

# Računarska grafika (i svašta još nešto)

Stefan Nožinić [stefan@lugons.org](mailto:stefan@lugons.org)

# Agenda

- Reprezentacija 2D slike u računaru
- transformacije 2D slike
- vektorska grafika i rendering 2D slike
- uvod u 3D grafiku
- transformacije u 3D grafici
- arhitektura GPU
- fizika u video igrima

# Slika i RGB sistem boja

- sliku predstavljam kao 2D matricu gde svaka ćelija sadrži trojku (r,g,b)
- brojevi u trojci reprezentuju količinu crvene, zelene i plave boje (3 bajta)
- Razlog za izbor RGB je aditivnost boja (ekran računara radi tako što meša crvenu, zelenu i plavu svetlost da napravi ceo spektrum boja)
- RYB nije moguće koristiti kada radimo sa svetlošću jer, na primer, ne možemo dobiti zelenu boju (za razliku od mastila)

# Drugi sistemi boja

- RGBA
- HSV
- CMYK

# HSV

- $$V = \frac{R + G + B}{3}$$
- $$1 - \frac{3}{R + G + B} \min(R, G, B)$$
- $$H = \cos^{-1}\left(\frac{2R - B - G}{2\sqrt{(R - G)^2 + (R - B)(G - B)}}\right)$$
- za  $G > B$
- $H = 360 - H$  za  $B > G$

# Transformacije nad slikom

- kako možemo posmatrati sliku
- translacija
- skaliranje
- rotacija

# Slika kao signal

- model
- interpolacija uvod

# Translacija

- svaku tačku pomerimo za  $(w_x, w_y)$



# Skaliranje

- računamo vrednost signala između tačaka
- nearest neighbour interpolacija
- bilinearna interpolacija

# Rotacija

- rotacija oko tačke (0,0)

$$x' = \cos \theta x - \sin \theta y$$

$$y' = \sin \theta x + \cos \theta y$$

- rotacija oko centra?

# Vektorska grafika

- ne čuvamo piksele već informacije o oblicima
- linija
- luk
- ostali oblici

# Crtanje linije na ekranu

- 

$$y = kx + n$$

- 

$$k = \frac{y_2 - y_1}{x_2 - x_1}$$

- 

$$n = y_1 - kx_1$$

# 3D grafika

- verteksi  $(x,y,z)$
- objekti se reprezentuju trouglovima
- svaki trougao čine 3 verteksa

# Algebarska definicija vektora

- predstava vektora
- absolutna vrednost vektora
- skalarni proizvod algebarski i geometrijski
- vektor kroz dve tačke
- radius vektor
- ortogonalni vektor (normalni)
- vektorski proizvod
- jedinični vektor

# Matrice

- sistemi linearnih jednačina i matična reprezentacija
- determinante
- množenje matrice vektorom
- množenje matrica
- jedinična matrica
- inverzna matrica

# Rang matrice

- definicija ranga preko sistema linearnih jednačina



# Vektorski prostori

- linearna kombinacija vektora
- linearna (ne)zavisnost
- definicija vektorskog prostora
- baza vektorskog prostora
- kolona prostor

# Vektorska notacija

- verteks možemo predstaviti kao vektor  $(x,y,z)$
- sve linearne transformacije (rotacija i skaliranje) možemo predstaviti u matričnom obliku

# Matrični oblici transformacija

- rotacija
- skaliranje
- translacija
- 

$$x' = TRSx$$

# Različiti pogledi na objekat

- model space
- camera space
- perspektivna projekcija

# Camera space

- $C = R_C T_C$
- imamo up vektor i normalni vektor ravni kamere
- dobijamo novi vektor iz normalnog i up vektora
- pravimo matricu A od vektora u,v,n

- $C = A^T T$

- $T = (-x_c, -y_c, -z_c)$

# Clipping

- skaliranje na prostor (-1, 1)

# Perspektivna projekcija

- dalji objekti izgledaju manje
- sličnost trouglova

- 

$$y_p = -\frac{y}{z/d}$$

- 

$$x_p = -\frac{y}{z/d}$$

- 

$$z = -d$$

# GPU pipeline

- vertex shader
- generisanje primitiva i clipping
- rasterizacija
- fragment shader

