

# **GeoGL3D v1.0 – térgeometriai szerkesztőprogram**

**Szerző:**  
Rana Dávid

**Komárom**

**2007**

# Tartalom

1.	BEVEZETÉS.....	3
2.	A DELPHI ELŐNYEI ÉS LEHETŐSÉGEI.....	4
3.	AZ OPENGL GRAFIKA.....	5
4.	A „GEOGL3D V1.0“ TÉRGEOMETRIAI SZERKESZTŐPROGRAM .....	6
4.1.	A SZERKESZTŐPROGRAM BEMUTATÁSA .....	6
4.2.	A SZERKESZTŐPROGRAM HASZNÁLATA.....	8
4.3.	MENÜPONTOK .....	9
4.4.	ADATLAP.....	12
4.5.	AZ OBJEKTUMOK SZERKESZTÉSEINEK LEHETŐSÉGEI.....	13
4.6.	A SZERKESZTŐPROGRAM ÉS AZ OPENGL KAPCSOLATA .....	15
5.	FELHASZNÁLT IRODALMAK ÉS INTERNETES FORRÁSOK .....	18

# 1. Bevezetés

A "GeoGL3D v1.0" Delphi2005 (Opengl)-ben megírt térgeometriai szerkesztőprogram, mely a különböző sík-, ill. téridomok megjelenítésére szolgál.

Az első és legfontosabb dolog, amit a szerkesztőprogramról tudni érdemes az, hogy bázispontokat vehetünk fel, azokra objektumokat. Pl. két pontra szakaszt, egyenest, kúpot, hengert, stb. A legfontosabb a számítógépes szerkesztésben az, hogy a kész szerkesztés elemei mozgathatók, miközben a származási viszonyok megmaradnak. Különféle színeket és vonalvastagságokat használhatunk a szebb ábra és a jobb áttekinthetőség érdekében. Továbbá lehetőség van az objektumok láthatóságát módosítani, így az egyes objektumok áttetszők, ill. teljesen elrejtethők. Az elkészült szerkesztéseket saját fájlformátumban (\*.g3d) elmenthetők.

Az alakzat megjelenítését is szabályozhatjuk. Lehetőség van csúcsok-, élek- és lapok megjelenítésére. Továbbá a koordináarendszer és a segédvonalak módosítására, ill. eltüntetésére. Az alakzatok kirajzolásánál használhatjuk a konstans és a folytonos árnyalást. Konstans árnyalás esetén a rendszer egyetlen csúcspont színével színezi ki az egész objektumot, folytonos árnyalás esetén a csúcspontok színét lineárisan interpolálja. Folytonos árnyalás alkalmazásával el tudjuk tüntetni a képről a görbült felületeket közelítő poligonháló éleit. A kamera mozgását az alap, csúszás, forgatás, +/- eszköztár nyomógombjaival változtathatjuk (vagy menüpontokból). E lehetőségeket kihasználva akár az alakzat belsejében is "sétálhatunk".

## 2. A Delphi előnyei és lehetőségei

A Delphi alapja az Objekt Pascal programozási nyelv, amely az ismert Turbo Pascal objektumos felépítménye. Éppen ezért a Delphiben feltudjuk használni a Turbo Pascal ciklusait, elágazásait, változók alaptípusait, illetve parancsait.

Többek közt azért választottam a Delphi programozási nyelvet, mert a Delphi fejlesztői környezet jobb más hozzá hasonló programozási nyelvtől. A Delphi az egyik legeffektívebb eszköz, mellyel a Windows operációs rendszer alatt alkalmazásokat hozhatunk létre. Az alkalmazás létrehozásának maximálisan leegyszerűsített fázisa van. A programokat jóval kevesebb idő alatt meg lehet írni Delphiben, mint Turbo Pascalban, a rengeteg vizuális eszköznek és integrált környezetnek köszönhetően.

A Windows alatti programozás eseményekkel irányított programozás. A programozónak csak a rendszer különféle eseményeire kell reagálnia. A program irányítását az operációs rendszer végzi. Jómagam a Delphi 2005-ös verzióját választottam, melyben többek közt megírtam a „GeoGL3D v1.0“ c. térgeometriai szerkesztőprogramot.

### 3. Az OpenGL grafika

Az OpenGL hardverfüggetlen programozási felületet biztosító háromdimenziós grafikus alprogramrendszer, melyet a Silicon Graphics Inc. fejlesztett ki. A GL több programozási nyelvből elérhető függvénykönyvtárakon keresztül. Az SG munkaállomások közismert előnye a gyors és igényes grafika, amit hardver oldalról a grafikus kártyába épített egy vagy több geometriai társprocesszor támogat. Ez a koncepció olyan sikeresnek bizonyult, hogy vezető hardver- és szoftvergyártó cégek (többek között a DEC, IBM, Intel, Microsoft és Silicon Graphics) összefogásával létrehoztak egy munkacsoportot, amely ez alapján specifikálta az OpenGL-t.

