ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Semestrální práce z předmětu KIV/KPG

Prohlížeč fraktálních terénů

Jméno a příjmení	**Petr Brož**
Studijní číslo	**A03238**
E-mail	**pebro@students.zcu.cz**
Datum vypracování	**4.4.2005**

I ZADÁNÍ

Pokud už máte znalosti z 3D grafiky, např. ze ZPG, můžete si vydělat řešením této úlohy: vytvořte uživatelsky příjemný zobrazovací program pro vizualizaci souborů vytvářených vašimi kolegy v úloze 3, s jednoduchým osvětlením a možností prohlížet si terén z různých stran. Aby měl váš program využití již pro řešení úlohy 3, je zapotřebí jej odevzdat již 14 dní po zadání. Můžete k tomuto účelu upravit i svoji semestrálku na ZPG.

II REALIZACE

Program jsem napsal v jazyce C# s využitím funkcí rozhraní Direct3D. Při načtení vstupního souboru aplikace vytvoří instance tříd VertexBuffer a IndexBuffer. Do VertexBufferu se ze souboru načtou souřadnice jednotlivých vrcholů a přidají se k nim ještě barvy vrcholů, které se dodefinují podle výšky vrcholu. Do IndexBufferu se poté načtou trojice indexů definující tři vrcholy každého vykreslovaného trojúhelníku.

Před každým vykreslením bude (pokud si to uživatel bude přát) zařízení D3D kromě ostatních transformačních matic nastavena matice pro transformaci světových souřadnic – konkrétně pro jejich rotaci kolem Y-ové osy, aby se tak dosáhlo efektu rotace terénu.

Zásadní částí programu je právě načtení informací z textového souboru do VertexBufferu a IndexBufferu. Kód této metody následuje:

```
/// <summary> Privátní metoda pro načtení dat fraktalizovaného terénu. </summary>
private void loadTerrain(String fileName)
{
  TextReader tr = new StreamReader(fileName);       // Vstupní proud pro čtení ze souboru
  CustomVertex.PositionColored[] vertices;  // Pole vrcholů pro zapsání do VB

                      // Zjištění počtu vrcholů a jejich načtení do uzamčené paměti VB
  verticesCount = Int32.Parse(tr.ReadLine());
  vertexBuffer = new VertexBuffer(typeof(CustomVertex.PositionColored),
                                  verticesCount, device, Usage.WriteOnly,
                              CustomVertex.PositionColored.Format, Pool.Default);

  vertices = (CustomVertex.PositionColored[]) vertexBuffer.Lock(0, 0);
  for (int i = 0; i < verticesCount; i++)
  {
       String[] readedWords = tr.ReadLine().Split(' ');
       vertices[i].X = (float) Double.Parse(readedWords[0]);
       vertices[i].Z = (float) Double.Parse(readedWords[1]);
       vertices[i].Y = (float) Double.Parse(readedWords[2]);

       int greenAmount = (int)((2 + vertices[i].Y) * 100);
       if (greenAmount > 255) greenAmount = 255;
       if (greenAmount < 0) greenAmount = 0;
       vertices[i].Color = Color.FromArgb(0, greenAmount, 0).ToArgb();
  }

  vertexBuffer.Unlock();
  indicesCount = Int32.Parse(tr.ReadLine());
  int[] indices = new int[3 * indicesCount];
  indexBuffer = new IndexBuffer(typeof(int), 3 * indicesCount, device, Usage.WriteOnly,
                                Pool.Default);

  for (int i = 0; i < indicesCount; i++)
  {
       String[] readedWords = tr.ReadLine().Split(' ');
       indices[3 * i + 0] = Int32.Parse(readedWords[0]);
       indices[3 * i + 1] = Int32.Parse(readedWords[1]);
       indices[3 * i + 2] = Int32.Parse(readedWords[2]);
  }
  indexBuffer.SetData(indices, 0, LockFlags.None);

  tr.Close();
}
```

III UŽIVATELSKÁ PŘÍRUČKA

Na formuláři aplikace se nachází tlačítko „Load Terrain" pro načtení terénu z připraveného textového souboru a tlačítko „Background Color", které vyvolá dialog pro výběr barev a případnou zvolenou barvu nastaví jako pozadí vykreslovanému terénu. Kromě těchto tlačítek má uživatel k dispozici 3 radiobutony pro výběr stylu vykreslování (vykreslení samostatných bodů, drátěného modelu a plného modelu) a také zaškrtávací políčko pro vypnutí/zapnutí rotace terénu.