



Bewertung:

Programmierung 8 Pt	Modellierung 18 Pt	Rasterisierung 14 Pt	OpenGL 17 Pt	Beleuchtung 9 Pt	Raytracing 9 Pt	Summe 75 Pt

Klausur ECG SS 2010

Info

- In dieser Klausur können Sie insgesamt 75 Punkte erreichen. Für eine erfolgreiche Klausur sind mindestens 37 Punkte notwendig. Dafür haben Sie 90min Zeit.
- Bitte schalten Sie alle elektronischen Geräte aus und verstauen diese in Ihren Taschen.
- Erlaubte Hilfsmittel sind Stifte, Radiergummi, Lineal, Geodreieck und für ausländische Studenten ein Wörterbuch.
- Eigene Blätter sind nicht erlaubt. Zusätzliche leere Blätter sind von den Aufsichtspersonen mit einem Handzeichen zu erfragen.
- Der vorgegebene Platz im Klausurbogen reicht stets für eine vollständige Lösung aus. Nebenrechnungen können Sie auf der Rückseite des Deckblattes oder des letzten Blattes anfertigen.
- Zum Teil sind die Lösungen in vorhandene Skizzen einzuzeichnen. Falls Sie eine Skizze unwiderruflich falsch ausgefüllt haben, können Sie vom Aufsichtspersonal eine weitere Kopie der Skizze anfordern solange der Vorrat reicht.
- Bitte schreiben Sie Ihre Antworten auf die vorgegebenen Linien. Sollten Sie dennoch Platz für Ihre Lösungen auf weiteren leeren Blättern benötigen, beschriften Sie jedes Blatt unbedingt mit **Name, Matrikel- und Aufgabennummer**.
- Wenn Sie zu einer Aufgabe nicht sofort die Lösung bzw. den Lösungsweg kennen, lösen Sie zuerst die restlichen Aufgaben.
- Auf der Rückseite des Deckblatts findet sich ein Auszug aus der Tabelle für C++-Operatoren, die in der ersten Aufgabe benötigt werden.

Checkliste

- € Name und Matrikelnummer auf dem Deckblatt eingetragen?
- € Alle Aufgaben bearbeitet?
- € Nicht gültige Lösungen durchgestrichen?



Regeln für C++-Operatoren

Precedence	Operator	Description	Example	Overloadable	Associativity
1	::	Scope resolution operator	Class::age = 2;	no	left to right
3	compl	Alternate spelling for ~	flags = ~flags;	yes	right to left
	++	Pre-increment	for (i = 0; i < 10; ++i) cout << i;	yes	
	--	Pre-decrement	for (i = 10; i > 0; --i) cout << i;	yes	
	-	Unary minus	int i = -1;	yes	
	+	Unary plus	int i = +1;	yes	
	*	Dereference	int data = *IntPtr;	yes	
	&	Address of	int *IntPtr = &data;	yes	
	sizeof	Size (of the type) of the operand in bytes	size_t s = sizeof(int);	no	
	new	Dynamic memory allocation	long* pVar = new long;	yes	
	new []	Dynamic memory allocation of array	long* array = new long[20];	yes	
	delete	Deallocating the memory	delete pVar;	yes	
	delete []	Deallocating the memory of array	delete [] array;	yes	
4	(type)	Cast to a given type	int i = (int)floatNum;	yes	left to right
	->*	Member pointer selector	ptr->*var = 24;	yes	
5	.*	Member object selector	obj.*var = 24;	no	left to right
	*	Multiplication	int i = 2 * 4;	yes	
6	/	Division	float f = 10.0 / 3.0;	yes	left to right
	%	Modulus	int rem = 4 % 3;	yes	
7	+	Addition	int i = 2 + 3;	yes	left to right
	-	Subtraction	int i = 5 - 1;	yes	
8	<<	Bitwise shift left	int flags = 33 << 1;	yes	left to right
	>>	Bitwise shift right	int flags = 33 >> 1;	yes	
9	<	Comparison less-than	if (i < 42) ...	yes	left to right
	<=	Comparison less-than-or-equal-to	if (i <= 42) ...	yes	
	>	Comparison greater-than	if (i > 42) ...	yes	
	>=	Comparison greater-than-or-equal-to	if (i >= 42) ...	yes	
10	==	Comparison equal-to	if (i == 42) ...	yes	left to right
	eq	Alternate spelling for ==			
	!=	Comparison not-equal-to	if (i != 42) ...	yes	
	not_eq	Alternate spelling for !=			
11	&	Bitwise AND	flags = flags & 42;	yes	left to right
	bitand	Alternate spelling for &			
12	^	Bitwise exclusive OR (XOR)	flags = flags ^ 42;	yes	left to right
	xor	Alternate spelling for ^			
13		Bitwise inclusive (normal) OR	flags = flags 42;	yes	left to right
	bitor	Alternate spelling for			



