

Question # 1 Type: Single answer, Maximum score: 3, Score:0

Co platí o Coonsově kubice definované řídícími body $P_0P_1P_2P_3$?

- ☒ A Je-li $P_0 = P_1$, leží počátek křivky na úsečce P_0P_2 a křivka končí v těžišti trojúhelníka $P_1P_2P_3$
- ☐ B Křivka má v počátku tečnu rovnoběžnou s úsečkou P_0P_2 a při napojování oblouků lze zajistit maximálně spojitost C^1 .
- ☐ C Je-li $P_0 = P_1 = P_2$ pak Coonsova kubika degeneruje na úsečku ležící na spojnici bodů P_0 a P_3 **Correct answer**
- ☐ D Křivka začíná v antitěžišti trojúhelníka $P_0P_1P_2$ a končí v těžišti trojúhelníku $P_1P_2P_3$.

Question # 2 Type: Single answer, Maximum score: 2, Score:2

Hloubka transformovaného vrcholu na výstupu vertex shaderu (VS) se typicky

- ☒ A v těle vertex shaderu zapisuje jako třetí složka zabudované souřadnice vrcholu `gl_Position` **Correct answer**
- ☐ B v nezměněné podobě předává na vstup fragment shaderu
- ☐ C zapisuje na výstup VS po vydělení složkou w
- ☐ D na konci VS zapisuje do zabudované výstupní proměnné `gl_VertexDepth` přiřazovacím příkazem

Question # 3 Type: Single answer, Maximum score: 2, Score:2

Na definování čtyř trojúhelníků pomocí primitiv typu `GL_TRIANGLE_FAN` potřebujeme ... vrcholů(y)

- ☐ A 12
- ☐ B 8
- ☒ C 6 **Correct answer**
- ☐ D 4

Question # 4 Type: Single answer, Maximum score: 2, Score:2

Při míchání barev v OpenGL

- ☐ A nelze vstupy nijak ovlivnit; jediné, co můžeme měnit je tzv. míchací rovnice (blending equation)
- ☐ B se nastavují míchací činitele pro každý fragment samostatným příkazem `glBlendFunc()`
- ☐ C se používá míchací paměť (blending buffer)
- ☒ D se jedním příkazem nastaví míchací činitele, určující, která část vstupů se zúčastní míchání **Correct answer**

Question # 9 Type: Single answer, Maximum score: 3, Score:3

Které tvrzení je pravdivé?

- ☐ A Textura v OpenGL je vždy 2D obrázek.
- ☒ B Použitím vhodné projekční matice jsme schopni implementovat rovnoběžné promítání. **Correct answer**
- ☐ C OpenGL zobrazuje pouze objekty nacházející se v normalizovaných souřadnicích zařízení (NDC), tj. se souřadnicemi v intervalu $(0, 0, 0)$ až $(1, 1, 1)$.
- ☐ D OpenGL nepodporuje lineární interpolaci hodnot mezi VS a FS.

Question # 10 Type: Single answer, Maximum score: 3, Score:3

Mějme dvojici vektorů $u = (1, 2, 3)$ a $v = (4, 3, 2)$. Hodnota vektorového součinu $u \times v$ těchto vektorů je rovna

- ☒ A $(-5, 10, -5)$ **Correct answer**
- ☐ B $(4, 6, 6)$
- ☐ C 16
- ☐ D 15

Question # 5 Type: Single answer, Maximum score: 2, Score:2

Příkaz `glutPostRedisplay()`

- ☐ A Je synonymem pro proceduru, kterou jsme zaregistrovali pro vykreslování obrazovky
- ☐ B způsobí okamžité vykreslení obrazovky
- ☐ C zastaví provádění programu a čeká, než se dokončí všechny grafické příkazy
- ☒ D zařadí událost překreslení okna do fronty událostí **Correct answer**

