1.

Record Nr.	UNINA9910956648603321
Autore	Wolff David
Titolo	OpenGL 4.0 shading language cookbook : over 60 highly focused, practical recipes to maximize your use of the OpenGL shading language / / David Wolff
Pubbl/distr/stampa	Birmingham [England], : Packt Pub. Ltd., 2011
ISBN	9786613308184 9781283308182 1283308185 9781849514774 1849514771
Edizione	[1st ed.]
Descrizione fisica	1 online resource (767 p.)
Disciplina	006.6
Soggetti	Programming languages (Electronic computers)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes index.
Nota di contenuto	Table of Contents; OpenGL 4.0 Shading Language Cookbook; OpenGL 4.0 Shading Language Cookbook; Credits; About the Author; About the Reviewers; www.PacktPub.com; Support files, eBooks, discount offers and more; Why subscribe?; Free access for Packt account holders; Preface; What this book covers; What you need for this book; Who this book is for; Conventions; Reader feedback; Customer support; Downloading the example code for this book; Errata; Piracy; Questions; 1. Getting Started with GLSL 4.0; Introduction; The OpenGL Shading Language; Profiles: Core vs. Compatibility Using the GLEW Library to access the latest OpenGL functionalityGetting ready; How to do it...; How it works...; There's more...; GLEW visualinfo; GLEW glewinfo; Checking for extension availability at runtime; See also; Using the GLM library for mathematics; Getting ready; How to do it...; How it works...; There's more...; Using the GLM types as input to OpenGL; See also; Determining the GLSL and OpenGL version; How to do it...; How it works...; There's more...; See also; Compiling a shader; Getting ready; How to do it...; How it works...; There's more...; Deleting a shader object; See also

Linking a shader programGetting ready; How to do it...; How it works...; There's more...; Deleting a shader program; See also; Sending data to a shader using per-vertex attributes and vertex buffer objects; Getting ready; How to do it...; How it works...; There's more...; Using layout qualifiers; Using element arrays; Interleaved arrays; See also; Getting a list of active vertex input attributes and indices; Getting ready; How to do it...; How it works...; There's more...; See also; Sending data to a shader using uniform variables; Getting ready; How to do it...; How it works...

There's more...See also; Getting a list of active uniform variables; Getting ready; How to do it...; How it works...; There's more...; See also; Using uniform blocks and uniform buffer objects; Getting ready; How to do it...; How it works...; There's more...; Using an instance name with a uniform block; Using layout qualifiers with uniform blocks; See also; Building a C++ shader program class; Getting ready; How to do it...; How it works...; See also; 2. The Basics of GLSL Shaders; Introduction; Vertex and fragment shaders; Replicating the old fixed functionality

Implementing diffuse, per-vertex shading with a single point light sourceGetting ready; How to do it...; How it works...; There's more...; See also; Implementing per-vertex ambient, diffuse, and specular (ADS) shading; Getting ready; How to do it...; How it works...; There's more...; Using a non-local viewer; Per-vertex vs. Per-fragment; Directional lights; Light attenuation with distance; See also; Using functions in shaders; Getting ready; How to do it...; How it works...; There's more...; The const qualifier; Function overloading; Passing arrays or structures to a function; See also

Implementing two-sided shading

Sommario/riassunto	Over 60 highly focused, practical recipes to maximize your OpenGL Shading language use