

## 6. Tworzenie obrazu szkieletowego wymaga określenia

6	Rodzaju rzutu, parametrów rzutowania i powierzchni, na której rzut zostanie wyświetlony
	Jedynie powierzchni, na której rzut zostanie wyświetlony
	Jedynie rodzaju rzutu

## 7. Planarne rzuty geometryczne to

7	Dwupunktowe rzuty perspektywiczne i rzuty ukośne
	Widok z góry, widok z boku i widok z przodu
	Synonim rzutów równoległych

## 8. Pasy i wachlarze to

8	Figury geometryczne stosowane w grafice komputerowej
	Konstrukcje wykorzystywane przy rzutowaniu perspektywicznym
	Techniki wykorzystywania wspólnych wierzchołków prymitywów trójkątnych
	Wynik podziału ścian sześcianu na układ prymitywów

## 9. Ustawianie geometrii bryły widzenia to

9	Utworzenie „mapy” opisującej położenie wszystkich występujących na niej obiektów i ustalenie rozmiarów obiektów przy użyciu przekształceń elementarnych
	Określenie układ współrzędnych sceny i układu współrzędnych oka (rzutowania)
	Wprowadzenie obiektów na scenę graficzną

## 10. Pod pojęciem teksturowania rozumiemy

10	Pod pojęciem teksturowania rozumiemy	Dopasowywanie tekstur do wielkości odpowiadającej wielkości prymitywów trójkątnych
		Nakładanie na trójwymiarowy szkielet płaskich obrazków nazywanych teksturami
		Wyodrębnienie z kwadratowej tekstury trójkątów o żądanej wielkości

## 11. Pod pojęciem MIP mappingu rozumiemy

11	Pod pojęciem MIP mappingu rozumiemy	Powielanie pojedynczych teksceli w różnych miejscach prymitywu
		Utworzenie przed renderowaniem sceny z każdej tekstury wzorcowej kilku pomniejszonych bitmap
		Ujednolicianie barwy z wykorzystaniem teksceli pochodzących z dwóch różnych wielkości tekstur

## 12. Mapowanie wybojów polega na

12	Mapowanie wybojów polega na	Modyfikacji tekstury wzorcowej o elementy faktury powierzchni
		Drobnej zmianie kształtu prymitywu, powodującej utworzenie niewielkich wybruszeń.
		Nakładaniu na prymityw mapy obrazującej zmiany jasności poszczególnych pikseli po uprzednim nałożeniu tekstury obrazu

## 13. Mapowanie sferyczne to

13	Mapowanie sferyczne to	Nałożenie na obiekt sześciu różnych tekstur obrazu.
		Nałożenie na obiekt sześciu różnych map środowiska.
		Rodzaj mapowania z wykorzystaniem przestrzennych tekstur środowiska

## 14. Rendering można podzielić na

14	Rendering można podzielić na	wyświetlanie wielokątów i linii prostych, wypełnianie wielokątów i skalowanie okien
		MIP mapping, próbki punktowe, filtrowanie i korekcja perspektywy
		Teksturowanie, oświetlenie i cieniowanie oraz dodawanie efektów specjalnych

## 15. Pod pojęciem wokseli rozumiemy

15	Pod pojęciem wokseli rozumiemy	Piksele umieszczone na obrazie sceny trójwymiarowej	
		Trójwymiarowe odpowiedniki tekselfi umożliwiające mapowanie trójwymiarowe	
		Piksele po nałożeniu map obrazu i wypukłości	
		Tekselfe po procesie MIP mappingu	

## 16. Obraz foto-realistyczny w grafice komputerowej

16	Obraz foto-realistyczny w grafice komputerowej	To obraz powstały wskutek skanowania przestrzennego obiektów rzeczywistych	
		To zdjęcie fotograficzne umieszczone na obrazie wygenerowanym komputerowo	
		Często bywa idealizowany, jego elementy mogą być nierzeczywiste, a jego jakość bywa celowo obniżana	
		To obraz o najwyższej możliwej jakości i zgodności z rzeczywistością	

## 17. Generowanie grafiki komputerowej

17	Generowanie grafiki komputerowej	Zawsze odbywa się z udziałem jedynie procesora centralnego komputera	
		Zawsze odbywa się przy współudziale procesora centralnego komputera i układu graficznego	
		Zawsze odbywa się z udziałem jedynie układu graficznego	
		Przy udziale procesora centralnego i układu graficznego, lub z pominięciem jednej z wymienionych jednostek	

## 18. Procesory graficzne to

18	Procesory graficzne to	Urządzenia odciążające jednostkę centralną w procesie rzutowania perspektywicznego	
		Urządzenia odciążające jednostkę centralną w końcowym etapie obliczeń	
		To urządzenia odciążające jednostkę centralną w początkowym etapie obliczeń	
		Układy samodzielnie przetwarzające całość strumienia grafiki komputerowej	

## 19. Strumień grafiki komputerowej to

19	Strumień grafiki komputerowej to	Powiązane ze sobą operacje przekształceń geometrycznych sceny i operacji renderingu	
		Ciąg operacji utworzony z powiązanych ze sobą przekształceń elementarnych	
		Teksturowanie, oświetlanie i cieniowanie	
		Próbkowanie punktowe, filtrowanie i mapowanie wybojów	

## 20. Kamera syntetyczna i scena graficzna

20	Kamera syntetyczna i scena graficzna	Wymagają tego samego układu współrzędnych	
		Wymagają dwóch różnych niezależnych układów współrzędnych	
		Wymagają dwóch układów współrzędnych różniących się tylko jedną z osi	
		Kamera syntetyczna nie wymaga żadnego układu współrzędnych	

## 21. Obraz szkieletowy sceny graficznej

21	Obraz szkieletowy sceny graficznej	Jest wykreślany przed operacjami renderingu	
		Jest wykreślany po operacjach renderingu	
		Jest tworem pomocniczym i nie ma potrzeby go wykreślać	
		Zostanie wykreślony niezależnie od rodzaju rzutu	

## 22. Planarne rzuty geometryczne

22	Planarne rzuty geometryczne	Zawsze zachowują kształty i wymiary obiektów sceny graficznej i są wyświetlane na powierzchni płaskiej	
		Zachowują kształty obiektów sceny graficznej i mogą być wyświetlane na dowolnej powierzchni	
		Są synonimem rzutów prostokątnych	
		Zawsze zachowują kształty, ale nie zawsze wymiary obiektów sceny graficznej i są wyświetlane na płaskiej powierzchni	

## 23. Zapamiętanie pasów i wachlarzy o tej samej liczbie prymitywów trójkątnych wymaga

23	Zapamiętanie pasów i wachlarzy o tej samej liczbie prymitywów trójkątnych wymaga	Zdecydowanie różnego rozmiaru pamięci komputera	
		Tego samego rozmiaru pamięci komputera	
		Rozmiaru pamięci zależnego od konstrukcji obu tworów	
		Rozmiaru pamięci zależnego od kształtu obiektów jakie opisują	

## 24. Do określenia wymiarów i kątów obiektów sceny graficznej wymagane są

24	Do określenia wymiarów i kątów obiektów sceny graficznej wymagane są	Jeden, dwa lub trzy różne rzuty prostokątne	
		Perspektywiczny rzut trzypunktowy	
		Rzut izometryczny	
		Dowolny rzut ukośny	

## 25. Pod pojęciem tekscela rozumiemy

25	Pod pojęciem tekscela rozumiemy	Dyskretny punkt tekstury o dowolnej wielkości. 
		Najmniejszy, dyskretny punkt obrazu po operacji teksturowania.
		Najmniejszy, dyskretny punkt tekstury.
		Trójkątny układ punktów tekstury, z których każdy ma inną barwę składową

## 26. Mapy MIP to

26	Mapy MIP to	Tekstury o różnych rozmiarach, które są wynikiem skalowania tekstury wyjściowej. 
		Tekstury o różnych rozmiarach, które są kolejno wczytywane do pamięci komputera.
		Tekstury o identycznych rozmiarach, ale różnym stopniu rozmycia obrazu.
		Tekstury o różnych kształtach zależnych od kształtu prymitywów obrazu

## 27. Tłoczenie wybojów

27	Tłoczenie wybojów	Jest synonimem mapowania wybojów 
		Jest rodzajem mapowania wybojów 
		Nie ma nic wspólnego z mapowaniem wybojów
		Jest częścią procesu mapowania wybojów

## 28. Mapowanie sześciennne wymaga nałożenia na obiekt

28	Mapowanie sześciennne wymaga nałożenia na obiekt	Tekstury podstawowej i trzech różnych map środowiska. 
		Tekstury podstawowej i sześciu różnych map środowiska.
		Tekstury podstawowej i jednej mapy środowiska.
		Podstawowej tekstury i mapy wypukłości.

## 29. Do przedstawienia pełnej palety barw RGB i półprzezroczystości wymagany jest

29	Do przedstawienia pełnej palety barw RGB i półprzezroczystości wymagany jest	32 bitowy obszar pamięci 
		24 bitowy obszar pamięci
		Ściśle nieokreślony obszar pamięci
		Obszar zależny od rodzaju stosowanych monitorów

### 30. Pod pojęciem "alpha blending" rozumiemy

Pod pojęciem „alpha blending” rozumiemy	Technikę pozwalającą na zmianę jasności poszczególnych elementów obrazu
	Technikę uzyskiwania koloru poprzez mieszanie kolorów składowych
	Technikę określania stopnia przezroczystości tekstury z wykorzystaniem oddzielnego kanału.

### 31. Obraz foto-realistyczny w grafice komputerowej

31	Obraz foto-realistyczny w grafice komputerowej	To obraz obiektów rzeczywistych na tle obrazów generowanych komputerowo
		To zdjęcie fotograficzne umieszczone na obrazie wygenerowanym komputerowo
		To obraz generowany komputerowo, często idealizowany, zawierający obiekty rzeczywiste i/lub nierzeczywiste
		To obraz o najwyższej możliwej jakości i 100% zgodny z rzeczywistością

### 32. Złudzenia optyczne w grafice komputerowej

32	Złudzenia optyczne w grafice komputerowej	Są spowodowane błędami renderingu
		Wynikają z właściwości stosowanych urządzeń
		Wynikają ze stosowania niewłaściwych algorytmów
		Mogą być i na ogół są wykorzystywane w tworzeniu realizmu wirtualnego

### 33. Akceleratory grafiki płaskiej to urządzenia wspomagające

33	Akceleratory grafiki płaskiej to urządzenia wspomagające	Wspomagające jednostkę centralną
		Wspomagające procesory kart graficznych
		Wspomagające jednostkę centralną i procesory kart graficznych
		Zastępujące jednostkę centralną w obliczeniach związanych z grafiką 2D

### 34. Akcelerator geometrii to inna nazwa

34	Akcelerator geometrii to inna nazwa	Dowolnej karty graficznej
		Procesora graficznego
		Akceleratora grafiki płaskiej
		Akceleratora grafiki trójwymiarowej

**35. Przetwarzanie strumienia grafiki komputerowej przez poszczególne algorytmy zachodzi**

35	Przetwarzanie strumienia grafiki komputerowej przez poszczególne algorytmy zachodzi	Tylko i wyłącznie sekwencyjnie
		Tylko i wyłącznie równolegle
		Równolegle bądź sekwencyjnie - wybór ma programista
		Równolegle bądź sekwencyjnie w zależności od wymogów algorytmów



**36. Nowoczesne karty graficzne charakteryzują się znaczną liczbą procesorów**

36	Nowoczesne karty graficzne charakteryzują się znaczną liczbą procesorów	Specjalizowanych, z których każdy może wykonywać dowolne obliczenia
		Uniwersalnych, z których każdy może wykonywać dowolne obliczenia
		Specjalizowanych, z których każdy ma konkretne przeznaczenie
		Specjalizowanych, ale nie do wszystkich rodzajów obliczeń związanych z grafiką



**37. Przekształcenia geometryczne są wykonywane przy użyciu**

37	Przekształcenia geometryczne są wykonywane przy użyciu	Jednego układu współrzędnych
		Dwóch równorzędnego niezależnych układów współrzędnych
		Dwóch równorzędnego zależnych od siebie układów współrzędnych
		Trzech równorzędnego niezależnych układów współrzędnych



**38. Obraz szkieletowy sceny graficznej**

38	Obraz szkieletowy sceny graficznej	To układ prymitywów odwzorowujący obiekty sceny graficznej
		To układ linii pomocniczych w odwzorowujący obiekty sceny graficznej
		To układ prymitywów i linii pomocniczych odwzorowujący obiekty sceny graficznej



**39. Układ współrzędnych kamery syntetycznej i układ obserwatora**

39	Układ współrzędnych kamery syntetycznej i układ obserwatora	To dwa odrębne niezależne układy współrzędnych
		To dwa odrębne, ale zależne od siebie układy współrzędnych
		To dwie różne nazwy tego samego układu wsółrzędnych
		To układy, które nie biorą udziału w renderingu sceny graficznej



#### 40. Rzuty równoległe i rzuty prostokątne

40

Rzuty równoległe i rzuty prostokątne

To różne dwa rodzaje rzutów perspektywicznych

To różne dwa rodzaje rzutów aksonometrycznych

To różne dwa rodzaje płaskich rzutów geometrycznych



To różne dwa rodzaje rzutów geometrycznych, z których jeden jest rodzajem drugiego

#### 41. Wymiary i kąty zachowują

41

Wymiary i kąty zachowują

Niektóre rzuty prostokątne

Wszystkie rzuty prostokątne



Wszystkie rzuty równoległe

Rzuty Izometryczne

#### 42. Ustawienie geometrii bryły widzenia

42

Ustawianie geometrii bryły widzenia

Wymaga złożonych obliczeń całkowitoliczbowych

Nie wymaga żadnych skomplikowanych obliczeń, a jedynie ustawienia osi układów współrzędnych

