

**Odpovědi na kontrolní otázky z opory předmětu
Základy počítačové grafiky**

... a něco navíc ;)

Petr Zemek, xzemek02@fit.vutbr.cz

10. května 2007

Části textu zde uvedeného pocházejí ze studijní opory předmětu IZG a autor nemá žádný zájem na jakémkoliv možném porušování autorských práv - tento text slouží pouze pro studijní účely. Odkaz na studijní oporu a jiné použité materiály je uveden v referencích. Veškeré informace zde uvedené jsou bez jakékoliv záruky a nemusí být správné.

Obsah

1 Úvod	3
2 Základní principy počítačové grafiky	3
3 Barvy a barevné modely	4
4 Rasterizace vektorových entit ve 2D	8
5 Vyplňování 2D oblastí	10
6 Ořezávání vektorových 2D objektů	13
7 Geometrické transformace	16
8 2D křivky	20
9 Datová reprezentace 3D objektů	21
10 Řešení viditelnosti 3D objektů	24
11 Osvětlení a stínování 3D objektů	24
12 Texturování 3D objektů	25
13 Realistické zobrazení 3D objektů	30
14 Antialiasing, obraz jako vícerozměrný signál	31
15 Grafické formáty	31
16 Základy grafických 3D API	32

1 Úvod

V tomto dokumentu se nachází odpovědi na otázky uvedené ve studijní opoře k předmětu Základy počítačové grafiky (IZG) [1]. Snažil jsem se o správnost, ale nic není dokonalé, takže prosím omluvte případné chyby a překlepy. Dodatečné informace, náměty na zlepšení atd. mi můžete poslat na můj email uvedený na titulní straně. Jako zdroj informací jsem bral [1, 2, 3, 4, 5]. Přeju všem pohodové studium a úspěšné zvládnutí zkoušky.

2 Základní principy počítačové grafiky

1. Co je to Počítačová grafika?

Počítačová grafika se zabývá analýzou (interpretací) nebo tvorbou (syntézou, generováním) grafické obrazové informace.

2. Jaké dva základní druhy počítačové grafiky rozlišujeme, z hlediska reprezentace dat?

Rastrovou a vektorovou grafiku.

3. Co je podstatou rastrové počítačové grafiky?

Zpracovávané (zobrazované) informace popisujeme diskrétně ve formě rastrové matice. Tento popis dat získáme: manuálně, syntézou (generováním nebo převodem z jiného popisu) nebo snímáním (kamerou atd.). Jeden prvek matice nazýváme pixel.

4. Co je podstatou vektorové počítačové grafiky?

Popisujeme (ukládáme, definujeme) zpracovávané informace ve formě skupiny vektorových entit (úsečky, kružnice, křivky, polygony, atd.). Tento popis získáme: manuálně nebo syntézou (generováním nebo převodem z jiného popisu)

5. Která z těchto dvou základních druhů počítačové grafiky je lepší?

Bez bližší specifikace kritéria "lepší" nelze rozhodnout, každá z variant má své pro a proti a záleží na konkrétní situaci.

6. Jakým způsobem probíhá zobrazení v současné počítačové grafice?

Vektorová data se převádní pomocí rasterizace na rastrové zobrazení (rastrová data se ze zřejmých důvodů převádět nemusí). Naše oči (vidění) jsou založeny na rastrovém principu.

3 Barvy a barevné modely

1. Co je to barva?

Barva je subjektivní vjem vyvolaný působením světla daného spektra šířeného ze zdroje, které se odráží od povrchu objektu (podle jeho objektivních vlastností) a dopadá na sítnici našeho oka. Oko pak předává získanou informaci do mozku, který ji interpretuje jako barvu odpovídající charakteristickým rozsahům vlnových délek světla odraženého od povrchu objektu.

2. Co je podstatou aditivního míchání barev?

Aditivní míchání je charakteristické pro práci se světlem, u zařízení, které vyzařují světlo (monitory, projektory, displeje, kamery, atd.). Složením všech tří základních barev při jejich maximální intenzitě dostáváme bílou barvu. Aditivnímu skládání barev odpovídají základní barvy červená (red), zelená (green) a modrá (blue) a barevný model RGB.

3. Co je podstatou subtraktivního míchání barev?

Subtraktivní míchání barev je charakteristické pro práci s pigmenty u zařízení, které tisknou barvy (tiskárny, plotry, atd.). Složením všech tří základních barev při jejich maximální intenzitě dostáváme černou barvu. Aditivnímu skládání barev odpovídají základní barvy tyrkysová (cyan), fialová (magenta), žlutá (yellow) a barevný model CMY.

4. Jaká barva vznikne smícháním maximálních intenzit základních složek při aditivním míchání barev?

Bílá.

5. Jaká barva vznikne smícháním maximálních intenzit základních složek při subtraktivním míchání barev?

”Skoro” černá barva (nelze získat ”úplně” černou barvu).

6. Jaký je vztah barevných modelů RGB a CMY?

Převod z RGB na CMY je definován jako doplněk:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

7. Jaký je vztah barevných modelů CMY a CMYK?

V modelu CMYK je oproti CMY přidána další složka - černá barva (black). Význam je především u zobrazovacích zařízení, protože pouze pomocí CMY nejsme schopni vytvořit ”čistě” černou barvu. Dále u tiskáren tímto šetříme barvami (tonery/inkoustem).

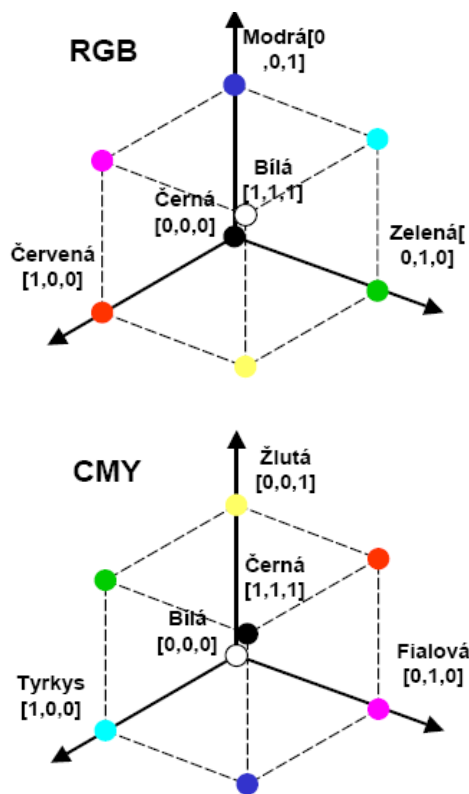
8. Jaký význam má použití barevných modelů HSV a HLS?

Jedná se o uživatelsky orientované modely.

HSV umožňuje intuitivně pracovat s barvou, vybírat sytost (saturation), barevný tón (hue) a světlost (values). Přímou ovládat tyto vlastnosti barvy není v jiných modelech možné.

HLS umožňuje intuitivně pracovat s barvou, vybírat sytost (saturation), barevný tón (hue) a jas (lightness). Přímou ovládat tyto vlastnosti barvy není v jiných modelech možné.

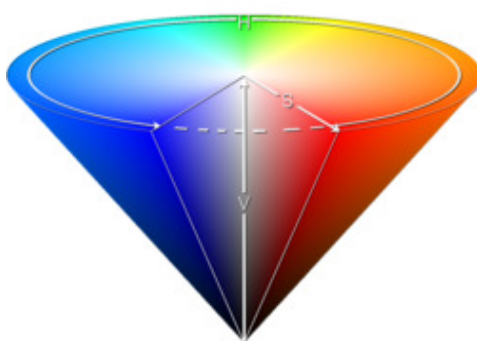
9. Nakreslete schéma barevných modelů RGB a CMY, v čem se liší?



Liší se hodnotami (barvami) ve vrcholech (vlastně ve všech bodech krychle).

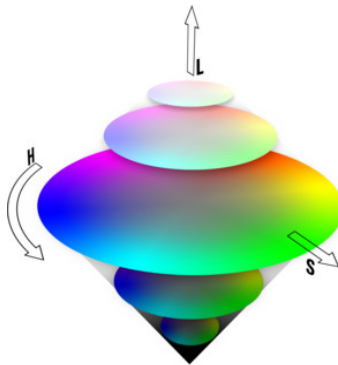
10. Nakreslete schéma barevného modelu HSV.

Viz. opora - str. 21.



11. Nakreslete schéma barevného modelu HLS.

Viz. opora - str. 20.



12. Jaký význam má chromatický diagram (CIE)?

Mezinárodní standard pro porovnávání barev a kalibraci barevných zařízení. Ostatní barevné modely můžeme brát jako podmnožinu chromatického diagramu nebo se dají takto zobrazit.

13. Proč je potřeba provádět redukci barevného modelu?

V mnoha případech nemá výstupní zařízení takový barevný rozsah, kterého jsme schopni dosáhnout (např. při 24b tj cca 16 miliónů barev). Dále také z důvodů komprese obrazu.

14. Jakým způsobem probíhá redukce barevného obrazu na stupně šedi?

Stupně šedi jsou většinou vzorkovány na 8b, což dává 256 úrovní šedi. Pro převod každého pixelu použijeme empirický vztah:

$$I = 0.299R + 0.587G + 0.114B$$

Výslednou intenzitu (I) uložíme do všech složek barvy daného pixelu (RGB).

15. Co je podstatou ditheringu?

Dithering je způsob převodu stupňů šedi (256) na černou a bílou barvu nahrazením původního rozsahu hodnot pixelů obrazu vhodnou distribucí černých a bílých bodů, které mají integrálně vizuálně maximálně odpovídající podobu. Nedochozí při tom ke zvětšení rozlišení obrazu. Význam ditheringu je v při zobrazení na monitorech a displejích, které mají omezené rozlišení.