I Programmierung (8 Pt)

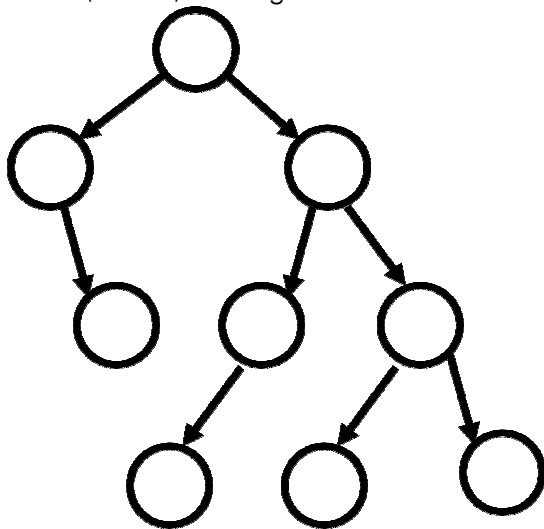
Frage I.1 (2 Pt)

Gegeben sei der Ausdruck: $e = (a+b) * (b-c)^{a \& b}$;

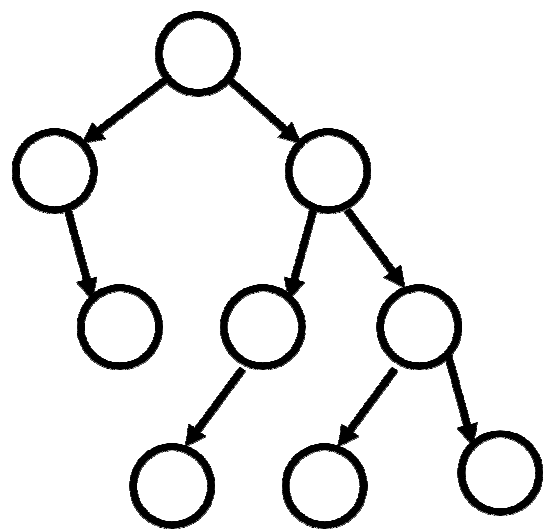
Geben Sie alle Auswertungsreihenfolgen für den Ausdruck an, die die Assoziations- und Prioritätsregeln von C++-Operatoren erfüllen, indem Sie für jede mögliche Reihenfolge die Liste der Operatoren ohne weitere Trennzeichen angeben:

Frage I.2 (2 Pt)

Nummerieren Sie die Knoten in den dargestellten Bäumen gemäß der Traversierungsreihenfolge (beginnend mit 1) bei der „Depth First“ (links) bzw. „Breadth First“ (rechts) Strategie!



Depth First



Breadth First

Frage I.3 (4 Pt)

Geben Sie Pseudo-Code an für eine Traversierungsfunktion die einen Baum in der Depth First Strategy durchwandert:



II Modellierung mit Kurven (18 Pt)

Frage II.1 (1 Pt)

Skizzieren Sie einen Fall, bei dem es beim Marching Squares Algorithmus mehrere Möglichkeiten gibt, Kurvensegmente zu erzeugen!

