

Modul 7

Input dan Animasi

I. Tugas Pendahuluan

1. Apa yang dimaksud dengan callback function?
2. Apa yang dimaksud dengan komputasi interaktif?
3. Apa yang dimaksud dengan simulasi?
4. Berikan penjelasan secara singkat sejarah animasi komputer!

II. Pengantar

Modul ini membahas tentang penambahan kemampuan untuk menerima input bagi program OpenGL. Selain itu, modul ini juga membahas tentang pembuatan animasi dengan menggunakan OpenGL.

Input

Yang dimaksud sebagai input di sini adalah fasilitas program untuk menerima sinyal dari perangkat input (keyboard dan mouse) ketika program dijalankan. Dengan fasilitas ini, program dan user dapat berinteraksi secara langsung (real-time), tanpa perlu melakukan kompilasi ulang tiap kali user ingin mengubah tampilan program.

Di GLUT, mekanisme input dijalankan dalam konsep *callback function*. Di konsep ini, fungsi main memanggil fungsi input glut dan programmer harus mendefinisikan isi fungsi input tersebut.

Program 7.1 menunjukkan contoh menerima input dari keyboard untuk merotasi 2 garis.

```
static float rotAngle = 0.1;

void init(void)
{
    glClearColor(0.0,0.0, 0.2, 0.0);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (0.0, 1.0, 0.0);

    glPushMatrix();
    glRotatef(-rotAngle, 0.0, 0.0, 0.1);
    glBegin (GL_LINES);
    glVertex2f (-0.5, 0.5);
    glVertex2f (0.5, -0.5);
    glEnd ();
    glPopMatrix();

    glColor3f (0.0, 0.0, 1.0);
```

```

    glPushMatrix();
    glRotatef(rotAngle, 0.0, 0.0, 0.1);
    glBegin (GL_LINES);
    glVertex2f (0.5, 0.5);
    glVertex2f (-0.5, -0.5);
    glEnd ();
    glPopMatrix();

    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        gluOrtho2D (-1.0, 1.0, -1.0*(GLfloat)h/(GLfloat)w, 1.0*(GLfloat)h/(GLfloat)w);
    else
        gluOrtho2D (-1.0*(GLfloat)w/(GLfloat)h, 1.0*(GLfloat)w/(GLfloat)h, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'r':
        case 'R':
            rotAngle += 20.;
            if (rotAngle >= 360.) rotAngle = 0.;
            glutPostRedisplay();
            break;
        case 27:
            exit(0);
            break;
        default:
            break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutCreateWindow (argv[0]);
    init();
    glutReshapeFunc (reshape);
    glutKeyboardFunc (keyboard);
    glutDisplayFunc (display);
    glutMainLoop();
    return 0;
}

```

Program 7.1 Garis Silang berotasi oleh penekanan tombol keyboard

Pada program 7.1, tiap kali tombol ‘r’ atau ‘R’ ditekan, kedua garis akan berotasi terhadap titik pusatnya. Perhatikan bahwa untuk menerima input dari keyboard, diperlukan:

1. Perintah `glutKeyboardFunc (keyboard);`

Iniilah fungsi callback yang diletakkan di dalam `main()`. `Keyboard()` sendiri adalah fungsi tempat input diproses. Nama fungsi ini bisa diganti-ganti sesuai keinginan, misal: `key()`, `inputKeyboard()`, dll.

2. `void keyboard(unsigned char key, int x, int y)` adalah format baku fungsi yang dipanggil oleh fungsi callback. Variabel `key` ini berisi kode tombol keyboard yang ditekan oleh user.

