

Beantworten Sie die Fragen in Aufgabe 1 mit einer kurzen, prägnanten Antwort. Die Fragen zu OpenGL beziehen sich auf die in der Vorlesung vorgestellte OpenGL Version 3.1 (core) sowie GLSL 1.4.

Aufgabe 1 (15 Punkte)

1. Welcher Punkt im \mathbb{R}^3 wird durch die homogenen Koordinaten $(9, 6, 12, 3)$ beschrieben?

$(3, 2, 4)$

2. In welchem Farbraum wird ein Bild bei Anwendung der JPEG-Kompression konvertiert?

YUV

3. Mit welchem Verfahren lassen sich auf Schwarz-Weiß-Bildschirmen Grauwerte nachbilden?

Dithering

4. Wo liegt eine Kante bezüglich des Clipping-Rechtecks beim Clipping nach Cohen und Sutherland, deren Region-Codes der Endpunkte ODER-verknüpft 0 ergibt ($code(P_1) || code(P_2) = 0$)?

komplett innerhalb des Clipping-Rechtecks

5. Nennen Sie einen Algorithmus für das Erzeugen einer möglichst guten bildbezogenen Farbtabelle (beispielsweise für die Bildkompression mit dem GIF-Verfahren).

Median-Cut

6. Mit welchem Texturierungs-Verfahren können geometrisch solide Flächen teilweise transparent gerendert werden?

Alpha-Mapping

7. Bei welchem Shading-Verfahren tritt der so genannte Mach-Band-Effekt besonders deutlich hervor?

Flat-Shading

8. Bei welchem Texturierungs-Verfahren wird die Geometrie der Objekte verändert?

Displacement-Mapping

9. Welches Rendering-Verfahren eignet sich besonders gut, um realistische Spiegelungen und Reflexionen an komplexen Objekten darzustellen?

Ray-Tracing

10. Was wird beim Phong-Shading interpoliert, um eine möglichst realistische Beleuchtung von Flächen zu erreichen?

die Normalen

11. Bei einem Drucker mit CMY-Farbpatrone sei die Magenta-Tinte leer. Welche Farbe ist zu sehen, wenn Sie ein schwarz-weiß Foto ausdrucken?

$1 - (1, 0, 1) = (0, 1, 0)$, also Grün

12. Nennen Sie eine *feste* Stage der OpenGL-Graphics-Pipeline.

Primitive Processing

13. Auf welche Daten kann ein Shader nur lesend zugreifen? Nennen Sie ein Beispiel.

Uniform-Daten

14. In welcher Stage der OpenGL-Graphics-Pipeline kann das Phong-Beleuchtungsmodell implementiert werden?

Fragment-Shader

15. Wie heißt die Stage der OpenGL-Graphics-Pipeline, die Primitives in Fragments überführt?

Rasterizer

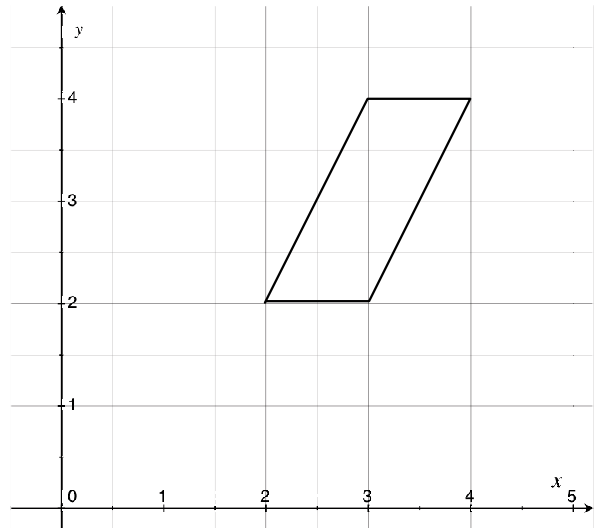
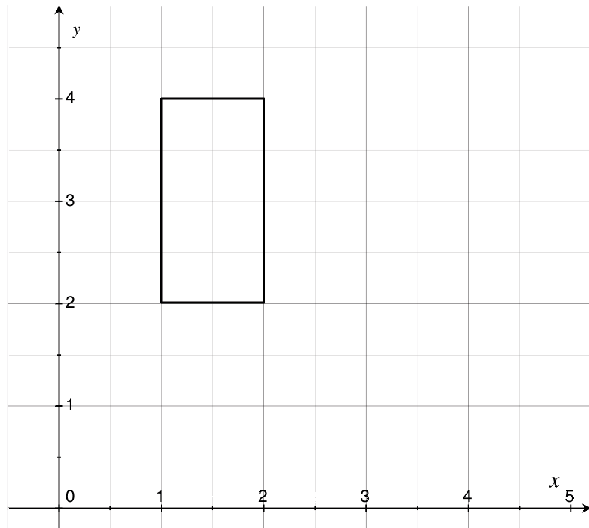
Aufgabe 2 (10 Punkte)