Question # 6 Type: Single answer, Maximum score: 2, Score:2

Normály se musí **ve fragment shaderu** před výpočtem osvětlení normalizovat,

- ☐ A normály se normalizovat nemusí.
- ☐ B protože se interpolací prodlouží.
- ☐ C protože se při interpolaci obrací jejich orientace.
- ☒ D protože se interpolací typicky zkrátí. **Correct answer**

Question # 7 Type: Single answer, Maximum score: 3, Score:3

Mipmapping textur

- ☐ A se typicky používá, pokud je rozlišení textury pro daný fragment příliš nízké.
- ☒ B se typicky používá, pokud je rozlišení textury pro daný fragment příliš vysoké. **Correct answer**
- ☐ C se prakticky nepoužívá kvůli několikanásobné paměťové náročnosti.
- ☐ D nemá smysl pro 1D textury.

Question # 8 Type: Single answer, Maximum score: 3, Score:3

Pro definici Coonsovy křivky stupně 3 se sedmi oblouky je potřeba

- ☒ A 10 řídících bodů. **Correct answer**
- ☐ B 12 řídících bodů.
- ☐ C 16 řídících bodů.
- ☐ D 44 řídících bodů.

Question # 1 Type: Single answer, Maximum score: 2, Score:2

O trilineární interpolaci textur v OpenGL platí:

- ☐ A Řeší antialiasing na hranách objektů.
- ☐ B Neřeší problém se skokovým přechodem mezi různými úrovněmi mipmap.
- ☒ C Aktivuje se použitím módu `GL_LINEAR_MIPMAP_LINEAR`. **Correct answer**
- ☐ D Odebere 3-krát více vzorků než lineární.

Question # 2 Type: Single answer, Maximum score: 3, Score:3

Struktura dat uložených v bufferu vrcholů (vertex buffer) se definuje:

- ☐ A Připojením bufferu příkazem `glBindBuffer(...)`.
- ☐ B Příkazem `glBufferData(...)`.
- ☒ C Příkazem `glVertexAttribPointer(...)`. **Correct answer**
- ☐ D Příkazem `glGenVertexArrays(...)`.

Question # 3 Type: Single answer, Maximum score: 3, Score:3

M je matice 4x4 transformující vrcholy do prostoru kamery (tzv. matice modelView) a N je normálová matice (matice pro transformaci normál do prostoru kamery).

- ☒ A Transponovaná matice N je rovna inverzní matici k matici M , tj. $N^T = M^{-1}$. **Correct answer**
- ☐ B Matice N je rovna inverzní matici k matici M , tj. $N = M^{-1}$.
- ☐ C Transponovaná matice N je vždy rovna transponované matici M , tj. $N^T = M^T$.
- ☐ D Matice N je vždy shodná s transponovanou maticí M , tj. $N = M^T$.

Question # 4 Type: Single answer, Maximum score: 3, Score:3

Aliasing je způsoben

- ☒ A vykreslováním vyšších frekvencí, než které můžeme reprezentovat pomocí vzorků o velikosti pixelu **Correct answer**
- ☐ B vykreslováním nižších frekvencí, než které můžeme reprezentovat pomocí vzorků o velikosti pixelu
- ☐ C vykreslováním menšího počtu primitiv, než je limit grafické karty
- ☐ D vykreslováním většího počtu primitiv, než je limit grafické karty

Question # 5 Type: Single answer, Maximum score: 3, Score:3

Phongův osvětlovací model

- ☒ A aproximuje některé jevy globálního osvětlení **Correct answer**
- ☐ B nevyžaduje ke svému vyhodnocení znalost směru od osvětlovaného bodu k pozorovateli-kameře
- ☐ C nelze aplikovat na osvětlení čistě šedotónových materiálů, funguje pouze v RGB
- ☐ D je jediný osvětlovací model aplikovatelný v OpenGL shaderech

Question # 6 Type: Single answer, Maximum score: 3, Score:3

Které tvrzení je pravdivé?

