



Interakcija čovek-računar

Laboratorijske vežbe – III termin

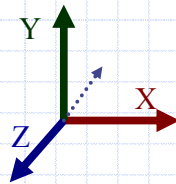
3D Interfejs i OpenGL

Zadaci

- ◆ Navigacija pomoću miša i tastature
- ◆ Izbor objekata u 3D sceni

Prva 3D scena

- ◆ Crtanje kocke
 - `glutSolidCube(1.0);`
- ◆ Zadavanje boje
 - `glColor3f(1.0, 0.0, 0.0);`
- ◆ Translacija
 - `glTranslatef(1.0, 0.0, -1.0);`
- ◆ Rotacija
 - `glRotatef(45.0, 0.0, 1.0, 0.0);`
- ◆ Čuvanje/vraćanje transformacija
 - `glPushMatrix() / glPopMatrix()`



Primer

```
void CGLRenderer::DrawScene()
{
    glClear(GL_COLOR_BUFFER_BIT |
           GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(0, 0, -3.0);

    // KOD ZA CRTANJE →

    glFlush();
}
```

```
glPushMatrix();

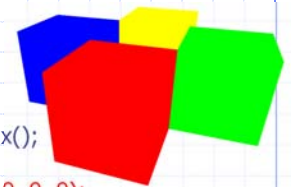
glColor3f(1.0, 0, 0);
glutSolidCube(1.0);

glTranslatef(1.0, 0.0, -1.0);
glColor3f(0, 1.0, 0);
glutSolidCube(1.0);

glTranslatef(-2.0, 0.0, 0.0);
glColor3f(0, 0, 1.0);
glutSolidCube(1.0);

glTranslatef(1.0, 0.0, -1.0);
glColor3f(1.0, 1.0, 0);
glutSolidCube(1.0);

glPopMatrix();
```



Navigacija

◆ Zumiranje

- skrol-tasterom
- tastaturom (+/-)

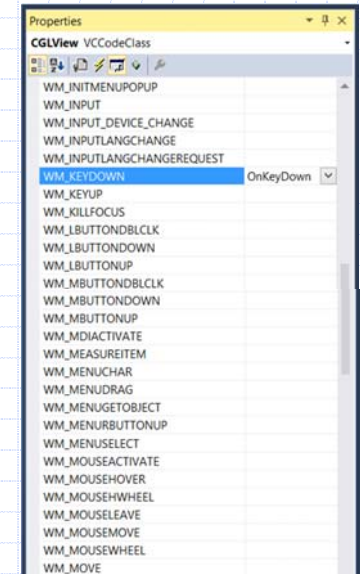
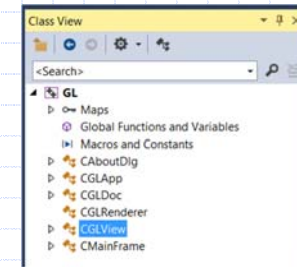
◆ Pomeranje prikaza

- mišem
- tastaturom (kursorski tasteri i WASD)

◆ Rotacija

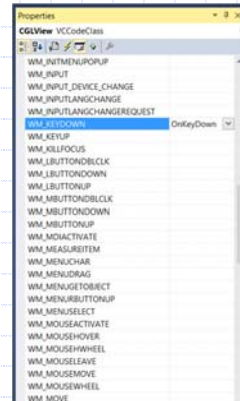
- mišem
- tastaturom (shift + kursorski tasteri)

Dodavanje rukovaoca



Windows poruke

- ◆ WM_KEYDOWN
- ◆ WM_KEYUP
- ◆ WM_LBUTTONDOWNBLCLK
- ◆ WM_LBUTTONDOWN
- ◆ WM_LBUTTONUP
- ◆ WM_MBUTTONDOWN.../ WM_RBUTTONDOWN...
- ◆ WM_MOUSEWHEEL



