

Technická univerzita v Liberci

Fakulta mechatroniky a mezioborových inženýrských studií

Ústav mechatroniky a technické informatiky

Akademický rok: 2007/08

Grafické a databázové aplikace 2007-2008

„Implementace aplikace DancingSignature“

Zadání

Pro zobrazování je použita knihovna OpenGL. Naprogramujte aplikaci DancingSignature pro zobrazování nápisu - vašeho příjmení nebo iniciál.

DancingSignature

1. Na bílé kreslicí ploše zobrazte daný nápis.
2. Jednotlivá písmena budou složena z primitiv OpenGL (zvolte sami).
3. Každé písmeno se bude otáčet kolem své svislé osy.
4. Celý nápis se bude otáčet kolem své vodorovné osy kolmé k nápisu.
 - Rychlosti rotací zvolte "rozumně".
 - Písmena se otáčejí rychleji než celý nápis.
5. Nasvícení scény je libovolné (musí být ale něco vidět).

----- Poznámka -----

6. Zadání je záměrně velice "volné". Máte tak velkou možnost seberealizace. Pokud vám není zadání srozumitelné, zeptejte se (mailem, konzultací, ...).

Vypracoval: **Martin Blížkovský**
M06000089

Popis realizace

Jako základ pro OpenGL je použita knihovna csgl.dll, která je přidána do programovacího prostředí pomocí Add References. Hlavní část programu, která vytvoří OpenGL okno pomocí výše uvedené knihovny jsem vyhledal na internetu a je pro všechny OpenGL aplikace s touto knihovnou stejná.

Písmena jsem vytvořil pomocí trojúhelníků (GL_TRIANGLES) a čtyřúhelníků (GL_QUADS). Pro příklad jsem písmeno T vytvořil ručně přímo vypsáním jeho souřadnic pro jednotlivé čtyřúhelníky do programovacího prostředí. Každý čtyřúhelník navíc má svoji normálu a materiál, o jejichž funkci se zmíním později. Zbylá písmena jsem vytvořil v programu 3ds max, následně jsem každý model převedl do textového souboru obsahující jednotlivé polygony s příslušnými souřadnicemi bodů, které jsem pak nakopíroval do mé aplikace. Z těchto důvodů jsou zbylá písmena vytvořena z trojúhelníků. Pro každé písmeno jsem vytvořil vlastní funkci pro vyšší přehlednosti a snazší přístup k nim. Jednotlivé funkce volám z funkce glDraw. Písmena na obrazovku rozmísťuji pomocí funkce glTranslatef, kterou knihovna csgl.dll obsahuje. Vzhledem k tomu, že jsem se rozhodl každé písmeno vytvořit jako samostatný model, kde každé má svůj vlastní souřadnicový počátek bylo je snadné pomocí glTranslatef na obrazovku rozmístit přesně na potřebná místa.

Pro rotaci písmen jsem vytvořil zvláštní funkci RotujJ() kde pomocí funkce glRotatef dochází k rotaci písmen kolem osy x a y. Přičemž kolem osy y se točí písmena 6x rychleji než kolem x. Tato funkce se volá těsně před vykreslením každého písmena a tím dochází k rotaci každého písmena kolem osy y a navíc celého nápisu kolem osy x.

