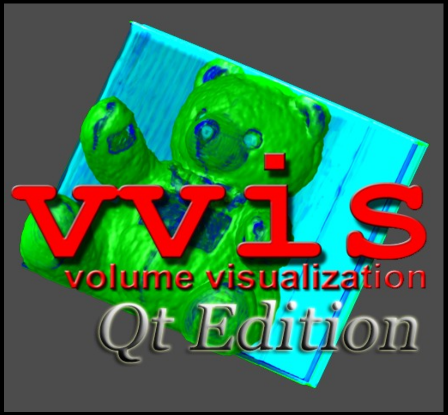


Qt-Interface For Volume Visualization

Practical Course Computer Graphics For Advanced
Supervising
Dr. Susanne Krömker

Stefan Becker & Ronald Lautenschläger

Outline

1. Terms you should know
2. Comparison of GUI Designers
3. QtOpenGL Module
4. Callback methods
5. Threads
6. Qt vs. GTK+ (overview)
7. Boy‘s Surface
8. QTvvis
9. Problems during porting
10. Summary
11. Sources

OpenGL [Wikipedia]

- OpenGL (Open Graphics Library) is a specification defining a cross-language cross-platform API for writing applications that produce 3D and 2D computer graphics
  - Interface of over 250 different function calls
  - competes with Direct3D on Microsoft Windows platforms
  - widely used in CAD, virtual reality, scientific visualization, information visualization, and video game development.

GTK(MM) [Wikipedia]

- C++ interface for the GUI library GTK+
- typesafe callbacks
- widgets extensible via inheritance
- You can create user interfaces either in code or with the Glade Interface Designer, using libglademm
- gtkmm is free software distributed under the GNU Library General Public License (LGPL)

Qt [Wikipedia]

- Cross-platform graphical widget toolkit for the development of GUI programs
- Produced by the Norwegian company Trolltech
- Qt uses an extended version of the C++ programming language
- Bindings exist for Python, Ruby, C, Perl and Pascal
- It runs on all major platforms
- Features
  - SQL database access
  - XML parsing
  - thread management

Qt [Wikipedia]

Platforms:

- **Qt/X11** — Qt for X Window System
- **Qt/Mac** — Qt for Apple Mac OS X
- **Qt/Windows** — Qt for Microsoft Windows
- **Qt/Embedded** — Qt for embedded platforms (PDA, Smartphone, ...)
- **Qt Console** — edition for *non-GUI* development
- **Qt Desktop Light** — entry level GUI edition, stripped of network and database support
- **Qt Desktop** — complete edition
- **Qt Open Source Edition** — complete edition, for open-source development

Qt - Elements

- **QtCore** — Core Classes used by other Qt-Modules
- **QtGui** — Component for GUI creation
- **QtOpenGL** — OpenGL support
- **QtDesigner** — Qt-Designers extension classes
- **Qt3Support** — Special classes to ensure compatibility to Qt 3
- **qmake** — generator for makefiles
- **moc** — creates meta-information of classes used in program
- **uic** — creates c++ code from UI-Classes
- **rcc** — resource-Compiler

Qt - Hierarchy

- QObject
  - Base class of all Qt-Objects
- QWidget
  - Inherits from QObject
  - Base class of all Widgets

Glade Interface Designer

- Glade Interface Designer is a graphical user interface creator for GTK+
- programming language–independent
- does not produce code for events
- created XML file, and optionally one or more C programming language files into which programmers insert their code.
- free software, distributed under the GNU General Public License

Glade Interface Designer

XML Glade-GUI description example:

```
<?xml version="1.0" standalone="no"?> <!--*- mode: xml -*-->
<!DOCTYPE glade-interface SYSTEM "http://glade.gnome.org/glade-2.0.dtd">

<glade-interface>

<widget class="GtkWindow" id="windowToVrend">
  <property name="title" translatable="yes">window1</property>
  <property name="type">GTK_WINDOW_TOPLEVEL</property>
...
  <property name="skip_pager_hint">False</property>
  <property name="type_hint">GDK_WINDOW_TYPE_HINT_NORMAL</property>
...

  <child>
    <widget class="GtkVBox" id="vboxToVrend">
      <property name="visible">True</property>
...

      <child>
                <widget class="GtkLabel" id="labelTitle">
                  <property name="visible">True</property>
                  <property name="label" translatable="yes">&lt;b&gt;convert to Vrend&lt;/b&gt;</property>
...
                  <property name="width_chars">-1</property>
                  <property name="single_line_mode">False</property>
                  <property name="angle">0</property>
                </widget>
...
```

Short Demo Qt-Designer

Construct user interfaces for the Qt library with this easy-to-use GUI design program.
Qt Designer
TROLLTECH