Die in der Vorlesung vorgestellte Viewing-Pipeline zur Projektion von dreidimensionalen Szenen auf ein zweidimensionales Ausgabegerät ist in mehrere Schritte unterteilt. Welche Namen haben diese Schritte? Was passiert im jeweiligen Schritt und wie lauten die üblichen Abkürzungen für die Koordinatensysteme, zwischen denen transformiert wird?

Name des Schrittes	kurze Beschreibung der Aktion im Schritt	Startkoordinaten-system	Endkoordinaten-system
Modeling	Positionierung der Körper in der Welt	MC	WC
View Orientation	Definition des Betrachterstandpunktes	WC	VRC
View Mapping	Projektion in Einheitswürfel	VRC	NPC
Device Mapping	Projektion auf Bildschirm	NPC	DC

Aufgabe 3 (6 Punkte)

Scheren Sie das Viereck in der folgenden Zeichnung um den Scherwert $S_x = 0,5$ und tragen Sie das Ergebnis in das rechte Koordinatensystem ein. (4 Punkte)



Wie lautet die 3×3 -Transformationsmatrix für zweidimensionale Transformationen mit homogenen Koordinaten, für die obige Scherung? (2 Punkte)

Die Matrix lautet:
$$\begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Aufgabe 4 (9 Punkte)

Das unten dargestellte Viereck soll mit ganzzahligen Grauwerten zwischen 0 und 255 eingefärbt werden. Für die Eckpunkte des Vierecks sind die jeweiligen Grauwerte bereits durch die Beleuchtungsgleichung berechnet worden.

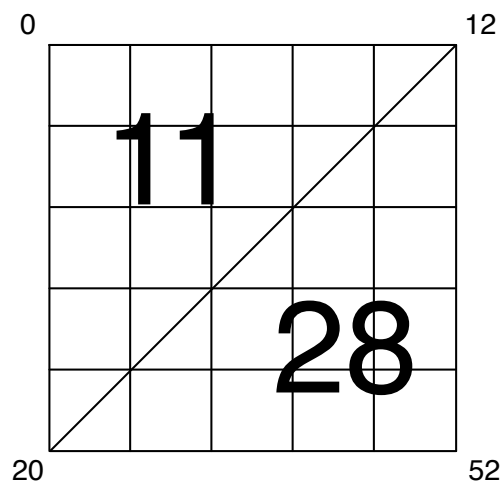
Triangulieren Sie das Viereck und färben Sie es in Vorlage a) nach dem Flat-Shading-Verfahren ein (3 Punkte).

Hinweis: Sie können selbst entscheiden, ob Sie die Pixel auf der Diagonalen mit den Farbwerten des einen oder des anderen Dreiecks einfärben.

Färben Sie es in Vorlage b) nach dem Gouraud-Shading-Verfahren ein (6 Punkte).

Schreiben Sie die errechneten Farbwerte in die jeweils einen Pixel repräsentierenden Kästchen.

a) Flat-Shading



b) Gouraud-Shading

0	3	6	9	12
5	8	11	14	22
10	13	16	24	32
15	18	26	34	42
20	28	36	44	52

Aufgabe 5 (5 Punkte)

Sie möchte in Ihrer Szene ein Backface-Culling durchführen. Betrachten Sie das durch die homogenen Eckpunkte $(1, 0, 1, 1)$, $(2, 1, 0, 1)$, $(2, 2, 1, 1)$ definierte Dreieck. Um das Culling durchführen zu können, müssen Sie die Normale der Fläche (unter der Annahme, dass diese nicht gekrümmt ist) errechnen.

Wie lautet die *normierte* Normale der Fläche unter Berücksichtigung der Tatsache, dass die Eckpunkte gegen den Uhrzeigersinn angeordnet sind, um eine eindeutige Bestimmung der Vorder- und Rückseite zu ermöglichen. Etwaige Brüche und Wurzeln können Sie stehen lassen.

