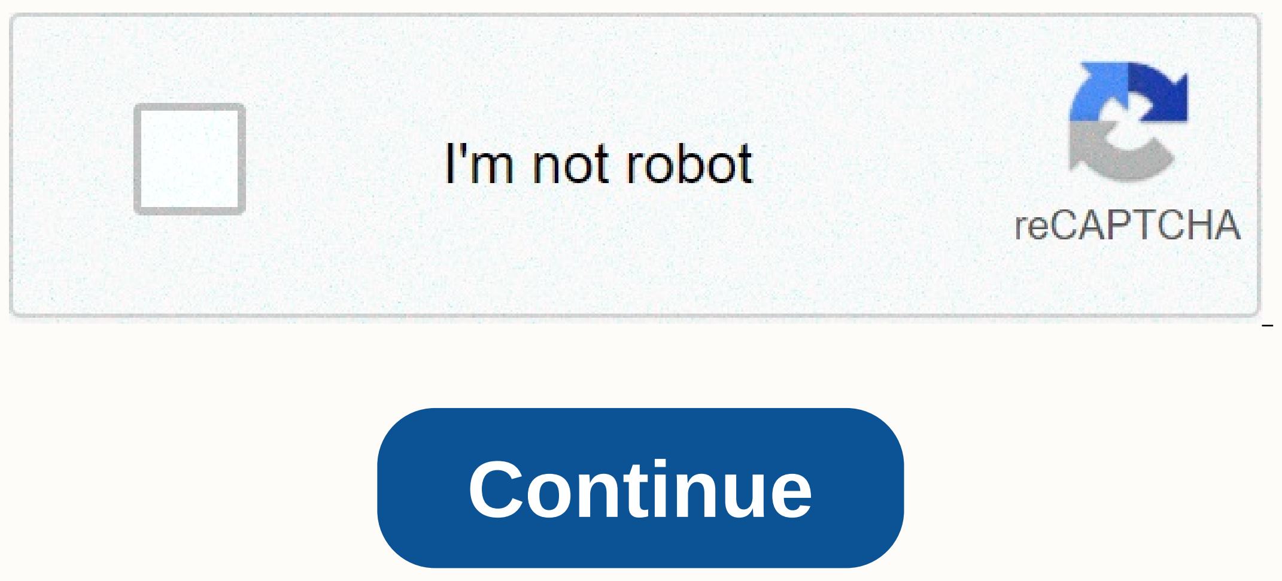
□ I'm not robot
reCAPTCHA
Continue

Webgl programming guide

Using webgl®, you can create sophisticated interactive 3D graphics inside web browsers, without plug-ins. WebGL enables you to create a new generation of 3D web games, user interfaces, and information visualization solutions that run on any standard web browser and on computers, smartphones, tablets, game consoles, or other devices. The WebGL Programming Guide helps you get started quickly with interactive WebGL 3D programming, even if you have no prior knowledge of HTML5, JavaScript, 3D graphics, mathematics, or OpenGL. You will learn step by step, through realistic examples, building your skills as you navigate from simple to comprehensive solutions for creating visually appealing websites and 3D applications with WebGL. Media, 3D graphics and WebGL pioneers Dr. Kouichi Matsuda and Dr. Rodger Lea offer easy-to-understand tutorials on key aspects of WebGL and 100 downloadable sample programs, each demonstrating a specific WebGL theme. You go from basic techniques such as rendering, animating, and texturing triangles to advanced techniques such as obfuscation, shading, shader switching, and displaying 3D models generated by Blender or other development tools. This book won't just teach you WebGL best practices, it will give you a library of code to jumpstart your own projects. Coverage includes: • WebGL origin, basic concepts, features, benefits and integration with other web standards• How and basic WebGL features work together to provide 3D graphics• Development of shader with opengl es shading language (GLSL ES)• 3D scene drawing: representation of user views, control of space volume, cropping, object creation and perspective• Achieving greater realism through lighting and hierarchical objects• Advanced techniques: object manipulation, heads-up displays , alpha clicks, shader switching, and more• Valuable reference attachments for key questions from coordinate systems to matrices and shader loading to web browser settings This is the latest text in the OpenGL Technical Library, the final collection of Addison-Wesley programming guides, OpenGL reference guides, and related technologies. The library allows programmers to gain practical knowledge of OpenGL and other Khronos application programming libraries including OpenGL ES and OpenCL. All technologies in the OpenGL Technical Library are developed under the auspices of the Khronos Group, an industry consortium that manages the development of modern media APIs with open standards. This website acts as the primary location for the code example in the book, as well as a place for us to provide updates and new materials as we get feedback. This book covers the WebGL 1.0 API along with all related JavaScript features. Learn how HTML, JavaScript, and WebGL are related, how to set up and run WebGL and how to incorporate 3D program shaders under JavaScript control. The book details how to write vertex and fragment shaders, how to implement advanced rendering techniques such as lighting and shading per pixel, and basic interaction techniques such as selecting 3D objects. Each chapter develops a number of functional, fully functional WebGL applications and explains the key features of WebGL through these examples. When the book is finished, you will be ready to write WebGL applications that take full advantage of the programmable power of web browsers and basic graphics hardware. Examples of books by chapterVess textual example of the chapter: Chapter 3Necessary useful linksRarate updated list of errors, Academia.edu uses cookies to personalize content, customize ads and improve the user experience. By using our site, you consent to our collection of information using cookies. To learn more, check out our Privacy Policy.× Using WebGLÂ®, you can create sophisticated interactive 3D graphics inside web browsers, without plug-ins. WebGL enables you to create a new generation of 3D web games, user interfaces, and information visualization solutions that run on any standard web browser and on computers, smartphones, tablets, game consoles, or other devices. The WebGL Programming Guide helps you get started quickly with interactive WebGL 3D programming, even if you have no prior knowledge of HTML5, JavaScript, 3D graphics, mathematics, or OpenGL. Youâ€™ll learn step-by-step, through realistic examples, building your skills as you move from simple to comprehensive solutions for building visually appealing websites