

Základy počítačové grafiky

Základy vykreslování 3D scény, perspektivní a paralelní projekce

Michal Španěl
Tomáš Milet



Ústav počítačové grafiky a multimédií

Brno 2023

Cíl přednášky

Seznámit se s principy zobrazování 3D grafiky...

Co tvoří 3D scénu?

Jak se zobrazí 3D scéna na 2D obrazovce?

Co je to perspektivní a paralelní projekce?

Obsah

1 Úvod

- Zobrazení 3D scény
- Práce s transformacemi
- Grafická pipeline

2 Projekce

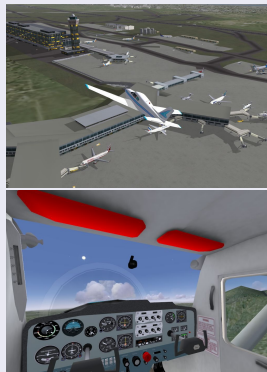
- Paralelní projekce
- Perspektivní projekce
- Knihovna GLM

Motivační příklady

Architektura



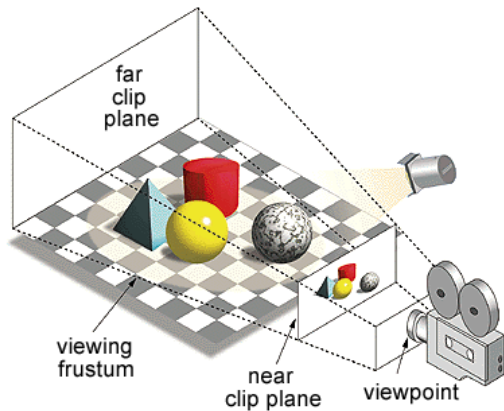
Simulátory, hry, atd.



Co potřebujeme pro zobrazení 3D scény?



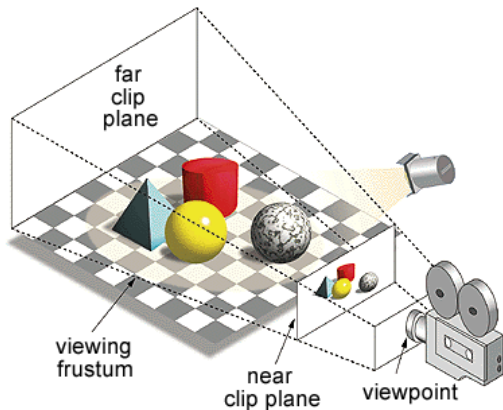
Prvky 3D scény



Objekty

- Geometrie, materiál, průhlednost, atd.

Prvky 3D scény



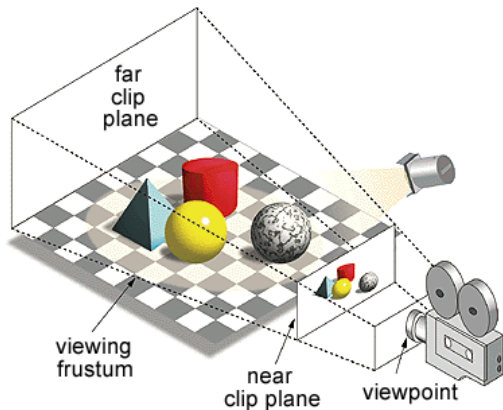
Objekty

- Geometrie, materiál, průhlednost, atd.

Zdroj světla

- Bodový, plošný, ...

Prvky 3D scény



Objekty

- Geometrie, materiál, průhlednost, atd.

Zdroj světla

- Bodový, plošný, ...

Kamera

- Pozice, orientace, FOV, ...

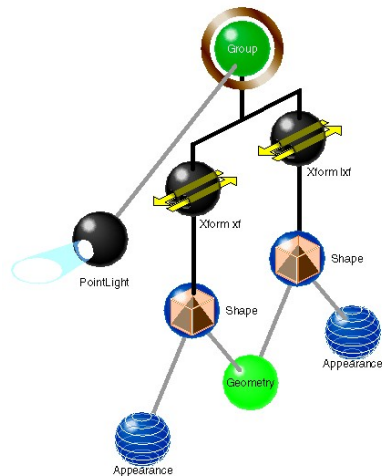
Graf scény

Hierarchická reprezentace 3D scény

- Odpovídá logické struktuře scény
- Knihovny OpenSceneGraph, Ogre3D, ...

