Érdekes informatika feladatok

XX. rész

Az első OpenGL példaprogram Visual C++-ban

Ha OpenGL programot szeretnénk létrehozni VisualC++-ban, három lehetőségünk van: *Win32 alkalmazás*, *Win32 konzol alkalmazás* és MFC *platformon történő programozás*.

Ha az első kettőt választjuk, akkor a GLUT (OpenGL Utility Toolkit) feladata az ablakozó rendszer kezelése és a grafika megjelenítése. A harmadik esetben az ablakozó rendszert a Visual C++ MFC osztályhierarchiája oldja meg és a grafika egy Windows-os kontrollban jelenik meg.

Jelen példaprogramunkban az első (Win32 alkalmazás) lehetőséget választjuk. Ehhez a következőket kell tenni:

- Elindítjuk a Visual C++ 6.0-ást
- *File* / *New...* / *Projects* / *Win 32 Application* utat járjuk be a menüből kiindulva
- Beírjuk a projekt nevét: *Project name: Elso*
- Beállítjuk a mentési útvonalat.
- OK gomb, majd:
- *A simple Win32 application.*
- Így a következő főprogram-modul jött létre:

```
// Elso.cpp : Defines the entry point for the application.
//

#include "stdafx.h"

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
  // TODO: Place code here.

  return 0;
}
```

- Ha ezzel megvagyunk (a varázsló befejeződött), előjön a Visual C++ programozói felülete, és elkészült a projektnek megfelelő könyvtárstruktúra is.
- Ha nincs OpenGL bekonfigurálva Visual C++ alá, akkor ezt a következőképpen tehetjük meg:
  - Például a *http://www.xmission.com/~nate/glut.html* honlapról töltsük le a *glut-3.7.6-bin.zip* állományt (vagy, ha közben frissítették, akkor az újabb verziót)
  - Kicsomagolás után öt állományt kapunk, amelyből három fontos számunkra: *glut.h*, *glut32.lib*, valamint *glut32.dll.*
  - Ha nincs írásjogunk rendszerkönyvtárakhoz, akkor másoljuk be a *glut.h*-t és a *glut32.lib*-et a projekt könyvtárába, a *glut32.dll*-t pedig a projekt *Debug* könyvtárába.
  - Ha van írásjogunk a rendszerkönyvtárakhoz, akkor véglegesen is feltelepíthetjük az OpenGL-t (így minden projekt tudja használni a fent említett állományokat): másoljuk a *glut32.dll*-t a *Windows* / *system32* könyvtárba, a *glut32.lib*-et a Visual Studio Library könyvtárába (pl. *c:\Program*

Files\Microsoft Visual Studio\VC98\Lib), *glut.h* állománynak pedig hozzunk létre egy saját *GL* nevű könyvtárat a Visual Studio Include könyvtárában (pl. c:\Program Files\Microsoft Visual Studio\VC98\Include\GL\).

- A Visual C++ menüjéből kiindulva, a *Project* / *Settings* beállításoknál, a *Link* fülnél írjuk hozzá a már meglévő *Object/libary modules* sorhoz a következőket: glut32.lib glu32.lib opengl32.lib glaux.lib.
- A fenti főprogram-modul *include* sorába írjuk be az OpenGL headerállományát is: `#include <GL\glut.h>`, vagy `#include "glut.h"`, ha a *glut.h* a projekt könyvtárában van.

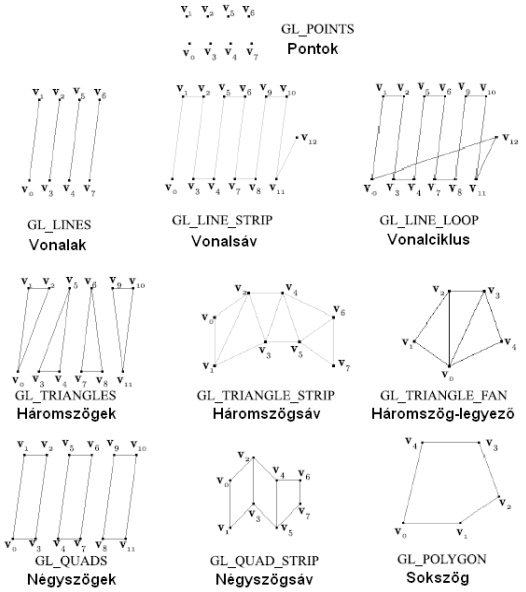
Ha bekonfiguráltuk és használható az OpenGL, akkor megírhatjuk az első példaprogramunkat, amely az OpenGL geometriai primitíveit mutatja be.