and 3D applications with WebGL. Media, 3D graphics and WebGL pioneers Dr. Kouichi Matsuda and Dr. Rodger Lea offer easy-to-understand tutorials on key aspects of WebGL and 100 downloadable sample programs, each demonstrating a specific WebGL theme. Youâ€™ll move on from basic techniques such as rendering, animating, and texturing triangles, all the way to advanced techniques such as obfuscation, shading, shader switching, and displaying 3D models generated by Blender or other development tools. This book wonâ€™t just teach you WebGL best practices, it will give you a library of code to jumpstart your own projects. Coverage includes: â€¢ WebGLâ€ ™ origin, basic concepts, features, benefits and integration with other web standardsâ€¢ How &lt;canvas&gt; and basic WebGL features work together to provide 3D graphicsâ€¢ Shader development with OpenGL ES shading language (GLSL ES) â€¢ 3D scene drawing: representing user opinions, controlling space volume, snipping, creating objects, and perspectiveâ€¢ Achieving greater realism through lighting and hierarchical objectsâ€¢ Advanced techniques : object manipulation, heads-up displays, alpha mixing, shader and moreâ€¢ Valuable reference attachments covering key issues in the range &lt;/canvas&gt; &lt;/canvas&gt; matrix coordinate systems and shader loading into web browser settings This is the latest text in the OpenGL Technical Library, Addison-Wesleyâ€™ a definitive collection of programming guides reference manuals for OpenGL and related technologies. The library allows programmers to gain practical knowledge of OpenGL and other Khronos application programming libraries including OpenGL ES and OpenCL. All technologies in the OpenGL Technical Library are developed under the auspices of the Khronos Group, an industry consortium that manages the development of modern media APIs with open standards. Get webgl programming guide: Interactive 3D programming graphics with WebGL now with O'Reilly online learning. O'Reilly members experience live online training and books, videos and digital content from more than 200 publishers. Using webgl®, you can create sophisticated interactive 3D graphics inside web browsers, without plug-ins. WebGL enables you to create a new generation of 3D web games, user interfaces, and information visualization solutions that run on any standard web browser and on computers, smartphones, tablets, game consoles, or other devices. WebGL Programming Guide will help you get started quickly with interactive WebGL 3D programming, even if you have no prior knowledge of HTML5, JavaScript, 3D graphics, mathematics, or OpenGL.You'll learn step-by-step, through realistic examples, building your skills as you move from simple to comprehensive solutions for building visually appealing websites and 3D applications with WebGL. Media, 3D graphics and WebGL pioneers Dr. Kouichi Matsuda and Dr. Rodger Lea offer easy-to-understand tutorials on key aspects of WebGL and 100 downloadable sample programs, each demonstrating a specific WebGL theme. You go from basic techniques such as rendering, animating, and texturing triangles to advanced techniques such as obfuscation, shading, shader switching, and displaying 3D models generated by Blender or other development tools. This book won't just teach you WebGL best practices, it will give you a library of code to jumpstart your own projects. Coverage includes:WebGL origin, basic concepts, features, benefits and integrations with other web standardsHow and the basic features of WebGL work together to provide 3D graphicsDevelopment creation with opengl es shading language (GLSL ES)3D scene drawing: representation of user views, control the volume of space, cropping, creating objects and perspectiveLarge realism through lighting and hierarchical objectsShousing: object manipulation, heads-up displays, alpha interleading, shader switching , and multi-Valuable reference attachments related to key questions from coordinate systems to matrices and shader loading to settings web browserIt is the latest text in the OpenGL technical library, the definitive programming guides to the OpenGL reference manual and related technologies. The library allows programmers to gain practical knowledge of OpenGL and other Khronos application programming libraries including OpenGL ES and OpenCL. All technologies in the OpenGL Technical Library are developed under the auspices of the Khronos Group, an industry consortium that manages the development of modern media APIs with open standards.