+/-

- Intuitivní práce se scénou
- Struktura není úplně vhodná pro současná grafická API a maximální výkon/FPS



Tvoříme svět...

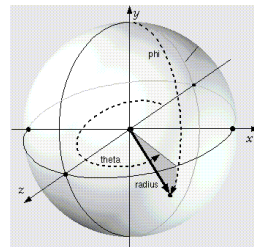
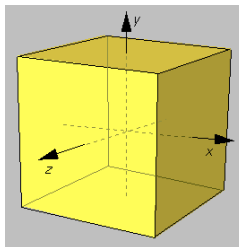
- 3D modely jednotlivých objektů.
- Model scény (rozmístění objektů, poloha kamery, apod.).

Tvoříme svět...

- 3D modely jednotlivých objektů.
- Model scény (rozmístění objektů, poloha kamery, apod.).

3D model objektu (*model space*)

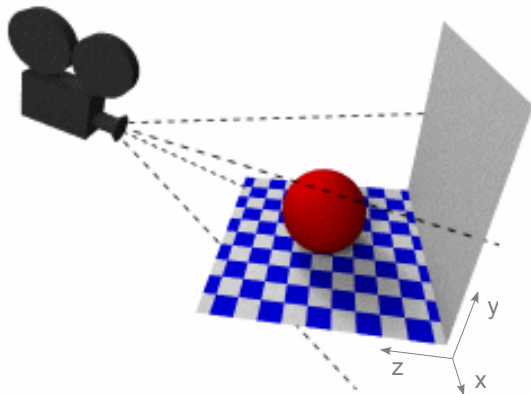
- Vhodně **zvolený lokální souřadný systém**
- Také *local coordinates* nebo *local space*



Tvoříme svět...

Vytvoření 3D scény - souřadný systém scény (*world coordinates*)

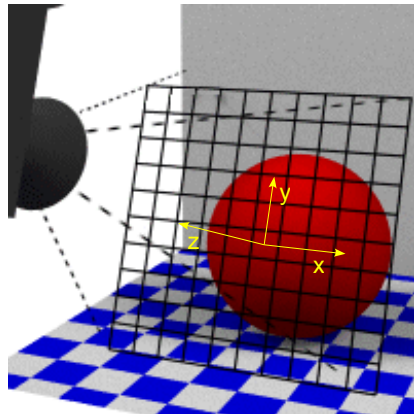
- Umístění objektu do scény → **transformace do prostoru scény** (posunutí, rotace, apod.).
- Definice kamery (poloha, směr pohledu, směr "nahoru", zorný úhel).



Tvoříme svět...

Zobrazení scény (*view space*)

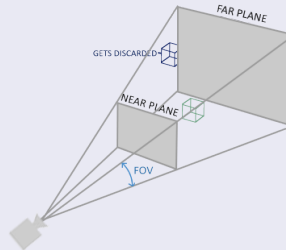
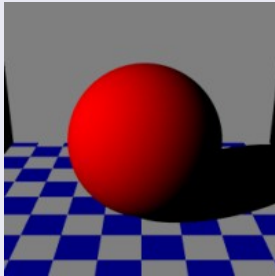
- Souřadný systém kamery.
- Transformace do souřadného systému kamery.



Tvoříme svět...

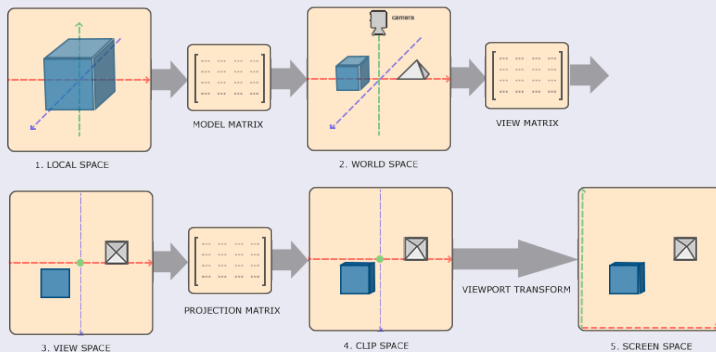
(Perspektivní) projekce

- Převod do 2D souřadného systému v rovině obrazovky (*NDC (Normalized Device Coordinates) space* společně s *clip space*).
- Ořezání primitiv hranicemi okna (*view frustum*).