Nie ma miejsca - bryła widzenia jest tworem stałym

Wymaga złożonych obliczeń zmiennoprzecinkowych

**Ustawienie geometrii bryły widzenia Wymaga złożonych obliczeń zmiennoprzecinkowych.**

#### 43. Ustawianie geometrii bryły widzenia

43

Ustawianie geometrii bryły widzenia

Wymaga zastosowania rzutowania perspektywicznego



Wymaga zastosowania rzutowania równoległego

Wymaga zastosowania rzutowania perspektywicznego lub równoległego

Wymaga zastosowania transformacji geometrycznych

#### 44. Wycinanie i zakrywanie linii niewidocznych

44

Wycinanie i zakrywanie linii niewidocznych

Zachodzi zawsze i usuwa wszystkie linie niewidoczne



Zachodzi zawsze, ale może pozostawić niektóre linie niewidoczne

Zachodzi w niektórych przypadkach i usuwa wszystkie linie niewidoczne

Zachodzi w niektórych przypadkach i może pozostawić niektóre linie niewidoczne

45. W przypadku animacji przekształcenia geometryczne wykonywane są

45	W przypadku animacji przekształcenia geometryczne wykonywane są	<p>Dla każdej klatki</p> <p>Przy znaczących zmianach sceny graficznej</p> <p>Przy znaczących zmianach sceny graficznej lub położenia obserwatora</p> 
----	---	--

46. Trzy główne procesy renderingu to

46	Trzy główne procesy renderingu to	<p>Teksturowanie, oświetlanie i cieniowanie</p> <p>Teksturowanie, oświetlanie i cieniowanie, dodawanie efektów specjalnych</p>  <p>Teksturowanie, mapowanie środowiskowe i dodawanie efektów specjalnych</p> <p>Teksturowanie, oświetlanie i cieniowanie, mapowanie środowiskowe</p>
----	-----------------------------------	---

47. Teksturowanie to

47	Teksturowanie to	<p>Nakładanie płaskich obrazów na trójwymiarowy szkielet</p>  <p>Nakładanie płaskich obrazów na płaski obraz szkieletu</p> <p>Nakładanie obrazów przestrzennych na płaski obraz szkieletu</p> <p>Nakładanie obrazów przestrzennych na trójwymiarowy szkielet</p>
----	------------------	--

48. Wycinanie (clipping) to

48	Wycinanie (ang. clipping) to	<p>Wyodrębnienie trójkąta z kwadratowej tekstury połączone ze skalowaniem do żądanej wielkości</p> <p>Wyodrębnienie trójkąta o żądanej wielkości z kwadratowej tekstury</p> <p>Nadawanie pikselom prymitywów atrybutów teksceli</p> <p>Wyodrębnienie prymitywu z przestrzennego szkieletu sceny graficznej</p> 
----	------------------------------	--

może  
to?



49. Próbkowanie punktowe, filtrowanie i korekcja perspektywy

49	Próbkowanie punktowe, filtrowanie i korekcja perspektywy	<p>Procesy składowe teksturowania</p> <p>Procesy składowe mapowania środowiskowego</p> <p>Procesy składowe alfa-blendingu</p> <p>Procesy składowe rzutowania</p> 
----	--	--

## 50. Wybór odpowiedniej MIP-mapy zależy od

50	Wybór odpowiedniej MIP-mapy zależy od	Wielkości MIP-mapy	
		Rozdzielcości MIP-mapy	
		Odległości prymitywu od obserwatora	
		Odległości prymitywy powierzchni rzutowania	

## 51. W praktyce do teksturowania najczęściej

51	W praktyce do teksturowania najczęściej	Wybierana jest jedna MIP-mapa zbliżona rozmiarami do prymitywu	
		Wybierana jest jedna MIP-mapa będąca wynikiem interpolacji dwóch MIP-map najbliższych prymitywowi	
		Wybierane są dwie MIP-mapy najbliższe prymitywowi odlegością od obserwatora	
		Wybierane są dwie MIP-mapy najbliższe prymitywowi rozmiarami	

## 52. Tworzenie MIP-map jest

52	Tworzenie MIP-map jest	Jednorazowe przy pierwszym użyciu tekstuury, po czym wszystkie MIP-mapy są zapamiętywane	
		Wielorazowe - MIP-mapy są przetrzymywane w pamięci do czasu utworzenia nowych z innej tekstuury	
		Jednorazowe lub wielorazowe w zależności od rozmiaru pamięci procesora graficznego	
		Niepotrzebny - są one wczytywane wraz z tekstuurą	

## 53. Filtrowanie tekstur polega na

53	Filtrowanie tekstur polega na	Dopasowaniu obszaru pobierania atrybutów tekstuury do rozmiaru prymitywu	
		Dopasowaniu obszaru pobierania atrybutów tekstuury do wielkości piksela obrazu	
		Dopasowaniu obszaru pobierania atrybutów tekstuury do odległości teksturowanego prymitywu od obserwatora	
		Dopasowaniu obszaru pobierania atrybutów tekstuury do odległości teksturowanego prymitywu od powierzchni rzutu	

## 54. W wyniku filtrowania tekstur metodą najbliższego sąsiada z MIP-mappingiem

54	W wyniku filtrowania tekstur metodą najbliższego sąsiada z MIP-mappingiem	Piksel przyjmuje kolor czterech sąsiadujących teksele MIP-mapy najbliższej pikselowej	
		Piksel przyjmuje interpolowany kolor teksele dwóch najbliższych MIP-map	
		Piksel przyjmuje kolor teksele najbliższej MIP-mapy, którego centrum leży najbliżej centrum piksela	
		Piksel przyjmuje kolor układu czterech tekselej najbliższej MIP-mapy, których centrum geometryczne leży najbliżej centrum piksela	

## 55. Filtrowanie trójliniowe korzysta

55	Filtrowanie trójliniowe korzysta	Z jednej MIP-mapy
		Z dwóch MIP-map
		Z trzech MIP-map
		Z teksturowej podstawowej

## 56. Korekcja perspektywy jest wykonywana w fazie

56	Korekcja perspektywy jest dokonywana w fazie	Przekształceń geometrycznych
		Nakładania teksturow
		Mapowania wypukłości
		Oświetlania i cieniowania

## 57. Mapowanie wypukłości odwzorowujące cienie powstające na powierzchni obiektu jest elementem

57	Mapowanie wypukłości odwzorowujące cienie powstające na powierzchni obiektu jest elementem	Przekształceń geometrycznych
		Oświetlania i cieniowania
		Teksturowania
		Nakładania efektów specjalnych

## 58. Środowiskowe mapowanie wybojów polega na połączeniu procesów

58	Środowiskowe mapowane wybojów polega na połączeniu procesów	Mapowania wypukłości oraz mapowania środowiska, ale nie podstawowej tekstuury
		Nakładania podstawowej tekstuury, mapy wypukłości oraz mapy środowiska
		Nakładania podstawowej tekstuury, mapy wypukłości, mapy środowiska i cieniowania
		Nakładania mapy wypukłości, mapy środowiska i cieniowania, ale nie podstawowej tekstuury

## 59. Mapowanie sferyczne wymaga pobierania atrybutów

59	Mapowanie sferyczne wymaga pobierania atrybutów	Z płaskiej mapy środowiska odzwierciedlającej całość otoczenia
		Z płaskiej mapy środowiska odzwierciedlającej otoczenie obiektu od stromy obserwatora
		Z sześciu map środowiska odzwierciedlających całość otoczenia
		Z przestrzennej mapy środowiska odzwierciedlającej otoczenie obiektu od stromy obserwatora

## 60. Mapowanie sześcienne wymaga pobierania atrybutów

Z płaskiej mapy środowiska będącej komplikacją sześciu map środowiska

Z płaskiej mapy środowiska odzwierciedlającej otoczenie obiektu od strony obserwatora

Równocześnie z sześciu map środowiska odzwierciedlających całość otoczenia

Z jednej przestrzennej mapy środowiska odzwierciedlającej otoczenie obiektu od strony obserwatora



## 61. Mapowanie trójwymiarowe to

Inna nazwa mapowania sześciennego

Inna nazwa mapowania sferycznego

Mapowanie w oparciu o tekstury przestrzenne

Mapowanie sześcienne połączone z nakładaniem tekstuury powierzchni



## 66. Podobne efekty wizualne uzyskuje się stosując

Cieniowanie płaskie i cieniowanie Gourauda



Cieniowanie Phonga i cieniowanie Gourauda

Cieniowanie płaskie i cieniowanie Phonga

Cieniowanie interpolacyjne i płaskie

## 67. Cieniowanie płaskie polega na

Przyporządkowaniu całej powierzchni trójkąta średniego poziomu jasności trójkątów sąsiednich

Interpolacji jasności wierzchołków trójkąta do jego wnętrza

Przyporządkowaniu całej powierzchni trójkąta średniego poziomu jasności jego wierzchołków



Przyporządkowaniu całej powierzchni trójkąta poziomu jasności jednego z wierzchołków

## 68. Cieniowanie Gourauda zakłada, iż pojedynczy odcień wnętrza każdego trójkąta powstaje

Z uśrednienia kolorów i natężenia światła występujących w każdym z jego wierzchołków



Z uśrednienia kolorów występujących w każdym z jego wierzchołków przy stałym natężeniu światła

Poprzez obliczenie kolorów i natężenia światła osobno dla każdego z pikseli trójkąta

Z uśrednienia kolorów i natężenia światła występujących w każdym z sąsiednich trójkątów

68

Cieniowanie Gourauda zakłada, iż pojedynczy odcień wnętrza każdego trójkąta powstaje

## 69. Cieniowanie z odbiciem rozproszonym dotyczy

69	Cieniowanie z odbiciem rozproszonym dotyczy	Wszystkich modeli cieniowania
		Modeli Gourauda i Phonga
		Jedynie modelu Gourauda
		Jedynie modelu Phonga

## 70. Algorytm śledzenia promieni analizuje

70	Algorytm śledzenia promieni analizuje	Promienie odbite i załamane w kierunku od źródła światła do obserwatora
		Tylko promienie odbite w kierunku od źródła światła do obserwatora
		Promienie odbite i załamane w kierunku od obserwatora do źródła światła
		Tylko promienie odbite i w kierunku od obserwatora do źródła światła

## 71. Energetyczna metoda cieniowania zakłada

71	Energetyczna metoda cieniowania zakłada	Dzieli obiekty sceny graficznej na niewielkie płaty i analizuje bilans energii pomiędzy poszczególnymi płatami
		Dzieli powierzchnie na niewielkie płaty i analizuje bilans energii pomiędzy płatami i poszczególnymi obiektami
		Analizuje bilans energii emitowanej i pochłanianej przez całe obiekty sceny graficznej
		Dokonuje bilansu energii pomiędzy poszczególnymi obiektami sceny graficznej

## 72. Do tworzenia mgły wolumetrycznej wykorzystywane są

72	Do tworzenia mgły wolumetrycznej wykorzystywane są	Funkcje wykładnicze obliczające są na bieżąco stopień zamglenia
		Specjalne tablice, w których zapisany jest stopień zamglenia, zależny od odległości od obserwatora
		Funkcje liniowe zamglenia obiektów wraz ze wzrostem ich odległości od obserwatora
		Półprzezroczyste tekstury trójwymiarowe

## 73. Sygnalizacja głębokości polega na

73	Sygnalizacja głębokości polega na	Wykładniczej zmianie koloru i odcienia przedmiotu w zależności od rozmiaru z-bufora
		Zmianie koloru i odcienia przedmiotu w zależności od dystansu od źródła światła
		Wykładniczej zmianie koloru i odcienia przedmiotu w zależności od dystansu od obserwatora
		Wykładniczej zmianie koloru i odcienia przedmiotu w zależności od kąta patrzenia wirtualnej kamery

#### 74. W trakcie rasteryzacji każdemu punktowi sceny graficznej przyporządkowywane są

74	W trakcie rasteryzacji każdemu punktowi sceny graficznej przyporządkowywane są	Dwie współrzędne istotne dla bufora ramki
		Trzy współrzędne istotne dla bufora ramki
		Dwie współrzędne, z których pierwsza jest wykorzystywana w buforze ramki
		Trzy współrzędne, z których dwie pierwsze są wykorzystywane w buforze ramki

#### 75. Antialiasing krawędziowy polega na usuwaniu „schodkowości” sceny graficznej

75	Antialiasing krawędziowy polega na usuwaniu „schodkowości” sceny graficznej	Z wykorzystaniem dwóch lub czterech identycznych, przesuniętych względem siebie ramek obrazu
		Poprzez odpowiednie rozmywanie krawędzi wzduż wszystkich rysowanych linii czy granic kolorów
		Poprzez odpowiednie rozmywanie krawędzi wzduż niektórych rysowanych linii czy granic kolorów
		Zwiększenie rozdzielczości bufora ramki

#### 76. Głębia ostrości w scenach generowanych sztucznie to

76	Głębia ostrości w scenach generowanych sztucznie to	Zjawisko niepożądane, które należy eliminować
		Zjawisko pożąданie, które wymaga zastosowania specjalnych algorytmów rozmywania wybranego obszaru
		Zjawisko niepożądane, którego nie można wyeliminować
		Zjawisko pożąданie, wynikające z właściwości ludzkiego sposobu widzenia

#### 77. Kalkulacja kolorów poszczególnych pikseli bufora ramki następuje z wykorzystaniem

77	Kalkulacja kolorów poszczególnych pikseli bufora ramki następuje z wykorzystaniem	Wyłącznie algorytmów oświetlenia
		Wyłącznie algorytmów cieniowania
		Wyłącznie algorytmów oświetlenia i cieniowania
		Algorytmów oświetlenia i cieniowania przy współudziale innych algorytmów

