

# Računarska grafika

JavaFX - mreža,  
tekstura



# Pojam proizvoljne geometrije

- JavaFX nudi samo 3 konkretne potklase geometrijskih tela:
  - za kvadar (`Box`), valjak (`Cylinder`) i loptu (`Sphere`)
  - klase su izvedene iz klase `Shape3D`
- Da bi se formirale površi sa proizvoljnom geometrijom
  - potrebno je definisati mrežu trouglova koja određuje površ
  - površ može biti zatvorena (telo) ili otvorena
- Da bi se površ prikazala, potrebne su sledeće informacije :
  - niz temena površi
  - niz koordinata karakterističnih tačaka teksture
  - niz stranica (trouglova)
    - stranice su određene indeksima temena i tačaka teksture

# Trougaona mreža i prikaz mreže

- Proizvoljno geometrijsko telo obrazovano od mreže trouglova
  - tipa `MeshView` izvedene klase iz `Shape3D`
- Klasa `TriangleMesh` – definisanje mreže trouglova
- Objektu klase `MeshView` se postavlja mreža trouglova
- Obe klase nalaze se u paketu `javafx.scene.shape`
- Formiranje proizvoljne površi (tela) tipa `MeshView`:
  - stvore se objekti klase `TriangleMesh` i klase `MeshView`
  - dohvata se zbirka postojećih temena (prazna) i dodaju nova temena
  - dohvata se zbirka postojećih i dodaju nove teksturne koordinate
  - dohvata se zbirka postojećih i dodaju nove stranice (trouglovi)
  - postavi se objektu klase `MeshView` formirani objekat klase `TriangleMesh`
  - postavi se objektu klase `MeshView` materijal sa difuznom mapom (teksturom)

# Obrazac za formiranje površi

- Površ se formira na sledeći način:

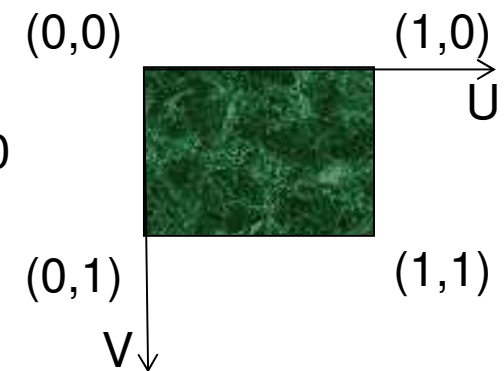
```
TriangleMesh mreža = new TriangleMesh();
mreža.getPoints().addAll(temena);
mreža.getTexCoords().addAll(teksturneKoordinate);
mreža.getFaces().addAll(stranice);
MeshView telo = new MeshView();
telo.setMesh(mreža);
PhongMaterial materijal = new PhongMaterial();
materijal.setDiffuseMap(new Image("Tekstura.png"));
telo.setMaterial(materijal);
```

# Temena površi (tela)

- Temena se zadaju nizom njihovih koordinata
- Koordinate su tipa `float`
- Svako teme je predstavljeno sa 3 koordinate u nizu
- Koordinate se navode po redosledu: `x, y, z`
- Temena se navode uzastopno: `x1, y1, z1, x2, y2, z2, ...`
- Nebitan je redosled temena u nizu
- Površi treba definisati u lokalnom koordinatnom sistemu
- Centar površi treba da bude u koordinatnom početku
  - po analogiji sa “standardnim” telima koja su tako definiisana
  - olakšava manipulisanje svim površima/telima u sceni

# Tekstura

- Teksture su matrice piksela kojima se boji mreža trouglova
- Definišu se:
  - na osnovu učitane slike ili programski
  - u 2D (U-V) koordinatnom sistemu
- U-V koordinatni sistem
  - opseg U i V koordinata teksture je od 0.0 do 1.0
  - kordinata U raste sleva-udesno
  - koordinata V raste odozgo-naniže
- Koordinate četiri ugaona temena teksture, počevši od gornjeg levog ugla, u smeru kazaljke na časovniku:
  - (0,0), (1,0), (1,1), (0,1)



