

Základy počítačové grafiky

Přednáška 1

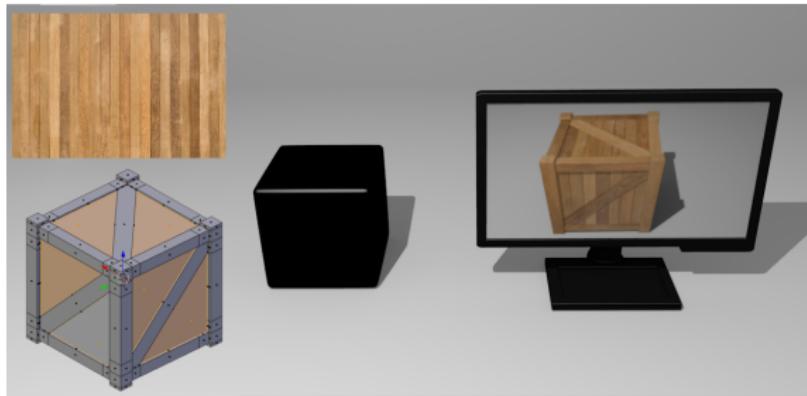
Martin Němec

VŠB-TU Ostrava

2025

Cíl předmětu

Seznámit studenty se základními principy 3D počítačové grafiky s využitím grafického API a práce se shadery. Studenti získají základní znalosti o činnosti grafické karty a zobrazovacího řetězce, osvojí si vybrané základní algoritmy a prakticky si vyzkouší implementaci klíčových technik.





"Vysoká škola není o tom, co se naučíte, ale o tom, že jste schopni zvládnout náročné úkoly a vytrvat." – anonymní autor

- Metody a nástroje pro realizaci grafických uživatelských rozhraní (kognitivní schopnosti člověka, mentální modely, základní pravidla designu, barevné prostory, volba barev a prezentace textu).
- Standardní zobrazovací řetězec (realizace jednotlivých kroků řetězce, modelovací a zobrazovací transformace, Phongův osvětlovací model, řešení viditelnosti, identifikace těles, stručná charakteristika standardu OpenGL a jazyka GLSL).
- Geometrické modelování (affinní a projektivní prostory, popis těles a možnosti jejich reprezentace, základní křivky používané v počítačové grafice, jejich vyjádření, vlastnosti a použití, Fergusonova kubika, Bézierova křivka).

Příklad otázky: Popište možnosti reprezentace těles a způsob jejich vykreslení pomocí standardního zobrazovacího řetězce v kontextu grafického rozhraní OpenGL.

Bakalářské studium

- Základy počítačové grafiky
- Modelování v grafických aplikacích

Magisterské studium

- Počítačová grafika I
- Počítačová grafika II
- Analýza obrazu
- Geometrie pro počítačovou grafiku
- Vizualizace dat
- Digitální zpracování obrazu
- Algoritmizace geometrických úloh

Předměty z PG na FEI

V předmětu **MGA** (volitelný, příští semestr) budeme modelovat v prostředí Blender a s využitím Unreal Enginu (vizualizaci za nás bude dělat UE).

Předmět **PG1** je zaměřen na realistický rendering (ray tracing).



Wasttron Base Raytracer
PG1 Ray Tracer, ©2023 Tomáš Fabian

Zápočet celkem 45 bodů (min. 20 bodů)

- Postupné vytváření 3D aplikace (C++ a OpenGL), udílení bodů a obhajoba v zápočtovém týdnu.
- Účast na cvičení je **povinná** a bude průběžně kontrolována.

Zkouška celkem 55 bodů (min. 20 bodů)

- Písemná část - 25 bodů (min. 10 bodů)
zápočtový týden na přednášce.
- Ústní zkouška - 30 bodů (min. 10 bodů).
zkouškové období (2x otázka).

