

# Vorträge

 Wichtig: Unbedingt die PowerPoint-Einführung unter <http://cg.informatik.uni-freiburg.de/teaching/powerpoint.ppt> lesen!

## 1. Woche - 19.10.

**Einführungsveranstaltung**

## 2. Woche - 26.10.

*Selbständige Vorbereitung!*

## 3. Woche - 02.11.

*Selbständige Vorbereitung!*

## 4. Woche - 09.11.

### 1. Mathematische Datentypen, Variablen, Zuweisungen, Kontrollstrukturen

- Erklärung float, double, int, long
- Erklärung Definition von Variable
- casting
- Zuweisung (+, +=, ++)
- Bitoperationen &, |, ~, ^
- for, while, do while
- if, ?:
- &&, ||, inkl. Vorrang d.Operatoren
- Boolean-Test (if (x), if (!x))
- switch
- Ressourcen:
  - <http://www.cplusplus.com/doc/tutorial/>
  - <http://cplus.about.com/od/learningc/ss/clesstontwo.htm>
  - <http://www.cprogramming.com/tutorial.html#c++tutorial>  
(nur die ersten fünf Tutorials)
  - [http://www.cprogramming.com/tutorial/bitwise\\_operators.html](http://www.cprogramming.com/tutorial/bitwise_operators.html)

### 2. Funktionen und Programmstruktur

- Definition einer Funktion, void
- Signatur einer Funktion
- Default-Parameter
- main()-Funktion
- Geltungsbereich von Variablen (innerhalb {}-Blöcken)
- Globale Variablen, extern
- Konstanten (const)
- Ressourcen:
  - <http://www.cplusplus.com/doc/tutorial/>
  - [http://cprogramming.com/tutorial/const\\_correctness.html](http://cprogramming.com/tutorial/const_correctness.html)

## 5. Woche - 16.11.

### 1. Speicherverwaltung

- Erklärung von Memory, wie Speicherzellen angeordnet sind, Nummerierung
- Was ist ein Bit, Byte, integer, long integer
- Adresse einer Variable, &-Operator, \*-Operator
- Pointer, inkl. graphischer Syntax für Pointer (beim Tutor nachfragen!)
- Ressourcen:
  - <http://cplus.about.com/od/learningc/ss/clessontwo.htm>
  - <http://cplus.about.com/od/glossar1/g/bitdefinition.htm>
  - <http://www.tutorialpage.de/Cpp/index.php>--> Zeiger und Strukturen (6.1 und 6.2)

### 2. Arrays und Pointer in C++

- Arrays, Definition, Initialisierung, Zugreifen auf Elemente
- Offset-Operatoren [], +, -
- Wie Arrays und Pointer miteinander verwandt sind
- sizeof
- Nullpointer
- evtl. Zeichenketten (char\*), Initialisierung (nur kurz!)
- Ressourcen:
  - <http://www.cprogramming.com/tutorial.html#c++tutorial>
  - <http://www.tutorialpage.de/Cpp/index.php>
  - <http://cplus.about.com/od/learningc/ss/clessonthree.htm>
  - <http://www.cplusplus.com/doc/tutorial/pointers.html>

## 6. Woche - 23.11.

### 1. OpenGL Einführung und Geschichte

- Theorie, geschichtlicher Abriss
- Kurz erklärtes Beispielprogramm (z.B. aus dem OpenGL Programming Guide, Seite 6)
- OpenGL Rendering Pipeline
- State Machine Konzept von OpenGL
- Ressourcen:
  - [http://www.cprogramming.com/tutorial/opengl\\_introduction.html](http://www.cprogramming.com/tutorial/opengl_introduction.html)
  - <http://de.wikipedia.org/wiki/Rendering-Pipeline>
  - <http://www.informatik.uni-augsburg.de/lehrestuehle/dbis/pmi/lectures/ss06/graphikprogrammierung/script/opengl.pdf> (Kap. 1.1 und 1.2)