16. Co je podstatou halftoningu?

Halftoning je způsob převodu stupňů šedi (256) na černou a bílou barvu nahrazením každého původního pixelu s rozsahem hodnot vhodnou distribucí černých a bílých bodů, které mají integrálně vizuálně maximálně odpovídající podobu. Dochází při tom ke zvětšení rozlišení obrazu. Význam halftoningu je především při tisku, kde je ve srovnání s monitory mnohem větší rozlišení. Můžeme proto kompenzovat ztrátu barevné informace zvětšením rozlišení.

17. Co je můžeme realizovat halftoning?

Trošku nechápu otázku, ale zřejmě tím je myšleno např. maticové rozptýlení, kdy danou distribuční matici "přiložíme" do každého pixelu obrazu (rozlišení se zvětší).

18. Jak pracuje redukce barevného prostoru prahováním?

Jedná se o metodu převodu obrazu ve stupních šedi na černou a bílou. Tato metoda je založena na porovnání všech pixelů obrazu s danou hodnotou prahem (threshold). Všechny pixely, které mají hodnotu pod daným prahem, budou černé, ostatní bílé. Je to metoda jednoduchá a rychlá, ale relativně destruktivní pro výsledný obraz. Pro určité typy obrazů, s výrazným kontrastem oblastí, však může dávat uspokojivé výsledky.

19. Jak pracuje redukce barevného prostoru náhodnou distribucí?

Jedná se opět o metodu převodu obrazu ve stupních šedi na černou a bílou. Tato metoda je založena na porovnání všech pixelů obrazu s hodnotou prahu, která je pro každý pixel vždy znova náhodně generována. Rozsah náhodného generátoru odpovídá rozsahu hodnot obrazu. Pixely, které mají hodnotu pod daným náhodným prahem, budou opět černé, ostatní bílé. Je to opět metoda jednoduchá a rychlá, která se velmi osvědčuje pro obrazy s velkými konstantními plochami nebo pozvolnými barevnými přechody.

20. Jak pracuje redukce barevného prostoru maticovou distribucí?

Jedná se o další metodu převodu obrazu ve stupních šedi na černou a bílou. Tato metoda je založena na porovnání pixelů obrazu s odpovídajícími hodnotami vhodné distribuční matice. Pixely, které mají hodnotu pod příslušnou hodnotou matice budou černé, ostatní bílé. Je možné tuto metodu aplikovat jako dithering nebo halftoning.

21. Jak pracuje redukce barevného prostoru distribucí chyby?

Jedná se o poslední uvedenou metodu převodu obrazu ve stupních šedi na černou a bílou. Tato metoda je v základu založena na metodě prahování. K tomu je přidána distribuce vznikajících chyb okolním, dosud nezpracovaným pixelům podle různých schémat (např. Floyd-Steinberg). Je-li aktuální pixel převeden na černý, pak se vzniklá chyba rovná původní hodnotě pixelu. Je-li aktuální pixel převeden na bílou, pak se vzniklá chyba rovná rozdílu maximální hodnoty obrazu a původní hodnoty pixelu.

22. Co je podstatou redukce barevného prostoru paletou 332?

Paleta 332 je jedinou uvedenou metodou pro redukci počtu barev plně barevného RGB obrazu. Tato metoda je založena na převzorkování standardního RGB 24b rozsahu, kdy máme na každý barevný kanál 8b. V paletě 332 máme 3b na zelenou a červenou barvu a 2b na modrou barvu. Celkově tak dostaneme 8b rozsah celého obrazu. Při relativně rovnoměrném rozložení barev v obrazu dává tato metoda poměrně dobré výsledky.

4 Rasterizace vektorových entit ve 2D

1. Co je podstatou procesu rasterizace vektorových entit?

Převod vektorové reprezentace dat na jejich rastrovou formu s cílem dosáhnout maximální možnou kvalitu a zároveň rychlost výsledného zobrazení.

2. Proč je důležité, aby rasterizační metody pracovaly maximálně efektivně a s použitím celočíselné aritmetiky?

Maximálně efektivně z důvodu využití zdrojů (procesor, paměť, ...), protože většinou je třeba rasterizaci provádět na velmi vysokém počtu (velkých) objektů. Celočíselná aritmetika je jednodušší na hardwarovou implementaci, než aritmetika s pohyblivou řadovou čárkou (a tím pádem i rychlejší zpracování).

3. Jaký je základní princip algoritmu DDA pro rasterizaci úsečky?

Tento algoritmus pracuje s čísly v pohyblivé řadové čárce. Úsečka se bude vykreslovat postupně po jednotlivých pixelech ve směru od počátečního bodu P_1 ke koncovému bodu P_2 podle následujících rekurentních vztahů:

$$\begin{aligned}x_{n+1} &= x_n + dx, & dx &= 1 \\y_{n+1} &= y_n + dy, & dy &= k = \frac{y_2 - y_1}{x_2 - x_1}\end{aligned}$$

Ve směru osy X postupujeme po jednotlivých pixelech s přírůstkem $dx = 1$. Ve směru osy Y je přírůstek dán velikostí směrnice úsečky (nutné zaokrouhlit).

4. Jaký je rozdíl mezi algoritmem DDA pro úsečku a Vykreslováním úsečky s hlídáním chyby?

Neprovádíme výpočet skutečné souřadnice v ose Y , ale pracujeme s hodnotou relativní odchylky (chyby), aktuální celočíselné souřadnice od skutečné souřadnice v ose Y . Jestliže tato chyba překročí hodnotu 0.5, přesuneme se na další řádek, na další celočíselnou souřadnici v ose Y . Po každém posunu na nový řádek je hodnota chyby snížena o hodnotu 1.0, abychom stále pracovali s relativní odchylkou od aktuálního řádku.

5. Jaký je základní princip Bresenhamova algoritmu pro rasterizaci úsečky?

Tento algoritmus pracuje pouze s celými čísly. Úsečku budeme opět vykreslovat postupně po jednotlivých pixelech ve směru od počátečního bodu P_1 ke koncovému P_2 . Ve směru osy X postupujeme po jednotlivých pixelech s přírůstkem $dx = 1$. O posunu ve směru osy Y rozhodujeme podle znaménka, nikoli hodnoty, prediktoru p_i . Jestliže je hodnota prediktoru $p_i < 0$, pak nový pixel leží na aktuální úrovni y_i . Pokud je však hodnota prediktoru $p_i \geq 0$, pak se nový pixel posune na další úroveň $y_i + 1$.

6. Proveďte kompletní odvození vztahů pro Bresenhamův algoritmus rasterizace úsečky.

Viz. opora str. 31.

7. Jaký je základní princip algoritmu pro rasterizaci kružnice po bodech?

Nejjednodušší způsob vykreslení kružnice, ale z důvodů nízké efektivity (především výpočet odmocniny), se nepoužívá. Následující vztah lze odvodit z rovnice kružnice a jedná se o výpočet souřadnic v ose Y (R je poloměr kružnice):

$$y = \sqrt{R^2 - x^2}$$

V ose X se pohybujeme ve směru hodinových ručiček po jenom pixelu od hodnoty 0 do okamžiku, kdy se souřadnice $x = y$ (pro úhel 45 stupňů, kdy je možné použít symetrie). Vypočtené souřadnice v ose Y se musejí zaokrouhlovat na nejbližší celá čísla.

8. Jaký je základní princip algoritmu pro vykreslení kružnice jako N-úhelník?

Jedná se o variantu algoritmu DDA pro kružnici. Kružnici budeme vykreslovat jako N-úhelník pomocí úseček. Vycházíme přitom z rotačních transformací bodů. Prostřednictvím následujících rekurentních vztahů:

$$\begin{aligned}x_{n+1} &= x_n \cos \alpha - y_n \sin \alpha \\y_{n+1} &= x_n \sin \alpha + y_n \cos \alpha\end{aligned}$$

budeme počítat souřadnice X, Y pro daný konstantní přírůstek úhlu $\alpha (\sim 1/R)$. Hodnoty $\cos \alpha$ a $\sin \alpha$ mohou být předpočítány dopředu pouze jednou jako konstanty. Výsledné souřadnice musíme opět zaokrouhlit na nejbližší celá čísla.

9. Jaký je základní princip Midpoint algoritmu pro rasterizaci kružnice?

Jeho princip je v podstatě stejný s Bresenhamovým algoritmem (také pracuje pouze s celými čísly). Rozdíl je ve stanovení kritéria pro určení prediktoru p_i . Kružnici budeme vykreslovat postupně po jednotlivých pixelech ve směru hodinových ručiček pouze pro její 1/8. Zbývající body získáme pomocí symetrie. Ve směru osy X se pohybujeme od hodnoty 0 do okamžiku, kdy se souřadnice $x = y$ (pro úhel 45 stupňů), s přírůstkem $dx = 1$. Ve směru osy Y klesáme od hodnoty R (poloměr). O posunu ve směru osy Y rozhodujeme opět podle znaménka prediktoru p_i .

Předpokládejme, že jsme v celočíselné pozici $[x_i, y_i]$ a rozhodujeme, jestli následující pixel leží na pozici $[x_i + 1, y_i]$ nebo $[x_i + 1, y_i - 1]$. Toto rozhodnutí provedeme podle polohy středního bodu $[x_i + 1, y_i - 1/2]$ (midpointu) vůči kružnici (uvnitř nebo vně).

Jestliže je hodnota prediktoru $p_i < 0$, pak je midpoint uvnitř kružnice a nový pixel leží na aktuální úrovni y_i . Pokud je však hodnota prediktoru $p_i \geq 0$, pak je midpoint vně kružnice a nový pixel se posune na další úroveň $y_i - 1$.