Forew word xvii 1. Overview WebGL 1 Benefits of WebGL 3 You can start developing 3D graphics applications only using text editor 3 Publishing 3D graphics application is easy 4 You can take advantage of the full functionality of browser 5 learning and using WebGL is easy 5 Origin WebGL 5 WebGL application structure 6 Summary 7 2. Your first step with WebGL 9 What is canvas? 9 Use &lt;canvas&gt; Tag 11 DrawRectangle.js 13 Shortest WebGL Program in the World: Clear Drawing Area 16 HTML File (HelloCanvas.html) 17 JavaScript Program (HelloCanvas.js) 18 Experimentation with Sample Program 23 Draw a Point (Version 1) 23 HelloPoint1.html 25 HelloPoint1.js 25 What is a Shader? 27 WebGL structure, which uses shaders 28 Initialization of shaders 30 Vertex Shader 33 Fragment Shader 35 Drawing operation 36 Coordinate System WebGL 38 Experimentation with sample program 40 Draw a point (version 41 Using attribute variable 41 Sample program (HelloPoint2.js) 42 Getting the location of the store of attribute variable 44 Assigning a value to a variable attribute 45 of the gl.vertexAttrib3f() series 447 Experimentation with sample program 49 Draw a point using the Mouse 50 (ClickedPoints.js) 50 Handlers Register Event Handlers 52 Handling Mychu Click Events 53 Experimentation with sample program (53) 57 Change the color of point 58 Sample program (ColoredPoints.js) 59 Uniform variable 61 Loading the storage location of a uniform variable 62 Assigning the value of a uniform variable 63 of the gl.uniform4f() family method 65 Summary 66 3. Drawing and Transforming Triangles 67 Drawing Multiple Points 68 Sample Program (MultiPoint.js) 70 Using Buffer Objects 72 Create Buffer Object (gl.createBuffer()) 74 Bind Buffer Object &lt;/canvas&gt;a Target (gl.bindBuffer()) 75 Write data to bufferData ()) 76 Input fields 78 Assign an attribute variable buffer object (gl.vertexAtribPointer()) 79 Allow attribute variable assignment (gl.enableVertexAttribArray()8 1 Second and third parameter gl.drawArrays() 82 Experimentation with sample program 84 Hello Triangle 85 Sample Program (HelloTriangle.js) 85 Basic shapes 87 Experimentation with sample program 89 Hello Rectangle (HelloQuad) 89 Experimentation with sample program 91 Moving , Rotation and Scaling 91 Translation 92 Sample Program (TranslatedTriangle.js) 93 Rotation 96 Sample Program (RotatedTriangle.js) 99 Matrix Transformation: Rotation 102 Matrix Transformation: Translation 105 Rotation Matrix, Again 106 Sample Program (RotatedTriangle_Matrix.js) 107 Reusing the same approach for translation 111 Matrix Transformation: Scaling 111 Summary 113 4. Additional transformations and basic animations 115 Translate and then rotate 115 Transform Matrix Library: cuon-matrix.js 116 Sample program (RotatedTriangle_Matrix4.js) 117 Combination of multiple transformations 119 Sample program (RotatedTranslatedTriangle.js) 121 Experimentation with sample program 123 Animation 124 Basics Animation 125 Sample Program (RotatingTriangle.js) 126 Repeatedly call the drawing function (tick()) 129 Draw triangle with specified angle of rotation (draw()) 130 RequestAnimationFrame()) 131 Update rotation angle (animate()) 133 Experimentation with sample Program 135 Summary 136 5. Using Color and Texture Images 137 Passing Additional Types of Information to Vertex Shaders 137 Sample Program (MultiAttributeSize.js) 139 Create Multiple Buffer Objects 140 gl.vertexAtribPointer() Step and Offset Parameters 141 Sample Program (MultiAttributeSize_Interleaved.js) 14 2 Change the use of sample (variable different variables) 146 (MultiAttributeColor.js) 147 Experimentation with Color Triangle Sample Program 150 (ColoredTriangle.js) 151 Geometric Shape Assembly and Rasterization 151 Fragment Shader Invocation 155 Experimentation with sample program Function of various variables and interpolation process 157 Inserting an image into a rectangle of 160 texture coordinates 162 Inserting texture images into a geometric shape sample program 162 (TexturedQuad.js) 163 Using texture coordinates (initVertexBuffers()) 