Frage II.2 (2 Pt)

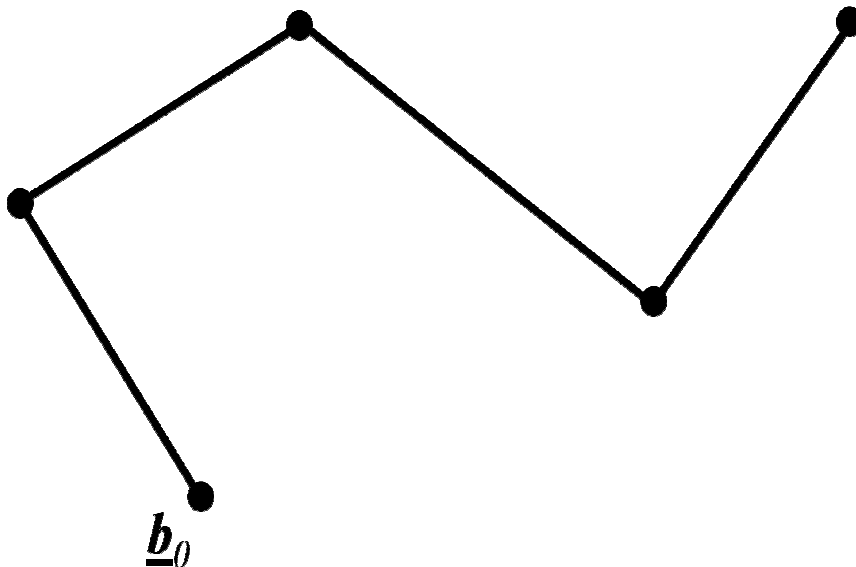
Wie kann man die Vereinigung bzw. Schnittmenge von zwei implizit repräsentierten Formen F und G berechnen, wenn negative Werte das Innere beschreiben und positive das Äußere und die Formen durch die Funktionen $f(x,y)=0$ und $g(x,y)=0$ definiert sind?

$F \cup G$: _____

$F \cap G$: _____

Frage II.3 Beziér-Kurven (3+1+2=6 Pt)

- a) Konstruieren Sie für folgendes Kontrollpolygon den Kurvenpunkt einer Bezier-Kurve graphisch für den Parameterwert $t = 0,3$! (3 Pt)



- b) Welchen Grad hat die Bezier-Kurve, die durch obiges Kontrollpolygon definiert ist? (1 Pt)



- c) Geben Sie die Formeln für die Bernsteinbasis vom Grad 3 an! Schreiben Sie sich dazu zuerst den benötigten Teil des Pascal'schen Dreiecks auf? **(2 Pt)**

Frage II.4 Lagrange-Interpolation (2 Pt)

Wie ist die 3-te Lagrange-Basisfunktion vom Grad 4 bei einem Stützstellenvektor von $U = (0, 1, 2, 4, 7)$ definiert?

Frage II.5 B-Splines (2+2+1+2=7 Pt)

- a) In welchem Zusammenhang steht die Anzahl K der Kontrollpunkte mit der Anzahl n der Kurvensegmente bei einem offenen bzw. geschlossenen B-Spline vom Grad g ? **(2 Pt)**