A legtöbb munkaállomás kategóriájú számítógép támogatja az OpenGL-t, de a PC- s világban használt WIN'95, WIN'98, Windows NT, Windows 2000 és Windows XP operációs rendszerek alatt az IBM PC (és vele kompatibilis) gépeken is futtathatunk ilyen alkalmazásokat. Természetesen a UNIX operációs rendszernek a PC-ken használatos különböző változatai (pl. Linux) is támogatják az OpenGL alkalmazások futtatását. Az egyes grafikus kártyák eltérő képességűek a beépített geometriai társprocesszorok (grafikai gyorsítók) típusától és számától függően.

Az OpenGL nem geometriai modellezőrendszer, tehát nincsenek benne összetett geometriai objektumok megadására alkalmas parancsok. Az OpenGL nem interaktív rajzoló- vagy grafikus rendszer. Az OpenGL azonban rendkívül alkalmas a geometriai modellezőrendszerek, CAD rendszerek, grafikus rendszerek, de még az interaktív játékprogramok grafikai igényeinek kielégítésére is. Ennek köszönhetően széles körben elterjedt. Az OpenGL-t használó alkalmazások több géptípuson, így minden munkaállomás kategóriájú gépen és a nálunk legnépszerűbb PC-ken is futtathatók, még a programok forrásnyelvi szintű portabilitása is megoldható.

Az OpenGL segítségével egyszerű, térben elhelyezkedő 3D-s grafikus primitíveket jeleníthetünk meg a képernyőn. Ennek során megválaszthatjuk a térből síkba való leképezés módját, az objektumok színét, megvilágítását és mintázatát. A síkbeli képen többféle módon figyelembe vehetjük azt is, hogy az elemek a térbeli elhelyezkedésből következően takarják egymást.

## 4. A „GeoGL3D v1.0“ térgeometriai szerkesztőprogram

A szerkesztőprogram neve a **Geo** (geometria), **GL** (OpenGL), **3D** (3 dimenzió-tér) származik. A **v1.0** a szerkesztőprogram verziójának a számát jelöli. A program megírása Delphi 2005-ben OpenGL parancsok segítségével történt. Célja a különböző sík-, ill. téridomok megjelenítése.

### 4.1. A szerkesztőprogram bemutatása

A matematikaórákon gyakran gondot okoz a tanulóknak, hogy elképzeljék a feladatban szereplő térgeometriai alakzatokat. A szerkesztőprogram segítségével e problémák könnyen orvosolhatóak, hiszen a tanulók az előre elkészített fájlokat megnyitva maguk vizsgálhatják meg a kérdéses testeket, sőt a megfelelő háttérismeret birtokában akár maguk is megszerkeszthetik azokat. Kitűnő segítséget nyújt elsősorban ábrázoló-, ill. analitikus geometria tantárgyakhoz, mivel a programban síkokkal, ill. vektorokkal dolgozhatunk.

A legfontosabb dolog az, hogy „bázispontokat“ kell létrehozunk és ezekre objektumokat. Pl. 2 pontra szakaszt, egyenest, stb. Mivel térben dolgozunk, fontos, hogy az objektumokat bármilyen kameraállásból megtudjuk jeleníteni, ezért a szerkesztőprogramban magunk választhatjuk ki a kamera stílusát. Választhatunk az *alap*, *csúszás* (a térben lévő csúszás a képernyő x,y tengelyéhez viszonyítva), *forgatás* (az OpenGL koordinátarendszer x és y tengelye körül) és +/- (nagyítás, kicsinyítés: a térben előre ill. hátra lévő mozgás) közül. A jobb áttekinthetőséget és egyértelműséget a programban a vonalvastagság, a szín és a láthatóság (átlátszóság) szabályozza. Továbbá kiválaszthatjuk azt is, hogy az objektumokat milyen stílusban szeretnénk megjeleníttetni. Lehetőség van csúcsok-, élek- és lapok megjelenítésére (egyidejűleg csak az egyikre), továbbá a koordinátarendszer- és a segédvonalak megjelenítésére is. Magunk választhatjuk ki az egész „munkaablak“ színét és stílusát. A munkaablak (főablak) tartalmaz eszköztárakat, melyeket szintén a saját munkastílusunkhoz igazíthatjuk, mivel lehetőség van az eszköztárak színének és elhelyezkedésük módosítására, melyek a program indításakor visszaolvasódnak.

## **Adatok a rendszerben:**

A szerkesztőprogram adatbázissal dolgozik, a filekat megtaláljuk a könyvtáron belül különböző mappákban. A mappák tartalma:

„*adat/koordinatarendszer.txt*“ – A koordinátarendszer adatait (a tengelyek láthatósága és vonalvastagsága) tartalmazza. A program indításakor ezek az adatok visszaolvasódnak.

„*adat/segedvonalak.txt*“ – A segédvonalak adatait (beosztási távolságok a tengelyek szerint, vonalvastagság és láthatóság) tartalmazza. A program indításakor ezek az adatok visszaolvasódnak.

„*adat/szinbeallitas.txt*“ – Színek (háttérszín, tengelyek-, segédvonalak-, üzenetablakok színe, ablak háttérszín, sík-, tér- és kamera/megjelenítés eszköztár színe) számát tartalmazza 16-os számrendszerben. A program indításakor ezek az adatok visszaolvasódnak.

„*adat/Geogl3Dv10.chm*“ – HTML Help. A szerkesztőprogram súgója.