- ☐ A Textura v OpenGL je vždy 2D obrázek.
- ☒ B Použitím vhodné projekční matice jsme schopni implementovat rovnoběžné promítání. **Correct answer**
- ☐ C OpenGL zobrazuje pouze objekty nacházející se v normalizovaných souřadnicích zařízení (NDC), tj. se souřadnicemi v intervalu (0, 0, 0) až (1, 1, 1).
- ☐ D OpenGL nepodporuje lineární interpolaci hodnot mezi VS a FS.

Question # 7 Type: Single answer, Maximum score: 2, Score:2

Dělení hodnotou w se v OpenGL provádí

- ☐ A ve fragment shaderu, jako poslední operace s fragmentem před vykreslením.
- ☒ B ve fixní části mezi vertex shaderem a fragment shaderem. **Correct answer**
- ☐ C ve fixní části mezi hlavním procesorem a vertex shaderem.
- ☐ D ve vertex shaderu nastavením projekční matice.

Question # 8 Type: Single answer, Maximum score: 2, Score:0

Která forma popisu uzavřeného tělesa popsaného jedinou rovnicí bude nejvhodnější pro testování, zda nějaký daný bod leží uvnitř tělesa?

- ☐ A Parametrický zápis.
- ☒ B Explicitní zápis.
- ☐ C Implicitní zápis. **Correct answer**
- ☐ D Ani jeden z uvedených.

Question # 9 Type: Single answer, Maximum score: 2, Score:2

O globálních osvětlovacích metodách můžeme říci, že

- ☐ A patří mezi ně rasterizace
- ☐ B generují vždy obrázky bez šumu
- ☒ C simulují vzájemné osvětlení i odvrácených ploch **Correct answer**
- ☐ D jsou typicky rychlé na výpočet

Question # 10 Type: Single answer, Maximum score: 2, Score:2

Lineární transformace ve 3D lze vyjádřit následovně (zvolte nejmenší možnou velikost)

- ☐ A Maticí 3x4
- ☒ B Maticí 3x3 **Correct answer**
- ☐ C Maticí 4x3
- ☐ D Maticí 4x4

Question # 3 Type: Single answer, Maximum score: 2, Score:2

Knihovna OpenGL

A pracuje s algoritmem sledování paprsku (Ray tracing)

B je určena k fotorealistickému zobrazování

C obsahuje funkce pro nahrávání modelů ze souborů

D využívá při zobrazování rasterizaci **Correct answer**

Clear answer

Question # 4 Type: Single answer, Maximum score: 3, Score:0

Jak bude vypadat výsledná transformační matice M , chceme-li provést rotaci okolo bodu P , za předpokladu, že matice T_1 reprezentuje posunutí do počátku O , tj. posunutí o vektor $O - P$, T_2 posunutí z počátku o vektor $P - O$ a matice R vyjadřuje požadovanou rotaci? Předpokládejte, že v okamžiku provedení rotace je aktuální modelovací transformace uložena v matici C .

- ☐ A $M = T_2, R, T_1, C$ **Correct answer**
- ☐ B $M = C, T_1, R, T_2$
- ☐ C $M = C, T_2, R, T_1$
- ☒ D $M = T_1, R, T_2, C$
- ☐ Clear answer

Question # 5 Type: Single answer, Maximum score: 2, Score:2

Je-li bod ve Phongově osvětlovacím modelu osvětlen více světly, výsledné osvětlení spočítáme jako:

- A Součet příspěvků jednotlivých světél** **Correct answer**
- B Aritmetický průměr příspěvků jednotlivých světél vážený jejich intenzitou.
- C Příspěvek náhodně zvoleného světla s pravděpodobností úměrnou jeho intenzitě
- D Čistý aritmetický průměr příspěvků jednotlivých světél.

Clear answer

Question # 6 Type: Single answer, Maximum score: 3, Score:3

Test šablony (stencil test)

- A Propustí všechny fragmenty, které padnou do obdélníkového okna na obrazovce.
- B Používá přední rovinu paměti barvy (front buffer).
- C Propustí fragmenty, které vyhovují zadané porovnávací funkci** **Correct answer**
- D Propustí jen ty fragmenty, které mají barvu shodnou se zadanou referenční hodnotou.

