

Računarska grafika - vežbe

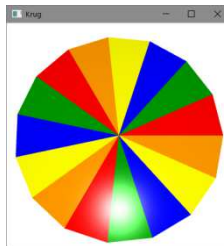
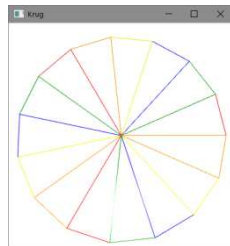
8 – JavaFX 3D mreža i tekstura

Zadatak 1: Mreža kruga

Formirati trougaonu mrežu kruga poluprečnika R i N podela kružnice, u X-Z ravni, sa centrom u koordinatnom početku, a zatim formirati krug opisan datom trougaonom mrežom kao 3D objekat i prikazati ga perspektivnom kamerom u obliku mreže (žični model) i popunjeno (neproziran objekat) koristeći najpre difuznu boju, a zatim teksturu sa priložene slike.



Priložen je očekivani izgled kruga za broj podela $N=15$, za sliku perspektivne kamere sa lampom na nosaču rotiranom oko X-ose za ugao -80° i materijalom kruga sa mapom difuzije na osnovu priložene slike i odrazom belog svetla snage 150:



Rešenje: Mreža kruga (1/10)

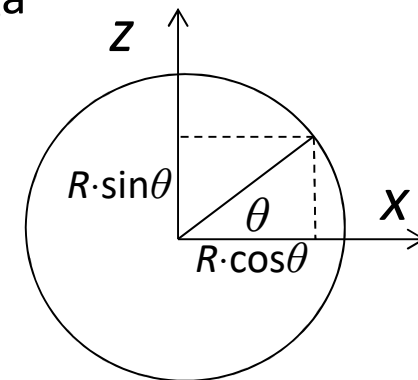
Koordinate temena na kružnici:

- dobijaju se na osnovu parametarskih jednačina kruga
- krug je u XoZ ravini:

$$x = R \cdot \cos(\varphi)$$

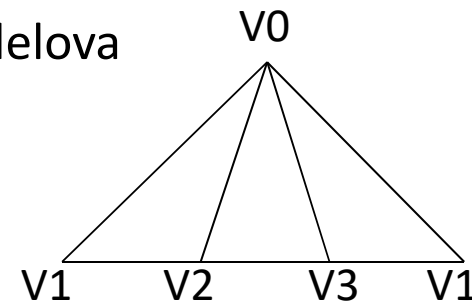
$$y = 0$$

$$z = R \cdot \sin(\varphi)$$



Trougaona mreža se formira od trouglova:

- zajedničko teme (v_0) u centru kružnice
- po 2 temena (v_i, v_{i+1}) na kružnici, podeljenoj na N delova
- lica: v_0, v_1, v_2 ; v_0, v_2, v_3 ; v_0, v_3, v_1
- naličja: v_0, v_2, v_1 ; v_0, v_3, v_2 ; v_0, v_1, v_3




Rešenje: Mreža kruga (2/10)

...




```
import javafx.scene.shape.MeshView;
import javafx.scene.shape.TriangleMesh;
import javafx.scene.shape.DrawMode;
import javafx.scene.shape.CullFace;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.paint.Color;
import javafx.scene.PerspectiveCamera;
import javafx.scene.PointLight;
import javafx.scene.transform.Rotate;
import javafx.event.EventHandler;
import javafx.scene.input.MouseEvent;
import javafx.scene.input.ScrollEvent;
import javafx.scene.image.Image;
```

Rešenje: Mreža kruga (3/10)



```
public class Krug1 extends Application {
    public static final int R = 100;
    public static final int N = 15;
    private static final double DALJA_ODSECAJUĆA_RAVAN = 2000.0;
    private static final double POČETNO_RASTOJANJE_KAMERE = -400.0;
    private static final double POČETNI_UGAO_KAMERE = -80;
    private static final double ŠIRINA_SCENE = 400;
    private static final double VISINA_SCENE = 400;
    private final Rotate rx = new Rotate();
    private final Rotate ry = new Rotate();
    private final Rotate rz = new Rotate();
    private double pozX;           private double pozY;
    private double staraPozX;     private double staraPozY;
    private PerspectiveCamera kamera;
```

