

# **OpenGL Programming**

# OpenGL Programming

[http://en.wikibooks.org/wiki/OpenGL\\_Programming](http://en.wikibooks.org/wiki/OpenGL_Programming)

This Book Is Generated By **Wb2PDF**

using

**RenderX XEP**, XML to PDF XSL-FO Formatter

---

# Table of Contents

1. OpenGL Programming.....	5
Introduction.....	5
Setting Up OpenGL.....	6
Modern OpenGL.....	6
The basics arc.....	6
The lighting arc.....	7
The scientific arc.....	10
Selected topics.....	11
The post-processing arc.....	12
Mini-portal.....	12
Glescraft.....	13
Using the accumulation buffer.....	14
Cutting-edge OpenGL.....	14
Code quality.....	15
Appendices.....	15
Legacy OpenGL 1.x.....	15
Starting Tutorial.....	16
Basics.....	16
Intermediate.....	17
Advanced.....	17
Appendices.....	18
External links.....	18
Wikibooks.....	18
Ports.....	19

# OpenGL Programming

---

- Freely-licensed documentation and samples..... 19
- Non-freely-licensed documentation..... 19
- Websites..... 20
- Further reading..... 20

# OpenGL Programming

Welcome to the **OpenGL Programming** book. OpenGL is an API used for drawing 3D graphics. OpenGL is not a programming language; an OpenGL application is typically written in C or C++. What OpenGL does allow you to do is draw attractive, realistic 3D graphics with minimal effort. The API is typically used to interact with a GPU, to achieve hardware-accelerated rendering.

You are free, and encouraged, to **share** and **contribute** to this wikibook: it is written in the spirit of [free documentation](#), that belongs to humanity. Feel free to make copies, teach it in school or professional classes, improve the text, write comments or even new sections.

**We're looking for contributors.** If you know about OpenGL, feel free to leave comments, expand TODO sections and write new ones!



Also see the [GLSL Programming](#) book.



[Wikipedia](#) has related information at [OpenGL](#)

## Introduction

1. [About this book](#)
2. [History and Evolution of OpenGL](#)

## Setting Up OpenGL

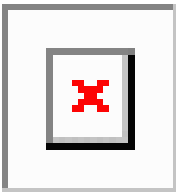
- [Installation on GNU/Linux](#)
- [Installation on Macintosh](#)
- [Installation on Windows with Code::Blocks](#)
- [Installation for Android NDK development](#)
  - with a GLUT-like wrapper to follow the exercises
- [Installation for iPhone development](#)
- [Installing GLUT](#)

## Modern OpenGL

"Modern" OpenGL is about OpenGL 2.1+, OpenGL ES 2.0+ and WebGL, with a programmable pipeline and shaders.

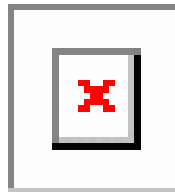
## The basics arc

01



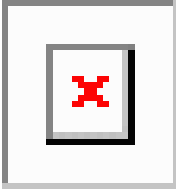
**Tutorial 01:** newcomer's introduction, first dive into shaders

02



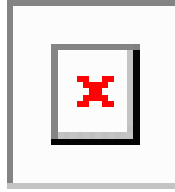
**Tutorial 02:** adding more robustness to our code, transparency

03



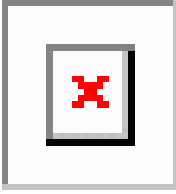
**Tutorial 03:** passing information to shaders: attributes, varying and uniforms

04



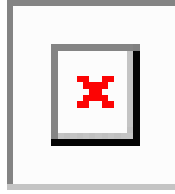
**Tutorial 04:** transformation matrices: positioning and rotating

05



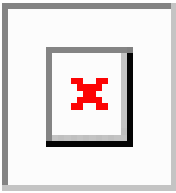
**Tutorial 05:** adding the 3rd dimension: a cube, plus a camera

06



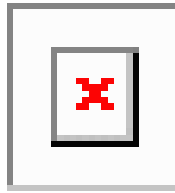
**Tutorial 06:** textures: displaying a wooden cube

