

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ljiljana Pintarić

Mario Paravac

JAVAFX

SEMINARSKI RAD IZ KOLEGIJA RAČUNALNA GRAFIKA

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ljiljana Pintarić
Mario Paravac

JAVAFX
SEMINARSKI RAD

Mentor:

doc.dr.sc. Ivan Hip

Varaždin, siječanj 2015.

Sadržaj

1.	Uvod.....	1
2.	Teoretski uvod.....	2
2.1.	3D oblici i tijela.....	2
2.2.	Kamera.....	2
2.3.	Svjetla.....	3
2.4.	Materijal.....	3
2.5.	Organizacija čvorova i grupa.....	3
3.	Instalacija i pokretanje aplikacije.....	5
5.1.	Instalacija.....	5
5.2.	Pokretanje aplikacija.....	5
5.3.	Pokretanje primjera.....	5
5.3.1.	Kocka.....	6
5.3.2.	Piramida.....	7
5.3.3.	Vjetromjer.....	9
5.3.4.	Slovo F.....	11
5.3.5.	Videokocka.....	12
5.3.6.	Test dubine(Z-buffer).....	12
4.	Zaključak.....	13
5.	Literatura.....	14

1. Uvod

Tema ovog rada je JavaFX i njene mogućnosti vezane uz računalnu grafiku. U radu smo se koncentrirali na izradu primjera poput onih na vježbama i iz zadaća, te još nekih koji prikazuju dodatne mogućnosti JavaFX. Kod sadrži dodatne komentare koji nisu navedeni u seminaru budući da njihovo značenje ima smisla samo u kontekstu koda.

Od primjera odlučili smo napraviti neke jednostavnije primjere poput kocke, piramide i sličnih, te neku integraciju svega toga kao što su vjetromjer i slovo F. Ovi primjeri pokazuju rad sa predefiniranim oblicima, svjetlom, kamerom, materijalima, videom i zvukom i drugim zanimljivim stvarima koje nude biblioteke JavaFX.

2. Teoretski uvod

JavaFX 3D graphics API osigurava trodimenzionalnu grafičku biblioteku za JavaFX platformu. Omogućava korištenje 3D geometrije, kamera i svjetla za kreiranje, prikaz i manipulaciju objektima u trodimenzionalnom prostoru.

2.1. 3D oblici i tijela

JavaFX sadrži SHAPE3D API dostupan u JavaFX 3d Graphics biblioteci. POstoje dva tipa oblika: predefimirani oblici i korisnički definirani. Predefimirani oblici su kocke odnosno "kutije", valjci i sfere.

U nastavku vidimo Shape3D hijerarhiju klasa:

```
java.lang.Object
  javafx.scene.Node
    javafx.scene.shape.Shape3D
      javafx.scene.shape.MeshView
      javafx.scene.shape.Box
      javafx.scene.shape.Cylinder
      javafx.scene.shape.Sphere
```

Korisnički definirani oblici koriste se TriangleMesh podklasom, koja je najtipičnija vrsta "mreže" koja se koristi u 3D layoutima. U nastavku vidimo JavaFX Mesh hijerarhiju klasa.

```
java.lang.Object
  javafx.scene.shape.Mesh (abstract)
    javafx.scene.shape.TriangleMesh
```

TriangleMesh sadrži odvojena polja točaka, koordinata tekstura i naličja koja opisuju mrežu geometrijskog tijela.

2.2. Kamera

Kamera je čvor koji je moguće dodati JavaFX scenskom grafu, te tako omogućuje pokretanje kamere u 3D layoutu. U JavaFX koordinatnom prostoru scene, kamera je pozicionirana na Z=0, X-os usmjerena je desno, Y-os usmjerena je prema dolje, a Z-os u dubinu ekrana. Budući da je kamera sada čvor, moguće je nad njom primjenjivati transformacije kao i na drugim čvorovima: rotaciju, translaciju itd.