#### 78. Model oświetlenia Warna zakłada

78	Model oświetlenia Warna zakłada	Oświetlenie obiektów niezależne od ich pozycji, ale zależne od ich orientacji
		Oświetlenie obiektów niezależne od ich kształtu i koloru
		Stosowanie klap i stożków
		Oświetlenie obiektów niezależne od ich orientacji, ale zależne od ich pozycji

### 79. Odległość i kierunek źródła światła oraz pozycja obserwatora mają znaczenie przy

79	Odległość i kierunek źródła światła oraz pozycja obserwatora mają znaczenie przy	Oświetleniu punktowym z odbiciem rozproszonym	
		Oświetleniu punktowym z odbiciem zwierciadlanym	
		Oświetleniu kierunkowym z odbiciem rozproszonym	
		Oświetleniu kierunkowym z odbiciem zwierciadlanym	

### 80. Model cieniowania płaskiego zakłada

80	Model cieniowania płaskiego zakłada	Istnienie punktowych źródeł światła i odbić rozproszonych	
		Brak pozycji źródeł światła i istnienie odbić rozproszonych	
		Brak pozycji źródeł światła i istnienie odbić zwierciadlanych	
		Istnienie kierunkowych źródeł światła i odbić zwierciadlanych	

### 81. Interpolacja wartości węzłowych prymitywu trójkątnego ma znaczenie przy

81	Interpolacja wartości węzłowych prymitywu trójkątnego ma znaczenie przy	Cieniowaniu płaskim i cieniowaniu Gourauda	
		Cieniowaniu Phonga i cieniowaniu Gourauda	
		Cieniowaniu płaskim i cieniowaniu Phonga	
		Tylko przy cieniowaniu płaskim	

### 82. Przy cieniowaniu Phonga decydujące znaczenie ma

82	Przy cieniowaniu Phonga decydujące znaczenie ma	Interpolacja normalnych do powierzchni obiektów w wierzchołkach prymitywów do ich wnętrz	
		Średni poziom jasności trójkątów sąsiednich	
		Interpolacja poziomu jasności wierzchołków trójkąta do jego wnętrza	
		jasność jednego z wierzchołków prymitywu	

### 83. Cieniowanie Phonga zakłada, iż kolor i natężenie światła wewnętrz trójkąta

83	Cieniowanie Phonga zakłada, iż kolor i natężenie światła wewnętrz trójkąta	Są wynikiem uśrednienia kolorów i natężenia światła występujących w każdym z jego wierzchołków	
		Są wynikiem uśrednienia kolorów występujących w każdym z jego wierzchołków	
		Są obliczane osobno dla każdego z pikseli trójkąta	
		Są wynikiem interpolacji kolorów i natężenia światła występujących w każdym z jego wierzchołków	

#### 84. Złudzenie gładkości sferycznych obiektów złożonych nawet z niewielkiej liczby trójkątów powstaje

84	Złudzenie gładkości sferycznych obiektów złożonych nawet z niewielkiej liczby trójkątów powstaje	Wyłącznie przy cieniowaniu Gourauda
		Wyłącznie przy cieniowaniu Phonga
		Zarówno przy cieniowaniu Gourauda, jak i przy cieniowaniu Phonga
Przy trzech typach cieniowania: płaskim, Gourauda i Phonga		

#### 85. Algorytm śledzenia promieni pracuje poprawnie w przypadku

85	Algorytm śledzenia promieni pracuje poprawnie w przypadku	Przeźroczystych obiektów sceny graficznej	
		Przeźroczystych i nieprzeźroczystych obiektów sceny graficznej	
Nieprzeźroczystych obiektów sceny graficznej			
Obiektów sceny graficznej rozpraszających światło			

87.

Do tworzenia mgły wykładowiczej **wykorzystywane są nieliniowe funkcje obliczające na bieżąco stopień zamglenia.**

#### 88. Rasteryzacja to

88	Rasteryzacja to	Zamiana wszystkich parametrów generowanej sceny na zbiór pikseli gotowych do wysłania na monitor
		Zamiana wybranych obiektów generowanej sceny na zbiór pikseli gotowych do wysłania na monitor
		Zamiana zbioru pikseli reprezentujących poszczególne obiekty generowanej sceny na zbiór pikseli gotowych do wysłania na monitor
Zamiana wszystkich tekstuów na zbiór pikseli gotowych do mapowania tekstur		

89. Pod pojęciem z-bufora rozumiemy obszar pamięci RAM karty graficznej odpowiadający swoją wielkością

89	Pod pojęciem z-bufora rozumiemy obszar pamięci RAM karty graficznej odpowiadający swoją wielkością	Rozdzielczości ekranu i zastosowanej głębi sceny graficznej
		Rozdzielczości ekranu i modelowi koloru
		Zastosowanej głębi sceny graficznej
Rozdzielczości ekranu, zastosowanej głębi sceny graficznej i modelowi koloru		

#### 90. Bufor szablonowy to

90	Bufor szablonowy to	Obszar pamięci służący do zmniejszania obciążenia procesora graficznego i magistrali pamięci
		Obszar pamięci służący do przechowywania informacji o odległości obiektów od obserwatora
		Obszar pamięci służący do przechowywania informacji o atrybutach niektórych pikseli
Obszar pamięci służący do przechowywania informacji o obszarach cieni występujących na scenie graficznej		

## 91. Dithering polega na

91 Dithering polega na

Zawęźleniu liczby kolorów dostępnych w systemie do kolorów z dostępnej palety

Rozszerzeniu dostępnej w systemie palety kolorów do palety 24-bitowej

Symulacji koloru niedostępnego w systemie poprzez kompozycję kilku barw zbliżonych z dostępnej palety



Zmianie dostępnej palety kolorów na paletę o większej liczbie kolorów

## 92. Moduł T&L karty graficznej wykorzystuje najbardziej czasochłonne obliczenia związane

92 Moduł T&L karty graficznej wykonuje najbardziej czasochłonne obliczenia związane

Z oświetleniem sceny graficznej



Z cieniowaniem

Z oświetleniem sceny graficznej i cieniowaniem

Z teksturowaniem i nakładaniem map środowiska

## 93. Oświetlenie światłem otoczenia polega na

93 Oświetlenie światłem otoczenia polega na

Zastosowaniu rozproszonego odbicia światła

Zastosowaniu cieniowania płaskiego

Zastosowaniu światła odbitego od otoczenia obiektu

Stałym oświetleniu obiektów niezależnie od ich pozycji i orientacji



## 94. Oświetlenie Warna to

94 Oświetlenie Warna to

Oświetlenie mające podłożę fizyczne

Dowolne oświetlenie z ograniczaniem światła przez stożki i klapy



Odbicie zwierciadlane od punktu na płaszczyźnie oświetlanej przez punktowe źródło światła

Odbicie rozproszone od punktu na płaszczyźnie oświetlanej przez punktowe źródło światła

## 95. Cieniowanie interpolacyjne to

95 Cieniowanie interpolacyjne to

Cieniowanie płaskie, cieniowanie Gourauda i cieniowanie Phong'a

Cieniowanie Gourauda i cieniowanie Phong'a



Cieniowanie Gourauda, cieniowanie Phong'a i cieniowanie ze śledzeniem promieni

Cieniowanie ze śledzeniem promieni

## 96. Wielokrotne odbicia i załamania są uwzględniane

96	Wielokrotne odbicia i załamania światła są uwzględniane	Jedynie przez algorytm śledzenia promieni
		Jedynie przez algorytm metody energetycznej
		Przez algorytm śledzenia promieni i algorytm metody energetycznej

## 97. Wykorzystanie promieni odbitych i załamanych zakładają

97	Wykorzystanie promieni odbitych i załamanych zakładają	Wszystkie algorytmy śledzenia promieni
		Algorytm rekursywnego śledzenia promieni i metoda energetyczna
		Jedynie algorytmy metody energetycznej
		Każdy algorytm śledzenia promieni i metoda energetyczna



## 98. Metoda energetyczna opiera się na zasadzie

98	Metoda energetyczna opiera się na zasadzie	Odbić promieni świetlnych w środowisku zamkniętym
		Wpływu energii poszczególnych źródeł światła na odbicia i załamania promieni świetlnych
		Zachowania energii promieni świetlnych w środowiskach zamkniętym i otwartym

Zachowania energii promieni świetlnych w środowisku zamkniętym

## 99.

Mianem efektów atmosferycznych określane są **wszystkie efekty specjalne**.

## 100. W trakcie rasteryzacji następuje

100	W trakcie rasteryzacji następuje	Przeniesienie dwóch pierwszych współrzędnych punktów sceny graficznej na obraz dwuwymiarowy, podczas gdy trzecia jest tracona
		Przeniesienie dwóch wybranych współrzędnych punktów sceny graficznej na obraz dwuwymiarowy, podczas gdy trzecia jest tracona
		Przeniesienie dwóch pierwszych współrzędnych punktów sceny graficznej na obraz dwuwymiarowy, a trzeciej do oddzielnego bufora

## 101. Bufory: ramki, szablonowy i głębokości to bufory

101	Bufory: ramki, szablonowy i głębokości to bufory	O podobnym rozmiarze
		O różnym rozmiarze
		Z których dwa mają podobny rozmiar, a trzeci jest znacznie mniejszy

Z których dwa mają podobny rozmiar, a trzeci jest znacznie większy

### 103. Algorytm Z-bufora wykorzystuje

103

Algorytm Z-bufora wykorzystuje

Bufor głębokości i bufor szablonowy

Bufor głębokości i bufor ramki

Bufor głębokości, bufor ramki i bufor szablonowy

Jedynie bufor głębokości



### 104. Algorytm tworzenia cieni na scenie graficznej wykorzystuje bufor

104

Algorytm tworzenie cieni na scenie graficznej wykorzystuje bufor

Ramki



Głębokości

Akumulacyjny

Szablonowy

### 105. Bufor akumulacyjny jest wykorzystywany przez algorytm

105

Bufor akumulacyjny jest wykorzystywany przez algorytm

Teksturowania



Nakładania efektów specjalnych

Antialiasingu

Ditheringu

### 106. Złożoność obliczeń z antialiasingiem pełnoekranowym jest

106

Złożoność obliczeń z antialiasingiem pełnoekranowym jest

Dwu lub czterokrotnie większa niż bez antialiasingu



Zawsze dwukrotnie większa niż bez antialiasingu

Zawsze czterokrotnie większa niż bez antialiasingu

Porównywalna z tymi bez antialiasingu

### 110. Dla tego samego stopnia aproksymacji liczba płatów

110

Dla tego samego stopnia aproksymacji liczba płatów wielomianowych jest

Znacznie mniejsza niż płatów wielokątowych



Znacznie większa niż płatów wielokątowych

Zbliżona do liczby płatów wielokątowych

Identyczna z liczbą płatów wielokątowych

#### 111. Wielokąt stosowany w grafice 3d

111	Wielokąt stosowany w grafice 3D jest ograniczony	Zamkniętą sekwencją krawędzi	
		Dowolną sekwencję krawędzi	
		Zamkniętą lub otwartą sekwencję krawędzi	
		Sekwencję krawędzi lub luków	

#### 112. Wśród siatek wielokątowych

112	Wśród siatek wielokątowych reprezentacja bezpośrednia relacji pomiędzy wierzchołkami, krawędziami i wielokątami wymaga	Najkrótszych czasów operacji i najmniej miejsca w pamięci	
		Najdłuższych czasów operacji i najwięcej miejsca w pamięci	
		Najkrótszych czasów operacji i najwięcej miejsca w pamięci	
		Najdłuższych czasów operacji i najmniej miejsca w pamięci	

#### 113. Reprezentacja siatki wielokątowej

113	Reprezentacja siatki wielokątowej w postaci wskaźników na listę wierzchołków jest	Bardziej złożona niż reprezentacja w postaci wskaźników na listę krawędzi i wymaga algorytmów o krótszych czasach obliczeń	
		Mniej złożona niż reprezentacja w postaci wskaźników na listę krawędzi i wymaga algorytmów o krótszych czasach obliczeń	
		Bardziej złożona niż reprezentacja w postaci wskaźników na listę krawędzi i wymaga algorytmów o dłuższych czasach obliczeń	
		Mniej złożona niż reprezentacja w postaci wskaźników na listę krawędzi i wymaga algorytmów o dłuższych czasach obliczeń	

#### 114. Metoda rzutów wielokąta na płaszczyznę prostopadłe do osi układu współrzędnych pozwala na

114	Metoda rzutów wielokąta na płaszczyznę prostopadłe do osi układu współrzędnych pozwala na	Jedynie na sprawdzenie płaskości dowolnego wielokąta	
		Obliczenie pola dowolnego wielokąta bez względu na jego orientację w przestrzeni	
		Na wyznaczenie równania płaszczyzny aproksymującej płaszczyznę dowolnego wielokąta	
		Na wyznaczenie równania płaszczyzny aproksymującej płaszczyznę prymitywu, ale tylko dla prymitywu trójkątnego	

## 115 Krzywe Hermita są opisywane

115

Krzywe Hermita są opisywane iloczynem

Kwadratowej macierzy geometrii, macierzy bazowej Hermite'a i wektora kolejnych potęg parametru

Macierzy geometrii, macierzy bazowej Hermite'a o stałych elementach i wektora kolejnych potęg parametru



Macierzy geometrii, macierzy bazowej Hermite'a o zmiennych elementach i wektora kolejnych potęg parametru

Macierzy geometrii, macierzy bazowej Hermite'a o zmiennych elementach i wektora pochodnych kolejnych potęg parametru