A főprogramban a GLUT-re bízzuk az ablakozást: `glutInitDisplayMode` (az ablak beállításai), `glutInitWindowSize` (az ablak mérete), `glutInitWindowPosition` (az ablak bal-felső sarkának a koordinátái), `glutCreateWindow` (az ablak létrehozása). Szintén itt hívjuk meg az OpenGL-t inicializáló függvényt: `init`, majd az eseménykezelő Callback-függvényeit állíthatjuk be. A `glutDisplayFunc`-kal beállított `display` függvény mindig meghívódik az ablak frissítésekor, tehát itt rajzoljunk, a `glutKeyboardFunc`-kal beállított `keyboard` függvény pedig a billentyűzet eseménykezelőjét regisztrálja. A főprogram végén belépünk a fő eseményhurokba: `glutMainLoop`.

Természetesen, a főprogram előtt nekünk kell megírnunk az `init`, `display`, `keyboard` függvényeket.

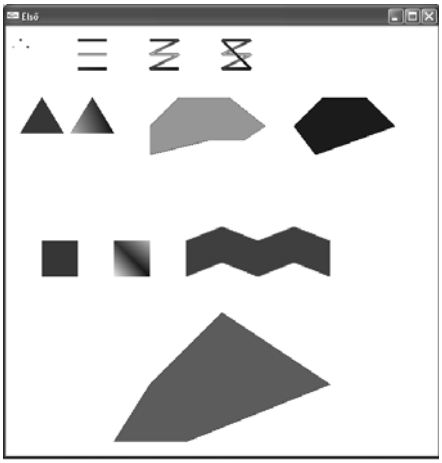
A `display` függvényben történik az effektív rajzolás, itt specifikálhatjuk a vertexeket (csúcspontokat), színeket `glBegin()`, `glEnd()` közé zárva egy-egy primitívet (Begin-End objektum). A primitívek a következők: GL_POINTS, GL_LINES, GL_LINE_STRIP, GL_LINE_LOOP GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_QUADS, GL_QUAD_STRIP és GL_POLYGON.

A primitívek funkcióit és rajzolási módjukat a következő ábra mutatja (figyeljünk a csúcspontok – vertexek specifikálási sorrendjére):



A specifikálás után a `glFlush` paranccsal kényszeríthetjük ki a rajzolást.

A fent elmondottak alapján a program eredménye:



A program a következő:

```
// Elso.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include <GL\glut.h>

void init(void)
{
  glClearColor(0.0, 0.0, 0.0, 0.0); // a törlőszín a fekete
  glMatrixMode(GL_PROJECTION);      // az aktuális mátrix mód a vetítési mátrix
  glLoadIdentity();                 // betölti az egységmátrixot
  gluOrtho2D(-300,300,-300,300);    // párhuzamos vetítés, origó a képernyő közepén
}

void display(void)
{
  glClear(GL_COLOR_BUFFER_BIT);  // letöröljük a képernyőt
  glPointSize(3);                // 3-as nagyságú pontjaink legyenek
  glLineWidth(3);                // 3-as vastagságú egyeneseink legyenek
  glBegin(GL_POINTS);            // pontokat fogunk specifikálni
    glColor3f(1.0, 0.0, 0.0);    // piros szín
    glVertex2i(-280, 280);       // egy pont a (-280, 280) koordinátába
    glColor3f(0.0, 1.0, 0.0);    // zöld szín
    glVertex2i(-290, 270);       // még egy pont
    glColor3f(0.0, 0.0, 1.0);    // kék szín
    glVertex2i(-270, 270);       // még egy pont
  glEnd();                       // több pont nem lesz
  glBegin(GL_LINES);             // vonalakat specifikálunk (hármat)
    glColor3f(1.0, 0.0, 0.0);    // a két pont által meghatározott vonal egyszínű
    glVertex2i(-200, 280);       // első végpont
    glVertex2i(-160, 280);       // második végpont
    glColor3f(0.0, 1.0, 0.0);    // zöld szín
```

```
  glVertex2i(-200, 260);        // első végpont
  glVertex2i(-160, 260);        // második végpont
  glColor3f(1.0, 0.0, 0.0);     // a vonal színét interpolációval számoljuk ki
  glVertex2i(-200, 240);        // első végpont
  glColor3f(0.0, 0.0, 1.0);     // új szín
  glVertex2i(-160, 240);        // második végpont
glEnd();
glBegin(GL_LINE_STRIP);         // vonalsávot specifikálunk (összekötött vonalak)
  glColor3f(1.0, 0.0, 0.0);
  glVertex2i(-100, 280);
  glVertex2i(-60, 280);
  glColor3f(0.0, 1.0, 0.0);
  glVertex2i(-100, 260);
  glVertex2i(-60, 260);
  glColor3f(1.0, 0.0, 0.0);
  glVertex2i(-100, 240);
  glColor3f(0.0, 0.0, 1.0);
  glVertex2i(-60, 240);
glEnd();
glBegin(GL_LINE_LOOP);          // vonalciklust specifikálunk (vissza az elsőhöz)
  glColor3f(1.0, 0.0, 0.0);
  glVertex2i(0, 280);
  glVertex2i(40, 280);
  glColor3f(0.0, 1.0, 0.0);
  glVertex2i(0, 260);
  glVertex2i(40, 260);
  glColor3f(1.0, 0.0, 0.0);
  glVertex2i(0, 240);
  glColor3f(0.0, 0.0, 1.0);
  glVertex2i(40, 240);
glEnd();
glBegin(GL_TRIANGLES);          // egyszínű háromszög
  glColor3f(1.0, 0.0, 0.0);     // piros szín
  glVertex2i(-250, 200);        // egy pont
  glVertex2i(-280, 150);        // még egy pont
  glVertex2i(-220, 150);        // még egy pont
glEnd();
  glBegin(GL_TRIANGLES);        // színháromszög
  glColor3f(1.0, 0.0, 0.0);     // piros szín
  glVertex2i(-180, 200);        // egy pont
  glColor3f(0.0, 1.0, 0.0);     // zöld szín
  glVertex2i(-210, 150);        // még egy pont
  glColor3f(0.0, 0.0, 1.0);     // kék szín
  glVertex2i(-150, 150);        // még egy pont
glEnd();
  glBegin(GL_TRIANGLE_STRIP);   // háromszögsáv
  glColor3f(1.0, 0.5, 0.25);
  glVertex2i(-100, 160);        // első háromszög: v0, v1, v2
  glVertex2i(-100, 120);
  glVertex2i(-60, 200);
  glVertex2i(-20, 140);         // egy pont a következőhöz: v3
  glVertex2i(10, 200);          // egy pont a következőhöz: v4
  glVertex2i(30, 140);          // egy pont a következőhöz: v5
  glVertex2i(60, 160);          // egy pont a következőhöz: v6
glEnd();
  glBegin(GL_TRIANGLE_FAN);     // háromszög-legyező
  glColor3f(0.5, 0.0, 0.0);
  glVertex2i(130, 120);         // első háromszög: v0, v1, v2
  glVertex2i(100, 160);
```

```
    glVertex2i(140, 200);
    glVertex2i(200, 200);          // egy pont a következőhöz: v3
    glVertex2i(240, 160);          // egy pont a következőhöz: v4
  glEnd();
    glBegin(GL_QUADS);             // egyszínű négyszög
    glColor3f(1.0, 0.0, 0.0);      // piros szín
    glVertex2i(-250, 0);           // egy pont
    glVertex2i(-200, 0);           // még egy pont
    glVertex2i(-200, -50);         // még egy pont
    glVertex2i(-250, -50);         // és az utolsó
  glEnd();
    glBegin(GL_QUADS);             // sokszínű négyszög
    glColor3f(1.0, 0.0, 0.0);      // piros szín
    glVertex2i(-150, 0);           // egy pont
    glColor3f(0.0, 1.0, 0.0);      // zöld szín
    glVertex2i(-100, 0);           // még egy pont
    glColor3f(0.0, 0.0, 1.0);      // kék szín
    glVertex2i(-100, -50);         // még egy pont
    glColor3f(1.0, 1.0, 0.0);      // sárga szín
    glVertex2i(-150, -50);         // és az utolsó
  glEnd();
    glBegin(GL_QUAD_STRIP);        // négyszögsáv
    glColor3f(1.0, 1.0, 0.0);
    glVertex2i(-50, 0);            // első négyszög: v0, v1, v2, v3
    glVertex2i(-50, -50);
    glVertex2i(0, 20);
    glVertex2i(0, -30);
    glVertex2i(50, 0);             // két pont a következőhöz: v4, v5
    glVertex2i(50, -50);
    glVertex2i(100, 20);           // két pont a következőhöz: v6, v7
    glVertex2i(100, -30);
    glVertex2i(150, 0);            // két pont a következőhöz: v8, v9
    glVertex2i(150, -50);
  glEnd();
    glBegin(GL_POLYGON);           // sokszöget rajzol
    glColor3f(0.0, 0.5, 0.0);
    glVertex2i(-100, -200);
    glVertex2i(-150, -280);
    glVertex2i(-50, -280);
    glVertex2i(50, -240);
    glVertex2i(150, -200);
    glVertex2i(0, -100);
  glEnd();
  glFlush();                       // rajzolj!
}

void keyboard(unsigned char key, int x, int y)
{                                  //billentyűkezelés
  switch(key)
  {
    case 27:                       // ha escape-et nyomtunk
      exit(0);                     // lépjen ki a programból
      break;
  }
}

// Főprogram
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
```

```
                    LPSTR     lpCmdLine,
                    int       nCmdShow)

{
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
                    // az ablak egyszeresen bufferelt, RGB módú
  glutInitWindowSize(600, 600);
                    // az ablak 600x600-as
  glutInitWindowPosition(100, 100);
                    // az ablak bal felső sarkának koordinátája
  glutCreateWindow("Első");
                    // neve: Első
  init();
                    // inicializálás
  glutDisplayFunc(display);
                    // a képernyő események kezelése (Callback)
  glutKeyboardFunc(keyboard);
                    // billentyűzet események kezelése (Callback)
  glutMainLoop();
                    // belépés az esemény hurokba...
  return 0;
}
```