2.3. Svjetla

Dakle, svjetla su također jedan od čvorova u scenskom grafu. Ukoliko nema posebno definiranih svjetala u aplikaciji, osigurano je neko "defaultno" svjetlo. Svako svjetlo odnosi se na neki skup čvorova, a ako je taj skup prazan, onda se svjetlo primjenjuje na sve čvorove mreže. Također, ukoliko je korijenski čvor u skupu čvorova tada se na sve druge primjenjuje svjetlo.

Postoje dvije vrste svjetla:

- **Ambijentalno** svjetlo koje dolazi iz svih smjerova
- **Usmjereno** svjetlo koje se nalazi na nekoj fiksnoj točki u prostoru i isijava svjetlo jednakomjerno u svom smjerovima

Hijerarhija klase svjetala:

```
javafx.scene.Node
    javafx.scene.LightBase (abstract)
```

2.4. Materijal

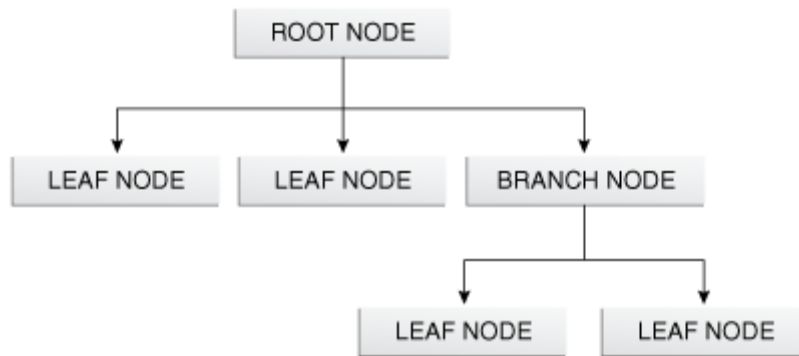
U nastavku vidimo hijerarhiju klase materijala:

```
java.lang.Object
    javafx.scene.paint.Material (abstract)
        javafx.scene.paint.PhongMaterial
```

PhongMaterial klasa osigurava definicije svojstava koja reprezentiraju formu Phong materijala: diffuse color, diffuse map, specular map, specular color, specular power, bump map or normal map, self-illumination map. Jednom kada definiramo materijal moguće ga je pridijeliti bilo kojem obliku na sceni.

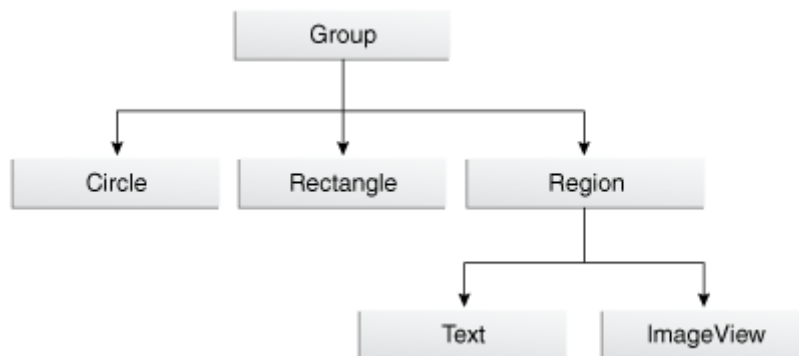
2.5. Organizacija čvorova i grupa

Na idućoj slici vidi se primjer organizacije čvorova odnosno grupa. Kao što vidimo, čvorovi su organizirani u stablo. Kao i svako stablo, postoji jedan čvor koji je korijen, kojem je zatim moguće dodijeliti druge čvorove. Čvorovi koji nemaju djece su listovi.



Slika 1, Čvorovi i grupe, Izvor: <http://docs.oracle.com/javafx/2/scenegraph/img/figure1.png>

Na idućoj slici vidimo konkretan primjer – u korijenu je konkretna grupa kojoj su zatim dodijeljeni drugi čvorovi: krug, pravokutnik te još jedna grupa, koja pak sadrži tekst i ImageView. Prednost ovakve organizacije je, osim same preglednosti, da se sve transformacije primijenjene nad roditeljskim čvorom odnose i na djecu čvorove.