## 116. Wielomiany Bernsteina

116

Wielomiany Bernsteina są

Funkcjami wagowymi krzywych Hermite'a i wszystkie są dodatnio określone

Funkcjami wagowymi krzywych Hermite'a, ale nie wszystkie są dodatnio określone

Funkcjami wagowymi krzywych Beziera i wszystkie są dodatnio określone



Funkcjami wagowymi krzywych Beziera, ale nie wszystkie są dodatnio określone

## 117. Płyty bikubiczne to przestrzenne

Płyty bikubiczne to przestrzenne rozwinięcie:

Dowolnych parametrycznych krzywych wielomianowych



Jedynie parametrycznych krzywych Hermite'a i Beziera

Jedynie trzech typów krzywych parametrycznych

Dowolnych krzywych wielomianowych, niekoniecznie parametrycznych

## 118. Krzywe stożkowe są

118

Krzywe stożkowe są do celów grafiki komputerowej modelowane przy pomocy

Parametrycznych wielomianów trzeciego stopnia



Parametrycznych wielomianów drugiego stopnia

Prostych funkcji nieparametrycznych nie będących wielomianami

Prostych funkcji parametrycznych nie będących wielomianami

## 119. Metody fraktalne i gramatyki grafowe

119	Metody fraktalne i gramatyki grafowe to	Specjalizowane metody grafiki komputerowej generowania obiektów o dowolnej budowie
		Specjalizowane metody grafiki komputerowej generowania obiektów o symetrycznej budowie
		Specjalizowane metody grafiki komputerowej zakładające dokładne lub statystyczne samopodobieństwo generowanych obiektów
		Specjalizowane metody grafiki komputerowej zakładające dokładne samopodobieństwo generowanych obiektów

## 120. Modele oparte o gramatyki grafowe wymagają

120	Modele oparte o gramatyki grafowe wymagają:	Zawsze reprezentacji gramatycznej i geometrycznej, a w niektórych przypadkach rejestracji wieku
		Zawsze reprezentacji gramatycznej i geometrycznej oraz rejestracji wieku
		Zawsze reprezentacji gramatycznej i rejestracji wieku, a niekiedy reprezentacji graficznej
	Zawsze reprezentacji gramatycznej, a niekiedy reprezentacji graficznej	
	5 / 27	

## 121. Graftalami nazywamy

121	Graftalami nazywamy	Obiekty utworzone przy pomocy specjalizowanych metod modelowania kształtu
		Obiekty zbudowane przy pomocy fraktalnych metod opisu roślin
		Obiekty zbudowane przy pomocy metody opisu roślin opartej o metajęzyki równoległych gramatyk grafowych
	Obiekty zbudowane z wykorzystaniem płatów bikubicznych	

## 122. Modelowanie obiektów rzeczywistych

122	Modelowanie obiektów rzeczywistych	Jest zawsze aproksymacją ich kształtów
		Jest na ogół aproksymacją ich kształtów
		Jest zawsze dokładnym opisem ich kształtów
		Jest na ogół dokładnym opisem ich kształtów

123. Model powierzchni znanego z OpenGL czajniczka Martina Newella jest reprezentowany przez

123	Model powierzchni znanego z OpenGL czajniczka Martina Newella jest reprezentowany przez	Zbiór gładkich powierzchni krzywoliniowych 
		Zbiór gładkich elementów płaszczyzn
		Zbiór prymitywów trójkątnych
		Siatkę wielokątową

124. Siatka wielokątów to

124	Siatka wielokątów to	Zbiór figur płaskich wykorzystywanych w grafice komputerowej
		Zbiór figur płaskich lub przestrzennych ułożonych w odpowiedni wzór
		Zbiór połączonych powierzchni płaskich graniczonych zamkniętymi łamankami 

125. Złożoność algorytmów opisujących

125	Złożoność algorytmów opisujących płaty wielomianowe jest	Zbyt wysoka, aby płyty takie stosować
		Porównywalna ze złożonością algorytmów opisujących płaty wielokątowe
		Znacznie większa od złożoności algorytmów opisujących płaty wielokątowe 

126. W przypadku siatek

126	W przypadku siatek wielokątowych	Każda krawędź łączy dwa wierzchołki i jest wspólna najwyżej dla dwóch wielokątów
		Każda krawędź łączy dwa wierzchołki i jest wspólna przynajmniej dla dwóch wielokątów 
		Każda krawędź łączy przynajmniej dwa wierzchołki i jest wspólna najwyżej dla dwóch wielokątów
		Każda krawędź łączy dwa wierzchołki i może, ale nie musi być wspólna przynajmniej dla dwóch wielokątów

## 127. Reprezentacja bezpośrednia relacji

127	Reprezentacja bezpośrednią relacji pomiędzy wierzchołkami, krawędziami i wielokątami zakłada, iż	Każdy wielokąt jest opisyany przez listę współrzędnych wierzchołków zapamiętyanych w dowolnej kolejności
		Każdy wielokąt jest opisyany przez listę współrzędnych wierzchołków zapamiętyanych w kolejności, w jakiej napotyka się je poruszając się wokół wielokąta
		Każdy wielokąt jest opisyany przez listę krawędzi zapamiętyanych w kolejności, w jakiej napotyka się je poruszając się wokół wielokąta
		Każdy wielokąt jest opisyany przez listę krawędzi zapamiętyanych w dowolnej kolejności

## 128. Reprezentacja siatki wielokątowej

128	Reprezentacja siatki wielokątowej za pomocą wskaźników na listę wierzchołków jest	Bardziej złożona od reprezentacji bezpośredniej i mniej złożona od reprezentacji na listę krawędzi
		Mniej złożona od reprezentacji bezpośredniej i bardziej złożona od reprezentacji na listę krawędzi
		Bardziej złożona od reprezentacji bezpośredniej i bardziej złożona od reprezentacji na listę krawędzi
		Mniej złożona od reprezentacji bezpośredniej i mniej złożona od reprezentacji na listę krawędzi

## 129. Metoda rzutów wielokąta na płaszczyznę prostopadłe do osi układu współrzędnych wyznacza

129	Metoda rzutów wielokąta na płaszczyznę prostopadłe do osi układu współrzędnych wyznacza	Współrzędne rzutów węzłów wielokąta na płaszczyznę aproksymującą płaszczyznę wielokąta
		Współczynniki płaszczyzny aproksymującej układ węzłów prymitywu wielokątowego
		Kosinusy kierunkowe nachylenia płaszczyzny prymitywu do osi układu współrzędnych
		Wektor styczny do płaszczyzny wielokąta

## 130. Równania opisujące krzywe Beziera

130	Równania opisujące krzywe Beziera różnią się od równań opisujących krzywe Hermite'a	Jedynie macierzami bazowymi
		Macierzami bazowymi i macierzami geometrii
		Macierzami bazowymi, macierzami geometrii i wektorem potęg parametru
		Jedynie macierzami geometrii

### 131. Funkcje wagowe krzywych Hermite'a

131

Funkcje wagowe krzywych Hermite'a

Są symetryczne i wszystkie są dodatnio określone

Są symetryczne i w większości dodatnio określone,



Nie są symetryczne ani dodatnio określone

Nie są symetryczne, ale wszystkie są dodatnio określone

### 132. Koncepcja krzywych NURBS

132

Koncepcja krzywych NURBS zakłada

Podział krzywych na segmenty o równych bądź nierównych zakresach parametru, ale równych wagach węzłów

Podział krzywych na segmenty o różnych zakresach parametru i różnych wagach poszczególnych węzłów

Podział krzywych na segmenty o różnych zakresach parametru, ale dowolnych wagach poszczególnych węzłów

Podział krzywych na segmenty o równych bądź nierównych zakresach parametru i dowolnych wagach węzłów



### 133. Krzywe stożkowe to krzywe powstałe z przecięcia

133

Krzywe stożkowe to krzywe powstałe z przecięcia stożka

Płaszczyznami o dowolnych nachyleniach



Płaszczyznami pionowymi bądź poziomymi

Płaszczyznami o dowolnych nachyleniach z wyjątkiem płaszczyzn pionowych

Płaszczyznami o dowolnych nachyleniach z wyjątkiem płaszczyzn poziomych

6 / 27

### 134. Modele fraktalne i gramatyki grafowe są generowane w trakcie

134

Modele fraktalne i gramatyki grafowe są generowane w trakcie

Skończonych deterministycznych procesów iteracyjnych bądź nieiteracyjnych

Skończonych procesów iteracyjnych z udziałem funkcji losowych

Skończonych deterministycznych procesów iteracyjnych

Skończonych procesów iteracyjnych bez udziału funkcji losowych



### 135. System funkcji iterowanych (IFS) w grafice komputerowej to:

135

System funkcji iterowanych (IFS) w grafice komputerowej to:

Biblioteka zawierająca procedury wykorzystywane przez algorytmy gramatyk grafowych

Biblioteka zawierająca procedury wykorzystywane przez algorytmy budowy obiektów fraktałnych

Rodzina funkcji, za pomocą których konstruuje się fraktały samopodobne



Rodzina funkcji, za pomocą których konstruuje się gramatyki grafowe

### 136. Gramatyki Reffey'a

136

Gramatyki Reffey'a

Są narzędziem do budowy roślinnych obiektów graficznych w oparciu o dowolne gramatyki grafowe

Są narzędziem do budowy roślinnych obiektów graficznych w oparciu o gramatyki grafowe uwzględniające informację biologiczną



Są narzędziem do budowy dowolnych obiektów graficznych w oparciu o gramatyki grafowe uwzględniające ich środowisko

Są narzędziem do budowy dowolnych obiektów graficznych w oparciu o gramatyki grafowe uwzględniające ich środowisko

### 154. Współrzędne jednorodne

154

Współrzędne jednorodne:

Rozszerzają rozmiar przestrzeni współrzędnych kartezjańskich o jeden



Zawężają rozmiar przestrzeni współrzędnych kartezjańskich o jeden

Zachowują rozmiar przestrzeni współrzędnych kartezjańskich

Rozszerzają przestrzeń współrzędnych kartezjańskich o wielkość zależną od rozmiaru przestrzeni rzeczywistej

### 155. Translacja, rotacja i zmiana

155

Translacja, rotacja i zmiana skali mają reprezentacje macierzowe

Jedynie we współrzędnych kartezjańskich

Jedynie we współrzędnych jednorodnych

Zarówno we współrzędnych kartezjańskich jak i jednorodnych



Wszystkie we współrzędnych jednorodnych, a niektóre również we współrzędnych kartezjańskich

### 156. Translacja i rotacja są prze

156

Translacja i rotacja są przekształceniami

Multiplikatywnymi

Addytywnymi



Translacja jest addytywna, a rotacja multiplikatywna

Translacja jest multiplikatywna, a rotacja addytywna

### 157. Macierz zastępuje dwóch kolejnych przekształceń elementarnych jest:

157

Macierz zastępuje dwóch kolejnych przekształceń elementarnych jest:

Sumą macierzy przekształceń elementarnych

Iloczynem macierzy przekształceń elementarnych - pierwsza z macierzy iloczynu jest macierzą drugiego przekształceń



Sumą lub iloczynem macierzy przekształceń elementarnych - zależnie od rodzaju przekształceń

Iloczynem macierzy przekształceń elementarnych - pierwsza z macierzy iloczynu jest macierzą pierwszego przekształceń

### 158. Aby obiekty w trakcie przekształcenia

158

Aby obiekty w trakcie przekształcenia zachowywały się jak ciała sztywne, to macierz takiego przekształcenia musi być:

Macierzą ortonormalną

Dowolną macierzą kwadratową o rozmiarze przestrzeni przekształcenia



Macierzą ortogonalną

Macierzą przekształcenia afinycznego

### 159. Translacja i rotacja względem wybranego

159

Translacja i rotacja względem wybranego punktu obiektu są przekształceniami

Elementarnymi

Złożonymi

Translacja elementarnym, a rotacja złożonym



Rotacja elementarnym, a translacja złożonym

### 160. Elementarne przekształcenie zmiany

160

Elementarne przekształcenie zmiany skal powoduje

Powiększenie lub pomniejszenie obiektów wraz z ich oddalaniem się od początku układu współrzędnych

Powiększenie lub pomniejszenie obiektów bez zmiany miejsca ich położenia

Powiększenie lub pomniejszenie obiektów wraz z odpowiadającą im zmianą położenia tych obiektów względem początku układu współrzędnych



Powiększenie lub pomniejszenie obiektów wraz z ich przybliżaniem się do początku układu współrzędnych

### 161. Współrzędne kartezjańskie punktu o współrzędnych

161	Współrzędne kartezjańskie punktu o współrzędnych jednorodnych $(x, y, w)$ to:	Trójka $(x/w, y/w, 1)$
		Trójka $(x/w, y/w, w)$
		Para $(x/w, y/w)$
		Para $(x, y)$

### 162. We współrzędnych kartezjańskich reprezentacje macierzowe mają następujące przekształcenia

162	We współrzędnych kartezjańskich reprezentacje macierzowe mają następujące przekształcenia elementarne	Translacja, rotacja i zmiana skali
		Translacja i rotacja
		Rotacja i zmiana skali
		Translacja i zmiana skali

### 163. Spośród trzech przekształceń elementarnych

163	Spośród trzech przekształceń elementarnych: translacja, rotacja i zmiana skali multiplikatywne są	Wszystkie trzy
		Translacja i rotacja
		Rotacja i zmiana skali
		Tylko zmiana skali

### 164. Dowolne złożenie macierzy obrotu, przesunięcia i skalowania:

164	Dowolne złożenie macierzy obrotu, przesunięcia i skalowania	Jest przekształceniem afnicznym
		Thumbs up
		Jest przekształceniem ortonormalnym
		Nie jest przekształceniem ortonormalne ani afnicznym
		Jest przekształceniem równocześnie ortonormalnym i afnicznym

### 165. Przekształcenia pochylające

165	Przekształcenia pochylające	Są zawsze przekształceniami elementarnymi
		Są w pewnych warunkach przekształceniami elementarnymi
		Nie są przekształceniami elementarnymi
		Thumbs down
		Nie ma takich przekształceń

## 166. Rysowanie elips o dowolnych osiach w grafice

166

Rysowanie elips o dowolnych osiach w grafice komputerowej następuje z wykorzystaniem

Reprezentacji funkcyjne elipsy w postaci uwikłanej

Reprezentacji funkcyjne elipsy w postaci parametrycznej

Algorytmu rysowania okręgu przy równoczesnych: zmianie skali, rotacji i translacji



Algorytmu rysowania okręgu przy równoczesnych: zmianie skali i rotacji

## 361. Nowoczesna OpenGL: OpenGL jest odpowiedzialna za

361

Nowoczesna OpenGL: OpenGL jest odpowiedzialna za:

rendering oraz tworzenie kontekstu i okna renderingu.



rendering i tworzenie kontekstu, ale nie okna renderingu.

rendering i tworzenie okna renderingu, ale nie kontekstu.

rendering, ale nie za tworzenie kontekstu ani okna renderingu.