10. Proveďte kompletní odvození vztahů pro Midpoint algoritmu rasterizace kružnice.

Viz. opora str. 34.

11. Jaký je základní princip Midpoint algoritmu pro rasterizaci elipsy?

Princip je v podstatě stejný jako u algoritmu Midpoint u kružnice. Protože je elipsa pouze 4x symetrická, musíme vypočítat body pro celou její 1/4. Zbývající body získáme symetrií. Během výpočtu bodů 1/4 se však mění velikost parciálních derivací v ose X a v ose Y . Proto rozdělíme 1/4 elipsy na dvě oblasti. V oblasti I je větší parciální derivace v ose Y a řídící osou proto bude osa X ($x_{i+1} = x_i + 1$). V oblasti II je větší parciální derivace v ose X a řídící osou proto bude osa Y ($y_{i+1} = y_i - 1$). Přejít nastává v okamžiku rovnosti parciálních derivací, kdy má směrnice normály elipsy hodnotu 1 (úhel 45 stupňů).

12. Proveďte kompletní odvození vztahů pro Midpoint algoritmu rasterizace elipsy.

Viz. opora str. 36.

5 Vyplňování 2D oblastí

1. Jaká je souvislost vyplňování 2D oblastí s zobrazování 3D polygonálních modelů?

V současné době je 3D zobrazování založeno na polygonálních modelech, ve kterém jsou objekty popsány množinou mnohoúhelníků. Většinou se používají sítě trojúhelníků. Proto je třeba umět rychle a efektivně zobrazovat vyplněné plošné (2D) objekty.

2. Jakým způsobem jsou popsány vyplňované oblasti?

Většinou se jedná o popis hranic.

3. Jakým způsobem mohou být definovány hranice vyplňované oblasti?

Rozlišujeme:

- Rastrové oblasti – hranice je popsána rastrově, v rastrové matici nastavením bodů hranice nebo barvou vyplňované oblasti
- Vektorové oblasti – hranice je popsána seznamem vektorových entit (úseček, kruhových oblouků, křivek atd.)

4. Jaký je princip řádkového vyplňování?

Základní myšlenkou algoritmu řádkového vyplňování je procházení dané oblasti Ω po jednotlivých řádcích výsledného rastru komplexně přes celou oblast. Úseky řádků ležící uvnitř oblasti jsou pak vyplněny barvou nebo vzorem oblasti. Vyplňovaná oblast Ω je popsána seznamem hraničních entit (hran) $\Omega = \{e_1, e_2, \dots, e_n\}$.

5. Jakým způsobem se můžeme vypořádat s lichým počtem průsečíků řádku při řádkovém vyplňování?

Pro řešení problému lichého počtu průsečíků v řádku při protínání vrcholu hrany máme k dispozici dvě řešení:

- Zkrácení dolního okraje všech hran hranice oblasti o 1 ve směru osy Y. Vodorovné hrany se vypouštějí. Při tom může dojít ke zkrácení dolního okraje oblasti (hrana AB). Proto je nutné nakonec zvlášť vykreslit obrys oblasti rasterizací hranic.
- Analýza typu extrémů protínaných vrcholů během generování průsečíků řádků a následné generování správného počtu průsečíků. Tím se však implementace algoritmu komplikuje a zpomaluje.

6. Jakým způsobem můžeme optimalizovat řádkový vyplňovací algoritmus?

Jedná se o *řádkové vyplňování se seznamem aktivních hran*. Tento seznam obsahuje pouze ty hrany, které mají průsečík s právě zpracovávaným řádkem. Hrany se v tomto seznamu udržují podle souřadnic x . Projeví se zde především časová úspora metody při přechodu na nový řádek - ve většině případů zůstává uspořádání průsečíků beze změny.

7. Jaký je princip Inverzního řádkového vyplňování?

Odstraňuje nutnost třídít průsečíky pro každý řádek (u řádkového vyplňování se jedná o zpomalování, proto je třeba ho odstranit). Základní myšlenkou algoritmu inverzního řádkového vyplňování je procházení dané oblasti po jednotlivých hranách $e_i \in \Omega$. Od každé hrany jen napravo po řádcích provedeno vyplnění s inverzí aktuálních hodnot pixelů. Vyplňovaná oblast Ω je opět popsána seznamem hraničních entit (hran) $\Omega = \{e_1, e_2, \dots, e_n\}$.

8. Jaké jsou nevýhody (slabosti) Inverzního řádkového vyplňování?

První nevýhodou metody je zkrácení hranic vyplněné oblasti. Při prvním vyplnění od aktuální hrany s inverzí přes nevyplněné pozadí se hranice vykreslí. Při vyplnění od aktuální hrany s inverzí přes již vyplněné pozadí se hranice naopak smaže. Pro vyplnění celé oblasti je proto potřeba provést vykreslení obrysu oblasti rasterizací jeho hranic.

Druhou nevýhodou metody je ovlivňování okolí vyplněné oblasti napravo od ní až do její maximální souřadnice X_{max} . Pokud provádíme vyplňování přímo do obrazové paměti (framebufferu), pak může dojít k porušení již dříve existující obrazové informace.

9. Jaký je principiální rozdíl mezi řádkovým vyplňováním a Inverzním řádkovým vyplňováním?

Není nutné třídit průsečíky pro každý řádek a inverzní řádkové vyplňování provádí vyplnění s inverzí hodnot pixelů, při použití řádkového vyplňování k žádné inverzi nedochází.

10. Jaký je princip Pinedova vyplňovacího algoritmu?

Tento algoritmus pracuje pouze s konvexními mnohoúhelníky. Základní myšlenkou Pinedova algoritmu vyplňování je rozdělení roviny dané oblasti Ω na poloroviny hran $e_i \in \Omega$. Všechny body roviny oblasti Ω , které leží na kladné straně všech polorovin hran e_i jsou pak uvnitř oblasti Ω .

11. Odvoďte hranovou funkci pro Pinedův vyplňovací algoritmu.

Viz. opora str. 44.

12. Jaké máme k dispozici strategie procházení řádků u Pinedova vyplňovacího algoritmu?

- procházení opsaného obdelníka po řádcích
- procházení od maximálního vrcholu s obratem a přechodem na další řádek
- procházení po řádcích podél svislého plotu, který spustíme ze středu opsaného obdelníka nebo z prostředního vrcholu (možnost paralelizace)

13. Jaký je princip Semínkového vyplňování?

Základní myšlenkou semínkového vyplňování je obarvování sousedních pixelů (4 nebo 8) od daného semínka. Obarvené pixely se rekurzivně stávají dalšími semínky. Pixely, které představují hranici oblasti nebo nemají vlastnosti oblasti se vyřazují.

14. Jaké typy obarvování okolí se semínka znáte?

- hraniční – oblast je definována spojitou hranicí z pixelů dané barvy.
- záplavové – Oblast definována spojitou množinou vnitřních pixelů dané barvy.
- měkké – oblast je definována spojitou hranicí z pixelů daného rozptylu barev.
- prahové – oblast je definována spojitou množinou vnitřních pixelů jejichž barva leží v daném rozmezí okolo barvy semínka.

15. Jaké druhy spojitosti rastrové hranice oblasti znáte?

- 4-okolí – každý pixel má 4 sousedy (po hraně). Při vyplňování testujeme a obarvujeme tyto 4 sousedy. Hranice oblasti je uzavřená, pokud všechny její hraniční pixely sousedí po hraně.
- 8-okolí – každý pixel má 8 sousedů, po hraně a přes vrcholy. Při vyplňování testujeme a obarvujeme těchto 8 sousedů. Hranice oblasti je uzavřená, pokud všechny její hraniční pixely sousedí po hraně nebo přes vrchol.

16. Jaký je princip řádkového semínkového vyplňování?

Místo vyplňování po jednotlivých pixelech se postupuje po řádcích. Pro dané semínko se nalezne hranice vyplňovaného řádku x_L, x_P , vykreslí se úsečka a na vyšší úsečce $(y + 1)$ se najdou všechny souvislé nevyplněné oblasti a pro každou z nich se do zásobníku vloží semínko. To stejné na nižší úsečce.

17. Jaký je přínos řádkového semínkového vyplňování proti klasickému Semínkovému vyplňování?

Do zásobníku se ukládá méně semínek, takže může dojít k urychlení vyplňování.

18. Jakým způsobem může být popsána rastrová oblast pro Semínkové vyplňování?

- spojitou hranicí z pixelů dané barvy
- spojitou množinou vnitřních pixelů dané barvy
- spojitou hranicí z pixelů daného rozptylu barev
- spojitou množinou vnitřních pixelů jejichž barva leží v daném rozmezí okolo barvy semínka

6 Ořezávání vektorových 2D objektů

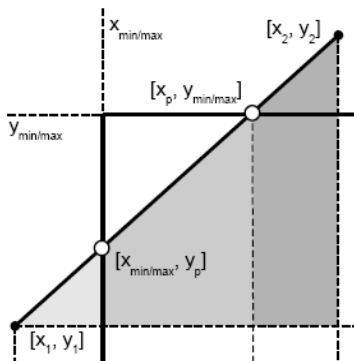
1. Proč je důležité ořezávat polygonální objekty zobrazovacím oknem před jejich vykreslením?

Je zbytečné zobrazovat to, co “není vidět”, především z výkonnostních důvodů. Po ořezání lze efektivně pracovat pouze s částí, která je v danou chvíli viditelná (potřebná).