Rešenje: Mreža kruga (4/10)



```
private MeshView napraviPovrš(float r, int n) {
    float[] temena = new float[3*(n+1)];
    temena[0] = 0f; temena[1] = 0f; temena[2] = 0f;
    // indeksi temena, teksturnih koordinata i stranica
    int iV = 3, iT = 0, iS = 0;


    double deltaUgao = 2*Math.PI/n;
    double ugao=0.0;
    for (int i=0; i<n; i++) {
        temena[iV++]= (float)(r*Math.cos(ugao));
        temena[iV++]= 0;
        temena[iV++]= (float)(r*Math.sin(ugao));
        ugao+=deltaUgao;
    }
}
```

Rešenje: Mreža kruga (5/10)




```
// Teksturane koordinate preko skale boja
float[] tekstura = {
    0.1f, 0.5f, // 0 crvena
    0.3f, 0.5f, // 1 zelena
    0.5f, 0.5f, // 2 plava
    0.7f, 0.5f, // 3 žuta
    0.9f, 0.5f // 4 narandžasta
};
```

Rešenje: Mreža kruga (6/10)



```
// Stranice
int[] stranice = new int [12*n];  iV=1;
for (int i=0; i<n; i++) {
    // lice
    stranice[iS++]=0;                stranice[iS++]=iT;
    stranice[iS++]=iV;                stranice[iS++]=iT;
    stranice[iS++]=(iV+1)%(n+1)!=0?iV+1:1; stranice[iS++]=iT;
    //naličje
    stranice[iS++]=0;                stranice[iS++]=iT;
    stranice[iS++]=(iV+1)%(n+1)!=0?iV+1:1; stranice[iS++]=iT;
    stranice[iS++]=iV;                stranice[iS++]=iT;
    iV++; // indeks temena
    iT=(iT+1)%(tekstura.length/2); // indeks teksture
}
```


Rešenje: Mreža kruga (7/10)




```
TriangleMesh mreža = new TriangleMesh();
mreža.getPoints().addAll(temena);
mreža.getTexCoords().addAll(tekstura);
mreža.getFaces().addAll(stranice);
MeshView površ = new MeshView();  površ.setMesh(mreža);
PhongMaterial mat = new PhongMaterial();
mat.setDiffuseColor(Color.ORANGE);
//     mat.setDiffuseMap(new Image("skalaboja.png"));
mat.setSpecularPower(150);
površ.setMaterial(mat);
//     površ.setDrawMode(DrawMode.LINE);
//     površ.setCullFace(CullFace.FRONT);
return površ;
}
```

Rešenje: Mreža kruga (8/10)

```
private Group napraviKameruNaNosacuSaLampom(){
    kamera = new PerspectiveCamera(true);
    kamera.setFarClip(DALJA_ODSECAJUĆA_RAVAN);
    kamera.setTranslateZ(POČETNO_RASTOJANJE_KAMERE);
    PointLight lampa = new PointLight();
    lampa.setColor(Color.WHITE);
    lampa.setTranslateX(kamera.getTranslateX());
    lampa.setTranslateY(kamera.getTranslateY());
    lampa.setTranslateZ(kamera.getTranslateZ());
    Group nosač = new Group(kamera, lampa);
    rx.setAxis(Rotate.X_AXIS); rx.setAngle(POČETNI_UGAO_KAMERE);
    ry.setAxis(Rotate.Y_AXIS); ry.setAngle(0);
    rz.setAxis(Rotate.Z_AXIS); rz.setAngle(0);
    nosač.getTransforms().addAll(rz, ry, rx); return nosač;
}
```

19.04.2017.


