



Proseminar *Graphikprogrammierung*

WS 09/10

Contact



- Matthias Teschner
Marc Gissler

Computer Graphics

Georges-Koehler-Allee 052 / 01-005 / 01-024

teschner@informatik.uni-freiburg.de

gisslerm@informatik.uni-freiburg.de

<http://cg.informatik.uni-freiburg.de/teaching.htm>

Computer Science -> Computer Graphics -> Teaching

Registration - 1



- check handout for topics
- choose a topic at the end of this introduction
- [Die Themen werden in der ersten Veranstaltung am 19.10. vergeben. Anwesende, angemeldete Studenten werden bei der Themenvergabe vorrangig behandelt. Wenn angemeldete Studenten am 19.10. nicht anwesend sind, können anwesende, nicht angemeldete Studenten bei der Themenvergabe berücksichtigt werden. Dadurch können angemeldete Studenten, die am 19.10. nicht anwesend sind, eventuell bei der Themenvergabe nicht berücksichtigt werden.]
<http://cg.informatik.uni-freiburg.de/teaching.htm>

Registration - 2



- check the web page for available topics
- choose a topic
- send an email to Marc Gissler with name, matriculation number, topic
- Marc updates the web page and sends a confirmation

Outline



- introduction
- organization
- presentation
- project
- sources

Introduction to C



- programming language
- developed by Dennies M. Ritchie at Bell Laboratories in the early 70s
- development of C is related to the UNIX operating system which is programmed in C
- C is independent of platform and operating system
- C is used for system and application programming
- standardization of C in 1989 (ANSI-C)

Introduction to C++



- extension to C
- object-oriented programming language
- developed by Bjarne Stroustrup at Bell Laboratories in 1980
- standardization started in 1989 (ANSI-C++)

Properties of C



- data types
 - int, char, float, double
 - array, structure, class, file, pointer
 - no strict type binding
- control structures
 - if, switch, while, do ... while, for
 - jump: break, continue, return
- functions
- input / output

Introduction to OpenGL



- Open Graphics Library - software interface to graphics hardware
- introduced in 1992
- premier environment for portable graphics applications
- industry's most widely used and supported 2D and 3D graphics application programming interface
- supported on a variety of computer platforms
- incorporates several hundreds rendering, special effects, and visualization functions
- specification guided by the OpenGL Working Group, Khronos Group

Introduction to OpenGL



- ARB - OpenGL Architecture Review Board (until Aug 2006)
 - founded in 1992
 - independent standards consortium
 - As of November 2004, Promoter-level members (with a formal vote in the body) include 3Dlabs, Apple, ATI, Dell Computer, IBM, Intel, NVIDIA, SGI, and Sun Microsystems. A partial list of Contributor-level members includes Evans and Sutherland, Imagination Technologies, Matrox, Quantum3D, S3 Graphics, Spinor GmbH, Tungsten Graphics, and Xi Graphics.
- OpenGL Working Group under the Khronos Group consortium for open standard APIs (since Sep 2006)
- current version 3.2 (August 2009)