### 2. OpenGL Linien und Flächen

- Punkte, Linien
- Dreieck zeichnen. Funktion glVertex, glBegin und glEnd
- Wireframe vs. gefüllte Flächen
- Unterschiede GL\_POINTS, GL\_LINES etc.
- Dreiecksnetz, inkl. Datenstruktur (ggf. beim Tutor nachfragen!)
- Dreiecksnetz zeichnen
- Ressourcen:
  - <http://www.cs.uccs.edu/~semwal/geometric.html#GeometricObjects> (nur bis und mit 'Drawing 3d-objects')

- <http://fly.cc.fer.hr/~unreal/theredbook/chapter02.html>  
(nur Kapitel 'Describing Points, Lines ...')
- <http://www.falloutsoftware.com/tutorials/gl/gl2p5.htm> und /gl/gl3.htm
- <http://www.informatik.uni-augsburg.de/lehrestuehle/dbis/pmi/lectures/ss06/graphikprogrammierung/script/opengl.pdf> (Kap. 5)
- <http://www.siggraph.org/education/materials/HyperGraph/modeling/polymesh/polymesh.htm>

## 7. Woche - 30.11.

### 1. Shader

- Was sind Shader?
- Übersicht OpenGL Rendering Pipeline
- Welche Teile der Pipeline sind durch Shader veränderbar?
- Beispielshader, welche die ursprüngliche Pipeline simulieren
- Shader in der eigenen Applikation einbinden
- kleines Anwendungsbeispiel
- Ressourcen:
  - Randi J. Rost : **OpenGL shading language**. 2. edition, 2007.  
(z.B. Kapitel 4 und 6)
  - [http://cg.informatik.uni-freiburg.de/intern/cg\\_shader.pdf](http://cg.informatik.uni-freiburg.de/intern/cg_shader.pdf)
  - <http://www.lighthouse3d.com/opengl/gsl/>
  - auf dem Vortrag **OpenGL Einführung und Geschichte** aufbauen

### 2. OpenGL GLUT

- Was ist GLUT?
- Einfaches Einführungsbeispiel
- Öffnen eines Fensters mit GLUT
- Mainloop
- Wie wird schlussendlich gezeichnet (glutDisplayFunc, glutReshapeFunc)
- Animation (mit idle)
- Evtl. Benutzerevents verarbeiten
- Ressourcen:
  - <http://www.lighthouse3d.com/opengl/glut/index.php?1> (nicht ins Detail gehen)
  - <http://mindfuck.de-brauer.be/articles/glut/>

## 8. Woche - 07.12.

### 1. Stack, Heap, Lokale Variablen

- Erklärung von Stack und Heap
- Stackframes, Rekursion
- Lokale Variablen
- Variablen im Heap
- new, delete Operator im Kontext von Arrays (noch keine Klassen)
- Ressourcen:
  - [http://tutorial.schornboeck.net/stack\\_heap.htm](http://tutorial.schornboeck.net/stack_heap.htm)
  - [http://de.wikibooks.org/wiki/C++-Programmierung:\\_Speicherverwaltung](http://de.wikibooks.org/wiki/C++-Programmierung:_Speicherverwaltung)

## 2. Pointer als Parameter, mehrdimensionale Vektoren

- Pointer als Funktionsargumente
- Rückgabe von Pointern, Problem mit Rückgabe von Pointer auf lokale Variable
- Mehrdimensionale Vektoren
- Ressourcen:
  - <http://www.codeguru.com/forum/showthread.php?t=343473>
  - <http://www.cprogramming.com/tutorial.html#c++tutorial>
  - <http://www.cplusplus.com/doc/tutorial/arrays.html>
  - <http://courses.cs.vt.edu/~cs1044/fall00/mcquain/Notes/Supplemental/S03.MultidimensionalArrays.pdf>