Clear answer

Question # 8 Type: Single answer, Maximum score: 2, Score:2

O globálních osvětlovacích metodách můžeme říci, že

- A generují vždy obrázky bez šumu
- B patří mezi ně rasterizace
- C simulují vzájemné osvětlení i odvrácených ploch** **Correct answer**
- D jsou typicky rychlé na výpočet

Clear answer

Question # 9 Type: Single answer, Maximum score: 3, Score:0

Vstupní proměnné *fragment shaderu* s kvalifikátorem *in*

- A jsou načítány z vertex bufferu**
- B jsou načítány z textury
- C lze nastavit na konstantní hodnotu v hlavním programu (v OpenGL)
- D vzniknou interpolací výstupů z předchozích shaderů** **Correct answer**

Clear answer

Question # 10 Type: Single answer, Maximum score: 2, Score:2

Algoritmus de Casteljau

- A Slouží k adaptivnímu vykreslování NURBS křivek.
- B Rozdělí Bézierovu křivku na oblouk a část, kterou lze vykreslit jako úsečku.
- C Slouží k vykreslování Bézierových křivek zadaným počtem stejně dlouhých úseček.
- D Vypočítává polohu bodu na Bézierově křivce pro zadanou hodnotu parametru t** **Correct answer**

Clear answer

1. Poslední akce která se provádí ve fixním zobrazovacím řetězci OpenGL → míchání barev
2. Depth buffer slouží hlavně → k zajištění aby vzdálenější objekty nepřekryly bližší
3. Ambientní světlo → reprezentuje v konkrétním bodě scény základní kladinu osvětlení a vliv všech světél
4. Rozsah texturovacích souřadnic pro pokrytí čtvercové textury je 0-1
5. Mezi základní primitiva OGL nepatří → torus (jinak lichoběžník patří)
6. Ve vykreslovacím modu GL_SELECT → se negeneruje obraz vůbec
7. Při rovnoběžném promítání → promítání zachovává úhly, rovnoběžnost úseček, střed úsečky se promítne na střed promítnuté
8. $M = C.T1.S.T2$ → změnu měřítka okolo P – t2-posune do P, S – scale, T1 posune zpět, pozice
9. Máme dvě bezierky (prochází konc body) jdou definičovány stejnými body, pak →
 $B'_1(1) = -B'_2(0)$ a $B'_2(1) = -B'_1(0)$
10. Reprezentace scény pomocí orientované acyklického grafu značí → že se geometrie neopakuje
11. Je vhodnější vykreslovat pomocí GL_TRIANGLE_STRIP, protože zadáváme méně vertexů
12. Stínování v OGL → interpoluje barvy mezi barvami ve vrcholech
13. O překreslení obrazovky požádáme pomocí glutPostRedisplay() → zařadí vykreslení do fronty požadavků
14. Příkaz glutPostRedisplay() → zařadí událost překreslení okna do fronty událostí
15. První operace s fragmentem ve fixním zobrazovacím řetězci → Aplikace textury
16. O globálních osvětlovacích metodách můžeme říci → že simulují vzájemné osvětlení i odvrácených ploch
17. U gimbal locku dojde → když dojde ke splnutí osy díky předchozímu otočení
18. Aliasing je způsoben → vykreslováním vyšších frekvencí, než které můžeme reprezentovat pomocí vzorků o velikosti pixelu
19. Perspektivní promítání → střed úsečky se promítne na střed promítnuté úsečky
20. glutLookAt → 1x posunutí, 3x rotace
21. lineární transformace lze vyjádřit už → 3x3 maticí
22. která forma popisu uzavřeného tělesa popsaného jedinou rovnicí bude nejvhodnější pro testování, zda je bod uvnitř → implicitní zápis
23. o trilineární interpolaci textur v OGL platí → aktivuje se pomocí GL_LINEAR_MIPMAP_LINEAR
24. Struktura dat uložených v bufferu vrcholů (vertex bufferu) se definuje →
glVertexAttribPointer
25. Určete správné tvrzení → Použitím vhodné projekční matice jsme schopni implementovat rovnoběžné promítání

