


Računarska grafika

JavaFX – Uvod



Uvod

- JavaFX – skup paketa za grafiku, multimediju i veb
 - omogućava razvoj desktop i bogatih internet aplikacija (RIA)
- Razvijana kao posebno izdanje do verzije 2.2
- Od verzije Java SE 7u6 – u standardnom izdanju
- Od verzije Java SE 11, ponovo posebno izdanje
- Od posebnog interesa za 2D i 3D grafiku na kursu

2D grafika u Javi FX

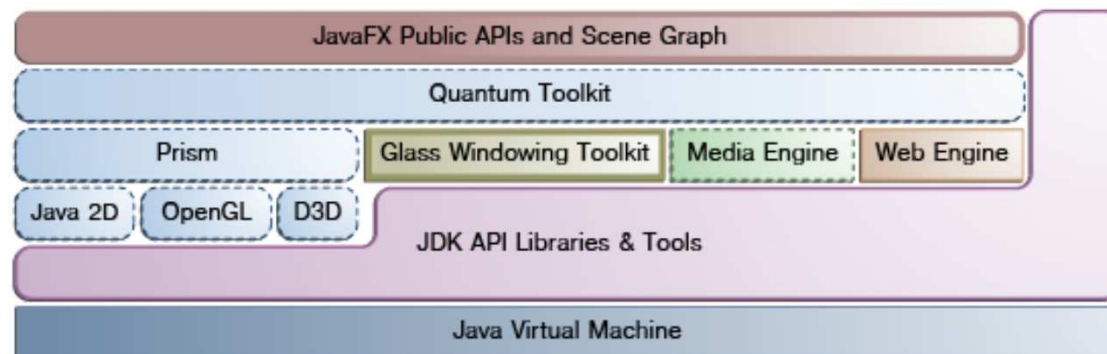
- JavaFX podrška za 2D grafiku:
 - geometrijski 2D oblici
 - primitive
 - putanje
 - kombinovani oblici
 - atributi stila primitiva
 - transformacije
 - animacija
 - interakcija
 - tekst
 - efekti

3D grafika u Javi FX

- 3D grafika u Javi FX
 - geometrijski 3D oblici (kvadar, valjak, lopta)
 - kamere
 - svetla
 - teksture
 - materijali
 - mreže površi
 - interakcija sa 3D objektima

Arhitektura

- JavaFX – javni API zasnovan na sloju radnih alata:
 - Quantum Toolkit – omotač oko:
 - Prism – rad sa grafikom (Graphics Engine)
 - Glass Windowing Toolkit – rad sa prozorima
 - Media Engine – rad sa multimedijalnim sadržajima
 - Web Engine – rad sa sadržajima na Vebu



Osnovni pojmovi

- Čvor (*Node*)
 - element scene: komponenta, geometrijska primitiva, tekst
- Graf scene (*Scene Graph*)
 - stablo čvorova
- Scena (*Scene*)
 - kontejner grafa scene
- Pozornica (*Stage*)
 - kontejner najvišeg nivoa, sadrži scenu (npr. prozor)

Niti izvršnog okruženja

- Lansirna nit
 - pokreće se po stvaranju objekta aplikacije, izvršava `init()`
- Aplikativna nit
 - kod aplikacije je primarno u ovoj niti
 - izvršava metode `start()` i `stop()`
 - stvara scenu i pridružuje pozornici, održava prikazani graf scene
 - obrada događaja (*Glass Windowing Toolkit, GWT*), animacija
- Nit za iscrtavanje scene (nit *Prism*)
 - sinhronizuje prikaz scene sa grafom scene
- Nit za reprodukciju multimedijalnih sadržaja