Die Normale lässt sich aus dem Kreuzprodukt zweier Kanten des Dreiecks berechnen:

$\vec{u} = (1, 0, 1) - (2, 1, 0) = (-1, -1, 1)$ und $\vec{v} = (2, 2, 1) - (2, 1, 0) = (0, 1, 1)$. Die Normale ist jetzt $\vec{n} = \vec{v} \times \vec{u} = (2, -1, 1)$, normiert: $(\frac{2}{\sqrt{6}}, \frac{-1}{\sqrt{6}}, \frac{1}{\sqrt{6}})$

Aufgabe 6 (5 Punkte)

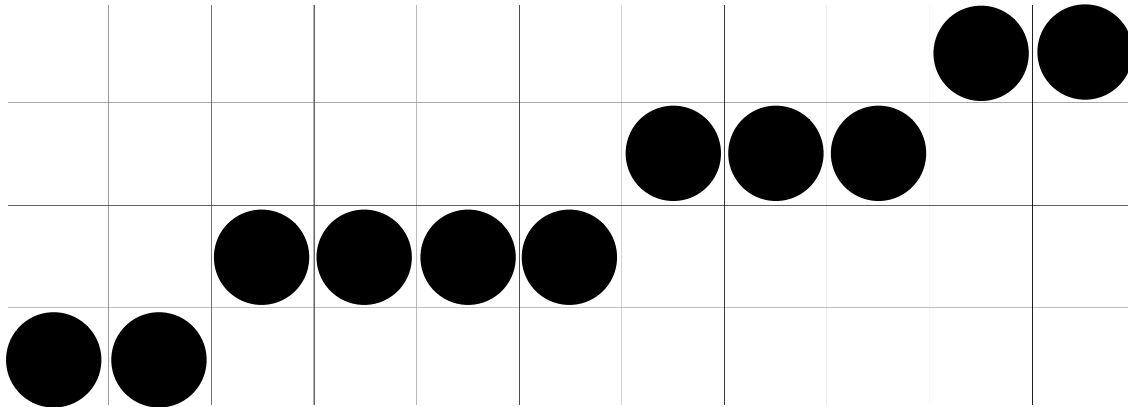
Fortsetzung des Backface-Cullings: Für eine andere Fläche ihrer Szene haben Sie die Normale $(-3, 1, -2, 0)$ berechnet. Der Augenpunkt befindet sich bei $(0, 10, 1, 1)$. Berechnen Sie, ob der Eckpunkt der Fläche $(-1, 1, -1, 1)$ unter Berücksichtigung der Position des Augenpunktes sichtbar bzw. dem Betrachter zugewandt ist.

Ihre Rechnung muss nachvollziehbar sein. Für ein nicht nachvollziehbares *Ja* oder *Nein* als Antwort werden keine Punkte vergeben.

Sei $p \cdot n - a \cdot n = e$ (vgl. Skript). Folglich $(0, 10, 1) \cdot (-3, 1, -2) - (-1, 1, -1) \cdot (-3, 1, -2) = 8 - 6 = 2$. Da $e > 0$ folgt, dass der Eckpunkt vom Augenpunkt **sichtbar** bzw. dem Betrachter zugewandt ist.

Aufgabe 7 (7 Punkte)

Zeichnen Sie mit Hilfe des in der Vorlesung vorgestellten Bresenham-Algorithmus eine Linie zwischen den beiden markierten Punkten. Füllen Sie nur die Kästchen aus, bei denen der Bresenham-Algorithmus einen Pixel setzen würde. Hinweis: Die Steigung der Linie beträgt $\frac{3}{10}$. (4 Punkte)



Der Bresenham-Algorithmus hat den Nachteil, dass Linien unterschiedlicher Länge mit der gleichen Anzahl an Pixeln gezeichnet werden, was zu einem unterschiedlichen Eindruck der Dicke von Linien führt. Nennen Sie ein Verfahren, dass dieses Problem behebt (1 Punkt):

