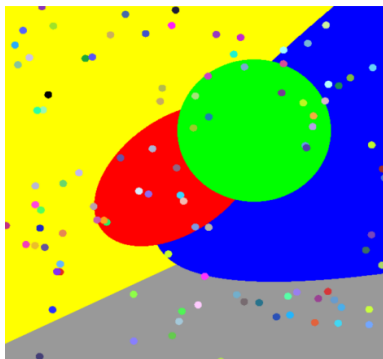
Homogén lineáris transzformációk

- Affin transzformációk (elfolás, forgatás, stb.)
- Egyenestartó transzformációk: **mat4**
- Perspektíva



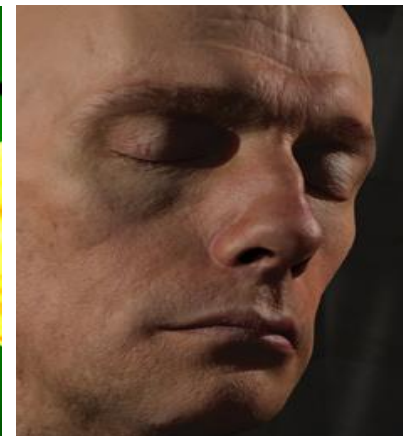
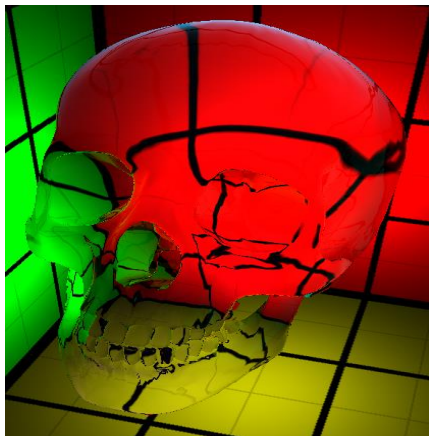
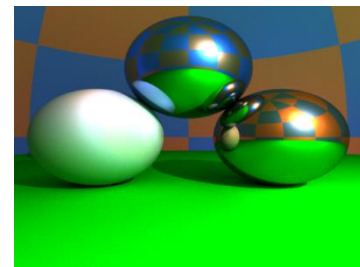
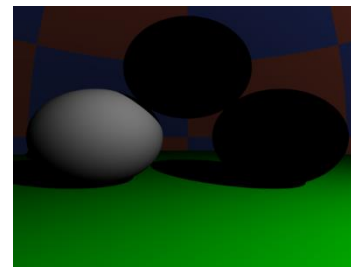
2D képszintézis

- Pixel vezérelt képszintézis
- Objektum vezérelt képszintézis
 - Vektorizáció és háromszögesítés
 - Vágás
 - Raszterizáció



3D grafika fizikai alapjai

- Színek
- Sugársűrűség és számítása
- Optikailag sima anyagok
- Optikailag rücskös anyagok

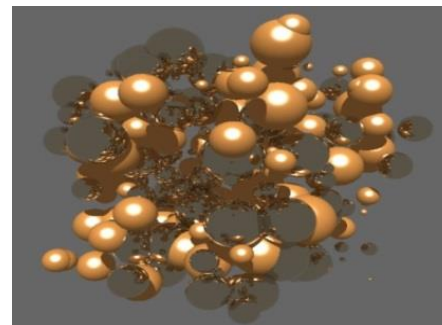
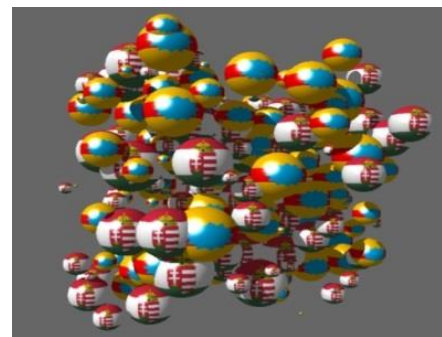
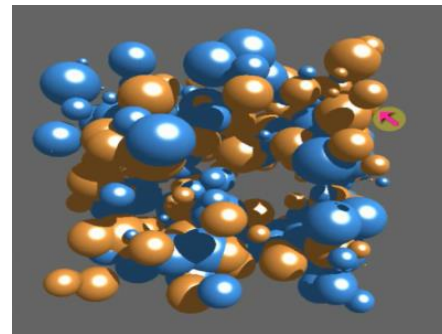