# Teksturne koordinate

- Potrebno je zadati tačke teksture u koje se preslikavaju temena
- Tačke se zadaju nizom njihovih koordinata
- Koordinate su tipa `float`
- Koordinate se zadaju u U-V koordinatnom sistemu
- Svaka bitna tačka teksture je predstavljena sa 2 koordinate u nizu
- Koordinate se navode po redosledu: `u, v`
- Tačke teksture se navode uzastopno: `u1, v1, u2, v2, ...`
- Nebitan je redosled tačaka teksture u nizu

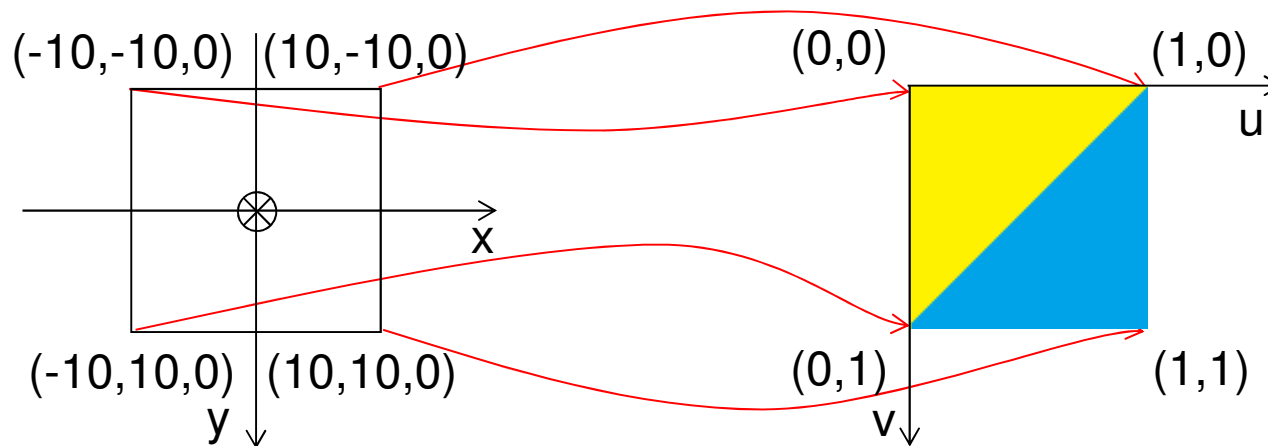
# Stranice – trouglovi u mreži

- Stranice se zadaju nizom temena trouglova
- Teme trougla je predstavljeno parom indeksa:
  - indeks temena trougla u nizu temena
  - indeks u nizu teksturnih koordinata u koje se preslikava teme trougla
- Indeksi se navode redosledom obilaska temena koji određuje normalu na površ po pravilu desne zavojnice
- Svaka stranica mreže se predstavlja sa dva trougla (šest temena)
- Jedan trougao predstavlja spoljnu, a drugi unutrašnju površ stranice
- Jedan trougao u mreži se predstavlja sa 12 indeksa:
  - 2 trougla  $\times$  3 temena  $\times$  2 indeksa



# Primer – mreža kvadrata (1)

- Kvadrat je zadat temenima: i teksturom:



$v_0 = (-10, -10, 0)$	$t_0 = (0, 0)$	$f_{0_{\text{lice}}} = (0, 0, 1, 1, 3, 3)$ - žuta
$v_1 = (-10, 10, 0)$	$t_1 = (0, 1)$	$f_{0_{\text{naličje}}} = (0, 0, 3, 3, 1, 1)$ - žuta
$v_2 = (10, 10, 0)$	$t_2 = (1, 1)$	$f_{1_{\text{lice}}} = (1, 1, 2, 2, 3, 3)$ - plava
$v_3 = (10, -10, 0)$	$t_3 = (1, 0)$	$f_{1_{\text{naličje}}} = (1, 1, 3, 3, 2, 2)$ - plava

