



OpenGL. Programowanie gier

Kevin Hawkins, Dave Astle

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki:

"OpenGL. Programowanie gier"

Jest to lista błędów znalezionych po opublikowaniu książki, zgłoszonych i zaakceptowanych przez naszą redakcję. Pragniemy, aby nasze publikacje były wiarygodne i spełniały Twoje oczekiwania. Zapoznaj się z poniższą listą. Jeśli masz dodatkowe zastrzeżenia, możesz je zgłosić pod adresem

<https://ebookpoint.pl/user/erraty>

Strona	Linia	Jest	
43	43	BeginPaint(hwnd, &ps); EndPaint(hwnd, &ps);	BeginPaint(h &paintStru EndPaint(h &paintStru
55	1. w ramce	MSDrN	MSDN
55	ramka, 4 wiersz	http://msdn.microsoft.com	http://msdn.microsoft.com
68	5. od góry	<code>memset(&dmScreenSettings,0,sizeof(dmScreenSettings));</code>	<code>memset(&devModeScreen,0,s</code>
68	6. od góry	<code>devModeScreen.dmSize = sizeof(dmScreenSettings);</code>	<code>devModeScreen = sizeof(devMode</code>
74	19	A.B>0, jeśli kąt pomiędzy wektorami A i B jest większy od 90 stopni	A.B

76	4 i 5 od dołu	$C = k * M = 2 * \begin{vmatrix} 1 & 3 \\ 2 & 6 \end{vmatrix}$ $= \begin{vmatrix} (2*1) & (2*3) \end{vmatrix} = \begin{vmatrix} 2 & 6 \\ (2*2) & (2*0) \end{vmatrix} = \begin{vmatrix} 4 & 0 \end{vmatrix}$	$C = k * M = 2 * \begin{vmatrix} 1 & 3 \\ 2 & 6 \end{vmatrix}$ $= \begin{vmatrix} (2*1) & (2*3) \\ (2*2) & (2*0) \end{vmatrix} = \begin{vmatrix} 2 & 6 \\ 4 & 0 \end{vmatrix}$
78	1	wiersz n00 n01 n02 jest umieszczony jako pierwszy wiersz macierzy M	powinien to pierwszy wiersz macierzy
81	2	$M_{OZ} = \begin{vmatrix} \cos(f_i) & -\sin(f_i) \\ 0 & 0 \end{vmatrix}$ $\begin{vmatrix} \sin(f_i) & \cos(f_i) & 0 & 0 \end{vmatrix}$ $\begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix}$ $\begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix}$	$M_{OZ} = \begin{vmatrix} \cos(f_i) & -\sin(f_i) \\ 0 & 0 \end{vmatrix}$ $\begin{vmatrix} -\sin(f_i) & \cos(f_i) & 0 & 0 \end{vmatrix}$ $\begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix}$ $\begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix}$
94	1. od góry	<pre>void geGetBooleanv(GLenum pname,GLboolean *params);</pre>	<pre>void glGetBooleanv(pname,GLbo *params</pre>
94	2. od góry	<pre>void geGetDoublev(GLenum pname,GLdouble *params);</pre>	<pre>void glGetDoublev(pname,GLd *params</pre>
94	3. od góry	<pre>void geGetFloatv(GLenum pname,GLfloat *params);</pre>	<pre>void glGetFloatv(G pname,GLf *params</pre>
94	4. od góry	<pre>void geGetIntegerv(GLenum pname,GLint *params);</pre>	<pre>void glGetIntegerv(pname,GLint *p</pre>
107	6-8 od dołu	<pre>glVertex(-2.0, -1.0, 0.0); glVertex(3.0, 1.0, 0.0); glVertex(0.0, 3.0, 0.0);</pre>	<pre>glVertex3f(-2. 0.0); glVertex3f(3. 0.0); glVertex3f(0. 0.0);</pre>