## 362. Nowoczesna OpenGL: Przy przetwarzaniu strumienia

362

Nowoczesna OpenGL: Przy przetwarzaniu strumienia grafiki komputerowej przez OpenGL opcjonalne są następujące szadery:

wszystkie z wymienionych w pozostałych punktach



wierzchołków

geometrii

fragmentów

## 363. Nowoczesna OpenGL: Program szadera geometrii analizuje

363

Nowoczesna OpenGL: Program szadera geometrii analizuje atrybuty:

wierzchołków



fragmentów

wierzchołków i fragmentów

żadnych z wymienionych w pozostałych punktach

## 364. Nowoczesna OpenGL: Każdy Vertex Buffer

364

Nowoczesna OpenGL: Każdy Vertex Buffer Object (VBO) zawiera dane wystarczające do

narysowania dowolnego prymitywu funkcją glDrawArrays

narysowania dowolnego prymitywu funkcją glDrawElements

zdefiniowania kształtu dowolnego prymitywu



zdefiniowania kształtu i koloru dowolnego prymitywu

### 365. Nowoczesna OpenGL: OpenGL Shading Language to

język programowania podobny do C++ kompliowany przez OpenGL



365

Nowoczesna OpenGL: OpenGL Shading Language to

język programowania podobny do C kompliowany przez OpenGL

język programowania podobny do C kompliowany przez oprogramowanie karty graficznej

język programowania odmienny od C i C++ kompliowany przez OpenGL

### 366. Nowoczesna OpenGL: Szader fragmentów wymaga atrybutów

zwykle są mu przekazywane poprzez szader wierzchołków bez żadnych zmian

zwykle są mu przekazywane poprzez szader wierzchołków po zmianach

366

Nowoczesna OpenGL: Szader fragmentów wymaga atrybutów, takich jak kolor czy współrzędne tekstury, które

zawsze są przekazywane szaderowi fragmentów bezpośrednio przez kod programu OpenGL



zawsze są mu przekazywane przez szader wierzchołków, a nigdy bezpośrednio przez kod programu OpenGL

### 367. Nowoczesna OpenGL: Instrukcje A i B języka GLSL

A: `vec4(position, 0.0, 1.0);`  
B: `vec4(position.x, position.y, 0.0, 1.0);`

są zawsze równoważne

są instrukcjami z których pierwsza jest błędna

są instrukcjami z których druga jest błędna

są równoważne, gdy position jest typu vec2



### 368. Nowoczesna OpenGL: Szader fragmentów generuje atrybuty fragmentów na podstawie

jedynie koloru wierzchołków

jedynie współrzędnych tekstury

koloru wierzchołków, współrzędnych tekstury oraz innych danych wspólnych dla wszystkich wierzchołków

jedynie koloru wierzchołków i współrzędnych tekstury

368

Nowoczesna OpenGL: Szader fragmentów generuje atrybuty fragmentów na podstawie

### 369. Nowoczesna OpenGL: SFML jest

369

Nowoczesna OpenGL: SFML jest

wieloplatformową biblioteką multimedialną języka C



wieloplatformową biblioteką multimedialną języka C++

wieloplatformową biblioteką graficzną języka C++

wieloplatformową biblioteką graficzną języka C

### 370. Nowoczesna OpenGL: Przedstawiona instrukcja

370

```
sf::Window window(sf::VideoMode(800, 600),  
                  "Moje okno", sf::Style::Close, settings);
```

to instrukcja z biblioteki SFML tworząca kontekst OpenGL w istniejącym oknie

Nowoczesna OpenGL: Przedstawiona instrukcja

to instrukcja z biblioteki SFML tworząca okno renderingu, ale nie kontekst OpenGL



to instrukcja z biblioteki SFML tworząca okno renderingu i kontekst OpenGL

to instrukcja z biblioteki SFML zmieniająca rozmiar okna renderingu

### 371. Nowoczesna OpenGL: Przedstawiona klasa pozwala określić:

371

```
sf::ContextSettings
```

rozmiary okna renderingu, poziom antialiasingu i parametry buforów:  
głębokości i szablonowego



poziom antialiasingu i parametry buforów: głębokości i szablonowego

Nowoczesna OpenGL: Przedstawiona klasa pozwala określić:

poziom antialiasingu, ale nie parametry buforów

parametry buforów: głębokości i szablonowego, ale nie poziom antialiasingu

### 372. Nowoczesna OpenGL: Przy przetwarzaniu strumienia grafiki komputerowej przez OpenGL niezbędne

372

Nowoczesna OpenGL: Przy przetwarzaniu strumienia grafiki komputerowej przez OpenGL niezbędne są szadery:

wierzchołków, geometrii i fragmentów

wierzchołków i fragmentów



wierzchołków i geometrii

tylko fragmentów

### 373. Nowoczesna OpenGL: Szader geometrii

373	Nowoczesna OpenGL: Szader geometrii	ogranicza objętość danych przekazywanym dalej w potoku graficznym
		zwiększa objętość danych przekazywanym dalej w potoku graficznym
		zwiększa bądź zmniejsza objętość danych przekazywanym dalej w potoku graficznym

### 374. Nowoczesna OpenGL: Vertex Buffer Object (VBO) może zawierać dane

374	Nowoczesna OpenGL: Vertex Buffer Object (VBO) może zawierać dane dotyczące:	jedynie pozycji wierzchołków
		pozycji wierzchołków i ich kolorów
		pozycji wierzchołków i kolorów fragmentów
		pozycji wierzchołków i typu prymitywów

### 375. Nowoczesna OpenGL: obsługa shaderów

375	Nowoczesna OpenGL: Obsługa shaderów	rozpoczęła się od wersji 3.2 OpenGL
		rozpoczęła się od wersji 2.1 OpenGL
		rozpoczęła się od wersji 2.0 OpenGL
		była możliwa w każdej wersji OpenGL

### 376. Nowoczesna OpenGL: Oprócz typów oferowanych

376	Nowoczesna OpenGL: Oprócz typów oferowanych przez język C, GLSL posiada wbudowane	typy wektorowe identyfikowane przez "vec1" ... "vec4", ale nie typy macierzowe
		typy wektorowe i macierzowe identyfikowane przez "vec1" ... "vec4" i "mat1" ... "mat4"
		typy wektorowe i macierzowe identyfikowane przez "vec2" ... "vec4" i "mat2" ... "mat4"

### 377. Nowoczesna OpenGL: Przedstawiona instrukcja jest instrukcją

```
vec4(position, 1.0);
```

377

Nowoczesna OpenGL: Przedstawiona instrukcja jest Instrukcją GLSL, która

zawsze tworzy wektor dwuelementowy

może utworzyć wektor o różnej liczbie elementów w zależności od typu position

zawsze tworzy wektor czteroelementowy

jest błędna



### 378. Nowoczesna OpenGL: Kolory w OpenGL są reprezentowane przez

trzy liczby całkowite z zakresu( 0 , 255)

trzy liczby zmiennoprzecinkowe z zakresu (0.0, 1.0)

cztery liczby całkowite z zakresu ( 0 , 255)

cztery liczby zmiennoprzecinkowe z zakresu (0.0, 1.0)

378

Nowoczesna OpenGL: Kolory w OpenGL są reprezentowane przez:



### 379. Nowoczesna OpenGL: Szader wierzchołków dokonuje:

przekształceń, rzutowań i łączenia wierzchołków w prymitywy



przekształceń i rzutowań, a wynik przekazuje jednostkom łączącym wierzchołki w prymitywy

przekształceń, a wynik przekazuje jednostkom dokonującym rzutowań i łączenia wierzchołków w prymitywy

przekształceń wcześniej zrzutowanych współrzędnych wierzchołków oraz łączenia wierzchołków w prymitywy

379

Nowoczesna OpenGL: Szader wierzchołków dokonuje

### 380. Nowoczesna OpenGL: Przedstawiony fragment kodu jest przeznaczony do instrukcji

```
#version 150
in vec2 position;
void main(){
    gl_Position = vec4(position, 0.0, 1.0);
}
```

380

bazowego programu OpenGL

szadera wierzchołków



Nowoczesna OpenGL: Przedstawiony fragment kodu jest przeznaczony dla

### 381. Nowoczesna OpenGL: Zaprezentowany fragment programu służy do

```
GLuint vertexShader = glCreateShader(GL_VERTEX_SHADER);  
glShaderSource(vertexShader, 1, &vertexSource, NULL);
```

381

Nowoczesna OpenGL: Zaprezentowany fragment programu służy do

utworzenia obiektu szadera i komplikacji jego kodu

utworzenia obiektu szadera i jego kodu

utworzenia obiektu szadera i załadowania do niego kodu

utworzenia obiektu szadera i rezerwacji pamięci karty graficznej



### 382. Nowoczesna OpenGL: Przedstawione instrukcje powodują

```
GLint status;  
glGetShaderiv(vertexShader, GL_COMPILE_STATUS,  
               &status);
```

382

Nowoczesna OpenGL: Przedstawione instrukcje powodują

sprawdzenie poprawności programu OpenGL

sprawdzenie poprawności komplikacji szadera



pobranie informacji o błędach komplikacji szadera

utworzenie informacji o błędach komplikacji szadera

### 383. Nowoczesna OpenGL: Przedstawiona funkcja służy openg

```
glVertexAttribPointer(posAttrib, 2, GL_FLOAT,  
                      GL_FALSE, 0, 0);
```

383

Nowoczesna OpenGL: Przedstawiona funkcja służy do

określania sposobu przenoszenia danych z tablicy wierzchołków do bufora wierzchołków

reorganizacji danych w buforze buforze wierzchołków

ustalenia sposobu komunikacji pomiędzy tablicą danych wierzchołkowych, a buforem bufora wierzchołków



określanie sposobu pobierania danych z bufora wierzchołków

384. Nowoczesna OpenGL: Przedstawione instrukcje powodują (utworzenie bufora przechowującego powiązania między atrybutami).

14 / 27

```
GLuint vao;  
 glGenVertexArrays(1, &vao);
```

384

Nowoczesna OpenGL: Przedstawione instrukcje powodują

utworzenie bufora szablonowego

utworzenie bufora przechowującego dane wierzchołkowe



utworzenie bufora przechowującego powiązania między atrybutami

utworzenie bufora przechowującego dane wierzchołkowe i powiązania między atrybutami

385. Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje

```
glDrawArrays(GL_TRIANGLES, 0, 5);
```

385

Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje

wykreślenie jednego trójkąta



wykreślenie dwóch trójkątów

wykreślenie pięciu trójkątów

błąd komplikacji

386. Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje

```
GLuint shaderProgram = glCreateProgram();
```

386

Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje

utworzenie programu szaderów z wcześniej załadowanych szaderów



utworzenie pustego programu szaderów

kompilację i linkowanie programu szaderów

aktywacją istniejącego programu szaderów

387. Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje

```
glBindFragDataLocation(shaderProgram, 0, "outColor");
```

387

Nowoczesna OpenGL: Przedstawiona instrukcja spowoduje pobranie uchwytu do wektora "outColor" umieszczonego w

programie szaderów

programie szdera wierzchołków

programie szdera fragmentów



programie bazowym

### 388? Nowoczesna OpenGL: Przedstawione instrukcje spowodują (outcolor)

```
gl_Position = vec4(position, 0.0, 1.0);  
outColor = vec4(1.0, 1.0, 1.0, 1.0);
```

przypisanie wektorów do wbudowanych zmiennych "gl\_Position" i "outColor"

przypisanie wektorów do zdefiniowanych przez użytkownika zmiennych "gl\_Position" i "outColor"



przypisanie wektorów do wbudowanej zmiennej "gl\_Position" i zdefiniowanej przez użytkownika zmiennej "outColor"

przypisanie wektorów do wbudowanej zmiennej "outColor" i zdefiniowanej przez użytkownika zmiennej "gl\_Position"

388

Nowoczesna OpenGL: Przedstawione instrukcje spowodują

### 957. Obrazy są dwuwymiarowymi macierzami pikseli i należy

957

Obrazy są dwuwymiarowymi macierzami pikseli i należy przywiązać je do obiektu typu

GL\_TEXTURE lub GL\_TEXTURE\_2D



GL\_TEXTURE\_2D lub GL\_TEXTURE2D

GL\_TEXTURE

GL\_TEXTURE\_2D

### 958. OpenGL zakłada, że współrzędne tekstury obrazu leżą w przedziale:

958

OpenGL zakłada, że współrzędne tekstury obrazu leżą w przedziale

od 0.0 do 1.0, gdzie adres (0.0, 0.0) jest jej górnym lewym rogiem



od 0.0 do 1.0, gdzie adres (0.0, 0.0) jest jej dolnym lewym rogiem

od -1.0 do 1.0, gdzie adres (-1.0, -1.0) jest jej dolnym lewym rogiem

od -1.0 do 1.0, gdzie adres (-1.0, -1.0) jest jej górnym lewym rogiem

### 959. Przypisanie współrzędnym tekstury

959

Przypisanie współrzędnym tekstury wychodzącym poza zakres koloru obramowania następuje po wybraniu opcji

GL\_REPEAT

GL\_CLAMP\_TO\_BORDER

GL\_MIRRORED\_REPEAT

GL\_CLAMP\_TO\_EDGE



960. Przedstawiony fragment kodu prowadzi do

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S,  
                GL_REPEAT);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T,  
                GL_REPEAT);
```

960

Przedstawiony fragment kodu prowadzi do

ustawienia parametrów skalowania tekstur

ustawienia parametrów filtrowania tekstur

ustawienia parametrów jednostki teksturowej

ustawienia parametrów wycinania tekstur



961. Metody filtrowania GL\_LINEAR

A: glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER,  
 GL\_LINEAR);  
B: glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER,  
 GL\_LINEAR);

961

A

B

A;B

każdym innym oprócz A i B



962. Generowanie MIP-map następuje poprzez wywołanie funkcji:

962

Generowanie MIP-map następuje poprzez wywołanie funkcji:

A

B

C

D



10 / 37

963. Filtrowanie trójliniowe następuje przy użyciu następującej metody

GL\_NEAREST\_MIPMAP\_NEAREST

GL\_LINEAR\_MIPMAP\_NEAREST

GL\_LINEAR\_MIPMAP\_LINEAR

GL\_NEAREST\_MIPMAP\_LINEAR

963

Filtrowanie trójliniowe następuje przy użyciu następującej metody filtrowania



#### 964. Przedstawiony fragment kodu powoduje

```
float pixels[] = {  
    0.0f, 0.0f, 0.0f,    1.0f, 1.0f, 1.0f,  
    1.0f, 1.0f, 1.0f,    0.0f, 0.0f, 0.0f  
};  
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, 2, 2, 0,  
             GL_RGB, GL_FLOAT, pixels);
```

964

Przedstawiony fragment kodu powoduje

załadowanie obrazu tekstury do pamięci operacyjnej komputera

załadowanie obrazu tekstury do pamięci karty graficznej

nałożenie testury przez shader wierzchołków



nałożenie testury przez shader fragmentów

#### 965. Wykonywanie zaprezentowanego fragmentu kodu wymaga współpracy

```
int width, height;  
unsigned char* image = SOIL_load_image("img.png",  
                                       &width, &height, 0, SOIL_LOAD_RGB);  
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height,  
             0, GL_RGB, GL_UNSIGNED_BYTE, image);
```

965

Wykonanie zaprezentowanego fragmentu kodu wymaga współpracy

biblioteki OpenGL z szaderem wierzchołków

biblioteki OpenGL z szaderem fragmentów

biblioteki OpenGL z biblioteką pomocniczą bez udziału szaderów



biblioteki OpenGL z biblioteką pomocniczą przy udziale szaderów

#### 966. Obrazy tekstur mogą być przesyłane do

966

Obrazy tekstur mogą być przesyłane do szadera

fragmentów bezpośrednio i tam nakładane



wierzchołków bezpośrednio i tam nakładane

fragmentów za pośrednictwem szadera wierzchołków i tam nakładane

fragmentów bez udziału szadera wierzchołków, ale za pośrednictwem szadera geometrii

### 967. Przedstawiony fragment kodu szadera fragmentów

```
uniform sampler2D texKitten;
uniform sampler2D texPuppy;
void main(){
    vec4 colKitten = texture(texKitten, Texcoord);
    vec4 colPuppy = texture(texPuppy, Texcoord);
    ...
};
```

967

Przedstawiony fragment kodu szadera fragmentów

wymaga pracy jednej jednostki teksturowującej o nazwie texture

wymaga pracy jednej jednostki teksturowującej o nazwie sampler2D

wymaga pracy dwóch jednostek teksturowujących



nie wymaga pracy żadnej jednostki teksturowującej

### 968. Funkcja “glActiveTexture” określa obiekt tekstury jest związany z konkretną

968

Funkcja “glActiveTexture” określa obiekt tekstury jest związany z konkretną jednostką teksturowującą

i jej jawne wywołanie jest wymagane w każdym z przypadków teksturowania

i jej jawne wywołanie jest wymagane w niektórych przypadkach teksturowania

i jej wywołanie podczas teksturowania następuje automatycznie

i jej jawne wywołanie jest wymagane nawet wtedy, gdy nie następuje proces teksturowania



### 969. Przedstawiony kod obrazu jest kodem

```
void main(){
    vec4 colKitten = texture(texKitten, Texcoord);
    vec4 colPuppy = texture(texPuppy, Texcoord);
    outColor = mix(colKitten, colPuppy, 0.5);
}
```

969

Przedstawiony kod programu jest kodem

szadera wierzchołków

szadera fragmentów



szadera geometrii

podstawowym biblioteki OpenGL

## 972. Kamera w programie OpenGL jest standardowo

	nieruchoma, umieszczona w punkcie (0,0,0, 1,0) i zwrócona w kierunku osi Oz
972	<input type="checkbox"/>
Kamera w programie OpenGL jest standardowo:	nieruchoma, umieszczona w punkcie (0,0,0, 0,0) i zwrócona w kierunku osi Oz
	nieruchoma, umieszczona w punkcie (0,0,0, 0,0) i zwrócona w kierunku przeciwnym do kierunku osi Oz
	nieruchoma, umieszczona w punkcie (0,0,0, 1,0) i zwrócona w kierunku przeciwnym do kierunku osi Oz

## 973. Rzutowanie przekształca wierzchołek do układu współrzędnych przycinania, które są równocześnie unormowanymi współrzędnymi urządzenia

	w przypadku rzutowania perspektywicznego
973	<input type="checkbox"/>
Rzutowanie przekształca wierzchołek do układu współrzędnych przycinania, które są równocześnie unormowanymi współrzędnymi urządzenia	w przypadku rzutowania prostokątnego
	w przypadku rzutowań perspektywicznego i prostokątnego
	w przypadku, gdy wierzchołek jest podany we współrzędnych świata

## 974. W układzie odniesienia kamery ekran znajduje się macierz

	lewa i prawa krawędź są dowolnie zorientowane
974	<input type="checkbox"/>
W układzie odniesienia kamery ekran znajduje się w płaszczyźnie prostopadłej do osi Oz, a jego	lewa i prawa krawędź są równoległe do osi Oy
	<input type="checkbox"/>
	lewa i prawa krawędź są równoległe do osi Ox
	górną i dolną krawędź są równoległe do osi Ox

### 975. Zaprezentowana macierz jest

$$\begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

macierzą rzutowania perspektywicznego i przekształca współrzędne obserwatora w unormowane współrzędne urządzenia

macierzą rzutowania prostokątnego i przekształca współrzędne obserwatora w unormowane współrzędne urządzenia

Zaprezentowana macierz jest

macierzą rzutowania perspektywicznego i przekształca współrzędne obserwatora we współrzędne przycinania



macierzą rzutowania prostokątnego i przekształca współrzędne świata we współrzędne przycinania

23 / 27

### 976. Zaprezentowana macierz jest

$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

976

macierzą rzutowania perspektywicznego i przekształca współrzędne obserwatora w unormowane współrzędne urządzenia



Zaprezentowana macierz jest

macierzą rzutowania prostokątnego i przekształca współrzędne obserwatora w unormowane współrzędne urządzenia

macierzą rzutowania perspektywicznego i przekształca współrzędne obserwatora we współrzędne przycinania

macierzą rzutowania prostokątnego i przekształca współrzędne świata we współrzędne przycinania

977. Przedstawione równanie jest:

$$\begin{pmatrix} X_{NDC} \\ Y_{NDC} \\ Z_{NDC} \\ 1 \end{pmatrix} = \frac{1}{w_c} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ w_c \end{pmatrix}$$

977

Przedstawione równanie jest

 równaniem dzielenia perspektywicznego, gdzie indeks "c" określa współrzędne układu przycinania

równaniem dzielenia prostokątnego, gdzie indeks "c" określa współrzędne układu przytaczania

równaniem dzielenia perspektywicznego, gdzie indeks "c" określa współrzędne układu kamery



 równaniem dzielenia prostokątnego, gdzie indeks "c" określa współrzędne układu kamery

978. Biblioteka GLM

978

Biblioteka GLM(GL Math)

nie jest typu "header only" oraz nie jest obiektywnie zorientowana, ale obsługuje wszystkie niezbędne operacje matematyczne.

nie jest obiektywnie zorientowana, ale jest typu "header only" i obsługuje wszystkie niezbędne operacje matematyczne.

nie jest typu "header only", ale zawiera szereg klas wektorowych i macierzowych oraz obsługuje wszystkie niezbędne operacje matematyczne.

jest typu "header only" i zawiera szereg klas wektorowych i macierzowych oraz obsługuje wszystkie niezbędne operacje matematyczne.



979. Zaprezentowany fragment kodu

979

Zaprezentowany fragment kodu

```
glm::mat4 model;  
model = glm::rotate(model, glm::radians(1.57f),  
glm::vec3(0.0f, 0.0f, 1.0f));
```

 jest wykonywany przez szader wierzchołków i powoduje obrót dookoła osi Oz o kąt 1.57 radiana

 jest wykonywany przez szader wierzchołków i powoduje obrót dookoła osi Oz o kąt 1.57 stopnia



 jest wykonywany przez szader geometrii i powoduje obrót dookoła osi Oz o kąt 1.57 radiana

 jest wykonywany przez szader geometrii i powoduje obrót dookoła osi Oz o kąt 1.57 stopnia

980. Obrót o kąt prosty dookoła osi Oz zgodnie z ruchem wskazówek zegara dokonywany jest równaniem

- A: `model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 0.0f, 1.0f))`
- B: `model = glm::rotate(model, glm::radians(0.785f), glm::vec3(0.0f, 0.0f, 1.0f))`
- C: `model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 0.0f, -1.0f))`
- D: `model = glm::rotate(model, glm::radians(0.785f), glm::vec3(0.0f, 0.0f, -1.0f))`

980

Obrót o kąt prosty dookoła osi Oz zgodnie z ruchem wskazówek zegara dokonywany jest równaniem:

A

B



C

D

981. Zaprezentowana instrukcja powoduje

```
glm::mat4 view = glm::lookAt(  
    glm::vec3(1.2f, 1.2f, 1.2f),  
    glm::vec3(0.0f, 0.0f, 0.0f),  
    glm::vec3(0.0f, 1.0f, 0.0f));
```

981

Zaprezentowana instrukcja powoduje

utworzenie macierzy rzutowania perspektywicznego, a poszczególne wektory oznaczają kolejno: położenie kamery, punkt środkowy ekranu i oś "do góry"



utworzenie macierzy widoku, a poszczególne wektory oznaczają kolejno: położenie kamery, punkt środkowy ekranu i oś "do góry"

utworzenie macierzy rzutowania perspektywicznego, a poszczególne wektory oznaczają kolejno: punkt środkowy ekranu, położenie kamery i oś "do góry"

utworzenie macierzy widoku, a poszczególne wektory oznaczają kolejno: punkt środkowy ekranu, położenie kamery i oś "do góry"

982. Zaprezentowana instrukcja jest funkcją tworzącą macierz

```
proj = glm::perspective(glm::radians(alfa),  
    800.0f / 600.0f, 1.0f, 1.5f);
```

982

Zaprezentowana instrukcja jest funkcją tworzącą macierz rzutu perspektywicznego, a poszczególne jej parametry oznaczają

kąt pionowy pola widzenia, proporcje ekranu oraz bliską i daleką płaszczyznę obcinania



kąt poziomy pola widzenia, proporcje ekranu oraz współczynniki powiększenia w kierunku x i y

kąt pionowy pola widzenia, proporcje ekranu oraz współczynniki powiększenia w kierunku x i y

kąt poziomy pola widzenia, proporcje ekranu oraz bliską i daleką płaszczyznę obcinania

### 983. OpenGL dokonuje powiększenia (zoom) obrazu zazwyczaj z wykorzystaniem

983

OpenGL dokonuje powiększenia (zoom) obrazu zazwyczaj z wykorzystaniem

funkcji "glm::perspective" poprzez zmianę bliskiej i dalekiej płaszczyzny obcinania

funkcji "glm::perspective" poprzez zmianę kąta pola widzenia

funkcji "glm::lookAt" poprzez zmianę położenia kamery

### 984. Zaprezentowany kod jest

```
uniform mat4 model;
uniform mat4 view;
uniform mat4 proj;
void main(){
    Color = color;
    gl_Position = proj * view * model * vec4(position,
                                                0.0, 1.0);
}
```

984

fragmentem kodu szadera wierzchołków łączącym przekształcenia w przestrzeni 2D lub 3D

fragmentem kodu szadera geometrii łączącym przekształcenia w przestrzeni 2D lub 3D

Zaprezentowany kod jest

fragmentem kodu szadera wierzchołków łączącym przekształcenia, ale jedynie w przestrzeni 2D



fragmentem kodu szadera geometrii łączącym przekształcenia, ale jedynie w przestrzeni 2D

### 985. Tekstura może być obiektem

985

Tekstura może być obiektem...

tylko 2D



1D, 2D lub 3D

tylko 1D i 2D

tylko 3D

### 986. Wykonanie przedstawionej funkcji spowoduje

