ELEKTROTEHNIČKI FAKULTET 22.04.2018.
Katedra za RTI

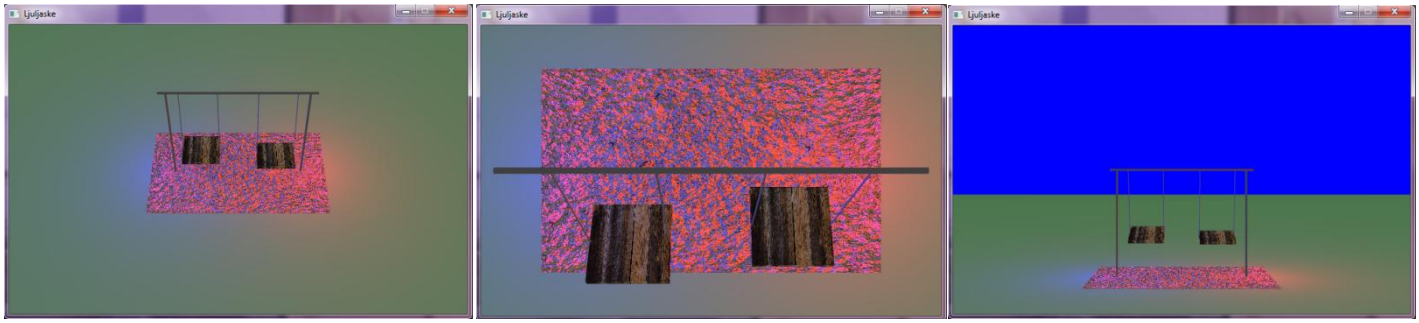
RAČUNARSKA GRAFIKA

drugi kolokvijum – praktični deo

1) [70] Napisati program koji koristi grafičku biblioteku JavaFX i prikazuje animirane ljuljaške. Pozadina je plave boje, dok se na sredini zelene podloge (Color.LIGHTGREEN), ispod ljuljaški, nalazi pesak. Postaviti peščanoj podlozi teksturu na osnovu priložene slike "sand.png" i reljef na osnovu priložene slike "sand_n.png". Dimenzije usvojiti da model približno odgovara slikama, pri čemu je visina stubova jednaka 3/4 dužine nosača, a visina šipki kojim su ljuljaške zakačene za nosač približno 1/2 dužine nosača. Nosač, stubovi i šipke su sive boje, boja odraza svetlosnog izvora je žuta, a refleksivnost materijala visoka. Otklon ljuljaški se kreće do 30 stepeni, pri čemu je kretanje sporije sa povećanjem ugla otklona. Postaviti sedištu ljuljaški teksturu na osnovu priložene slike "wood.jpg". Ambijentalno osvetljenje je sive boje (Color.LIGHTGRAY), a dva svetla tačkastog izvora, postavljena na oba kraja iznad peščane podloge, su plave i crvene boje. Pritiskom na tastere P i C se isključuje/uključuje plavi i crveni tačkasti izvor svetla, respektivno. Pokretanje i zaustavljanje ljuljaški vrši se klikom miša na sedište ljuljaške, ili tasterima 4 (leva ljuljaška) i 5 (desna ljuljaška). Koristi se projekcija sa perspektivom. Potrebno je obezbediti sledeće poglede:

- podrazumevani, kao na slici levo (taster 1);
- pogled odozgo, gde posmatrač gleda vertikalno naniže, kao na slici u sredini (taster 2);
- pogled spreda, kao na slici desno (taster 3).

Priložen je .jar fajl sa demonstracijom programa koji treba napisati.



<u>Napomene:</u>

1. Izrada praktičnog dela traje 100 minuta.
2. Rešenje zadatka se predaje u obliku NetBeans projekta u predviđenom folderu na računaru.
3. Dozvoljena je upotreba literature koja je stavljena na raspolaganje i pristup *Oracle* sajtu.
4. Nije dozvoljeno uz sebe imati mobilni telefon, bez obzira da li je uključen ili isključen.