„*adat/ujprojekt.g3d*“ – GeoGL3D v1.0-ás fájlformátumú állomány, mely egy üres projektet tartalmaz (3 síkkal, 7 vektorral és 1 ponttal). Ezek az adatok bekerülnek az adatbázisba és kirajzolódnak a program indításakor vagy új projekt nyitásakor.

„*kurzor/\*.cur*“ – Kurzorokat (alap-„*alap.cur*“, csúszás-„*tolas.cur*“, forgatás-„*forgatas.cur*“, nagyítás/kicsinyítés-„*kozeltavol.cur*“) tartalmaz, melyet a szerkesztőprogram a kamera stílusánál használ.

„*kepek/gombok*“ – A szerkesztőprogram nyomógombjainak a képét (\*.bmp) tartalmazza.

„*kepek/szerkeszt/sik*“ – Az adatlap síkidomok információ sorába olvasódnak be a program futása közben.

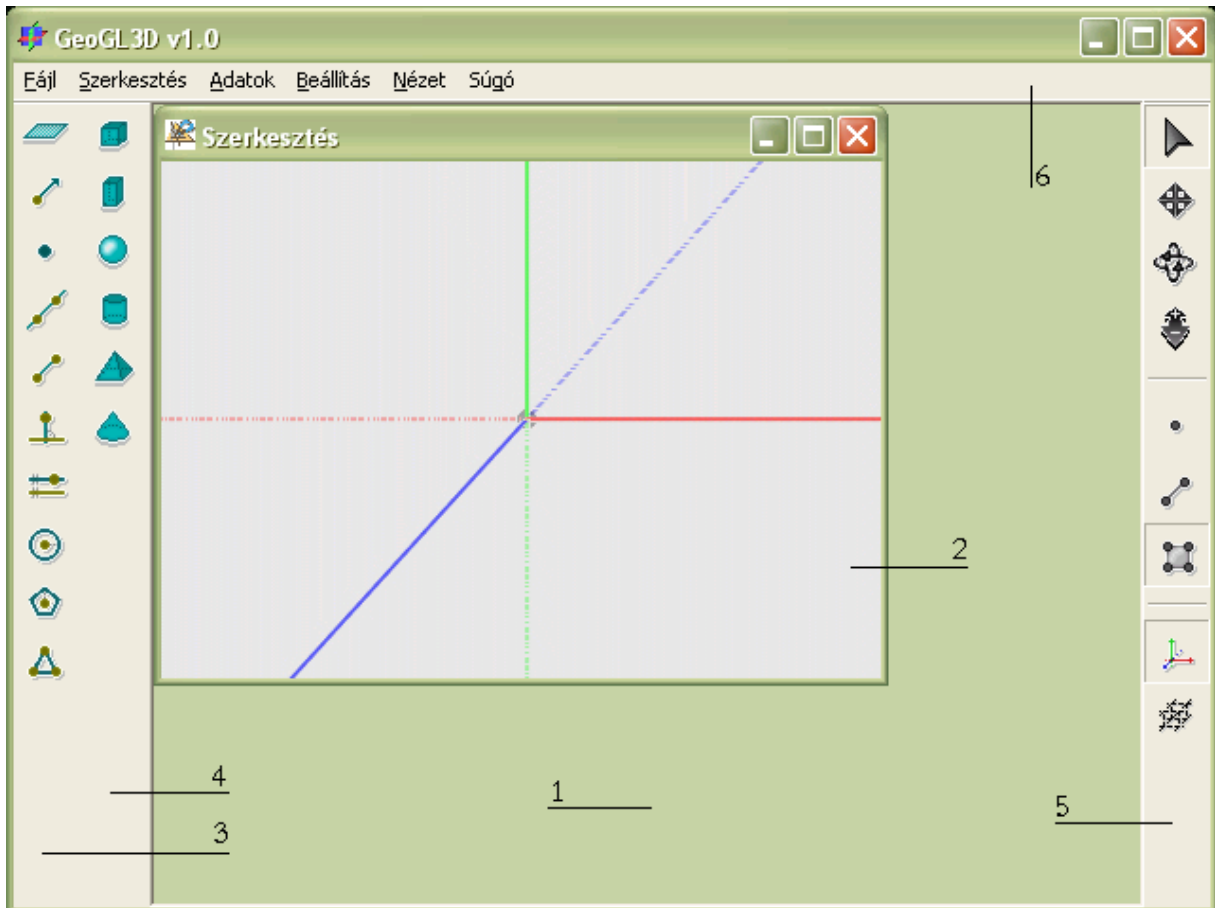
„*kepek/szerkeszt/ter*“ – Az adatlap téridomok információ sorába olvasódnak be a program futása közben.

„*kepek/\*.bmp*“ – A szerkesztőprogramban használt bitmap képek.

„*kepek/\*.ico*“ – A szerkesztőprogramban használt ikonok.