Obsah přednášek

- Úvod do standardního zobrazovacího řetězce (OpenGL).
- Homogenní souřadný systém, transformace.
- Promítání, kamera, souřadné systémy.
- Světlo, intenzita osvětlení, lokální osvětlovací modely.
- Textury v OpenGL, mapování.
- Řešení viditelnosti (z-buffer, malířův algoritmus), skybox.
- Optická iluze nerovnosti, normal mapping.
- Identifikace objektů.
- Stíny v počítačové grafice.
- Úvod do křivek a ploch.

Pořadí i obsah se mohou změnit.

Zápočet

Během semestru budeme postupně na cvičeních vytvářet aplikaci (jazyk C++ a OpenGL 3.3+), která bude umožňovat načíst 3D scénu (modely, textury, světla apod.), načtenou scénu následně vizualizovat a procházet s využitím základního vykreslovacího řetězce.

Seznámíme se s jednotlivými kroky OpenGL, transformacemi, osvětlením, animacemi, shadery atd.

Budeme předpokládat základní znalosti:

- probraná SŠ a VŠ matematika (matematická analýza, lineární algebra);
- programování (primárně v C++);
- objektově orientované programování.

Zápočet

- Na OpenGL aplikaci budeme pracovat průběžně na jednotlivých cvičeních, včetně kontroly přes systém kelvin.cs.vsb.cz.
- V čem spočívá "obtížnost"? Zvládne to AI?
- Přednášky jsou doporučené (chodit si je "odsedět").
- Cvičení jsou povinná (diskutujte, ptejte se).

Úkol 1

Select multiple files using CTRL or SHIFT or drag them to this window

Assignment

Zadání

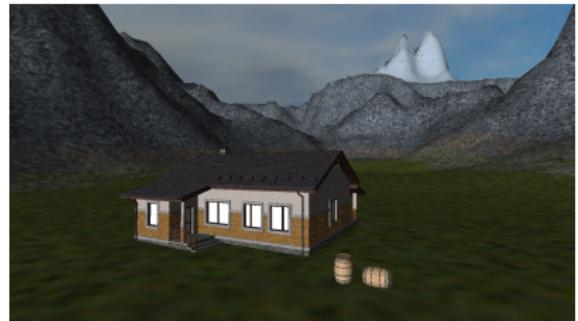
1. Upravte stávající projekt z prvního cvičení na moderní OpenGL a objektový kód (viz přednáška a cvičení).
2. Funkce `main` bude mít pouze inicializaci základního objektu `Application` a případné nastavení a spouštění. Doplňte třídy minimálně pro modely, shadery, kontrolu callback funkci atd. Základní požadavky byly probrány na přednášce.

Odevzdání úkolu

1. Odešlete zdrojové soubory `CPP` a `H` (neposílejte knihovny), které můžete zabalit do `zip` souboru.
2. Přiložte screen obrazovky s výsledky (trojúhelník s pozici na barvě, čtverec atd.).

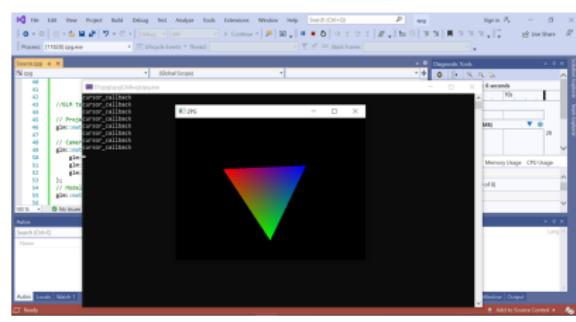
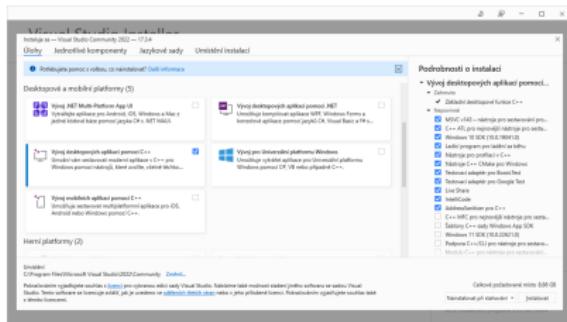
Zápočet

V aplikaci se pokusíme vykreslit základní 3D prostředí (světla, modely, textury, pozadí atd.), které budeme moci pomocí myši a klávesnice procházet. V projektu budeme používat různé modely, zkusíme si vytvořit i vlastní.