**Koordinate temena**

**Koordinate teksture**

**Stranice**

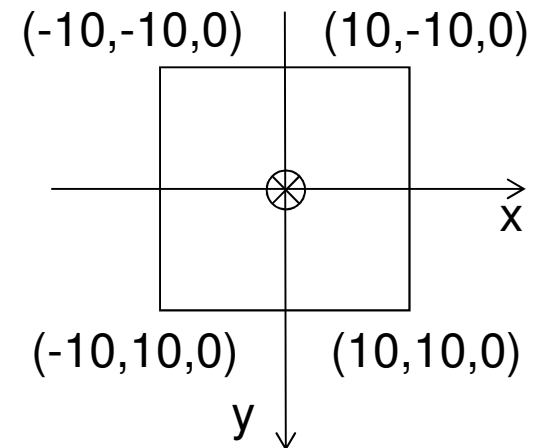
## Primer – mreža kvadrata (2)

```
...  
import javafx.scene.shape.MeshView;  
import javafx.scene.shape.TriangleMesh;  
import javafx.scene.paint.PhongMaterial;  
import javafx.scene.image.Image;  
  
public class Mreza extends Application {  
    @Override public void start(Stage prozor) {  
        MeshView kvadrat = napraviKvadrat();  
        // kod za postavljanje kamere, scene, prozora  
    }  
}
```



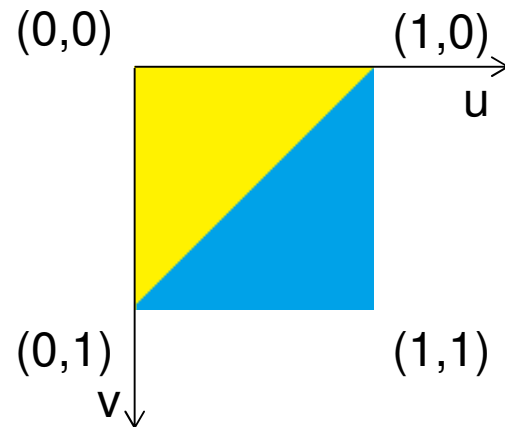
## Primer – mreža kvadrata (3)

```
public MeshView napraviKvadrat() {  
    // Temena: za svako se daju X, Y i Z koordinata  
    float[] temena = {  
        -10f, -10f, 0f, // v0  
        -10f, 10f, 0f, // v1  
        10f, 10f, 0f, // v2  
        10f, -10f, 0f // v3  
    };  
};
```



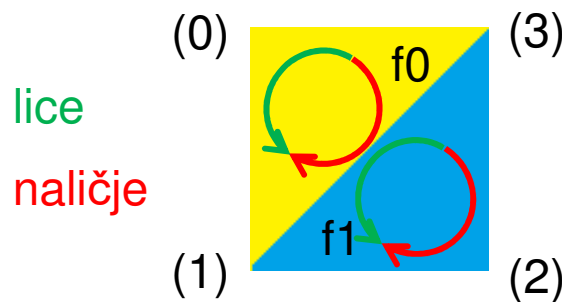
# Primer – mreža kvadrata (4)

```
// Teksturane koordinate  
float[] teksturneKoordinate = {  
    0.0f, 0.0f, // t0  
    0.0f, 1.0f, // t1  
    1.0f, 1.0f, // t2  
    1.0f, 0.0f // t3  
};
```



## Primer – mreža kvadrata (5)

```
// Stranice: po 2 indeksa za svako teme,  
// 3 temena za trougao,  
// 2 trougla za stranicu (lice i naličje)  
int[] stranice = {  
    0, 0, 1, 1, 3, 3, // iv0,it0,iv1,it1,iv3,it3 (f0 lice)  
    0, 0, 3, 3, 1, 1, // iv0,it0,iv3,it3,iv1,it1 (f0 naličje)  
    1, 1, 2, 2, 3, 3, // iv1,it1,iv2,it2,iv3,it3 (f1 lice)  
    1, 1, 3, 3, 2, 2 // iv1,it1,iv3,it3,iv2,it2 (f1 naličje)  
};
```