Rešenje: Mreža kruga (9/10)



```
private void obradaDogađaja(Scene scena) {
    // obrada događaja miša i točkića miša
    // ...
    scena.addEventHandler(MouseEvent.MOUSE_PRESSED, r1);
    scena.addEventHandler(MouseEvent.MOUSE_DRAGGED, r2);
    scena.addEventHandler(ScrollEvent.SCROLL, r3);
}

private Scene napraviScenu(Group koren) {
    SceneAntialiasing glatko=SceneAntialiasing.BALANCED;
    Scene scena = new Scene(koren, ŠIRINA_SCENE, VISINA_SCENE,
                            true, glatko);
    scena.setCamera(kamera); obradaDogađaja(scena);
    return scena;
}
```

Rešenje: Mreža kruga (10/10)

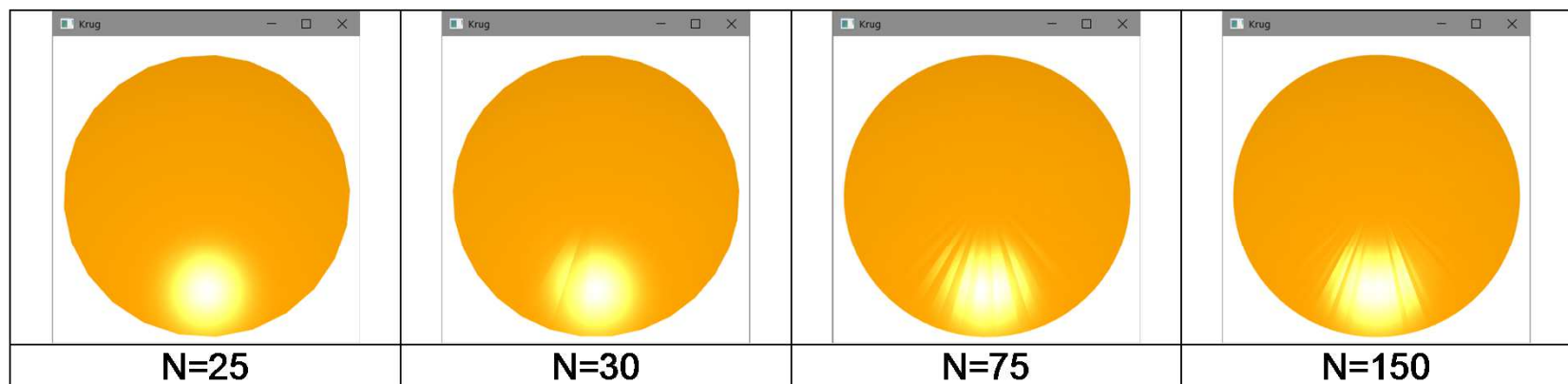


```
private Group napraviGrafScene(){
    MeshView krug=napraviPovrš(R,N);
    Group nosač = napraviKameruNaNosačuSaLampom();
    Group graf = new Group(krug,nosač);
    return graf;
}
@Override public void start(Stage prozor) {
    Group koren = napraviGrafScene();
    Scene scena = napraviScenu(koren);
    prozor.setTitle("Krug");
    prozor.setScene(scena);    prozor.show();
}
public static void main(String[] args) { launch(args); }
}
```

Rezultat: Mreža kruga



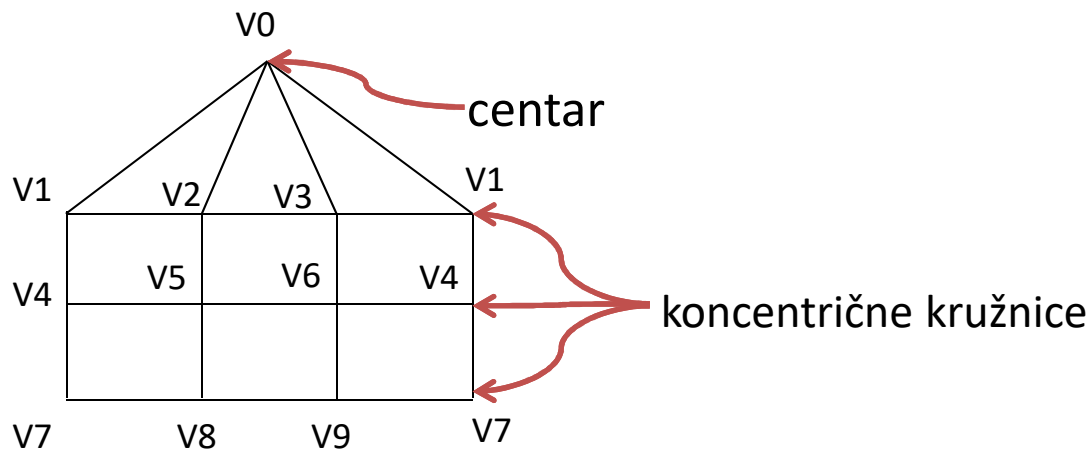
Problem:



2. Rešenje: Mreža kruga (1/9)

Trougaona mreža kruga se formira na sledeći način:

- od trouglova koje obrazuje N tačaka na svakoj od koncentričnih kružnica sa poluprečnicima R/N
- izuzev trouglova koje obrazuju tačke na kružnici sa najmanjim poluprečnikom i centrom kruga



2. Rešenje: Mreža kruga (2/9)

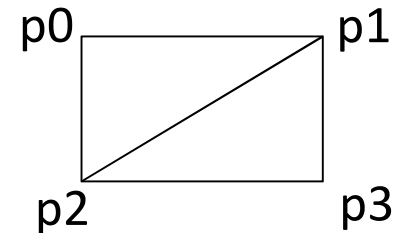
Indeksi temena na stranicama koje čine trouglovi sa temenom u centru kružnice – određuju se kao u prethodnom rešenju:

lica: $0,1,2; 0,2,3; \dots; 0,n,1$

naličja: $0,2,1; 0,3,2; \dots; 0,1,n$

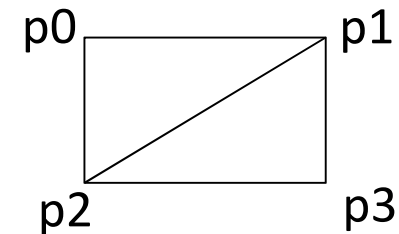
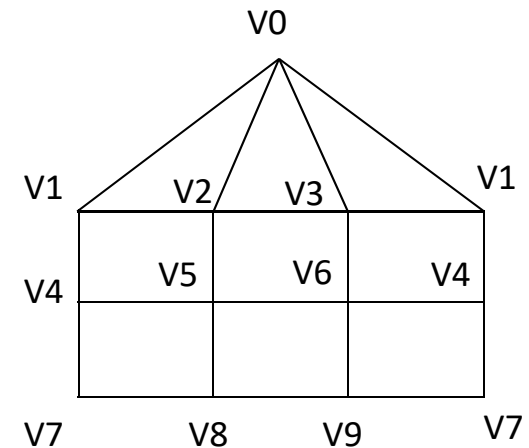
Indeksi temena na koncentričnom kružnicama:

- indeks temena u posmatranom pravougaoniku je p_i ,
gde se i kodira na osnovu binarnog dvocifrenog broja
- prva binarna cifra (koordinata) odgovara pravcu horizontalne ose
(indeks tačke na kružnici)
- druga cifra pravcu vertikalne ose (indeks kružnice)
- p_0 (p_{00_2}), p_1 (p_{01_2}), p_2 (p_{10_2}), p_3 (p_{11_2})




2. Rešenje: Mreža kruga (3/9)

```
for (int i = 0; i < n-1; i++) { // i: indeks kružnice
    for (int j = 0; j < n; j++) { // j: indeks tačke na i-toj kružnici
        p0 = i * n + j;
        p = p0 + 1;    p1= p%n != 0 ? p : p-n;
        p2 = p0 + n;
        p = p2 + 1;    p3= p%n != 0 ? p : p-n;
        p0++; p1++; p2++; p3++;
        //gornji trougao, lice:    p2,p1,p0
        //gornji trougao, naličje: p0,p1,p2
        //donji trougao, lice:    p1,p2,p3
        //donji trougao, naličje: p3,p2,p1
    }
}
```