2. Jak můžeme snadno získat průsečík úsečky s hranicí obdélníkového okna?

Vztah pro výpočet průsečíku $[x_p, y_p]$ odvodíme z podobnosti trojúhelníků.

$$\frac{x_2 - x_1}{y_2 - y_1} = \frac{x_p - x_1}{y_p - y_1}$$



Dosadíme polohy hranice okna $x_p = x_{min/max}$ pro svislé nebo $y_p = y_{min/max}$ pro vodorovné hranice okna.

$$x_p = x_{min/max} \rightarrow y_p = \frac{(x_{min/max} - x_1)(y_2 - y_1)}{x_2 - x_1} + y_1$$
$$y_p = y_{min/max} \rightarrow x_p = \frac{(x_2 - x_1)(y_{min/max} - y_1)}{y_2 - y_1} + x_1$$

3. Jaký je princip algoritmu Cohen-Sutherland pro ořezání úseček?

Rozšíření primitivního testování polohy úsečky vůči oknu. Koncové body $P1, P2$ označíme binárními kódy $C1, C2$ oblastí. Určíme průsečíky, rozdělíme úsečku na části a znovu testujeme.

4. Jakým způsobem algoritmus Cohen-Sutherland pro ořezání úseček určuje polohu úsečky vůči oknu?

Podle hodnot binární kombinace kódů koncových bodů můžeme určit potenciální polohu úsečky vůči oknu:

- $C1 \text{ or } C2 = 0 \rightarrow$ celá úsečka leží uvnitř okna
- $C1 \text{ and } C2 \neq 0 \rightarrow$ celá úsečka leží mimo okno
- $C1 \text{ and } C2 = 0 \rightarrow$ úsečka potenciálně protíná hranice okna

5. Jakým způsobem určuje algoritmus Cohen-Sutherland pro ořezání úseček průsečíky úsečky s hranicemi okna?

Pokud úsečka potenciálně protíná hranice okna, vypočteme příslušné průsečíky. Podle nich úsečku rozdělíme na části, které znovu testujeme na polohu vůči oknu. Průsečíky počítáme pouze pro ty hranice okna, kterým odpovídá nastavený bit v hodnotě binárního výrazu: $C1 \text{ or } C2$.

6. Jaký je princip algoritmu Liang-Barsky pro ořezání úseček?

Tento algoritmus pracuje v parametrickém prostoru. Úsečka dána body $P[x_P, y_P]$, $Q[x_Q, y_Q]$ a rovnicemi:

$$\begin{aligned}x &= x_P + (x_Q - x_P) \cdot u = x_P + \Delta x \cdot u \\y &= y_P + (y_Q - y_P) \cdot u = y_P + \Delta y \cdot u, \quad 0 \leq u \leq 1\end{aligned}$$

Hledáme část úsečky ležící uvnitř okna, nebo-li hodnoty parametru u .

7. Jakým způsobem algoritmus Liang-Barsky pro ořezání úseček určuje polohu úsečky vůči oknu?

Podle hodnot parametrů p_k a q_k (z rovnic viz. opora str. 53).

8. Jakým způsobem určuje algoritmus Liang-Barsky pro ořezání úseček průsečíky úsečky s hranicemi okna?

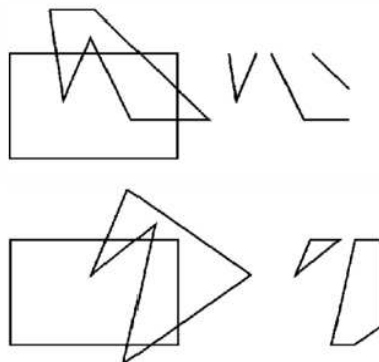
Nalezením parametrů u_R a u_S (viz. opora str. 54).

9. Je možné ořezávat polygony prostým ořezáním jejich hraničních úseček?

Ne, protože bychom ztratili informaci o části polygonu ležící uvnitř okna (musí se zachovat uzavřené oblasti).

10. Co se stane, když ořežeme polygon prostým ořezáním jeho hraničních úseček?

Nedojde k zachování uzavřené oblasti, viz. obrázek.



11. Jaký je princip algoritmu Sutherland-Hodgman pro ořezání polygonů?

Algoritmus Sutherland-Hodgman provádí ořezání polygonů obdélníkovým oknem se stranami rovnoběžnými se osami X a Y . Komplexní úkol ořezání polygonu celým oknem je zjednodušen na ořezání pouze jednou stranou okna (přímku procházející příslušnou stranou okna). Po oříznutí jednou stranou okna provedeme oříznutí druhou, třetí a čtvrtou stranou. Tak postupně po jednodušších krocích dosáhneme oříznutí celého polygonu celým oknem.

12. Jaké má algoritmus Sutherland-Hodgman pro ořezání polygonů omezení a nevýhody?

+ je relativně jednoduchý a snadno implementovatelný
– problém při generování překrývajících se hran ořezaného polygonu podél hranic a rohů okna
Ořezávaný polygon při tom může být definován pouze jednou uzavřenou smyčkou hraničních úseček. Je tedy komplikovaná realizace polygonů s vnitřními otvory.

13. Jaký je princip algoritmu Weiler-Atherton pro ořezání polygonů?

Obecné ořezání polygonů i s vnitřními otvory (vnitřní smyčky mají opačnou orientaci) obecným oknem definovaným polygonem. Jeho základem je několik pomocných seznamů vrcholů polygonu a okna včetně průsečíků. Jejich vhodným průchodem získáme části polygonu uvnitř nebo vně okna.

14. Jaké druhy polygonů je schopen zpracovat algoritmus Weiler-Atherton pro ořezání polygonů?

Obecné polygony i s vnitřními vrcholy.

7 Geometrické transformace

1. Proč jsou geometrické transformace pro počítačovou grafiku tak důležité?

Umožňují nám provádět posunutí, rotaci, změnu měřítka a zkosení jednotlivých objektů. Jsou jednou z nejvíce používaných operací v současné počítačové grafice.

2. Jaký rozměr mají transformační matice ve 2D/3D a proč?

Transformační matice mají rozměr o 1 větší, než bychom čekali - ve 2D to je matice 3x3 a ve 3D to je 4x4. Je to z důvodu jednotné práce s transformacemi, kdy za použití *homogenních souřadnic* můžeme provádět všechny geometrické transformace jednotně pomocí maticového zápisu.

3. Proč mají transformační matice pro 3D rozměr 4x4, když máme pouze tři kartézské souřadnice (x, y, z) ?

Je to z důvodu použití homogenních souřadnic, viz. odpověď výše.

4. Jak vypadá 3D homogenní souřadnice pro bod a jak pro vektor?

Homogenní souřadnice bodu ve 3D s kartézskými souřadnicemi $[x, y, z]$ je uspořádaná čtveřice $[X, Y, Z, w]$ pro kterou platí $x = X/w$, $y = Y/w$, $z = Z/w$. Bod je svými homogenními souřadnicemi určen jednoznačně. Souřadnici w nazýváme váhou bodu. Hodnota váhy je většinou $w = 1$ (v případě lineárních transformací). Vektory $\vec{v} = (x, y, z)$ reprezentujeme čtveřicí $\vec{v} = (x, y, z, w)$, kde $w = 0$.

5. Proč pro transformace zavádíme homogenní souřadnice, proč nám nestačí kartézské souřadnice?

Umožňují nám pracovat se všemi druhy základních transformací jednotně - pomocí maticového zápisu. Kdybychom používali kartézské souřadnice, pak bychom museli rozlišovat, zda provádíme posun (zde se matice sčítají) nebo zbývající transformace (zde se matice násobí).

6. Co se stane, bude-li váhová souřadnice homogenních souřadnic $w = 0$?

Potom se jedná o vektor, viz. 4. otázka.

7. Jaké jsou základní 3D afinní transformace?

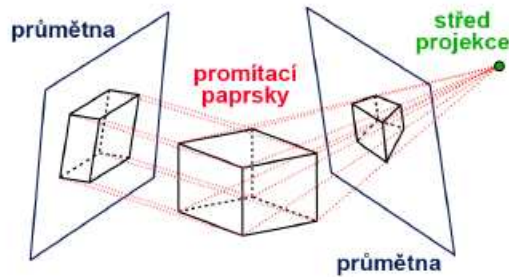
Afinní znamená, že se zachovává rovnoběžnost hran a jedná se o posunutí, rotaci, změnu měřítka a zkosení.

8. Jakým způsobem provádíme skládání transformací a k čemu je to dobré?

Při praktickém nasazení transformací si většinou nevystačíme se základními transformacemi. Naštěstí se dá každá požadovaná obecná lineární transformace rozložit na posloupnost základních transformací. Obecná transformace se pak dá vyjádřit jednou transformační maticí. Tuto obecnou transformační matici získáme násobením dílčích základních transformačních matic. Toto násobení matic musí být z pravé strany a v přesném pořadí provádění jednotlivých transformací.

9. Co je to projekce a jaké základní druhy projekcí znáte?

Projekce je transformace realizující redukci dimenze 3D prostoru na 2D prostor při zachování parametrů použitého zobrazení. Projekce provádí redukci dimenze prostoru v definovaném směru (směru pohledu), nejčastěji ve směru osy Z. Pracuje se s vrcholy projektovaných objektů a zachovává se druh jejich reprezentace (rastrová, vektorová). Rozlišujeme *paralelní* (rovnoběžnou) a *perspektivní* (středovou) projekci, viz. dále.



Paralelní (vlevo) a perspektivní (vpravo) projekce.

10. Co je to paralelní projekce a jaké její základní druhy znáte?

Paralelní projekce je lineární projekce, která zobrazuje vrcholy promítaných objektů prostřednictvím rovnoběžných paprsků. Velikost průmětů objektů není závislá na jejich vzdálenosti od průmětny. Rovnoběžnost promítaných hran je v této projekci zachována.

