


# Računarska grafika

JavaFX – kamera

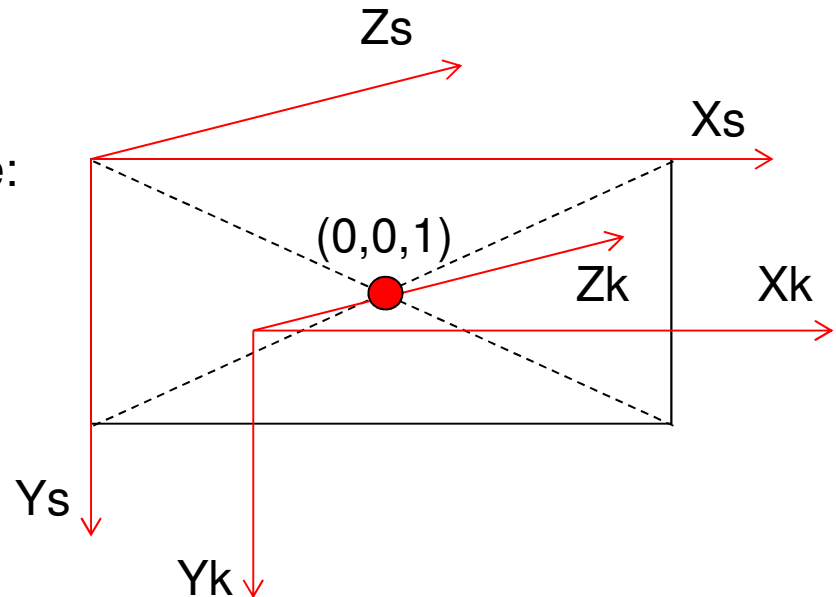


# JavaFX kamera

- Kamera je objekat kojim se snima scena
- Nalazi se u koordinatnom sistemu realnog sveta 3D scene
- Ima i svoj lokalni koordinatni sistem
- Snimak kamerom
  - predstavlja projekciju 3D scene na projekcionu ravan kamere
  - projektuje se deo scene u vidnom polju kamere
- Kao i drugi čvorovi u sceni, kamera ima položaj i orijentaciju
- Na kameru mogu da se primenjuju transformacije
  - da se pozicionira i orijentiše u koordinatnom sistemu scene
- Kamera inicijalno ima:
  - položaj – (x,y) koordinate centra scene, z se izračunava
  - orijentaciju – pravac i smer pozitivne Z-ose

# Koordinatni sistemi scene i kamere

- Koordinatni sistem scene:  $X_s$ - $Y_s$ - $Z_s$
- Koordinatni sistem kamere:  $X_k$ - $Y_k$ - $Z_k$
- Centar prozora u projekcionoj ravni
  - u koordinatnom sistemu scene:  $(\text{sirina}/2, \text{visina}/2, 0)$
  - u koordinatnom sistemu kamere:  $(0, 0, 1)$
- Pozicija (oka) kamere:
  - u koordinatnom sistemu scene  $(\text{sirina}/2, \text{visina}/2, z_k)$ ,  $z_k < 0$
  - u koordinatnom sistemu kamere  $(0, 0, 0)$



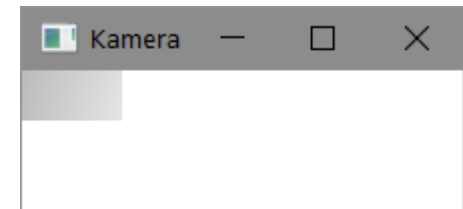
# Klase kamera, pridruživanje sceni

- JavaFX kamera je opisana apstraktnom klasom `Camera`
  - iz paketa `javafx.scene`
  - izvedena je iz klase `Node`
    - zato može da predstavlja čvor u grafu scene
- Definisane su 2 izvedene klase:
  - `ParallelCamera` – za paralelnu ortografsku kameru
  - `PerspectiveCamera` – za kameru sa perspektivnom projekcijom
  - podrazumevana kamera u sceni je paralelna ortografska kamera
- Nije dovoljno (ni potrebno za prikaz) dodati kameru u graf scene
  - mora se pridružiti sceni da bi se primenila na prikaz scene
  - metod scene: `setCamera(Camera kamera)`

# Primeri podrazumevanih kamera

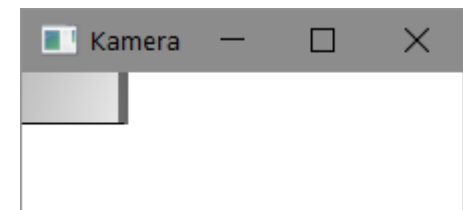
- Podrazumevana kamera (i svetlo):

```
Box kvadar = new Box(100, 50, 10);  
koren.getChildren().addAll(kvadar);  
Scene scena = new Scene(koren, 220, 100);
```