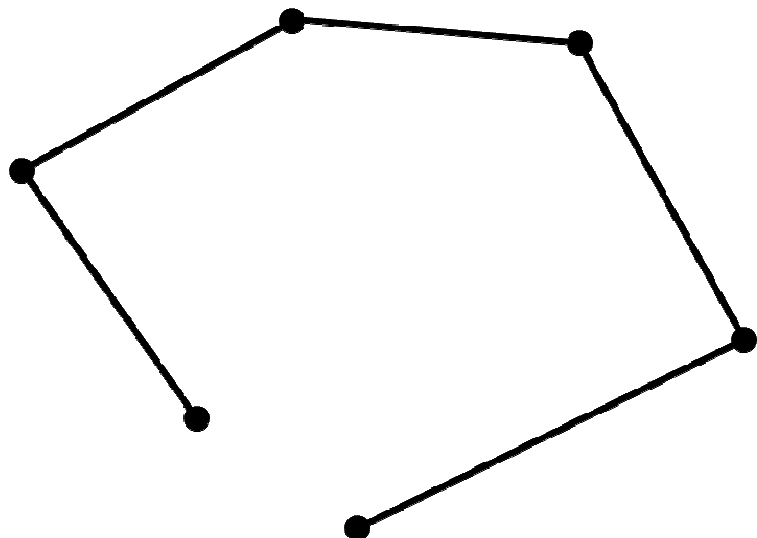
offen: _____

geschlossen: _____

- b) Geben Sie einen Knotenvektor für einen offenen B-Spline vom Grad 4 mit 8 Kontrollpunkten an, der die Eigenschaft der Endpunktinterpolation besitzt! **(2 Pt)**
[Tipp überlegen Sie sich zuerst die Anzahl der Kurvensegmente und schließen Sie daraus auf die Anzahl der unterschiedlichen Knotenvektoreinträge.]

- c) Wieviele Kontrollpunkte beeinflussen jeweils ein Kurvensegment in einem geschlossenen B-Spline vom Grad 3? **(1 Pt)**

- d) Zeichnen Sie im rechten Kontrollpolygon die Vereinigung der lokalen konvexen Hüllen ein innerhalb der ein offener B-Spline vom Grad 2 liegen muss! **(2 Pt)**





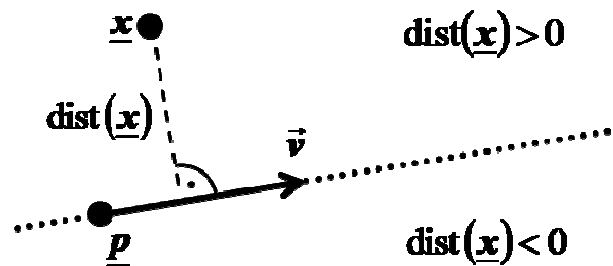
III Rasterisierung (14 Pt)

Frage III.1 (4 Pt)

Nennen Sie zwei Vor- und zwei Nachteile von Vektordisplays!

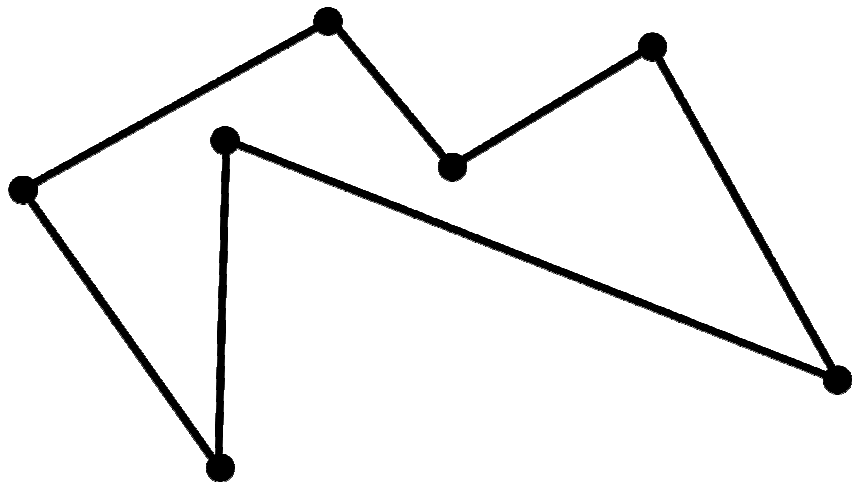
Frage III.2 (4 Pt)

Wie kann man den senkrechten, mit Vorzeichen behafteten Abstand $\text{dist}(\underline{x})$ eines gegebenen 2D Punktes \underline{x} von einer Geraden durch den Punkt \underline{p} in Richtung $\vec{v} = (v_x \ v_y)^T$ berechnen?