Slika 2, Primjer organizacije čvorova, Izvor: <http://docs.oracle.com/javafx/2/scenegraph/img/figure2.png>

3. Instalacija i pokretanje aplikacije

3.1. Instalacija

Za instalaciju potrebno je skinuti Java Development Kit (JDK) 8 sa <http://www.oracle.com/technetwork/java/javase/downloads/> . (JDK 8 uključuje JavaFX API-je koji uključuju 3D grafičke funkcionalnosti)

Također je potrebno instalirati NetBeans IDE dostupan na <https://netbeans.org/downloads/> .

3.2. Pokretanje aplikacija

Nakon pokretanja, sve što je potrebno za otvaranje projekta je odabrati Open sa izbornika File te navigirati do željenog projekta.

3.3. Pokretanje primjera

Za promjenu datoteka koje se izvršavaju prve, potrebno je u Properties izborniku pod Run odabrati željenu klasu.

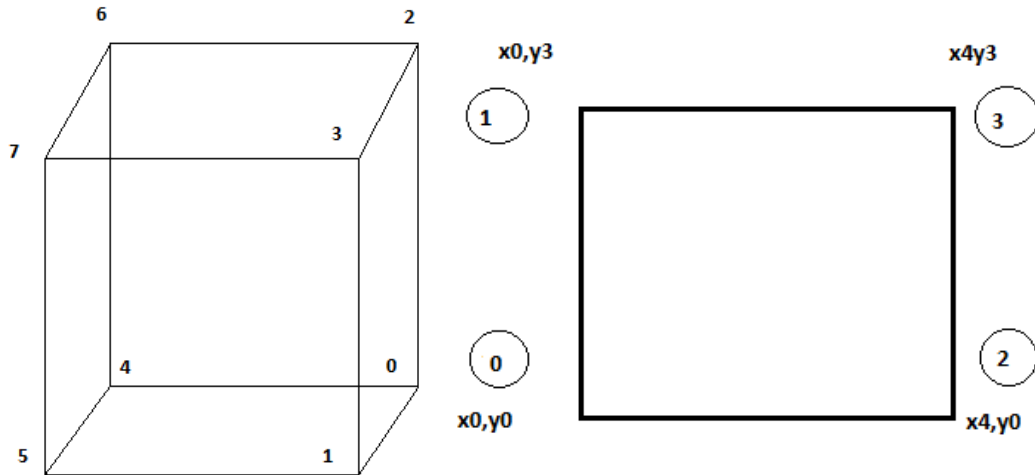
Sljedeća tri primjera (kocka, piramida, vjetromjer) imaju zajedničke odrednice vezano uz kameru i tipke koje je moguće koristiti u primjeru. Primjer koristi perspektivnu kameru. Budući da je kamera čvor, nad njom se mogu provoditi različite transformacije. Primjeri imaju tri osi različitih boja: os x je crvene, os y zelene i os z narančaste boje. Tipke kojima je moguće upravljati kamerom su sljedeće:

```
ALT + SHIFT + PRIMARY_BUTTON - rotacija po z osi
ALT + PRIMARY_BUTTON - rotacija po x i y osi
ALT + SECONDARY_BUTTON - zoomIn/zoomOut
ALT + MIDDLE_BUTTON - premještanje kocke po sceni
F - centriranje, zumiranje i postavljanje na početne postavke
A - postavljanje na početne postavke
```

Primjeri također sadrže i svjetlo, koje je također zaseban čvor kojim je također moguće upravljati.

