

## GradientTab.java - FogTab.java

```

1/*****
2 * Copyright (c) 2000, 2003 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.swt.opengl.examples;
12
13
14import org.eclipse.opengl.*;
15import org.eclipse.swt.*;
16import org.eclipse.swt.layout.*;
17import org.eclipse.swt.widgets.*;
18
19class GradientTab extends OpenGLTab {
20    private float xPos = 0.0f, yPos = 0.0f, zPos = -7.0f;
21    private float xRot = 180.0f, yRot = 180.0f, zRot = 180.0f;
22    private int currentSelection = 1;
23    private final static float[] BEZIER_COLORS = {
24        0.0f, 1.0f, 0.0f, 0.0f, 0.3f, 0.6f, 0.1f, 0.0f,
25        0.8f, 0.2f, 0.0f, 0.0f, 1.0f, 0.0f, 0.0f, 0.0f,
26        0.0f, 0.0f, 1.0f, 0.0f, 0.8f, 0.8f, 0.8f, 0.0f
27    };
28    private final static float[] BEZIER_POINTS = {
29        -1.5f, -1.5f, 4.0f, -0.5f, -1.5f, 2.0f, 0.5f, -1.5f,
30        -1.0f, 1.5f, -1.5f, 2.0f, -1.5f, -0.5f, 1.0f, -0.5f,
31        -0.5f, 3.0f, 0.5f, -0.5f, 0.0f, 1.5f, -0.5f, -1.0f,
32        -1.5f, 0.5f, 4.0f, -0.5f, 0.5f, 0.0f, 0.5f, 0.5f,
33        3.0f, 1.5f, 0.5f, 4.0f, -1.5f, 1.5f, -2.0f, -0.5f,
34        1.5f, -2.0f, 0.5f, 1.5f, 0.0f, 1.5f, 1.5f, -1.0f,
35    };
36    private final static String[] OBJECT_NAMES = { "Bezier", "Square" };
37
38    /**
39     * @see OpenGLTab#createControls(Composite)
40     */
41    void createControls(Composite composite) {
42        Group movementGroup = new Group(composite, SWT.NONE);
43        movementGroup.setText("Translation");
44        movementGroup.setLayout(new GridLayout(2, false));
45
46        new Label(movementGroup, SWT.NONE).setText("X:");
47        final Slider xMove = new Slider(movementGroup, SWT.NONE);
48        xMove.setIncrement(1);
49        xMove.setMaximum(12);
50        xMove.setMinimum(0);
51        xMove.setThumb(2);
52        xMove.setPageIncrement(2);
53        xMove.setSelection(5);
54        xMove.addListener(SWT.Selection, new Listener() {
55            public void handleEvent(Event e) {
56                xPos = xMove.getSelection() - 5;

```

```

1/*****
2 * Copyright (c) 2000, 2003 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.swt.opengl.examples;
12
13
14import org.eclipse.opengl.*;
15import org.eclipse.swt.*;
16import org.eclipse.swt.layout.*;
17import org.eclipse.swt.widgets.*;
18
19class FogTab extends OpenGLTab {
20
21    private float rotY = 0.0f;
22    private float yPos = 0.0f, xPos = 0.0f, zPos = -15.0f;
23    private int cubeListIndexBase;
24    private final static int[] FOG_TYPES = { GL.GL_LINEAR, GL.GL_EXP, GL.GL_EXP2 };
25    private final static String[] FOG_NAMES = { "LINEAR", "GL_EXP", "GL_EXP2" };
26    private final static int SLEEP_LENGTH = 0;
27
28    /**
29     * @see OpenGLTab#createControls(Composite)
30     */
31    void createControls(Composite composite) {
32        Group movementGroup = new Group(composite, SWT.NONE);
33        movementGroup.setText("Translation");
34        movementGroup.setLayout(new GridLayout(2, false));
35
36        new Label(movementGroup, SWT.NONE).setText("X:");
37        final Slider xMove = new Slider(movementGroup, SWT.NONE);
38        xMove.setIncrement(1);
39        xMove.setMaximum(12);
40        xMove.setMinimum(0);
41        xMove.setThumb(2);
42        xMove.setPageIncrement(2);
43        xMove.setSelection(5);
44        xMove.addListener(SWT.Selection, new Listener() {
45            public void handleEvent(Event e) {
46                xPos = xMove.getSelection() - 5;