Frage III.3 (1 Pt)

Kreisen Sie im rechten Bild die Eckpunkte des Polygons ein, deren Polygonecken ein Ohr im Sinne des Ear-Cutting-Algorithmus darstellen.



Frage III.4 (1 Pt)

Wie viele Ohren haben Polygone mit mehr als 6 Ecken mindestens?



Frage III.5 (4 Pt)

Geben Sie Pseudo-Code für einen FloodFill Algorithmus, der auch bei sehr großen Füllbereichen nicht zum Stapelüberlauf führt! (Es gibt auch Punkte für Pseudo-Code eines rekursiven Floodfill Algorithmus, der zu Stapelüberläufen führt)

IV OpenGL und Transformationen (17 Pt)

Frage IV.1 (2 Pt)

Erklären Sie das Prinzip des Triple Buffering!



Frage IV.2 (3 Pt)

Erklären Sie den Unterschied zwischen Flat-, Gouraud- und Phong-Shading!

Frage IV.3 (3 Pt)

Schreiben Sie unter die folgenden **homogenen** 3x3-Matrizen, welche Art der Transformation sie repräsentieren. Zur Auswahl stehen:

- reine Rotation (rR),
- reine nichtuniforme Skalierung (rnuS),
- reine uniforme Skalierung (ruS),
- reine Punktspiegelung (rPS),
- reine Translation (rT),
- reine Scherung (rS),
- reine perspektivische Transformation (rP),
- keine reine Transformation (krT).

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

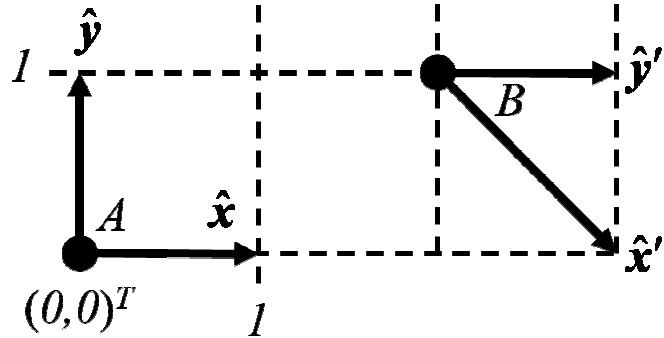
Frage IV.4 (1 Pt)

Wann werden Geradensegmente bei einer perspektivischen Transformation in zwei Teile zerlegt?



Frage IV.5 (3 Pt)

Geben Sie die **homogene** Matrix der affinen Transformation an, die Vektoren im Koordinatensystem A so transformiert, dass sich die Basisvektoren von A auf die Basisvektoren von B abbilden. Wie kommt man auf die Systemtransformation von A nach B ?



Frage IV.6 (2 Pt)

Wie kann man homogene Vektoren interpretieren, wenn man sie zur Darstellung von affinen Transformationen einsetzt und wie beim Einsatz für perspektivische Transformationen?

Frage IV.7 (3 Pt)

Transformieren Sie den 2D Punkt $\underline{p} = (2 \ 1)^T$ in den in der Vorlesung behandelten drei Schritten mit der perspektivischen Transformation, die durch die homogene Matrix $\tilde{M} = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \\ \frac{1}{2} & 0 & 1 \end{pmatrix}$ repräsentiert ist. Benennen sie die drei Teilschritte!



V Beleuchtung (9 Pt)

Frage V.1 (2 Pt)

Erklären Sie den Unterschied zwischen Light und Visibility Tracing anhand einer Skizze!