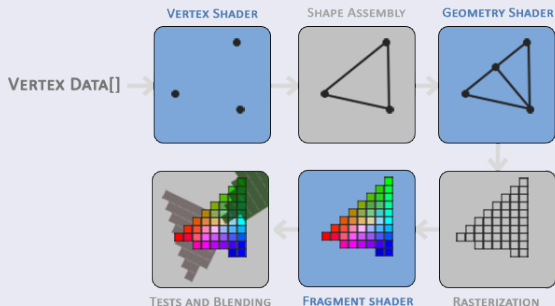
Tvoříme svět...

Shrnutí z pohledu transformačních matic



A zobrazujeme...

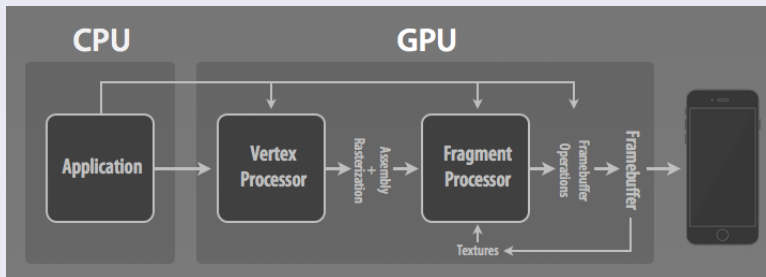
Základní představa grafické pipeline



- Podrobnosti v přednášce o OpenGL

A zobrazujeme...

Co se kde dělá?



- Kde se chystají transformace?
- Kde se počítají/aplikují transformace?

Obsah

1 Úvod

- Zobrazení 3D scény
- Práce s transformacemi
- Grafická pipeline

2 Projekce

- Paralelní projekce
- Perspektivní projekce
- Knihovna GLM

Projekce

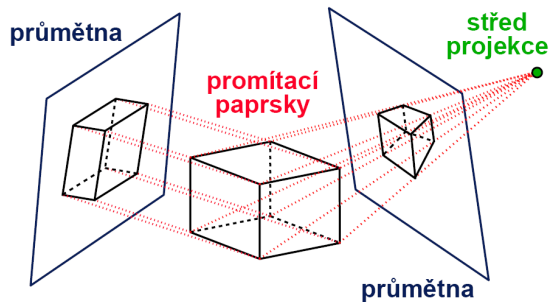
- 3D objekty zobrazujeme na 2D výstupu.
- Projekce (též. promítání) = *transformace ze 3D do 2D prostoru*.
- Dochází ke ztrátě informace.
- **Projekční paprsek promítá body na průmětnu.**

Projekce

- 3D objekty zobrazujeme na 2D výstupu.
- Projekce (též. promítání) = *transformace ze 3D do 2D prostoru*.
- Dochází ke ztrátě informace.
- **Projekční paprsek promítá body na průmětnu.**

Základní druhy projekcí

- Paralelní (rovnoběžná).
- Perspektivní (středová).



Projekce, pokr.

Promítání na rovinnou průmětnu

- *Úsečky se projektují na úsečky.*
- Projekce pouze vrcholů úseček.

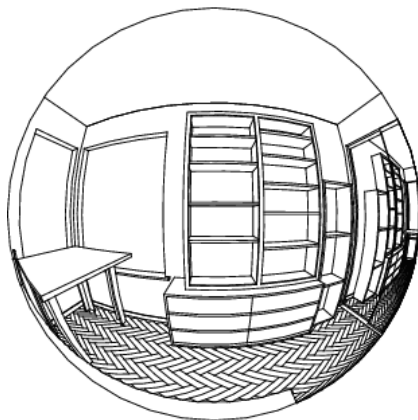
Projekce, pokr.

Promítání na rovinnou průmětnu

- *Úsečky se projektují na úsečky.*
- Projekce pouze vrcholů úseček.

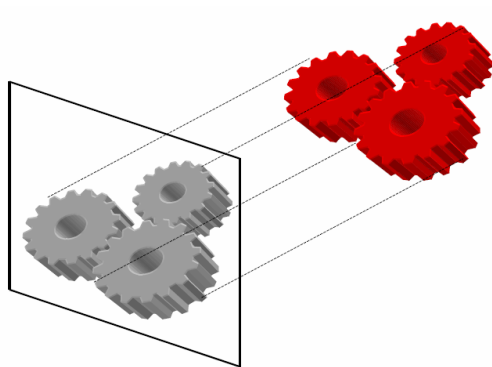
Promítání na zakřivenou průmětnu

- *Úsečky se projektují na křivky (panorama, rybí oko).*
- Projekce *všech* rastrovaných bodů objektu.



