

## Mintakérdések a Számítógépes grafika és képfeldolgozás tárgy vizsgájára való felkészüléshez

### Analitikus geometria és transzformációk

1. Írja fel azon művelet mátrixát, amely egy  $ax+by+cz+d=0$  egyenletű síkra merőlegesen vetít.
2. Bizonyítsa be, hogy a projektív sík egyenesének  $aX+bY+ch = 0$  egyenlete összhangban van a projektív geometria azon axiómáival, hogy „két különböző egyenes egy pontba metszi egymást”, és hogy „két pont meghatároz egy egyenest”.
3. Írja fel három pontra illeszkedő sík egyenletét az euklideszi és a projektív térben.
4. Bizonyítsa be, hogy a homogén lineáris transzformációk a konvex kombinációkat konvex kombinációkká képezik le.
5. Írja fel azon síkpontok mértani helyének egyenletét, amelyek egy egyenestől és egy ponttól ugyanolyan távolságra vannak. Milyen alakzat ez? Terjessze ki az alakzatot a projektív síkra. Mík az ideális pontjai.
6. Írjon C függvényt, amely egy egyenes és egy pont távolságát kiszámítja.
7. Írjon C függvényt, amely két térbeli egyenes távolságát kiszámítja.
8. Írjon C függvényt, amely egy implicit egyenletével adott egyenes, és két végpontjával adott szakasz metszéspontját kiszámítja.
9. Írja fel a projektív sík egyenesének implicit (azaz nem paraméteres) egyenletét homogén koordinátákban. Bizonyítsa be, hogy a sík minden invertálható homogén lineáris transzformációja ezt az egyenest egyenesre képezi le. Melyek azok a transzformációk, amelyek az egyenest önmagára képezik le. Mi történik az egyenes normálvektorával?
10. Lehet-e egy affin – azaz párhuzamos egyeneseket párhuzamos egyenesekbe leképző – transzformáció mátrixának negyedik oszlopa  $[0, 0, 0, 2]$ ?
11. A projektív tér síkjainak homogén lineáris transzformációi. A projektív tér síkjának egyenlete. A transzformáció mibe viszi át a síkot (feltételezheti, hogy a transzformációs mátrix invertálható), bizonyítás. Mi történik a sík normálvektorával. Mi ennek a következménye az OpenGL működésére.
12. Tekintsük a projektív térben a  $[1, 2, 3, 4]$  és  $[-4, -3, -2, -1]$  homogén koordinátákkal azonosított végpontok konvex kombinációiként előálló szakaszt! Mekkora a szakasz hossza?
13. Írja fel azon homogén lineáris transzformáció mátrixát, amely egy síkbeli pontot az  $xc, yc$  vetítési középponttal az  $ax+by+c=0$  egyenletű egyenesre vetít.
14. Tekintsük a következő homogén lineáris transzformációs mátrixot (a transzformálandó pontot a mátrix bal oldalára kell írni):

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Mit csinál a transzformáció? Mi keletkezik a transzformáció után a  $[2, 1]$  és  $[-1, 1]$  pontokat összekötő szakaszból.

15. Adott a következő homogén lineáris transzformáció (a transzformálandó pont helyvektorát sorvektornak kell tekinteni).

$$\begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 8 & 6 & 4 & 2 \end{bmatrix}$$

Adja meg azon ponthalmaz egyenletét, amelyre ez a transzformáció a  $8x+6y+8z+6 = 0$  egyenletet kielégítő ponthalmazt leképezi! Először írja le a megoldás menetét, azután végezze el a szükséges számításokat.

16. Adott két pont Descartes koordinátákkal:  $(3,6)$ ,  $(-2,5)$ . Adja meg erre a két pontra illeszkedő egyenes ideális pontját homogén koordinátákban!
17. Bizonyítsa be, hogy ha egy invertálható transzformációs mátrix 4. oszlopa  $[0, 0, 0, 1]$ , akkor a transzformáció **affin**, azaz párhuzamos egyeneseket párhuzamos egyenesekre képez le (a transzformálandó pont homogén koordinátáit sorvektornak tekintjük és a mátrixszal jobbról szorozzuk)!
18. Írja fel a 3D projektív tér összes ideális síkjának és ideális egyenesének egyenletét (az ideális térelem csak ideális pontokat tartalmaz). Hány ideális sík és egyenes van?
19. Írja fel a projektív sík egy körének egyenletét! Mi lesz abból a körből, amelynek középpontja egy ideális ponton van? Segítség: a kör egyenlete homogén koordinátákban *annak analógiájára, ahogyan az egyenes egyenletét a síkban, és a sík egyenletét a térben bevezettük.*
20. Tekintsük a projektív tér  $[a,b,c,d]$  homogén négyessel megadott síkját. Írja fel ezen sík ideális egyenesének paraméteres egyenletét!
21. Írjon C++ függvényt, amely a bemeneti paraméteréül a projektív sík két egyenesét kapja, kiszámítja a két egyenes metszéspontját, a metszéspontot vetíti az origó középpontú, egység sugarú körre, és a vetület Descartes koordinátáit adja vissza.
22. Adott a következő  $\mathbf{r} \rightarrow \mathbf{r}'$  2D transzformáció, ahol  $\mathbf{r}$ ,  $\mathbf{r}'$  és  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  a sík vektorai:
 
$$\mathbf{r}' = [(\mathbf{a} \cdot \mathbf{r})/(\mathbf{c} \cdot \mathbf{r}), (\mathbf{b} \cdot \mathbf{r})/(\mathbf{c} \cdot \mathbf{r})].$$
 Mibe vihet át ez a transzformáció egy szakaszt?
23. Adott a következő  $\mathbf{r} \rightarrow \mathbf{r}'$  2D transzformáció, ahol  $\mathbf{r}$ ,  $\mathbf{r}'$  és  $\mathbf{b}$  a sík vektorai:
 
$$\mathbf{r}' = \mathbf{r}/|\mathbf{r}| + \mathbf{b}.$$
 A képletben  $|\mathbf{r}|$  az  $\mathbf{r}$  vektor abszolút értékét jelenti. Mibe vihet át ez a transzformáció egy szakaszt?
24. Írja fel azt a mátrixot, amely egy  $\mathbf{r}$  helyvektorú pontot  $\mathbf{a} \times (\mathbf{a} \times \mathbf{r})$  pontba képezi le.
25. Hova kerül a  $(0, 0, 1)$  pont, ha az origón átmenő  $(1, 1, 0)$  irányú tengely körül  $+45$  fokkal elforgatjuk.