Motivation

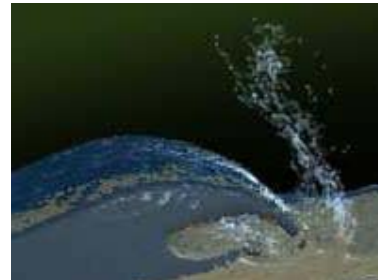
Graphical Simulation



- physically-based animation of
 - rigid bodies, deformable objects, fluids



rigid bodies



fluids



contact handling

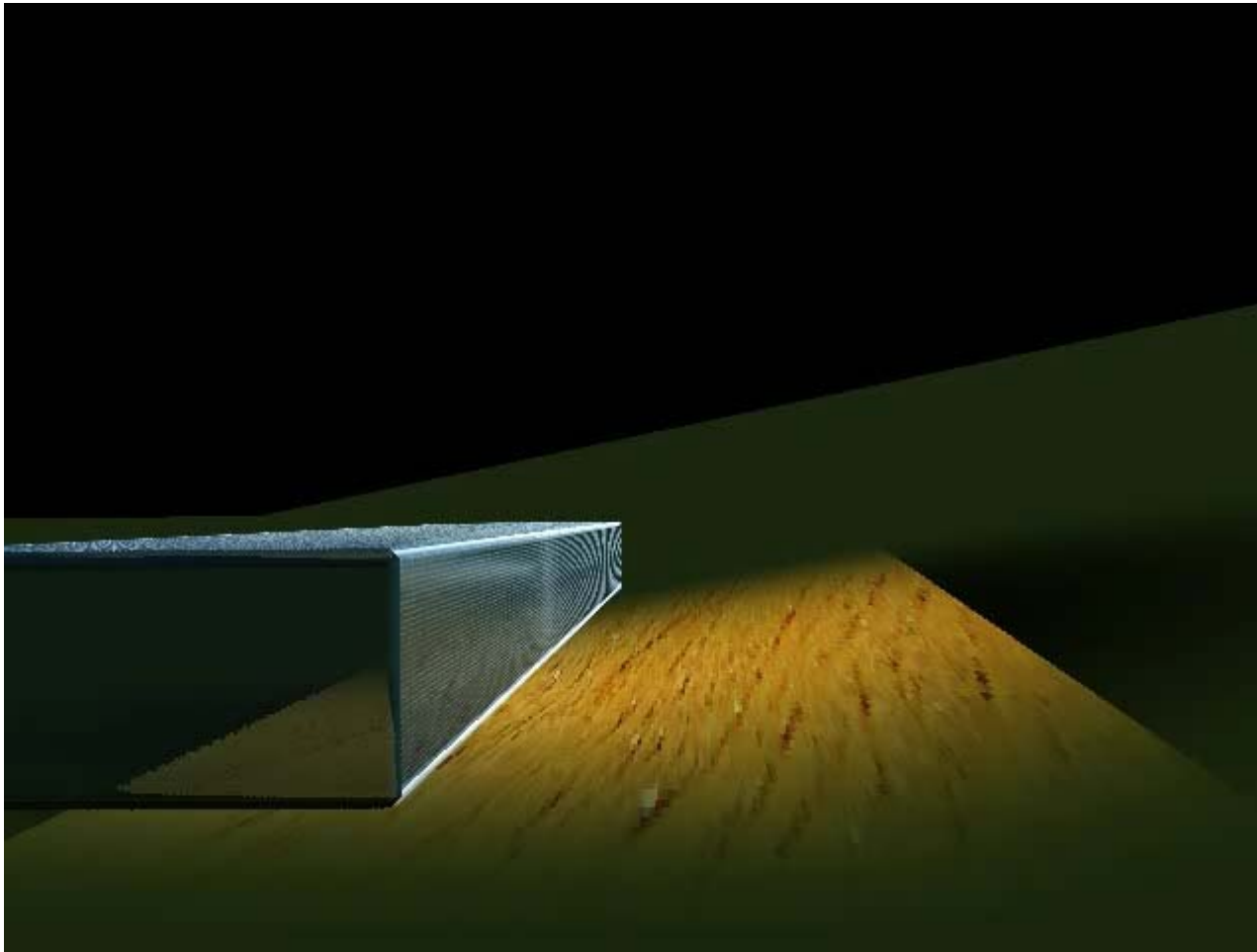


deformable objects

Graphical Simulation Deformable Solids



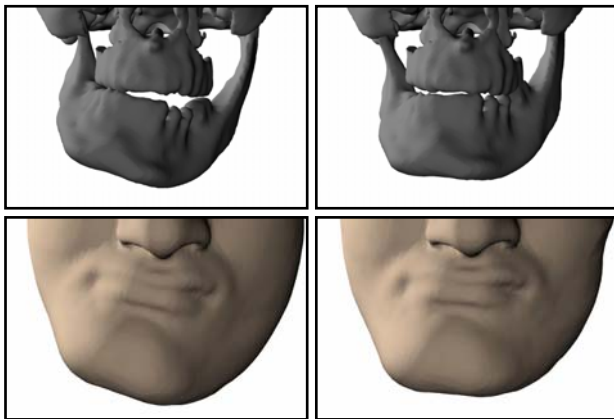
Graphical Simulation Fluids



Graphical Simulation Applications



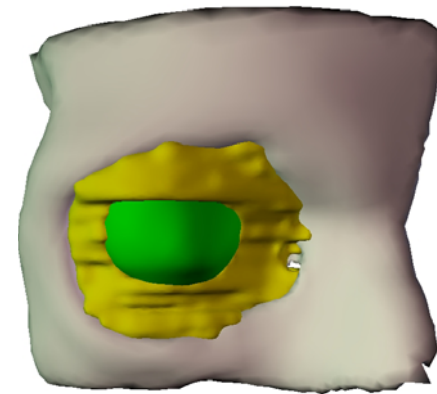
- computational medicine



pre-operative planning in
cranio-maxillofacial surgery



interactive hysteroscopy
simulation for educational
purposes

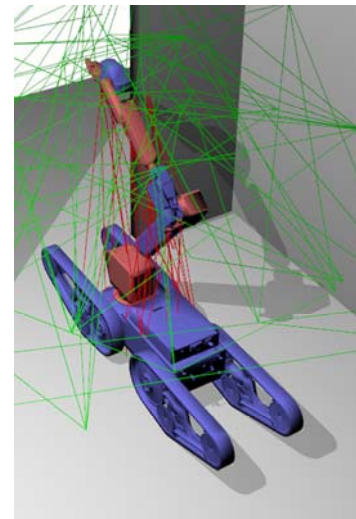


intra-operative support
in orbital reconstruction

Graphical Simulation Applications



- robotics



support of robot navigation
(with Wolfram Burgard and Bernhard Nebel, University of Freiburg)

Outline



- introduction
- organization
- presentation
- project
- sources

Organization



- presentation of a C++ / OpenGL topic
- project in C++ using OpenGL
- short presentation of the implementation

- implementation can be realized in groups
- attendance of all presentations is mandatory

- recent information on
<http://cg.informatik.uni-freiburg.de/teaching.htm>

Outline



- introduction
- organization
- presentation
- project
- sources

Presentation



- 25 min - 35 min presentation
- 10 min - 20 min discussion
 - form of the presentation
 - technical questions

Topics - Example



- structures and classes in C++
 - definition of structure and class
 - difference of structure and class
 - ...
 - <http://www.cprogramming.com/tutorial/lesson7.html>