Cvičení 1

- Primárně budeme používat vývojové prostředí Visual Studio Community, které je na všech učebnách a je zdarma (lze použít i jiné vývojové prostředí, podle preferencí, operačního systému apod.).
- Budeme potřebovat externí knihovny: pro práci s okny, matematické operace, načítání obrázků, modelů atd.
- Vše budeme postupně řešit na cvičeních, kde budeme naši aplikaci vyvíjet. Cvičení jsou povinná.



Výsledná 3D aplikace

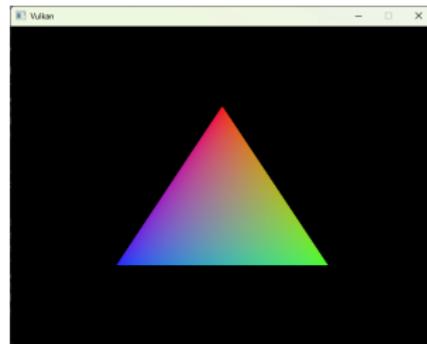
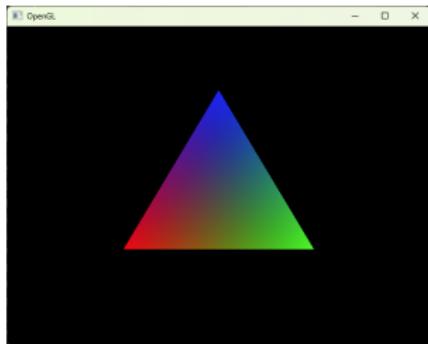
- Součástí předmětu je i zopakování dříve získaných znalostí (počítačová grafika, programování, matematika, geometrie atd.).
- Vyjdeme z čistého projektu ve VS a postupně jej budeme na cvičeních vytvářet.
- Nejde jen o to "dopsat někam kousek kódu" nebo využít AI.
- Snahou není jen "něco zobrazit", ale porozumět tomu jak to funguje.
- Nedílnou součástí práce bude i refactoring, optimalizace, testování atd.
- Na cvičeních se budeme domlouvat na společných pravidlech týkající se projektu ZPG.

Mezi nejčastěji používaná rozhraní pro programování grafických aplikací, tzv. API (Application Programming Interface) patří:

- **OpenGL** - (Khronos Group) vyšší úroveň abstrakce, méně efektivní, jednodušší, vhodné pro začátečníky.
- **Vulkan** - (Khronos Group) nástupce OpenGL, nízkoúrovňové API, navržen pro vysoký výkon na moderním vícevláknovém hardwaru.
- **DirectX** - (Microsoft) obdobně jako Vulkán je DirectX 12 nízkoúrovňové API, optimalizované pro moderní více jádrové procesory a grafické karty. Primárně pro Windows a Xbox.

Není OpenGL zastaralé?

Hello World (Triangle)



OpenGL - přibližně 100 řádků
zdrojového kódu.

- high-level API
- méně kódu
- vyšší abstrakce

Vulkan - přibližně 1000 řádků
zdrojového kódu.

- low-level API
- plná kontrola
- vyšší výkon

Vznik OpenGL

Open Graphics Library (OpenGL) je multiplatformní rozhraní (API) pro tvorbu grafických aplikací. Od roku 2006 spadá pod Khronos Group.

- první vydání v červnu 1992 (Silicon Graphics);
- zaměřen čistě na vykreslování (neřeší ovládání, okna atd.);
- během vývoje postupně upravován (fixní a programovatelný);
- aktuální verze 4.6 (2017);

OpenGL Shading Language (zkráceně GLSL) je programovací jazyk pro psaní shaderů.