07



**OBJ format:** loading Suzanne the monkey from Blender

08



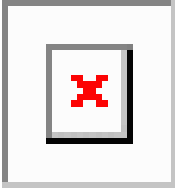
**Navigation:** navigate in 3D space and manipulate objects in our model viewer

[Tutorial\\_drafts](#): ideas and notes for upcoming tutorials

## The lighting arc

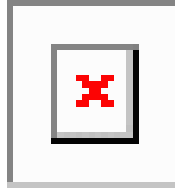
This series of tutorials is a C++ port of the [GLSL wikibook Basic Lighting tutorials](#).

01



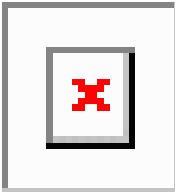
**Diffuse Reflection:**  
about per-vertex diffuse lighting and multiple light sources of different kinds

02



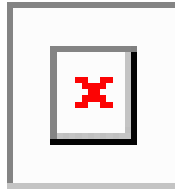
**Specular Highlights:**  
about per-vertex lighting

03



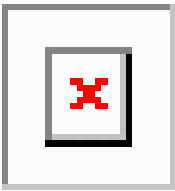
**Two-Sided Surfaces**  
(about two-sided per-vertex lighting)

04



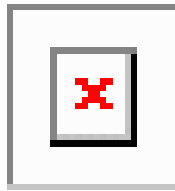
**Smooth Specular Highlights**  
(about per-pixel lighting)

05



**Two-Sided Smooth Surfaces**  
(about two-sided per-pixel lighting)

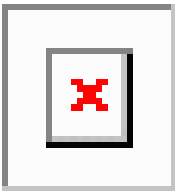
06



**Multiple Lights**  
(about for-loops for handling multiple light sources)

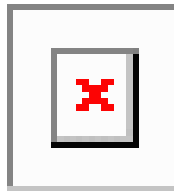
This series of tutorials is a C++ port of the [GLSL wikibook Basic Texturing tutorials](#).

01



**Textured Spheres:**  
about texturing a sphere

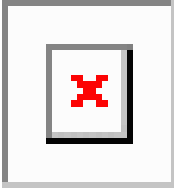
02



**Lighting Textured Surfaces:**  
about textures for diffuse lighting

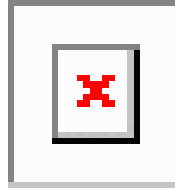


03



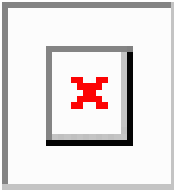
**Glossy Textures:** about gloss mapping

04



**Transparent Textures:** about using alpha textures for discarding fragments, alpha testing, and blending

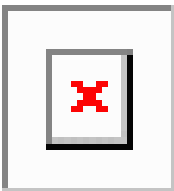
05



**Layers of Textures:** about multi-texturing

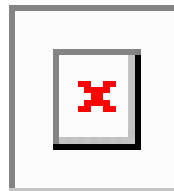
This series of tutorials is a C++ port of the [GLSL wikibook tutorials about Textures in 3D](#).

01



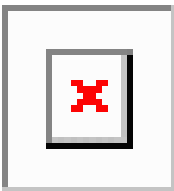
**Lighting of Bumpy Surfaces:** about normal mapping

02



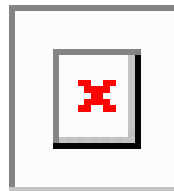
**Projection of Bumpy Surfaces:** about parallax mapping

03



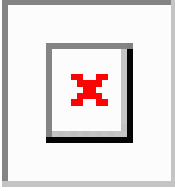
**Cookies:** about projective texture mapping for shaping light

04



**Light Attenuation:** about texture mapping for light attenuation and lookup tables in general

05

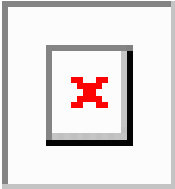


**Projectors:**  
about projective texture mapping for projectors

There are more tutorials to port at the [GLSL wikibook!](#)