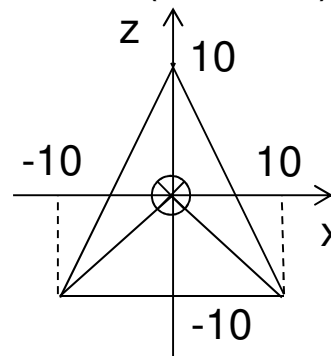
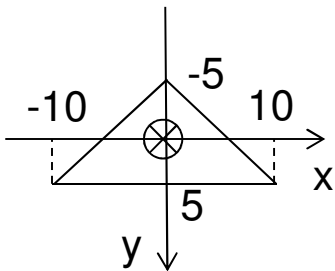
## Primer – mreža kvadrata (6)

```
TriangleMesh mreža = new TriangleMesh();
mreža.getPoints().addAll(temena);
mreža.getTexCoords().addAll(teksturneKoordinate);
mreža.getFaces().addAll(stranice);
MeshView kvadrat = new MeshView();
kvadrat.setMesh(mreža);

PhongMaterial mat = new PhongMaterial();
mat.setDiffuseMap(new Image("ZutoPlaviKvadrat.png"));
kvadrat.setMaterial(mat);
return kvadrat;
}
```

# Primer – mreža tetraedra (1)

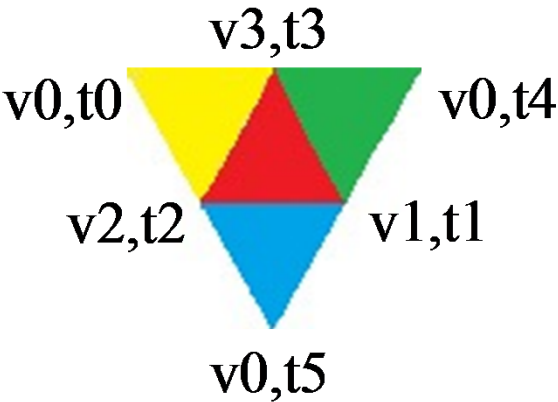

- Opis tetraedra:
  - osnovica (crvena) u ravni  $y=5$
  - vrh na Y-osi u tački  $y=-5$
  - prednja (plava) stranica  
ivica na osnovici paralelna X-osi,  $x \in [-10,10]$ ,  $z=-10$ ,  $y=5$
  - bočna desna (zelena) i bočna leva (žuta) stranica  
imaju zajedničko teme u tački  $(0, 5, 10)$



mreža

# Primer – mreža tetraedra (2)

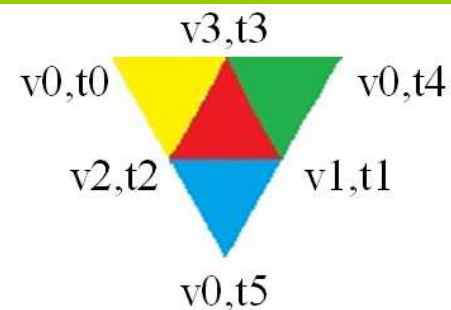
- Temena i tekstura

 <p>A diagram of a tetrahedron mesh. The vertices are labeled as follows: <math>v_0, t_0</math> (top-left), <math>v_3, t_3</math> (top-right), <math>v_0, t_4</math> (top-right), <math>v_2, t_2</math> (middle-left), <math>v_1, t_1</math> (middle-right), and <math>v_0, t_5</math> (bottom). The faces are colored: yellow (top-left), green (top-right), red (top), and blue (bottom).</p>	 <p>A diagram showing an alternative texture for the tetrahedron mesh. The texture is a square divided into four colored regions: yellow (top-left), green (top-right), red (top), and blue (bottom).</p>
Teksturirana mreža	Alternativna tekstura



# Primer – mreža tetraedra (3)

- Koordinate temena i teksture
- Stranice
  - indeksi temena i tekstura



$v_0 = ( 0, -5, 0)$   
 $v_1 = ( 10, 5, -10)$   
 $v_2 = (-10, 5, -10)$   
 $v_3 = ( 0, 5, 10)$

$t_0 = (0.00, 0.00)$   
 $t_1 = (0.75, 0.50)$   
 $t_2 = (0.25, 0.50)$   
 $t_3 = (0.50, 0.00)$   
 $t_4 = (1.00, 0.00)$   
 $t_5 = (0.50, 1.00)$

$f_{0_{\text{lice}}} = (0, 5, 2, 2, 1, 1)$  - plava  
 $f_{0_{\text{naličje}}} = (0, 5, 1, 1, 2, 2)$  - plava  
 $f_{1_{\text{lice}}} = (1, 1, 3, 3, 0, 4)$  - zelena  
 $f_{1_{\text{naličje}}} = (1, 1, 0, 4, 3, 3)$  - zelena  
 $f_{2_{\text{lice}}} = (0, 0, 3, 3, 2, 2)$  - žuta  
 $f_{2_{\text{naličje}}} = (0, 0, 2, 2, 3, 3)$  - žuta  
 $f_{3_{\text{lice}}} = (1, 1, 2, 2, 3, 3)$  - crvena  
 $f_{3_{\text{naličje}}} = (1, 1, 3, 3, 2, 2)$  - crvena