## Geometriai modellezés

26. Bizonyítsa be, hogy a Bézier görbét érinti az első két kontrolpontokra fektetett egyenes.
27. Adja meg a kvadrátikus felületek általános definícióját. Milyen konkrét tagjai vannak ennek a családnak? Hogyan számolható ki a normálvektor a kvadrátikus felület egy pontjában. Hogyan változik az egyenlet, ha egy homogén lineáris transzformációt hajtunk rajta végre?
28. Mik az Euler operátorok és miért van rájuk szükség? Általánosítsa az Euler tételt több részből álló, illetve lyukas objektumokra.
29. Írja fel a tóruszfelület egység hosszúra normalizált normálvektorát az  $u, v$  paraméterű pontjában?
30. A Catmull-Rom spline kontrollpontjai és hozzájuk tartozó csomóértékei rendre  $\mathbf{r}_1, t_1; \mathbf{r}_2, t_2; \mathbf{r}_3, t_3; \mathbf{r}_4, t_4$ . Mi lesz a spline érintőjének iránya a  $t=t_2$  paraméternél?
31. Egy ellipszoid középpontja  $(cx, cy, cz)$ , a tengelyeinek hossza  $a, b, c$ . Írja fel az ellipszoid normálvektorát annak  $(x, y, z)$  pontjában! (1 pont)
32. Adott egy 9 vezérlőpontra illeszkedő  $3 \times 3$ -as Bézier felület. A vezérlőpontok rendre  $p_1, \dots, p_9$ . Adja meg a felület normálvektorát a  $(0, 0.5)$  paraméterértékek mellett! Hogyan változik az előző feladatban kiszámolt normálvektor, ha a vezérlőpontokat a

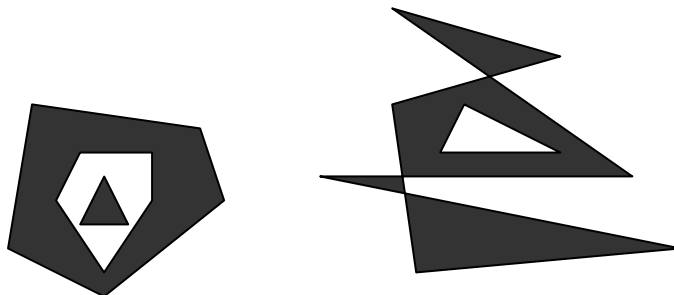
$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 6 & 7 & 8 & 1 \end{pmatrix}$$

homogén lineáris transzformációs mátrixszal megszorozzuk?

## 2D képszintézis

33. Írjon DDA szakaszrajzoló függvényt,  
`DDADraw(int x1, int y1, int x2, int y2, int R, int G, int B)`  
amely bármilyen **pozitív meredekségű** szakaszt képes felrajzolni. Használhat lebegőpontos aritmetikát. Egy rasztertárbeli pixel színét a  
`PutPixel(int x, int y, int R, int G, int B)`  
függvénnyel változtathatja meg.
34. Írjon programot, amely egy szakaszt egy konvex sokszögre vág.
35. Írjon programot, amely egy tetszőleges sokszöget egy konvex sokszögre vág.
36. Írjon háromszög kitöltő programot, amely háromszögenként 3 osztást és fixpontos összeadásokat tartalmazhat.
37. Írjon C++ programot, amely eldönti, hogy egy  $p_1, p_2, p_3$  és egy  $v_1, v_2, v_3$  síkbeli háromszögnek van-e közös része. Feltételezheti, hogy a vektor összeadás (+), kivonás (-), skaláris szorzás (\*), és 90 fokkal való elforgatás ( $[x, y] \rightarrow [-y, x]$ , `.Rot90`) műveletek a vektorokra (`Vec`) rendelkezésre állnak. Bónusz: általánosítsa a programot két konvex sokszögre.
38. Írjon háromszögvágó programot, amely a csúcaival definiált háromszögből eltávolítja azokat a pontokat, amelyek a paraméterként kapott  $y_{min}$  érték alatt vannak, és visszatérési értékében jelzi, hogy maradt-e egyáltalán valami a vágás után. Az eredményt a három-szögek csúcsait tartalmazó tömbben adja vissza. A megoldás során feltételezheti, hogy rendelkezésre áll egy `Intersect` eljárás, ami két egyenes metszéspontját kiszámítja.
39. Írjon programot, amely egy paraméterként kapott, vezérlő pontjaival ( $x[], y[], npoint$ ) definiált Bézier görbét töröttvonalal közelít.
40. Írjon 2D felosztott görbe (Catmull-Clark subdivision curve) rajzoló rutint C++ nyelven.
41. Írja fel azt az időfüggő homogén lineáris transzformációs mátrixot, amely egy  $[2, 0]$  sebességű autó 2 egység sugarú kerekét mozgatja és forgatja. Az animáció kezdetén a kerék transzformációs mátrixa az egységmátrix.
42. Írjon szakaszvágó programot, amely a szakaszból eltávolítja azokat a pontokat, amelyek a paraméterként kapott  $y_{min}$  érték alatt vannak és visszatérési értékében jelzi, hogy maradt-e egyáltalán valami a vágás után.
43. Rajzolja fel annak a hardvernek a vázlatos kapcsolási rajzát, amely minden órajelciklusra egy erősen emelkedő ( $x_2 > x_1, y_2 > y_1, y_2 - y_1 > x_2 - x_1$ ) egyenes szakasz újabb pixelének  $x, y$  címét előállítja. Milyen értékekkel kell inicializálni a tárolókat?
44. Területkitöltés geometriai reprezentáció alapján (belső pont azonosítása, naív algoritmus, gyorsítási lehetőségek, inkrementális elv, aktív él tábla, él tábla).
45. Területek vágása ablakra (algoritmus, több részre eső területek kezelése)
46. Írja fel egy 2D autókerék transzformációs mátrixát az idő függvényében, ha az autó az  $x$  tengely irányába halad  $v$  sebességgel, a talaj  $y$  koordinátája zérus, a kerék sugara  $r$ . A lokális modellezési-koordináta-rendszerben a kerék tengelye az origóban van, és  $t=0$ -ban az  $x$  koordináta ugyancsak zérus. A kerék a talajon gördül.
47. Adott egy zárt töröttvonal a síkban. A csúcsok koordinátái a  $px[n], py[n]$  tömbökben vannak, a csúcsok száma az  $n$  változóban. Írjon C++ függvényt, amely egy  $(x, y)$  pontról eldönti, hogy a töröttvonal által definiált sokszög belsejében van-e.