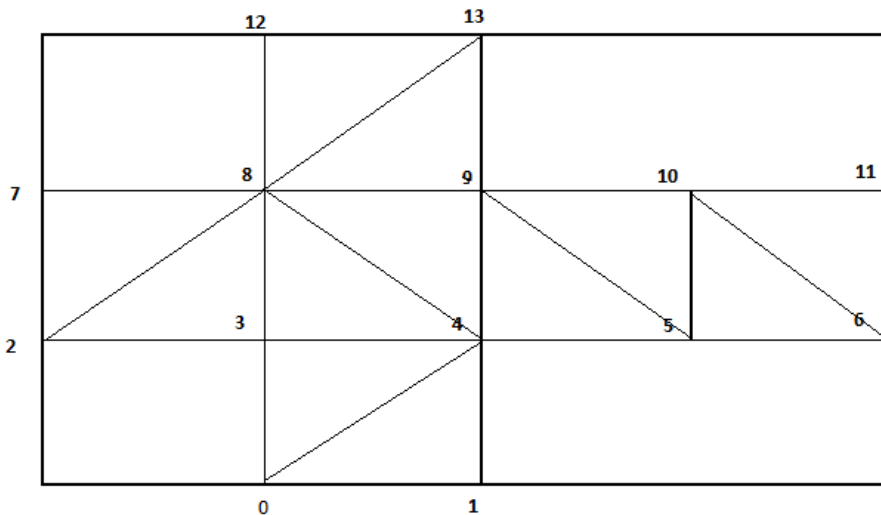
3.3.1. Kocka

Ovaj primjer koristi TriangleMesh za crtanje kocke te konkretnu teksturu čije koordinate sadrži. Alternativa je crtanje kocke pomoću već gotove klase Box, no u tom slučaju je nemoguće upravljati lijepljenjem teksture. Tekstura je nalijepljena na sljedeći način:



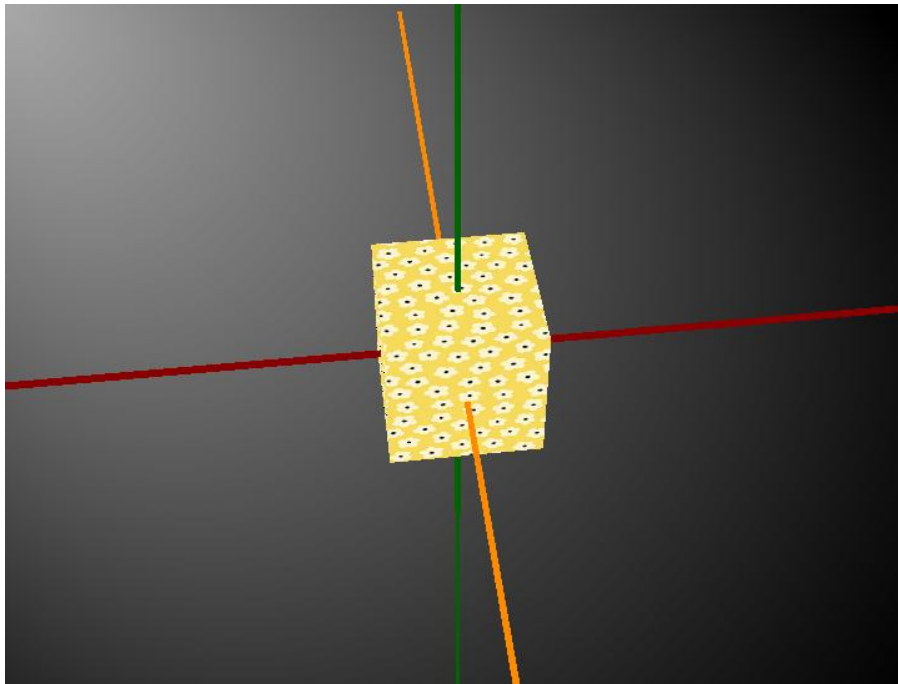
Slika 3. Vrhovi kocke

Slika 4. Vrhovi kocke(1. slučaj) i koordinate teksture



Slika 5. Rezanje teksture

Kao što vidimo, iz teksture je izrezana određena površina i nalijepljena na kocku. Trebamo uzeti u obzir da se u javiFX koristi trokut kao osnovni lik za crtanje (triangle mesh) pa tako i za određivanje strukture koordinata. Rezultat je sljedeća kocka:



Slika 6. Kocka

Također, postoji alternativa – moguće je i ne izrezivati teksturu nego samo naljepiti cijelu teksturu na stranicu kocke. U tom slučaju bismo dobili malo manje cvjetiće. Ukoliko odaberemo raditi kocku već predefiniranim oblikom Box, tada nemamo izbora što se tiče koordinata tekstura – automatski se koristi ovaj drugi pristup lijepljenju tekstura.

