void GLRenderer::**DrawPyramid**(double side, double height, int n)

{

double angleStep = (2 \* piconst) / n;

double currAngle = 0;

double halfAngle = angleStep / 2;

double r = (side / 2) / tan(halfAngle);

double L = sqrt(r \* r + height \* height);

double R = sqrt(r \* r + (side / 2) \* (side / 2));

float ny = r / L;

float nr = height / L;

glBegin(GL\_TRIANGLES);

{

for (int i = 0; i < n; i++)

{

glNormal3f(nr \* cos(currAngle + (angleStep / 2)), ny, -nr \* sin(currAngle + (angleStep / 2)));

float x1 = R \* cos(currAngle);

float x2 = R \* cos(currAngle + angleStep);

float z1 = -R \* sin(currAngle);

float z2 = -R \* sin(currAngle + angleStep);

glVertex3f(0, height, 0);

glVertex3f(x1, 0, z1);

glVertex3f(x2, 0, z2);

currAngle += angleStep;

}

}

glEnd();

currAngle = 0;

glBegin(GL\_TRIANGLE\_FAN);

{

glNormal3f(0.0, -1.0, 0.0);

glVertex3f(0.0, 0.0, 0.0);

for (int i = 0; i < n + 1; i++)

{

glVertex3f(R \* cos(currAngle), 0.0, R \* sin(currAngle));

currAngle += angleStep;

}

}

glEnd();

}

**void GLRenderer::DrawCylinder(float r, float h, float nr, float nh, bool axes) //sa TexCord**

{

float texStepY = 1.0 / nh;

float texStepX = 1.0 / nr;

float texY = 0;

float angStep = (2 \* piconst) / nr; //za x i z(i nalazenje normala)

for (int i = 0; i < nh; i++) // **omotac**

{

float currAng = 0;

float texX = 0;

glBegin(GL\_QUAD\_STRIP);

{

for (int j = 0; j < nr + 1; j++)

{

float x = r \* cos(currAng);

float z = -r \* sin(currAng);

float nx = x / r;

float nz = z / r;

glNormal3f(nx, 0, nz);

glTexCoord2f(texX, texY);

glVertex3f(x, h/2, z);

glTexCoord2f(texX, texY + texStepY);

glVertex3f(x, -h/2, z);

texX += texStepX;

currAng += angStep;

}

}

glEnd();

texY += texStepY;

}

glBegin(GL\_TRIANGLE\_FAN); **//osnova**

{

glNormal3f(0.0, 1.0, 0.0);

glTexCoord2f(0.5, 0.5);

glVertex3f(0, h / 2, 0);

float currAng = 0;

for (int i = 0; i < nr + 1; i++)

{

float x = r \* cos(currAng);

float z = -r \* sin(currAng);

float texX = 0.5 + 0.5 \* cos(currAng);

texY = 0.5 - 0.5 \* sin(currAng);

glTexCoord2f(texX, texY);

glVertex3f(x, h / 2, z);

currAng += angStep;

}

}

glEnd();

glBegin(GL\_TRIANGLE\_FAN);

{

glNormal3f(0.0, -1.0, 0.0);

glTexCoord2f(0.5, 0.5);

glVertex3f(0, -h / 2, 0);

float currAng = 0;

for (int i = 0; i < nr + 1; i++)

{

float x = r \* cos(currAng);

float z = r \* sin(currAng);

float texX = 0.5 + 0.5 \* cos(currAng);

texY = 0.5 - 0.5 \* sin(currAng);

glTexCoord2f(texX, texY);

glVertex3f(x, -h / 2, z);

currAng += angStep;

}

}

glEnd();

float texStepY = 1.0 / nh;

float texStepX = 1.0 / nr;

float texY = 0;

float angStep = (2 \* piconst) / nr;

void GLRenderer::**DrawTube**(double r1, double r2, double h, int n) //**valjak sa razlicitim polupr. osnove**

{

glBegin(GL\_QUAD\_STRIP);

{

float texPart = 1.0 / (float)n; // kad imamo n segmenata nekog tela

float angleStep = (2 \* pi) / (float)n; //kad nam treba neki ugao, okrugli objekat

float currAngle = 0;

float y1 = h / 2, y2 = -h / 2;

for (int i = 0; i <= n; i++)

{

float x1 = r1 \* cos(currAngle);

float z1 = r1 \* sin(currAngle);

float x2 = r2 \* cos(currAngle);

float z2 = r2 \* sin(currAngle);

float rDiff = r2 - r1;// ove dve pisem da bih nasla ny i nr (normale)

float L = sqrt(h \* h + rDiff \* rDiff);

float ny = rDiff / L;

float nr = h / L;

float nx = nr \* cos(currAngle);

float nz = nr \* sin(currAngle);

glNormal3f(nx, ny, nz);

glTexCoord2f(i \* texPart, 0.0);

glVertex3f(x1, y1, z1);

glTexCoord2f(i \* texPart, 1.0);

glVertex3f(x2, y2, z2);

currAngle += angleStep;

}

}

glEnd();

}

void GLRenderer::**DrawCone**(double r, double h, int n) //**Kupa** bez dna

{

glBegin(GL\_TRIANGLES);

{

float currAngle = 0;

float angleStep = (2 \* pi) / n;

float L = sqrt(h \* h + r \* r);

float ny = r / L;

float nr = h / L;

for (int i = 0; i < n; i++)

{

float nx = nr \* cos(currAngle + angleStep / 2); // zasto + angleStep / 2??

float nz = nr \* sin(currAngle + angleStep / 2); // zato sto se vrh trougla nalazi na sredini stranie, tj. izmedju tacaka koje su definisane dole

glNormal3f(nx, ny, nz);

glVertex3f(0, h, 0);

nx = nr \* cos(currAngle); // normale racunamo sa nr; prva tacka bez rastojanja

nz = nr \* sin(currAngle);

glNormal3f(nx, ny, nz);

float x = r \* cos(currAngle); //a koordinate sa obicno r

float z = r \* sin(currAngle);

glVertex3f(x, 0, z);

nx = nr \* cos(currAngle + angleStep); //druga tacka sa plus rastojanje, tj. ugao

nz = nr \* sin(currAngle + angleStep);

glNormal3f(nx, ny, nz);

x = r \* cos(currAngle + angleStep);

z = r \* sin(currAngle + angleStep);

glVertex3f(x, 0, z);

currAngle += angleStep;

}

}

glEnd();

}