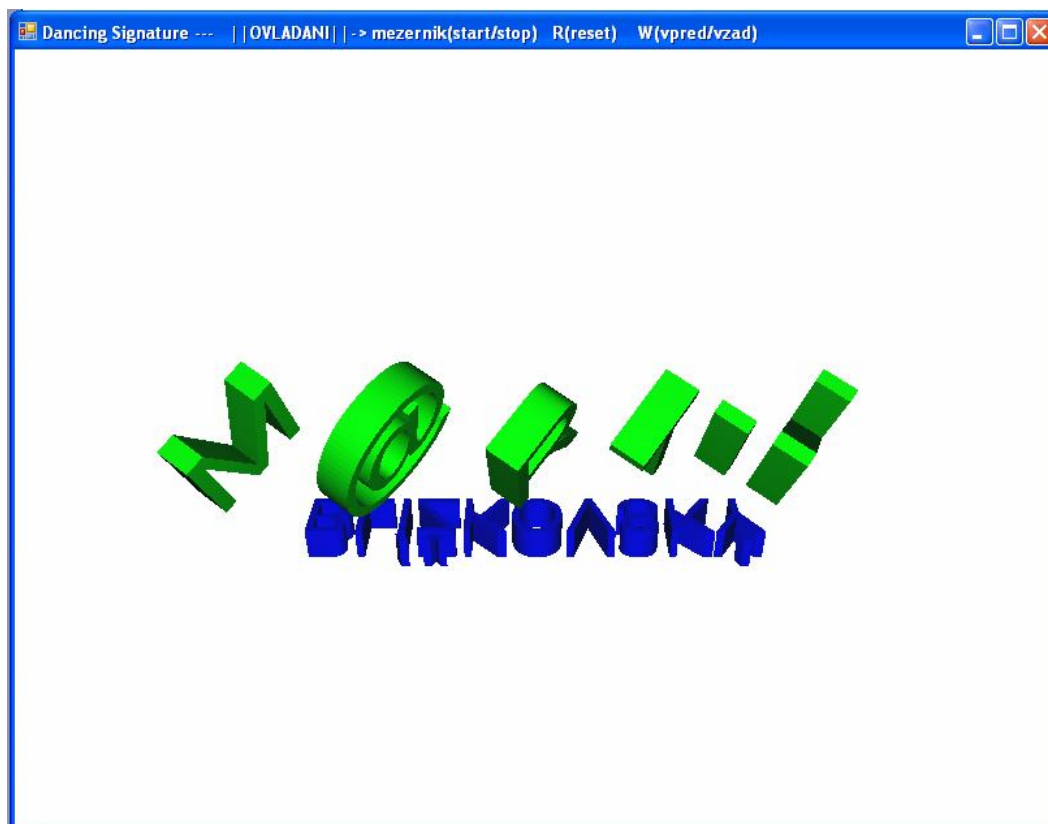
Pro rozpočívání celé animace jsem si vytvořil Timer, který v určitém intervalu volá funkci glDraw a refreshuje okno aplikace. Dále se ve funkci glDraw inkrementuje proměnná rotace, která udává úhel otočení jednotlivých písmen.

Pro nabuzení prostorového efektu jsme všechna písmena vytvořil 3D, ale výsledek ještě nebyl dokonalý, proto jsem se rozhodl scénu osvětlit. Pro osvětlení scény bylo potřeba použít přirozené světlo, jehož intenzita je všude stejná a neudává se jeho pozice (Ambient Light) a rozptylující světlo, které se zachytává na objektech (tvorí stíny) a je potřeba zadat jeho pozici (Diffuse Light). Osvícená scéna, ale nepromítá barvu vertexů pomocí glColor a písmena jsou následkem toho bílá, proto jsem musel vytvořit materiál pomocí glMaterial, který se zobrazí na žádaných objektech po vykreslení scény. Dalším problémem byly stíny. Jelikož plochy (triangly a quady) neměli nadefinované normály, program nemohl správně dopočítat rozložení stínů. Z toho důvodu jsem ke každé ploše musel dopsat ještě její směr (normálu) pomocí glNormal. Tím jsem dosáhl výsledku stínovaných 3D písmen, se kterými jsem byl konečně spokojen.

Celá animace se rozpočívá nebo zastaví stiskem mezerníku, pro změnu směru rotace slouží klávesa W a pro nastavení do výchozí polohy je R.

Navíc je v aplikaci vytvořen 3d model příjmení, který krouží kolem jména pohybujícího se dle zadání.