```
glGenTextures(1, &tex);
```

986

Wykonanie przedstawionej funkcji spowoduje...

ustawienie tekstury jako bieżącej

ustawienie tekstury jako głównej

wygenerowanie pojedynczej tekstury



ustawienie filtrowania tekstury

987. Wykonanie przedstawionej funkcji spowoduje | glBindTexture(GL\_TEXTURE\_2D, tex);

glBindTexture(GL\_TEXTURE\_2D, tex);

987

Wykonanie przedstawionej funkcji spowoduje...

połączenie dwóch tekstur

załadowanie tekstuury do pamięci



wygenerowanie pojedynczej tekstuury

przywiązanie tekstuury do obiektu typu GL\_TEXTURE\_2D

988. Co to jest próbkowanie (sampling)

988

Co to jest próbkowanie (sampling) tekstuury?

Testowanie wartości kolorów tekstuury

Operacja, która wykorzystuje współrzędne tekstuur do pobierania informacji o kolorze pikseli



Przydzielenie adresów teksturom

Mieszanie tekstuur

989. Podczas procesu nakładania tekstuur

989

Podczas procesu nakładania tekstuur (wrapping) parametr GL\_REPEAT oznacza, że...

część całkowita współrzędnej zostanie zignorowana i zostanie utworzony powtarzający się wzór

tekstura zostanie również powtórzona, ale obraz będzie obrazem zwierciadlanym, gdy część całkowita współrzędnej jest nieparzysta



współrzędna zostanie przekształcona do obszaru pomiędzy 0 a 1

współrzednym poza zakresem zostanie przypisany wybrany kolor obramowania

990. Podczas procesu nakładania tekstuur

990

Podczas procesu nakładania tekstuur jest używany parametr GL\_MIRROR\_OED\_REPEAT oznaczający, że...

część całkowita współrzędnej zostanie zignorowana i zostanie utworzony powtarzający się wzór

tekstura zostanie również powtórzona, ale obraz będzie obrazem zwierciadlanym, gdy część całkowita współrzędnej jest nieparzysta

współrzędna zostanie przekształcona do obszaru pomiędzy 0 a 1

współrzednym poza zakresem zostanie przypisany wybrany kolor obramowania

## 991. Podczas procesu nakładania tekstur (wrapping) parametr GL\_CLAMP\_TO\_EDGE

	część całkowita współrzędnej zostanie zignorowana i zostanie utworzony powtarzający się wzór
991	Podczas procesu nakładania tekstur (wrapping) parametr GL_CLAMP_TO_EDGE oznacza, że... tekstura zostanie również powtórzona, ale obraz będzie obrazem zwierciadlanym, gdy część całkowita współrzędnej jest nieparzysta współrzędna zostanie przekształcona do obszaru pomiędzy 0 a 1
	współrzędny poza zakresem zostanie przypisany wybrany kolor obramowania

## 992. Podczas procesu nakładania tekstur (wrapping) parameter GL\_CLAMP\_TO\_BORDER

	część całkowita współrzędnej zostanie zignorowana i zostanie utworzony powtarzający się wzór
992	Podczas procesu nakładania tekstur (wrapping) parametr GL_CLAMP_TO_BORDER oznacza, że... tekstura zostanie również powtórzona, ale obraz będzie obrazem zwierciadlanym, gdy część całkowita współrzędnej jest nieparzysta współrzędna zostanie przekształcona do obszaru pomiędzy 0 a 1 współrzędny poza zakresem zostanie przypisany wybrany kolor obramowania

## 993. Co to jest filtrowanie tekstur

	Zastosowanie różnych metod decydujących o próbkowanych kolorach tekstuury
993	Co to jest filtrowanie tekstur?
	Zmiana rozmiaru tekstuury
	Generowanie mipmap
	Nakładanie tekstuury

## 994. Wykonanie przedstawionej funkcji prowadzi do

	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,GL_LINEAR);
994	skalowania tekstuur w dół i filtrowania
	skalowania tekstuur w górę i filtrowania
	filtrowania bez skalowania
	skalowania brz. filtrowania

995. Wykonanie przedstawionej funkcji spowoduje

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, 2, 2, 0,  
             GL_RGB, GL_FLOAT, pixels);
```

995

Wykonanie przedstawionej funkcji spowoduje...

wygenerowanie obrazu tekstury

załadowanie obrazu tekstury



przefiltrowanie obrazu tekstury

sformatowanie obrazu tekstury

996. Co umożliwia dodanie następującej deklaracji

```
uniform sampler2D tex;
```

996

Co umożliwia dodanie następującej deklaracji do shadera fragmentów?

Dostęp shadera do tekstury



Załadowanie tekstury

997. Wykonanie przedstawionej funkcji spowoduje

```
glActiveTexture(GL_TEXTURE0);
```

przypisanie jednostki teksturującej



997

Wykonanie przedstawionej funkcji spowoduje...

aktywowanie tekstury

aktywowanie samplera

aktywowanie shadera

998. Wykonanie przedstawionej funkcji wymusza

```
glGenerateMipmap(GL_TEXTURE_2D);
```

generowanie wszystkich MIP-map

generowanie dwóch MIP-map

generowanie filtrów dla mipmap



generowanie nowej tekstury

998

Wykonanie przedstawionej funkcji wymusza ...

999. Co umożliwia dodanie następującej deklaracji do shadera fragmentów? In vec2 texcoord;

in vec2 texcoord;

Interpolowane współrzędnych koloru dla każdego fragmentu



Co umożliwia dodanie następującej deklaracji do shadera fragmentów?

Interpolowane współrzędnych tekstury dla każdego fragmentu

Interpolowane współrzędnych tekstury dla wybranych fragmentów

Interpolowane współrzędnych tekstury dla wybranych wierzchołków

1000. Do czego służy macierz modelu

Przekształca pozycję we współrzędnych świata na pozycję we współrzędnych modelu

1000

Do czego służy macierz modelu?

Przekształca pozycję we współrzędnych modelu na pozycję we współrzędnych świata



Przekształca widok modelu

Zmienia ustawienie kamery

1001. Do czego służy macierz widoku?

Przekształca wszystkie współrzędne świata na współrzędne widoku



1001

Do czego służy macierz widoku?

Przekształca wszystkie współrzędne widoku na współrzędne świata

Przekształca wszystkie współrzędne modelu na współrzędne świata

Przekształca wszystkie współrzędne świata na współrzędne modelu

1002. Macierz projekcji decyduje o

sposobie modelowania

sposobie rzutowania



sposobie kolorowania

sposobie cieniowania

1002

Macierz projekcji decyduje o...

### 1003. Do czego służy biblioteka GLM?

1003

Do czego służy biblioteka GLM?

Ułatwia pracę z modelowaniem

Ułatwia pracę z teksturami

Ułatwia pracę z macierzami



Ułatwia pracę z cielowaniem

### 1004. Co określa wykonanie przedstawionej linijki przez shader

1004

```
gl_Position = proj * view * model * vec4(position, 1.0);
```

Co określa wykonanie przedstawionej linijki przez shader?

Sposób generowania punktów

Sposób cieniowania

Kolor punktów

Położenie punktów

### 1005. Co określa wykonanie przedstawionej linijki kodu

1005

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

Co określa wykonanie przedstawionej linijki kodu?

Macierz Modelu

Macierz widoku

Macierz rzutowania

Macierz pochylenia

### 1006. Co określa wykonanie przedstawionej linijki kodu

1006

```
glm::mat4 proj = glm::perspective(glm::radians(45.0f),  
                                    800.0f / 800.0f, 0.06f, 100.0f);
```

Co określa wykonanie przedstawionej linijki kodu?

Macierz Modelu

Macierz widoku

Macierz projekcji

Macierz pochylenia



1007. Co spowodują instrukcje GLint uniTrans = glGetUniformLocation(shaderProgram, "model");  
glUniformMatrix4fv(uniTrans, 1, GL\_FALSE, glm::value\_ptr(model));

```
GLint uniTrans = glGetUniformLocation(shaderProgram,  
                                     "model");  
glUniformMatrix4fv(uniTrans, 1, GL_FALSE,  
                   glm::value_ptr(model));
```

1007	Dodanie macierzy modelu
	Wysłanie macierzy modelu do shadera
	Utworzenie macierzy modelu
	Zmianę macierzy modelu

1008. Co spowodują zaprezentowane instrukcje?

```
GLint uniView = glGetUniformLocation(shaderProgram,  
                                    "view");  
glUniformMatrix4fv(uniView, 1, GL_FALSE,  
                   glm::value_ptr(view));
```

1008	Dodanie macierzy widoku
	Wysłanie macierzy widoku do shadera
	Utworzenie macierzy widoku
	Zmianę macierzy widoku

1009. Co spowodują przedstawione instrukcje?

```
GLint uniProj = glGetUniformLocation(shaderProgram,  
                                    "proj");  
glUniformMatrix4fv(uniProj, 1, GL_FALSE,  
                   glm::value_ptr(proj));
```

1009	Dodanie macierzy projekcji
	Wysłanie macierzy projekcji do shadera
	Utworzenie macierzy projekcji
	Zmianę macierzy projekcji

### 1010. Jakiego typu używa GLSL do przechowywania

1010

Jakiego typu używa GLSL do przechowywania macierzy przekształceń?

mat2



mat3

mat4

mat5

### 1011. Jakiego typu używa GLSL do przechowywania wektora koloru?

1011

Jakiego typu używa GLSL do przechowywania wektora koloru?

26 / 27

Brak kodu

vec2

vec3

vec4



vec5

27 / 27

### 1012. Czy zmienne typu uniform są dostępne na każdym etapie potoku: TAK

1012

Czy zmienne typu uniform są dostępne na każdym etapie potoku

Tak

Nie

Tylko w części

### 1013. Co wykonuje funkcja glm:normalize(..)

1013

Co wykonuje funkcja glm:normalize(..)?

Zmienia długość wektora na 1

Zmienia zwrot wektora

Zmienia wartość wektora

Mnoży wektory

### 10\*\*. Co oblicza przedstawiona funkcja? Glm :: cross

glm::cross(cameraFront, cameraUp);

iloczyn wektorowy wektorów

iloczyn skalarny wektorów

Normę wektorów

Przełoczenie wektorów



Co oblicza przedstawiona funkcja?

- **DOT3 mapping wymaga nałożenia na obiekt** podstawowej tekstury i mapy wypukłości.
- **Obraz wygenerowany komputerowo w porównaniu do fotografii** może być realistyczny, w stopniu nie pozwalającym odróżnić go od rzeczywistości
- **Kamera syntetyczna to** program komputerowy, który symuluje kamerę rzeczywistą, obiekty sceny w układzie niezależnym
- **Akceleratory grafiki trójwymiarowej są to** urządzenia odciażające jednostkę centralną w końcowym etapie obliczeń.
- **Wrażenie przestrzenności sceny graficznej uzyskiwane jest głównie z wykorzystaniem algorytmów oświetlania i cieniowania łącznie.**
- **Jasność obiektu zmienia się w zależności od kierunku i odległości od źródła światła przy oświetleniu punktowym**
- **Metoda energetyczna pozwala na** zabarwienie powierzchni rozpraszających barwami innych powierzchni.
- **Do tworzenia mgły liniowej wykorzystywane są** funkcje liniowe zależne od odległości.
- **Algorytm Z-bufora to algorytm** ukrywania powierzchni niewidzialnych.

### **1. Modelowanie powierzchni w grafice trójwymiarowej polega na:.**

- Dokładnym opisie powierzchni funkcjami matematycznymi
- Zastosowaniu wyłącznie płytów bikubicznych
- **Wykorzystaniu szeregu metod - najczęściej siatek wielokątów, powierzchni parametrycznych i powierzchni drugiego stopnia**
- Wykorzystaniu szeregu metod - najczęściej siatek wielokątów, powierzchni stożkowych i powierzchni popularnych brył

### **2. Modelowanie obiektów nierzeczywistych polega na tym, iż:.**

- Obiekt aproksymuje swoją reprezentację i stanowi jej urzeczywistnienie
- **Obiekt przybliża dokładnie swoją reprezentację, gdyż stanowi jedynie Jej urzeczywistnienie**
- Obiekt jest zawsze zbudowany z szeregu brył geometrycznych stanowiących jego urzeczywistnienie
- Modelowanie takie jest niemożliwe

### **3. Wielomianowe krzywe parametryczne definiują punkty na krzywych za pomocą:.**

- Uwikłanych funkcji wielomianowych
- Trzech wielomianów parametru t, wspólnych dla każdej ze współrzędnej
- Wielomianów określających wzajemne zależności współrzędnych
- **Trzech wielomianów parametru t, oddzielnie dla każdej współrzędnej**

#### **4. Model powierzchni znanego z OpenGL czajniczka Martina**

- Z uśrednienia kolorów występujących w każdym z jego wierzchołków
- przy stałym natężeniu światła
- **Poprzez obliczenie kolorów i natężenia światła osobno dla każdego piksela trójkąta**
- Z uśrednienia kolorów i natężenia światła występujących w każdym z sąsiednich trójkątów

#### **5. Jeżeli zbiór punktów S jest skończony, to jego otoczka:**

- **Jest najmniejszym wielokątem wypukłym o wierzchołkach ze zbioru S.**
- Jest dowolnym wypukłym lub wklęsłym wielokątem o wierzchołkach ze zbioru S.
- Jest dowolnym wielokątem wypukłym o wierzchołkach ze zbioru
- Jest najmniejszym wypukłym lub wklęsłym wielokątem o wierzchołkach ze zbioru S.