26. Phongův osvětlovací model → aproximuje některé jevy globálního osvětlení
27. M je matice 4x4 transformující vrcholy do prostoru kamery (ModelViewMatice) N je normálová matice (transformace normál do prostoru kamery → transponovaná N = inverzní $M N^T = M^{-1}$)
28. Na 4 trojúhelníky v triangle_fan potřebujeme → 6 vrcholů
29. Normály se musí ve fragment shaderu před výpočtem osvětlení normalizovat → interpolací se typicky zkrátí
30. Hloubka transformovaného vrcholu na výstupu VS → se typicky zapisuje v těle shaderu jako třetí složka `gl_Position`
31. Při míchání barev v OpenGL → se jedním příkazem nastaví míchací činitele, určující, která část vstupu se zúčastní míchání
32. Mipmapping textur → se typicky používá, když je rozlišení textury pro daný fragment příliš velké
33. Pro definici Coonsovy křivky stupně 3 se sedmi oblouky potřebujeme → 10 bodů
34. Co platí o Coonsově kubice definované řídícími body P_0, P_1, P_2, P_3 ? → Je-li $P_0 = P_1 = P_2$, pak Coonsova kubika degeneruje na úsečku ležící na spojnici bodů P_0 a P_3
35. Mějme dvojici vektorů $u = (1, 2, 3)$ a $v = (4, 3, 2)$. Hodnota vektorového součinu $u \times v$ těchto vektorů je rovna → $(-5, 10, -5)$

Question # 1 Type: Single answer, Maximum score: 3, Score:3

Struktura dat uložených v bufferu vrcholů (vertex buffer, `GL_ARRAY_BUFFER`) se definuje:

- ☐ A Příkazem `glGenVertexArrays(...)`.
- ☐ B Připojením bufferu příkazem `glBindBuffer(...)`.
- ☐ C Příkazem `glBufferData(...)`.
- ☒ D Příkazem `glVertexAttribPointer(...)`. Correct answer
- ☐ Clear answer

Question # 2 Type: Single answer, Maximum score: 2, Score:2

Metoda Monte Carlo sledování světla (light tracing)

- ☒ A Patří mezi metody, které sledují cestu paprsků vysílaných ze světelných zdrojů. Correct answer
- ☐ B Funkci BRDF vůbec nepoužívá.
- ☐ C Generuje jen velmi málo paprsků (jeden na pixel).
- ☐ D Využívá matici konfiguračních faktorů mezi ploškami.
- ☐ Clear answer

Question # 4 Type: Single answer, Maximum score: 2, Score:0

Knihovna OpenGL

- ☐ A je určena k fotorealistickému zobrazování
- ☐ B pracuje s algoritmem sledování paprsku (Ray tracing)
- ☒ C postupuje při zobrazování po jednotlivých pixelech obrazovky (image order)
- ☐ D postupuje při zobrazování po zobrazovaných tělesech (object order) Correct answer
- ☐ Clear answer

Question # 5 Type: Single answer, Maximum score: 3, Score:3

Jsou dány dvě Bézierovy kubické křivky $Q_1(t)$ a $Q_2(t)$. První křivka $Q_1(t)$ je definována řídicími body P_1, P_2, P_3 a P_4 . Druhá křivka $Q_2(t)$ je definována stejnými body, ale v opačném pořadí P_4, P_3, P_2 a P_1 . Které z následujících tvrzení platí?