- Perspektivna kamera sa podrazumevanim parametrima (i podrazumevano svetlo):

```
Box kvadar = new Box(100, 50, 10);  
PerspectiveCamera kamera = new PerspectiveCamera();  
koren.getChildren().addAll(kvadar);  
scena.setCamera(kamera);
```



# Prikazne zapremine

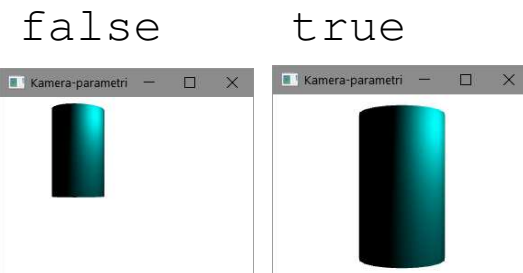
- Oba tipa kamere prikazuju deo scene iz prikazne zapremine
- Prikazna zapremina
  - kod ortografske kamere – kvadar
  - kod perspektivne kamere – zarubljena piramida
- Kamera se nalazi na podužnoj osi prikazne zapremine
- Ivice prozora pripadaju bočnim stranicama prikazne zapremine
- Obe kamere su okarakterisana svojstvima:
  - `nearClip` – rastojanje od kamere do prednje odsecajuće ravni
  - `farClip` – rastojanje od kamere do zadnje odsecajuće ravni
- Podrazumevano se zadaju u koordinatnom sistemu (oka) kamere
- Podrazumevane vrednosti:
  - `nearClip = 0.1, farClip = 100.0`

# Paralelna kamera

- Kod paralelne kamere
  - projekcioni zraci su paralelni i ortogonalni (normalni) na projekcionu ravan u kojoj se prikazuje scena
- Mesto kamere na Z pravcu, normalnom na projekcionu ravan
  - irelevantno za geometriju prikaza objekta
  - utiče na podrazumevano svetlo i na prikaznu zapreminu
- Prikazna zapremina je kvadar
- Prikaz je veoma efikasan, ali nije dovoljno realističan
- Često se koristi kada se scena prikazuje u tri 2.5D prikaza:
  - spreda/otpozadi, odozgo/odozdo, sleva/sdesna
- Postoji samo podrazumevani konstruktor

# Perspektivna kamera

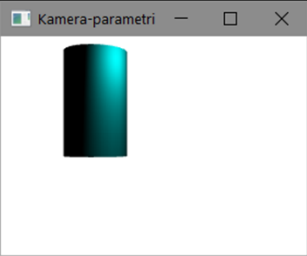
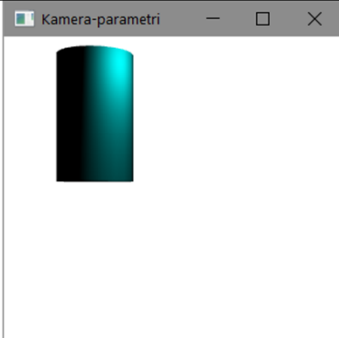
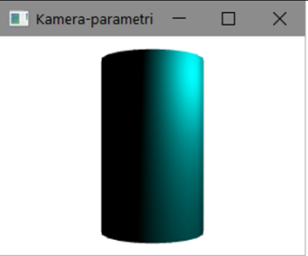
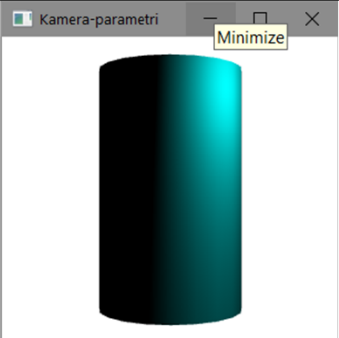
- Prikaz scene je realističan, odgovara čovekovom čulu vida
- Pored podrazumevanog, ima i konstruktor:  
`PerspectiveCamera(booleann fiksnoOkoKamereUNuli)`
- Parametar `fiksnoOkoKamereUNuli`
  - `true`: oko kamere se ne pomera iz koordinatnog početka kamere pri transformacijama kamere u koordinatnom sistemu roditelja (scene)
  - `false`: omogućava da se oko kamere, nakon transformacija, pomeri iz koordinatnog početka, tako da se prilagodi obuhvatu scene
  - podrazumevano: `false`
- U praksi:
  - kada se radi sa 3D scenom i računa se sa transformacijama kamere parametar treba da bude `true`





# Primeri za `fixsnoOknoKamereUNuli`

- Centar valjka:  $z=600$ ,  $R=75$
- Perspektivna kamera, podrazumevane vrednosti svojstava