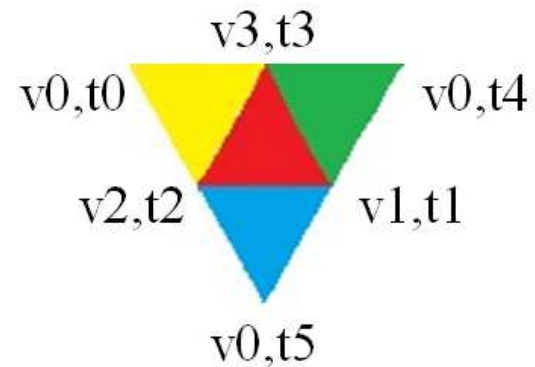
**Koordinate temena**

**Koordinate teksture**

**Stranice tetraedra**

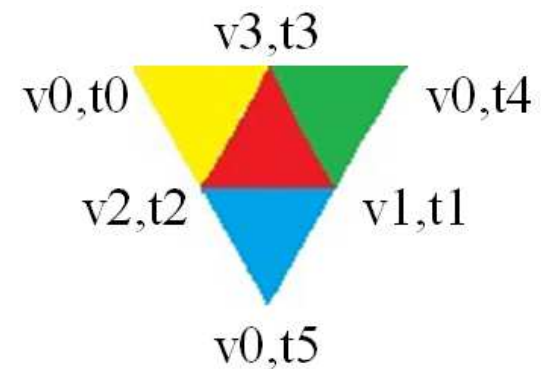
## Primer – mreža tetraedra (4)

```
import javafx.scene.shape.MeshView;  
import javafx.scene.shape.TriangleMesh;  
import javafx.scene.paint.PhongMaterial;  
...  
public MeshView napraviTetraedar() {  
    // Koordinate temena: za svako teme se daje x, y, z  
    float[] temena = {  
        0f, -5f, 0f, // v0  
        10f, 5f, -10f, // v1  
        -10f, 5f, -10f, // v2  
        0f, 5f, 10f // v3  
    };  
};
```



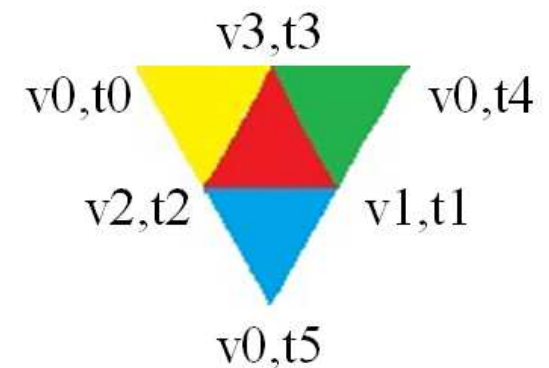
## Primer – mreža tetraedra (5)

```
// Teksturane koord.: U, V koord. karakterističnih tačaka
float[] teksturneKoordinate = {
    0.00f, 0.0f, // t0
    0.75f, 0.5f, // t1
    0.25f, 0.5f, // t2
    0.50f, 0.0f, // t3
    1.00f, 0.0f, // t4
    0.50f, 1.0f, // t5
};
```



## Primer – mreža tetraedra (6)

```
// Stranice: par indeksa za svako teme, 3 temena za trougao,  
//           2 trougla za stranicu (lice i naličje)  
int[] stranice = {  
    0, 5, 2, 2, 1, 1, // f0 lice  
    0, 5, 1, 1, 2, 2, // f0 naličje  
    1, 1, 3, 3, 0, 4, // f1 lice  
    1, 1, 0, 4, 3, 3, // f1 naličje  
    0, 0, 3, 3, 2, 2, // f2 lice  
    0, 0, 2, 2, 3, 3, // f2 naličje  
    1, 1, 2, 2, 3, 3, // f3 lice  
    1, 1, 3, 3, 2, 2, // f3 naličje  
};
```