Vulkan - nástupce OpenGL, nízkoúrovňové API pro moderní grafické karty a procesory (paralelizace). Spadá pod Khronos Group. Aktuálně Vulkan 1.4 (prosinec 2024).

- zjistit typ paměti, alokovat VkDeviceMemory, namapovat, nahrát data;
- synchronizace mezi frontami (fences, semaphores);
- pro příkazy command buffery (zámkы, koordinace);
- shadery ve formátu SPIR-V (Standard Portable Intermediate Representation);

Do formátu SPIR-V lze kompilovat i jiné jazyky, např. HLSL.

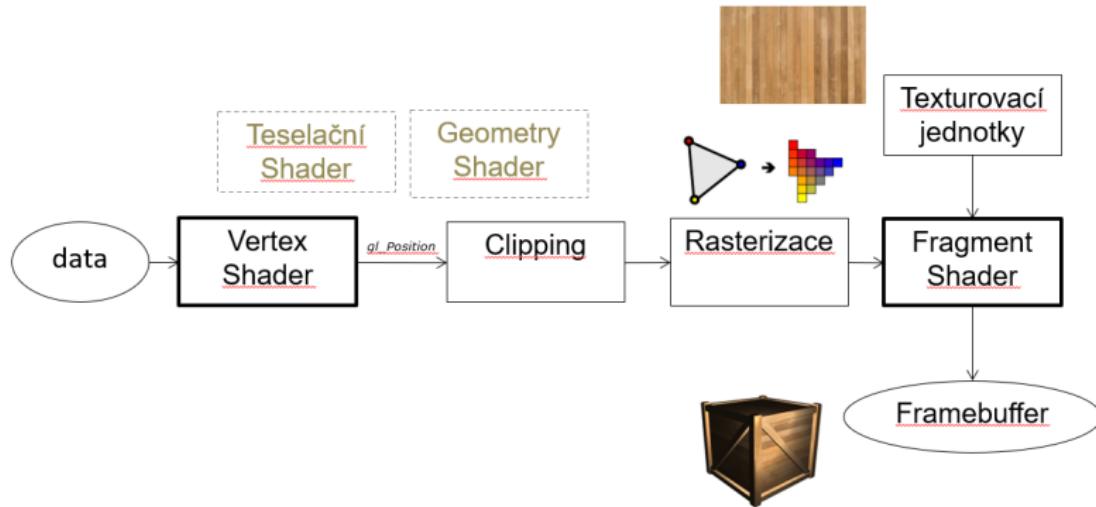
Projekt

Postupně budeme na cvičeních:

- vytvoření projektu, knihovny;
- vykreslení trojúhelníku;
- práce s shadery;
- transformace;
- kamera;
- osvětlení;
- načítání modelů (FBX);
- identifikace modelů;
- skybox;
- atd.

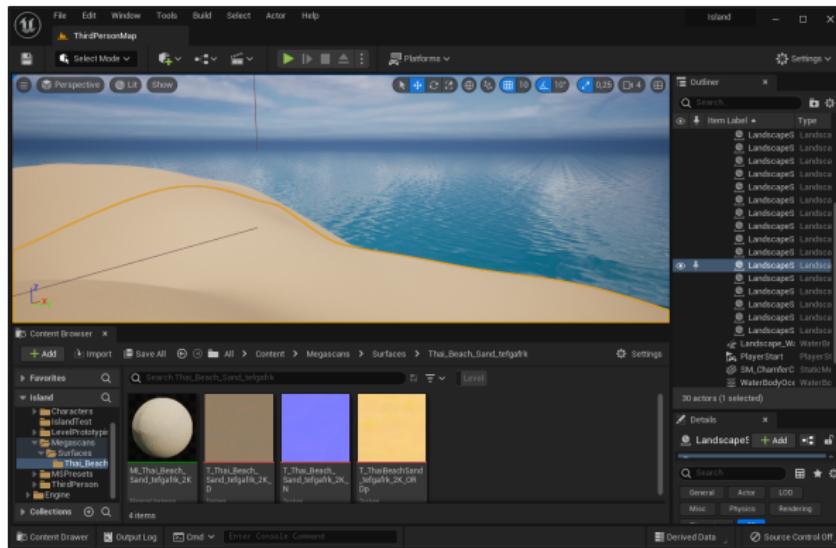
Vykreslovací řetězec

- Vertex Shader - převážně k aplikaci transformací (`gl_Position`)
- Clipping - ořezání neviditelných částí
- Rasterizace - rozklad těles na fragmenty (vektor-rastr)
- Fragment Shader - barvení, hloubka apod.