Zumiranje na skrol taster miša

```
BOOL CGLView::OnMouseWheel(UINT nFlags, short zDelta, CPoint pt)
{
    // TODO: Add your message handler code here and/or call default

    if (zDelta > 0.0)
        m_glRenderer.m_z += 0.2;

    if (zDelta < 0.0)
        m_glRenderer.m_z -= 0.2;

    Invalidate();

    return CView::OnMouseWheel(nFlags, zDelta, pt);
}
```

Promena u crtanju scene

```
void CGLRenderer::DrawScene()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(m_x, m_y, -3.0+m_z);

    glPushMatrix();

    ...
}
```

Zumiranje preko tastature

```
void CGLView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    if (nChar == VK_ADD)
        m_glRenderer.m_z += 0.2;

    if (nChar == VK_SUBTRACT)
        m_glRenderer.m_z -= 0.2;

    Invalidate();

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

Pomeranje preko tastature

```
void CGLView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    if (nChar == VK_RIGHT)
        m_glRenderer.m_x += 0.2;

    if (nChar == VK_LEFT)
        m_glRenderer.m_x -= 0.2;

    if (nChar == VK_UP)
        m_glRenderer.m_y += 0.2;

    if (nChar == VK_DOWN)
        m_glRenderer.m_y -= 0.2;

    Invalidate();

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

Rotacija preko tastature

```
void CGLView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    bool bShift = GetKeyState(VK_SHIFT) & 0x8000;

    if (nChar == VK_RIGHT)
        if (bShift)
            m_glRenderer.m_Ry += 5.0;
        else
            m_glRenderer.m_x += 0.2;
}
```

Promena u crtanju scene

```
void CGLRenderer::DrawScene()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(m_x, m_y, -3.0+m_z);

    glRotatef(m_Rx, 1.0, 0.0, 0.0);
    glRotatef(m_Ry, 0.0, 1.0, 0.0);

    glPushMatrix();

    ...
}
```

Izbor objekata u 3D sceni

Koraci:

- ◆ Specificiranje polja za smeštaj zapisa "pogodaka" (*hit records*) - **glSelectBuffer**
- ◆ Prelazak u mod za izbor – **glRenderMode(GL_SELECT)**
- ◆ Inicijalizacija steka imena – **glInitNames()**
- ◆ Definisanje pogleda, tj. prostora koji se koristi za selekciju
- ◆ Naizmenično vršiti iscrtavanje i izdavati komande za manipulaciju stekom imena
- ◆ Izaći iz moda za selekciju i obraditi podatke koje je vratila selekcija

Kreiranje polja za smeštanje imena i prelazak u mod za izbor

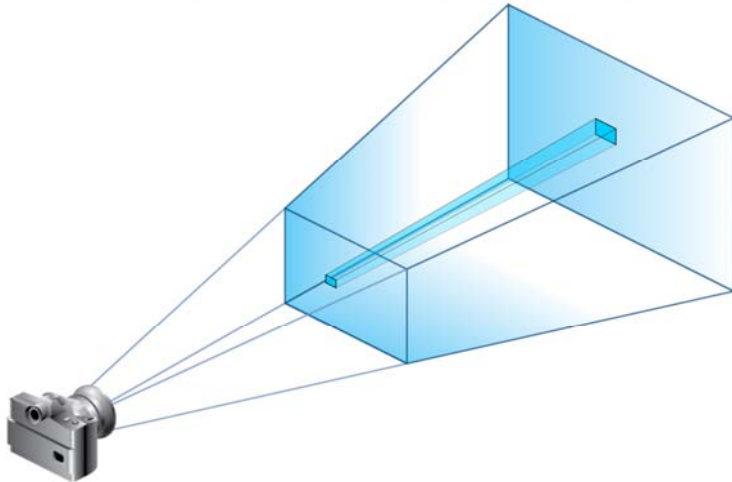
- ◆ **void glSelectBuffer(GLsizei size, GLuint *buffer);**
 - Specficira polje za smeštanje selektovanih podataka. buffer – pointer na bafer, size – veličina bafera
- ◆ **GLint glRenderMode(GLenum mode);**
 - Prevodi OpenGL u mod za:
 - ◆ iscrtavanje (GL_RENDER)
 - ◆ selekciju (GL_SELECT)
 - ◆ odziv (GL_FEEDBACK)