XML-GUI Description created by Qt-Designer

```
<ui version="4.0" >
 <author></author>
 <comment></comment>
 <exportmacro></exportmacro>
 <class>CDialogAbout</class>
 <widget class="QDialog" name="CDialogAbout" >
  <property name="geometry" >
   <rect>
    <x>0</x>
    <y>0</y>
    <width>337</width>
    <height>423</height>
   </rect>
  </property>
  <property name="minimumSize" >
   <size>
    <width>337</width>
    <height>423</height>
   </size>
  </property>
…
<property name="baseSize" >
<pixmapfunction></pixmapfunction>
 <resources/>
 <connections/>
</ui>
```

Qt-Designer embedding UI file

1. Compile ui file created by Qt Designer with UIC (Compiler for ui-classes/files)
2. Usage (example)
   - In your Ui-Folder:

     ```
     uic – o TargetFileName.h UiFileName.ui
     ```

   - Include compiled header file in your cpp or header file

     ```
     #include <TargetFileName.h>
     ```

   - Create reference to objects of namespace Ui

     ```
     private:
         Ui::TargetFileName ui;
     ```

   - In cpp you have to setup the generated form like this

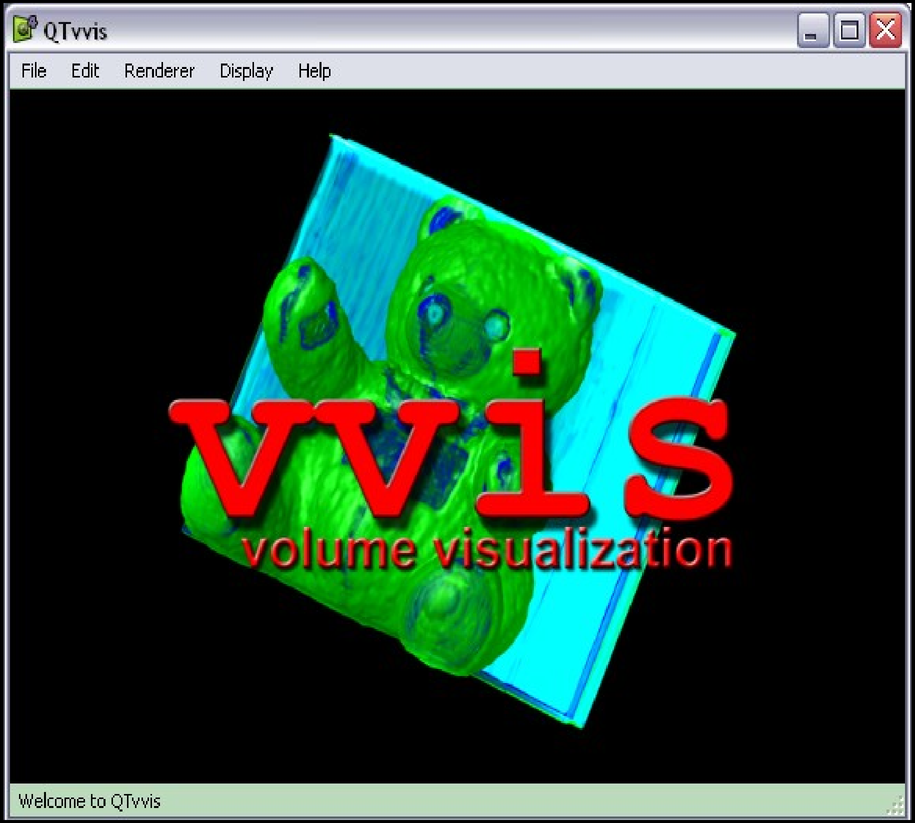
     ```
     ui.setupUI(this);
     ```

QtOpenGL Module

- QGLColormap
  - custom colormaps for QGLWidgets
- QGLContext
  - OpenGL rendering context
- QGLFormat
  - display format of OpenGL rendering context
- QGLWidget
  - widget for rendering OpenGL graphics

QtOpenGL Module

- QGLWidget
  - initializeGl()
  - paintEvent()
  - resizeGl()
  - swapBuffers()
  - QGLFormat format()
    - Rertuns GLContext



QtOpenGL Module

- How to display OpenGL within a Qt window
  1. Your class needs to inherit from QGLWidget
     - Classname : public QGLWidget
  2. Initialization in cpp
     - Classname(QGLFormat format, QWidget* parent)

QtOpenGL Module

- QGLFormat
  - specifies the display format of an OpenGL rendering context
  - Methods:
    - *depth(), doubleBuffer(), setRgba() etc…*
- GtkGlContext
  - *Glib::RefPtr<Gdk::GL::Config> &refGLConfig*

