

RAČUNARSKA GRAFIKA
prvi kolokvijum – teoretski deo

- 1) [20] Moodle test
- 2) [10] Izvesti analitičku formulu kvadratne Bezjeove krive definisane nad tačkama P_0 , P_1 i P_2 , u zavisnosti od realnog parametra t . Tačke P_0 i P_2 su početna i krajnja tačka krive, respektivno. Napisati vektorski i matrični oblik kvadratne Bezjeove krive.

Napomene:

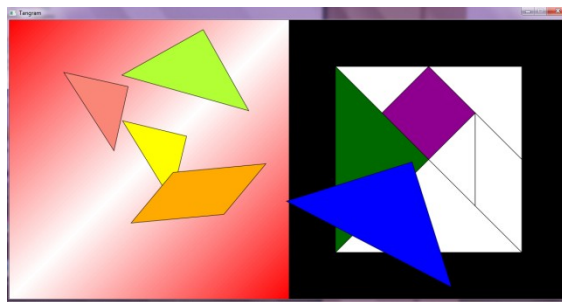
1. Izrada teoretskog dela (Moodle testa i zadatka) traje 20 minuta.
2. Rešenje 2. zadatka se predaje na ovom listu.
3. Nije dozvoljeno korišćenje literature i podsetnika u bilo kom obliku.
4. Nije dozvoljeno uz sebe imati mobilni telefon, bez obzira da li je uključen ili isključen.

REŠENJE:

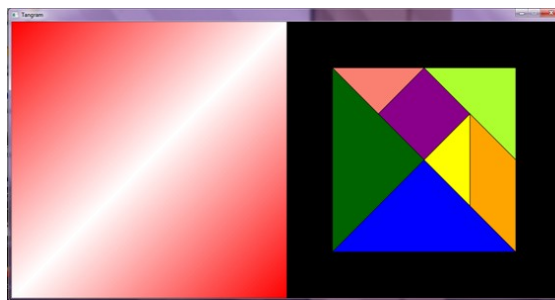
RAČUNARSKA GRAFIKA

prvi kolokvijum – praktični deo

- 1) [70] Napisati program koji koristeći grafičku biblioteku JavaFX prikazuje animaciju rešavanja slagalice Tangram. Tangram je igra slagalice za jednog igrača. Igrač ima zadatak da sve raspoložive geometrijske figure (pet trouglova, jedan kvadrat i jedan romboid) rasporedi tako da u potpunosti prekrivaju samo površ zadate figure (slagalice) složenog oblika. Program animira rešavanje slagalice, smeštanjem jedne po jedne figure sa levog dela ekrana na svoje mesto na desnom delu ekrana. Na Slici 1 dat je prikaz za vreme slaganja figure. Na početku su sve figure u levoj polovini ekrana, na slučajnim pozicijama i zarotirane za slučajnu vrednost ugla rotacije. Pozadina je prelaz crvene i bele boje. Desna polovina ekrana prikazuje finalnu figuru (kvadrat) koju treba sklopiti od raspoloživih figura. Pomeranje svake figure traje po 3s. Na Slici 2 dat je finalni izgled slagalice. Boje figura odabrati tako da odgovaraju bojama sa slike. Priložen je .jar fajl sa demonstracijom programa koji treba napisati.



Slika 1



Slika 2

Napomene:

1. Izrada praktičnog dela traje 100 minuta.
2. Rešenje zadatka se predaje u obliku .java fajla i rezultujućeg .jar fajla 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.

Rešenja zadataka
prvi kolokvijum – teoretski deo

Videti materijale sa predavanja (Predavanja 03 – Krive).

Rešenja zadatka

prvi kolokvijum – praktični deo

1)

```
package RG_1819_k1;
import java.util.Random;
import javafx.animation.*;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.paint.CycleMethod;
import javafx.scene.paint.LinearGradient;
import javafx.scene.paint.Stop;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;
import javafx.util.Duration;

class Tangram extends Group {
    private static final double T = 3; private static final int A = 600;
    private static final Polygon[] delovi = new Polygon[7];
    private static final Color[] boje = new Color[]{Color.DARKMAGENTA, Color.DARKGREEN,
    Color.BLUE, Color.GREENYELLOW, Color.SALMON, Color.YELLOW, Color.ORANGE};
    private static final Random rand = new Random();

    public Tangram(){
        Rectangle tabla = new Rectangle(0,0,A,A);
        Rectangle resenje = new Rectangle(A, 0, A, A);
        Stop[] stanice = {new Stop(1, Color.WHITE), new Stop(0, Color.RED)};
        tabla.setFill(new LinearGradient(0,0,0.5,0.5,true, CycleMethod.REFLECT, stanice));
        getChildren().addAll(tabla, resenje);

        delovi[0] = new Polygon(200.0,0.0,300.0,100.0,200.0,200.0,100.0,100.0);
        delovi[1] = new Polygon(0.0,0.0,0.0,400.0,200.0,200.0);
        delovi[2] = new Polygon(0.0,400.0,400.0,400.0,200.0,200.0);
        delovi[3] = new Polygon(200.0,0.0,400.0,0.0,400.0,200.0);
        delovi[4] = new Polygon(0.0,0.0,200.0,0.0,100.0,100.0);
        delovi[5] = new Polygon(200.0,200.0,300.0,100.0,300.0,300.0);
        delovi[6] = new Polygon(400.0,200.0,400.0,400.0,300.0,300.0,300.0,100.0);

        for(int i=0; i<7; i++){
            delovi[i].setFill(boje[i]); delovi[i].setStroke(Color.BLACK);
            delovi[i].setTranslateX(50+rand.nextInt(100));
            delovi[i].setTranslateY(50+rand.nextInt(100));
            delovi[i].setRotate(rand.nextDouble()*360);

            Polygon p = new Polygon();
            p.setStroke(Color.BLACK); p.setFill(Color.WHITE);
            p.getPoints().addAll(delovi[i].getPoints());
            p.setTranslateX(A+100); p.setTranslateY(100);
            getChildren().addAll(p);

            RotateTransition rt = new RotateTransition(Duration.seconds(T), delovi[i]);
            rt.setToAngle(0); rt.setDelay(Duration.seconds(i*T));
            TranslateTransition tt = new TranslateTransition(Duration.seconds(T), delovi[i]);
            tt.setToX(A+100); tt.setToY(100); tt.setDelay(Duration.seconds(i*T));
            tt.play(); rt.play();
        }
        for(int i=0; i<7; i++) getChildren().add(delovi[i]);
    }
}

public class K1_2019 extends Application {
    @Override
    public void start(Stage primaryStage) {
        Group koren = new Group();
        koren.getChildren().add(new Tangram());
        Scene scena = new Scene(koren, 1200, 600);
        primaryStage.sizeToScene(); primaryStage.setResizable(false);
        primaryStage.setTitle("Tangram"); primaryStage.setScene(scena);
        primaryStage.show();
    }
    public static void main(String [] args){launch(args);}
}
}
```