FIKSNO_OKO=false originalni prozor	FIKSNO_OKO=false povećan prozor	FIKSNO_OKO=true originalni prozor	FIKSNO_OKO=true povećan prozor
			

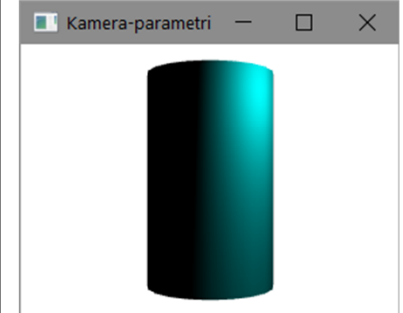
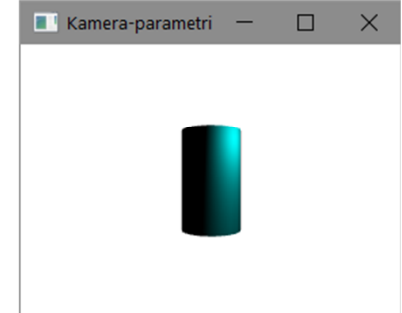
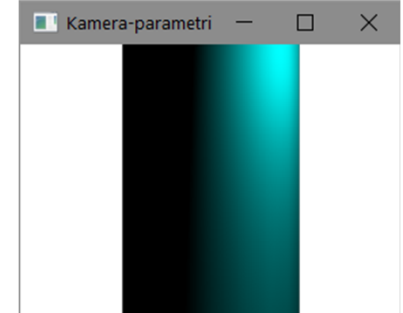
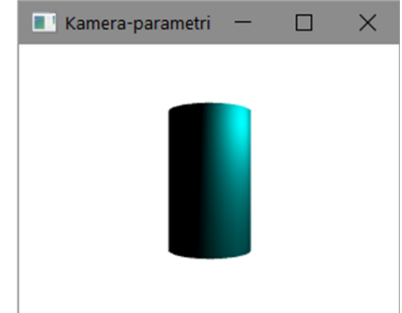
- Za `FIKSNO_OKO=false` (podrazumevano)
  - veličina objekata u projekcionoj ravni se ne menja sa promenom veličine scene, menja se obuhvat prikazne zapremine

# Svojstva perspektivne kamere

- Perspektivna kamera ima dva specifična svojstva:
  - `fieldOfView` i `verticalFieldOfView`
- **Realno (double) svojstvo** `fieldOfView`
  - vidni ugao kamere, meri se u stepenima
  - podrazumevano je 30°
- **Logičko svojstvo** `verticalFieldOfView`
  - određuje da li se podrazumevana z-koordinata kamere računa tako da vrednost `fieldOfView` odgovara visini ili širini 2D scene
  - `true`: računa se prema vertikalnoj dimenziji scene (visini)
  - `false`: računa se prema horizontalnoj dimenziji scene (širini)
  - podrazumevana vrednost svojstva je `true`

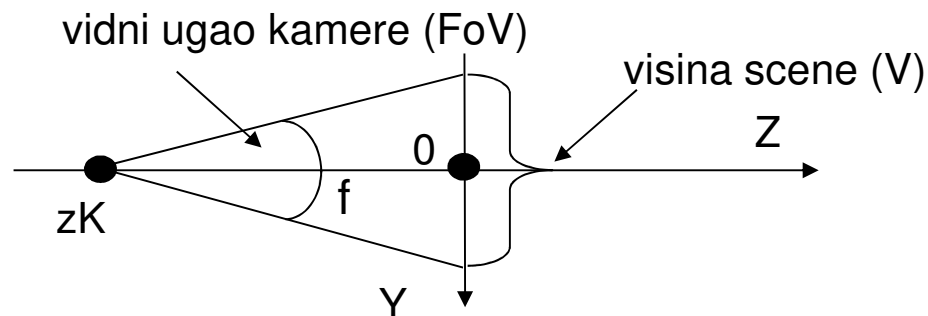
# Primeri vidnog ugla

- Centar valjka  $Z=600$ ,  $R=75$
- Scena:  $280 \times 200$  ( $V=200$ ,  $H=280$ )
- Menja se `fieldOfView` i `verticalFieldOfView`

FIKSNO_OKO=true VIDINI_UGAO=30 VERTIKALNO=true	FIKSNO_OKO=true VIDINI_UGAO=60 VERTIKALNO=true	FIKSNO_OKO=true VIDINI_UGAO=30 VERTIKALNO=false	FIKSNO_OKO=true VIDINI_UGAO=60 VERTIKALNO=false
			