# Verteks shader

```
in vec4 vPosition;  
  
void main()  
{  
    gl_Position = vPosition;  
}
```



# Fragment shader

```
void main()  
{  
    gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);  
}
```

# Komunikacija sa GPU

- vertex buffer
- index buffer
- color buffer
- texture
- transformacije

# Teksture u fragment shaderu

```
in vec2 st;  
in vec4 color;  
  
uniform sampler2D texMap;  
  
void main()  
{  
    gl_FragColor = color * texture2D(texMap, st);  
}
```

# Model u 3D

- vertex buffer
- index buffer
- texture coordinates buffer
- default color buffer

# Svetlo

- podešava se u fragment shaderu jer on određuje boju
- ambijentalno svetlo
- Tačkasti izvor

- 

$$I(p) = \frac{1}{|p - p_0|^2} I(p_0)$$

- reflektor

- 

$$I_R = I_0 \cos^d \theta$$

# Svojstva materijala

- ambijentalno svojstvo
- difuzija
- tačkasto svetlo

- 

$$I_a = K_A I_A$$

- 

$$R_d = \frac{k_d}{\alpha + \beta d + \gamma d^2} (I \cdot n) L_d$$

- 

$$I_s = k_s L_s \max((r \cdot v)^\alpha, 0)$$

# Verteks shader za Fongov model

```
in vec4 vPosition;  
in vec4 Normal;  
  
uniform mat4 ModelView;  
uniform vec4 LightPosition;  
uniform mat4 Projection;  
  
out vec3 N;  
out vec3 L;  
out vec3 E;  
  
void main()  
{  
    gl_Position = Projection*ModelView*vPosition;  
    N = Normal.xyz;  
    L = LightPosition.xyz - vPosition.xyz;  
    if (LightPosition.w == 0) L = LightPosition.xyz;  
    E = vPosition.xyz;  
}
```

# Fragment shader

```
uniform vec4 ambientProduct, diffuseProduct, specularProduct;
uniform mat4 ModelView;
uniform vec4 LightPosition;
uniform float shininess;

in vec3 N;
in vec3 L;
in vec3 E;

void main()
{
    vec3 NN = normalize(N);
    vec3 EE = normalize(E);
    vec3 LL = normalize(L);
    vec4 ambient, diffuse, specular;

    vec3 H = normalize(LL+EE);
    float kd = max(dot(LL,NN), 0.0);
    float ks = pow(max(dot(NN, H), 0.0), shininess);

    ambient = ambientProduct;
    diffuse = kd*diffuseProduct;
    specular = ks*specularProduct;
    gl_FragColor = vec4((ambient + diffuse + specular).xyz, 1.0);
}
```



# Strukture podataka u 3D grafici

- vokseli
- quadtree, octree,
- parametarska reprezentacija

# Fizika u video igrama

- kretanje (pozicija, brzina, ubrzanje)
- sile

# Izvod funkcije jedne promenljive

- tangenta na grafiku
- izvod linearnih funkcija, kvadratne funkcije
- izvod zbira, izvod proizvoda
- viši izvod
- Tejlorova aproksimacija

# Diferencijalne jednačine

- nepoznata funkcija

- 

$$\frac{dx}{dt} = k$$

- 

$$\frac{dx}{dt} = kt$$

- 

$$\frac{dx}{dt} = kx$$

# Modeli fizičkih sistema

- opisani su diferencijalnim jednačinama
- translatorno kretanje tela - review
- rotaciono kretanje
- kretanje tkanine - Hook-ov zakon

# Metode za rešavanje diferencialnih jednačina

- jednačine prvog reda: Ojlerov metod
- jednačine višeg reda: Ojlerov metod

# Sudari

- detekcija upotrebom bounding ox-a
- detekcija sudara kod kompleksnih nekonveksnih objekata
- rezolucija sudara

# Problem n tela

- opis problema
- model
- simulacija



# Paralelizacija na GPU

- geometrijska dekompozicija
- SIMD arhitektura
- primer sa n tela

# ostali fizički modeli

- fluidi
- vatra
- projektili
- vožnja automobila
- generisanje realističnog nasumičnog terena terena
- erozija
- zemljotres

# šta još postoji

- AI za video igre
- distribuirano računarstvo

**Pitanja?**