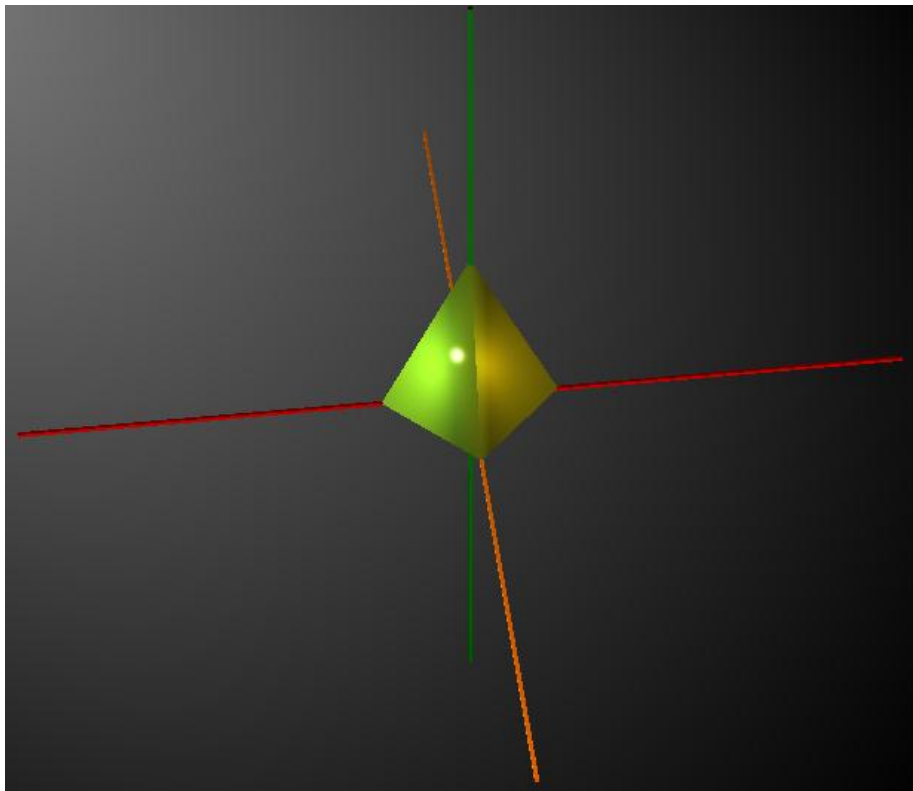
3.3.2. Piramida

Za crtanje piramide se također koristi TriangleMesh, no nisu definirane koordinate tekstura (odnosno postavljene su na 0). U ovom primjeru, kao i u prošlom, svjetlo je statički čvor sa ambijentalnim svjetlom te svjetlom usmjerenim na točno određene točke. Konkretno, koristili smo sljedeća svjetla:

```
Group lightGroup = new Group();
    //lijevo svijetlo
PointLight light = new PointLight();
light.setColor(Color.WHITE);
light.setTranslateX(-600);
light.setTranslateY(200);
light.setTranslateZ(-600);
lightGroup.getChildren().add(light);
    //desno svijetlo
```

```
PointLight light2 = new PointLight();
light2.setColor(Color.DARKORANGE);
light2.setTranslateX(600);
light2.setTranslateY(200);
light2.setTranslateZ(-600);
lightGroup.getChildren().add(light2);
AmbientLight ambientLight = new AmbientLight(Color.rgb(20, 20, 20));
```