Paralelní projekce

- Lineární projekce prostřednictvím rovnoběžných paprsků – *rovnoběžné promítání*.
- **Zachovává rovnoběžnost hran!**
- Použití pro technické CAD/CAM aplikace – výkresová dokumentace, technická schémata, apod.
- *Vzdálenost od průmětny neovlivňuje velikost průmětu!*
- Nejčastěji tzv. *kolmé promítání* – paprsky kolmé na průmětnu.

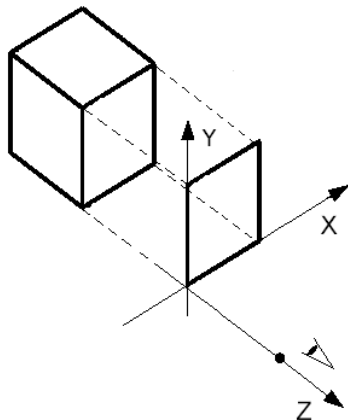


Paralelní projekce, pokr.

Kolmá projekce, průmětna rovnoběžná s rovinou XY

- Promítnutí do roviny XY v ose Z = *zanedbání* (= 0) souřadnice Z.

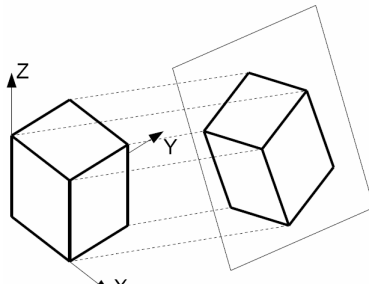
$$P_{XY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Paralelní projekce, pokr.

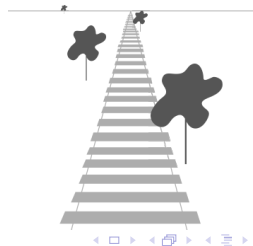
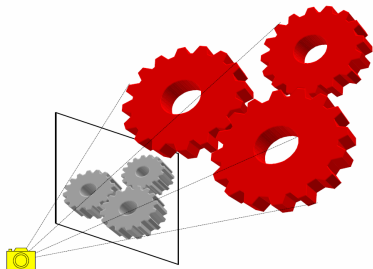
Obecná kolmá projekce

- Přenést objekty do souřadného systému, kde je průmětna rovnoběžná s rovinou XY (složená lineární transformace).
- Provést promítnutí do roviny XY.



Perspektivní projekce (odpovídá promítání v realitě)

- *Nelineární středová projekce* – paprsky vycházejí z jednoho bodu (střed projekce, místo pozorovatele).
- **Nezachovává rovnoběžnost hran!**
- Použití pro virtuální realitu, architekturu, hry, atd.
- *Vzdálenost od středu projekce ovlivňuje velikost průmětu!*



Transformační matice pro perspektivní projekci

- Jednobodová perspektiva, průmětna rovnoběžná s rovinou XY.

Parametrické vyjádření usečky PS

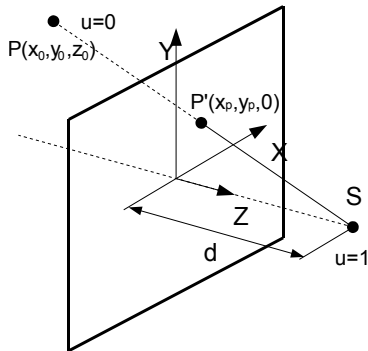
- Body $P = (x_0, y_0, z_0)$ a $S = (0, 0, d)$, $u \in \langle 0, 1 \rangle$.

$$x = x_0 - x_0 u$$

$$y = y_0 - y_0 u$$

$$z = z_0 - (z_0 + d)u$$

- V rovině XY platí
 $z = 0 \rightarrow u = z_0 / (z_0 + d)$



Transformační matice pro perspektivní projekci, pokr.

Odvození matice

- Po dosazení $u = z_0/(z_0 + d)$

$$x' = x_0 \frac{1}{1 + z_0/d}, \quad y' = y_0 \frac{1}{1 + z_0/d}$$

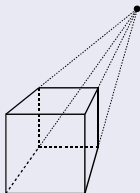
- Maticový zápis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

Vícebodová perspektivní projekce

Jednobodová perspektiva

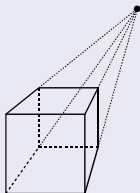
- Průmětna protíná jedinou souřadnicovou osu.
- Úsečky kolmé na průmětnu se sbíhají do jednoho bodu - *úběžníku*.