Gtk::GL::DrawingArea

- GTK+ DrawingArea widget supports OpenGL rendering
- Your class needs to inherit from Gtk::GL::DrawingArea
- Initialization
  - set_gl_capability(refGLConfig)

Callback-Methods Introduction

- Defines behaviour of program
  - A method which is passed to another by argument and is executed at certain conditions

Callback-Methods

- Libsigc++ :
  - Implements typesafe callback system for C++
  - Signals and slots
  - Uses Callback-Mechanism. The programmer passes the adress of a method to another which can execute the method behind the adress

Qt – Signals & Slots

- Signals / Slot mechanism
  - Communication between objects of the program
  - Alternative to callbacks

Qt – Signals & Slots

```
#include <QObject>
class Number : public QObject
{
      Q_OBJECT
private:
      int _value;
public:
      Number() { _ value = 0; }
      int value() const { return _value; }
public slots:
      void setValue( int value)
      {
        if( value != _value )
        {
                 _value = value;
                  emit valueChanged ( value );
         }
      }
signals:
      void valueChanged( int new Value);
};
```

Qt – Signals & Slots

```
Number a, b;
    QObject::connect(&a, SIGNAL(valueChanged(int)), &b,
                        SLOT(setValue(int)));


 a.setValue(12);
// a.value() == 12
// b.value() == 12 because signal of a is connected
 b.setValue(48);
// a.value() == 12 because signal of b is not connected
// b.value() == 48
```

Comparison

- Libsigc++
  - Signal is connected to method that should be invoked
  - Easy callback mechanism
- Qt Signals & Slots
  - Easy mechanism
  - signal is connected to slot
    - When signal is emitted the connected slot is invoked

Threads

- Definition
  - A way for a program to split itself
  - A thread can be executed parallel with other threads
  - Used for heavy computation processes for example:
    - Rendering and event processing at the same time

Glib::Thread

Declaration in header:

```
Glib::Thread *m_pThread;
```

Implementation in sourcefile:

```
m_pThread = Glib::Thread::create(
                    sigc::mem_fun(*this,
   &ClassName::doSomething()), true);
```

→Easy creation of thread

QThread

- *Declaration:*

  ```
  class MyThread : public QThread
  {
  public:
          void run();
  };
  ```

  Implementation:

  ```
  void MyThread::run()
  {
          doSomething();
          exec();
  }
  ```

- No possibility to create thread without creating new class deriving from QThread

QCoreApplication ::processEvents

- Alternative to dispense with QThread
- Processes all events of Qt application for time of invocation
- This function is called normally when program is busy performing a long operation i.e.
  - Converting
  - Rendering
  - Opening file
  - etc.

  Otherwise application would freeze

Thread comparison

- Glib::Thread
  - Easy to use
  - Use *Glib::Thread::create()* to instanciate your thread and simply connect it to destination method of any object
- QThread
  - Precondition is creation circumstantially of new class that inherits from QThread
  - QThreads can “talk with each other” by connection of signals and slots
  - QThreads share data with all other threads within the process

Qt vs. GTK+ (overview)

Web Site	http://www.trolltech.com	http://www.gtk.org
Programming Language	Based on C++	Based on C
Language Bindings	Perl, PHP, Python	Java, Perl, Python and many more
IDE	Has good RAD Tool (rely on Qt Designer and KDevelop)	Glade (has poorer documentation)

Qt vs. GTK+ (overview)

Web Site	Qt	GTK+
Supported OS	Runs on Linux, Windows, Macintosh	Mainly developed for Linux, ports for Win32, DirectFB, BeOS are available, Macintosh in development (far from complete)
Scripting	Has internal support for scripting (QAS)	Developer has to embed his own scripting language (Python, LUA…)
Database	Has integrated data-aware widget	Needs separate project like gnome-db
Usage	Qt Toolkit everything is included	Less functionality (needs many other libraries like libsigc++ for signals i.e.)