## 9. Woche - 14.12.

### 1. OpenGL Transformationen und Kamera 1

- Unterschied zwischen 'Kamera bewegen' und 'Objekte bewegen'.
- Erklärung Weltkoordinaten vs. Objektkoordinaten
- Wie werden Objekte bewegt (glTranslate, glRotate, glScale)
- Setzen der Kamera mit gluLookAt, Erklärung der Parameter der Funktion
- Perspektive, glFrustum
- Ressourcen:
  - <http://www.cs.uccs.edu/~semwal/geometric.html#GeometricObjects> (nur Kapitel 'Transformations')
  - <http://www.cs.uccs.edu/~semwal/viewing.html#Viewing> (nur Kapitel Viewing Transformation, Projection, Viewport Transformations)
  - <http://fly.cc.fer.hr/~unreal/theredbook/chapter03.html> (ohne Kapitel 'Viewing and Modeling Transformations', nur bis 'Transformations Troubleshooting')
  - <http://www.informatik.uni-augsburg.de/lehrstuehle/dbis/pmi/lectures/ss06/graphikprogrammierung/script/opengl.pdf> (Kap. 3)
  - <http://www.falloutsoftware.com/tutorials/gl/gl0.htm>

### 2. OpenGL Transformationen und Kamera 2

- siehe 1. (Aufteilung bitte mit dem Tutor absprechen)

## 10. Woche – 21.12.

### 1. Strukturen

- Definition Struktur, . Operator
- Struktur im Kontext von Memory
- Pointer auf Struktur, -> Operator.
- Strukturen als Parameter, Problem mit Copy
- sizeof im Kontext von Strukturen
- Ressourcen:
  - <http://www.cprogramming.com/tutorial/lesson7.html>
  - [http://cprogramming.com/tutorial/size\\_of\\_class\\_object.html](http://cprogramming.com/tutorial/size_of_class_object.html)
  - <http://www.tutorialpage.de/Cpp/index.php>

- <http://randu.org/tutorials/c/structs.php>

## 2. Klassen

- Unterschied Struktur <-> Klasse
- Definition von Klassen im Header
- Protected, private, public
- Deklaration von Klassenmember im Sourcecode
- Erzeugen von Klassen auf dem Stack, new, delete
- Konstruktoren, Destruktoren
- Default Parameter bei Methoden und Konstruktoren
- Ressourcen:
  - <http://www.cplusplus.com/doc/tutorial/classes.html>
  - [http://www.codersource.net/cpp\\_tutorial\\_class.html](http://www.codersource.net/cpp_tutorial_class.html)
  - [http://de.wikipedia.org/wiki/Objektorientierte\\_Programmierung](http://de.wikipedia.org/wiki/Objektorientierte_Programmierung)

## Weihnachtsferien

### 13. Woche - 11.01.

#### 1. OpenGL Farben und Beleuchtung

- Wie wird eine Farbe zusammengesetzt (RGB)
- Wie wird eine Farbe gesetzt (glColor3f)
- Ambientes Licht
- Gerichtete und ungerichtete Lichtquellen in OpenGL
- Setzen von Lichtquellen mit glLightfv (Fälle w=0 und w=1 diskutieren)
- Diffuse und spekulare Lichtquellen
- Setzen von Materialeigenschaften (Farben, Reflektion, ohne Mathematik der Reflektion)
- Schattierung: Flat vs. Gourad
- evtl. Normalen
- Ressourcen:
  - <http://www.falloutsoftware.com/tutorials/gl/gl8.htm>
  - <http://fly.cc.fer.hr/~unreal/theredbook/chapter05.html>  
(ohne Kapitel 'Dithering' und 'Color Index Mode')