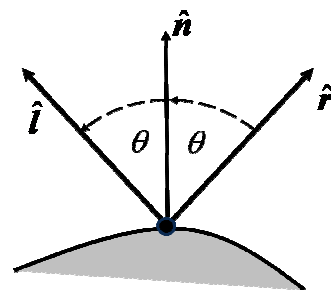
Frage V.2 (2 Pt)

Kreuzen Sie in der rechten Tabelle an, wovon die jeweiligen Beleuchtungsanteile eines lokalen Beleuchtungsmodells abhängig sind:

	Richtung zur Lichtquelle	Richtung zum Beobachter
ambient	<input type="checkbox"/>	<input type="checkbox"/>
diffus	<input type="checkbox"/>	<input type="checkbox"/>
spekular	<input type="checkbox"/>	<input type="checkbox"/>

Frage V.3 (2 Pt)

Geben Sie in Vektornotation an wie sich \hat{r} aus \hat{l} und \hat{n} berechnet! [Alle drei Vektoren sollen normiert sein.]





Frage V.4 (3 Pt)

Zeichnen Sie eine Skizze und erklären Sie das Gesetz von Snellius mit der entsprechenden Formel!

VI Raytracing (9 Pt)

Frage VI.1 (1 Pt)

Wieso werden beim Raytracing meist zwei unterschiedliche Schnittberechnungsfunktionen implementiert?

Frage VI.2 (3 Pt)

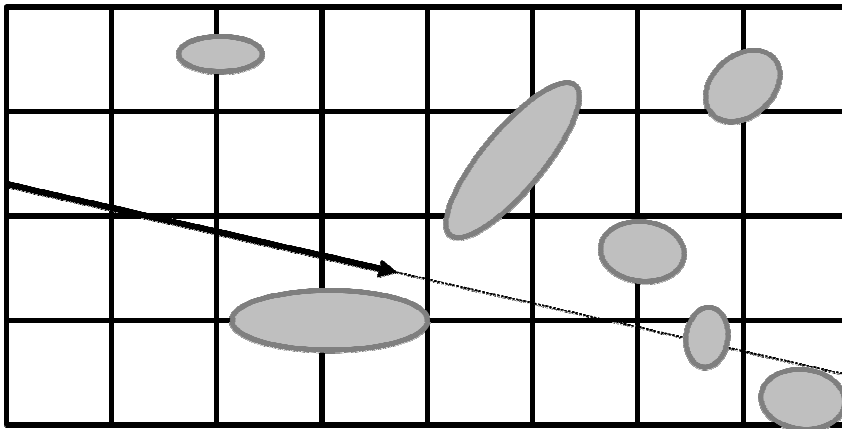
Erklären Sie ohne auf Formeln einzugehen, wie man die Schnittberechnung zwischen Strahl und Kugel implementiert!



Frage VI 3 (2 Pt)

Markieren Sie in folgender Skizze, in welcher Reihenfolge die Gitterzellen durchlaufen werden und in welcher Reihenfolge der jeweils erste Schnitttest mit jedem Primitiv durchgeführt wird (maximal eine Zahl pro Zelle und eine Zahl pro Primitiv)!

[Vorsicht: Primitive, für die keine Schnittberechnung durchgeführt wird, dürfen auch keine Zahl erhalten]



Frage VI 4 (3 Pt)

Zeichnen Sie in folgendem Bild die ersten drei Levels einer Hüllvolumenhierarchie ein, die nach dem Top-Down Prinzip aufgebaut wird und mit achsenparallelen (parallel zu den Gitterlinien im Hintergrund ausgerichtet) Rechtecken arbeitet. Spalten Sie jeweils entlang der Richtung der größten Rechtecksausdehnung. Falls eine ungerade Anzahl an Primitiven zu splitten ist, splitten Sie so, dass unten bzw. links mehr Primitive zugewiesen werden.