2. Rešenje: Mreža kruga (4/9)



```
private MeshView napraviPovrš(float rKrug, int n) {
    float[] temena = new float[3*(n*n+1)];
    temena[0] = 0f; temena[1] = 0f; temena[2] = 0f; // centar
    int iV = 3, iT = 0, iS = 0;
    float deltaR = rKrug/n; double deltaUgao = 2 *Math.PI/n;
    float r=deltaR;
    for (int i=0; i<n; i++) { // i - indeks kružnice
        double ugao=0.0;
        for (int j=0; j<n; j++) { // j - indeks temena na kružnici
            temena[iV++]=((float)(r*Math.cos(ugao))); // x
            temena[iV++]=0; // y
            temena[iV++]=((float)(r*Math.sin(ugao))); // z
            ugao+=deltaUgao;
        } r+=deltaR;
    }
}
```


2. Rešenje: Mreža kruga (5/9)



```
// Teksturne koordinate preko skale boja
float[] tekstura = {
    0.1f, 0.5f, // 0 crvena
    0.3f, 0.5f, // 1 zelena
    0.5f, 0.5f, // 2 plava
    0.7f, 0.5f, // 3 žuta
    0.9f, 0.5f // 4 narandžasta
};


int[] stranice = new int [12*(n+(n-1)*n*2)]; // 24*n*n-12*n
iV=1;
```

2. Rešenje: Mreža kruga (6/9)



```
for (int i=0; i<n; i++) {  
    // lice  
    stranice[iS++] = 0;           stranice[iS++] = iT;  
    stranice[iS++] = iV;        stranice[iS++] = iT;  
    stranice[iS++] = (iV+1)%(n+1) != 0 ? iV+1 : 1;  stranice[iS++] = iT;  
  
    // naličje  
    stranice[iS++] = 0;           stranice[iS++] = iT;  
    stranice[iS++] = (iV+1)%(n+1) != 0 ? iV+1 : 1;  stranice[iS++] = iT;  
    stranice[iS++] = iV;        stranice[iS++] = iT;  
  
    iV++;  
    iT = (iT+1)%(tekstura.length/2); // indeks teksture  
}
```

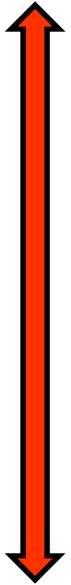
2. Rešenje: Mreža kruga (7/9)



```
int p0, p1, p2, p3; // indeksi temena četvorougla: GL,GD,DL,DD
int p; // pomoćna promenljiva za izračunavanje indeksa
int t0 = 0, t1 = 0, t2 = 0, t3 = 0; // indeksi teksturnih tačaka
for (int i = 0; i < n-1; i++) { // i - indeks kružnice
for (int j = 0; j < n; j++) { // j - indeks tačke na kružnici
    p0 = i * n + j;
    p = p0 + 1;    p1= p%n != 0 ? p : p-n;
    p2 = p0 + n;
    p = p2+1;    p3= p%n != 0 ? p : p-n;
    p0++; p1++; p2++; p3++;

    t0=t1=t2=t3=iT;
```


2. Rešenje: Mreža kruga (8/9)



```
//gornji trougao, lice  
stranice[iS++] = p2;      stranice[iS++] = t2;  
stranice[iS++] = p1;      stranice[iS++] = t1;  
stranice[iS++] = p0;      stranice[iS++] = t0;
```

```
//gornji trougao, naličje  
stranice[iS++] = p0;      stranice[iS++] = t0;  
stranice[iS++] = p1;      stranice[iS++] = t1;  
stranice[iS++] = p2;      stranice[iS++] = t2;
```