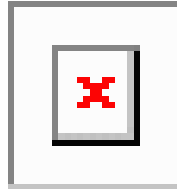
## The scientific arc

01



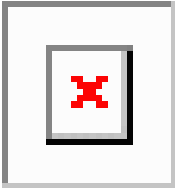
**Graph 01:** plotting a simple function, using vertex buffer objects and point sprites

02



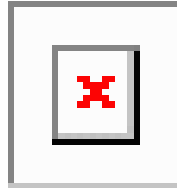
**Graph 02:** plotting a graph from data in a texture

03



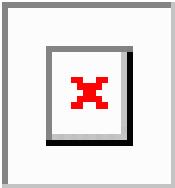
**Graph 03:** plotting borders and axes, clipping

04



**Graph 04:** plotting a three-dimensional graph

05

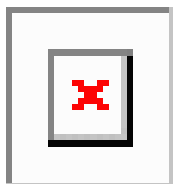


**Graph 05:** plotting a surface with hidden line removal

And more to come.

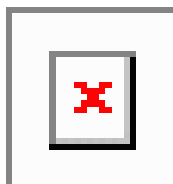
# Selected topics

01



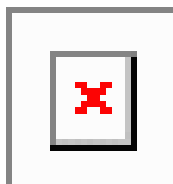
**Arcball:** intuitive object rotation with the mouse

02



**Bounding box:** draw a cube around your object for editing or debugging purposes

03



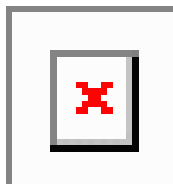
**2D-on-3D:** hardware-accelerated 2D programming

04



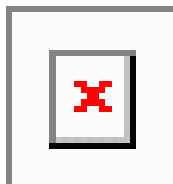
**Video Capture:** using apitrace to capture your animation

05



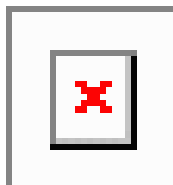
**Tea time:** generating a HD teapot from Bézier surfaces

06



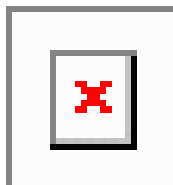
**Stencil buffer:** masking and combining

07



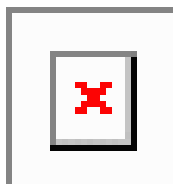
**Quadrics:** creating simple shapes with a bit of maths

08



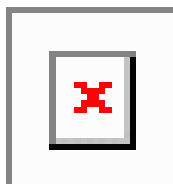
**Basic text:** rendering text using the FreeType library

09



**Optimized text rendering:** using a texture atlas containing all glyphs

10

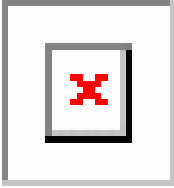


**Object selection:** unprojecting coordinates and object identification using

the stencil  
buffer

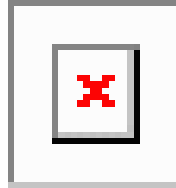
## The post-processing arc

01



Concepts: how to perform full-screen post-processing, first example with a simple animated wave

02

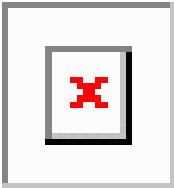


???: next effect to be decided!

## Mini-portal

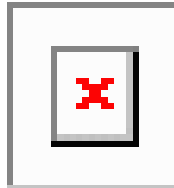
This series shows how to implement a teleportation system similar to Valve's Portal, step-by-step, using OpenGL.

01



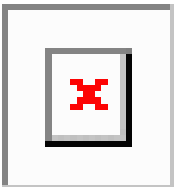
Mini-Portal: a first working see-through portal

02



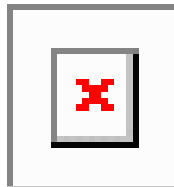
Mini-Portal Smooth: smooth transition, understanding the camera

03



Mini-Portal Recursive: recursive portals - display portals

04



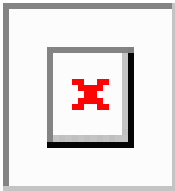
Mini-Portal Optimization: optimization with scissors

within  
portals

# Glescraft

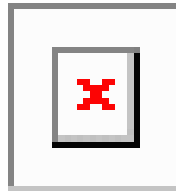
This series shows how to render a voxel based world, similar to [Minecraft](#).