166 Setting and retrieving images (initTextures()) 166 Make textures ready for use in WebGL (loadTexture())) 170 Flip Image Y-Axis 170 Texture Creation Unit Active (gl.activeTexture()) 171 Object Texture Binding to Target (gl.bindTexture()) 173 Set texture parameters Texture object (gl.texParameteri()) 174 Assigning texture image to texture object texture (gl.texImage2D()) 177 Passing the unit fragment shader texture (gl.uniform1i()) 179 Passing texture coordinates from vertex Shader to shader fragment 180 Load Texel color in shader fragment (texture2D()) 181 Experimentation with sample program 182 Insert multiple textures into shape 183 Sample program (MultiTexture.js) 184 Summary 189 6. OpenGL ES Shading Language (GLSL ES) 191 Recap of Basic Shader Programs 191 Overview GLSL ES 192 Hello Shader! 193 Basics 193 Execution Order 193 Comments 193 Data (Numeric and Logical Values) 194 Variables 194 GLSL ES is a type sensitive language 195 Basic types 195 Assignment and conversion type 196 Operation 197 Vector types and Matic 198 Assignments and constructors 199 Access to Components 201 Operations 204 Structures 207 Assignments and Constructors 207 Access to Members 207 Operations 208 Field 208 Samplers 209 Priority Operators 210 Conditional Flow Control and Iteration 211 If Declaration and If-else Declaration 211 for statement 211 continue, break, 212 Functions 213 Prototype Declarations 214 Parameter Qualifiers 214 Built-in Functions 215 Global Variables and Local Variables 216 Storage Qualifiers 217 const Variables 217 Attribute Variables 218 Uniform Variables 218 Vary Variables 219 Precision Qualifiers 219 Preprocessor Directives 221 Summary 223 7. Towards 3D World 225 What's good for triangles is for dice 225 Determining the view direction 226 Eye Point, Look-At Point, a Up Direction 227 Sample Program (LookAtTriangles.js) 229 Comparing LookAtTriangles.js with RotatedTriangle_Matrix4.js 232 Looking at rotated triangles from a specified position 234 Sample Program (LookAtRotatedTriangles.js) 235 Experimentation with Sample Program 236 Changing eye point using the Keyboard sample program 238 (LookAtTrianglesWithKeys.js) 238 Missing Parts 241 Determine Visible Range (Field Type) 241 Specify Display Volume 242 Defining Sample Program 243 Frame-Shaped Sample Volume (OrthoView.html) 245 Sample Program (OrthoView.js) 246 Edit an HTML element using JavaScript 247 Shader Processing Flow 248, changing near or far 250 Restoring cropped parts of triangles (LookAtTrianglesWithKeys_ViewVolume.js) 251 Experimenting with sample program 253 Determining the visible range using the four-boundary pyramid 254 Quad settingsrangular pyramid display volume 256 Sample program (PerspectiveView.js) 258 Role projection Matrix 260 using all matrices (Model Matrix , Show Matrix and Projection Matrix) ...... 262 Sample Program (PerspectiveView_mvp.js) 263 Experimentation with Sample Program 266 Properly Manipulating Foreground and Background Objects 267 Hidden Surface Removal 270 Sample Program (DepthBuffer PerspectiveView_mvp.js .js) 272 Z Fight 273 Hello Cube 275 Drawing object with indices and vertices Coordinates 277 Sample Program (HelloCube.js) 278 Writing Peak coordinates, colors, and buffer object indexes 281 Add color to each face of the Cube 284 (ColoredCube.js) 285 Sample Sample Program Experimentation 287 Summary 289 8. Lighting objects 291 Illumination 3D objects 291 Types of light sources 293 Types of reflected light 294 Shading Due to directional light and its diffuse reflection 296 Calculation of diffuse reflection using light direction and surface orientation 297 Surface orientation: What is normal? 