- lines and polygons in OpenGL
 - functions glVertex, glBegin, glEnd
 - difference of GL_POINTS and GL_LINES
 - ...
 - <http://fly.cc.fer.hr/~unreal/theredbook/chapter02.html>

Preparation



- know your topic
 - examine relevant material thoroughly
 - do not try to circumvent problems
- prepare slides
 - allow 1 to 2 minutes per slide
 - slides should be uniform and not too dense
 - incorporate illustrations
 - slide titles should be helpful
- rehearse your presentation
 - gather feedback
 - adapt your presentation accordingly
 - check your slides with Marc Gissler

Presentation



- introduction
 - introduce yourself and the title of your presentation
- overview
 - give an idea, but not too detailed
- motivation
 - illustrate the principle and / or applications
 - explain the goal of your presentation
 - the audience should be eager to listen your presentation

Presentation



- main part
 - should consist of distinguished parts
 - separate different parts of the presentation explicitly
 - each part should be introduced and summarized
- summary
 - tell the audience what you have told them
 - ask for questions

Presentation - Summary



- introduce the title and yourself
- motivate and introduce your topic thoroughly
 - it is essential to arouse the interest of the audience right at the beginning
- give a brief overview
 - avoid too many details
- structure your presentation
 - introduce and summarize parts of your presentation
- summarize the entire presentation
- clearly mark the end of your presentation
 - e. g. “Thank you for your attention.”