01



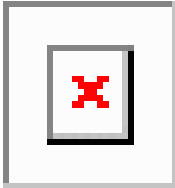
**Glescraft 1:**  
basic voxel  
rendering

02



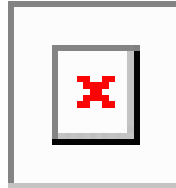
**Glescraft 2:**  
removing  
unnecessary  
voxel faces

03



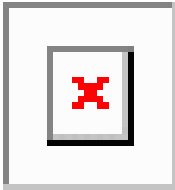
**Glescraft 3:**  
texturing,  
lighting, fog,  
transparency

04



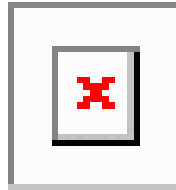
**Glescraft 4:**  
first person  
camera con-  
trols

05



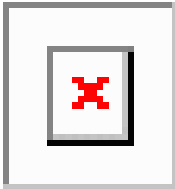
**Glescraft 5:**  
drawing only  
what is on  
screen

06



**Glescraft 6:**  
adding and  
removing  
voxels

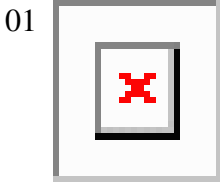
07



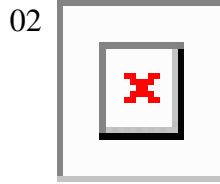
**Glescraft 7:**  
using geo-  
metry shaders



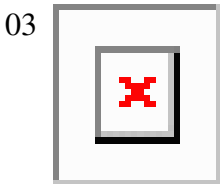
## Using the accumulation buffer



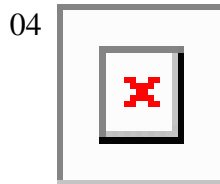
**Tutorial 01:** motion blur



**Tutorial 02:** full-screen supersampling anti-aliasing



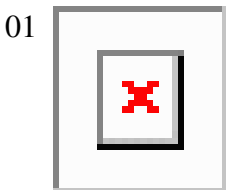
**Tutorial 03:** depth of field



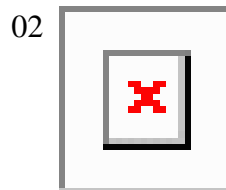
**Tutorial 04:** order-independent transparency

## Cutting-edge OpenGL

If you do not need to target mobile devices or the web, you can upgrade to OpenGL 4.x. It notably introduces 3 new kinds of shaders: Geometry, Tessellation Control and Tessellation Evaluation.



**Tutorial 01:** modify and create vertices on the fly with geometry shaders

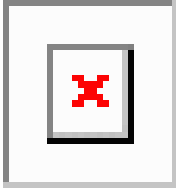


**Tutorial 02:** dynamic mesh quality with tessellation

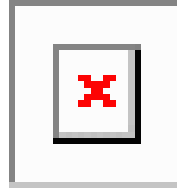
and lots of other features.

## Code quality

01



**Debugging:** tips to help debug your OpenGL code



**Performance:** measuring and improving your application performance

## Appendices

- [OpenGL ES 2.0 Overview](#): OpenGL ES 2.0 (OpenGL for Embedded Systems 2.0) concepts and its differences to normal OpenGL
- [Migrating from 1.x to 2.x](#): how to upgrade your code to use modern OpenGL
- [Glossary](#): what do all those new weird words mean?
- [APIs, Libraries and acronyms](#): how all acronyms relate to each others
- [OpenGL Shading Language](#): elements of GLSL programming
- [Shaders reference](#): input and output variables list
- [Team](#): contributors to this wikibook
- Download code: [Gitorious wikibooks-opengl project](#)

## Legacy OpenGL 1.x

"Legacy" OpenGL is about OpenGL 1.x and OpenGL ES 1.x, with a fixed pipeline and no shaders.