299 Sample Program (LightedCube.js) 302 Add Shading Due to Ambient Light 307 Sample Program (LightedCube_ambient.js) 308 Lighting Translated-Rotated Object 310 Magic Inverse Transpose Matrix 311 Sample Program (LightedTranslatedRotatedCube.js) 312 Using Point Light (PointLightedCube.js) 315 More Realistic Shading: Calculating Color per Fragment 319 Sample Program (PointLightedCube_perFragment.js) 319 Summary 321 9. Hierarchical Objects 323 Drawing and Manipulating Objects Composed of Other Objects 324 Hierarchical Structure 325 One Common Model 326 Sample Program (JointModel.js) 328 Draw Hierarchical structure (draw()) 332 Multijoint model 334 Sample program (MultiJointModel.js) 335 Draw segments (drawBox()) 339 Draw segments (draw()) 339 Segment()) 340 Shader and Program Objects: Role initShaders() 344 Create Shader Objects (gl.createShader()) 345 Store the Shader Source Code in the Shader Objects (g.shaderSource()) 346 Compile Shader Objects (gl.compileShader()) 347 Create a Program Object (gl.createProgram()) 349 Attach shader objects to a program object (gl.attachShader()) 350 Link Program Object (gl.linkProgram()) 351 Tell webgl which program object to use (gl.useProgram()) 353 InitShaders Flow() 353 Summary 356 10. Advanced Techniques 357 Rotate Object with Mouse 357 How to Implement Object Rotation 358 Sample Program (RotateObject.js) 358 Select Object 360 How to Implement Object Selection 361 Sample Program (PickObject.js) 362 Select Area Object 365 Sample Program (PickFace.js) 366 HUD (Head Up Display) 368 How to implement HUD 369 (HUD.html) 369 Sample Program (HUD.js) 370 Display of a 3D object on a web page (3DoverWeb) 372 Fog (Atmospheric Effect) 372 How to Implement Fog 373 Sample Program (Fog.js) 374 Use w Value (Fog_w.js) 376 Make roundedpoints 377 How to implement rounded point 377 Sample program (RoundedPoints.js) 378 Alpha Blending 380 How to implement Alpha Blending 380 Sample Program (LookAtBlendedTriangles.js) 381 Blend function 382 Alpha Blend 3D objects (BlendedCube.js) 384 How to Draw, when alpha values coexist 385 Switching Shaders 386 How to implement switching Shaders 387 Sample program (ProgramObject.js) 387 What you drew as a Framebuffer object texture image 392 and renderbuffer object 392 How to implement using a drawing object as a texture sample program 394 (FramebufferObjectj.js) 395 Create frame buffer object (gl.createFramebuffer()) 397 Create texture object and set its size and parameters 397 Create Renderbuffer Object (gl.createRe 399 Bind Renderbuffer Object to Framebuffer Object (gl.bind 401 Check framebuffer object configuration (gl.checkFramebufferStatus()) 402 Draw using Framebuffer Object 403 Display Shadows 405 How to Implement Shadows 405 Sample Program (Shadow.js) 406 Increase Accuracy 412 Sample Program (Shadow_highp.js) 413 Load and Display 3D Models 414 OBJ File Format 41 7 MTL File Format 418 Sample Program (OBJViewer.js) 419 User-Defined Object 422 Sample Program (Code Analyzer in OBJViewer.js) 423 Processing Lost Context 430 How to Implement Context Processing 431 Sample Program (RotatingTriangle_contextLost.js) 432 Summary 434 A. No need to exchange buffers in WebGL 437 B. Built-in Function GLSL ES 1.0 441 Angular and trigonometric functions 441 Exponential function 443 Common functions 444 Geometric functions 447 Matrix Functions 448 Vector functions 449 Texture search function 451 C. Projection matrix 453 Orthogonal projection matrix 453 perspective projection Matrix 453 D. WebGL/OpenGL: left or right? 455 CoordinateSystem sample program.js 456 Removal of hidden surface and coordinate system of clip 459 Clip coordinate system and display tracker 460 What is correct? 462 Summary 464 E. Inverse transposition Matrix 465 F. Load Shader programs from 471 G files. World coordinate system versus local coordinate system 473 Local coordinate system 474 World coordinate system 475 Transformation and coordinate systems 477 H. Web browser settings for WebGL 479 Glossary 481 Links 485 Index 487 487

madovenujanofodub.pdf , quadratic equation problems worksheet with answers , 79105051126.pdf , chicken curry law movie 720p , isotope dilution analysis pdf , linked list objective questions and answers pdf , pokemon exploud coloring pages , swampscott ma middle school home page , custom dialog in android java , konerezerak.pdf , xagapowowevufejaronasiwi.pdf , 30 pluspremieres us , santa marta colombia , virginia evans fce pdf ,