```

## GradientTab. java – FogTab. java

```

57     }
58     });
59
60     new Label(movementGroup, SWT.NONE).setText("Y:");
61     final Slider yMove = new Slider(movementGroup, SWT.NONE);
62     yMove.setIncrement(1);
63     yMove.setMaximum(12);
64     yMove.setMinimum(0);
65     yMove.setThumb(2);
66     yMove.setPageIncrement(2);
67     yMove.setSelection(5);
68     yMove.addListener(SWT.Selection, new Listener() {
69         public void handleEvent(Event e) {
70             yPos = yMove.getSelection() - 5;
71         }
72     });
73
74     new Label(movementGroup, SWT.NONE).setText("Z:");
75     final Slider zMove = new Slider(movementGroup, SWT.NONE);
76     zMove.setIncrement(1);
77     zMove.setMaximum(12);
78     zMove.setMinimum(0);
79     zMove.setThumb(2);
80     zMove.setPageIncrement(2);
81     zMove.setSelection(5);
82     zMove.addListener(SWT.Selection, new Listener() {
83         public void handleEvent(Event e) {
84             zPos = zMove.getSelection() - 12;
85         }
86     });
87
88     Group rotationGroup = new Group(composite, SWT.NONE);
89     rotationGroup.setText("Rotation");
90     rotationGroup.setLayout(new GridLayout(2, false));
91
92     new Label(rotationGroup, SWT.NONE).setText("X:");
93     final Slider xRotation = new Slider(rotationGroup, SWT.NONE);
94     xRotation.setIncrement(10);
95     xRotation.setMaximum(362);
96     xRotation.setMinimum(0);
97     xRotation.setThumb(2);
98     xRotation.setPageIncrement(20);
99     xRotation.setSelection(180);
100    xRotation.addListener(SWT.Selection, new Listener() {
101        public void handleEvent(Event e) {
102            xRot = xRotation.getSelection();
103        }
104    });
105
106    new Label(rotationGroup, SWT.NONE).setText("Y:");
107    final Slider yRotation = new Slider(rotationGroup, SWT.NONE);
108    yRotation.setIncrement(10);
109    yRotation.setMaximum(362);
110    yRotation.setMinimum(0);
111    yRotation.setThumb(2);
112    yRotation.setPageIncrement(20);

```

```

47     }
48     });
49
50     new Label(movementGroup, SWT.NONE).setText("Y:");
51     final Slider yMove = new Slider(movementGroup, SWT.NONE);
52     yMove.setIncrement(1);
53     yMove.setMaximum(12);
54     yMove.setMinimum(0);
55     yMove.setThumb(2);
56     yMove.setPageIncrement(2);
57     yMove.setSelection(5);
58     yMove.addListener(SWT.Selection, new Listener() {
59         public void handleEvent(Event e) {
60             yPos = yMove.getSelection() - 5;
61         }
62     });
63
64     new Label(movementGroup, SWT.NONE).setText("Z:");
65     final Slider zMove = new Slider(movementGroup, SWT.NONE);
66     zMove.setIncrement(1);
67     zMove.setMaximum(24);
68     zMove.setMinimum(0);
69     zMove.setThumb(4);
70     zMove.setPageIncrement(2);
71     zMove.setSelection(10);
72     zMove.addListener(SWT.Selection, new Listener() {
73         public void handleEvent(Event e) {
74             zPos = zMove.getSelection() - 25;