2. Rešenje: Mreža kruga (9/9)



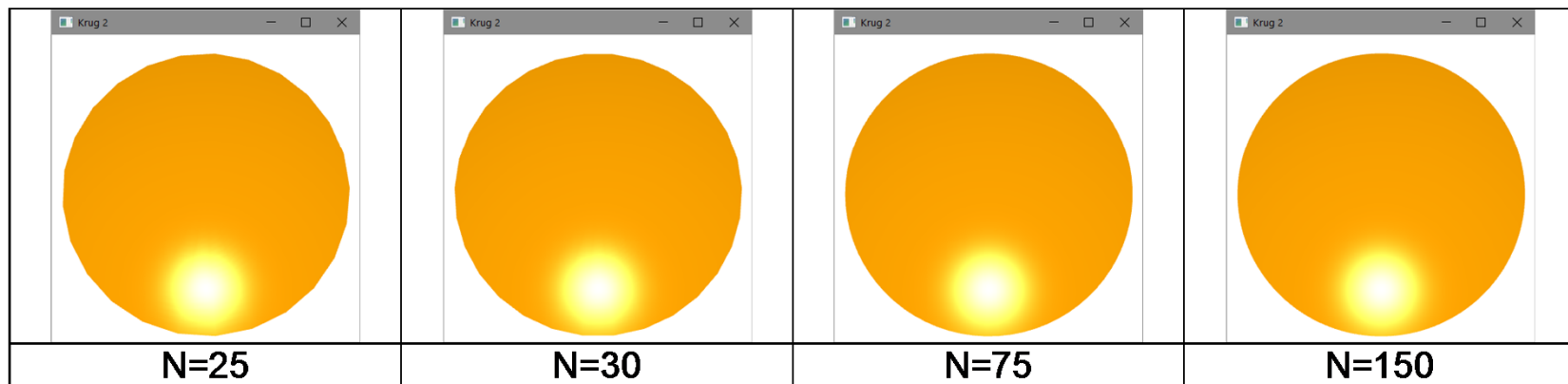
```
//donji trougao, lice
stranice[iS++] = p1;      stranice[iS++] = t1;
stranice[iS++] = p2;      stranice[iS++] = t2;
stranice[iS++] = p3;      stranice[iS++] = t3;

//donji trougao, naličje
stranice[iS++] = p3;      stranice[iS++] = t3;
stranice[iS++] = p2;      stranice[iS++] = t2;
stranice[iS++] = p1;      stranice[iS++] = t1;
}
iT=(iT+1)%(tekstura.length/2); // indeks teksture
}
// Preostali deo metoda je isti kao i u prethodnom rešenju
}
```

2. Rezultat: Mreža kruga



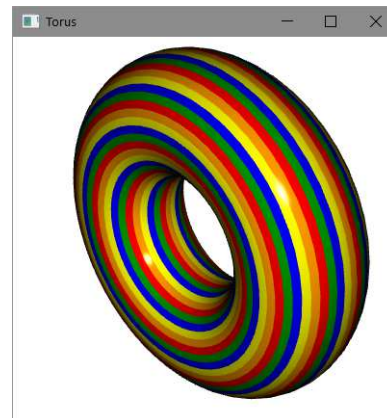
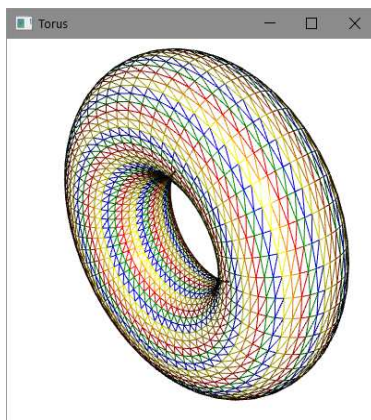
Ne postoji problem sa odrazom, ali je mreža značajno kompleksnija:



Zadatak 2: Mreža torusa

Formirati trougaonu mrežu torusa čija je srednja linija u ravni X-O-Y, sa centrom u koordinatnom početku. Zadati su poluprečnik kružnice srednje linije torusa R i poluprečnik kružnice poprečnog preseka r , kao i brojevi podela po srednjoj liniji torusa i kružnici poprečnog preseka, nU i nP , respektivno. Koristiti sliku iz prethodnog zadatka za teksturu torusa.

Priložen je očekivani izgled torusa:



Rešenje: Mreža torusa (1/7)