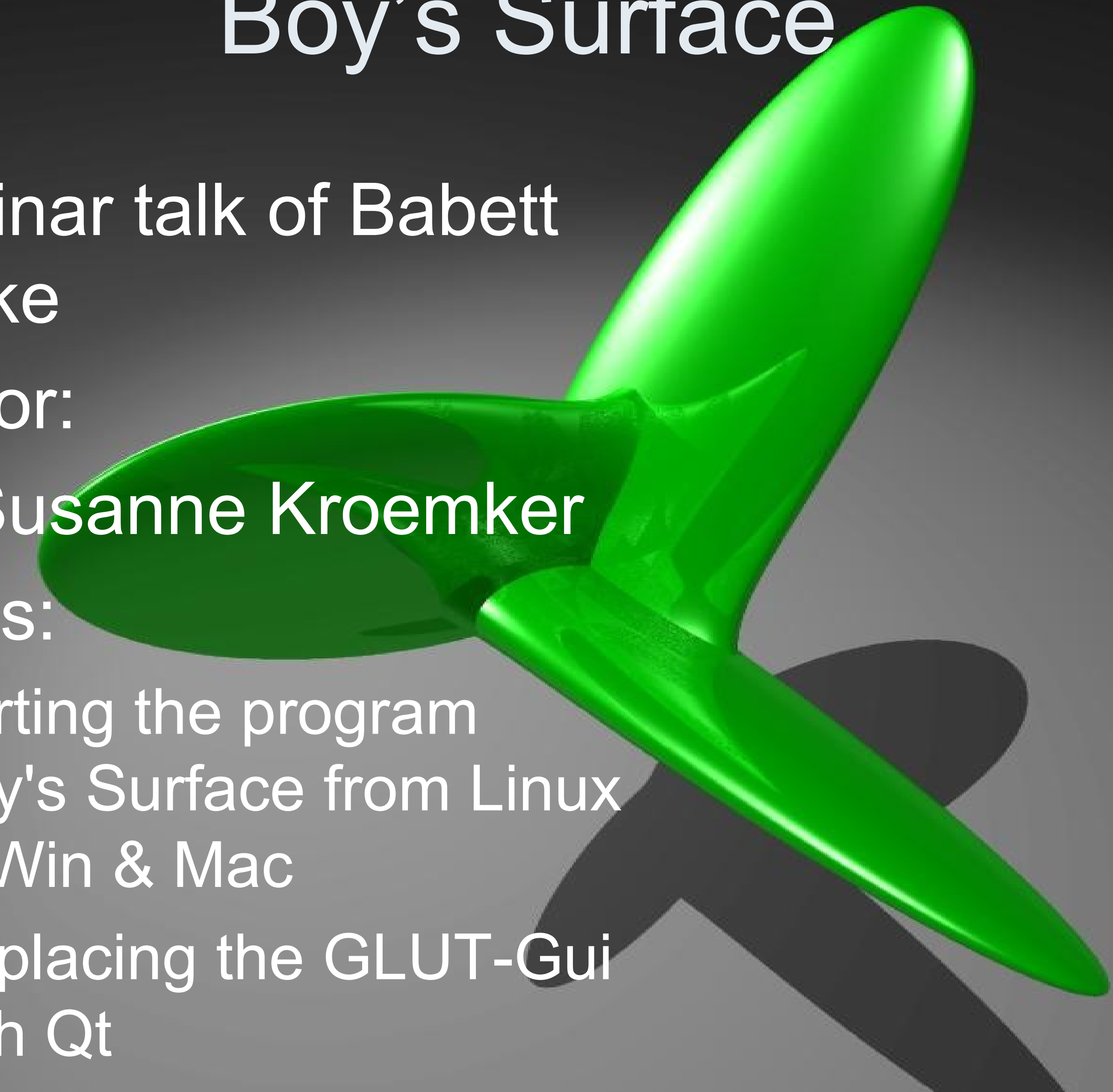
Qt vs. GTK+ (overview)

Web Site	Qt	GTK+
Communication	Talks via DCOP with other Qt Apps	Talks via CORBA
Documentation	Very good documentation and many source codes from Qt and KDE Team	Comprehensive reference (most time behind API documentation) many source codes from GTK Team
License	Qt is only freeware on Linux and for open source projects	GTK is freeware, GNU LGPL

Qt vs. GTK+ (overview)

Web Site	Qt	GTK+
Strengths & Features	Integrated internationalization and translation features Many helpful tools bundled RAD Tools with nice wizards Many advanced widgets available	Internationalization features (Unicode etc.) gtkmm: Good namespaces gtkmm: well defined, clean, modular API
Weaknesses	Huge sources and binaries. Takes ages to compile Objects not referred by namespace but simple literal prefix “Q”	Win32 Port unstable, Macintosh version far from completion