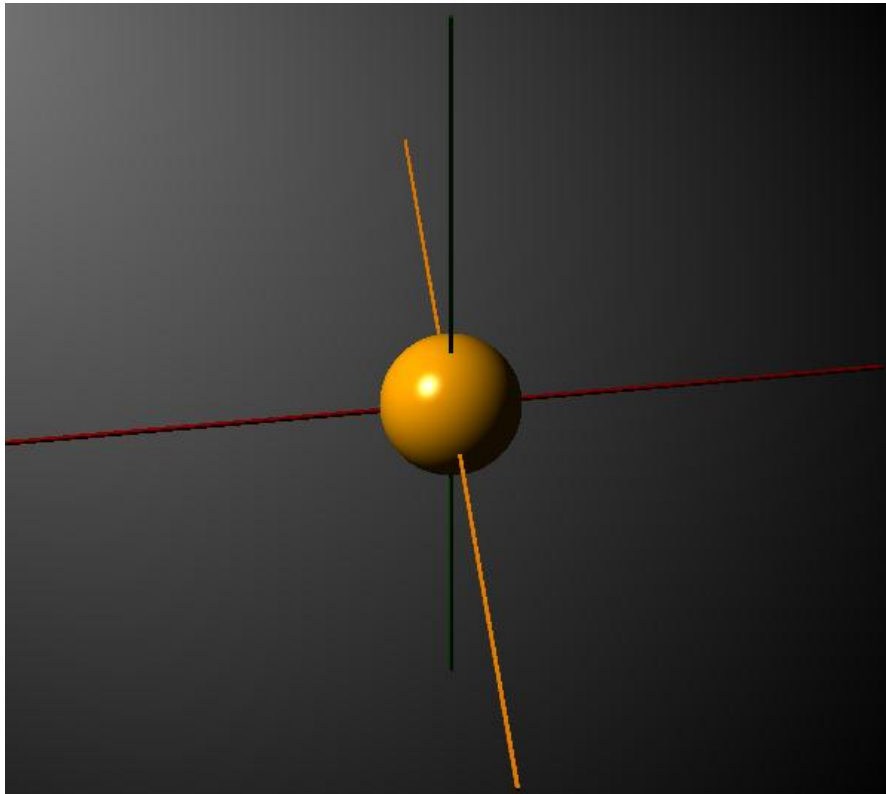
Kao što vidimo, na vrhu je čvor koji sadrži svjetla. Nakon toga definirana su dva izvora usmjerenog svjetla („PointLight“), kojima su dodijeljene boje, te su translirana tako da optimalno osvijetle piramidu. Također je definirano i ambijentalno svjetlo, koje je neke srednje tamno sive boje, jer se u suprotnom svjetlo previše ili premalo odbija od površine.



Slika 7. Piramida

3.3.3. Kugla

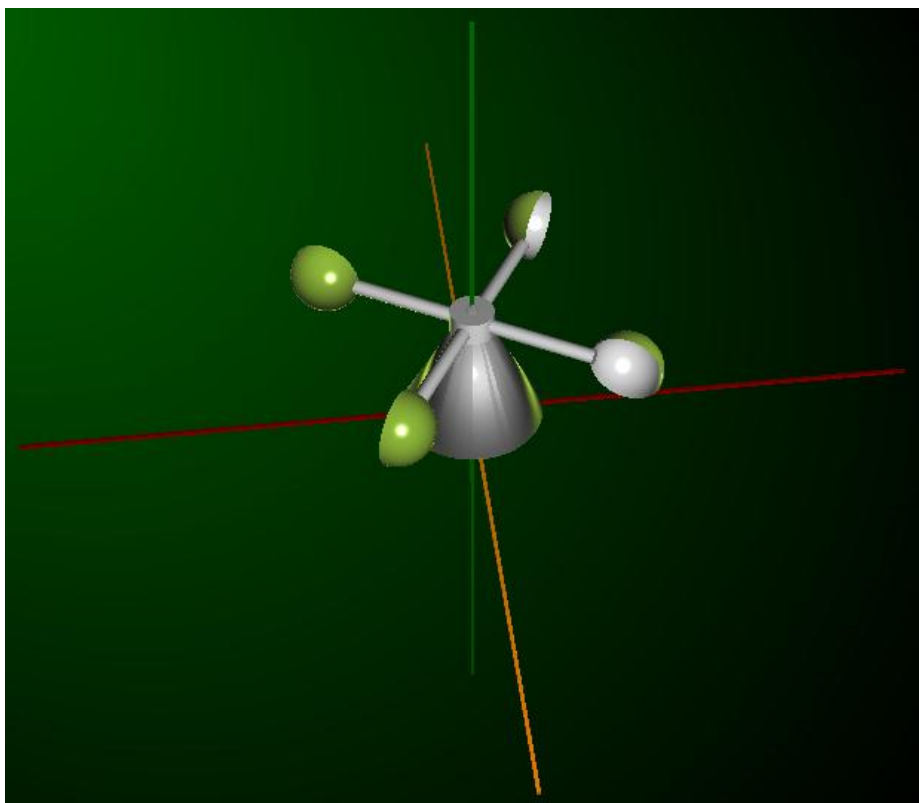
Na primjeru kugle možemo vidjeti gašenje i paljenje svjetla. Pritiskom na tipku L svjetlo se gasi ili pali, ovisno o trenutnom stanju. Pritom ambijentalno svjetlo ostaje upaljeno, dok gasimo samo usmjereno svjetlo (tzv. „PointLight“).



