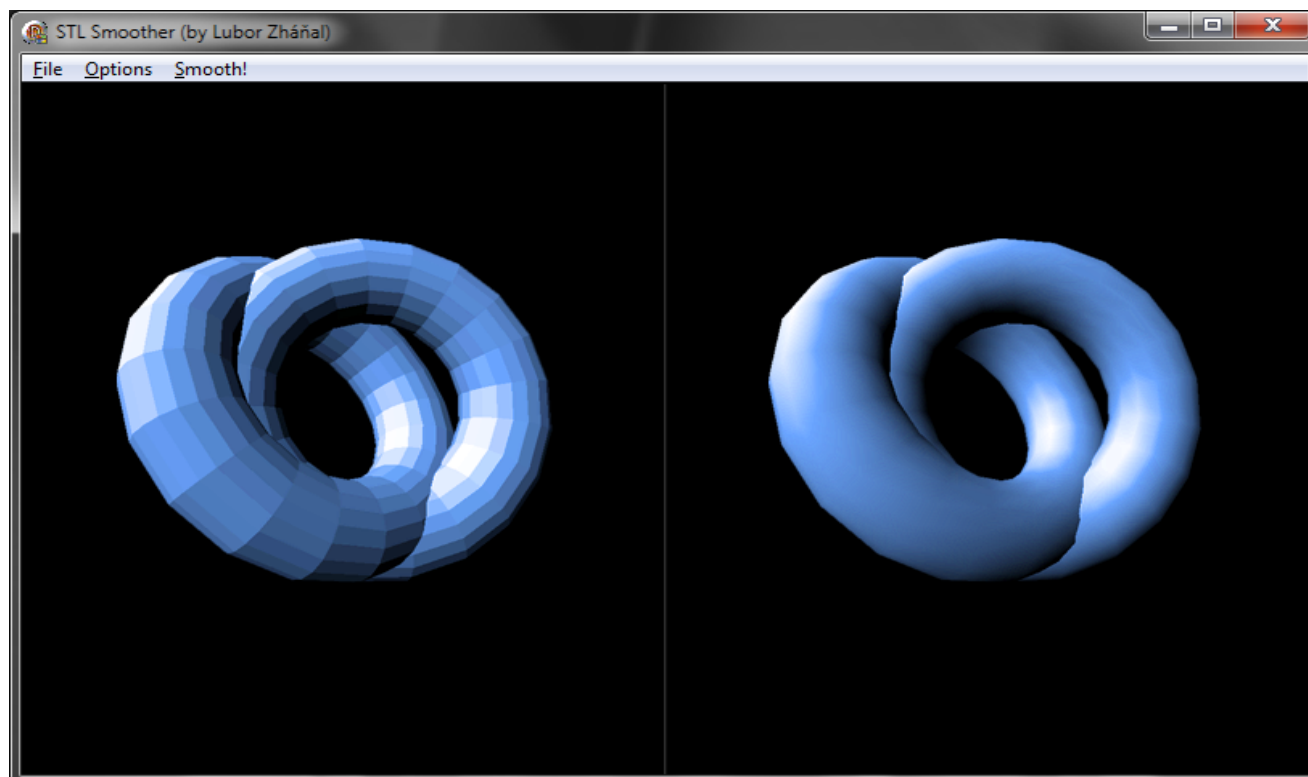


Vysoké učení technické v Brně

Fakulta strojního inženýrství



Ústav automobilního
a dopravního inženýrství



Autorizovaný software

Název autor. sw v originále

STL Smoother for Virtual Reality

Název autor. sw česky

Vyhlazování STL pro virtuální realitu

Id. Číslo (Apollo)

23860

Interní označení

STL Smoother VR

Autoři

1
2
3
4

Ing. Lubor Zháňal

Datum

24.6.2009

Popis v originále

STL Smoother can be used for analysis and improvement of STL models. It can compute a missing normal vectors on all of vertex points. A region-based algorithm can be used for a calculation speed-up on high detail models.

Popis česky

Aplikace STL Smoother slouží ke analýze a vylepšení geometrie STL modelů prostřednictvím dopočítávání chybějících normálových vektorů ve všech vrcholech polygonální sítě. Pro zrychlení výpočtu je možno u složitějších modelů provádět výpočet v pohyblivém regionu.

Klíčová slova v originále

geometry smoothing, STL, 3D, virtual reality

Klíčová slova česky

vyhlazování geometrie, STL, 3D, virtuální realita

Parametry technické

Aplikace je napsána ve vývojovém prostředí Turbo Delphi. Optimalizované matematické rutiny jsou napsány v assembleru pro platformu x86.

Parametry ekonomické

Software je zdarma k dispozici všem uživatelům po domluvě s autorem aplikace.

Využití mimo autorský kolektiv

Aplikace je využívána na učebně virtuální reality VUT a také partnerské ČVUT pro zpracování a zvýšení kvality prohlížených STL souborů.

Předáno za projekt

MSM 0021630518 Simulační modelování mechatronických soustav

Rok

2009

Kontaktní osoba

Ing. Lubor Zháňal

Místnost

A1/622

Telefon

kl. 2264

The screenshot shows the Turbo Delphi IDE with the following components:

- Structure View:** Shows a project named "STLSmoothing - Turbo Delphi - qOGL" containing folders for Classes, Procedures, Types, Variables/Constants, and Uses.
- Object Inspector:** Displays properties for the selected component, "Form1 Tform1". Key properties include:
 - Action: Action
 - ActiveControl: alNone
 - Align: False
 - AlphaBlend: False
 - AlphaBlendValue: 255
 - Anchors: [akLeft, akTop]
 - AutoScroll: False
 - AutoSize: False
 - BIDMode: bdLeftToRight
 - BorderIcons: [biSystemMenu, biMinimize]
 - BorderStyle: bsSizeable
 - BorderWidth: 0
 - Caption: er (by Lubor Zhaňal)
 - ClientHeight: 465
 - ClientWidth: 800
 - Color: clBtnFace
 - Constraints: (TSizeConstraints)
 - Ctl3D: True
 - Cursor: crDefault
 - DefaultMonitor: dmActiveForm
 - DockSite: False
 - DragKind: dkDrag
 - DragMode: dmManual
 - Enabled: True
 - Font: (TFont)
 - FormStyle: fsNormal
 - Height: 523
 - HelpContext: 0
 - HelpFile: htContext
 - HelpKeyword: htContext
 - Hint: (TControlScrollBar)
 - Icon: (None)
 - KeyPreview: False
 - Left: 0
 - Margins: (TMargins)
- Main Editor:** Contains the source code for "qGLoadShaders.pas". The code defines several procedures and functions for loading and managing OpenGL shaders. Notable features include:
 - qglLoadShaders(VSFileName, FSFileName): Cardinal;**: A function that loads vertex and fragment shaders from files and compiles them.
 - qglSwitchMode2D3D(Mode2D: Boolean);**: A procedure that switches between 2D and 3D rendering modes, disabling/enabling depth testing and setting up projection matrices.
 - qglUnProjectXY(X, Y: Integer; var WS: TVector);**: A procedure for unprojecting screen coordinates (X, Y) into world space coordinates (WS).
 - qglUnProjectXYZ(X, Y: Integer; Z: Single; var WS: TVector);**: A procedure for unprojecting screen coordinates (X, Y) along with a depth value (Z) into world space coordinates (WS).