Koordinate temena mreže torusa su:

$$x=(R+r\cdot\cos\varphi)\cdot\cos\theta$$

$$y=(R+r\cdot\cos\varphi)\cdot\sin\theta$$

$$z=r\cdot\sin\varphi$$

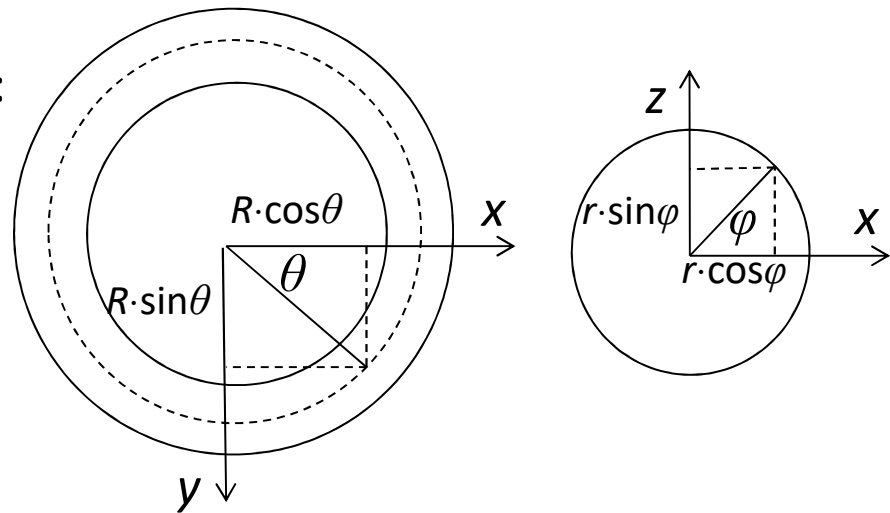
gde su:

R – poluprečnik kružnice koja predstavlja srednju liniju torusa

r – poluprečnik poprečnog preseka torusa

θ – ugao odklona do poprečnog preseka na kružnici srednje linija torusa

φ – ugao odklona do temena na kružnici poprečnog preseka

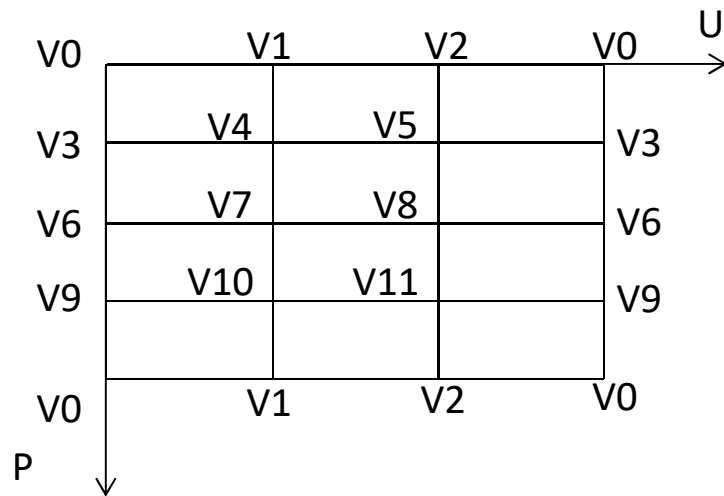


Rešenje: Mreža torusa (2/7)

Mreža torusa se formira na osnovu mreže pravougaonika

- svaki pravougaonik je podeljen glavnom dijagonalom na 2 trougla

Razmotana mreža torusa, posmatrana „iznutra“:



v_i – i -to teme u vektoru temena

U – uzdužna osa mreže

(odgovara srednjoj liniji)

P – poprečna osa mreže

(odgovara kružnici poprečnog preseka)

n_U – broj podela po U osi ($n_U=3$)

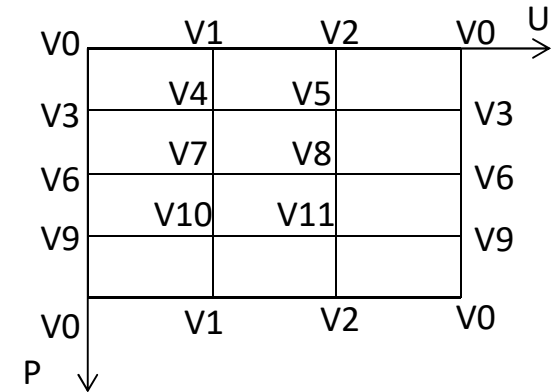
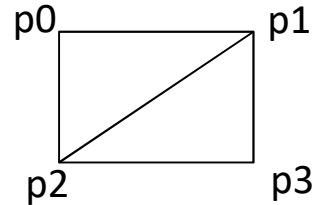
n_P – broj podela po P osi ($n_P=4$)