Sugárkövetés

- Láthatóság, árnyékszámítás
- Tükröződés és törés

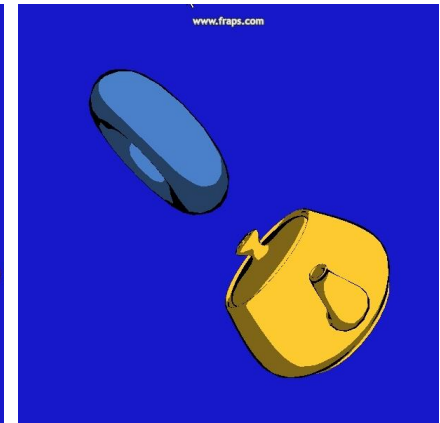
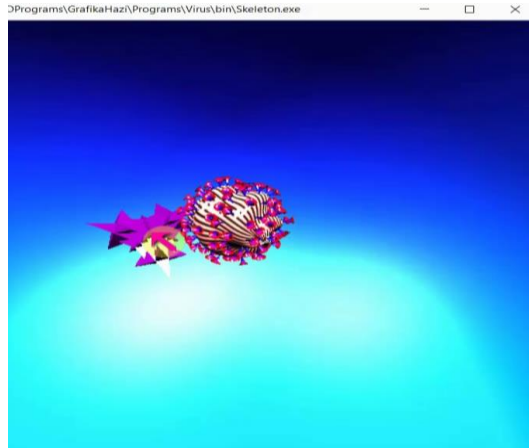
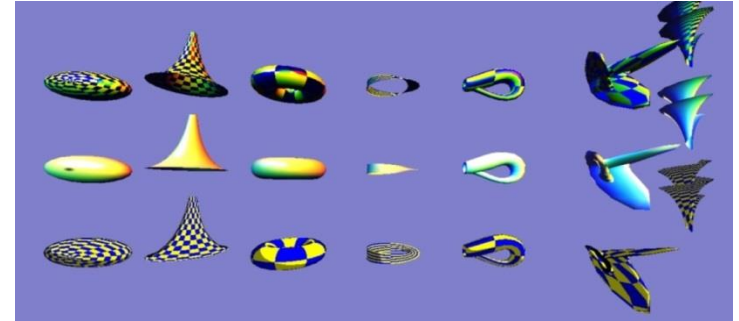
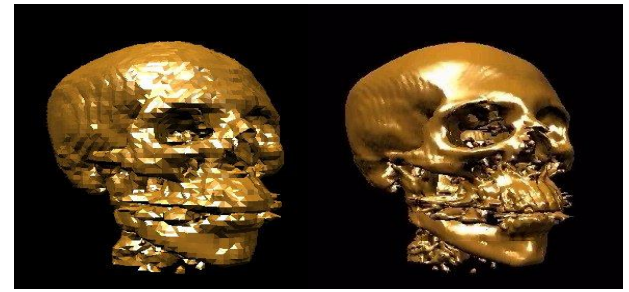
```
typedef struct{double x,y,z}vec;vec U,black,amb={.02,.02,.02}; struct sphere{ vec cen,color;double rad,kd,ks,kt,kl,ir}*s,*best,sph[]={0.,6.,.5,1.,1.,1.,.9,.05,.2,.85,0.,1.7,-1.,8.,-.5,1.,.5,.2,1.,.7,.3,0.,.05,1.2,1.,8.,-.5,1.,.8,.8,1.,.3,.7,0.,0.,1.2,3.,-6.,15.,1.,.8,1.,7.,0.,0.,.6,1.5,-3.,-3.,12.,.8,1.,1.,.5,.0.,0.,.5,1.5,};yx; double u,b,tmin,sqrt(),tan();double vdot(A,B)vec A ,B;{return A.x*B.x+A.y*B.y+A.z*B.z;}vec vcomb(a,A,B)double a;vec A,B; {B.x+=a*A.x;B.y+=a*A.y;B.z+=a*A.z;return B;}vec vunit(A)vec A;{return vcomb(1./sqrt( vdot(A,A)),A,black);}struct sphere *intersect(P,D)vec P,D;{best=0;tmin=1e30;s= sph+5;while(s-->sph)b=vdot(D,U=vcomb(-1.,P,s->cen)),u=b*b-vdot(U,U)+s->rad*s ->rad,u=u>0?sqrt(u):1e31,u=b-u>1e-7?b-u:b+u,tmin=u>=1e-7&&u<tmin?best=s,u: tmin;return best;}vec trace(level,P,D)vec P,D;{double d,eta,e;vec N,color;struct sphere*s,*l;if(!level--)return black;if(s=intersect(P,D));else return amb;color=amb;eta=s->ir;d= -vdot(D,N=vunit(vcomb(-1.,P=vcomb(tmin,D,P),s->cen))) );if(d<0)N=vcomb(-1.,N,black),eta=1/eta,d= -d;l=sph+5;while(l-->sph)if((e=l->kl*vdot(N,U=vunit(vcomb(-1.,P,l->cen))))>0&&intersect(P,U)==l)color=vcomb(e ,l->color,color);U=s->color;color.x*=U.x;color.y*=U.y;color.z*=U.z;e=1-eta* eta*(1-d*d);return vcomb(s->kt,e>0?trace(level,P,vcomb(eta,D,vcomb(eta*d-sqrt(e),N,black))):black,vcomb(s->ks,trace(level,P,vcomb(2*d,N,D)),vcomb(s->kd, color,vcomb(s->kl,U,black))));}
```

```
main(){printf("%d %d\n",32,32);while(yx<32*32) U.x=yx%32-32/2,U.z=32/2-yx++/32,U.y=32/2/tan(25/114.5915590261),U=vcomb(255., trace(3,black,vunit(U)),black),printf("%.0f %.0f %.0f\n",U);}/*minray!*/
```



Inkrementális képszintézis

- GPU csővezeték
 - Tesszelláció
 - Transzformációk és vágás
 - Láthatóság és árnyalás



Játékfejlesztés

- Játékmotor, avatár, plakátok, részecske rendszerek



Fraktálok és káosz

- Hausdorff dimenzió
- Lindenmayer rendszerek
- Perlin zaj
- Iterált függvények