- ☐ A obě křivky $Q_1(t)$ a $Q_2(t)$ mají odlišný tvar, ale stejné počáteční a koncové derivace, tj. $Q_1'(1) = Q_2'(0)$ a $Q_2'(1) = Q_1'(0)$
- ☒ B obě křivky $Q_1(t)$ a $Q_2(t)$ mají shodný tvar, a pro počáteční a koncové derivace platí $Q_1'(1) = -Q_2'(0)$ a $Q_2'(1) = -Q_1'(0)$ Correct answer
- ☐ C obě křivky $Q_1(t)$ a $Q_2(t)$ mají shodný tvar, a pro počáteční a koncové derivace platí $Q_1'(0) = Q_2'(0)$ a $Q_2'(1) = Q_1'(1)$
- ☐ D jedná se o naprosto odlišné křivky
- ☐ Clear answer

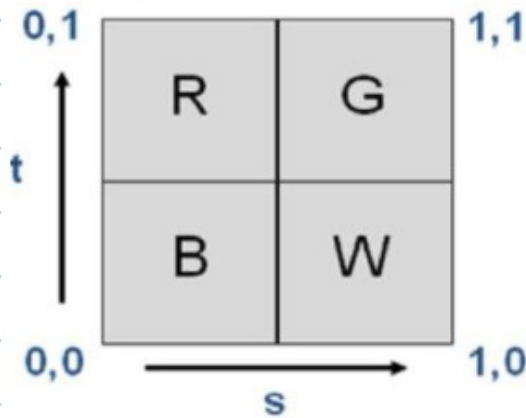
Question # 6 Type: Single answer, Maximum score: 2, Score:0

Ambientní světlo

- ☐ A se projeví pouze v místech, odkud jsou vidět světelné zdroje
- ☐ B reprezentuje v konkrétním bodě scény základní hladinu osvětlení a vliv všech světelných zdrojů Correct answer
- ☒ C je nezávislé na použitých světelných zdrojích
- ☐ D se odráží od všech materiálů stejně jako zrcadlové
- ☐ Clear answer

Question # 7 Type: Single answer, Maximum score: 3, Score:3

Jaká bude výsledná barva fragmentů po aplikaci textury (viz obrázek) za předpokladu, že budeme kreslit trojúhelník jehož vrcholy budou mít přiřazené stejné texturovací souřadnice $(1.55, -1.66)$, použijeme míchání v režimu ADD, barva fragmentů před aplikací textur bude $F = (0, 1, 1)$ a opakování textury ve směru s bude nastaveno na GL_REPEAT a pro směr t na GL_CLAMP?



- ☐ A $R = (1, 0, 0)$.
- ☒ B $W = (1, 1, 1)$. Correct answer
- ☐ C Žádná z uvedených.
- ☐ D $G = (0, 1, 0)$.
- ☐ E $B = (0, 0, 1)$.
- ☐ Clear answer

Question # 8 Type: Single answer, Maximum score: 3, Score:3

Pokrytí (coverage) fragmentu udává:

- ☐ A jak velkou část obrazovky v pixelech fragment pokrývá
- ☐ B kolik pixelů přes sebe bylo vykresleno na obrazovce/okně na pozici odpovídající danému fragmentu
- ☒ C jak velkou relativní část plochy fragmentu primitivum zakrývá Correct answer
- ☐ D jak velké mají být fragmenty generované během rasterizace
- ☐ Clear answer

Question # 9 Type: Single answer, Maximum score: 2, Score:2

Která forma popisu uzavřeného tělesa popsaného jedinou rovnicí bude nejvhodnější pro testování, zda nějaký daný bod leží uvnitř tělesa?

☒ A Implicitní zápis. Correct answer

☐ B Explicitní zápis.

☐ C Ani jeden z uvedených.

☐ D Parametrický zápis.

☐ Clear answer

Question # 10 Type: Single answer, Maximum score: 3, Score:3

Jak bude vypadat výsledná transformační matice M , chceme-li provést *rotaci* okolo bodu P , za předpokladu, že matice T_2 reprezentuje posunutí do počátku O , tj. posunutí o vektor $O - P$, T_1 posunutí z počátku o vektor $P - O$ a matice R vyjadřuje požadovanou rotaci? Předpokládejte, že v okamžiku provedení rotace je aktuální modelovací transformace uložena v matici C .

☒ A $M = T_1 R T_2 C$ Correct answer

☐ B $M = T_2 R T_1 C$

☐ C $M = C T_2 R T_1$

☐ D $M = C T_1 R T_2$

☐ Clear answer