1992 Version 1.0 Version 1.1 - Vertex Arrays 1997 - Texture Objects - Polygon Offset Version 1.2 - 3D-Texturen 1998 - Neue Pixelformate - Level-Of-Detail-Texturen Version 1.3 - ARB Extensions - ARB Multitexture Version 1.3 - Komprimierte Texturen 2001 - Cube-Maps - Multitexturing Version 1.4 - Tiefentexturen 2002 - Autom. Mip-Map-Erzeugung Nebelkoordinaten Version 1.5 2003 - Pufferobjekte - Occlusion Queries Version 2.0 - Shaderprogramme (GLSL) 2004 - Multiple Render Targets - Texturen belieber Größe Version 2.1 - Pixel Buffer Objects 2006 - GLSL 1.20 - sRGB-Texturen Version 3.0 "Longs Peak" Codebasis aufgeräumt 2008 - GLSL 1.30 - Entfernung von Altlasten Version 3.1 - Schichtmodell-Architektur "Long Peak Reloaded" 2009 - Entfernung von Altlasten - GLSL 1.40 - Uniform Buffer Objects - Vertex Shader erlauben mindetens 16 Texture-lookups - Primitive Restart - Instancing - CopyBuffer API Version 3.2 - Ausführungsgeschwindigkeit erhöht - Darstellungsqualität verbessert - Geometry Processing beschleunigt - Portierung von Direct3D-Anwendungen erleichtert - GLSL 1.50 - Direkte Open-GL Kern Unterstützung für Geometry-Shader - Fünf ARB-Erweiterungen Version 3.3 - Einbindung von OpenCL - GLSL 3.30 2010 Version 4.0 - Tessellation - GLSL 4.00 Version 4.1 - Laden von binären Shaker-Programmen - 64-Bit Floating-Point-Komponenten bei Vertex-Shader-Eingabe - Mehrere Viewports für ein Render Surface - GLSL 4.10 Version 4.2 2011 - GLSL 4.20 Version 4.3 2012 - Compute Shader-Programme - GLSL 4.30 - Plattformübergreifende Version 4.4 **Texturkompression** - Mehr Kontrolle über die - Mehr Stabilität zwischen den Platzierung der Buffer im OpenGL-Anwendungen Speicher - GLSL 4.40 2013 - Mehrere Objekte gleichzeitig an Context binden - Compute Shader mit variabler Version 4.5 Work-Group Größe - Direct State Access (DSA) - Flush Control 2014 - Robustheit - OpenGL ES 3.1 API und Shader- Kompatibilität - DX11 Emulations Feature