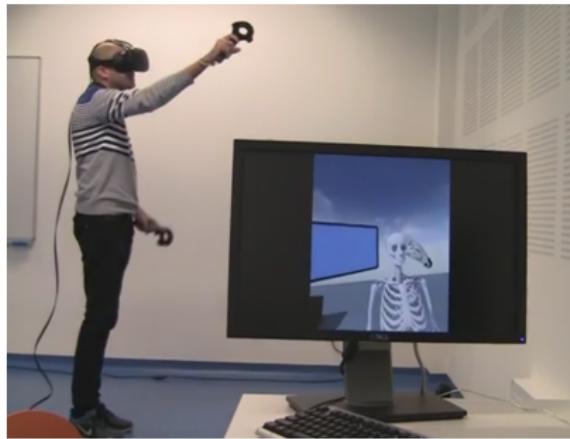
Unreal Engine

Proč si implementovat vlastní vykreslovací řetězec, když lze profesionální 3D aplikace vyvíjet i jinak?

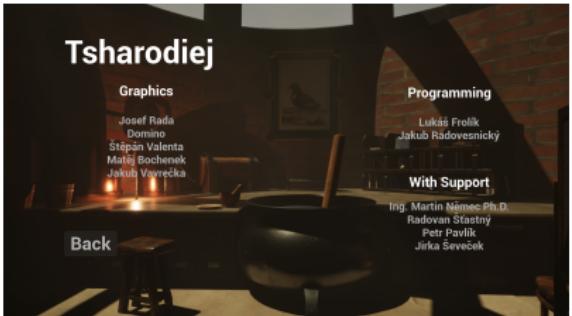
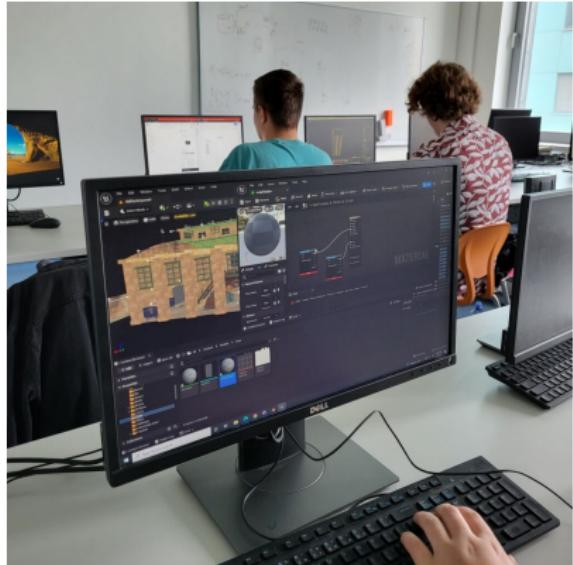


Co nestihneme?

Počítačová grafika má spoustu zajímavých částí, ale vše nestihneme.



Projekty na středních školách



Jan Polach a Jiří Vašica (<https://peruncreative.com>)

- Hobo: Tough Life
- Kromlech



Mall Craze

Milan Timčenko



Datum vydání: 19. září 2023

Kvark

Radovan Šťastný, Petr Pavlík, Perun Creative



Datum vydání: 2. června 2023

Učebna EB408



Processor: 13th Gen Intel(R) Core(TM) i7-13700KF, 3400 MHz,
16 Core(s), 24 Logical Processor(s). RAM: 64.0 GB. Graphics
card: NVIDIA GeForce RTX 4080.