48. Az erdészetnél vállalt munkát, ahol tölgyfaerdőt telepítettek egy szabályos rács szerint  $N$  sorban és  $M$  oszlopban. Sajnos távvezeték építés miatt az erdőből fákat kell kivágni a távvezeték egyenes vonala alatt. Tudjuk, hogy a távvezeték melyik sorban/oszlopban lép be az erdőbe és melyikben távozik. Valamilyen szakaszrajzoló algoritmus adaptálásával írjon C++ programot, amely a sztandard outputra kiírja, hogy mely sorban és oszlopban lévő fákat kell kivágni. Természetesen, csak minimális számú fát szeretnénk kivágni, ezért egy kivágott fának legfeljebb két, ugyancsak kivágott szomszédja lehet (két fa szomszéd, ha sor és oszlopszámuk külön-külön legfeljebb eggyel térnek el egymástól!
49. Tőzsdeügynöknek csapott fel, és a részvényekről egy nagyon nagy méretű adatbázissal rendelkezik. Egy részvényt a sorszámaival, a hozammal (0..1000) és a törzstőkével (0..1000) jellemezhetünk. A kuncsaftok azt a kérdést tehetik fel, hogy melyek azon részvények sorszámai, amelyek hozama az általuk megadott hozamtól legfeljebb 10-zel tér el, ÉS az általuk megadott törzstőkétől ugyancsak legfeljebb 10-zel tér el. Szerencsére még emlékszik a sugárkövetés szabályos rács felosztó eljárására, így tőzsdeügynökként is megállja a helyét. Írjon olyan programot, amely egy előfeldolgozási lépésből és egy lekérdező lépésből áll C++ nyelven. Az előfeldolgozási fázis időbonyolultsága nem érdekes, úgymint csak egyszer kell végrehajtani. Viszont a nagy piaci verseny miatt csak olyan lekérdező lépés engedhető meg, amelynek válaszideje nem nő a részvények számával abban az esetben, ha a részvényadatok valószínűségi eloszlása egyenletes (olyan megoldások, ahol a lekérdezési idő a részvények számával nő, különösen, ha azzal egyenesen arányos, nem értékelhetők ezen a ZH-n sem). A bemeneti adatok a részvények tömbje és száma. A lekérdezési fázis bemenete a kívánt hozam és tőke, a kimenete pedig a kívánalomnak eleget tevő részvények sorszámainak tömbje. A megoldás során a tömbök allokálásával nem kell foglalkozni.
50. Írjon C++ függvényt, amely egy paraméterként kapott  $x, y$  középpontú ellipszist töröttvonalal közelít.
51. Írjon görbesimító programot C++-ban OpenGL és GLUT felhasználásával. Az windows ablak háttérszíne kék, felbontását szabadon megválaszthatják. A program egy "pontok.txt" nevű ascii fájlt olvas be, amelynek első sorában egy int van (a pontok száma), amit ennyi sor követ, minden sorban két float számmal, amelyek a pontok  $x, y$  világkoordinátáit tartalmazzák. A program ezen csúcspontokra egy zárt töröttvonalat illeszt és egy  $(0,0), (100,100)$  sarokpontokkal definiált ablakú kamerával lefényképezi, az eredményt a képernyő windows ablakában megjeleníti. A zárt töröttvonalat fehér színnel kell felrajzolni. Ha a felhasználó a bal egér gombbal ráklikkel a töröttvonalra, a program Catmull-Clark algoritmussal simít egyet a töröttvonalon. Jobb egérklick pedig csökkenti a simítás mértékét.
52. Írjon C++ függvényt, amely egy zárt sokszögre eldönti, hogy rámutatással kijelöltük-e. Akkor tekintjük a sokszöget kijelöltnek, ha a  $(x,y)$  kijelölési pont középpontú, 10 egység oldalhosszúságú téglalap tartalmazza a sokszögből egy részt (ez a rész bármilyen kicsiny lehet, a vizsgálatot tehát nem elég egész koordinátákra elvégezni). A függvény deklarációja:
- ```
struct Vertex { float x, y; };
bool Talalat(float x, float y, int n, Vertex * p);
```
- ahol  $n$  a csúcspontok száma,  $p$  pedig a csúcspontok tömbje.
53. Írjon C++ függvényt, amely egy többszörösen összefüggő sokszöget kitölt (akkor nevezünk egy sokszöget többszörösen összefüggőnek, ha a határát több zárt töröttvonalal lehet megadni). A függvény bemenete az egyes töröttvonalak csúcsszáma, illetve a töröttvonalak csúcspontjai (2 dimenziós tömb). Felételezheti, hogy a művelet előtt az ablakra vágunk. Az ablak sarokpontjai  $(0,0), (1000,1000)$ . Egyetlen pixel a SetPixel( $x,y$ ) függvénnyel színezhethet át a sokszög színére. Példák kitöltött, többszörösen összefüggő sokszögekre:



## Illumináció

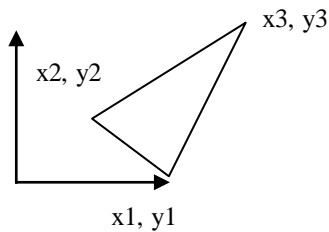
54. Írjon egy BRDF osztályt, ami diffúz és Phong-Blinn szerinti illuminációs modellt valósít meg.
55. A fény egy ideális tükörrre  $V$  irányból érkezik, a tükör normálvektora  $N$ . Milyen irányba halad tovább?
56. A fény egy ideális fénytörő felületre a  $V$  irányból érkezik, a tükör normálvektora  $N$ . Milyen irányba halad tovább, ha a törésmutató  $n$ ?
57. Az országgyűlés felülírta a Snellius-Descartes törvényt, innentől abban nem szinuszok, hanem tangensek aránya szerepel. Oldja meg az előző feladatot az új törvény szerint.
58. Egy tökéletesen sík felületre érkezik a fény  $[3,4,5]$  irányból. Milyen irányban verődik vissza, ha a sík normálvektora  $[-3,8,1]$ , és a fény a felületre merőlegesen polarizált?
59. Írjon egy BRDF függvényt, amely a fény, nézeti és normálvektorokat kapja meg és a BRDF tényezőt adja vissza RGB hullámhosszokon. A függvény a diffúz és az eredeti Phong-Blinn szerinti illuminációs modellt valósítja meg.
60. Adott egy irányfényforrás, amely a  $[3,4,0]$  irányból világít, intenzitása  $[200, 10,100]$ . Milyen színűnek látja a megfigyelő a  $z=0$  sík  $[3,5,0]$  pontját, ha ő a  $[7,5,3]$  pontban van, és a felület Phong-Blinn illuminációs modell szerint veri vissza a fényt. A spekuláris visszaverődési  $k_s = [25/9,25/9,25/9]$ , a fényesség (shininess) 2.
61. Egy  $(0,0,1)$  normálvektorú piros műanyag (nem fém!!!) felületről tudjuk, hogy Phong-Blinn modell szerint veri vissza a fényt. A  $(1,1,1)$  irányból érkező  $(1,1,1)W/st/m^2$  sugársűrűségű (R,G,B hullámhosszain) megvilágítás hatására a piros hullámhosszán a  $(1,1,1)$  irányból ránézve a felületre  $0.3 W/st/m^2$  sugársűrűséget, a  $(-1,-1,1)$  irányból ránézve pedig  $0.5 W/st/m^2$  sugársűrűséget észlelünk. Mekkora a  $kd$  diffúz visszaverődési tényező és a  $k_s$  spekuláris visszaverődési tényező az R,G,B hullámhosszokon?
62. Tekintsük a  $12x + 4y + 3z = 19$  egyenletű sík  $(1, 1, 1)$  koordinátájú pontját, amit a  $(13, 6, 1)$  pontbeli az r, g, b hullám-hosszokon egyaránt  $169^2 * 4 * \pi W$  teljesítményű fényforrás világít meg, és a  $(4, 13, 5)$  pontból nézzük. A sík anyagát diffúz + Phong-Blinn modellel írjuk le. Milyen intenzitású fényt érzékel a szem az r, g, b hullámhosszokon, ha a felület diffúz visszaverődési tényezője ugyanezekben a hullámhosszokon  $[0.1, 0.2, 0.3]$ , a spekuláris visszaverődési tényező minden hullámhosszon 2.89, a fényesség (shininess) pedig 2. Numerikus eredményt várunk, nem elég csak a képleteket felírni, ki is kell az eredményt számítani (nem hiszünk el semmit, csak ami le van írva). *Segítség: A pontszerű fényforrás által keltett sugársűrűség a teljesítmény osztva távolság négyzetével és  $4\pi$ -vel. A Phong-Blinn modell, amit az OpenGL is használ, a felezővektor és normálvektor közötti szöggel dolgozik.*
63. Az ideális tükörgömb  $p$  pontjára  $l$  irányból érkező fénysugár milyen irányba verődik vissza, ha a gömb középpontja  $c$ .