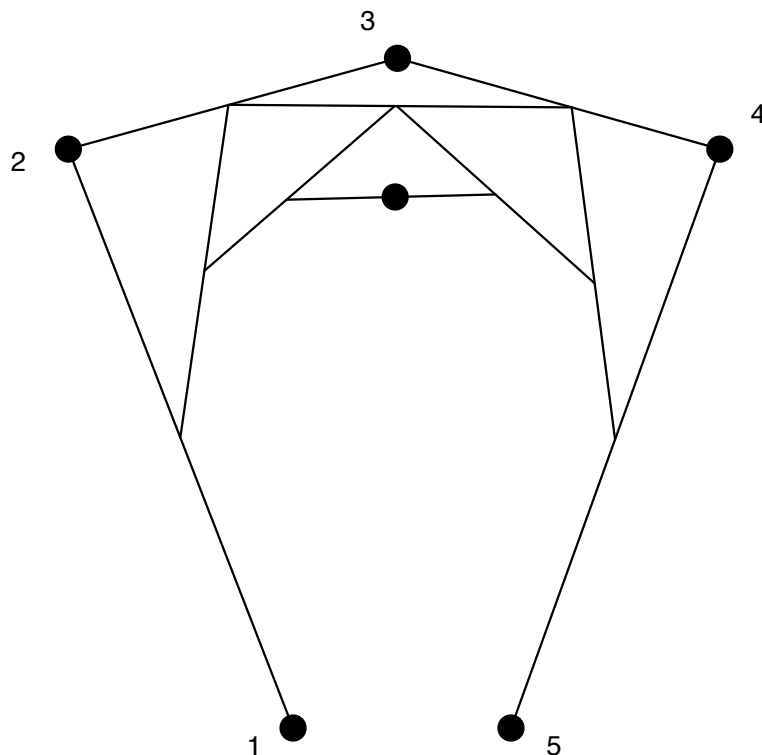
Anti-Aliasing

Eine gerade Bresenham-Linie (ohne Steigung) sei 100 Pixel lang und wird folglich auch mit 100 Pixeln gezeichnet. Wie lang kann eine Bresenham-Linie maximal sein, die ebenfalls mit 100 Pixeln gezeichnet wird und entsprechend dünner wirkt? Geben Sie das genaue Ergebnis oder auf ganze Pixel gerundet an. (2 Punkte)

$$\sqrt{2} \cdot 100 \text{Pixel} = \sqrt{100^2 + 100^2}$$

Aufgabe 8 (7 Punkte)

Sie möchten eine Bézier-Kurve mit den unten dargestellten Stützpunkten zeichnen. Nutzen Sie den Algorithmus von de Casteljau, um approximativ für die Bézier-Kurve mit der Laufvariablen $0 \leq t \leq 1$ den Punkt der Kurve für $t = \frac{1}{2}$ zu bestimmen und zeichnen Sie ihn unten ein. Ihre Zeichnung soll auch Ihr Vorgehen verdeutlichen. Das Einzeichnen eines nicht nachvollziehbar entstandenen Punktes wird mit 0 Punkten bewertet. (4 Punkte)



Wie viele Bernstein-Polynome sind für die genaue Berechnung des oben genannten Punktes notwendig? (1 Punkt)

Welchen Anteil der Kurve müssen Sie neu auswerten, wenn Sie bei einer bereits gezeichneten Bézier-Kurve mit 5 Stützpunkten den letzten Stützpunkt verändern und die Kurve anschließend neu zeichnen wollen? Geben Sie das Ergebnis in Worten, Prozent oder als Intervall (von t) an. (2 Punkte)

Aufgabe 9 (5 Punkte)

Sie möchten eine schwarz-weiße Grafik komprimieren und dazu den LZW-Algorithmus nach Lempel, Ziv und Welch einsetzen. Gehen Sie davon aus, dass Ihre Kodierungs-Tabelle anfangs aussieht, wie unten angegeben. Die Codes 0 und 1 sind also bereits für die Farben schwarz (**s**) und weiß (**w**) festgelegt.

Ein Ausschnitt ihrer Datei enthält den Abschnitt **s w s w s s w s**. Wenden Sie den LZW-Algorithmus an und geben Sie das kodierte Ergebnis obigen Abschnitts sowie die resultierende Code-Tabelle an. (4 Punkte)

Wenn Sie lieber mit der Notation für die Tabelle mit Prä- und Suffixen arbeiten möchten, legen Sie eine eigene Tabelle auf der Rückseite eines der Aufgabenblätter an und verweisen hier darauf.

