

Costruzione di Interfacce

Lezione 15

Moebius 3

cignoni@isti.cnr.it
<http://vcg.isti.cnr.it/~cignoni>

update

- ❖ nelle versioni precedenti c'era un piccolo bug:
- ❖ in risposta ad un evento resize si faceva anche una `setvideomode` (come consigliato dalla doc `sdl`) ma questo distrugge lo stato di `opengl`
- ❖ quindi dopo ogni resize bisogna resettare TUTTO lo stato di `opengl` (`lighting` ecc.)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

2

Intro

- ❖ Passiamo dalla rappresentazione a cubetti ad una mesh vera e propria.
- ❖ Useremo feature del `c++`
- ❖ vedremo come fare una classe semplice che rappresenta una mesh.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

3

Stato

- ❖ Modifichiamo la funzione che disegna la pallina in modo da sostituire il blocchetto con una sfera.
- ❖ Per fare una banda colorata sulla pallina ne disegno due di differenti colori leggermente scalate.
- ❖ Attenzione alla precisione dello `Zbuffer`
 - ❖ fate esperimenti con differenti `near` e `far plane`...

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

4

Pallina che sembri tale

- ❖ usiamo `gluQuadric`
 - ❖ serve per disegnare generiche superfici quadriche (sfere, cilindri e dischi);
 - ❖ 1 creare un oggetto `gluNewQuadric`
 - ❖ `GLUquadricObj* sphere=gluNewQuadric()`
 - ❖ 2 settare come lo si disegna
 - ❖ `gluQuadricNormals(sphere, GLU_SMOOTH);`
 - ❖ 2 disegnare la quadrica
 - ❖ `gluSphere(radius,slice,stacks);`

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

5

Stato

- ❖ Fare attenzione allo stato di `opengl`
- ❖ la pallina va disegnata con
 - ❖ `glShadeModel(GL_SMOOTH);`
- ❖ mentre i cubetti con
 - ❖ `glShadeModel(GL_FLAT);`
- ❖ Tenere traccia delle modifiche allo stato di `opengl`

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

6

glPushAttrib/glPopAttrib

- ❖ salvano e ripristinano porzioni dello stato di opengl
 - ❖ glPushAttrib(mascheradibit)
 - ❖ glPopAttrib()
- ❖ dove maschera di bit serve a specificare quali porzioni dello stato salvare
 - ❖ GL_CURRENT_BIT
salva il colore corrente, la normale corrente ecc.
 - ❖ GL_ENABLE_BIT
salva tutto quello che si puo abilitare con glEnable

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

7

DrawBall()

```
void DrawBall(float radius)
{
    glPushAttrib(GL_LIGHTING_BIT);
    glPushMatrix();
    glShadeModel(GL_SMOOTH);
    static GLUquadricObj* sphere=0;
    if(!sphere) {
        sphere=gluNewQuadric();
        gluQuadricNormals(sphere, GLU_SMOOTH);
    }
    glColor3f(0,0,1);
    gluSphere(sphere,radius,16,8);
    glScalef(1.02,1.02,0.7);
    glColor3f(1,0,0);
    gluSphere(sphere,radius,16,8);
    glPopMatrix();
    glPopAttrib();
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

8

Da blocchi a mesh

- ❖ Si procede per passi:
 1. generare solo i vertici del bordo dell'anello (moebius02.cpp)
 2. costruirci la superficie che li connette con una triangle strip e fare le normali giuste (moebius03.cpp)
 3. costruire l'estrusione di un quadrato (moebius04.cpp)
 4. costruire l'estrusione di un generica forma (moebius05.cpp)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

9

Generare solo i vertici

- ❖ si usa le matrici di Opengl stesse per trasformare i vertici come vogliamo
- ❖ utilizziamo due classi accessorie point3 e matrix44 per trasformare punti con matrici in maniera semplice
- ❖ basta prendere la modelviewmatrix degli scorsi esempi e applicarla a dei vertici che stanno sul segmento
 - ❖ (-1,0,0)-(1,0,0)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

10

Point3/Point4

- ❖ Classe minimale per rappresentare punti3d
- ❖ templatata sul tipo usato per rappresentare le coordinate
- ❖ overloading di tutti gli operatori che ci si immagina +-*^ ecc.
- ❖ shorthand per I tipi comuni
 - ❖ typedef Point3<float> Point3f;
 - ❖ typedef Point3<double> Point3d;
 - ❖ typedef Point4<double> Point4d;

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

11

Matrix44

- ❖ Ultra classe minima per gestire matrici 4x4
- ❖ Soliti operatori +- tra matrici
- ❖ prodotto matrice*vettore -> vettore
- ❖ funzioni per prendere le matrici da opengl
- ❖ glGetModelViewMatrix
- ❖ glMultMatrix
- ❖ ecc.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

12

Memento matrici in opengl

- ❖ Opengl memorizza le matrici in *column major order*