## Starting Tutorial

1. [Setting Up A Programming Environment On Windows](#)
  - ▣
2. [Setting Up OpenGL In The Programming Environment](#)
  - ▣
3. [Drawing Primitives](#)
  - ▣
    1. Immediate Mode
    2. Display Lists
    3. Vertex Arrays
4. [Basic Transformations](#)
  - ▣
    1. Translation
    2. Rotation
    3. Scaling
    4. Custom Transformations

## Basics

1. [Structure of a Typical OpenGL Application](#)
  - ▣
2. [Drawing Rectangles](#)
  - ▣
3. [Drawing Lines and Points](#)
4. [Drawing Simple 2D Shapes](#)





5. OpenGL Naming Conventions



6. Using Color



7. Viewing Transformations



8. Drawing Simple 3D Objects

9. Perspective versus Orthographic Projections

## Intermediate

1. Smoothing Polygons with Normals

2. Adding Lights

3. Using Materials

4. Using Textures

5. Using Mipmaps

6. Drawing Complex Polygons Using Tessellation

## Advanced

1. Optimizing OpenGL Code

2. Drawing Shadows

3. Drawing Using Quadrics

4. Drawing Using NURBS and Curves

5. Ambient Occlusion

# Appendices

1. [Coordinate Transformations](#)
2. [Understanding Transformation Matrices](#)
3. [OpenGL Library Reference](#). functions and type reference for gl.h glu.h and glut.h
4. [Why OpenGL Exists and What It's Good For](#)
5. [Migrating from 1.x to 2.x](#): how to upgrade your code to use modern OpenGL

# External links



Also see the *GLSL Programming* book.

# Wikibooks

Related WikiBooks:

- [GLSL Programming](#) : wikibook on the use of the OpenGL Shading Language (GLSL) in Unity 3 and Blender 2.5, with much information on lighting and texturing
- [Blender 3D: Noob to Pro](#): comprehensive book on using the Blender 3D modeling environment

## Ports

The following websites provide conversion of the tutorials to other programming languages or platforms:

- <https://github.com/alihelmy/lwjglTutorial> : port to Lightweight Java Game Library (LWJGL) ([blog](#))
- <http://littlefootlandmines.blogspot.com/search/label/opengl%20tutorials> : not really a port, but commented, progressive version of the tutorials code as the author follows them

## Freely-licensed documentation and samples

- [Learning Modern 3D Graphics Programming](#): another modern OpenGL tutorial, requiring v3.3 or later, MIT license ([source repository](#))
- [OpenGL Samples Pack](#): written in C++ based on the "core profile" specifications, aimed at easing upgrades to new OpenGL versions and features, MIT license ([source repository](#))
- [Introduction and examples for OpenGL 3.3](#): GFDL license
- [A modern OpenGL FAQ](#): GFDL license
- [OpenGLContext Python Tutorials](#): basic tutorials using PyOpenGL (BSD-style licenses)
- [Paul's projects](#): OpenGL 1.x, two tutorials on shadow mapping and bump mapping, with code under the MIT license

## Non-freely-licensed documentation

- [NeHe's OpenGL tutorial](#): OpenGL 1.x

# OpenGL Programming

---

- [Collection of OpenGL fundamentals tutorials using C++ and Windows API: OpenGL 1.x](#)
- [ZeusCMD OpenGL Tutorials: OpenGL 1.x](#)
- [OpenGL Programming Examples: OpenGL 1.x,](#)

## Websites

- [Official OpenGL Website](#)
- [Mesa, an open-sourced 3D graphics library almost identical to OpenGL](#)

## Further reading

- [OpenGL Architecture Review Board, et al: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2, Fifth Edition, Addison-Wesley, ISBN 0-321-33573-2](#)
- [OpenGL Architecture Review Board, et al: OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.4, Addison-Wesley, ISBN 0-321-17383-X](#)
- [Wright, Richard S. Jr and Lipchak, Benjamin: OpenGL SuperBible, Third Edition, Sams Publishing, ISBN 0-672-32601-9](#)



[< OpenGL Programming](#)

Browse & download [complete code](#)