Vícebodová perspektivní projekce

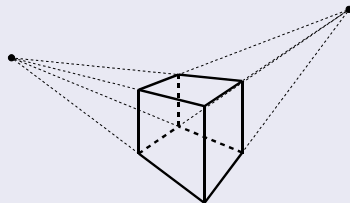
Jednobodová perspektiva

- Průmětna protíná jedinou souřadnicovou osu.
- Úsečky kolmé na průmětnu se sbíhají do jednoho bodu - *úběžníku*.



Dvoubodová perspektiva

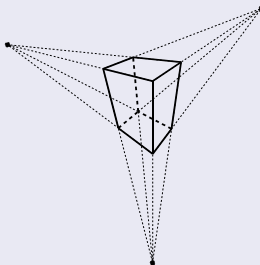
- Průmětna protíná dvě ze souřadných os.
- Hrany rovnoběžné s těmito osami směřují do dvou úběžníků.



Vícebodová perspektivní projekce, pokr.

Tříbodová perspektiva

- Nejobecnější případ, průmětna protíná všechny tři osy.
- Hrany rovnoběžné s osami se sbíhají do tří úběžníků.



OpenGL Mathematics (GLM)

- A C++ mathematics library for graphics programming
- Stejné pojmenování a funkcionality jako OpenGL Shading Language (GLSL)
- <http://glm.g-truc.net/>

OpenGL Mathematics (GLM)

- A C++ mathematics library for graphics programming
- Stejné pojmenování a funkcionality jako OpenGL Shading Language (GLSL)
- <http://glm.g-truc.net/>

Transformace vrcholu

$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + T_x \\ y + T_y \\ z + T_z \\ 1 \end{pmatrix}$$

```
glm::vec4 vec(x, y, z, 1.0f);  
glm::mat4 trans;  
trans = glm::translate(trans, glm::vec3(tx, ty, tz));  
vec = trans * vec;
```

OpenGL Mathematics (GLM), pokr.

Skládání transformací (*model matrix*)

```
glm::mat4 trans;  
trans = glm::rotate(trans, 90.0f, glm::vec3(0.0, 0.0, 1.0));  
trans = glm::scale(trans, glm::vec3(0.5, 0.5, 0.5));
```

OpenGL Mathematics (GLM), pokr.

Skládání transformací (*model matrix*)

```
glm::mat4 trans;  
trans = glm::rotate(trans, 90.0f, glm::vec3(0.0, 0.0, 1.0));  
trans = glm::scale(trans, glm::vec3(0.5, 0.5, 0.5));
```

- V jakém pořadí se budou aplikovat?

OpenGL Mathematics (GLM), pokr.

Pozice kamery a transformace do souřadného systému kamery (*view matrix*)

```
glm::mat4 view;  
view = glm::lookAt(glm::vec3(0.0f, 0.0f, 3.0f),  
                  glm::vec3(0.0f, 0.0f, 0.0f),  
                  glm::vec3(0.0f, 1.0f, 0.0f));
```

- Pozice kamery
- Cílová pozice
- Vektor směrem nahoru

OpenGL Mathematics (GLM), pokr.

Perspektivní projekce (*projection matrix*)

```
glm::mat4 projection;  
projection = glm::perspective(45.0f, screenWidth / screenHeight, 0.1f, 100.0f);
```

- FOV (zorné pole)
- Aspect ratio (poměr stran pohledového jehlanu)
- Near plane
- Far plane

OpenGL Mathematics (GLM), pokr.

Všechny transformace provádíme ve Vertex shaderu...

```
#version 330 core
layout (location = 0) in vec3 position;
...
uniform mat4 model;
uniform mat4 view;
uniform mat4 projection;

void main()
{
    // Note that we read the multiplication from right to left
    gl_Position = projection * view * model * vec4(position, 1.0f);
    ...
}
```

Další info

- Learn OpenGL - Transformations

- <https://learnopengl.com/#!Getting-started/Transformations>

- Learn OpenGL - Coordinate Systems

- <https://learnopengl.com/#!Getting-started/Coordinate-Systems>