Podle úhlu dopadu promítacích paprsků na průmětnu rozlišujeme dva základní druhy paralelní projekce:

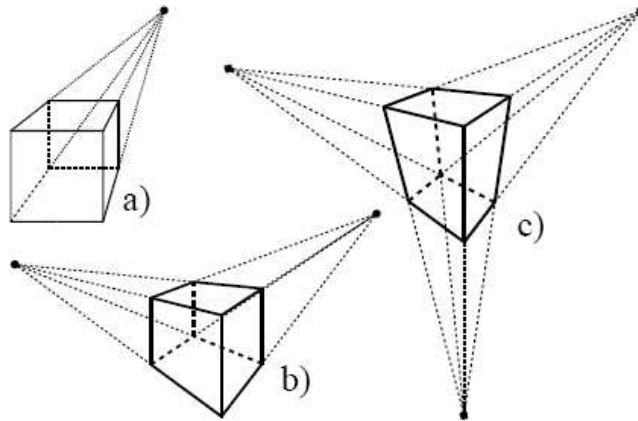
- Kolmá projekce – promítací paprsky jsou kolmé na průmětnu:
 - Mongeova projekce
 - Axonometrická projekce
- Kosoúhlá projekce – úhel dopadu paprsků na průmětnu < 90 stupňů. Rovina XY je rovnoběžná s průmětnou, osa Z je sklopena pod úhlem.
 - Kavalírská projekce
 - Kabinetní projekce

11. Co je to perspektivní projekce a jaké její základní druhy znáte?

Perspektivní projekce je nelineární neafinní projekce, která zobrazuje vrcholy promítaných objektů prostřednictvím paprsků protínajících se v jednom bodě, ve středu projekce. Střed projekce je zároveň většinou místem pozice pozorovatele. Velikost průmětů objektů je nepřímo úměrně závislá na jejich vzdálenosti od průmětny. Čím je objekt blíže průmětně, tím je jeho obraz větší a naopak. Rovnoběžnost promítaných hran není v této projekci zachována.

Průmětna může mít obecně jakoukoli polohu vůči souřadnému systému zobrazovaných objektů. Pro zjednodušení však rozlišujeme tři následující případy:

- Jednobodová perspektiva – průmětna je rovnoběžná se dvěma souřadnými osami. Hrany objektů rovnoběžné se třetí osou se sbíhají do jednoho bodu - úběžníku.
- Dvoubodová perspektiva – průmětna je rovnoběžná s jednou souřadnou osou. Hrany objektů rovnoběžné se zbylými dvěma osami se sbíhají do dvou úběžníků.
- Tříbodová perspektiva – průmětna má vůči souřadnému systému obecnou polohu. Hrany objektů rovnoběžné se souřadnými osami se sbíhají do tří úběžníků.



Druhy perspektivní projekce: a) jednobodová, b) dvoubodová, c) třibodová.

12. Jak prakticky realizujeme paralelní projekce?

Nejčastěji pomocí tzv. kolmého promítání - paprsky kolmé na průmětnu. Praktickou realizací kolmé projekce je zanedbání (nastavení na 0) souřadnice promítaných vrcholů, které odpovídají ose promítání (nejčastěji osa Z). Zbylé dvě osy pak tvoří průmětnu (XY). Tuto operaci lze snadno realizovat prostřednictvím jednoduché transformační matice (viz. opora str. 68).

13. Jak prakticky realizujeme perspektivní projekce?

Stejně jako paralelní projekci můžeme i perspektivní projekci vyjádřit prostřednictvím transformace s maticí P_{per} (viz. opora str. 71). Její odvození vychází z parametrického vyjádření promítacího paprsku.

14. Co je to pohledový objem a k čemu je to dobré?

Pohledový objem je část 3D prostoru, která obsahuje zobrazované (viditelné) objekty. Pohledový objem je ohraničen okraji okna protaženými do prostoru do tvaru hranolu (při paralelní projekci) nebo komolého jehlanu (při perspektivní projekci). V směru pohledu je objem uzavřen přední a zadní ořezávací stěnou.

15. Jaké je praktická oblast použití paralelní projekce?

Použití pro technické aplikace (CAD/CAM/CAE). Výkresová dokumentace, zobrazení 3D objektů, technická schémata, apod.

16. Jaké je praktická oblast použití perspektivní projekce?

Použití pro virtuální realitu, architekturu, hry, atd.

17. Je paralelní projekce lineární a jaké to má důsledky?

Ano, je. Zachovává rovnoběžnost hran a vzdálenost od průmětny neovlivňuje velikost průmětu. (Tady si nejsem jistý, jestli je to důsledek linearitu, možná to bude něco jiného, ale kromě toho, že se zachovávají lineární kombinace mě nic nenapadá ;x).

18. Je perspektivní projekce lineární a jaké to má důsledky?

Ne, není. Nezachovává rovnoběžnost hran a vzdálenost od průmětny ovlivňuje velikost průmětu.

19. Jak se chovají promítací paprsky v paralelní projekci?

Promítací paprsky jsou rovnoběžné a nejčastěji kolmé na průmětnu.

20. Jak se chovají promítací paprsky v perspektivní projekci?

Paprsky vycházejí z jednoho bodu (střed projekce, místo pozorovatele) a nejsou rovnoběžné.

21. Jaká je velikost obrazu promítaných objektů v paralelní projekci a proč?

Velikost obrazu je nezávislá na vzdálenosti objektu od průmětny (rovnoběžné paprsky).

22. Jaká je velikost obrazu promítaných objektů v perspektivní projekci a proč?

Velikost obrazu je dána vzdáleností objektu od průmětny (paprsky vycházejí z jednoho bodu a nejsou rovnoběžné).

8 2D křivky

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám.

1. Vyjmenujte a stručně popište požadované vlastnosti křivek.

Požadované vlastnosti křivek jsou:

- Invariance k lineárním transformacím – rotace řídicích bodů nemá vliv na tvar křivky
- Konvexní obálka – křivka leží v konvexní obálce svých řídicích bodů
- Interpolace krajních bodů – křivka prochází krajními body
- Lokalita změn – pokud změním/přidám řídicí bod, tak se křivka změní jen v dané lokální části

2. Jaké znáte druhy křivek z pohledu závislosti na zadaných bodech?

Rozlišujeme křivky:

- Interpoláčnické – křivka přímo prochází body (“proložení” bodů křivkou)
- Aproximační – neprochází body, v tomto případě tzv. *řídicí body*

3. Jaké další možné rozdělení křivek znáte?

Můžeme rozdělit křivky na:

- Racionální – *váhové koeficienty* w_i řídicích bodů, jsou invariantní vůči perspektivní projekci
- Neracionální – tvar křivky ovlivňujeme pouze změnou polohy řídicích bodů, speciální případ racionální křivky, kde $w_i = 1$, nejsou invariantní vůči perspektivní projekci

4. Jaký je rozdíl mezi Beziérovými kubikami a racionálními Beziérovými křivkami?

Pro vykreslení racionálních Beziérových křivek nelze (narozdíl od Beziérových kubik) použít algoritmus *de Casteljau* - polynomy nemají rekurentní definici.

5. Co musíme udělat pro to, aby koncové body Coonsových křivek ležely v prvním a posledním řídicím bodu?

Musíme dané řídicí body ztrojit, tzn. $P_0 = P_1 = P_2$, to stejné pro poslední bod.

6. Napište základní vlastnosti NURBS křivek.

Jedná se o zobecnění B-spline křivek. NURBS křivky jsou racionální, mají neuniformní uzlový vektor. Umožňují vkládání řídicího bodu při zachování tvaru a jsou invariantní vůči lineárním transformacím. Lze pomocí nich přesně vyjádřit např. kuželosečky.

9 Datová reprezentace 3D objektů

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám.

1. Jaký je základní rozdíl mezi manifold a nonmanifold objekty?

Manifold objekty jsou vyrobitelné, každá hrana sdílí pouze 2 stěny. Nonmanifold objekty nejsou vyrobitelné.

2. K čemu nám slouží tzv. Eulerovy rovnice?

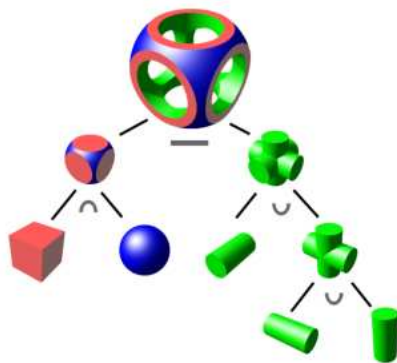
Umožňují kontrolovat topologii objektu. Platí pro manifold tělesa, kde hrana spojuje dva vrcholy, ve vrcholu jsou minimálně tři hrany a stěny se neprotínají.

3. Proč se zavádí tzv. regularizované množinové operace místo klasických?

Cílem je, aby množinové operace neměnily topologickou dimenzi objektu, takže výsledkem množinové operace s tělesy je těleso - to nám právě zajišťují regularizované množinové operace. Mezi regularizované množinové operace patří (a značí se) \cap^* , \cup^* , $-^*$.

4. Co je to konstruktivní geometrie a k čemu slouží?

Konstruktivní geometrie (CSG - constructive solid geometry) slouží k popisu 3D modelů. Objekt je popsán stromem složeným ze 3D primitiv, transformací a booleovských operací (uzly stromu). Nejsou informace o povrchu objektu \rightarrow převod na B-rep. Dále je zde možnost vzniku singularit - nutnost využívat regularizované operace.



Příklad CSG stromu.

5. K čemu slouží tzv. oktalový strom?

CSG model (prostor) můžeme rozdělit právě pomocí oktalového stromu (*octtree*). Slouží to k urychlení regenerace stromu při lokálních úpravách.