Presentation



- check the presentation environment prior to the presentation
- do not occlude the projection
- avoid idiosyncrasies
- stay in time

Presentation



- do not learn your talk by heart
- do not read your talk
- do not read slides,
but explain every item on your slide
- do not be shy or quiet
- communicate self-confidence

Outline



- introduction
- organization
- presentation
- project
- sources

Environment and Organization



- GLUT-based viewer, one object, one light source (see web page)
- alternatively, you can use your own environment
- in general, indicate and cite all third-party libraries and source code that you use
- up to three participants per group
- contributions of each member have to be indicated

Requirements per Person



- **minimum**
 - two or more objects
 - object animation (objects move relative to each other)
 - different material for each object
 - two or more light sources
 - light source animation (light sources move relative to each other)
- **optional**
 - texture mapping
 - object morphing / geometric deformation
 - fog, blending
 - user interaction
 - being creative

Presentation



- projects are presented in the last meeting on Feb 8
- slides, images, software
- per person: 1 min preparation, 2 min presentation, 1 min discussion
- groups have more time, accordingly
- focus on essential aspects of your project
- cite third-party code
- check the presentation environment
 - which laptop to use (if you do not have a laptop, ask Marc Gissler)
 - make sure your presentation including software is working on the laptop and the projector

Outline



- introduction
- organization
- presentation
- project
- sources

Sources



- <http://cplusplus.about.com/od/beginnerctutoria1/1/blctut.htm>
- C Programming Tutorial
 - Hello World, Variables, Constants, Input / Output, Conditional Processing, Looping, Pointers, Arrays, Strings, Structures, Memory Allocation, File I/O, Command Line Arguments, Functions, Scope and Program Structure
- C++ Programming Tutorial
 - Classes, References, Function Overloading, Function Templates, Constructors, Destructors, Operator Overloading, Inheritance, Polymorphism, Abstract Data Types, Exceptions, Class Templates

Sources



- <http://cplusplus.com>
- C++ Programming Tutorial
 - Basics (program structure, variables, data types, operators)
 - Control Structures and Functions
 - Advance Data (arrays, strings, pointers, dynamic memory, structures, user-defined data types)
 - Object-oriented Programming (classes, constructors, destructors, overloading, inheritance, virtual members)
 - Advanced concepts (templates, namespaces, exception handling, preprocessor directives)
 - C++ Standard Library (file input / output)

Sources



- <http://cprogramming.com/tutorial.html>
- C Tutorial
 - intro, if, loops, functions, switch, pointers, structures, arrays, strings, file I/O, typecasting, command line arguments, applications: linked list, recursion, binary trees
- Advanced C / C++
 - references, inheritance, exception handling, namespaces, templates
- C++ Standard Template Library (STL)
 - vector container, map container, list container, iterators

Sources



- <http://fly.cc.fer.hc/~unreal/theredbook/>
- OpenGL Programming Guide (The Red Book)
 - introduction, drawing, viewing, display lists, color, lighting, blending, antialiasing, fog, bitmaps, fonts, textures, framebuffer
- <http://www.cs.uccs.edu/~semwal/indexGLTutorial.html>
- OpenGL tutorial
 - introduction (rendering pipeline, libraries, include files, compilers)
 - basics (initialization, creating a window, display function, main loop)
 - objects (points, lines, polygons, transformations, display lists)
 - viewing (color, shading, light, transformations: viewing, projections)
 - input devices, selection

Getting Set Up – C++



- compiler (Borland C++, Microsoft C++, GNU C++)
- front-end environments, e. g.
 - Microsoft Visual C++
 - http://poolmgr.informatik.uni-freiburg.de/?Software:Software_von_Microsoft

Summary



- presentation (25 min - 35 min)
- software project

- start preparation in time
- employ various sources
- rehearse your talk
- check presentation with Marc Gissler