Počítačová grafika

Počítačová grafika je obor informatiky, který se od svých počátků v 70. letech rozvinul do samotné vědní disciplíny.

- Vývoj je úzce propojen s výpočetními a zobrazovacími vlastnostmi počítačů (velký rozvoj v posledních letech).
- Počítačová grafika pronikla do různých odvětví (film, lékařství, vojenství, stavebnictví, projektování, modelování, apod.).
- Dnes se stává již nepostradatelnou součástí.



Různá rozdělení

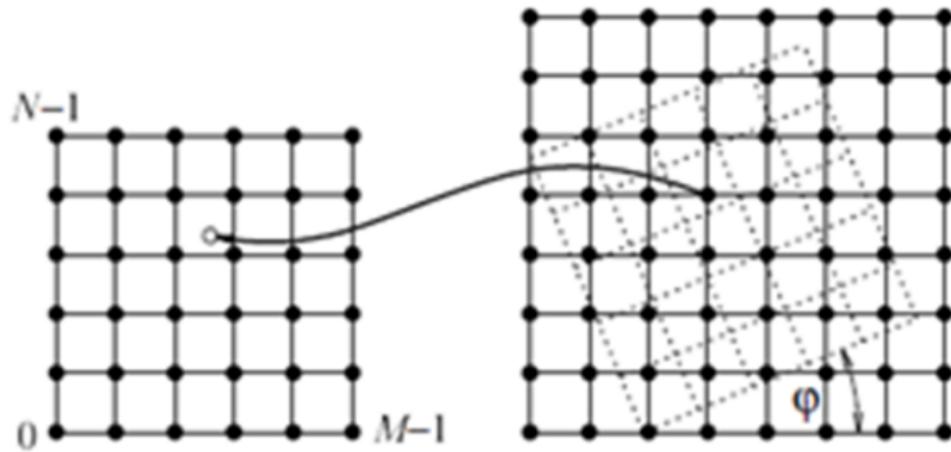
Počítačovou grafiku můžeme dělit podle různých kritérií:

- podle dimenze - 2D grafika, 3D grafika.
 - podle popisu - rastrová grafika, vektorová grafika;
 - podle rychlosti - real-time rendering, fotorealistický rendering.



Afinní transformace - rotace

Diskrétní obraz (DZO - digitální zpracování obrazu)

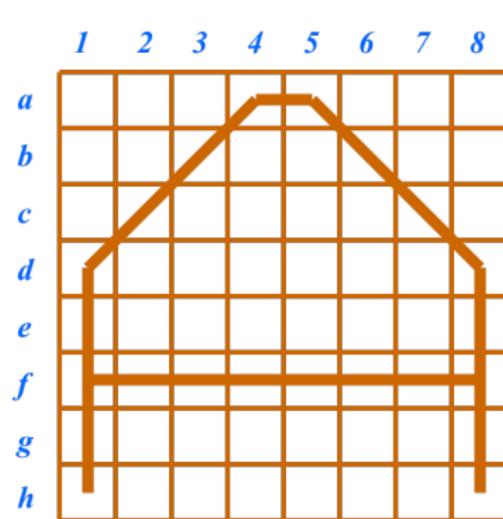
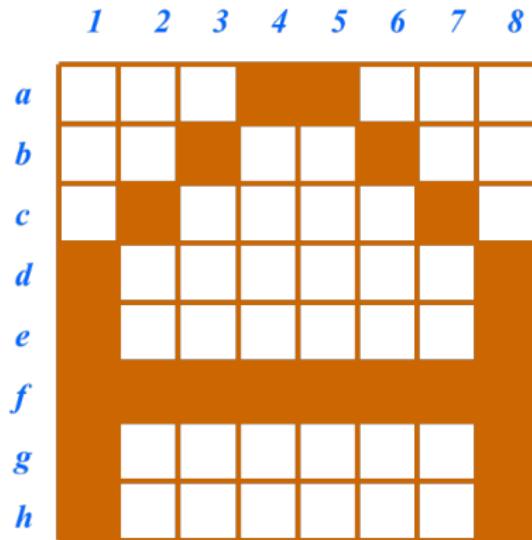