```
a0 a4 a8 a12
a1 a5 a9 a13
a2 a6 a10 a14
a3 a7 a11 a15
```

- ❖ mentre noi teniamo un float[4][4]

```
a0 a1 a2 a3
a4 a5 a6 a7
a8 a9 a10 a11
a12 a13 a14 a15
```

- ❖ quindi si deve sempre trasporre se non si usano le funzioni apposite della libreria...

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

13

Strip di punti

```
void GenerateMoebiusRing(vector<Point3d> &Strip, int step,
float r, int twist)
{
    Strip.clear();
    float angle=360.0f/step;
    float angletwist=(180.0f*twist)/step;
    for(int i=0;i<step;++i)
    {
        Matrix44d mm;
        glPushMatrix();
        glLoadIdentity();
        glRotatef(i*angle,0,1,0);
        glTranslatef(r,0,0);
        glRotatef(i*angletwist,0,0,1);
        glGetModelViewMatrix(mm);
        Strip.push_back(mm*Point3d(-1,0,0));
        Strip.push_back(mm*Point3d( 1,0,0));
        glPopMatrix();
    }
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

14

Classe myMesh

- ❖ Iniziamo a scrivere una classe per tenere una superficie fatta di triangoli
- ❖ Inizialmente pensiamo solo a metter dentro I vertici e a disegnare la superficie con un hack

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

15

```
class myVertex {
public:
    myVertex(Point3d &pp, Point4d &nn) (p=pp; n=Point3d(nn[0],nn[1],nn[2]));

    Point3d p;
    Point3d n;
};

class myFace {
public:
    myVertex *v[3];
    Point3d n;
};

class myMesh {
public:
    vector< myVertex > vert;
    vector< myFace > face;
    void ComputeFaceNormal();
    void DrawStrip() const;
    bool empty() const (return vert.empty());
    void clear() (vert.clear(); face.clear());
};
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

16

Vertici

- ❖ Questa volta la classe vertici contiene anche la normale...
- ❖ Come facciamo a calcolare la normale trasformata?
- ❖ al solito facendo passare una normale in coordinate omogenee dentro la matrice di modellazione
 - ❖ le traslazioni non avranno effetto..

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

17

```
void GenerateMoebiusRing(myMesh &m, int step, float r, int twist)
{
    m.clear();
    float angle=360.0f/step;
    float angletwist=(180.0f*twist)/step;
    for(int i=0;i<step;++i)
    {
        Matrix44d mm;
        glPushMatrix();
        glLoadIdentity();
        glRotatef(i*angle,0,1,0);
        glTranslatef(r,0,0);
        glRotatef(i*angletwist,0,0,1);
        glGetModelViewMatrix(mm);
        m.vert.push_back(myVertex(mm*Point3d(-1,0,0),mm*Point4d(0,1,0,0)));
        m.vert.push_back(myVertex(mm*Point3d( 1,0,0),mm*Point4d(0,1,0,0)));
        glPopMatrix();
    }
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

18

Disegniamo la strip

```
void myMesh::DrawStrip() const
{
    glBegin(GL_TRIANGLE_STRIP);

    for(int i=0;i<vert.size();++i){
        glNormal(vert[i].n);
        glVertex(vert[i].p);
    }

    glEnd();
}
```

- ❖ Non funge bene, perche?
- ❖ Perche' non e' possibile definire bene le normali di un anello di moebius a spessore 0...
- ❖ costruiamo effettivamente la mesh di un anello fatto da due superfici leggermente spostate
- ❖ usiamo una funzione di disegno generica

Generate vertexes...

```
void GenerateMoebiusRing(myMesh &m, int step, float r, int twist)
{
    m.clear();
    float angle=360.0f/step;
    float angletwist=(180.0f*twist)/step;
    int i;
    for(i=0;i<step*2;++i)
    {
        Matrix44d mm;
        glPushMatrix();
        glLoadIdentity();
        glRotatef(i*angle,0,1,0);
        glTranslatef(r,0,0);
        glRotatef(i*angletwist,0,0,1);
        glGetModelViewMatrix(mm);
        m.vert.push_back(myVertex(mm*Point3d(-1,.2,0),mm*Point4d(0,1,0,0)));
        m.vert.push_back(myVertex(mm*Point3d( 1,.2,0),mm*Point4d(0,1,0,0)));
        glPopMatrix();
    }
}
```