Rešenje zadatka

drugi kolokvijum – praktični deo

1)

```
package RG_1718_k2;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.animation.*;
import javafx.event.EventHandler;
import javafx.scene.*;
import javafx.scene.image.Image;
import javafx.scene.input.*;
import javafx.scene.paint.Color;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.shape.*;
import javafx.scene.transform.*;
import javafx.util.Duration;

class Ljuljaska extends Group{
    private Timeline animacija;
    private Box sediste;
    Ljuljaska(PhongMaterial matSipke) {
        Cylinder sipka1 = new Cylinder(1, 190); sipka1.getTransforms().add(new Translate(-49, -5, 0));
        Cylinder sipka2 = new Cylinder(1, 190); sipka2.getTransforms().add(new Translate(49, -5, 0));
        sipka1.setMaterial(matSipke); sipka2.setMaterial(matSipke);
        sediste = new Box(100,10,100); sediste.setTranslateY(95);
        PhongMaterial mat = new PhongMaterial(); mat.setDiffuseMap(new Image("wood.jpg"));
        sediste.setMaterial(mat);
        this.getChildren().addAll(sipka1, sipka2, sediste);
        Rotate r = new Rotate(); r.setPivotY(-100); r.setAxis(Rotate.X_AXIS); this.getTransforms().addAll(r);
        double ugao = 30;
        animacija = new Timeline(
                new KeyFrame(Duration.ZERO, new KeyValue(r.angleProperty(), -ugao)),
                new KeyFrame(Duration.seconds(1), new KeyValue(r.angleProperty(), 0, Interpolator.EASE_IN)),
                new KeyFrame(Duration.seconds(2), new KeyValue(r.angleProperty(), ugao, Interpolator.EASE_OUT))
        );
        animacija.setAutoReverse(true); animacija.setCycleCount(Timeline.INDEFINITE);
        animacija.play();
    }
    public void promeni() {
        if (animacija.getStatus() == Timeline.Status.PAUSED) animacija.play();
        else if (animacija.getStatus() == Timeline.Status.RUNNING) animacija.pause();
    }
    public void promeni(Node sediste){ if(sediste==this.sediste) promeni(); }
}

public class K2_2018 extends Application {
    private static PerspectiveCamera kamera, kamera1, kamera2, kamera3;
    private static PointLight crvenoSvetlo, plavoSvetlo;
    @Override
    public void start(Stage prozor){
        Group koren = new Group();
        Scene scena = new Scene(koren, 640, 400, true, SceneAntialiasing.BALANCED);
        Box trava = new Box(10000,1,10000); trava.setTranslateY(200);
        PhongMaterial matTrava = new PhongMaterial(); matTrava.setDiffuseColor(Color.LIGHTGREEN);
        trava.setMaterial(matTrava);
        Box pesak = new Box(500,1,300); pesak.setTranslateY(198);
        PhongMaterial matPesak = new PhongMaterial();
        matPesak.setBumpMap(new Image("sand_n.png")); setDiffuseMap(new Image("sand.png"));
        pesak.setMaterial(matPesak);
        Group ljuljaska = new Group();
        PhongMaterial matStubovi = new PhongMaterial(); .setDiffuseColor(Color.GRAY);
        matStubovi.setSpecularColor(Color.YELLOW); matStubovi.setSpecularPower(150);
        Ljuljaska ljuljaska1 = new Ljuljaska(matStubovi);
        ljuljaska1.getTransforms().addAll(new Translate(-100,0,0));
        Ljuljaska ljuljaska2 = new Ljuljaska(matStubovi);
        ljuljaska2.getTransforms().addAll(new Translate(100,0,0));
        Cylinder nosac = new Cylinder(3,400); nosac.setTranslateY(-100); nosac.setRotate(90);
        Box stub1 = new Box(5,300,5); stub1.getTransforms().addAll(new Translate(-180, 50, 0));
        Box stub2 = new Box(5,300,5); stub2.getTransforms().addAll(new Translate(180, 50, 0));
        nosac.setMaterial(matStubovi); stub1.setMaterial(matStubovi); stub2.setMaterial(matStubovi);
        ljuljaska.getChildren().addAll(ljuljaska1, ljuljaska2, nosac, stub1, stub2);
        koren.getChildren().addAll(trava, pesak, ljuljaska);
        koren.getChildren().addAll(napraviSvetla());
```

```
        scena.setCamera(napraviKamere());
        EventHandler<KeyEvent> r = dog -> {
            KeyCode k=dog.getCode();
            switch (k) {
                case DIGIT4: ljuljaska1.promeni(); break;
                case DIGIT5: ljuljaska2.promeni(); break;
                case DIGIT1: kamera = kamera1; scena.setCamera(kamera); break;
                case DIGIT2: kamera = kamera2; scena.setCamera(kamera); break;
                case DIGIT3: kamera = kamera3; scena.setCamera(kamera); break;
                case C: crvenoSvetlo.setLightOn(!crvenoSvetlo.isLightOn()); break;
                case P: plavoSvetlo.setLightOn(!plavoSvetlo.isLightOn()); break;
            }
        };
        EventHandler<MouseEvent> r1 = dog -> {
            PickResult rez = dog.getPickResult();
            ljuljaska1.promeni(rez.getIntersectedNode());
            ljuljaska2.promeni(rez.getIntersectedNode());
        };
        scena.addEventHandler(KeyEvent.KEY_PRESSED, r); scena.addEventHandler(MouseEvent.MOUSE_PRESSED, r1);
        scena.setFill(Color.BLUE); prozor.setScene(scena); prozor.setTitle("Ljuljaske");
        prozor.setResizable(false); prozor.show();
    }

    private PerspectiveCamera napraviKamere(){
        kamera1 = new PerspectiveCamera(true);
        kamera1.setTranslateZ(-1000); kamera1.setTranslateY(-1000);
        kamera1.setRotationAxis(Rotate.X_AXIS); kamera1.setRotate(-50);
        kamera1.setNearClip(0.1); kamera1.setFarClip(10000);
        kamera2 = new PerspectiveCamera(true); kamera2.setTranslateY(-600);
        kamera2.setRotationAxis(Rotate.X_AXIS); kamera2.setRotate(-90);
        kamera2.setNearClip(0.1); kamera2.setFarClip(10000);
        kamera3 = new PerspectiveCamera(true);
        kamera3.setTranslateY(-100); kamera3.setTranslateZ(-1500);
        kamera3.setNearClip(0.1); kamera3.setFarClip(10000);
        return kamera1;
    }

    private Group napraviSvetla(){
        Group svetla = new Group();
        crvenoSvetlo = new PointLight(); crvenoSvetlo.setColor(Color.RED);
        crvenoSvetlo.getTransforms().addAll(new Translate(200,100,0));
        plavoSvetlo = new PointLight(); plavoSvetlo.setColor(Color.BLUE);
        plavoSvetlo.getTransforms().addAll(new Translate(-200,100,0));
        AmbientLight svetlo = new AmbientLight(); svetlo.setColor(Color.LIGHTGRAY);
        svetla.getChildren().addAll(crvenoSvetlo, plavoSvetlo, svetlo);
        return svetla;
    }

    public static void main(String[] args) { launch(args); }
}
```