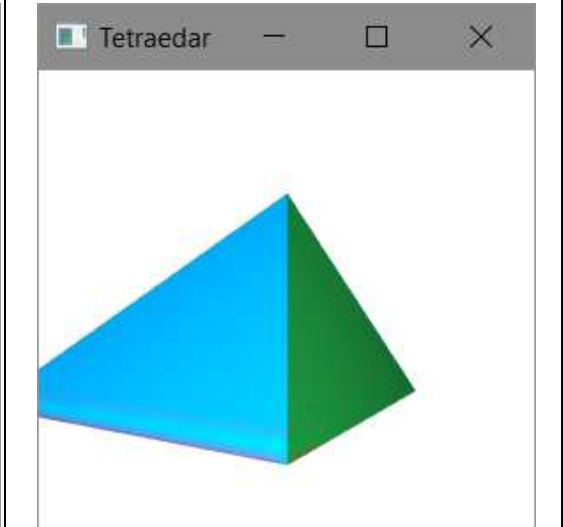
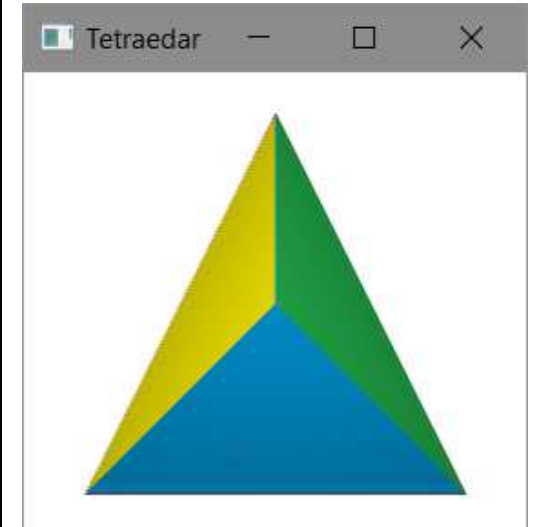
## Primer – mreža tetraedra (7)

```
TriangleMesh mreža = new TriangleMesh();
mreža.getPoints().addAll(temena);
mreža.getTexCoords().addAll(teksturneKoordinate);
mreža.getFaces().addAll(stranice);
MeshView tetraedar = new MeshView();
tetraedar.setMesh(mreža);

PhongMaterial mat = new PhongMaterial();
mat.setDiffuseMap(new Image("TeksturaTetraedar.jpg"));
tetraedar.setMaterial(mat);
return tetraedar;
}
```

# Primer – mreža tetraedra (8)



bez rotacije	<pre>tetraedar.setRotationAxis (Rotate.Y_AXIS); tetraedar.setRotate(45);</pre>	<pre>tetraedar.setRotationAxis (Rotate.X_AXIS); tetraedar.setRotate(90);</pre>
		

## Primer – mreža tetraedra (9)

- Druga (jednostavnija) mogućnost - skala boja
  - za slučaj da su stranice monolitno obojene



```
float[] teksturneKoordinate = {  
    0.1f, 0.5f, // 0 crvena  
    0.3f, 0.5f, // 1 zelena  
    0.5f, 0.5f, // 2 plava  
    0.7f, 0.5f // 3 žuta  
};
```

# Primer – mreža tetraedra (10)

- Stranice:

```
int[] stranice = {  
    0, 2, 2, 2, 1, 2, // f0 lice  
    0, 2, 1, 2, 2, 2, // f0 naličje  
    1, 1, 3, 1, 0, 1, // f1 lice  
    1, 1, 0, 1, 3, 1, // f1 naličje  
    0, 3, 3, 3, 2, 3, // f2 lice  
    0, 3, 2, 3, 3, 3, // f2 naličje  
    1, 0, 2, 0, 3, 0, // f3 lice  
    1, 0, 3, 0, 2, 0, // f3 naličje  
};
```



# Primer – mreža tetraedra (11)



bez rotacije	<code>tetraedar.setRotationAxis(Rotate.Y_AXIS); tetraedar.setRotate(45);</code>	<code>tetraedar.setRotationAxis(Rotate.X_AXIS); tetraedar.setRotate(90);</code>