Wort	Code
s	0
w	1
sw	2
ws	3
s w s	4
s s	5

Eingabe: **s w s w s s w s**, Ausgabe: 01204

Welches aus der Vorlesung bekannte Bildkompressions-Verfahren bzw. Bildformat nutzt den LZW-Algorithmus? (1 Punkt)

GIF

Aufgabe 10 (4 Punkte)

Sie möchten einen Vertex-Shader schreiben, der die übergebenen Vertices abhängig von der Zeit, repräsentiert durch das skalare Uniform `time`, periodisch auf und ab bewegt. Zum Zeitpunkt `time=0` soll der y-Wert des Vertex in Weltkoordinaten unverändert sein. Das Uniform wird nun fortlaufend erhöht und anhand des Wertes soll bestimmt werden, um welchen Betrag der y-Wert des Vertex verändert wird. Der y-Wert soll hierbei maximal um ± 1 verändert werden.

Schließlich wird das veränderte Vertex (`changedVertex`) mit der Model-View-Projection Matrix multipliziert und an den weiteren Verarbeitungsprozess übergeben (bereits vorgegeben). Ergänzen Sie den untenstehenden GLSL-Shadercode, um das oben beschriebene Verhalten umzusetzen.

Hinweis: GLSL stellt verschiedene Rechen-Funktionen zur Verfügung: beispielsweise die Elementarrechenoperationen: $+$, $-$, \cdot , \div , Vektoroperationen: das Skalarprodukt $dot(a, b)$, das Kreuzprodukt $cross(a, b)$, Vektornormierung $normalize(a)$, sowie Operationen für skalare Werte: Modulo $\%$ und trigonometrische Funktionen wie $sin(\alpha)$, $cos(\alpha)$ und $tan(\alpha)$.

```
#version 140

uniform float time;
uniform mat4.mvpMatrix;

in vec4 vertex;

void main() {

    vec4 changedVertex = vertex;

    changedVertex.y = changedVertex.y * sin(time);

    gl_Position =.mvpMatrix * changedVertex;

}
```

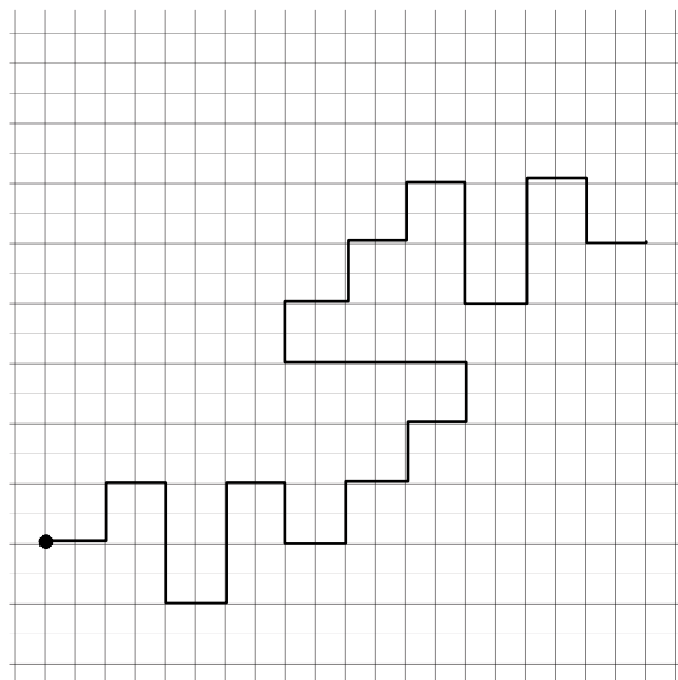
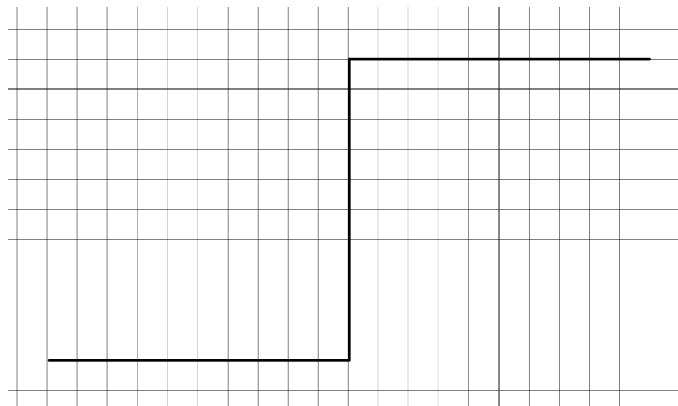
Aufgabe 11 (6 Punkte)

Ein Lindenmayer-System ist eine nicht-grafische Beschreibung, die auch für Fraktale genutzt wird. Ein einfaches L-System sei definiert durch $\Sigma = \{r, u, l, d\}$ für right, up, left, down, also der Richtung, in die eine Linie ausgehend von einem Punkt gezeichnet wird.