Program 7.2 menunjukkan contoh menerima input dari keyboard untuk menggerakkan simulasi lengan robot.

```
static int shoulder = 0, elbow = 0;

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glTranslatef (-1.0, 0.0, 0.0);
    glRotatef ((GLfloat) shoulder, 0.0, 0.0, 1.0);
    glTranslatef (1.0, 0.0, 0.0); glPushMatrix();
    glScalef (2.0, 0.4, 1.0); glutWireCube (1.0);
    glPopMatrix();

    glTranslatef (1.0, 0.0, 0.0);
    glRotatef ((GLfloat) elbow, 0.0, 0.0, 1.0);
    glTranslatef (1.0, 0.0, 0.0); glPushMatrix();
    glScalef (2.0, 0.4, 1.0);
    glutWireCube (1.0);
    glPopMatrix();

    glPopMatrix();
    glutSwapBuffers();
}

void reshape (int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(65.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef (0.0, 0.0, -5.0);
}
```

```

}

void keyboard (unsigned char key, int x, int y)
{
    switch (key)
    {
        case 's':
            shoulder = (shoulder + 5) % 360;
            glutPostRedisplay();
            break;
        case 'S':
            shoulder = (shoulder - 5) % 360;
            glutPostRedisplay();
            break;
        case 'e':
            elbow = (elbow + 5) % 360;
            glutPostRedisplay();
            break;
        case 'E':
            elbow = (elbow - 5) % 360;
            glutPostRedisplay();
            break;
        case 27:
            exit(0);
            break;
        default:
            break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (700, 600);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

Program 7.2 Simulasi lengan robot

Program diatas akan menggerakkan shoulder/lengan atas jika tombol ‘s’ atau ‘S’ ditekan; serta akan menggerakkan elbow/siku jika tombol ‘e’ atau ‘E’ ditekan.

Program 7.3 menunjukkan contoh menerima inputan dari keyboard untuk menggerakkan simulasi planet

```

static int year = 0, day = 0;

void init(void)

```

```

{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 1.0, 1.0);
    glPushMatrix();
    glutWireSphere(1.0, 20, 16); /* gambar matahari */
    glRotatef ((GLfloat) year, 0.0, 1.0, 0.0);
    glTranslatef (2.0, 0.0, 0.0);
    glRotatef ((GLfloat) day, 0.0, 1.0, 0.0);
    glutWireSphere(0.2, 10, 8); /* gambar planet kecil */
    glPopMatrix();
    glutSwapBuffers();
}

void reshape (int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0); glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
}

void keyboard (unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'd':
            day = (day + 10) % 360;
            glutPostRedisplay();
            break;
        case 'D':
            day = (day - 10) % 360;
            glutPostRedisplay();
            break;
        case 'y':
            year = (year + 5) % 360;
            glutPostRedisplay();
            break;
        case 'Y':
            year = (year - 5) % 360;
            glutPostRedisplay();
            break;
        case 27:
            exit(0);
            break;
        default:
            break;
    }
}

```

```

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

Program 7.3 Simulasi planet

Program diatas akan menggerakkan planet berotasi terhadap sumbunya atas jika tombol ‘d’ atau ‘D’ ditekan; serta akan menggerakkan planet untuk berotasi terhadap matahari jika tombol ‘y’ atau ‘Y’ ditekan.

Animasi.

Animasi adalah “Illusion Of Motion” yang dibuat dari sekumpulan image statis yang ditampilkan secara berurutan sehingga seolah-olah gambar-gambar diskontinyu tadi menjadi terlihat kontinyu. Animasi berkembang dari teori-teori/fenomena-fenomena terkait penglihatan manusia: phi phenomena, beta movement, & persistence of vision. Intinya adalah bahwa mata manusia cenderung menangkap urutan gambar-gambar yang ditampilkan pada tenggang waktu minimal tertentu sebagai suatu pola pergerakan.