Primer:
GLuint selectBuf[512];
glSelectBuffer(512, selectBuf);
glRenderMode(GL_SELECT);

Definisanje pogleda/prostora za selekciju

- ◆ Selekcija se ostvaruje proverom koje primitive presecaju modifikovani prostor pogleda
- ◆ Pogled treba da ima iste parametre kao i pogled koji prikazuje scenu, ali se sužava, tako da projekcija tog prostora bude **par×par** piksela (nije preporučljivo da bude samo 1 piksel, zbog preciznosti izbora)
- ◆ Prostor treba da bude centriran u odnosu na položaj kursora

Definisanje pogleda/prostora za selekciju



Definisanje pogleda/prostora za selekciju

- ◆ **void gluPickMatrix(GLdouble x, GLdouble y, GLdouble width, GLdouble height, GLint viewport[4]);**
 - definiše region (prostor) za selekciju, tj. ograničava "iscrtavanje" samo na mali region oko kursora
 - x,y – centar regiona selekcije u ekranskim koordinatama
 - width, height – širina i visina regiona za izbor
 - viewport – tekući viewport (pogled)



Definisanje pogleda/prostora za selekciju - primer

Primer:

```
GLint viewport[4];
glMatrixMode(GL_PROJECTION);
glPushMatrix();
glLoadIdentity();

// Preuzimanje tekućeg viewport-a
glGetIntegerv(GL_VIEWPORT, viewport);
// Definisanje (suženog) prostora selekcije
gluPickMatrix(x, viewport[3]-y, 3, 3, viewport);
// Definisanje projekcije na isti način kao pri crtanju
gluPerspective(45.0, m_aspect, 0.1, 100.0);
glMatrixMode(GL_MODELVIEW);
```



Definisanje steka imena

- ◆ **void glInitNames();**
 - Inicijalizuje stek imena
- ◆ **void glPushName(GLuint name);**
 - Postavlja *name* na vrh steka imena
- ◆ **void glPopName();**
 - Uklanja ime iz steka imena
- ◆ **void glLoadName(GLuint name);**
 - Zamenjuje ime na vrhu steka
 - void glLoadName(GLuint name){
 glPopName();
 glPushName(name);
}

Manipulacija stekom i iscrtavanje

Primer:

```
glInitNames();
glPushName(0); // često se na vrh stavlja nešto
                // da bi bilo moguće sukcesivno
                // korišćenja f-je glLoadName()

...
glLoadName(1);
drawSomeObject();
glLoadName(2);
drawAnotherObject();
```

Komentar: Potrebno je postaviti ime u stek pre iscrtavanja objekta koga reprezentuje to ime. Moguće je i više objekata iscrtati nakon navođenja jednog imena. U tom slučaju ne može se razlikovati koji objekat je izabran.

Zapis selekcije

- ◆ Zapis koji se dobija nakon izvršene selekcije sadrži sledeća polja:
- broj imena u steku kada se desio "pogodak"
- minimalna i maksimalna Z-vrednost svih temena primitiva koje su presekle prostor pogleda od prethodno zabeleženog pogotka (to su vrednosti od 0 do 1 pomnožene sa $2^{32}-1$ i zaokružene na najbliži neoznačeni ceo broj)
- sadržaj steka imena u trenutku kada se "pogodak" desio, čitanih od dna steka prema vrhu

br_imena	minZ	maxZ	Ime ₁	...	Ime _{br_imena}
----------	------	------	------------------	-----	-------------------------

