


Računarska grafika

JavaFX – 3D izbor



Uvod

- Da bi se interagovalo sa nekim objektom u sceni i menjao mu se:
 - položaj
 - orijentacija
 - svojstva materijala
 - način prikaza (žični model/neproziran objekat, lica/naličja)potrebno je najpre da se objekat izabere
- Izbor se po pravilu vrši mišem
 - kurzor se dovede iznad objekta koji treba da se izabere
 - pritisne se dugme miša za izbor
- Za izbor objekta se koristi pik-korelacija (*pick-correlation*)
 - proverava da li se kurzor nalazi unutar prikazane projekcije objekta

Izbor objekta u 3D sceni

- Osnovna ideja:
 - klikom miša dok je kurzor na nekoj poziciji u prikazu scene bira se objekat čija projekcija obuhvata poziciju kurzora
 - ako više projekcija objekata obuhvata poziciju kurzora bira se objekat koji je nablži posmatraču
- Problemi izbora: intuitivnost i određenost
 - ako je prikazan žični model, izbor bliskog objekta nije intuitivan kad je kliknuto na objekat u pozadini, vidljiv kroz mrežu bliskog objekta
 - ako je objekat složen, koji nivo objekta u kompoziciji delova se bira: najniži (najbliži listu u stablu) ili neki viši?
 - primer složenog objekta: robot-ruka-šaka-prst-članak; klik na šaku; da li se time bira šaka, ili ruka ili robot?

Izbor objekta u JavaFX

- Izbor (selekcija) oblika u 3D sceni
- Obrađuje se događaj miša prouzrokovan „klikom na oblik“
 - parametar rukovaoca (ili lambda izraza) je referenca:
`MouseEvent dog`
- Pozove se metod: `dog.getPickResult()`
- Metod vraća rezultat tipa `javafx.scene.input.PickResult`
- Klasa `PickResult` predstavlja apstrakciju rezultata pik-korelacije (za određivanje oblika u 3D sceni na koji je pokazivao kurzor)
- Ukoliko pokazana tačka pripada projekcijama više oblika:
 - rezultat `PickResult rez` će odgovarati najbližem obliku kameri

Dohvatanje izabranog objekta

- Za objekat rezultata (klase `PickResult`) pozove se metod `rez.getIntersectedNode()`
- Metod vraća referencu na izabrani čvor (klase `Node`) grafa scene
- Ako nije izabran neki čvor `getIntersectedNode()` vraća `null`
- Ukoliko su u sceni samo objekti klase `Shape3D`
 - konverzijom naniže može da se konvertuje rezultat tipa `Node` u referencu na objekat tipa `Shape3D`
 - nakon toga sa datim objektom može nešto da se radi kao sa 3D oblikom

Metodi klase `PickResult` (1)

- `Point3D getIntersectedPoint()`
 - vraća tačku preseka u lokalnom koord. sistemu izabranog oblika
 - tačka preseka se određuje tako što projekcioni zrak prolazi kroz poziciju kamere i dodirnutu tačku u prikaznoj ravni
 - tačka preseka je tačka u kojoj zrak prodire stranicu mreže oblika
- `double getIntersectedDistance()`
 - vraća rastojanje između kamere i tačke preseka na površi oblika
- `int getIntersectedFace()`
 - vraća indeks stranice izabranog oblika, kojoj pripada tačka preseka
 - ako oblik nema korisnički definisane stranice ili ako je oblik dodirnut na ivici, vraća se vrednost `FACE_UNDEFINED`

Metodi klase `PickResult` (2)

- `Point3D getIntersectedNormal()`
 - vraća normalu na površ u tački preseka
- `Point2D getIntersectedTexCoord()`
 - vraća teksturne koordinate u tački preseka izabranog objekta

Primer (1)

```
import javafx.scene.shape.Shape3D;
import javafx.scene.paint.Material;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.paint.Color;
import javafx.scene.input.PickResult;
...
Group koren = new Group(valjak, kvadar, lopta, svetlo);
...
enum Izbor {KVADAR, VALJAK, LOPTA};
private Shape3D telo;
private Material matStari;
...
```


Primer (2)

```
EventHandler<MouseEvent> r1 = dog -> { // obrada pritiska
    PhongMaterial mat = new PhongMaterial();
    mat.setDiffuseColor(Color.CHARTREUSE);
    Izbor izbor=null;
    PickResult rez = dog.getPickResult();
    if (rez!=null) telo = (Shape3D) rez.getIntersectedNode();
}
```

Primer (3)

```
if(telo != null){
    if (telo instanceof Box) izbor = Izbor.KVADAR;
    else if (telo instanceof Cylinder) izbor = Izbor.VALJAK;
    else if (telo instanceof Sphere) izbor = Izbor.LOPTA;
    System.out.println("Izabrano telo je: " + izbor);
    matStari = telo.getMaterial();
    telo.setMaterial(mat);
}
}; // obrada događaja pritiska miša
scena.addEventHandler(MouseEvent.MOUSE_PRESSED, r1);
```

Primer (4)

```
EventHandler<MouseEvent> r2 = dog -> { // obrada puštanja
    if (telo!=null) telo.setMaterial(matStari);
}; // obrada događaja puštanja miša
scena.addEventHandler(MouseEvent.MOUSE_RELEASED, r2);
```

Rezultat izvršenja