Boy’s Surface



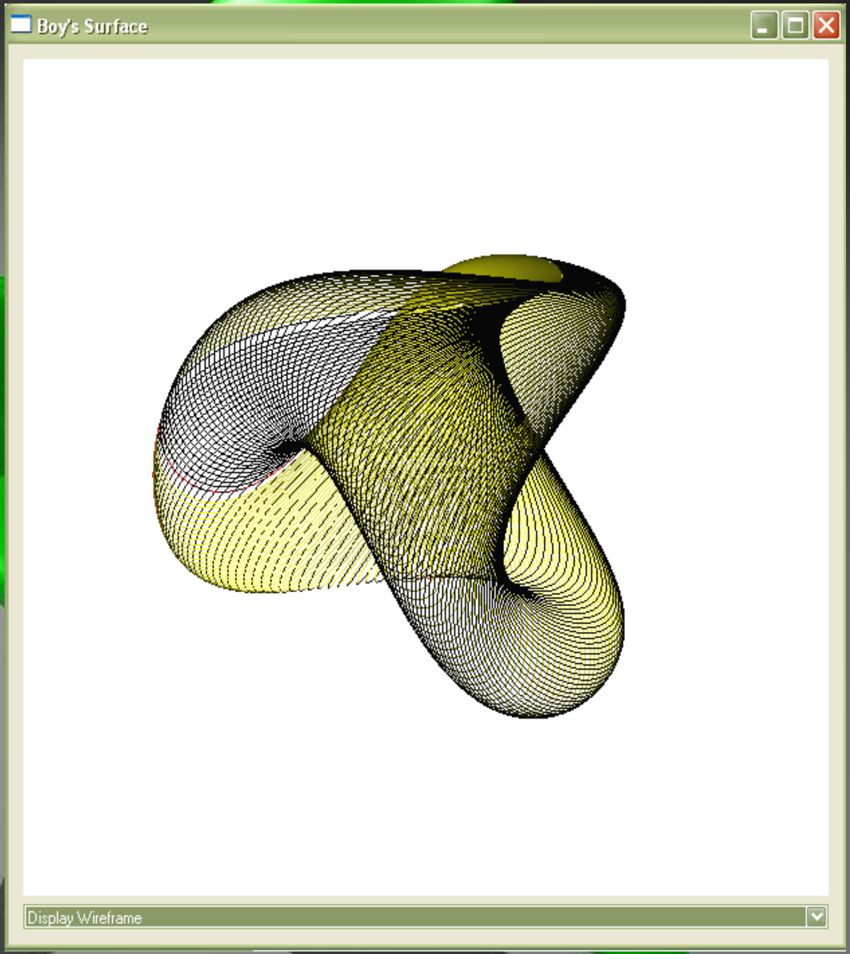
- Seminar talk of Babett Lemke
- Author:

  Dr. Susanne Kroemker
- Tasks:
  - Porting the program Boy's Surface from Linux to Win & Mac
  - Replacing the GLUT-Gui with Qt

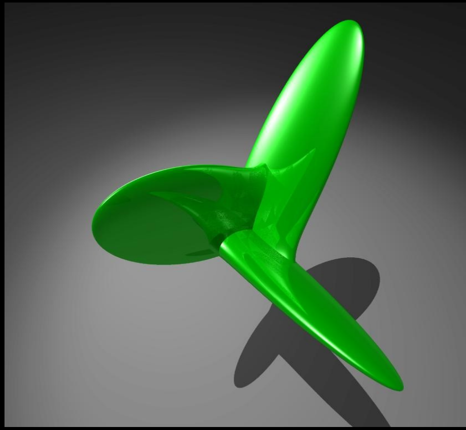
Boy's Surface

- Short Demo

  Boy's Surface with Qt



Boy's Surface
What is GLUT ?

