POKROČILÉ 3D OBJEKTY V OPENGL

Kurz: **Moderní počítačová grafika**

Lektor: Ing. Michal Švento

Náplň cvičení

1. Kvadriky GLU
2. Export modelu z Blenderu
3. OBJ Loader

Prostorové útvary a GLU

- definice složitějších tvarů s využitím knihovny GLU, která nabízí koncept kvadrik
  - primitivy typu QUAD, které vhodně zvoleným postupem generují základní prostorové útvary
- knihovna GLU sama zajišťuje korektní normálování, texturování a další vlastnosti vertexů

- kvadriky se generují pomocí objektu typu `GLUquadric`
  - vytvoření: `GLUquadric* nazev = gluNewQuadric()`
  - odstranění: `gluDeleteQuadric(nazev)`

Kvadriky a práce s nimi

- vlastnosti kvadrik nastavují následující funkce
  - `gluQuadricOrientation(GLUquadric* nazev, enum orientation)`
    - pro kvadriku danou označením `nazev` nastavuje orientaci vykreslených stěn polygonů
    - projeví se pouze při zapnuté volbě face culling
  - `gluQuadricDrawStyle(GLUquadric* nazev, enum draw)`
    - nastavuje způsob vykreslení stěn
  - `gluQuadricNormals(GLUquadric* nazev, enum normal)`
    - nastavuje způsob generování normál pro osvětlení
  - `gluQuadricTexture(GLUquadric* nazev, enum texture)`
    - nastavuje (či vypíná) generování texturovacích souřadnic

Příklady útvarů

- gluCylinder(GLUquadric* nazev, double base, top, height, int slices, stacks)
  - dutý kónický válec s osou v ose *z*
  - https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/gluCylinder.xml
- gluDisk(GLUquadric* nazev, double inner, outer, int slices, loops)
  - disk v rovině *xy* s osou v ose *z*
  - https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/gluDisk.xml
- gluSphere(GLUquadric* nazev, double radius, int slices, stacks)
  - koule se středem v nule
  - https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/gluSphere.xml

Příklady útvarů

- gluCylinder(GLUquadric* nazev, double base, top, height, int slices, stacks)
- gluDisk(GLUquadric* nazev, double inner, outer, int slices, loops)
- gluSphere(GLUquadric* nazev, double radius, int slices, stacks)

- obecně
  - `nazev` udává, ke kterému objektu je nově vytvořená kvadrika přiřazená
  - parametry typu `double` definují rozměry kvadriky
  - parametry typu `int` definují počet elementů při vykreslování (subdivisions)

Bodovaný úkol 1 [2 b]

1. V kódu `ukol1.cpp` vygenerujte následující kvadriky:
   - kouli se středem [-15, 0, 0] a poloměrem 5
   - využijte `glPushMatrix()` a `glPopMatrix()` spolu s translací pro umístění objektů na správná místa
2. Implementujte do scény osvětlení.

Nápověda:

- texturování lze zapínat/vypínat kdykoliv: `glEnable(GL_TEXTURE_2D)`
  `glDisable(GL_TEXTURE_2D)`
- nastavení aktuální textury: `glBindTexture(GL_TEXTURE_2D, int name)`

Nebodovaný úkol 1 - pokračování

1. V kódu `ukol1.cpp` vygenerujte následující kvadriky:
   - kouli se středem [-15, 0, 0] a poloměrem 5
   - disk se středem [0, 5, 0] a poloměry 5 a 7
   - válec se středem [15, 0, 0], konstantním poloměrem 6 a výškou 10
   - využijte `glPushMatrix()` a `glPopMatrix()` spolu s translací pro umístění objektů na správná místa
2. Implementujte do scény osvětlení.
3. Vyzkoušejte i zobrazení různých textur a změnu parametrů generovaných kvadrik.