#### **6. Algorytm Grahama wyznaczania otoczki wypukłej zbioru S jest oparty na założeniu, iż:**

- Wszystkie punkty otoczki wypukłej leżą po jednej stronie prostych wyznaczonych przez kolejne punkty otoczki.
- **Żaden wierzchołek otoczki wypukłej nie leży ani wewnętrz ani na brzegu trójkąta wyznaczonego przez trzy inne punkty zbioru S.**
- Każdy wierzchołek otoczki wypukłej leży na brzegu trójkąta wyznaczonego przez trzy inne punkty zbioru S.
- Każdy wierzchołek otoczki wypukłej leży wewnętrz lub na brzegu trójkąta wyznaczonego przez trzy inne punkty ze zbioru S.

#### **7. Algorytm z punktem środkowym w przypadku konwersji okręgu rozważa:**

- Jedną ósmą okręgu począwszy od punktu położonego na osi OX.
- **Jedna ósma okręgu począwszy od punktu położonego na osi OY.**
- Jedną czwartą okręgu począwszy od punktu położonego na osi OX.
- Jedną czwartą okręgu począwszy od punktu położonego na osi OY.

#### **8. Algorytm wypełniania wielokątów wykorzystuje:**

- Tablicę aktywnych krawędzi oraz zależną od niej globalną tablicę krawędzi.
- Rezultaty iloczynów skalarnych wektorów poprowadzonych z wierzchołków wielokąta.
- Jedynie globalną tablicę krawędzi.
- **Globalną tablicę krawędzi oraz zależną od niej tablicę aktywnych krawędzi.**

#### **9. Rozwiązywanie układu równań jest cechą następujących algorytmów obcinania odcinków:**

- Prymitywnego i Cyrusa-Becka.
- Prymitywnego i Cohena-Sutherlanda.
- Cohena-Sutherlanda i Cyrusa-Becka.
- **Jedynie prymitywnego.**

#### **10. W przypadku obcinania odcinków płaskiej sceny graficznej algorytm Cohena-Sutherlanda:**

- Stosuje podział przestrzeni na 9 niezależnych pól oraz wykonuje jedynie potrzebne obcinania odcinków.
- **Stosuje podział przestrzeni na 9 niezależnych pól oraz wykonuje nie zawsze potrzebne obcinania odcinków.**
- Stosuje podział przestrzeni na 4 niezależnych pola oraz wykonuje jedynie potrzebne obcinania odcinków.
- Stosuje podział przestrzeni na 9 niezależnych pól oraz nie wykonuje żadnego obcinania odcinków.

**11. Oprócz typów oferowanych przez język C, GLSL posiada wbudowane:**

- Typy wektorowe i macierzowe dowolnych rozmiarów.
- Typy wektorowe identyfikowane przez vec4, ale nie typy macierzowe.
- Typy wektorowe i macierzowe identyfikowane przez vec i mat.
- **Typy wektorowe i macierzowe identyfikowane przez vec\* i mat\*.**

**12. Najlepszą aproksymację dla idealnego odcinka linii prostej:**

- zapewnia użycie arytmetyki zmiennoprzecinkowej
- **zapewniają: podstawowy algorytm przyrostowy i algorytm z punktem środkowym 70%**
- zapewnia podstawowy algorytm przyrostowy
- **zapewnia algorytm z punktem środkowym - to na pewno, ale czy przyrostowy też to nwm**

**13. Piksel nie jest traktowany jako część wypełnianego wielokąta, jeśli półpłaszczyzna zdefiniowana przez krawędź i zawierająca prymityw:**

- **leży poniżej albo z lewej strony krawędzi**
- leży powyżej albo z prawej strony krawędzi
- leży powyżej albo z lewej strony krawędzi
- leży poniżej albo z prawej strony krawędzi

**14. Co definiuje funkcja void glViewport:**

- **Obszar renderingu**

**15. Dla prymitywu GL\_LINES:**

- **Każde dwa punkty definiują odcinek.**

**16. Wartość różnic II rzędu dla algorytmu z punktem środkowym w przypadku konwersji okręgu wynosi:**

- 2 lub 4, przy czym częściej 4
- **2 lub 4, przy czym częściej 2**
- $2 x_p + 3$
- $2 x_p - 2 y_p + 3$

**17. Translacja i rotacja są przekształceniami**

- **Addytywnymi**
- Translacja jest multiplikatywna, a rotacja addytywna
- Multiplikatywnymi
- Translacja jest addytywna, a rotacja multiplikatywna

**18. Algorytm konwersji z punktem środkowym może być stosowany dla**

- dowolnych odcinków i okręgów
- **dowolnych odcinków, ale okręgów całkowitoliczbowych**
- jedynie odcinków i okręgów całkowitoliczbowych
- dowolnych okręgów, ale odcinków całkowitoliczbowych

## **19. Metoda powielania pikseli, to metoda**

- rasteryzacji okręgów całkowitoliczbowych
- **poogrubiania prymitywów**
- wypełniania wielokątów
- rasteryzacji odcinka

## **20. Podstawowy algorytm przyrostowy konwersji odcinka jest nieefektywny gdyż w każdej iteracji:**

- **wykonywane jest zmiennopozycyjne mnożenie, dodawanie i zaokrąglanie wyniku**
- wykonywane jest zaokrąglanie wyniku
- wykonywanych jest szereg zmiennopozycyjnych mnożeń
- wykonywane jest zmiennopozycyjne mnożenie macierzy

## **21. Wartość zmiennej decyzyjnej algorytmu z punktem środkowym jest obliczana dla**

- punktu leżącego w równej odległości między dwoma pikselami
- położen dwóch sąsiednich pikseli
- **punktu przecięcia odcinka lub okręgu z linią wyznaczającą kolumnę pikseli**
- punktu przecięcia odcinka lub okręgu z linią wyznaczającą wiersz pikseli

## **22. Prymitywny algorytm obcinania odcinków**

- Dokonuje obliczeń przecięć każdego z odcinków sceny graficznej z wybraną krawędzią prostokąta obcinającego
- Nie dokonuje obliczeń przecięć odcinków sceny graficznej z krawędziami prostokąta obcinającego
- Dokonuje obliczeń przecięć każdego z odcinków sceny graficznej z każdą krawędzią prostokąta obcinającego
- **Dokonuje obliczeń przecięć niektórych odcinków sceny graficznej z każdą krawędzią prostokąta obcinającego**

## **23. Algorytm Cohena-Suterlanda w jednej iteracji**

- Dzieli każdy odcinek na część wewnętrzną i zewnętrzną w stosunku do prostokąta obcinającego
- Dzieli odcinek jedynie wtedy gdy oba jego końce leżą na zewnątrz prostokąta obcinającego
- Dzieli odcinek jedynie wtedy gdy jeden jego koniec leży wewnętrz, a drugi na zewnątrz prostokąta obcinającego
- **Dzieli niektóre odcinki na część wewnętrzną i zewnętrzną w stosunku do prostokąta obcinającego**

## **24. Algorytm Cohena-Suterlanda używa kodów**

- **czterobitowych**
- jednobitowych
- dwubitowych
- jednobajtowych

## **25. Algorytm Cyrusa-Becka obcinania odcinków sceny graficznej**

- Jest algorytmem prymitywnym
- **Jest algorytmem parametrycznym**
- Nie należy do grupy algorytmów parametrycznych ani iteracyjnych
- Jest algorytmem iteracyjnego dzielenia odcinków

**26. Algorytm Cohena-Sutherlanda jest najmniej efektywny jeśli obszar prostokąta obcinającego jest**

- Zbliżony do połowy obszaru zajmowanego przez prymitywy
- Porównywalny z obszarem zajmowanym przez prymitywy
- **Znacznie mniejszy od obszaru zajmowanego przez prymitywy**
- Większy od obszaru zajmowanego przez prymitywy

**27. Podczas pracy algorytm Cyrusa-Becka obcinania odcinków brane są pod uwagę:**

- Wewnętrzne wektory normalne do linii wyznaczających boki prostokąta obcinającego
- Wektory normalne do odcinków-prymitywów
- Zewnętrzne i wewnętrzne wektory normalne do linii wyznaczających boki prostokąta obcinającego
- **Zewnętrzne wektory normalne do linii wyznaczających boki prostokąta obcinającego**

**28. Algorytm Cyrusa-Becka obcinania odcinków sceny graficznej opiera się o**

- Wyznaczenie i posortowanie punktów przecięcia odcinka z brzegiem prostokąta obcinającego oraz nadanie im 4-bitowych kodów
- Wyznaczenie, posortowanie i sklasyfikowanie punktów przecięcia odcinka z brzegiem prostokąta obcinającego
- **Wyznaczenie, posortowanie i sklasyfikowanie parametrów punktów przecięcia odcinka z brzegiem prostokąta obcinającego**
- Wyznaczenie i posortowanie punktów przecięcia odcinka z brzegiem prostokąta obcinającego

**29. OpenGL 2.1: Jakie dwie podstawowe ścieżki przetwarzania można wyróżnić w OpenGL?**

- **Geometrii i obrazu**
- Geometrii i rasteryzacji
- Przetwarzania wstępnego właściwego
- Geometrii i renderowania

**30. OpenGL 2.1: Maszyna stanu przechowuje dane o zakresie**

- chronionym
- lokalnym
- **globalnym**
- prywatnym

**31. OpenGL 2.1: Co to jest rasteryzacja (rasterization)?**

- zapis współrzędnych do tablicy
- ukrywanie niewidocznych powierzchni
- wczytywania tekstur
- **konwersja danych pochodzących ze ścieżki geometrii i pikseli w tzw. fragmenty**

### **32. OpenGL 2.1: Dla prymitywu GL\_LINE\_STRIP:**

- każde dwa wierzchołki definiują odcinek.
- wierzchołki definiują linię łamana gdzie każdy następny punkt jest łączony z poprzednim, a ostatni punkt jest łączony z pierwszym.
- **wierzchołki definiują linię łamana gdzie każdy następny punkt jest łączony z poprzednim.**
- każdy zdefiniowany wierzchołek określa pojedynczy punkt.

### **33. OpenGL 2.1: Rozwojem OpenGL zajmuje się organizacja**

- **Architecture Review Board**

### **34. Algorytm konwersji z punktem środkowym poszukuje w jednej iteracji**

- dwóch pikseli
- **jednego piksela**
- 1 piksela w przypadku odnicków i 8 w przypadku okręgów
- 1 piksela w przypadku odnicków i 4 w przypadku okręgów

### **35. Przestrzenna spójność segmentów jest wykorzystywana w przypadku**

- **wypełniania wielokątów**
- rasteryzacji odcinków i okręgów
- obcinania prymitywów
- pogrubiania odcinków

### **36. Algorytm Cohena -Sutherlanda odrzuca odcinki**

- **Jeśli iloczyn bitowy obu kodów nie jest równy zeru**
- Jeśli iloczyn bitowy obu kodów jest równy zeru
- Jeśli kody obu końców są zerowe
- Jeśli kody obu końców są identyczne

### **37. Algorytm Cyrusa-Becka obcinania odcinków sceny graficznej klasyfikuje przecięcia jako wchodzące (PE) i wychodzące (PL) na podstawie**

- Znaku różnicy współrzędnych normalnej do obcinającej krawędzi
- **iloczynu skalarnego normalnej do krawędzi obszaru obcinającego i wektora łączącego dowolny punkt odcinka z wybranym punktem krawędzi**
- Iloczynu wektorowego normalnej do krawędzi obszaru obcinającego i wektora łączącego dowolny punkt odcinka z wybranym punktem krawędzi
- Znaku sumy współrzędnych normalnej do obcinającej krawędzi

### **38. Algorytm konwersji Wu i Rokne poszukuje w jednej iteracji**

- **Dwóch pikseli**
- Trzech pikseli
- Jednego piksela
- Czterech pikseli

**39. Zmienna decyzyjna algorytmu z punktem środkowym może przyjmować wartości:**

- rzeczywiste dodatnie i ujemne
- rzeczywiste dodatnie
- **całkowite dodatnie i ujemne**
- całkowite dodatnie

**40. Przyrost zmiennej decyzyjnej algorytmu z punktem środkowym jest w przypadku konwersji konkretnego odcinka”**

- ciągle zmieniającą się liczbą całkowitą
- zawsze stałą liczbą niecałkowitą
- zawsze liczbą stałą, ale niekoniecznie całkowitą
- **zawsze jedną z dwóch stałych liczb całkowitych**

**41. Wartość funkcji uwikłanej opisującej równanie odcinka dla algorytmu z punktem środkowym jest:**

- zawsze ułamkiem o mianowniku 4
- zawsze liczbą ułamkową o mianowniku 2
- zawsze liczbą całkowitą
- **liczbą całkowitą bądź ułamkiem o mianowniku 2**

**42. Algorytm z punktem środkowym działa poprawnie dla:**

- dowolnych odcinków o nachyleniu od -1 do 1 i całkowitoliczbowych odcinków
- całkowitoliczbowych odcinków i dowolnych okręgów
- dowolnych odcinków i całkowitoliczbowych okręgów
- **dowolnych odcinków o nachyleniu od 0 do 1 i całkowitoliczbowych okręgów**

**43. Podstawowy algorytm przyrostowy konwersji odcinka został opracowany dla:**

- odcinków o dowolnym nachyleniu, a obliczenia dokonywane w dziedzinie liczb całkowitych
- **odcinków o dowolnym nachyleniu, a obliczenia są dokonywane w dziedzinie liczb rzeczywistych**
- odcinków o nachyleniu od -1 do 1, a obliczenia są dokonywane w dziedzinie liczb rzeczywistych
- odcinków o nachyleniu od -1 do 1, a obliczenia są dokonywane w dziedzinie liczb całkowitych

**44. Translacja i rotacja względem wybranego punktu obiektu są przekształceniami:**

- **Złożonymi**