## 4.2. A szerkesztőprogram használata

Az adott meghajtóról a „GeoGL3D v1.0“ mappa teljes tartalmát (GeoGL3Dv1.0.exe, adat-, kepek- és kurzor mappákat) be kell másolni. A szerkesztőprogram indítása után a következőt kell, hogy lássuk:

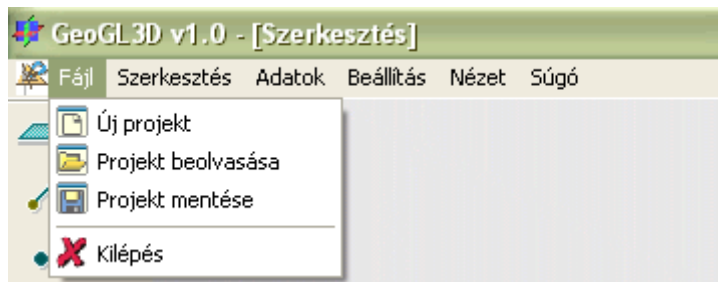


- 1 – Fő ablak: ezen helyezkedik el a szerkesztés ablak, az eszköztárak és a menüsor
- 2 – Szerkesztés ablak: itt rajzolódik ki a szerkesztés. Ez egyben az OpenGL ablaka
- 3 – Sík eszköztár: A síkidomokat tartalmazza
- 4 – Tér eszköztár: A térelemeket tartalmazza
- 5 – Kamera/megjelenítés eszköztár. A kameramódokat és a megjelenítéseket tartalmazza
- 6 – Menüsor

Indításkor a fő- és az OpenGL (szerkesztés) ablak felveszi a képernyő maximális méretét. Mivel a program MDI alkalmazásban íródott, lehetőségünk van a szerkesztés ablak méretein és pozícióján változtatni a fő ablakon belül.



### 4.3. Menüpontok

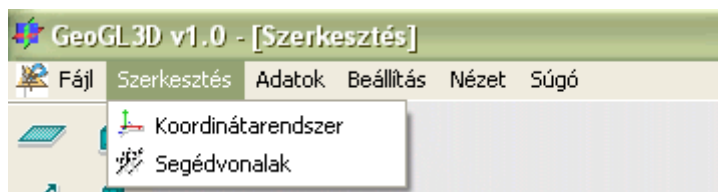


**Új projekt** – Egy új projektet hoz létre, mely a kezdő objektumokat tartalmazza (3 síkot, 7 irányvektort és 1 pontot), melyeket egy külső fájlból olvas be (“adat/ujprojekt.g3d”).

**Projekt beolvasása** – Egy létező GeoGL3D v1.0 –es fájlformátumot (\*.g3d) nyit meg.

**Projekt mentése** – A projektet GeoGL3D v1.0 –es fájlformátumba (\*.g3d) menti el.

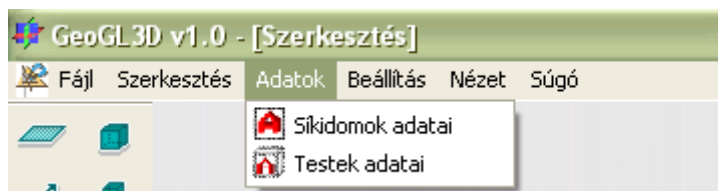
**Kilépés** – Kilép a szerkesztőprogramból.



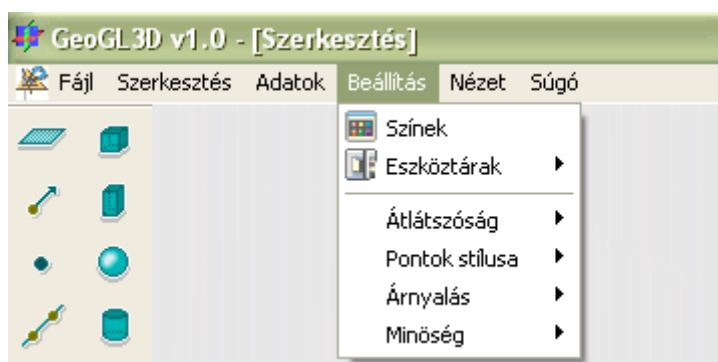
**Koordinátarendszer** – A koordinátarendszer módosítására szolgál. A tengelyek (x,y,z) láthatóságát, ill. vonalvastagságát lehet megváltoztatni. . Az “Eredeti” nyomógomb segítségével visszaállítja az eredeti tulajdonságokat (adatokat).

**Segédvonalak** – A segédvonalak módosítására szolgál. A beosztási távolságot x,y,z tengelyek szerint, a vonalvastagságukat, ill. láthatóságukat lehet megváltoztatni. Az “Eredeti” nyomógomb segítségével visszaállítja az eredeti tulajdonságokat (adatokat).

A megváltoztatott tulajdonságok külső fájlban tárolódnak (“adat/koordinatarendszer.txt”, „adat/segedvonalak.txt”). A szerkesztőprogram indításakor ezek az adatok visszaolvasódnak.



A síkidomok- és a testek adatait jeleníti meg. Táblázat segítségével írja ki az adatbázist (részleges adatbázist), mely a felhasználó számára fontos információkat közvetít. Tájékoztatja a felhasználót az objektumok “adatlapjáról”, vagyis hogy az egyes objektumok mely más objektumok segítségével lettek megszerkesztve, ill. hogy az adott objektum milyen sajátos adatokat tartalmaz (pl. a pont x,y,z koordinátái).