Für die Regelmenge gelte:

$$f = \{r \Rightarrow rurddruurdr, l \Rightarrow luldluulddl, u \Rightarrow ururulluru, d \Rightarrow dldrdlldrrd\}$$

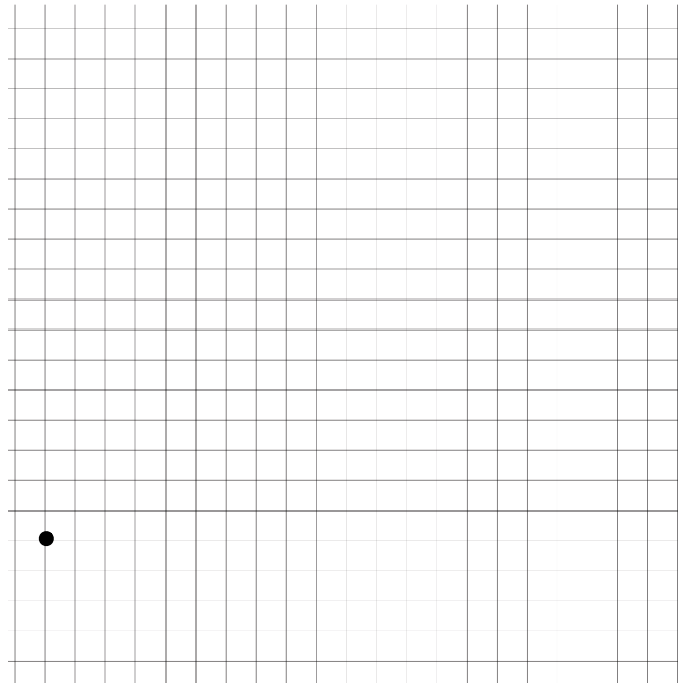
Das unten abgebildete Startwort w sei rur . Zeichnen Sie darunter das einmal iterierte Fraktal $f(w)$. Ein geeigneter Startpunkt ist unten markiert. Auf der nächsten Seite findet sich eine Ersatzvorlage, streichen Sie bei einer Nutzung die nicht zu wertende Vorlage deutlich durch. (4 Punkte)



Berechnen Sie die Fraktale Dimension (nach Vorschrift der Vorlesung) für obiges Fraktal. Etwaige Brüche und Logarithmen können Sie stehen lassen. (2 Punkte)

$$\frac{\log(11)}{\log(5)}, 11 \text{ Kopien, skaliert mit } \frac{1}{5}$$

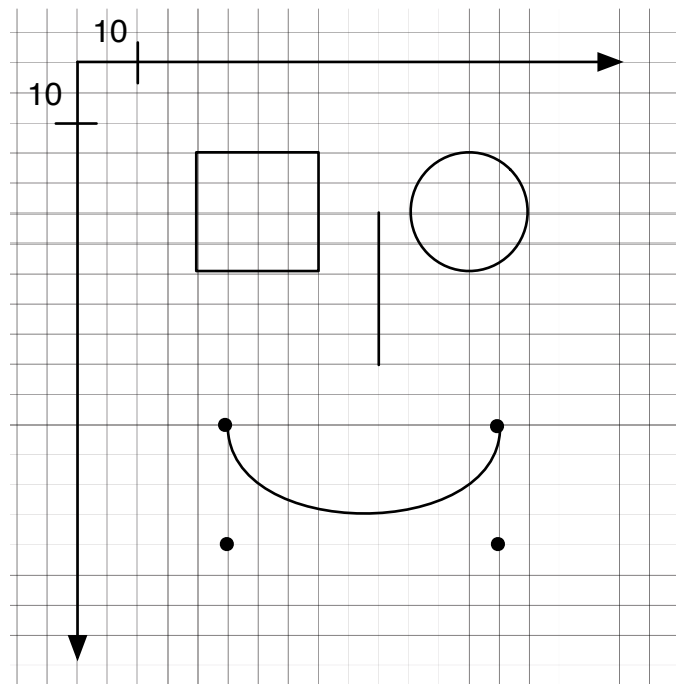
Ersatzvorlage für den ersten Aufgabenteil. Bei Nutzung auf jeden Fall die Originalvorlage deutlich durchstreichen!