Osvežavanje prikaza – puls

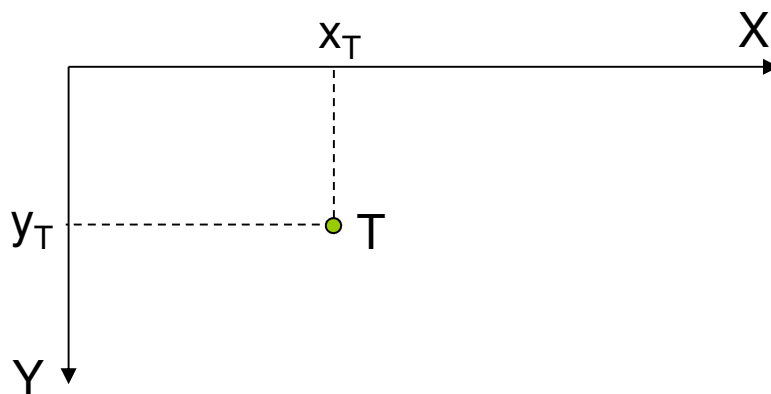
- GWT koristi tajmer za događaj otkucaja pulsa, $f = 60\text{fps}$
- Ukoliko nema aktivnih animacija, puls se obrađuje samo kada se nešto dogodi što zahteva crtanje scene
- Ulazni događaji se baferišu i obrađuju pri obradi pulsa
- Sprečava se degradacija performansi
 - graf scene se ne menja češće nego što je potrebno
 - scena se ne iscrtava češće nego što je potrebno
- Omogućena asinhrona obrada događaja i crtanje scene
 - obrada događaja u aplikativnoj, a crtanje scene u grafičkoj niti
 - dok se iscrtava N-ta slika, priprema se N+1 slika

Koraci izvršenja aplikacije

- Stvara se primerak klase izvedene iz klase `Application`
- Iz lansirne niti se poziva metod `init()`
- Iz aplikativne niti se poziva metod `start(javafx.stage.Stage)`
- U aplikativnoj niti se čekaju i obrađuju događaji i čeka se na kraj aplikacije koji se dešava:
 - kada aplikacija pozove `Platform.exit()`
 - kada se poslednji prozor zatvori i atribut `implicitExit` klase `Platform` ima vrednost `true`
- Iz aplikativne niti se na kraju poziva metod `stop()`

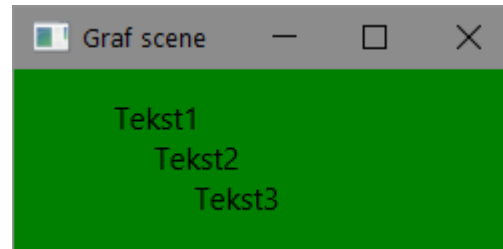
Koordinatni sistem

- Pravougli (kartezijanski) koordinatni sistem:
 - X koordinata raste sleva udesno
 - Y koordinata raste odozgo naniže
 - koordinate su tipa `double`



Primer (1)

- Rezultat izvršenja:



- Uvoz klasa iz JavaFX paketa:

```
import javafx.application.Application;  
import javafx.stage.Stage;  
import javafx.scene.Scene;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Text;
```

Primer (2)

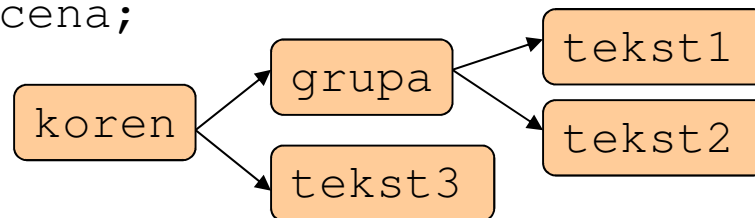
- **Struktura klase aplikacije:**

```
public class GrafScene extends Application {
    @Override
    public void start(Stage prozor) {
        Scene scena=formiranjeScene();
        podesavanjeProzora(prozor, scena);
    }
    ...
    public static void main(String[] args) {
        launch(args);
    }
}
```

Primer (3)

- Formiranje (grafa) scene:

```
private Scene formiranjeScene() {  
    Text tekst1 = new Text(50, 30, "Tekst1");  
    Text tekst2 = new Text(70, 50, "Tekst2");  
    Text tekst3 = new Text(90, 70, "Tekst3");  
    Group grupa = new Group();  
    grupa.getChildren().addAll(tekst1, tekst2);  
    Group koren = new Group();  
    koren.getChildren().addAll(grupa, tekst3);  
    Scene scena = new Scene(koren, 250, 90, Color.GREEN);  
    return scena;  
}
```



Primer (4)

- **Podešavanje i prikazivanje prozora:**

```
private void podesavanjeProzora (Stage prozor,  
                                Scene scena) {  
    prozor.setTitle ("Graf scene");  
    prozor.setScene (scena);  
    prozor.show ();  
}
```