116	3. od dołu	glRotatef(roll, 1.0f, 0.0f, 0.0f);	glRotatef(pitc 0.0f, 0.0f
117	5 od dołu	(5, 4,33)	(4,3)
131	9 od dołu	hwnd = CreateWindowEx(NULL,	hwnd = CreateWindowEx(
150	18	normal[0] = vector1[1]*vector2[2] - vector1[2]*vector2[2]	normal[0] vector1[1]*vec vector1[2]*ve
150	4	z wierzchołka P2	z wierzchoł
160	2. pod tabelą 6.3	glLightfv(GL_LIGHT0, GL_ATTENUATION, 4.0f);	glLightfv(GL_L GL_CONSTANT_AT 4.0f);
163	10 od dołu	void glMaterialf(GGLenum face, GGLenum pname, TYPE param);	void glMaterialfv(G face, GGLenum TYPE para
165	10 od dołu	glLightModel(GL_LIGHT_MODEL_AMBIENT, ambientLightModel);	glLightModelfv(GL_LIGHT ambientLight
165	9, 10	void glLightModel[i](GLenum pname, TYPE param); void glLightModel[i]v(GGLenum pname, TYPE *param);	void glLightModelfv(pname, TYPE p void glLightModelfv(pname, TYPE *
190	18 od dołu	BITMAPFILEHEADERbitmapFileHeader;	BITMAPFILEH bitmapFileHe
190	29	unisgned chartempRGB;	unisgned c tempRG

191	24	<pre> for (imageldx = 0; imageldx < biSizeImage; imageldx+=3) tempRGB = bitmapImage[imageldx]; bitmapImage[imageldx] = bitmapImage[imageldx + 2]; bitmapImage[imageldx + 2] = tempRGB; </pre>	<pre> for (imageldx = 0; imageldx < biSizeImage; imageldx+=3) tempRGB = bitmapImage[imageldx]; bitmapImage[imageldx] = bitmapImage[imageldx + 2]; bitmapImage[imageldx + 2] = tempRGB; </pre>
193	2 od dołu	wartość 0 x 00 = 24 bity, wartość 0 x 08 = 32-bity	wartość 0x00 = 24 bity, wartość 0x08 = 32 bity
208	22. od góry	Parametr param reprezentuje	Parametr ta reprezentuje
209	10. od góry	a parametr pname wartość GL_TEXTURE_ENV, która	a parametr pname wartość GL_TEXTURE_ENV, która
209	10,11	GL_TEXTURE_ENV, a parametr pname wartość GL_TEXTURE_ENV	GL_TEXTURE_ENV, a parametr pname wartość GL_TEXTURE_ENV
213	4 od dołu	glBuild2DMipmaps	gluBuild2DMipmaps
218	5	for(xldx = 35; xldx >=0; xldx--)	for(xldx = 35; xldx >=0; xldx--)
227	9-6 od dołu	<pre> #define MAP_X32 #define MAP_Z32 #define MAP_SCALE20.0f #define PI3.14159 </pre>	<pre> #define MAP_X32 #define MAP_Z32 #define MAP_SCALE20.0f #define PI 3.14159 </pre>
242	6 od dołu	PI 3.14195	PI 3.14159

259	4 od dołu	texture_t *LoadTextureFile(char *filename)	texture_ *LoadLightMa *filenam
269	6 od dołu	istnieją efektywniejsze polecenia obsługi tekstu, które omówione zostaną wkrótce).	istnieją efektyw polecenia ob tekstur, kt omówione zo wkrótce
273	6 od dołu	glNewList(g_cube, GLCOMPILE);	glNewList(g_ GLCOMPILE
281	4 od dołu	GLuint g_indexArray[MAP_X*MAP_Z*6]	GLuint g_indexArray[MAP_X*MAP_Z*6]
286	6 od dołu	glBaseList()	glListBas
289	10	glColor2f()	glColor3
322	13	glRotatef(objectAngle, 1.0f, 1.0f, 0.0f);	glRotatef(obje 1.0f, 0.0f, 0
337	3 od dołu	funkcja x	funkcja f
526	4 od dołu	$A*x + B*y + C*z - D =$	$A*x + B*y + C = 0$

Poniżej znajduje się lista błędów znalezionych przez czytelników, ale jeszcze nie potwierdzonych przez Redakcję:

Strona	Linia	Jest	Powinno
162	18 od góry	glLightfv(GL_LIGHT0, GL_DIFFUSE, ambientLight);	glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight);