Ovaldy | 161111062

1. #include "windows.h"
2. #include "GL/glut.h"
3. #include "stdio.h"
4. #include "math.h"
5. **float** view\_rotx = 0.0f, view\_roty = 180.0f;
6. **float** R\_Z=0.0f, R\_X=0.0f, R\_Y=0.0f;
7. **float** T\_Z=-4.0f, T\_X=0.0f, T\_Y=-0.0f;
9. **int** oldMouseX, oldMouseY;
11. **void** initGL(){
12. glShadeModel(GL\_FLAT);
14. **float** ambient[] = {1.0f,1.0f,1.0f,1.0f};
15. **float** diffuse[] = {1.0f,1.0f,1.0f,1.0f};
16. **float** specular[] = {0.2f,1.0f,0.2f,1.0f};
17. **float** position[] = {20.0f,30.0f,20.0f,0.0f};
19. glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient);
20. glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse);
21. glLightfv(GL\_LIGHT0, GL\_POSITION, position);
22. glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);
24. **float** mambient[] ={0.1745f, 0.01175f, 0.01175f, 0.55f};
25. **float** mdiffuse[] ={0.61424f, 0.04136f, 0.04136f, 0.55f };
26. **float** mspecular[] ={0.727811f, 0.626959f, 0.626959f, 0.55f };
27. **float** mshine =76.8f;
29. glMaterialfv(GL\_FRONT,GL\_AMBIENT,mambient);
30. glMaterialfv(GL\_FRONT,GL\_DIFFUSE,mdiffuse);
31. glMaterialfv(GL\_FRONT,GL\_SPECULAR,mspecular);
32. glMaterialf (GL\_FRONT,GL\_SHININESS,mshine);
34. glEnable(GL\_LIGHTING);
35. glEnable(GL\_LIGHT0);
36. glEnable(GL\_DEPTH\_TEST);
37. glEnable(GL\_NORMALIZE);
38. }
40. **void** timer(**int** value){
41. glutPostRedisplay();
42. glutTimerFunc(15, timer, 0);
43. }
45. **void** mouseControl(**int** button, **int** state, **int** x, **int** y){
46. oldMouseX = x;
47. oldMouseY = y;
48. }
50. **void** mouseMotion(**int** x, **int** y){
51. **int** getX = x;
52. **int** getY = y;
53. **float** thetaY = 360.0f\*(getX - oldMouseX)/640;
54. **float** thetaX = 360.0f\*(getY - oldMouseY)/480;
55. oldMouseX = getX;
56. oldMouseY = getY;
57. view\_rotx += thetaX;
58. view\_roty += thetaY;
59. }
60. **void** reshape(GLsizei width, GLsizei height){
61. **if** (height == 0) height = 1;
62. GLfloat aspect = (GLfloat)width / (GLfloat)height;
63. glViewport(30, 6, width, height);
64. glMatrixMode(GL\_PROJECTION);
65. glLoadIdentity();
66. gluPerspective(45.0f, aspect, 1.0f, 20.0f);
67. glMatrixMode(GL\_MODELVIEW);
68. glLoadIdentity();
69. }
71. **void** tube(){
72. **float** BODY\_LENGTH=2.0f;
73. **float** BODY\_RADIUS=1.0f;
74. **int** SLICES=30;
75. **int** STACKS=30;
76. GLUquadric \*q = gluNewQuadric();
77. gluCylinder(q, BODY\_RADIUS, BODY\_RADIUS, BODY\_LENGTH, SLICES, STACKS);
78. gluDisk(q, 0.0f, BODY\_RADIUS, SLICES, STACKS); //lingkaran untuk tutup atas
79. glTranslatef(0.0f, 0.0f, BODY\_LENGTH);
80. gluDisk(q, 0.0f, BODY\_RADIUS, SLICES, STACKS); //lingkaran untuk tutup bawah
81. }
83. **void** drawHemisphere(**int** scaley, **int** scalex, GLfloat r) {
85. **int** i, j;
86. GLfloat v[scalex\*scaley][3];
88. **for** (i=0; i<scalex; ++i) {
89. **for** (j=0; j<scaley; ++j) {
90. v[i\*scaley+j][0]=r\*cos(j\*2\*M\_PI/scaley)\*cos(i\*M\_PI/(2\*scalex));
91. v[i\*scaley+j][1]=r\*sin(i\*M\_PI/(2\*scalex));
92. v[i\*scaley+j][2]=r\*sin(j\*2\*M\_PI/scaley)\*cos(i\*M\_PI/(2\*scalex));
93. }
94. }
96. glBegin(GL\_QUADS);
97. **for** (i=0; i<scalex-1; ++i) {
98. **for** (j=0; j<scaley; ++j) {
99. glVertex3fv(v[i\*scaley+j]);
100. glVertex3fv(v[i\*scaley+(j+1)%scaley]);
101. glVertex3fv(v[(i+1)\*scaley+(j+1)%scaley]);
102. glVertex3fv(v[(i+1)\*scaley+j]);
103. }
104. }
105. glEnd();
106. }

109. **void** display\_pen(){
110. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);
111. glLoadIdentity();
112. glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
113. glTranslatef(0,0.0f, -10.0f);
114. glRotatef(view\_rotx, 1.0f, 0.0f, 0.0f);
115. glRotatef(view\_roty, 0.0f, 1.0f, 0.0f);
116. glRotatef(45, 0.0f, 0.0f, 1.0f);
117. glRotatef(90,0.0f,1.0f,0.0f);
118. tube();
119. glRotatef(90,1.0f,0.0f,0.0f);
120. drawHemisphere(30, 30, 1.0f);
121. glFlush();
122. glutSwapBuffers();
123. }
125. **void** display\_hemisphere(){
126. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);
127. glLoadIdentity();
128. glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
129. glTranslatef(0,0.0f, -10.0f);
130. glRotatef(view\_rotx, 1.0f, 0.0f, 0.0f);
131. glRotatef(view\_roty, 0.0f, 1.0f, 0.0f);
132. drawHemisphere(30,30,1.0f);
133. glFlush();
134. glutSwapBuffers();
135. }
137. **int** main(**int** argc, **char** \*\*argv){
138. glutInit(&argc, argv);
140. glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH);
141. glutInitWindowSize(480, 360);
142. glutInitWindowPosition(50, 50);
143. glutCreateWindow("hemisphere");
144. glutDisplayFunc(display\_hemisphere);
145. glutReshapeFunc(reshape);
146. initGL();
147. glutMouseFunc(mouseControl);
148. glutMotionFunc(mouseMotion);
149. glutTimerFunc(0, timer, 0);
151. glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH);
152. glutInitWindowSize(480, 360);
153. glutInitWindowPosition(50, 50);
154. glutCreateWindow("pen");
155. glutDisplayFunc(display\_pen);
156. glutReshapeFunc(reshape);
157. initGL();
158. glutMouseFunc(mouseControl);
159. glutMotionFunc(mouseMotion);
160. glutTimerFunc(0, timer, 0);
161. glutMainLoop();
162. **return** 0;
163. }