6. Jakým způsobem lze zobrazit CSG modely?

V CSG stromu není informace o povrchu objektu \rightarrow netriviální zobrazení objektů. Řešení zobrazení CSG modelů:

- Převod na polygonální model, rychlé 3D zobrazení, stínování
- Převod na spline hraniční model, geometrické operace, CAD/CAM

- Ray-casting, Ray-tracing
- Upravený Z-buffer (stencil buffer)

6. Co je to šablonování, k čemu slouží jaké jeho druhy znáte?

Jedná se o transformaci (posun) 2D profilu po křivce (obecně). Slouží k reprezentaci a ke generování 3D modelů. Druhy:

- Translate
 - Po přímce (přímkové plochy)
 - Po křivce
 - S měnícím se profilem (potahování)
- Rotace – využití vlastností NURBS křivek a ploch reprezentovat přesně kružnici

7. Co jsou to dekompoziční modely a jak se dají ukládat?

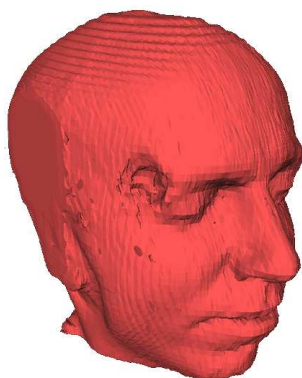
Diskrétní popis objektu dekompozicí jím obsaženého objemu na elementární objemové jednotky - krychle (hranoly), tzv. *voxely* - "3D pixel", volumetrická reprezentace (nejčastěji pravidelná kartézská mřížka, jedná se vlastně o 3D matici). Použití pro popis objemových diskretních objektů (mlha, mraky, oheň, ...), uložení naměřených 3D dat (geologie, CT/MR, ...).

Lze je ukládat následujícím způsobem:

- Oktalový strom (octree) – Rekurzivní dělení opsaného hranolu na osm částí. Vhodné pro malou hustotu dat. Problematické procházení.
- 3D pole diskretních hodnot – Prosté sekvenční pole hodnot. Velká náročnost na paměť. Vysoká rychlost přístupu.
- Subvoxely – Kombinace předchozích dvou možností. Různé možnosti kombinace a míry použití octree a 3D pole. Vhodné pro paralelizaci.

8. K čemu slouží algoritmus Marching cubes?

Slouží k převedení dekompozičních modelů na polygonální model.



Ukázka modelu po převedení na polygonální model pomocí algoritmu Marching cubes.

9. Popište hraniční reprezentaci 3D modelů.

B-rep (boundary representation) - objekt je popsán prostřednictvím 3D popisu svého povrchu - hranice. Informace o vnitřku tělesa nejsou uloženy - lze odvodit z povrchu. Objekty jsou definovány pomocí: vrcholů (body), hran (úsečky, křivky) a stěn (polygony, spline).

10. Jak je definován drátový model?

Objekty jsou definovány pomocí vrcholů (body) a hran (úsečky, křivky). Obsaženo málo topologických informací a model je nejednoznačný. Vhodné pro rychlé orientační zobrazení objektů.

11. Jak je definován polygonální model?

Objekty jsou definovány pomocí vrcholů (body), někdy hran (úsečky) a stěn (polygony). Obsažena úplná informace pro popis objektu. Dále nutnost dodržet a hlídat regulérnost - uzavřenost. Mezi výhody patří rychlost zobrazení, možnost stínování a hardwarová podpora zobrazení. Vhodné pro interaktivní zobrazení 3D objektů.

12. Co je to tzv. okřídelná hrana a k čemu slouží?

Jedná se o datovou strukturu pro uložení popisu polygonálního modelu. Obsahuje tři lineární seznamy - vrcholy, hrany a stěny. Definuje sousední vrcholy, hrany ohraničující stěnu a sousední stěny.

13. Jak je definován hraniční spline model?

Objekty jsou definovány pomocí vrcholů (body), někdy hran (úsečky, křivky) a stěn (spline plochy). Opět úplná informace pro popis objektu. Přesnost modelu je dána přesností aproximace spline ploch (NURBS). Nutnost dodržet a hlídat regulérnost - uzavřenost. Pro zobrazení se většinou převádí na polygonální model. Vhodné pro přesné geometrické modelování (CAD/CAM).

14. Co se rozumí pod zkratkou LOD?

Level of detail - vzdálené objekty není potřeba definovat (zobrazovat) v plné kvalitě detailu. Vytváříme několik úrovní detailu reprezentace objektů (LOD). Přepínání úrovně detailu podle vzdálenosti objektu. Dochází k redukci nepodstatných detailů (podle účelu, aplikace), např. redukce počtu prvků definujících hranici objektu.

15. Co jsou to 3D plochy?

Rozšíření teorie parametrických polynomiálních křivek v počítačové grafice. Plocha je definována bázovým polynomem a sítí řídicích bodů (matice). Pro zobrazení se převádí na polygony nebo používá ray-casting.

Typy ploch:

- Interpolační
 - Bikubické plochy
- Aproximační
 - Beziérové plochy
 - B-spline plochy
 - NURBS plochy

16. Co jsou to implicitní plochy (modely)?

Modelování pomocí kostry, kolem které je potenciální pole (pro každý prvek kostry). Není vyjádřen povrch, ale intenzita pole. Povrch je v místě, kde je hodnota pole = 0. Zobrazení převodem na B-rep (polygony, Marching cubes) nebo použit ray-casting (ray-tracing).

10 Řešení viditelnosti 3D objektů

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám. Odpovědi na níže uvedené otázky neuvádím (vše je pěkně popsáno v materiálech, viz. reference).

1. Jaké rozdělení algoritmů můžeme uvažovat z hlediska vstupní reprezentace objektů včetně příkladů algoritmů?
2. Znáte ještě nějaké jiné dělení (z hlediska hledání řešení v různých prostorech)?
3. Popište princip Robertsova algoritmu.
4. Popište princip algoritmu Plovoucího horizontu.
5. Popište princip Malířova algoritmu.
6. Popište princip algoritmu Dělení obrazu.
7. Popište princip Z-bufferu (Depth-bufferu).
8. Popište princip Ray-castingu.

11 Osvětlení a stínování 3D objektů

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám. Odpovědi na níže uvedené otázky opět neuvádím (řešení nechávám na čtenáři, kdyby měl někdo zájem, může vypracovat a poslat mě odpovědi, které zde uvedu).

1. Jaké znáte osvětlovací modely?
2. Jak se liší difúzní a zrcadlový odraz?
3. Popište Lambertův osvětlovací model.
4. Popište Phongův osvětlovací model. Čím se liší od Lambertova modelu?
5. Popište BRDF (dvousměrová odrazová distribuční funkce). Jak se liší od předchozích modelů?

Pozn. *Helmholtzův princip reciprocity* říká, že hodnota BRDF v daném bodě zůstává stejná, i když zaměníme směr dopadu a odrazu světelného paprsku.

6. Co je to stínování (shading) a k čemu jej potřebujeme?

Pozn. Neplést si s určováním vržených stínů (shadows).

7. Popište Konstantní stínování.
8. Popište Gourandovo stínování. Jak se liší od předchozí metody?

Pozn. Počítá se pouze ambientní a difúzní složka.

Pozn. Metoda založena na interpolaci barev z vrcholů zpracovávané plochy.

9. Popište Phongovo stínování. Jak se liší od předchozích metod?

Pozn. Neplést si s Phongovým osvětlovacím modelem.

Pozn. Metoda založena na interpolaci normálových vektorů zpracovávané plochy.

12 Texturování 3D objektů

1. Co je to textura, texturování, texel?

Textura je popis detailní struktury povrchu objektu, nezávislý na jeho geometrii. Jeden vzorek textury nazýváme texel. Proces nanášení textur na geometricky definovaný povrch objektu nazýváme texturování. Z hlediska vizuálního vnímání textura pomáhá vzájemně odlišovat a rozpoznávat jednotlivé objekty. Z hlediska zobrazování počítačových 3D modelů objektů umožňují textury dosáhnout řádově vyššího stupně realističnosti výsledného obrazu.

2. Jakým způsobem může být textura definována a v jaké dimenzi?

Definována:

- Datově – Diskrétně navzorkované a uložené v paměťovém poli, tabulce. Vhodné pro rychlé vykreslování, dochází však k aliasu.
- Procedurálně – Matematicky definované výpočtem podle daných parametrů. Vhodné pro kvalitní spojitý výsledek bez aliasu, ale pomalejší.

Dimenze:

- 2D – Popisují pouze povrch objektu.
- 3D – Popisují prostorové rozložení vlastností v celém objemu objektu.
- 4D – Jsou to 3D textury s časovou změnou, umožňují dynamické změny.

3. Jaké vlastnosti mohou být texturou mapovány na povrch objektu?

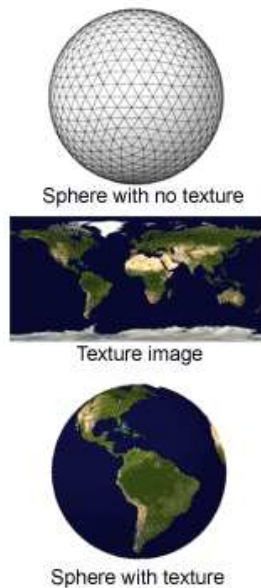
Barva povrchu, světelné vlastnosti povrchu (difuze a odrazivost), průhlednost, modifikace normály (*bump texture*), změna geometrie (*displacement texture*), hypertextura (oheň, vlasy, tráva), osvětlení (*light texture*), zrcadlení (*environment texture*).

4. Jakým způsobem provádíme interpolaci hodnot textury během rasterizace?

- Bez interpolace – zaokrouhlíme souřadnice v prostoru textury na nejbližší celočíselné hodnoty. Jde tedy o výběr nejbližší hodnoty. Je to nejrychlejší a nejjednodušší postup.
- Bilineární interpolace – provádíme lineární interpolaci z nejbližších čtyřech hodnot. Jde o základní interpolaci, požadují se minimální výpočty a zachovává se spojitost obrazových dat.
- Polynomická interpolace – provádí se interpolace z širšího okolí, polynomem vyššího stupně (nejčastěji kubickým). Je již náročnější na výpočty, ale velmi dobře vyhlazuje obrazová data.

5. Co je to mapování textur na povrch objektu?

Nanášení textury na povrch objektu (analogie tapetování).



6. Jaké znáte způsoby mapování textur na povrch objektu?

Pomocí inverzní mapovací funkce, prostřednictvím texturovacích souřadnic, mapování textur promítáním a mapování 3D textur.

7. Co je podstatou mapování textur prostřednictvím inverzní mapovací funkce a kdy je vhodné ji použít?

Je-li povrch objektu popsán jedinou analytickou plochou, pak můžeme jako inverzní mapovací funkci $M(x, y, z)$ použít inverzní funkci k parametrizaci plochy, pokud taková existuje. Je vhodné ji použít např. pro válec a kouli (využití cylindrických souřadnic).

8. Jaký je princip mapování textur prostřednictvím texturovacích souřadnic a kdy je vhodné je použít?

Texturovací souřadnice pro každý povrchový vrchol (např. uzly polygonální sítě) objektu definují odpovídající (u, v) souřadnice textury. Hodnoty texturovacích souřadnic mezi definovanými hodnotami ve vrcholech jsou lineárně interpolovány (přes jednotlivé polygony). Tak dokážeme snadno a jednoznačně mapovat texturu na povrch libovolného objektu i v případech, kdy není možné získat inverzní mapovací funkci.

Využití je především pro geometricky a topologicky jednodušší objekty, kdy je možné provádět kontrolu mapování ručně. V případě složitějších objektů toto již není možné a je vhodnější použít nějakou automatickou metodu, např. promítání textur.

9. Jak probíhá mapování textur promítáním a kdy je vhodné ho použít?

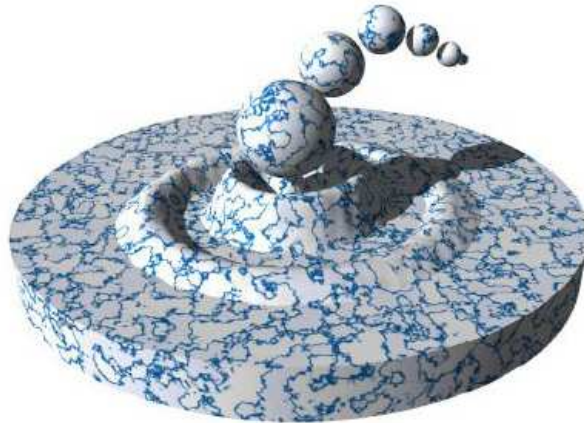
Místo nanášení textury přímo na požadovaný objekt nanese texturu s využitím inverzní mapovací funkce na pomocný objekt (rovinu, kouli, hranol, válec, atd.), který je topologicky jednoduchý, lze pro něj nalézt inverzní mapovací funkci a svým charakterem blízký texturovanému objektu. Pomocný objekt je opsán texturovanému objektu. Mezi povrchem texturovaného a pomocného objektu je potřeba nalézt vhodné zobrazení. Jeho prostřednictvím pak přeneseme texturovací souřadnice nanášené textury přes pomocný objekt na povrchové vrcholy texturovaného objektu.

Použití v případě geometricky složitějších objektů, kdy lze dosáhnout poměrně kvalitních výsledků i s navazováním mapovaných textur.

10. V čem spočítá mapování 3D textur a kdy je vhodné ho použít?

Při mapování 3D textur nepotřebujeme složitě hledat inverzní mapovací funkci nebo generovat texturovací souřadnice. Textura je definována jako jednotková krychle v prostoru u, v, w . Pro získání hodnoty textury v bodě $[x, y, z]$ provádíme pouze transformaci (posun, rotace, měřítko) prostoru objektu na prostor textury.

Aplikací 3D textur odstraníme problém kvality mapování a navazování 2D textur. Objekty s 3D texturou mohou mít bez problému velmi složitou prostorovou strukturu, jako je dřevo, mramor, atd. Využití při texturování volumetrických dat. Mezi nevýhody patří paměťová náročnost.



11. Který způsob mapování textur je nejvhodnější a proč?

Záleží na konkrétní situaci (topologie objektu, dimenze) a také na tom, zda požadujeme rychlost nebo kvalitu.

12. Jakým způsobem je dosaženo efektu hrbolatého povrchu u Bump mappingu?

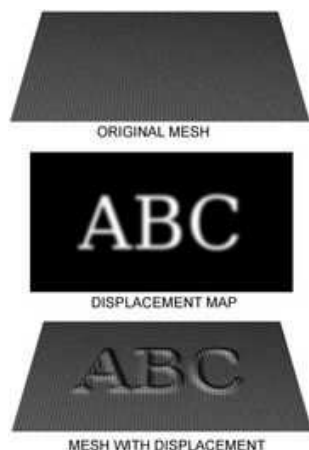
Mapování textury je prováděno některým ze standardních způsobů. Na povrch objektu jsou jako vlastnosti nanášeny požadované optické změny povrchu (gradienty). Efektu hrbolatého povrchu je dosažováno modifikací normál povrchu podle hodnot v textuře. Bump mapping je vhodný pro simulaci drobných nerovností na povrchu objektu, kdy je výsledek dostatečně realistický.



13. Jakým způsobem je dosaženo efektu hrbolatého povrchu u Displacement mappingu?

Podobné jako Bump mapping, ale nejedná se o čistě optický efekt. Dochází zde ke skutečnému posunu vrcholů objektu podle hodnot v textuře, ve směru normály povrchu. Obrisy objektu jsou

tedy odpovídajícím způsobem změněny. Mapování textury je prováděno některým ze standardních způsobů.



14. Jaký je rozdíl mezi Bump a Displacement mappingem?

U Bump mappingu nedochází k modifikaci obrysu objektu, pouze ke změně normál, kdežto u Displacement mappingu se povrch (obrysy) mění.

15. Kdy je vhodnější použití Bump mappingu a kdy Displacement mappingu?

Displacement mapping je vhodný pro simulaci větších změn (oproti Bump mappingu) ve struktuře povrchu objektu. Bump mapping je vhodnější pro jemnější doladění povrchu.

16. V čem je výhoda použití Bump mappingu a Displacement mappingu?

Vlastní geometrie objektu zabezpečuje jeho základní tvar a rozměry. Displacement mapping pak zajišťuje detaily povrchu. Bump mapping se spolu s barevnou texturou nakonec postará o jemnou plastickou a barevnou strukturu povrchu.

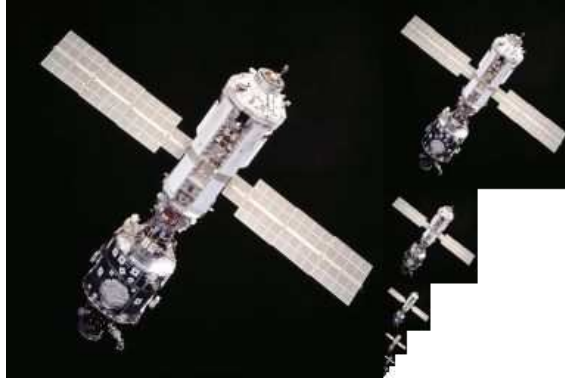
17. Jakého efektu dosáhneme aplikací Environmental mappingu?

Environmental mapping je způsob aplikace textury na objekt, který na lesklém povrchu vytváří efekt odrazu (odlesku/zrcadlení) svého okolí. Pokud zobrazovací metoda implicitně neřeší zrcadlení okolí na povrchu objektu, lze tímto způsobem příslušný efekt simulovat. Základem je textura, která představuje obraz okolí pro vybraný objekt. Tato textura je standardním způsobem s využitím inverzní mapovací funkce nanášena na obalové těleso (koule nebo krychli). Z obalového tělesa je textura promítáním odraženého paprsku mapována na povrch zrcadlicího objektu.



18. Co znamená MIP mapping a jakým způsobem se aplikuje při texturování?

MIP (Multum in Parvo) - mnoha v malém. Jde o způsob uložení jedné textury (jejího obrazu) v řadě velikostí do jedné datové matice. Textura je při tom rozložena na tři matice pro R, G a B barevný kanál. Každý kanál je potom samostatně uložen do 1/4 datové matice dvojnásobné velikosti. Do zbylé nevyužité 1/4 je rekurzivně uložena stejná textura poloviční velikosti.



19. Co je přínosem techniky MIP mapping při texturování objektů?

Použití MIP textur má dvojitý význam:

- Zrychlení zobrazení vzdálenějších objektů použitím menších verzí textur.
- Odstranění aliasu textur při zobrazení texturovaných objektů v perspektivní projekci. Předpokladem je vhodná filtrace při získávání menších rozlišeních obrazu textury.

20. Co je příčinou zkreslení textur při zobrazení texturovaných objektů v perspektivní projekci?

Perspektivní projekce není afinní transformace a proto nejsou zachovány relativní vzdálenosti. Projevuje se zkreslením mapovaných textur na jejich povrchu.