Kopie obrazovky



Dancing Signature --- || OVLADANI | -> mezernik(start/stop) R(reset) W(vpřed/vzad)

BLÍŽKOVSKÝ

NITR@M

Dancing Signature --- || OVLADANI | -> mezernik(start/stop) R(reset) W(vpřed/vzad)

BLÍŽKOVSKÝ

Výpis zdrojových textů

Deklarace použitých proměnných

```
public float rotuj = 0; //nulování a deklarace úhlu natočení
public bool vpřed = true; //defaultně rotace vpřed
public float[] LightAmbient = { 0.0f, 0.1f, 0.1f, 0.1f };//barva
přirozeného světla

public float[] LightDiffuse = { 1.0f, 1.0f, 1.0f, 1.0f };//barva
rozptylového světla(vytváří stíny)

public float[] LightPosition = { 0.0f, 0.0f, 0.0f, 1.0f };// umístění
světla

public float[] Mat_Jmeno = { 0.0f, 1.0f, 0.0f, 1.0f };//Vytvoření materiálu
pro jméno

public float[] Mat_Prijm = { 0.0f, 0.0f, 1.0f, 1.0f };//Vytvoření materiálu
pro příjmení
```

Popis funkce glDraw

```
public override void glDraw()
{
    GL.glClearColor(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT); //
Vymazání používaných bufferů

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(-80.0f, 0.0f, -250.0f); //přesun písmena na
požadovanou pozici

    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
    M(); //zavolání funkce pro vykreslení písmena

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(-40.0f, 0.0f, -250.0f);
    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
    A(); //zavolání funkce pro vykreslení písmena

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(0.0f, 0.0f, -250.0f);
    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
    R(); //zavolání funkce pro vykreslení písmena

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(30.0f, 0.0f, -250.0f);
    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
    T(); //zavolání funkce pro vykreslení písmena

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(50.0f, 0.0f, -250.0f);
    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
    I(); //zavolání funkce pro vykreslení písmena

    GL.glLoadIdentity();//reset vykreslovací matice (matice modelu)
    GL.glTranslatef(70.0f, 0.0f, -250.0f);
    RotujJ(); // zavolání funkce rotace pro příslušné písmeno
```

```

N(); //zavolání funkce pro vykreslení písmena

GL.glLoadIdentity();
GL.glTranslatef(0.0f, -75.0f, -220.0f);
RotujP(); // zavolání funkce rotace pro příjmení
prijmeni(); //zavolání funkce pro vykreslení příjmení

if (Timer1.Enabled==true) // pokud běží timer
{
    if (vpred) //zapnut chod vpřed
    {
        rotuj += 1.0f; //inkrementace úhlu natočení
    }
    else //chod zpět
    {
        rotuj -= 1.0f; //dekrementace úhlu natočení
    }
}
this.SwapBuffer(); //nahrání dat do bufferu
}

```

Funkce sloužící k rotaci objektů

```

public void RotujJ()
{
    if (rotuj >= System.Math.Abs(360)) rotuj = 0; //nulování úhlu
natočení každých 360 aby někdy náhodou nedošlo k přetečení proměnné
    GL.glRotatef(rotuj, 1f, 0f, 0f); //rotace kolem osy x o úhel
rotuj
    GL.glRotatef(6 * rotuj, 0f, 1f, 0f); //rotace kolem osy y o
úhel 6*rotuj
}
public void RotujP()
{
    GL.glTranslatef(0.0f, 75.0f, 0.0f); //přesun středu rotace
    GL.glRotatef(rotuj, 1f, 0f, 0f); //nastaví kroužení
příjmení kolem jména
    GL.glTranslatef(0.0f, -75.0f, 0.0f); //přesun středu rotace
zpět do středu písmen
    GL.glRotatef(6 * rotuj, 1f, 0f, 0f); //rotace příjmení kolem
osy x
}