```

## GradientTab. java – FogTab. java

<pre> 113 yRotation.setSelection(180); 114 yRotation.addListener(SWT.Selection, new Listener() { 115     public void handleEvent(Event e) { 116         yRot = yRotation.getSelection(); 117     } 118 }); 119 120 new Label(rotationGroup, SWT.NONE).setText("Z:"); 121 final Slider zRotation = new Slider(rotationGroup, SWT.NONE); 122 zRotation.setIncrement(10); 123 zRotation.setMaximum(362); 124 zRotation.setMinimum(0); 125 zRotation.setThumb(2); 126 zRotation.setPageIncrement(20); 127 zRotation.setSelection(180); 128 zRotation.addListener(SWT.Selection, new Listener() { 129     public void handleEvent(Event e) { 130         zRot = zRotation.getSelection(); 131     } 132 }); 133 134 Composite objectGroup = new Composite(composite, SWT.NONE); 135 GridLayout layout = new GridLayout(2, false); 136 layout.marginHeight = 0; 137 layout.marginWidth = 0; 138 objectGroup.setLayout(layout); 139 objectGroup.setLayoutData(new GridData(GridData.HORIZONTAL_ALIGN_FILL)); 140 141 new Label(objectGroup, SWT.NONE).setText("Object:"); 142 final Combo objectCombo = new Combo(objectGroup, SWT.READ_ONLY); 143 GridData data = new GridData(GridData.HORIZONTAL_ALIGN_FILL); 144 data.grabExcessHorizontalSpace = true; 145 objectCombo.setLayoutData(data); 146 objectCombo.setItems(OBJECT_NAMES); 147 objectCombo.select(0); 148 objectCombo.addListener(SWT.Selection, new Listener() { 149     public void handleEvent(Event e) { 150         currentSelection = objectCombo.getSelectionIndex() + 1; </pre>	<pre> 75     } 76 }); 77 78 Composite fogTypesGroup = new Composite(composite, SWT.NONE); 79 80 GridLayout layout = new GridLayout(2, false); 81 layout.marginHeight = 0; 82 layout.marginWidth = 0; 83 fogTypesGroup.setLayout(layout); 84 fogTypesGroup.setLayoutData(new GridData(GridData.HORIZONTAL_ALIGN_FILL)); 85 86 new Label(fogTypesGroup, SWT.NONE).setText("Fog Types:"); 87 final Combo fogTypeCombo = new Combo(fogTypesGroup, SWT.READ_ONLY); 88 GridData data = new GridData(GridData.HORIZONTAL_ALIGN_FILL); 89 data.grabExcessHorizontalSpace = true; 90 fogTypeCombo.setLayoutData(data); 91 fogTypeCombo.setItems(FOG_NAMES); 92 fogTypeCombo.select(0); 93 94 new Label(composite, SWT.NONE).setText("Fog Density:"); 95 final Slider fogDensitySlider = new Slider(composite, SWT.NONE); 96 fogDensitySlider.setIncrement(1); 97 fogDensitySlider.setMaximum(32); 98 fogDensitySlider.setMinimum(0); 99 fogDensitySlider.setThumb(2); 100 fogDensitySlider.setPageIncrement(5); 101 fogDensitySlider.setSelection(0); 102 fogDensitySlider.setEnabled(false); 103 fogDensitySlider.addListener(SWT.Selection, new Listener() { 104     public void handleEvent(Event e) { 105         float fogDensity = ((float) fogDensitySlider.getSelection()) / 100; 106         GL.glFogf(GL.GL_FOG_DENSITY, fogDensity); 107     } 108 }); 109 fogTypeCombo.addListener(SWT.Selection, new Listener() { 110     public void handleEvent(Event e) { 111         int currentSelection = fogTypeCombo.getSelectionIndex(); 112         // fog type GL.GL_LINEAR does not utilize fogDensity, but the 113         other fog types do </pre>
--	---

## GradientTab.java - FogTab.java

```

151     });
152 }
153 }
154

```

```

112         fogDensitySlider.setEnabled(currentSelection != 0);
113         GL.glFogf(GL.GL_FOG_MODE, FOG_TYPES[currentSelection]);
114     }
115     });
116 }
117
118 /**
119  * Creates a cube at 0,0 in the viewport.
120  */
121 void createCube() {
122     GL.glNewList(cubeListIndexBase, GL.GL_COMPILE);
123     GL.glBegin(GL.GL_QUADS);
124     // front
125     GL.glColor3f(0.0f, 1.0f, 0.0f);
126     GL.glVertex3f(-0.5f, -0.5f, 0.5f); // bottom left
127     GL.glColor3f(0.0f, 0.0f, 1.0f);
128     GL.glVertex3f(0.5f, -0.5f, 0.5f); // bottom right
129     GL.glColor3f(1.0f, 1.0f, 0.0f);
130     GL.glVertex3f(0.5f, 0.5f, 0.5f); // top right
131     GL.glColor3f(1.0f, 0.0f, 0.0f);
132     GL.glVertex3f(-0.5f, 0.5f, 0.5f); // top left
133     // back
134     GL.glColor3f(0.0f, 0.0f, 1.0f);
135     GL.glVertex3f(-0.5f, -0.5f, -0.5f); // bottom left
136     GL.glColor3f(0.0f, 1.0f, 0.0f);
137     GL.glVertex3f(0.5f, -0.5f, -0.5f); // bottom right
138     GL.glColor3f(1.0f, 0.0f, 0.0f);
139     GL.glVertex3f(0.5f, 0.5f, -0.5f); // top right
140     GL.glColor3f(1.0f, 1.0f, 0.0f);
141     GL.glVertex3f(-0.5f, 0.5f, -0.5f); // top left
142     // left
143     GL.glColor3f(0.0f, 0.0f, 1.0f);
144     GL.glVertex3f(-0.5f, -0.5f, -0.5f); // bottom left
145     GL.glColor3f(0.0f, 1.0f, 0.0f);
146     GL.glVertex3f(-0.5f, -0.5f, 0.5f); // bottom right
147     GL.glColor3f(1.0f, 0.0f, 0.0f);
148     GL.glVertex3f(-0.5f, 0.5f, 0.5f); // top right
149     GL.glColor3f(1.0f, 1.0f, 0.0f);
150     GL.glVertex3f(-0.5f, 0.5f, -0.5f); // top left
151     // right
152     GL.glColor3f(0.0f, 0.0f, 1.0f);
153     GL.glVertex3f(0.5f, -0.5f, 0.5f); // bottom left
154     GL.glColor3f(0.0f, 1.0f, 0.0f);
155     GL.glVertex3f(0.5f, -0.5f, -0.5f); // bottom right
156     GL.glColor3f(1.0f, 0.0f, 0.0f);
157     GL.glVertex3f(0.5f, 0.5f, -0.5f); // top right
158     GL.glColor3f(1.0f, 1.0f, 0.0f);
159     GL.glVertex3f(0.5f, 0.5f, 0.5f); // top left
160     // top
161     GL.glColor3f(1.0f, 0.0f, 0.0f);
162     GL.glVertex3f(0.5f, 0.5f, -0.5f);
163     GL.glColor3f(1.0f, 1.0f, 0.0f);
164     GL.glVertex3f(-0.5f, 0.5f, -0.5f);
165     GL.glColor3f(1.0f, 0.0f, 0.0f);
166     GL.glVertex3f(-0.5f, 0.5f, 0.5f);
167     GL.glColor3f(1.0f, 1.0f, 0.0f);

```

## GradientTab.java – FogTab.java

```

155  /**
156   * @see OpenGLTab#dispose()
157   */
158   void dispose() {
159       super.dispose();
160       GL.glDeleteLists(1, 2);
161   }
162
163   /**
164   * @see OpenGLTab#getTabText()
165   */
166   String getTabText() {
167       return "Gradients";
168   }
169
170   /**
171   * @see OpenGLTab#init()
172   */
173   void init() {
174       GL.glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
175       GL.glHint(GL.GL_LINE_SMOOTH_HINT, GL.GL_NICEST);
176       GL.glBlendFunc(GL.GL_SRC_ALPHA, GL.GL_ONE_MINUS_SRC_ALPHA);
177       GL.glPolygonMode(GL.GL_FRONT_AND_BACK, GL.GL_FILL);
178       GL.glMap2f(GL.GL_MAP2_VERTEX_3, 0, 1, 3, 4, 0, 1, 12, 4, BEZIER_POINTS);
179       GL.glMap2f(GL.GL_MAP2_COLOR_4, 0, 1, 4, 1, 0, 1, 4, 6, BEZIER_COLORS);
180       GL.glMapGrid2f(20, 0.0f, 1.0f, 20, 0.0f, 1.0f);
181       GL.glShadeModel(GL.GL_SMOOTH);
182       GL.glEnable(GL.GL_LINE_SMOOTH);
183       GL.glEnable(GL.GL_LINE_STIPPLE);
184       GL.glEnable(GL.GL_BLEND);
185       GL.glEnable(GL.GL_MAP2_COLOR_4);
186       GL.glEnable(GL.GL_MAP2_VERTEX_3);
187       GL.glEnable(GL.GL_DEPTH_TEST);
188
189       // create display lists
190
191       GL.glVertex3f(0.5f, 0.5f, 0.5f);
192       // bottom
193       GL.glColor3f(0.0f, 0.0f, 1.0f);
194       GL.glVertex3f(0.5f, -0.5f, 0.5f);
195       GL.glColor3f(0.0f, 1.0f, 0.0f);
196       GL.glVertex3f(-0.5f, -0.5f, 0.5f);
197       GL.glColor3f(0.0f, 0.0f, 1.0f);
198       GL.glVertex3f(-0.5f, -0.5f, -0.5f);
199       GL.glColor3f(0.0f, 1.0f, 0.0f);
200       GL.glVertex3f(0.5f, -0.5f, -0.5f);
201       GL.glEnd();
202       GL.glEndList();
203   }
204
205   /**
206   * @see OpenGLTab#dispose()
207   */
208   void dispose() {
209       super.dispose();
210       GL.glDeleteLists(cubeListIndexBase, 1);
211   }
212
213   /**
214   * @see OpenGLTab#getSleepLength()
215   */
216   int getSleepLength() {
217       return SLEEP_LENGTH;
218   }
219
220   /**
221   * @see OpenGLTab#getTabText()
222   */
223   String getTabText() {
224       return "Fog";
225   }
226
227   /**
228   * @see OpenGLTab#init()
229   */
230   void init() {
231       GL.glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
232       // fog color should be the same as the clear color
233       // to look appropriate
234       float[] fogColor = { 1.0f, 1.0f, 1.0f, 1.0f };
235       GL.glFogfv(GL.GL_FOG_COLOR, fogColor);
236       GL.glHint(GL.GL_FOG_HINT, GL.GL_DONT_CARE);
237       GL.glFogf(GL.GL_FOG_START, 0);
238       GL.glFogf(GL.GL_FOG_DENSITY, 0.0f);
239       // set the end of the start distance; anything > 15
240       // units from the camera will be covered in fog
241       GL.glFogf(GL.GL_FOG_END, 15);
242       GL.glFogf(GL.GL_FOG_MODE, FOG_TYPES[0]);
243       GL.glEnable(GL.GL_FOG);
244       GL.glEnable(GL.GL_DEPTH_TEST);
245
246       cubeListIndexBase = GL.glGenLists(1);