21. Jakým způsobem můžeme provést korekci zkreslení textur při zobrazení texturovaných objektů v perspektivní projekci?

Toto zkreslení je možné korigovat následujícími způsoby:

- Rozdělením každého texturovaného polygonu (taselace) na skupinu malých trojúhelníků a lineární interpolací texturovacích souřadnic do jejich vrcholů. Na malých trojúhelnících je chyba vznikající vlivem perspektivní projekce zanedbatelná.
- Hyperbolickou korekcí texturovacích souřadnic při jejich lineární interpolaci přes jednotlivé texturované polygony (během rasterizace polygonů). Tak posuneme texturovací souřadnice na správné místo i vzhledem k perspektivní projekci. Texturovací souřadnice jsou tak nejen lineárně interpolovány, ale ještě podrobeny perspektivnímu dělení vzdáleností.

13 Realistické zobrazení 3D objektů

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám. Odpovědi na níže uvedené otázky opět neuvádím (řešení nechávám na čtenáři, kdyby měl někdo zájem, může vypracovat a poslat mě odpovědi, které zde uvedu).

1. Co je to realistická vizualizace a na čem závisí?
2. Je realistická vizualizace možná? Jaké jsou důvody?
3. Popište objektové vizualizační metody.
4. Popište obrazové vizualizační metody.
5. Popište komplexní vizualizační metody.
6. Srovnejte výše uvedené metody. Pro co se hodí a pro co se nehodí?
7. Napište, jaké znáte reprezentace 3D scény a k nim možné způsoby vizualizace.
8. Popište obrazovou metodu Ray-tracing.
8. Jaké jsou výhody a nevýhody této metody (Ray-tracing)?.
8. Jakým způsobem lze tuto metodu (Ray-tracing) optimalizovat/urychlit?.
8. Popište obrazovou metodu Ray-casting.
9. Popište obrazovou metodu Particle-tracing.

Princip Particle-tracingu je takový, že se sledují cesty světelných částic (light patricles), které vycházejí ze světelných zdrojů. Tato metoda slibuje vysoký stupeň kvality výsledných obrázků s možnostmi jako jsou reflexe, refrakce a další světelné efekty jako například odraz světla v zrcadlech.

Výstupem této metody má být model s fotorealistickým osvětlením, vhodný hlavně pro obrovské scény v architektuře, interiérových návrzích a plant designu. Particle-tracing je jednoduše použitelná metoda s poměrně malými nároky na paměť a vysokou kvalitou výsledného zobrazení renderované scény. Jde o alternativu k tradiční radiozité, která také disponuje fotorealistickou kvalitou, ale má proti Particle-tracing mnohem vyšší nároky na paměť. Proto je tato metoda vhodná pro velké scény.

10. Popište princip metody zvané Radiozita.

Pozn. Pouze difúzní odraz světla.

11. Co jsou to tzv. konfigurační faktory a k čemu se využívají?
12. Co je to tzv. Nusseltova analogie a k čemu slouží?
13. Co je to tzv. postupující (progressive) radiozita?
14. K čemu se využívá tzv. adaptivní dělení ploch?
15. Srovnejte metody Ray-tracing a Radiozita z hlediska jejich výhod/nevýhod.
16. Na jakém principu jsou založeny tzv. metody Monte Carlo?

Jejich podstatou je aproximace řešení nějakého problému stochastickým vzorkováním. Hledá se náhodná proměnná, jejíž střední hodnota je řešením problému. Metody Monte Carlo se používají pro řešení široké škály (zejména fyzikálních) problémů. Proč to sem píšu - Dr. Kršek o tom mluvil na jedné přednášce, ale na stránkách k tomu nejsou žádné materiály, tak to sem dávám jen jako zmínku.

14 Antialiasing, obraz jako vícerozměrný signál

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám. Odpovědi na níže uvedené otázky opět neuvádím (řešení nechávám na čtenáři, kdyby měl někdo zájem, může vypracovat a poslat mě odpovědi, které zde uvedu).

1. Co je to alias? Kdy vzniká? Je to pro nás výhodná “vlastnost”, nebo není?
2. Napište a vysvětlete vzorkovací teorém.
3. Jaké jsou možnosti řešení problému aliasu?
4. Popište metodu supersamplingu. Jaké jsou výhody a nevýhody?
5. Popište metodu multisamplingu. Jaké jsou výhody a nevýhody? Jak se liší od supersamplingu?
6. Jaké jsou nevýhody antialiasingu?

15 Grafické formáty

Nejsou otázky - ani jsme tuto problematiku neprobírali, takže na zkoušce by se to nemělo objevit. Pro zájemce viz. [5] (strany 67 - 76) a [4], kde je spousta dalších informací.

16 Základy grafických 3D API

Následující otázky nejsou z opory (k tomuto tématu tam žádné nejsou), takže jsem je vymýšlel sám. Budu se zabývat převážně OpenGL. Odpovědi jsou pouze stručné, zbytek je si třeba přečíst v literatuře nebo přednáškách.

1. Co je to OpenGL a k čemu slouží?

Programové rozhraní pro grafické akcelerátory, API pro vykreslování, GL znamená Graphics Language/Library. Knihovna nezávislá na platformě ani hardwarovém vybavení.

2. Jaké zná OpenGL grafická primitiva? Jakto, že nádstavby nad OpenGL obsahují i kružnice atd.?

Pouze bod, úsečku a trojúhelník. Ostatní objekty se na tyto primitiva rozloží (např. obdelník se rozloží na dva trojúhelníky atd.).

3. OpenGL pracuje jako stavový stroj - co to znamená?

Pokud se něco nastaví, např. barva, tak poté se až do další změny vše vykresluje touto barvou.

4. Co je to tzv. "frame-buffer" a které buffery se používají v OpenGL?

Frame-buffer nebo taky "paměť snímku" - část paměti, do které OpenGL vykresluje výsledný obrázek. Používají se např. color, depth, stencil buffery.

5. Jaký mají význam předpony a přípony u názvu funkcí?

Mají prefix gl, glu nebo glut (podle knihovny) a suffix podle typu předávaných parametrů.

6. Proč se v OpenGL používají 2 color buffery a jak (kdy, proč) se mezi nimi přepíná?

Do jednoho se vykresluje a druhý je zobrazený. Poté se prohodí a situace se opakuje.

7. Co lze u každého vrcholu objektu zadávat?

Jeho barvu pomocí glColor(), normálu pomocí glNormal(), texturovací souřadnice pomocí glTexCoord(), materiál pomocí glMaterial() a další.

8. Jaké stínování se v OpenGL používá a co je nutné k jeho správnému fungování udělat?

Nejčastěji se používá Gouraudovo stínování, dále je zde možnost použít tzv. "Flat shading". Ke správnému fungování musíme zadat normály ve vrcholech grafických objektů (Gouraudova stínování), případně polygonu (Flat shading).

9. Popište nastavení kamery při vytváření snímku v OpenGL.

Umístění stavivu a zamíření kamerou do scény (viewing transformation). Postavení (naaranžování) scény - rozmístění a natočení objektů (modeling transformation). Výběr objektivu a zaostření (projection transformation). Určení výřezu výsledného 2D snímku (viewport transformation).

10. Jak se v OpenGL transformují objekty a k čemu slouží zásobníky matic?

Objekty se transformují pomocí transformačních matic. Modelovací a projekční matice jsou uvnitř knihovny OpenGL uloženy na zásobníku. Pro transformaci je použita matice z vrcholu zásobníku. Zásobník matic oceníme zejména při konstrukci hierarchických animovaných modelů.

11. Co je to barevný režim RGBA a k čemu slouží?

RGB složky existují tak, jak je známe, jen je přidán alpha kanál, který slouží pro dosažení průhlednosti objektů.

12. Co je to tzv. “blending” a k čemu se používá? Co je nutné dodržet při vykreslování objektů?

Využívá výše zmíněný alpha kanál k praktickému dosažení průhlednosti objektů. Při blendingu záleží na pořadí vykreslování objektů. Jako první (po všech neprůhledných objektech) by se měly vykreslit průhledné objekty nejhlouběji ve scéně (nutno ručně seřadit).

13. Co je to tzv. “color-buffer” a k čemu se využívá?

V color-bufferu (nebo bufferech) je uložena barevná informace o vykreslované scéně.

14. Co je to tzv. “depth-buffer” a k čemu se využívá?

Depth buffer, tedy paměť hloubky, je někdy také nazýván Z-buffer. Jedná se o paměť, v níž jsou většinou uloženy informace zajišťující vykreslení pouze viditelných částí těles, tj. vzdálenější plošky jsou překryty ploškami bližšími.

15. Co je to tzv. “stencil-buffer” a k čemu se využívá?

Stencil buffer neboli paměť šablony je používán pro určení, do kterých míst na obrazovce je povoleno vykreslování. Využití např. pro vykreslování zrcadlících ploch.

15. Co OpenGL samo o sobě neumí a je nutno “obcházet” programově?

Řekl bych, že vržené stíny a zrcadlení.

Reference

- [1] Kršek P.: *Studijní opora k předmětu "Základy počítačové grafiky"*,
<https://www.fit.vutbr.cz/study/courses/IZG/private/>
- [2] Kršek P., Španěl M.: *Materiály k přednáškám "Základy počítačové grafiky"*,
<https://www.fit.vutbr.cz/study/courses/IZG/private/>
- [3] Seeman M., Šiler O., Venera J., Štancl V., Španěl M.: *Podklady ke cvičením z IZG*,
<https://www.fit.vutbr.cz/study/courses/IZG/private/>
- [4] Wikipedia: internetová encyklopedie, <http://wikipedia.org/>
- [5] Žára J., Beneš B., Sochor J., Felkel P.: *Moderní počítačová grafika*, ComputerPress, 2004