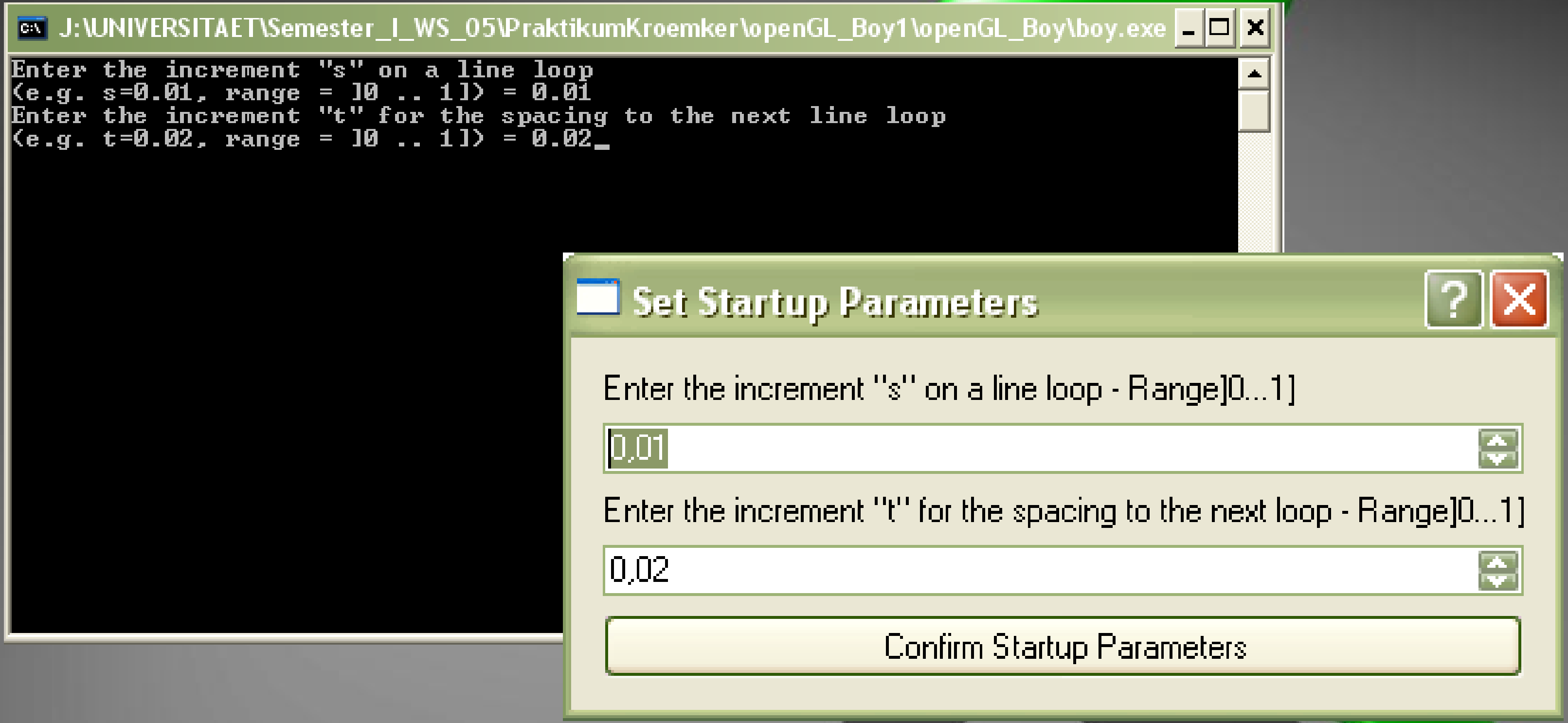


OpenGL Utility Toolkit (GLUT)

- library of utilities for OpenGL programs
- Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided
- limited support for creating pop-up windows
- At the moment no further development of GLUT

Boy's Surface

- Startwindow



J:\UNIVERSITAET\Semester_I_WS_05\PraktikumKroemker\openGL_Boy1\openGL_Boy\boy.exe

```
Enter the increment "s" on a line loop
(e.g. s=0.01, range = ]0 .. 1]) = 0.01
Enter the increment "t" for the spacing to the next line loop
(e.g. t=0.02, range = ]0 .. 1]) = 0.02_
```

Enter the increment "s" on a line loop - Range]0...1]

0,01

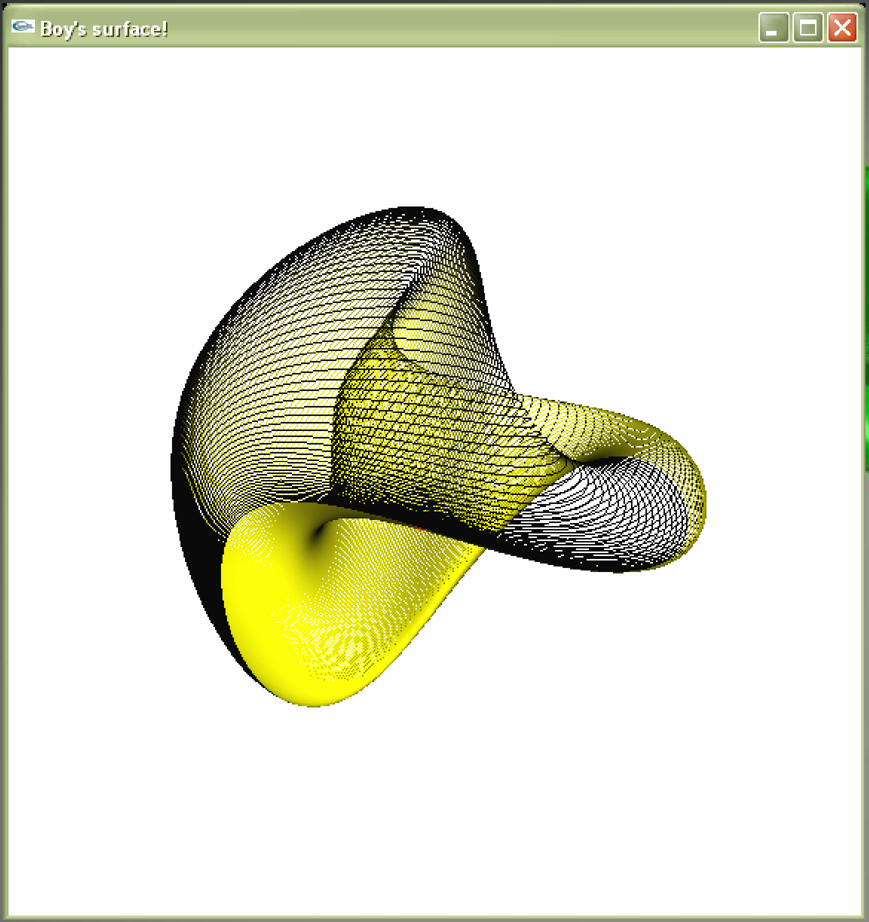
Enter the increment "t" for the spacing to the next loop - Range]0...1]