Kovács Lehel István

honlap-szemle

Honlapszemle

A KvízPart Online (*www.kvizpart.hu*) egy internetes szolgáltatás, amely szórakoztatással, tartalomszolgáltatással, kommunikációs, hirdetési és kereskedelmi tevékenységgel foglalkozik. Szolgáltatásainak túlnyomó többsége ingyenes, de az ingyenesen elérhető oldalak egy részének használatához regisztrálni kell.

Alapszolgáltatásai közé tartoznak a különböző kvízjátékok, levelező program, fórum.

A kvízjátékok során különböző témakörökben több ezer kérdés áll rendelkezésre. A program véletlenszerűen válogatja játékba a kérdéseket nehézségi foktól függetlenül. A kérdéseket adott idő alatt, egyenként kell megválaszolni úgy, hogy a felajánlott 4 lehetséges válasz közül ki kell választani a helytállót. Egy játék során megadott számú kérdést válaszolhatunk meg, ez témakörönként változhat. Témakörönként más lehet az is, hogy hány élet áll rendelkezésünkre, amelyeket rossz válasz megadása esetén elveszítünk. Plusz életeket is gyűjthetünk: ha 15 kérdésre zsinórban helyesen válaszoltunk, akkor 1 plusz életet kapunk! Ha minden kérdést megválaszoltunk, vagy az utolsó életünk is elveszett, a játéknak vége. Ha a játék végén maradt életünk, azt pontra váltja a rendszer.

A KvízPart home oldalán, az oldalmenüben felsorolt kvíztémakörök közül választhatjuk ki azt, amelyikkel játszani szeretnénk. A témakör nyitólapján lévő „Játék indul”