## Sugárkövetés

64. Írja le azon inkrementális algoritmus elvét (ötlet + képletek), amely egy 3D szabályos rács azon celláit kiírja, amelyet egy adott sugár metsz.
65. Írjon sugár és kvadratikus felület (a kvadratikus felület implicit egyenlete egy kvadratikus alak) metszési eljárást egy sugárkövető algoritmushoz. Bemenet: sugár kezdőpontja, irányvektora, a kvadratikus alak  $4 \times 4$ -es mátrixa.
66. Írjon sugárkövető programot, amely pontszerű fényforrások által megvilágított diffúz gömböket tud megjeleníteni. A szem a  $0,0,0$  pontban van, a  $z$  irányba néz. Az ablak közepe a  $0,0,1$  pontban van, a  $z$  irányra merőleges, mérete  $2 \times 2$ . Az képernyő felbontása  $1000 \times 1000$  pixel. Feltételezheti, hogy a 3d vektort és RGB szint megvalósító típusok rendelkezésre állnak, rájuk a szokásos operátorok működnek
67. A 2D síkon élők (Laposföld) lakói is sugárkövető programra vágnak. Segítsen nekik! A laposföldieknél a kamera mindig a koordinátarendszer origójában van, és az  $x$  irányba néz. A látószög  $90$  fokos. Az ablak a szentől egységnyi távolságban van, a felbontása  $100$ . Laposföldön a nap irányfénynek tekinthető, amelyet a  $(-1, 1)$  irányba tekintve érzékelünk, csak  $350$  nm-en sugároz, mégpedig egységnyi intenzitással. Laposföldön  $n$  darab kör található, amelyeket középpontjával és sugarával adunk meg. A tárgyak diffúzak, az  $i$ . gömb diffúz visszaverődési tényezője  $350$  nm-en éppen  $i$ . A laposföldi monitor csak  $350$  nm-en sugároz, a  $j$ . pixel intenzitását a  $\text{SetPixel}(j, \text{intenzitás})$  függvénnyel lehet beállítani. Készítse el a laposföldi sugárkövető programot C++ nyelven!
68. Írjon sugár háromszög metszéspontszámító függvényt C++-ban. A sugarat a kezdőpontjának helyvektorával, és irányvektorával adjuk meg. A háromszöget a három csúcának helyvektorával.
69. Írjon programot, amely egy  $s$  kezdőpontú és  $d$  irányvektorú sugárra  $0.01$  egység távolságon belüli pontossággal kiszámítja a sugár és a tórusz metszéspontját, feltételezve, hogy a sugár a tórusz belsejéből indul (12 pont). Adja meg analitikusan a tórusz normálvektorát a metszéspontban?
70. Hogyan találja meg az  $f(x) = 0$  nemlineáris egyenletet egy gyökét epsilon-nál kisebb hibával, ha tudja, hogy  $f$  folytonos,  $f(0) = -2$  és  $f(10) = 3$ .
71. Egy síkot három pontjának  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$  helyvektorával definiálunk. Hol (melyik pontban) metszi a síkot a  $\mathbf{ray}(t) = \mathbf{e} + \mathbf{v}t$  egyenletű sugár?

## Inkrementális képszintézis

72. A szem a koordináta rendszer origójában van és z irányba néz, a látószög x-ben és y-ban is 90 fokos, az első vágósík 0.1-en, a hátsó 2-n van. Adja meg azt a transzformációs mátrixot, amely a látható tartományt a  $[-1,-1,0]$ ,  $[1,1,1]$  sarokpontokkal definiált téglatestbe viszi át. Miért kell vágni a transzformáció végrehajtása előtt a  $z=0.1$  síkra?
73. Írjon 3D háromszög megjelenítőt, amely a képernyőkoordináta rendszerbe transzformált csúcspontokat kapja meg, és a takaráshoz z-buffer algoritmust használ. A háromszöget konstans színnel kell kitölteni. Feltételezheti, hogy a vektor műveletek rendelkezésre állnak és azt is, hogy a háromszög vetületében a koordináták az alábbi relációban állnak.



74. Phong árnyalás. Adja meg, hogy az egyes vektorok milyen koordináta-rendszerben értendők.
75. Phong árnyalással jelenít meg egy háromszöget, amelynek csúcspontjai világ-koordináta-rendszerben  $(6, 4, 0)$ ,  $(10, 4, 0)$ ,  $(6, 6, 0)$ . A háromszög csúcsaihoz rendre a következő normálvektorokat rendeltük  $(1,0,0)$ ,  $(0,1,0)$ ,  $(0,0,1)$ . A háromszög diffúz, a visszaverési együttható  $(1, 0.5, 0.5)$ . Egyetlen irányfényforrás van, amely a  $(11, 5, 4)$  irányból világít  $(R=3, G=1, B=2)$  intenzitással. A szem a  $(5, 6, 7)$  pontban van. Milyen színű lesz RGB színrendszerben a háromszög  $(8, 5, 0)$  pontja (az eredményt egy értékes jegy pontossággal kérjük)
76. Gouraud árnyalás. Milyen hardvertámogatás építhető hozzá?
77. Háromszög textúrázása inkrementális képszintézisben.
78. Írjon egy C függvényt, amely egy 3D háromszöget az  $x=0$  síkra vág. A függvény bemenete tehát három pont, kimenete pedig azon háromszögek tömbje és darabszáma, amelyek a kapott háromszögek az  $x=0$  sík feletti részét adják ki.
79. Implementálja a Sutherland-Hodgeman poligonvágó algoritmust 3D-ben, Descartes koordinátákban megadott háromszögekre. A megvalósítandó C++ `Vagas` függvény bemenete a háromszögek száma a háromszögeket tartalmazó tömb (`in`), a vágási tartomány két sarka (`Vector min`, `Vector max`), kimenete (referenciaváltozó) pedig a vágott háromszögeket tartalmazó tömb (`out`).
- A tömb típus (`Tomb`) a következő műveletekkel kezelhető:
- `Add( Haromszog h )`: egy új háromszög tömbhöz vétele
  - `int Size( )`: a tárolt háromszögek számának lekérdezése
  - `Elem(int i)`: az  $i$ . háromszög lekérése
- A háromszög típus a következő műveletekkel kezelhető:
- `Vector& P(int i)`: Az  $i$ . csúcs koordinátái,
  - `Vector& T(int i)`: Az  $i$ . csúcshoz tartozó uv textúra koordináták
- A `Vector`-on az összeadás, skalárral szorzás műveletek már implementáltak. Ha a megoldás hasonló blokkokból áll, akkor elegendő egyetlen blokkot leírni, és szövegesen utalni arra, hogy a többiben mik lennének az eltérések. *Segítség: a vágás során egy háromszögből négyszög is keletkezhet, amelyet két háromszögre kell bontani. Feltételezheti, hogy a Haromszog típus elbír 4 csúcspontot is.*