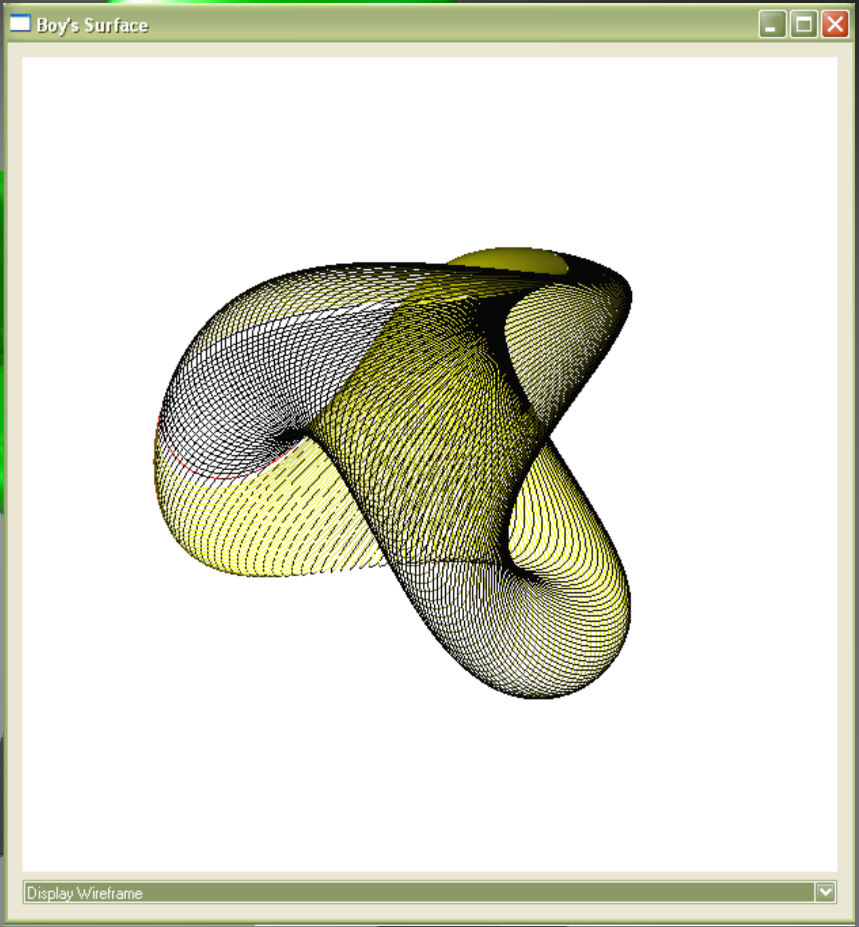
0,02

Confirm Startup Parameters

Boy's Surface

- Mainwindow





Display Wireframe

QTvvis

- Original Programm (GTK+ Version) by Johannes Lampel & Tobias Eberle http://vvis.sourceforge.net

QTvvis
What is vvis ?

- vvis renders 3D scalar fields like volume data
- Several file formats like OpenQvis or VRend are supported
- Supports TiFF conversion