$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + (N-1) \begin{bmatrix} -\sin(\varphi) \cos(\varphi) \\ -\sin^2(\varphi) \end{bmatrix}$$

Rastrový vs. vektorový popis

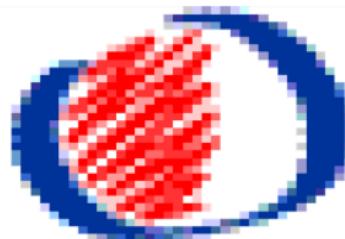
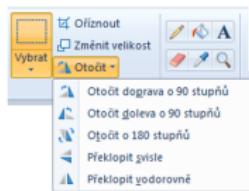
Rastrový formát - obrázek je popsán pomocí jednotlivých barevných bodů (pixelů).

Vektorový formát - obraz je složen z matematicky definovaných objektů (bod, kružnice, úsečka, křivka atd.).

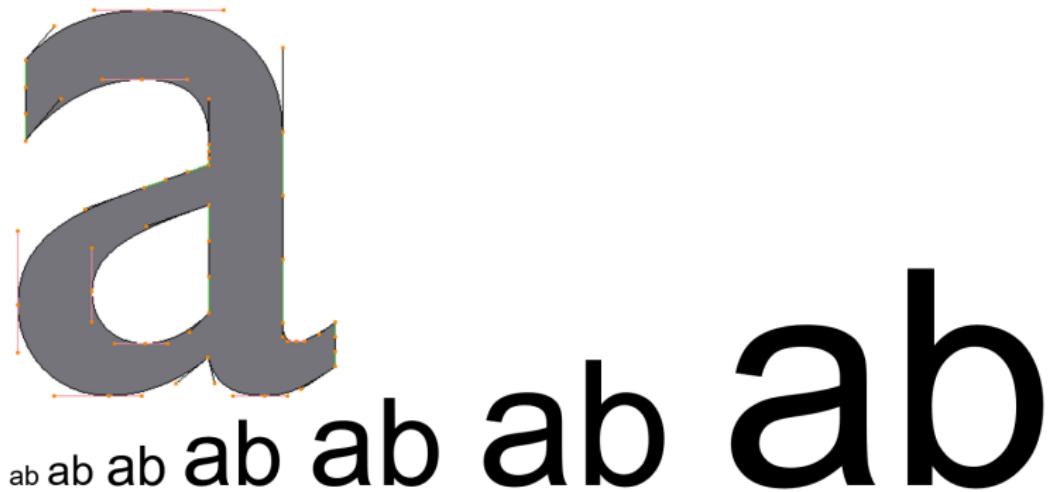


Rastrový vs. vektorový obraz

Srovnání výhod a nevýhod.



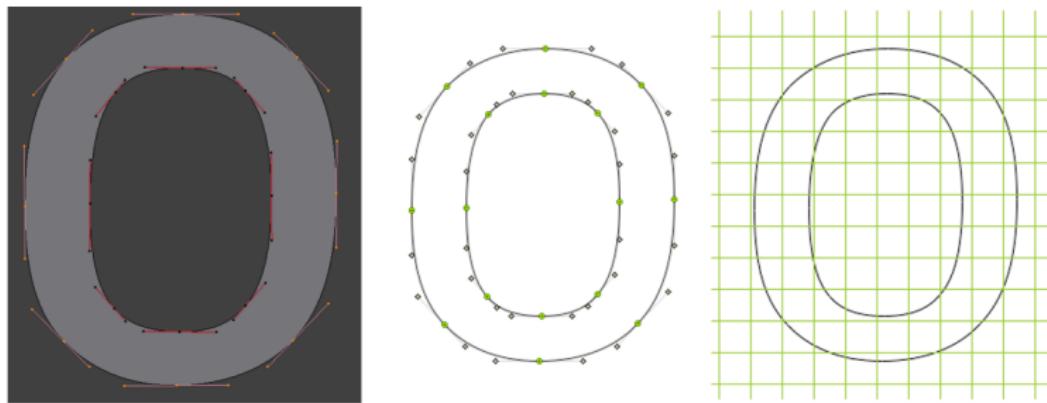
Jak jsou řešené fonty?