## OpenGL

75. A színtér két háromszögből áll, az egyik nem átlátszó, a másik átlátszósága 50%-os. Hogyan lehet megjeleníteni a színteret OpenGL-lel.
76. Mi a billboard ?
80. Miért használ float4 adatformátumot az OpenGL egy pont leírására?
81. Írjon OpenGL programot, amely a földgömböt rajzol fel, mégpedig úgy, hogy az egyes pontok vörös komponense a hosszúsági, a kék komponense pedig a szélességi körrel legyen arányos.
82. Rajzolja fel az OpenGL megjelenítési csővezetékét. Milyen operációkat hajt végre az OpenGL az egyes pontokon?
83. A grafikus kártya a modell-view és projektív transzformáció, vágás, homogén osztás és képernyő-transzformáció után egy  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ ,  $(x_3, y_3, z_3)$  csúcsú háromszög xy vetületét kitölti, és az egyes pixeleket a z koordinátájukkal a z-buffernek továbbítja a takarási feladat megoldásának érdekében. Mennyivel növeli meg a hardver a z értékét, két, ugyanabban a pásztában lévő (közös y koordinátájú), szomszédos (x és x+1 koordinátájú) pixel között?
84. Írja le, hogy mi történik a csúcspont árnyaló (vertex shader) kimenete és a pixel árnyaló (fragment shader) bemenete között egy `GL_TRIANGLES` primitív feldolgozása során! A leírásnak olyan részletesnek kell lennie, hogy egy koordináta geometriában jártas programozó implementálni tudja az egyes lépéseket.
85. Adott egy háromszög képernyő koordináta rendszerben a következő csúcsokkal:  $(0,0,0)$ ,  $(10, 0, 10)$ ,  $(8, 10, 4)$ . Mekkora azon két pont z koordinátáinak különbsége, amelyek az  $(5,5)$  illetve a  $(6,5)$  pixelen keresztül látszanak?
86. Készítsen OpenGL autót! Az autó karosszériája téglatest alakú, hosszúsága 4, szélessége 2, magassága 1 m. Az autónak négy kereke van, amelyek henger alakúak, szélességük 0.5 m, átmérőjük 1 m. A kerek tengelye az autó aljához van rögzítve, az autó elejétől illetve a végétől 1-1 méterre. Az autó az x-y síkon az x tengely irányába 1m/s sebességgel halad, t=0-ban az origóból indul. A kerekek a talajon gördülnek. Írjon egy C nyelvű `Render(float t)` kirajzoló függvényt, amely t időpontban felrajzolja az autót. Feltételezheti, hogy készen áll egy Kocka függvény, amely egységoldalú, origó középpontú, koordinátatengelyekkel párhuzamos élű kockát rajzol ki, valamint egy Henger függvény, amely egység sugarú és magasságú, origóban álló hengert rajzol ki (a körlap az x-y síkon van).
87. Készítsen OpenGL tengert! A tenger egy 100x100 m-es téglalap, amelynek közepén egy kis (pontoszerű) hajó 1 méter magasságú, 1/sec frekvenciájú hullámokat kelt. A hullámok 2 m/s sebességgel haladnak a tenger széle felé. A hullámozás amplitúdója a forrástól vett távolság négyzetével csökken. A tenger képe felülnézetből a hullámok ellenére állandó. Ezt a képet az id azonosítójú OpenGL textúra tartalmazza. Írjon egy C nyelvű `Render(float t)` kirajzoló függvényt, amely t időpontban felrajzolja a tengert! Feltételezheti, hogy a kamera már megfelelően be van állítva. A tenger felületét 2 millió háromszögre tesszellularja fel!
88. A kameraablakunk sarkai 2D világkoordinátákban  $(100, 100)$ ,  $(600, 600)$ . Írja fel azt a transzformációs mátrixot, ami egy x,y 2D pontot normalizált eszközköordináta rendszerbe visz át!
89. Egészítse ki az alábbi programot úgy, hogy a GPU vertex shader-ének 0. regiszterébe egy háromszögekre tesszellularált tórusz csúcspontjai kerüljenek be egymás után úgy, hogy egy `glDrawArrays(GL_TRIANGLES, 0, nVtx)`; hívással a teljes tórusz kirajzolható legyen (15 pont).

```
unsigned int vao, vbo;
glGenVertexArrays(1, &vao);
glBindVertexArray(vao);
glGenBuffers(1, &vbo);
glBindBuffer(GL_ARRAY_BUFFER, vbo);
float * vertices;
int sizeofVertices, nVtx;
...
glBufferData(GL_ARRAY_BUFFER, sizeofVertices, vertices, GL_STATIC_DRAW);
glEnableVertexAttribArray(0);
```

```
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, NULL);
```

90. Magyarozza meg, hogy mit csinál az alábbi programsor: `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);`
91. A raszterizáló egység az alábbi háromszög csúcspontokat kapja meg: (20, 20, 0), (40, 20, 0.5), (40, 40, 1). A hátsólap eldobás teszt (backface culling) szerint ki kell-e tölteni ezt a háromszöget, ha az OpenGL-ben első lapként az óramutató járásával megegyező irány lett kiválasztva? (2 pont) Mekkora lesz a (30, 21) és a (31, 21) koordinátájú pixelekre számított, a z-bufferbe tartó z értékek közötti különbség? Segítség: Triangle setup.
92. A kamera szempozíciója az (1, 1, 2) pontban van, a lookat pont az origóban. A kamera preferált függőleges iránya (0,1,0). Írja fel az kamera View mátrixát, azaz azt a homogén lineáris transzformációt, amely a világot kamerakoordináta rendszerbe képezi le!
93. Milyen árnyalást, fényforrásmodell és optikai anyagmodellt használ az alábbi vertex és fragment shader program? A modellezési transzformációt elfelejtették átadni, ezért a shader-t írja át úgy, hogy a referencia helyzethez képest az objektumot az x tengely mentén kétszeresen nagyítva, majd az y tengely mentén 3 egységet eltolva fényképezze le, valamint valósítson meg diffúz+Phong-Blinn típusú anyagot.

```
// Vertex shader
uniform mat4 VP; // View and Projection Transformation
uniform vec4 kd, ks, ka; // diffuse, specular, ambient ref
uniform vec4 La, Le; // ambient and point sources
uniform vec3 wLiPos; // pos of light source in world
uniform vec3 wEye; // pos of eye in world
uniform float shine; // shininess for specular ref
in vec3 vtxPos; // pos in modeling space
in vec3 vtxNorm; // normal in modeling space
out vec4 color; // computed vertex color

void main() {
    gl_Position = vec4(vtxPos, 1) * VP; // to NDC
    vec3 L = normalize(wLiPos - vtxPos);
    vec3 V = normalize(wEye - vtxPos);
    float cost = max(dot(vtxNorm, L), 0);
    color = ka * La + kd * cost * Le;
}

// Pixel shader
in vec4 color; // interpolated color of vertex shader
out vec4 fragmentColor; // output goes to frame buffer
void main() { fragmentColor = color; }
```