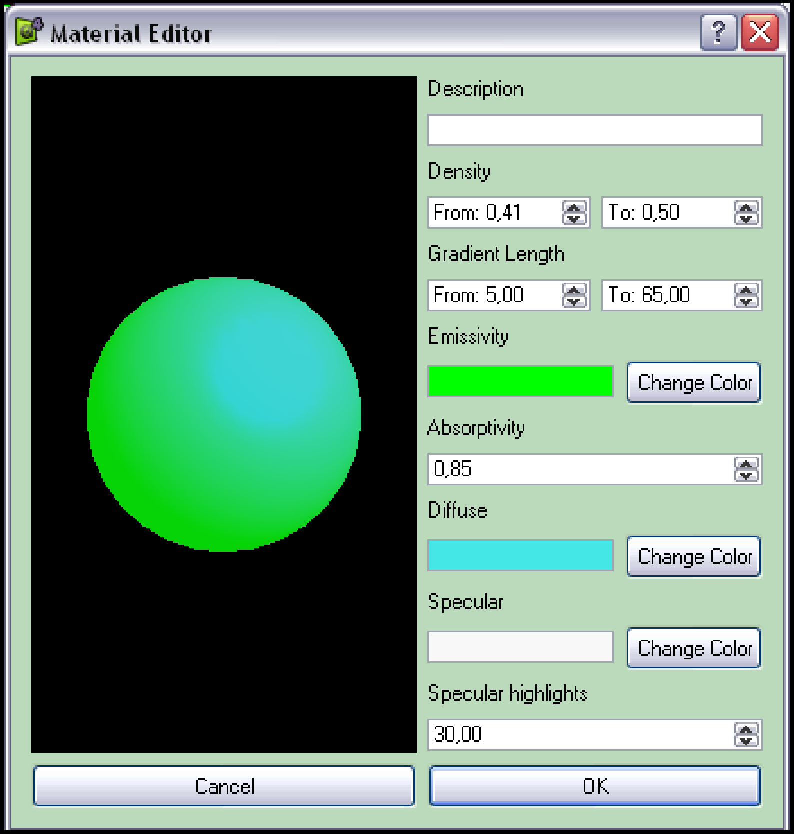
Demo QTvvis

QTvvis (Qt-Port)

- Requirements:
  - Qt 4.1.1 at least for compiling
  - OpenGL libraries (Windows)
  - Libtiff library (Win, Mac, Linux)

Problems during porting

- No QGLWidget in Qt-Designer (opensource edition)
- Layouting had to be done manually



Material Editor

Description:

Density: From: 0,41 To: 0,50

Gradient Length: From: 5,00 To: 65,00

Emissivity: 🟩

Absorptivity: 0,85

Diffuse

Specular

Specular highlights: 30,00

Problems during porting

- QThreads
  - No possibility to create a thread without creating new class deriving from QThread in Qt
- Vector and QWidgets
  - QWidgets have private constructor and aren’t storable into vector. Solution: Store reference
- Data type conversion
  - For example: std:String to QString
    - QString string = std_string.c_str();

Problems during porting

- Documentation
  - GTK+, GLib has poor Html documentation. Time-consuming search for all libraries documentation
- Installation of vvis
  - Research on Linux (no prev. knowledge existed)
  - vvis has many requirements (difficult installation)
  - At the beginning porting based on screenshots because original version wasn’t possible to run
- In general
  - Only little knowledge of Qt, C++ existed
  - No software development knowledge of Macintosh OS and Linux

Summary

- QT Support
  - Qt 4.1.1 has to be installed (due to bugs in previous versions)
  - Qt 4.x is not fully compatible with Qt 3.x
    - Special classes must be implemented if you want QT3 support
  - But: Qt 4 will soon be standard on all Linux distributions
  - Qt is still in development and some control elements doesn’t work properly on all platforms

- Porting of a GTK+ Application
  - GTK+ and Qt functionalities are comparable
  - Complexity should not be underestimated!

- Hope to enhance acceptance of vvis through:
  - Simple installation. vvis requires gtk(mm), glib, pango, atk, libsigc++, libglade, gtkglext(mm)
  - Less cost of configuration

Sources

- Wikipedia http://www.wikipedia.org
- vvis.net http://vvis.sourceforge.net
- Trolltech http://doc.trolltech.com
- Trolltech List Archives http://lists.trolltech.com/qt-interest/
- Trolltech Bug Tracking list http://www.trolltech.com/developer/tasktracker.html?meth
- GTK+ API Description http://www.gtk.org/api/
- GTKMM, GLIB Documentation http://www.gtkmm.org /docs/
- GTKMM List Archive http:// mail.gnome.org/archives/gtkmm-list/

Sources

- OpenGL Framebuffer Description http://www.delphi3d.net/articles/viewarticle.php?article=framebuf.htm
- OpenGL Texture Tutorial http://www.nullterminator.net/gltexture.html
- Delphi OpenGL Wiki http://wiki.delphigl.com/index.php/Hauptseite
- GluSphere Example Source code http://www.et.byu.edu/~merk/me570/qtRed/qtmaterial.cpp
- Libsigc++ 2 documentation http://libsigc.sf.net/libsigc2/docs/
- GTK+ for Mac
- http://developer.imendio.com/wiki/Gtk_Mac_OS_X/Things_to_do
- GTK+ vs. Qt
- http://mail.gnome.org/archives/gtk-list/2003-November/msg00037.html
- Freshmeat: GTK+ vs. Qt
- http://freshmeat.net/articles/view/928/