Zapis selekcije

- ◆ Nakon ulaska u mod selekcije, OpenGL upisuje od početka bafera zapise selekcije (vidi prethodni slajd) svaki put kada se desi "pogodak" (presek sa prostorom selekcije)
- ◆ Ako se premaši veličina bafera, samo se setuje fleg prekoračenja i prestaje sa upisom slogova
- ◆ Pri izlasku iz moda selekcije, glRenderMode() vraća broj zapisa u baferu (-1 ako je nastalo prekoračenje), briše stek imena i resetuje fleg prekoračenja

Izlazak iz moda selekcije i obrada

Primer:

```
int hits = glRenderMode(GL_RENDER);
unsigned int retVal = 0;
if(hits != 0)
    retVal = ProcessHits(hits,selectBuf);
```

Komentar: Svi selekcionni slogovi nalaze se u selectBuf-u i potrebno ih je obraditi. Za to se koristi "naša" funkcija ProcessHits.

Obrada selekcije

```
unsigned int CGLRenderer::ProcessHits(int hits, unsigned int buffer[])
{
    unsigned int i, j;
    unsigned int names, *ptr, minZ, *ptrNames, numberOfNames;
    ptr = (unsigned int *) buffer;
    minZ = 0xffffffff;
    for (i = 0; i < hits; i++) {
        names = *ptr;
        ptr++;
        if (*ptr < minZ) {
            numberOfNames = names;
            minZ = *ptr;
            ptrNames = ptr+2;
        }
        ptr += names+2;
    }
    ptr = ptrNames;
    return *ptr; // Vracam samo prvi iz liste
}
```

br_imena minZ maxZ Ime₁ ... Ime_{br_imena}

Šta je potrebno uraditi u vežbi

- ◆ Dodavanje imena u stek imena
- ◆ Implementacija funkcije ProcessHits()
- ◆ Poziv funkcije PickObject()

Zadatak

- ◆ Dopuniti projekat dostupan na sajtu mogućnošću:
 - zumiranje na skrol taster,
 - zumiranje preko tastature,
 - pomeranje preko tastature,
 - pomeranje pomoću miša,
 - rotaciju preko tastature,
 - rotaciju pomoću miša,
 - izbora klikom i ispisa poruke koja je kocka izabrana,
 - dopuniti scenu još jednom kockom, ali tako da bude deo neke druge kocke (logički, ne fizički) i omogućiti detekciju hijerarhije – npr. „Ljubičasta u okviru zelene kocke“.

Ispitni zadatak

U baferu selekcije nalaze se sledeće vrednosti:

3, 2247483647, 2247783647, 98, 56, 11, 2,
2147483647, 2147783647, 78, 12, 3, 2047483647,
2047783647, 43, 9, 7, 2, 2747483647, 2747783647, 2,
1, 2047483647, 2047483647, 2047483647

- ◆ Objasniti kako izgleda slog selekcije.
- ◆ Odrediti na koji je objekat kliknuto.
- ◆ Odrediti koliko je taj objekat daleko od posmatrača, ako su prednja i zadnja ravan odsecanja na rastojanjima 0.1 i 10.1, respektivno.

Ispitni zadatak

Na osnovu programskog koda 1, odrediti kako izgleda bafer selekcije, ako je prednja ravan odsecanja na rastojanju 2, a zadnja na 20 od posmatrača.

Funkcija **DrawBox(a, 0, 0, z)** crta kocku dužine stranice **a** sa centrom, na rastojanju **z** od posmatrača. Objasniti značenje pojedinih polja u slogu selekcije. Koji objekat je selektovan?

```
glPushName(302);  
DrawBox(4,0,0,8);  
DrawBox(2,0,0,5);  
glPushName(102);  
DrawBox(2,0,0,4);  
DrawBox(4,0,0,6);  
glLoadName(101);  
DrawBox(2,0,0,9);  
glPopName();  
glPushName(203);  
DrawBox(2,0,0,14);  
glPopName();
```

Kod 1.