94. Magyarozza meg, hogy mit jelentenek a megvastagított részek:  
`glBufferData(GL_ARRAY_BUFFER, sizeof(vertexCoords), vertexCoords, GL_STATIC_DRAW);`  
`glVertexAttribPointer(0, 2, GL_FLOAT, GL_FALSE, 0, NULL);`
95. Milyen árnyalást, fényforrásmodell és optikai anyagmodellt használ az alábbi vertex és fragment shader program? Írja át úgy, hogy Phong árnyalást valósítson meg, mégpedig távolság négyzetével csökkenő pozicionális fényforrással.

```
// Vertex shader
uniform mat4 MVP, M, Minv; // MVP, Model, Model-inverse
uniform vec4 kd, ks, ka; // diffuse, specular, ambient ref
uniform vec4 La, Le; // ambient and point sources
uniform vec3 wLiPos; // pos of light source in world
uniform vec3 wEye; // pos of eye in world
uniform float shine; // shininess for specular ref
in vec3 vtxPos; // pos in modeling space
in vec3 vtxNorm; // normal in modeling space
out vec4 color; // computed vertex color

void main() {
    gl_Position = vec4(vtxPos, 1) * MVP; // to NDC
    vec4 wPos = vec4(vtxPos, 1) * M;
    vec3 L = normalize(wLiPos);
```

```

vec3 V = normalize(wEye - wPos.xyz);
vec4 wNormal = Minv * vec4(vtxNorm, 0);
vec3 N = normalize(wNormal.xyz);
float cost = max(dot(N, L), 0);
color = ka * La + kd * cost * Le;
}

```

**// Pixel shader**

```

in vec4 color;           // interpolated color of vertex shader
out vec4 fragmentColor; // output goes to frame buffer
void main() { fragmentColor = color; }

```

96. Írja át az alábbi vertex és fragment shader programot úgy, hogy a bemenetre küldött tetszőleges 2D háromszögháló képén azon pontok legyenek pirosak, amelyek a p1, p2, p3 2D háromszög belsejében vannak, és azok kékek, amelyek kívül esnek a p1, p2, p3 2D háromszögön. Természetesen a 2D háromszögháló csúcspontjai ugyanabban a koordináta-rendszerben adottak, mint a p1, p2, p3.

**// Vertex shader**

```

in vec2 vtxPos;           // pos in modeling space
out vec4 color;          // computed vertex color

```

```

void main() {
    gl_Position = vec4(vtxPos, 0, 1);
    color = vec4(1, 1, 1, 1);
}

```

**// Pixel shader**

```

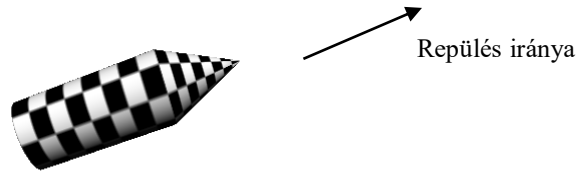
uniform vec2 p1, p2, p3; // háromszög három csúcsa
in vec4 color;           // interpolated color of vertex shader
out vec4 fragmentColor; // output goes to frame buffer
void main() { fragmentColor = color; }

```

Segítség: Ray-triangle intersection.