Nápověda:

- texturování lze zapínat/vypínat kdykoliv:
```
glEnable(GL_TEXTURE_2D)
glDisable(GL_TEXTURE_2D)
```
- nastavení aktuální textury:
```
glBindTexture(GL_TEXTURE_2D, int name)
```

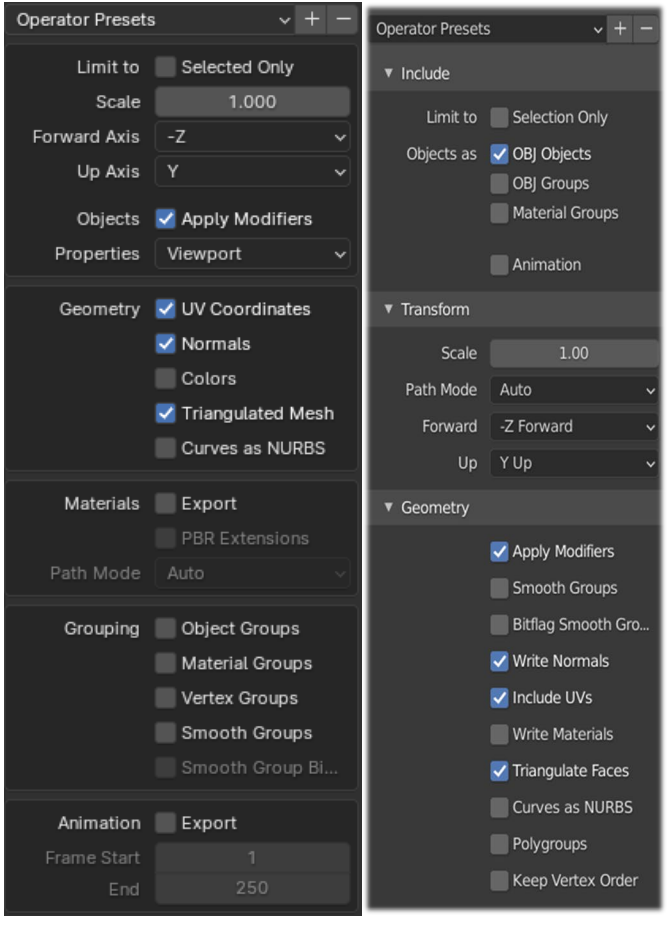
Blender

- 3D modelovací prostředí
- Freeware: https://www.blender.org/
- Existují různé tutoriály, např. zde: https://www.youtube.com/playlist?list=PLa1F2ddGya_-UvuAqHAksYnB0qL9yWDO6

- Vymodelované objekty lze exportovat a následně použít s pomocí OpenGL

Export modelu z Blenderu

- exportujeme soubor .obj
- zapisujeme i normály, aby mohl být objekt osvětlován
- zapisujeme i UV, aby mohl být objekt texturován
- triangulace stěn
  - vzhledem k principu vykreslování v OpenGL musíme vědět, jakou primitivu kreslíme



OBJ Loader

- třída pro načítání souborů .obj do projektu OpenGL
- https://github.com/Bly7/OBJ-Loader
- pro použití stačí stáhnout hlavičkový soubor `OBJ_Loader.h` a v projektu jej načíst
- třída `objl::Loader`
- použití

1. připojení hlavičkového souboru
```
#include "OBJ_Loader.h"
```
2. vytvoření načítacího objektu
```
objl::Loader loader
```
3. načtení
```
loader.LoadFile("path_to_object.obj")
```

OBJ Loader

- struktura dat po načtení
  - načtené sítě: `loader.LoadedMeshes`
    - objekty třídy `objl::Mesh`
    - nemusí být nutně jedna
  - každá síť obsahuje vertexy (a materiálové vlastnosti, které teď nevyužijeme)
    - `objl::Mesh sit`
    - `sit.Vertices`
  - každý vertex má souřadnice, normálu a texturovací souřadnice
    - `Position.X, Position.Y, Position.Z`
    - `Normal.X, Normal.Y, Normal.Z`
    - `TextureCoordinate.X, TextureCoordinate.Y`

Úkol 2

1. Exportujte Vámi vytvořený model v Blenderu jako soubor .obj s patřičným nastavením.
2. Načtěte a zobrazte svůj objekt v rámci VS projektu glut/OpenGL.
3. Objektu přidejte materiál a osvětlení.
   - normálování
   - spojování vertexů do primitiv typu `GL_TRIANGLES`
4. Objektu přidejte texturu.