

Grafika komputerowa i komunikacja człowiek-komputer

OpenGL – interakcja z użytkownikiem

Autor: Luka Mitrović

Numer indeksu: 253907

Grupa: PT 16:25 TN

Prowadzący: dr inż. Jan Nikodem

Wstęp teoretyczny

- `gluPerspective(fovy, aspect, zNear, zFar)` – funkcja do tworzenia macierzy opisującą perspektywę. Parametr `fovy` jest kątem pola widzenia względem osi Y. Parametr `aspect` opisuje stosunek szerokości do wysokości pola widzenia względem osi X. Parametr `zNear` opisuje odległość od obserwatora do jego najbliższego punktu, `zFar` opisuje odległość od obserwatora do najdalszego jego punktu.
- `gluLookAt(eyeX, eyeY, eyeZ, centerX, centerY, centerZ, upX, upY, upZ)` - funkcja ustawia punkt spojrzenia na obraz. Parametry `eyeX`, `eyeY` oraz `eyeZ` opisują położenie obserwatora, parametry `centerX`, `centerY` oraz `centerZ` opisują położenie obiektu na który patrzy obserwator, a ostatnie 3 parametry opisują gdzie jest góra względem obserwatora.
- `glutMouseFunc(callback)` - funkcja odpowiadająca za zdarzenia myszy. Przekazywana funkcja `callback` opisuje obsługę poszczególnych zdarzeń myszy.
- `glutMotionFunc(callback)` - funkcja odpowiadająca za obsługę ruchu myszy. Przekazywana funkcja `callback` opisuje przetwarzanie przemieszczania się kursora.
- `glRotateF(angle, x,y,z)` - obraca obraz o zadany kąt względem punktu o podanych parametrach współrzędnych.

Realizacja zadania:

Zmienne `old_theta` oraz `old_phi` są aktualizowane tylko podczas wciśnięcia lewego przycisku myszki i wykorzystywane podczas wyświetlania jajka. Oddalanie jajka odbywa się poprzez zmianę promienia. Po tym jak zostanie wciśnięty przycisk zapisuje się aktualna pozycja kursora. Zmienna `status` przechowuje dane o tym który przycisk został wciśnięty. Funkcja `Mouse()` odpowiada za obsługę guzików myszy.

```
void Mouse(int btn, int state, int x, int y)
{
    if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        x_pos_old = x;
        y_pos_old = y;
        status = 1;
    }
    else if (btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
        zoom_pos_old = y;
        status = 2;
    }
    else status = 0;
}
```

Funkcja `Motion()` oblicza przemieszczenie kursora względem miejsca w którym ostatnio się znajdował podczas wciśnięcia przycisku.

```
void Motion(GLsizei x, GLsizei y)
{
    delta_x = x - x_pos_old;
    delta_y = y - y_pos_old;
    delta_zoom = y - zoom_pos_old;
    x_pos_old = x;
    y_pos_old = y;
    zoom_pos_old = y;
    glutPostRedisplay();
}
```

Obracanie obserwatorem:

Podczas obracania zmieniają się dane lokalizacji obserwatora , które są przechowywane w tablicy viewer. Obserwator porusza się po strefie R, położeniu punktów strefy jest opisane wzorami podanymi poniżej.

$$x_s(\Theta, \Phi) = R \cos(\Theta) \cos(\Phi)$$

$$y_s(\Theta, \Phi) = R \sin(\Phi)$$

$$0 \leq \Theta \leq 2\pi$$

$$0 \leq \Phi \leq 2\pi$$

$$z_s(\Theta, \Phi) = R \sin(\Theta) \cos(\Phi)$$

Żeby uzyskać efekt przybliżenia i oddalania zmienia się promień strefy w której porusza się obserwator.

```
void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    if (status == 1) {
        fi += delta_y * pix2angleY;
        thetta += delta_x * pix2angleX;
        if (thetta >= 360.0) { thetta = 0.0; }
        if (fi >= 360.0) { fi = 0.0; }
    }

    else if (status == 2) {
        radius += 0.1 * delta_zoom;
        if (radius >= 25.0) { radius = 25.0; }
        if (radius <= 10.0) { radius = 10.0; }
    }

    glRotatef(thetta, 1.0, 1.0, 1.0);

    viewer[0] = radius * cos(thetta) * cos(fi);
    viewer[1] = radius * sin(fi);
    viewer[2] = radius * sin(thetta) * cos(fi);
    gluLookAt(viewer[0], viewer[1], viewer[2], 0.0, 0.0, 0.0, 0.0,
cos(fi), 0.0);
    glTranslated(0.0, -4.5, 0.0);
    if(eggDraw)
    {
        Egg();
    }
    if(teapotDraw)
    {
        glutSolidTeapot(3.0);
    }
    glFlush();
    glutSwapBuffers();
}
```

Wnioski:

Podczas obracania czajnikiem obserwator obraca się w niepoprawny sposób, niestety ten problem nie udało się rozwiązać, również jajko podczas obracanie obiektu tworzy efekt oddalania się w stronę przeciwną od ruchu obserwatora, może to być spowodowane niepoprawnym wyliczaniem kąta obracania lub nieuwzględnienie niektórych parametrów podczas realizacji zadania.