#### 2. OpenGL Texture Mapping

- Was ist eine Textur?
- Wad bedeutet Texturemapping (Gift-Wrapping Problem)?
- Bedeutung von Texturkoordinaten, Möglichkeiten der Texturierung
- Planares Texturemapping, sphärisches Texturemapping
- Laden und Erzeugen eines Texturobjektes
- Setzen von Texturkoordinaten
- Ressourcen:
  - <http://fly.cc.fer.hr/~unreal/theredbook/chapter09.html>  
(ohne 'Modulating and Blending', nur bis und mit 'Computing Appropriate Texture Coordinates')
  - <http://www.gamedev.net/reference/articles/article947.asp>
  - [http://www.cosy.sbg.ac.at/~held/teaching/graphik/ps\\_pdf\\_ppt/huber\\_ps\\_chernig.pdf](http://www.cosy.sbg.ac.at/~held/teaching/graphik/ps_pdf_ppt/huber_ps_chernig.pdf) (nicht ins Detail gehen)
  - <http://graphics.tu-bs.de/teaching/lectures/ss06/CG2/slides/08-texturing.pdf>

- <http://www.mvps.org/directx/articles/spheremap.htm>

## 14. Woche - 18.01.

### 1. OpenGL Transparenz

- Was ist Transparenz? Was ist Transluzenz?
- Z-Buffer
- Transparenz in OpenGL
- <http://de.wikipedia.org/wiki/Z-Buffer>
- <http://de.wikipedia.org/wiki/Transluzenz>
- <http://www.opengl.org/resources/faq/technical/transparency.htm>
- [http://www.codeworx.org/opengl\\_tut8.php](http://www.codeworx.org/opengl_tut8.php)

### 2. Kompilation, Linking, Headerdateien

- Was ist eine Headerdatei, warum?
- #include <...>, #include "..."
- Definition vs. Deklaration von Variablen und Funktionen
- Kompilation von Sourcen, ein Header, mehrere Sourcen
- Linking
- Nochmals globale Variablen
- #define, Makros
- Resource:
  - [1]
  - [http://en.wikipedia.org/wiki/Header\\_file](http://en.wikipedia.org/wiki/Header_file)
  - [http://www.cpp-tutor.de/cpp/le08/le08\\_04.htm](http://www.cpp-tutor.de/cpp/le08/le08_04.htm)

## 15. Woche 25.01.

### 1. STL Einführung 1

- Was ist STL
- Kurze Erklärung von Templates im Kontext von STL
- Erklärung der Klasse vector
- Definition von vector<> im Programm, Zugriff mit [], size()
- vector<> als Funktionsparameter im Kontext des &-Referenzoperators
- Unterschiede vector, list, deque, set, map
- Konzept von Iteratoren (nur Forward-Iteratoren)
- Ressourcen:
  1. <http://www.cprogramming.com/tutorial/stl/stlintro.html>
  2. <http://www.cprogramming.com/tutorial/stl/vector.html>
  3. [http://en.wikipedia.org/wiki/Standard\\_Template\\_Library](http://en.wikipedia.org/wiki/Standard_Template_Library)
  4. [http://en.wikipedia.org/wiki/Vector\\_\(STL\)](http://en.wikipedia.org/wiki/Vector_(STL))

### 2. STL Einführung 2

- siehe 1. (Aufteilung bitte mit dem Tutor absprechen)

## 16. Woche 02.02.

*Selbständiges Programmieren!*

## 17. Woche 08.02.

### Präsentation der Projekte

#### Allgemeine Referenzen:

[1] Bjarne Stroustrup: **Die C++-Programmiersprache**, deutsche Übersetzung der "Special Edition" von Nicolai Josuttis, 4. Auflage, Addison-Wesley, 2007.

[2] Dave Shreiner: **OpenGL programming guide: the official guide to learning OpenGL**, 6. Auflage, Addison-Wesley, 2008.

[3] Michael Mortenson: **Mathematics for Computer Graphics Applications**, 2. Auflage, Industrial Press, 2007.

[4] Randi Rost: **OpenGL shading language**, 2. Auflage, Addison-Wesley, 2007.