Pada OpenGL, animasi dapat dibuat dengan memanfaatkan proses transformasi pada obyek yang dilakukan secara terus-menerus/berulang-ulang. Program 7.4 memutar kotak yang ditrigger dan distop oleh penekanan tombol mouse.

```

static GLfloat spin = 0.0;

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glRotatef(spin, 0.0, 0.0, 1.0);
    glColor3f(1.0, 1.0, 1.0);
    glRectf(-25.0, -25.0, 25.0, 25.0);
    glPopMatrix();
    glutSwapBuffers();
}

void spinDisplay(void)
{
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
}

```

```

    glutPostRedisplay();
}

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void reshape(int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void mouse(int button, int state, int x, int y)
{
    switch (button)
    {
        case GLUT_LEFT_BUTTON:
            if (state == GLUT_DOWN)
                glutIdleFunc(spinDisplay);
            break;
        case GLUT_MIDDLE_BUTTON:
        case GLUT_RIGHT_BUTTON:
            if (state == GLUT_DOWN)
                glutIdleFunc(NULL);
            break;
        default:
            break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutMainLoop();
    return 0;
}

```

Program 7.4 Persegi 2 dimensi berputar oleh penekanan tombol mouse

Program 7.4 menggunakan double buffering untuk menampilkan animasinya. Mirip dengan input keyboard, program dengan kemampuan menerima input mouse memerlukan:

1. `glutMouseFunc(mouse);` sebagai fungsi callback-nya.
2. `void mouse(int button, int state, int x, int y)` yang berisi kode untuk memproses input dari mouse. Variabel `button` berfungsi untuk menyimpan informasi tombol mouse mana yang diklik. Sedangkan variabel `x` dan `y` berfungsi untuk menyimpan informasi posisi kursor pada layar.

III. Percobaan

1. Cobalah program-program diatas
2. Amati pada fungsi program inti
3. Amati pada fungsi masukan
4. Ubahlah program 7.1 supaya kedua garis yang muncul pada gambar menjadi bergerak searah jarum jam dengan sudut 90 derajat antara keduanya
5. Pada program 7.4 ubahlah program supaya kotak bergerak dan berhenti jika di tekan tombol keyboard “P” atau “p”
6. Pada program 7.3 coba ganti perintah `glutWireSphere(1.0, 40, 16);` dengan `glutWireCube(1.0)` dan `glutWireSphere(0.2, 10, 8);` dengan `glutWireCube(0.2);`
7. Pada program 7.3 buatlah garis lintasan planet yang mengelilingi matahari

IV. Tugas

1. Buatlah program untuk menampilkan gambar bintang segi 4 dengan warna yang dapat diubah dengan menggunakan tombol panah \uparrow dan \downarrow
2. Modifikasi program 7.4 supaya
 - a. ketika tombol mouse kiri ditekan, kotak berotasi terhadap sumbu z (rotasi yang sekarang)
 - b. ketika tombol mouse tengah ditekan, kotak berotasi terhadap sumbu y
 - c. ketika tombol mouse kanan ditekan, kotak berotasi terhadap sumbu x
 - d. start dan stop pergerakan dilakukan dengan menekan tombol ‘p’ atau ‘P’
3. Modifikasi program 7.4 dengan mengganti kotak yang diputar dengan checkerboard 8 x 8
4. Modifikasi program 7.4 supaya
 - a. Ketika ditekan tombol ‘u’, ukuran kotak mengecil menjadi 0.75 ukuran semula
 - b. Ketika ditekan tombol ‘U’, ukuran kotak membesar menjadi 1.5 ukuran semula
 - c. Ketika ditekan tombol ‘v’, kecepatan putaran kotak jadi melambat setengah kali lipat dari kecepatan semula

- d. Ketika ditekan tombol 'V', kecepatan putaran kotak jadi lebih cepat dua kali lipat dari kecepatan semula

V. Referensi

1. Edward Angel, "Interactive Computer Graphics Sixth Edition", Pearson, 2012, ch 2, p 98 – 106
2. F. S. Hill, Jr., Stephen M. Kelley, "Computer Graphics Using OpenGL Third Edition", Prentice Hall, 2007, ch 5, p 190 – 263