**Színek** – A felhasználó testre szabhatja a szerkesztőprogram munka felületét a színek beállításával. Beállíthatja a háttérszint, ablak háttérszint, x,y,z tengelyek-, segédvonalak-, üzenetablakok-, eszköztárak (sík, tér, kamera/megjelenítés) színét. Az “Eredeti” nyomógomb segítségével visszaállítja az eredeti színeket. A megváltoztatott tulajdonságok külső fájlban tárolódnak (“adat/szinbeallitas.txt”). A szerkesztőprogram indításakor ezek az adatok visszaolvasódnak.

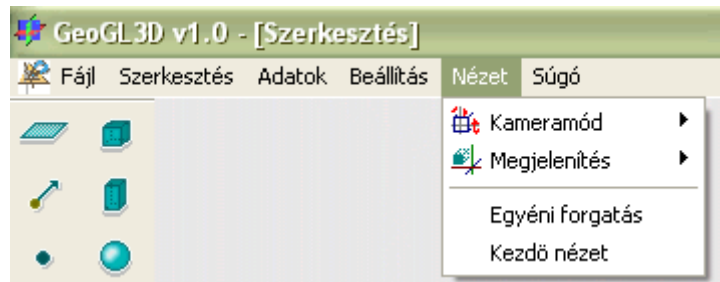
**Eszköztárak** – Az eszköztárak (sík, tér, kamera/megjelenítés) elhelyezésére szolgáló menüpont. A felhasználó maga választhatja ki az adott eszköztár pozícióját (fel, le, jobbra, balra, elrejt).

**Átlátszóság** – Az átlátszóság ki- és bekapcsolása. Ha nincs bekapcsolva az átlátszóság, akkor az összes objektum látható lesz, attól függetlenül hogy a “láthatóság” tulajdonsága milyen értékű.

**Pontok stílusa** – Az összes pont stílusát határozza meg. Lehet kör és négyzet alakú.

**Árnyalás** – Meghatározza, hogy a szakasz és poligon alapelemek hány színben jelenjenek meg. A konstans árnyalásnál egy és a folytonos árnyalásnál sok különböző színben jelenik meg az objektum.

**Minőség** – A kirajzolandó objektum minőségét lehet beállítani. A felhasználó választhat az alacsony, közepes és magas minőségek közül.

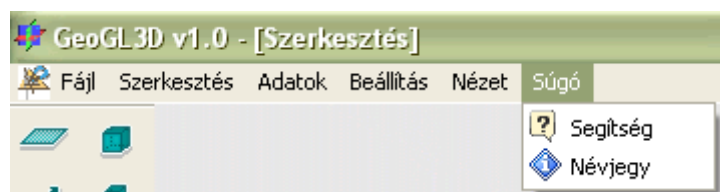


**Kameramód** – A felhasználó 4 kameramód közül választhat: alap, csúszás(a térben lévő csúszást teszi lehetővé a képernyő x,y tengelyéhez viszonyítva), forgatás(a koordináta-rendszer x és y tengelyek körüli forgatást teszi lehetővé), +/- (nagyítás, kicsinyítés: a térben előre ill. hátra lévő mozgást teszi lehetővé).

**Megjelenítés** – Az objektum megjelenítési stílusát teszi lehetővé. A felhasználó szabályozhatja az alakzat megjelenítését. Lehetőség van a csúcsok-, élek- és lapok megjelenítésére. Továbbá kiválaszthatja a koordináta-rendszer és a segédvonalak elrejtését, ill. megjelenítését.

**Egyéni forgatás** – A felhasználó maga választhatja ki a forgatás nagyságát az x és y tengelyek körül a kezdő nézethez viszonyítva.