## Animáció

97. Írjon Animate függvényt C++-ban, amely egy galaxist égitestjeinek a helyzetét (newtoni) fizikai animációval számítja ki a  $t$  időpillanatra. Összesen 100 égitest van, amelyek tömegei az  $m[100]$  tömbben, sugaraik az  $r[100]$  tömbben vannak. Az égitestek kezdeti helyzetei  $(x[100], y[100], z[100])$  tömbökben találhatóak. A  $t=0$  időpillanatban az égitestek állnak. Az égitestek ütközése teljesen rugalmatlan (figyelem, az impulzus megmarad!), azaz ütközéskor összetapadnak, inentől úgy tekintjük őket, mint egy gömböt, amelynek sugara a találkozó gömbök sugaraiknak összege. A Newton féle gravitációs állandó  $f$ .
98. Adott egy kör alakú biliárdasztal. Az asztal középpontjában vesszük fel a koordinátarendszerünket. Az asztal sugara  $R$ . Írjon Animate függvényt C++-ban, amely kiszámítja egy biliárdgolyó helyzetét a  $t$  időpillanatra, feltételezve, hogy  $t=0$ -ban a golyót az  $(x_0, y_0)$  pontból és  $(v_x, v_y)$  kezdősebességgel indítottuk el, csak ez az egyetlen golyó az asztalon, a golyó az asztal oldalával tökéletesen rugalmasan ütközik, és a súrlódást elhanyagoljuk.
99. Egy labda pattogását a magasságfüggvény kulcspozícióival (keyframe) adjuk meg:  
 $t=0, y=0; t=1, y=1; t=3, y=0; t=6, y=1;$   
Mi a labda  $y$  koordinátája  $t=2$ -kor, ha Catmull-Rom spline-t használunk interpolációra.
100. Egydimenziós mozgást kulcskeret animációval definiál. A tárgy a  $t=0$  időpillanatban az  $x=0$  pontban, a  $t=1$  időpillanatban az  $x=1$  pontban, a  $t=2$  időpillanatban megint az  $x=0$  pontban van. A mozgás nyugalmi helyzetből indul és így is fejeződik be (sebesség zérus). Hol van a tárgy a  $t=0.4$  időpontban, ha Catmull-Rom spline-nal interpolálunk.
101. Egy egység tömegű test tömegközéppontjának pályáját egy síkbeli Bézier görbével adhatjuk meg, amelynek vezérlőpontjai:  $(0,0), (1,1), (2,0)$ . Mekkora eredő erő hat a testre a  $t=0.33$  időpillanatban?
102. Egy labdát szeretne animálni **kulcskeret animációval**, **Catmull-Rom spline** alkalmazásával. A kulcskeretek a következők ( $t$ -vel az időt jelöljük és másodpercben értelmezzük):  
 $t=0$ : a labdát a  $[0, 1, 0]$  pontból  $[1, 1, 0]$  sebességgel elhajítjuk.  
 $t=1$ : a labda az  $[1, 2, 0]$  pontban repül  
 $t=2$ : a labda a  $[2, 0, 0]$  pontba csapódik  $[1, -1, 0]$  sebességgel és tökéletesen rugalmatlanul ütközik a talajjal.
103. Egy háromszög csúcsainak koordinátái a lokális modellezési koordinátarendszerben  $[0,0,0], [1,0,0], [0,1,0]$ . Animálja a háromszöget keyframe (kulcskeret) animáció segítségével úgy, hogy a háromszög orientációja állandó maradjon, a pozíciót pedig a kulcskeretekből lineáris interpolációval számítsa ki! Tételjeze fel, hogy a kulcskeretek időértékei a **float t[]** globális tömbben, az első csúcs pozíciója pedig ezekben a kulcskeretekben a **Vector v[]** tömbben vannak. A kulcskeretek száma **int nf**. A rajzolás OpenGL hívásokkal végezze el. Feltételezheti, hogy IdleCallback-ként az Idle függvényt regisztráltuk, tehát csak az Idle-t kell megírni, az inicializálási és kamerabeállító részt nem. A rajzolást **GL\_TRIANGLE** beállítással végezze, a háromszög színe fehér. Az alkalmazás indítása óta eltelt időt **glutGet(GLUT\_ELAPSED\_TIME)** hívással kérdezheti le. Mit és hogyan kell változtatni a programban ahhoz, hogy ne lineáris, hanem Catmull-Rom spline interpolációval dolgozzon (a Catmull-Rom spline a Kochanek Bartels spline speciális esete, amikor a tau zérus, beta pedig 0.5)?
104. Az alábbi lövedék röppül a hegyes végét elől tartva (úgy, ahogy egy tisztességes lövedéktől elvárható) a 3D virtuális térben (lásd a lenti rajzot). A lövedék hengeres részének magassága 4 m, a kúpos részének magassága 2 m, átmérője 1.5 m. A lövedék végének középpontja az  $[x_0, y_0, z_0]$  pontról indul  $[v_x, v_y, v_z]$  kezdősebességgel. A lövedék az ágyúcső hornyolása miatt  $f$  [fok/sec] szögsebességgel forog a főtengelye körül. A nehézségi gyorsulás  $g$  [m/sec<sup>2</sup>], a közegellenállás elhanyagolható, ütközés nincs. Írjon C függvényt, amely képletanimációval meghatározza a  $t$  pillanatban érvényes mozgásállapotot, és OpenGL függvényhívások segítségével fel is rajzolja a lövedéket. A rajzolóhoz felhasználhatja a Henger() függvényt, amely egy 1 m magasságú, origo középpontú, z tengelyű, 1 m átmérőjű hengert jelenít meg, és a Kúp() függvényt, amely egy xy síkon álló, 1 m magas, 1 m átmérős alapkörrel rendelkező kúpot rajzol fel.



105. Egy pontszerű test kétdimenziós mozgását Catmull-Rom spline-nal adja meg. A kulcspontok:

$t = 0$ :  $x = 0$ ,  $y = 0$ ,  $v_x = VX$ ,  $v_y = VY$

$t = 1$ :  $x = VX$ ,  $y = H$ ,

$t = 2$ :  $x = 2 * VX$ ,  $y = 0$ ,  $v_x = VX$ ,  $v_y = -VY$

A  $VX$ ,  $VY$ ,  $H$  értékek konstansok. Adja meg az  $x(t)$ ,  $y(t)$  mozgásgörbék algebrai alakján a  $[0, 1]$  időintervallumban!

106. Marcus Aurelius a barbárok ellen hadakozik. A barbárok kőhajítókkal támadnak, Marcus bátran és mozdulatlanul áll. A sziklák lényegesen kövérebbek Marcusnál, tehát nem lehetne Marcus belsejében elrejteni azokat, viszont nem feltétlenül magasabbak, mint Marcus. Készítsen C++/OpenGL függvényt, amely megjeleníti Marcust és a sziklát a képernyőn, és diszkrét idejű ütközésetektálási eljárással eldönti, hogy a szikla eltalálja-e Marcust. A függvény bemeneti paraméterei:

- Marcust definiáló háromszögek tömbje (`Tomb hMarcus`), Marcus jelenlegi állapotában,
- a sziklát definiáló háromszögek tömbje (`Tomb hSzikla`), a szikla referencia állapotában,
- a szikla jelenlegi helyzetét és orientációját definiáló homogén lineáris transzformáció (`float tSzikla[4][4]`).

Akkor mondjuk, hogy Marcust a szikla eltalálja, ha a szikla AABB-je (axis-aligned bounding box) Marcus valódi geometriájával ütközik (azaz az AABB Marcus bármely belső pontját tartalmazza). Feltételezheti, hogy Marcust definiáló háromszögháló zárt felület. Ütközés esetén a függvény TRUE értékkel tér vissza, egyébként FALSE-zal. Az ütközés felismeréshez használja fel az előző feladat `Vagas` függvényét. Marcus textúra azonosítója `tMarcus`, a szikláé `tSzikla`. A textúra kép betöltésével és a kamera, illetve az ablak beállításával nem kell foglalkozni.