TrueType fonty - kvadratické Bézierovy křivky
PostScriptové fonty - kubické Bézierovy křivky

Rasterizace

Převod z vektorového popisu (obrazu) na rastrový popis (mřížku).
Rastr musí mít zadanou velikost.



Vektor

Neformálně je **vektor** prvek vektorového prostoru, který má velikost a směr.

$$\vec{v} = (v_1, \dots, v_n)$$

Velikost vektoru, někdy také nazývaná norma vektoru (Euklidovská norma) $\|\vec{u}\|$ vektoru $\vec{u} \in \mathbb{R}^n$ je dána jako:

$$\|\vec{u}\| = \sqrt{u_1^2 + \dots + u_n^2}.$$



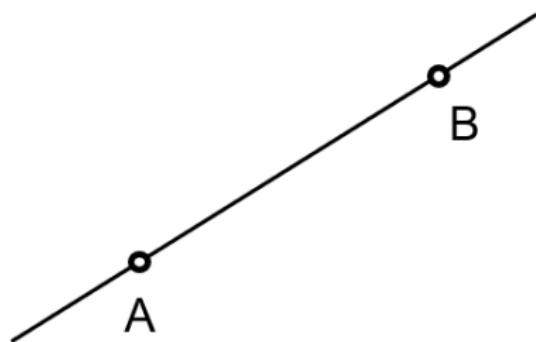
Parametrické vyjádření přímky

Základní rovnice pro popis přímky v rovině nebo v prostoru.

$$X(t) = A + \vec{u}t$$

$$X(t) = A + (B - A)t$$

$$X(t) = (1 - t)A + tB$$



$$\begin{aligned}x(t) &= a_x + \vec{u}_x t \\y(t) &= a_y + \vec{u}_y t\end{aligned}$$

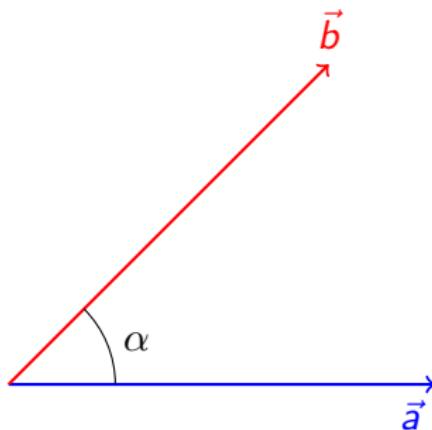
Jak je zadaná přímka, úsečka, polopřímka?

Skalární součin

Definuje se mezi dvěma vektory a zachycuje vztah mezi velikostí těchto vektorů a jejich úhlem.

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos \alpha$$

$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$$

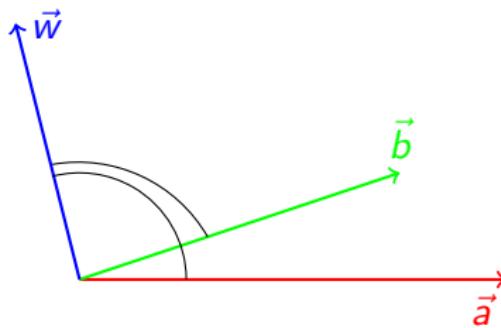


Vektorový součin

Vektor kolmý k vstupním vektorům s velikostí, která je rovna obsahu rovnoběžníku, který oba vektory určují.

$$\vec{a} \times \vec{b} = \begin{vmatrix} w_1 & w_2 & w_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$$

$$\vec{a} \times \vec{b} = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$



- J. Žára a kol.: Počítačová grafika. Computer Press, 2005, ISBN 80-251-0454-0
- E. Sojka, M. Němec, T. Fabián: Matematické principy PG. Ostrava
- Další informace: <http://lms.vsb.cz>

Dotazy?