Slika 8. Kugla

3.3.4. Vjetromjer

Vjetromjer je kompozicija od nekoliko elemenata – stošca, valjka, i polukugle. Također je nad njima primijenjena animacija rotacije. Nad čvorom svjetla primijenjena je animacija rotacije, pa se svjetlo vrti oko vjetromjera. U ovaj je primjer također dodan i zvuk vjetra, kako bi efekt bio potpun. Za ovaj se primjer koriste dodatne klase: Cone za crtanje stošca, te HalfSpheroid za crtanje polusfere.



Slika 9. Vjetromjer

U ovom je primjeru naglasak stavljen na rotaciju. Osim što su određeni elementi fiksno rotirani kako bi se dobila odgovarajuća slika, također je i dodana animacija rotacije koja služi za rotiranje elemenata čvorova nad kojima je primijenjena. U nastavku vidimo funkciju za rotaciju. Prosljeđujemo joj grupu elemenata, instanciramo klasu `RotateTransition` koja će nam omogućiti da odredimo karakteristike animacije, odredimo joj trajanje (iako ono kasnije gubi svoj smisao budući da se animacija vrti u beskonačnost), te dodamo grupu. Nakon toga odredimo kut rotacije (0-360 stupnjeva), os po kojoj želimo rotirati te dodatne postavke kao što su to vrijeme ponavljanja rotacije (`INDEFINITE`) te interpolator (`LINEAR`). Moguće je postaviti i nešto drugo, primjerice da se na početku onih 4.5 sekundi animacija vrti sporije, zatim ubrza i na kraju opet uspori ili nešto slično.

```
private void rotateAroundZAxis(Group elementGroup) {  
    RotateTransition rotate =  
        new RotateTransition(Duration.seconds(4.5), elementGroup);  
  
    rotate.setFromAngle(0);  
    rotate.setToAngle(360);  
    rotate.setAxis(Rotate.Z_AXIS);  
}
```

```

rotate.setCycleCount(RotateTransition.INDEFINITE);
rotate.setInterpolator(Interpolator.LINEAR);
rotate.play();
}

```

Nakon toga, definirana su svjetla:

```

PointLight light = new PointLight();
light.setColor(Color.WHITE);
Group lightGroup = new Group();
light.setTranslateX(-50);
light.setTranslateY(-100);
light.setTranslateZ(-800);
lightGroup.getChildren().add(light);
PointLight light2 = new PointLight();
light2.setColor(Color.YELLOWGREEN);
light2.setTranslateX(50);
light2.setTranslateY(200);
light2.setTranslateZ(400);
lightGroup.getChildren().add(light2);
AmbientLight ambientLight = new AmbientLight(Color.rgb(20, 20,
20));
rotateAroundZAxis(lightGroup);

```

te je nad njima primijenjena rotacija. Zbog toga se svjetla pomiču. Rotacija je također primijenjena nad grupom valjaka i grupom sfera koje se rotiraju oko osi z.