# Podrazumevana z-koordinata

- Projekciona ravan je ravan  $z=0$  u koordinatnom sistemu scene
- Rastojanje kamere od projekcione ravni – žižna daljina
- Žižna daljina (fokalno rastojanje)  $f$  u sistemu scene je određena:
  - visinom (ili širinom) scene ( $V$ )
  - vidnim uglom kamere ( $FoV$ )

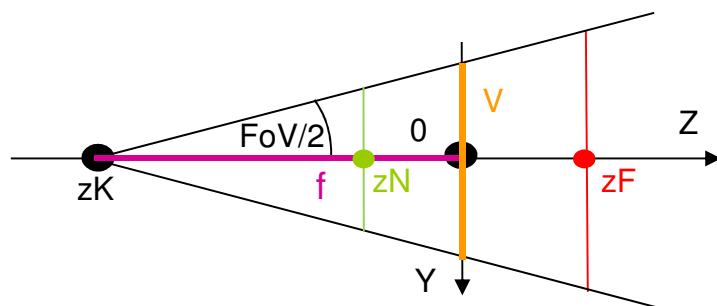


- Potrebno je odrediti  $z_K$  u sistemu scene

# Parametri kamere u sistemu scene

- Vrednosti svojstava `nearClip` i `farClip` se zadaju – u koordinatnom sistemu kamere
- Žižna daljina u sistemu kamere – rastojanje do prikazne ravni:  $F=1$
- U koordinatnom sistemu scene:

```
final double f = (V/2.0)/Math.tan(Math.toRadians(FoV)/2.0);  
final double zK = -1.0 * f;  
final double zN = f * NCP + zK;  
final double zF = f * FCP + zK;
```



## Zadato:

V - visina scene u sistemu scene

FoV - ugao kamere

NCP - nearClip u sistemu kamere

FCP - farClip u sistemu kamere

## Izračunato:

f - žižna daljina u sistemu scene

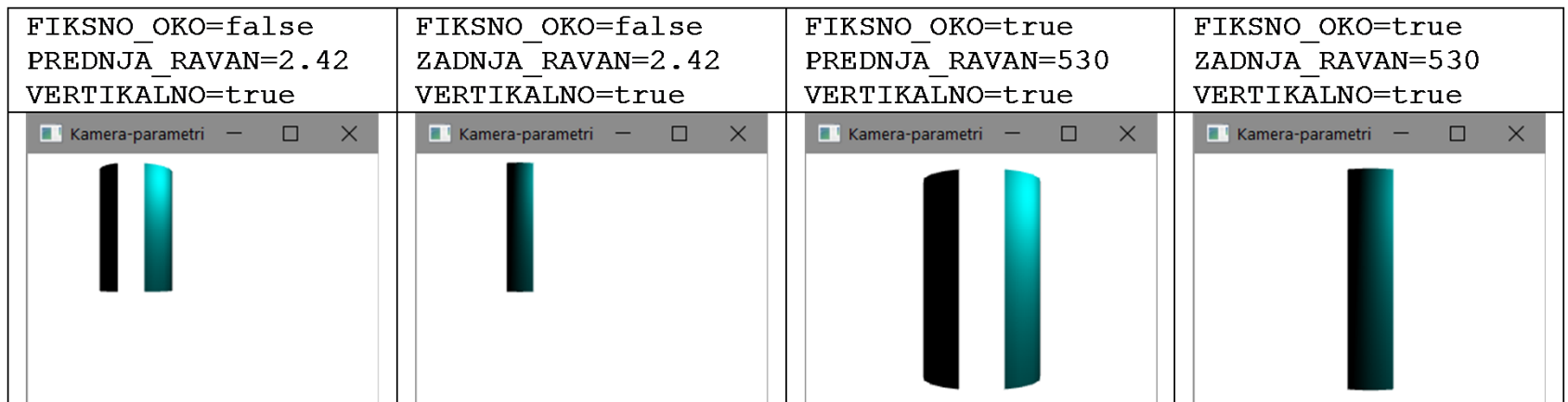
zK, zN, zF - z koord. u sistemu scene

# Primeri nearClip i farClip

- Centar valjka  $Z=600$ ,  $R=75 \rightarrow Z_{VALJAK}=[525,675]$ , ravan u  $Z=530$
- Scena:  $280 \times 200$  ( $V=200$ ,  $H=280$ )
- Menjaju se nearClip i farClip, za `FIKSNO_OKO=false/true`
  - za `FIKSNO_OKO=false`

$$f = (V/2) / \text{tg}(FoV/2) = 100 / \text{tg}(15^\circ) = 373.205, \quad zK = -f$$

$$NCP = (zN - zK) / f = (530 + 373.205) / 373.205 = 2.42$$



JavaFX - kamera

03.04.2017.