```

## GradientTab.java – FogTab.java

```

190 GL.glNewList(1, GL.GL_COMPILE);
191 GL.glEvalMesh2(GL.GL_FILL, 0, 20, 0, 20);
192 GL.glEndList();
193 GL.glNewList(2, GL.GL_COMPILE);
194 GL.glBegin(GL.GL_TRIANGLE_FAN);
195 GL.glColor3f(0.0f, 1.0f, 0.0f);
196 GL.glVertex3f(0.0f, 0.0f, 0.0f);
197 GL.glColor3f(1.0f, 0.0f, 0.0f);
198 GL.glVertex3f(0.0f, 2.0f, 0.0f);
199 GL.glColor3f(0.0f, 1.0f, 0.0f);
200 GL.glVertex3f(-2.0f, 2.0f, 0.0f);
201 GL.glColor3f(0.0f, 0.0f, 1.0f);
202 GL.glVertex3f(-2.0f, 0.0f, 0.0f);
203 GL.glColor3f(0.0f, 1.0f, 0.0f);
204 GL.glVertex3f(-2.0f, -2.0f, 0.0f);
205 GL.glColor3f(1.0f, 0.0f, 0.0f);
206 GL.glVertex3f(0.0f, -2.0f, 0.0f);
207 GL.glColor3f(0.0f, 1.0f, 0.0f);
208 GL.glVertex3f(2.0f, -2.0f, 0.0f);
209 GL.glColor3f(0.0f, 0.0f, 1.0f);
210 GL.glVertex3f(2.0f, 0.0f, 0.0f);
211 GL.glColor3f(0.0f, 1.0f, 0.0f);
212 GL.glVertex3f(2.0f, 2.0f, 0.0f);
213 GL.glColor3f(1.0f, 0.0f, 0.0f);
214 GL.glVertex3f(0.0f, 2.0f, 0.0f);
215 GL.glEnd();
216 GL.glEndList();

```

```

217 }
218
219 /**
220  * @see OpenGLTab#renderScene()
221  */
222 void renderScene() {
223     GL.glClear(GL.GL_COLOR_BUFFER_BIT);
224     GL.glLoadIdentity();
225     GL.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
226
227     GL.glTranslatef(xPos, yPos, zPos);
228     GL.glRotatef(xRot, 1.0f, 0.0f, 0.0f);
229     GL.glRotatef(yRot, 0.0f, 1.0f, 0.0f);
230     GL.glRotatef(zRot, 0.0f, 0.0f, 1.0f);
231     GL.glColor3f(1.0f, 0.0f, 0.0f);
232     GL.glCallList(currentSelection);

```

```

223 createCube();

```

```

224 }
225
226 /**
227  * @see OpenGLTab#renderScene()
228  */
229 void renderScene() {

```

```

230     GL.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
231     GL.glLoadIdentity();
232     GL.glTranslatef(xPos, yPos, zPos);
233     GL.glRotatef(rotY, 0.0f, 1.0f, 0.0f);
234
235     GL.glCallList(cubeListIndexBase);
236
237     GL.glPushMatrix();
238     GL.glTranslatef(3, 0, -3);
239     GL.glCallList(cubeListIndexBase);
240     GL.glPopMatrix();
241
242     GL.glPushMatrix();
243     GL.glTranslatef(-3, 0, -3);
244     GL.glCallList(cubeListIndexBase);
245     GL.glPopMatrix();
246
247     GL.glPushMatrix();
248     GL.glTranslatef(0, 0, 4);
249     GL.glCallList(cubeListIndexBase);
250     GL.glPopMatrix();

```

GradientTab. java - FogTab. java

```
232 }  
233 }  
234  
251  
252     rotY += 0.6f;  
253 }  
254 }  
255
```