U animaciju je također dodan i zvuk vjetra, sve da bi se dobilo na efektu pravog vjetromjera.

3.3.5. Slovo F

Slovo F je sastavljeno od nekoliko kocki. Na nekim kockama prikazani su različiti primjeri teksture .Primjer generira dvije gotovo identične scene. Druga scena se razlikuje po uključenom antialiasingu i dodanim video klipom . Funkcionalnosti primjera:

- Tipka „A“ - određuje način crtanja(žičano,punjeno)
- „S“ - rotacija kocki on/off
- „D“ - promjena scene

- „P“ - pokreće/pauzira video u drugoj sceni
- „Q“ - pokreće animaciju
- „W“ - pauzira animaciju
- „B“ - prikazuje teksturu bumpMap
- „-“ i „+“ – približavanje i udaljavanje kamere
- primary button + mouse dragged – upravljanje kamerom

3.3.6. Videokocka

Primjer prikazuje kocku koja na svakoj stranici prikazuje različiti video klip. Sastavljena od više media playera nad kojima su primjenjene transformacije translacije, rotacije i skaliranja te je primjenjena animacija rotacije glavnog čvora. Pritiskom tipke „A“ razdvajaju se stranice kocke. Moguće je upravljati kamerom: primary button + mouse dragged.

3.3.7. Test dubine(Z-buffer)

Primjer uključuje 3 kocke, crvena (prednja), zelena (srednja) i plava (stražnja).

Ukoliko je depth test isključen uočavamo da je poredak kocki drugačiji i prilikom rotacije zadržavaju svoju poziciju (npr. plava uvijek ostaje na prednjoj poziciji). Razlog tome je redosljed renderiranja. Depth buffer koristi se za sortiranje objekta na temelju njihove z vrijednosti. Sortiranje se vrši prema pikselima od strane GPU-a. Bez obzira kojim se redosljedom objekti renderiraju, onaj objekt čija je z vrijednost najbliža gledatelju biti će uvijek prikazan na vrhu.

4. Zaključak

Na kraju možemo reći da je JavaFX vrlo organizirana i user – friendly, te smo uživali u njenom korištenju. Također, Oracle nudi pristojnu dokumentaciju sa primjerima, te je moguće naći rješenja za sve probleme koji bi se mogli pojaviti početnicima u JavaFX poput nas.

Nažalost JavaFX nije sofisticirana kao OpenGL, te ne pokriva sve njegove funkcionalnosti, no to nadoknađuje jednostavnošću korištenja te kompaktnošću koda. Možemo je preporučiti za korištenje na kolegiju umjesto „obične“ jave.

5. Literatura

1. Oracle, JavaFX, <https://docs.oracle.com/javase/8/javafx/graphics-tutorial/javafx-3d-graphics.htm> , Dostupno 15. siječanj 2015.
2. Oracle, Animation and Other Effects, <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/animation.htm> , Dostupno 15. siječanj 2015.
3. Oracle, Transformation Types and Examples, <https://docs.oracle.com/javase/8/javafx/visual-effects-tutorial/transformation-types.htm#BABDEBGA> , Dostupno 15. siječanj 2015.
4. Oracle Learning Library, 3D Made Easy With JavaFX, <https://www.youtube.com/watch?v=EBKHdV-rIc> , Dostupno 15. siječanj 2015.
5. Sun Microsystems, Getting Started with Java 3D API, Creating Geometry, http://www2.hs-fulda.de/caelabor/inhalte/java/j3d/j3d_tutorials/j3d_tutorial_ch2.pdf , Dostupno 15. siječanj 2015.
6. Weiqi Gao, OCI, Experiments with JavaFX 3D Scenes, <http://sett.ociweb.com/sett/settMay2012.html> , Dostupno 15. siječanj 2015.
7. Narayan G. Maharjan, Java and FX, Lights in JavaFX 3D, <http://blog.ngopal.com.np/2013/06/14/lights-in-javafx-3d/> , Dostupno 15. siječanj 2015.