**Kezdő nézet** – A szerkesztőprogram előre definiált kamera pozíciója. Az x és y tengelyek körül a forgatás nagysága 0.

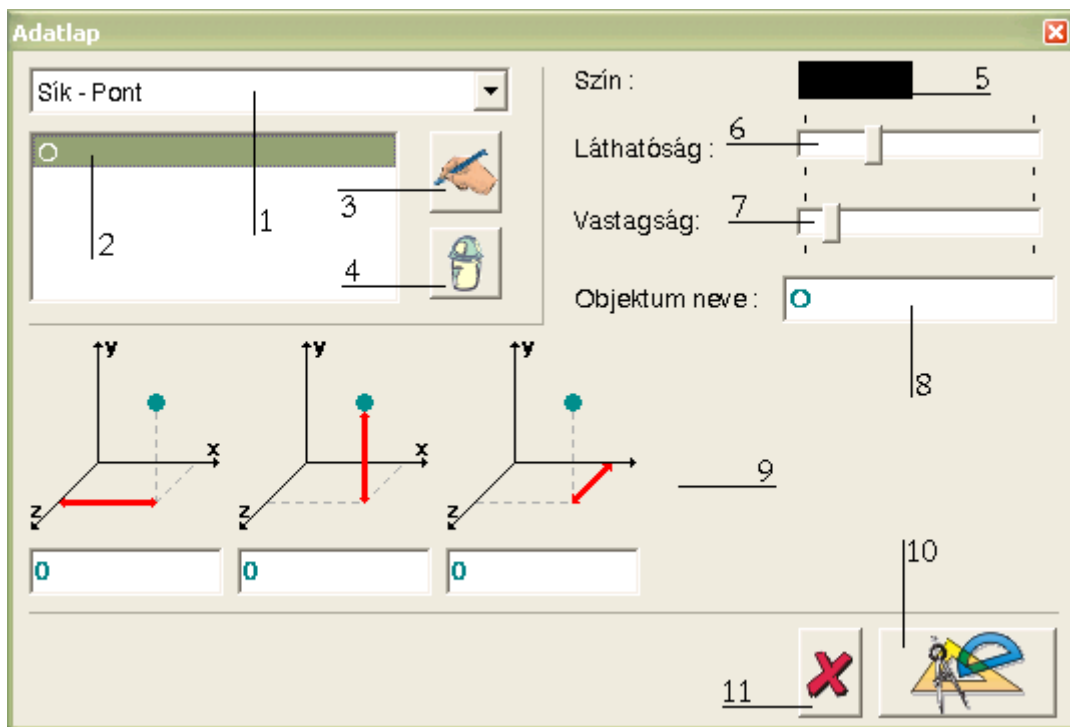


**Segítség** – A szerkesztőprogram felhasználói kézikönyvét jeleníti meg.

**Névjegy** – A szerző adatait jeleníti meg.

## 4.4. Adatlap

A fő ablak, vagy a szerkesztés ablak az egér jobb klikkje után, ill. a sík, vagy a tér eszköztár ikonjai klikkje után jelenik meg az adatlap. Az adatlap tartalmazza az összes objektum tulajdonságát. Vagyis a nevét, színét, láthatóságát, vastagságát, ill. hogy mely más objektum(ok) segítségével lett megszerkesztve, vagy sajátos adatait (pl. pontnál). Lényegében az adatlap tartalmazza az adatbázist.



- 1 – Az objektum kiválasztására szolgál.
- 2 – A kiválasztott objektum neveinek listája. A névre jobb egérgomb klikkelés után a szerkesztés ablakban az objektum piros színű lesz, így tájékoztatja a felhasználót, hogy mely objektum adatlapja látható.
- 3 – Új objektum hozzáadása. Egy új objektum kerül a listába. Az objektumot automatikusan elnevezi, beállítja az objektum színét, láthatóságát és vastagságát az adott objektumtól függően (pl. a síknál a láthatóság kezdő értéke 2).
- 4 – Az objektum törlése. Kitorlí az adott objektumot a listából, továbbá minden olyan objektumot, mely a kitörölendő objektumból lett megszerkesztve (pl. ha egy pont kerül törlésre és ez a pont egy kör középpontja, akkor a kör is törlésre kerül).
- 5 – Az objektum színe.

- 6 – Az objektum láthatósága. 0..10 között vehet fel értéket. Ez egyben az objektum átlátszóságát biztosítja (pl. ha értéke 0, az objektum teljesen átlátszó, tehát rejtve marad).
- 7 – Az objektum vastagsága. A pontnál, egyenesnél, szakasznál, merőleges- és párhuzamos egyeneseknél jelenik meg. 1..10 között vehet fel értéket.
- 8 – Az objektum neve, mely utólag is megváltoztatható.
- 9 – Az adott objektum szerkesztéséhez szolgáló információ sor. Tartalmaz képeket, melyek tájékoztatják a felhasználót az objektum megszerkesztéséhez szükséges információkra.
- 10 – Az objektum jóváhagyása, szerkesztése. Az adatlap adataiból megszerkeszti az objektumot.
- 11 – Elutasítás, bezárás. Az adatlap elrejtésére szolgál.

## 4.5. Az objektumok szerkesztéseinek lehetőségei

Egy objektumot többféleképpen is meg lehet szerkeszteni. A szerkesztőprogramban bázispontokat vehetünk fel és ezek segítségével szerkesszük meg a további objektumokat.



**Sík** – Kétféleképpen szerkeszthetjük meg: 3 különböző ponttal, melyek nem fekszenek egy egyenesen, vagy 1 ponttal és 2 vektorral. Az utóbbinál az információ sor első képe alatt lévő listából kell kiválasztani a pontot, különben hibaüzenetet jelenik meg.





**Vektor** – Kétféleképpen adhatunk meg egy vektort: 1 pont és 1 vektorral, vagy 2 ponttal. Az utóbbinál az információ sor első képe alatt lévő listából kell kiválasztani a pontot, különben hibaüzenetet jelenik meg, továbbá a szerkesztőprogram kiszámolja a vektor koordinátáit és megjeleníti azokat.






**Pont** – Koordinátáival, az x, y és z tengelyek szerint.





**Egyenes, szakasz** – 2 pont segítségével szerkeszthetjük meg.


 **Merőleges-**,  **párhuzamos egyenes** – 1 egyenessel, vagy szakasszal és egy rajtuk kívül fekvő ponttal szerkeszthetjük meg.


 **Kör**,  **sokszög** – 1 ponttal (középpont), sugárral, beosztással (sokszögnél, mely meghatározza az oldalak darabszámát, körnél a minőség értéke a mértékadó) és 1 vektorral (meghatározza hogy az objektum átlóira mely vektor legyen merőleges) szerkeszthetjük meg.


 **Háromszög** – 3 ponttal szerkeszthetjük meg, melyek nem fekszenek egy egyenesen.


 **Kocka** – 2 ponttal (az alaplapp és a fedőlap középpontjai) és 1 vektorral (az alaplappján lévő él iránya) szerkeszthetjük meg.

 **Hasáb** – 2 ponttal (az alaplapp és a fedőlap középpontjai), sugárral (az alaplapp és a fedőlap egyik átlójának a fele), beosztással (az alaplapp és fedőlap oldalainak darabszáma) és 1 vektorral (az alaplappján lévő él iránya) szerkeszthetjük meg.

 **Gömb** – 2 ponttal (meghatározza a sugarat, az első pont a középpont), vagy 1 ponttal (középpont) és sugárral szerkeszthetjük meg.

 **Henger** – 2 ponttal (az alaplapp és a fedőlap középpontjai) és sugárral (az alaplapp és a fedőlap sugara) szerkeszthetjük meg.

 **Gúla** – 2 ponttal (az alaplapp középpontja és a csúcsa), sugárral (az alaplapp egyik átlójának a fele), beosztással (az alaplapp oldalainak darabszáma) és 1 vektorral (az alaplappján lévő él iránya) szerkeszthetjük meg.

 **Kúp** – 2 ponttal (az alaplapp középpontja és a csúcsa), sugárral (az alaplapp sugara) szerkeszthetjük meg.

## 4.6. A szerkesztőprogram és az OpenGL kapcsolata

Az OpenGL segítségével egyszerű, térben elhelyezkedő 3D-s grafikus primitíveket jeleníthetünk meg a képernyőn. A primitívek segítségével bármilyen alakzatot megtudunk jeleníteni, csupán az OpenGL parancsok jól összehangolt sorára van szükség, ezért egy jó működő adatbázis nélkülözhetetlen. A szerkesztőprogram adatbázisa recordokból épül fel, melyek külön-külön tartalmazzák az egyes objektumok adatait, melyeket a GL parancsoknál használtam fel. A henger kirajzolása GL parancsal:

```
... var:henger:GLUquadricObj;
```

```
    gomb:GLUquadricObj;
```

```
    lemez:GLUquadricObj; ...
```

```
... Glucylinder(henger, {alapkör sugara}, {fedőlap sugara}, {magassága}, {alapkör és a fedőlap oldalainak darabszáma}, 1); ...
```

A „Glucylinder“ GL parancsot kihasználva nemcsak hengert, hanem kockát, hasábot, gúlát és kúpot is megtudunk szerkeszteni. A kockánál az alapkör és a fedőlap oldalainak darabszáma 4, hasábnál pedig  $n$  (tetszőleges, a felhasználótól függ) lehet. Gúlánál és kúpnál a fedőlap sugara 0. Ezek után ki kell rajzoltatni az alap- és a fedőlapot:

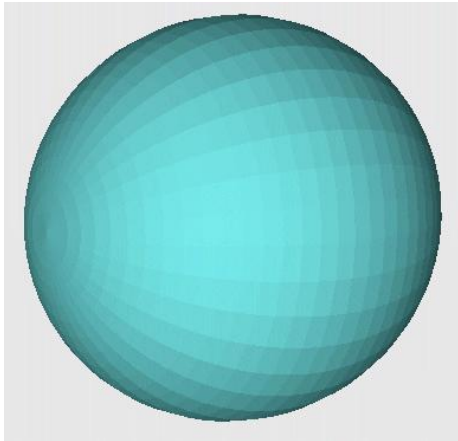
```
... Gludisk(lemez, {középső kör sugara}, {külső kör sugara}, {a lemez oldalainak darabszáma}, 1); ...
```

A „Gludisk“ parancs nélkül üreges objektumot kapnánk. A „lemez“ tulajdonságait a körnél és a sokszögnél is kihasználtam, melyek a síkidomokhoz tartoznak.

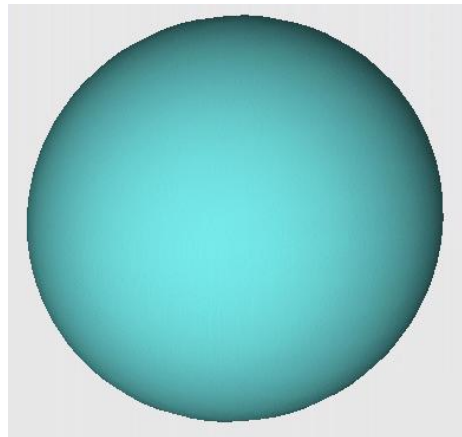
Miután kitöltöttük az adatlapot, az adatokat a program elküldi a számítások unitjába (ez az unit2), ott ezeket az adatokat kihasználva kiszámolja az objektum összes tulajdonságát (pl. egy kockánál ismerjük az alaplappal és a fedőlap középpontjait, ebből megkapjuk a magasságot, a magasságból viszont kiszámoljuk a sugarat, amire szükségünk lesz a kirajzolásnál) és bekerülnek az adatbázisba (rekordokba). A rekordokból kiolvasott adatokat használom fel az OpenGL parancsoknál, növelve ezzel a program gyorsaságát. Tehát amikor az objektumok kirajzolásra kerülnek a GL ablakon (unit1), már az objektum összes tulajdonsága ismert.

A szakasz és poligon alapelemeket megrajzolhatjuk egyetlen színnel (konstans árnyalás), vagy sok különbözővel (folytonos árnyalás). Az árnyalás az összes objektumot érinti, ezt egy gömb felületén a legmegfelelőbb szemléltetni:

`glShadeModel(GL_Flat);` {konstans}



`glShadeModel(GL_Smooth);` {folytonos}



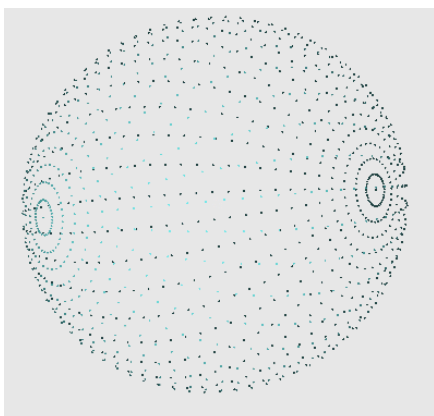
Ezt a lehetőséget a programban egyetlen változó (*arnyalas*) segítségével oldottam meg, melynek típusa byte. Ha az *arnyalas* változó értéke 0 konstans, ha 1 folytonos árnyalást hajt végre a program.

A poligon alapelem megjelenhet egy kitöltött síkidomként, a határát reprezentáló zárt töröttvonalaként vagy a csúcspontjait jelölő pontsorozatként is.

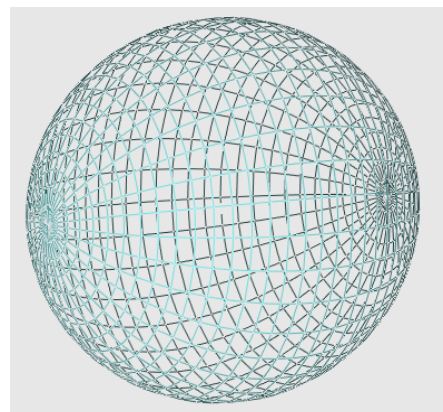
`glPolygonMode(GL_FRONT_AND_BACK, mode );`

A mode értéke `GL_Point`, `GL_Line` vagy `GL_Fill` lehet, mely azt jelzi, hogy a poligonnak csak a csúcspontjait kell megrajzolni, a határát kell megrajzolni vagy a belsejét ki kell tölteni. A `GL_Fill` megjelenítése függ az árnyalástól. A poligon alapelem megjelenítését is egy gömb felületén lehet a legmegfelelőbben szemléltetni:

*csúcspont megjelenítése*



*élek megjelenítése*





Ezt a lehetőséget a programban 3 változó (*csucsok*, *elek*, *lapok*) segítségével oldottam meg, melyeknek típusa byte. A program indításakor a *lapok* változó felveszi az 1 értéket, a többi 0-át. A program figyeli, hogy mely változó értéke módosul 1-re, mert ez alapján rajzolja ki az objektum megjelenítési stílusát. Ha valamely változó értéke 1-re módosult, akkor a többi felveszi a 0-át.

A szerkesztőprogramban lehetőség van a minőség értékének változtatására. Ez azt jelenti, hogy pl. egy kört hány darabra osztjon, vagyis alapköre hány oldalból álljon. Lehetőség van alacsony, közepes és magas minőség választására. Az alacsony minőségénél 20, a közepesnél 35 és a magasnál 50 darabra osztja szét az objektumot. A minőség nagyban befolyásolja a szerkesztőprogram gyorsaságát. A programban ezt egy változó (*minoseg*) segítségével oldottam meg, melynek típusa integer, mely felveszi a 20, 35, vagy 50 értéket. Ez a változó megjelenik a kör, gömb és kúp GL parancsainál.

## 5. Felhasznált irodalmak és internetes források

STOFFA, V.: *Algoritmizáció és programozás.* (Algoritmizácia a programovanie) 1. kiadás, Komárom : Selye János Egyetem, Tanárképző Kar. 2005. 174 s. ISBN 80-969251-7-2

STOFFOVÁ, V. *Databázové systémy v sledovaní úrovne vedomostí študentov.* Hradec Králové : Vysoká škola pedagogická Hradec Králové, 1995. 1. vyd.

STOFFOVÁ, V. et. al.: *Informatika, informačné technológie a výpočtová technika : Terminologický a výkladov slovník.* 1.vyd. Nitra : Fakulta prírodných vied UKF v Nitre, 2001. 230 s. ISBN 80-8050-450-4

OBÁDOVICS J. GYULA : *Matematika.* Tizennyolcadik, javított kiadás, Scholar Kiadó, 1994, ISBN 963-9534-28-5

[www.prog.hu](http://www.prog.hu)

[sulaco.co.za/opengl.htm](http://sulaco.co.za/opengl.htm)

[www.delhipages.com](http://www.delhipages.com)

[iit.bme.hu/~benedek/mirrors/OpenGL/opengl/](http://iit.bme.hu/~benedek/mirrors/OpenGL/opengl/)

[www.delphiforfun.org](http://www.delphiforfun.org)