```

Příklad vykreslení jednoho čtyřúhelníku z písmene T

```
public void T()
{
    GL.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, Mat_Jmeno);
//nastavení žadaného materiálu, je použit pro všechny plochy v této funkci
    GL.glBegin(GL.GL_QUADS); //začátek kreslení pomocí
quadů (čtyřúhelníků)

    GL.glNormal3f(0.0f, 0.0f, 1.0f); //nastavení normály
následující plochy
    GL.glVertex3f(-2.1f, -16.0f, 5.0f); //bod plochy
    GL.glVertex3f(2.1f, -16.0f, 5.0f); //bod plochy
    GL.glVertex3f(2.1f, 12.2f, 5.0f); //bod plochy
    GL.glVertex3f(-2.1f, 12.2f, 5.0f); //bod plochy

//obdobným způsobem jsou zapsány všechny zbylé plochy a jejich normály
    :
    :
    GL.glEnd(); //ukončení zápisu ploch pro písmeno T
}
```

Inicializace OpenGL okna

```
protected override void InitGLContext()
{
    GL.glClearColor(1.0f, 1.0f, 1.0f, 0.5f); // Barva pozadí
    GL.glClearDepth(1.0f); // nastavení příslušných parametrů pro
hloubku
    GL.glEnable(GL.GL_DEPTH_TEST); // nastavení příslušných
parametrů pro hloubku
    GL.glDepthFunc(GL.GL_LEQUAL); // nastavení příslušných
parametrů pro hloubku

    GL.glHint(GL.GL_PERSPECTIVE_CORRECTION_HINT, GL.GL_NICEST);
// kvalita vykreslování
    GL.glShadeModel(GL.GL_SMOOTH); //jemné vykreslování stínů
    GL.glEnable(GL.GL_LIGHTING); //povolení osvětlení
    GL.glLightfv(GL.GL_LIGHT1, GL.GL_AMBIENT, this.LightAmbient);
// Ambient Light
    GL.glLightfv(GL.GL_LIGHT1, GL.GL_DIFFUSE, this.LightDiffuse);
// Diffuse Light
    GL.glLightfv(GL.GL_LIGHT1, GL.GL_POSITION, this.LightPosition);
// pozice světla
    GL.glEnable(GL.GL_LIGHT1); // zapnutí Light1
    Timer1.Tick += new System.EventHandler(Timer1_Tick);
// vytvoření nového timeru
    Timer1.Interval = 20;
}
protected override void OnSizeChanged(EventArgs e)
{
    Base.OnSizeChanged(e);
    Size s = Size; //proměnná pro velikost okna
    double pomer = (double)s.Width / (double)s.Height; //poměr stran
    GL.glMatrixMode(GL.GL_PROJECTION); //projekční matice
    GL.glLoadIdentity(); // Reset matice modelu
    GL.gluPerspective(45.0f, pomer, 125.0f, 400.0f); //Nastavení
"zoomu", poměru stran okna, nejbližšího a nejvzdálenějšího vykreslovacího
bodů

    GL.glMatrixMode(GL.GL_MODELVIEW); // matice modelu
    GL.glLoadIdentity(); // Reset matice modelu
}
```

Popis sledování stisknutých kláves

```
protected void Prostredi_OnKeyDown(object Sender, KeyEventArgs key)
{
    if (key.KeyCode == Keys.Space) //start a pozastaveni animace
    {
        if (Timer1.Enabled == true)
        {
            Timer1.Enabled = false;
        }
        else
        {
            this.Timer1.Enabled = true;
        }
    }
    if (key.KeyCode == Keys.R) //reset animace do výchozí polohy
    {
        rotuj = 0;
        posun = 0;
        vpred = true;
        glDraw();
        this.Timer1.Enabled = false;
    }

    if (key.KeyCode == Keys.W) // volba směru rotace
    {
        if (vpred)
        {
            vpred = false;
        }
        else vpred = true;
    }
}

private void Timer1_Tick(object sender, EventArgs e)
//casovac pro opakovani vykreslovani
{
    glDraw(); //volani funkce vykreslit
    this.Refresh(); //refresh pohledu
}
```