Rešenje: Mreža torusa (3/7)

Lica trouglova:


gornji: p0, p1, p2

donji: p3, p2, p1




```
for (int i = 0; i < nP; i++) { // i: indeks tačke na preseku
    for (int j = 0; j < nU; j++) { // j: indeks poprečnog preseka
        p0 = i * nU + j;
        p = p0+1;           p1= p%nU != 0 ? p : p-nU;
        p = p0 + nU;       p2 = p < nP * nU ? p : j;
        p = p2+1;          p3= p%nU != 0 ? p : p-nU;
        // određivanje trouglova u mreži na osnovu p0,p1,p2 i p3
    }
}
```

Rešenje: Mreža torusa (4/7)



```
/* Prerađen metod autora J.Pereda
 *   FXyzLib/src/org/fxyz/shapes/primitives/TorusMesh.java
 */
private MeshView napraviTorus(
    int nU, // broj podela po uzdužnoj srednjoj liniji cevi torusa
    int nP, // broj podela kružnice poprečnog preseka cevi torusa
    float R, // poluprečnik uzdužne srednje linije cevi torusa
    float r) { // poluprečnik kružnice poprečnog preseka cevi torusa
    float[] temena = new float[nP * nU * 3];
    int[] stranice = new int [nP * nU * 12]; // bez naličja
    int iV = 0, iT = 0, iS = 0; // indeksi u nizovima tem.,tek.,str.
    int p0, p1, p2, p3; // indeksi temena tekućeg četvorougla
    int p; // pomoćna promenljiva za izračunavanje indeksa
    int t0 = 0, t1 = 0, t2 = 0, t3 = 0; // indeksi teksturnih tačaka
```

Rešenje: Mreža torusa (5/7)



```
// temena
for (int iP = 0; iP < nP; iP++) { // iP - po temenu preseka
    float phi = iP * 2.0f * (float)Math.PI/nP;
    for (int iU = 0; iU < nU; iU++) { // iU - po presecima
        float theta = iU * 2.0f * (float)Math.PI/nU;
        temena[iV++] = (float)((R + r*Math.cos(phi))*Math.cos(theta)); // x
        temena[iV++] = (float)((R + r*Math.cos(phi))*Math.sin(theta)); // y
        temena[iV++] = (float)(r* Math.sin(phi)); // z
    }
}
```


```
// teksturne koordinate kao u prethodnom zadatku
```

Rešenje: Mreža torusa (6/7)

```
//stranice
for (int i = 0; i < nP; i++) { // i - indeks na poprečnom preseku
    for (int j = 0; j < nU; j++) { // j - indeks poprečnog preseka
        p0 = i * nU + j;
        p = p0 + 1;  p1= p%nU != 0 ? p : p-nU;
//                p1= p%nU != 0 ? p : i*nU; // alternativa
        p = p0 + nU; p2 = p < nP*nU ? p : j;
//                p2 = p < nP * nU ? p : p%nP; // alternativa
        p = p2 + 1;  p3= p%nU != 0 ? p : p-nU;
//                p3= p%nU != 0 ? p : (i+1)%nP*nU; // alternativa

        t0=t1=t2=t3=iT;
```

Rešenje: Mreža torusa (7/7)



```
// Gornji trougao, lice
stranice[iS++] = p0;           stranice[iS++] = t0;
stranice[iS++] = p1;           stranice[iS++] = t1;
stranice[iS++] = p2;           stranice[iS++] = t2;

// Donji trougao, lice
stranice[iS++] = p3;           stranice[iS++] = t3;
stranice[iS++] = p2;           stranice[iS++] = t2;
stranice[iS++] = p1;           stranice[iS++] = t1;
}
iT=(iT+1)%(tekstura.length/2);
}
// Preostali deo koda kao u prethodnom zadatku
}
```

Rezultat: Mreža torusa

Perspektivna kamera sa lampom na nosaču rotiranom oko Y-ose za ugao = -45° i oko X-ose za ugao = -30°

Materijal ima refleksivnost snage 150.