and faces

```
...
int sz=m.vert.size();
for(i=0;i<step*2;++i)
{
    m.face.push_back(
        myFace(&m.vert[(i*2+0)%sz],
              &m.vert[(i*2+1)%sz],
              &m.vert[(i*2+2)%sz] ) );
    m.face.push_back(
        myFace(&m.vert[(i*2+1)%sz],
              &m.vert[(i*2+3)%sz],
              &m.vert[(i*2+2)%sz] ) );
}
// i*2+0      i*2+2
// | o-----o |
// | | / \ | |
// | | / \ | |
// | | / \ | |
// | o-----o |
// i*2+1      i*2+3
```

```
void myMesh::Draw() const
{
    glBegin(GL_TRIANGLES);

    for(int i=0;i<face.size();++i){
        glNormal(face[i].v[0]->n);
        glVertex(face[i].v[0]->p);
        glNormal(face[i].v[1]->n);
        glVertex(face[i].v[1]->p);
        glNormal(face[i].v[2]->n);
        glVertex(face[i].v[2]->p);
    }

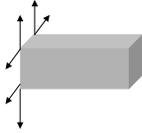
    glEnd();
}
```

Generalizziamo un minimo

- ❖ Facciamo che possa generare una qualunque sezione.
 - ❖ Una sezione viene passata come una sequenza di vertici nel piano xy
 - ❖ normali calcolate automaticamente e poi interpolate.
 - ❖ si usa doppi vertici per sharp features
- classe Shape

Gestione Crease

- ❖ Per fare gli angoli "vivi" occorre dare differenti normali per lo stesso vertice



- ❖ Quindi per fare un quadrato occorrono 8 vertici

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

25

Classe ausiliare myShape

- ❖ Per gestire poligoni nel piano
- ❖ Si duplicano semplicemente I vertici dove vogliamo crease
- ❖ calcolo delle normali ai vertici automatico
 - ❖ per trovare la normale ad un segmento basta ruotarlo di 90°
 - ❖ Normale al vertice
 - ❖ media se non crease
 - ❖ quella del segmento se crease

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

26

Classe myShape

```
class myShape
{
public:
    vector<myVertex> poly;
    void ComputeNormals();
    void BuildNGon(int n, double rad);
};

void myShape::BuildNGon(int n, double rad)
{
    poly.resize(n*2);
    double a = 2.0*M_PI/double(n);

    for(int i=0;i<n;i++) {
        poly[i*2+0].p=Point3d( rad*cos(i*a - a/2), rad*sin(i*a - a/2) ,0);
        poly[i*2+1].p=Point3d( rad*cos(i*a + a/2), rad*sin(i*a + a/2) ,0);
    }
    ComputeNormals();
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

27

Calcolo delle normali

```
void myShape::ComputeNormals()
{
    int i;
    int sz=poly.size();
    for(i=0;i<sz;++i)
    {
        Point3d eprev = poly[ i ] .p - poly[(i-1+sz)%sz].p;
        Point3d enext = poly[(i+1)%sz].p - poly[ i ] .p;
        // per trovare la normale basta ruotare di 90 gradi intorno
        // all'asse z cioe' scambiare x e y cambiando di segno alla y
        Point3d nprev,nnext;
        nprev.x() = eprev.y(); nprev.y() = -eprev.x(); nprev.z() = 0;
        nnext.x() = enext.y(); nnext.y() = -enext.x(); nnext.z() = 0;

        poly[i].n=Point3d(0,0,0);
        // se uno degli edge era nullo non si considera nel calcolo della norm.
        if(eprev.SquaredNorm() !=0) poly[i].n+=nprev;
        if(enext.SquaredNorm() !=0) poly[i].n+=nnext;

        poly[i].n.Normalize();
    }
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

28

Generate Ring Vertexes

```
void GenerateMoebiusRing(myMesh &m, myShape &s, int step, float r, float twistDeg)
{
    m.clear();
    float angle=360.0f/step;
    float angletwist=(twistDeg)/step;
    int i;
    for(i=0;i<=step;++i)
    {
        Matrix44d mm;
        glPushMatrix();
        glLoadIdentity();
        glRotatef(i*angle,0,1,0);
        glTranslatef(r,0,0);
        glRotatef(i*angletwist,0,0,1);
        glGetModelViewMatrix(mm);
        for(int j=0;j<s.poly.size();++j)
            m.vert.push_back(myVertex(mm*s.poly[j].p, mm*s.poly[j].n4()));
        glPopMatrix();
    }
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

29

Generate Ring Faces

```
int sz=m.vert.size();
int szs=poly.size();
for(i=0;i<step;++i)
{
    for(int j=0;j<s.poly.size();++j)
        if(s.poly[j].p != s.poly[(j+1)%sz].p)
        {
            m.face.push_back(myFace(
                &m.vert[(i+0)*szs+(j+0)%sz] ,
                &m.vert[(i+0)*szs+(j+1)%sz] ,
                &m.vert[(i+1)*szs+(j+1)%sz] ));
            m.face.push_back(myFace(
                &m.vert[(i+0)*szs+(j+0)%sz] ,
                &m.vert[(i+1)*szs+(j+1)%sz] ,
                &m.vert[(i+1)*szs+(j+0)%sz] ));
        }
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

30