Aufgabe 12 (8 Punkte)

Die folgende Abbildung stellt eine SVG-Grafik dar. Das rudimentäre Gesicht besteht aus einem Viereck und einem Kreis für die Augen, einer einfachen Linie für die Nase und einer Bezier-Kurve für den Mund. Die dargestellten Stützpunkte für den Mund gehören nicht zur eigentlichen Grafik.

Vervollständigen Sie die SVG-Datei auf der nächsten Seite, so dass diese die abgebildete Grafik definiert. Achten Sie darauf, dass keines der Objekte gefüllt ist und jedes der Objekte mit einem schwarzen Rand versehen ist. Ein Kästchen in der Vorlage entspricht 5 Pixeln.



```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="600" height="600" xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink">

    <rect x="20" y="15" width="20" height="20" stroke="black" fill="none" />

    <line x1="50" x2="50" y1="25" y2="50" stroke="black" />

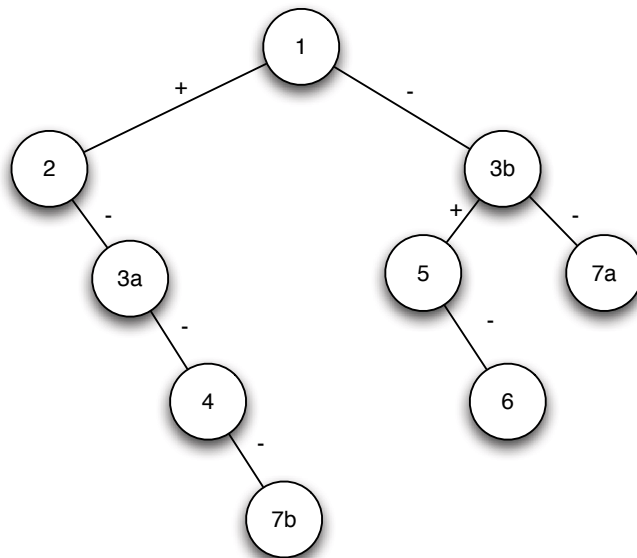
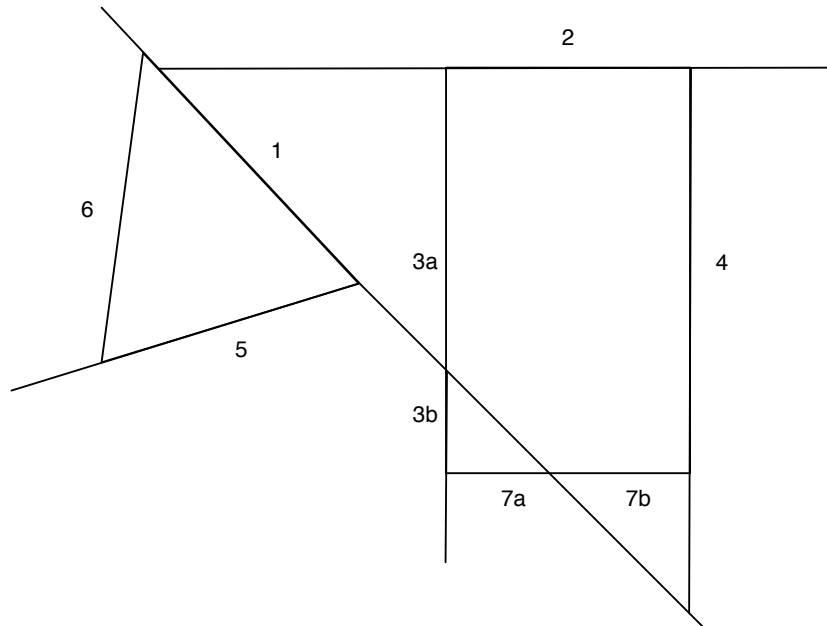
    <circle cx="65" cy="25" r="10" fill="none" stroke="black" />

    <path d="M 20,60 C 20,80 70,80 70,60" fill="none" stroke="black" />

</svg>
```

Aufgabe 13 (5 Punkte)

Die unten dargestellte Abbildung soll eine 2D-Szene sein, zu der Sie einen BSP-Tree erzeugen sollen. Handeln Sie die Kanten der angegebenen Reihenfolge nach ab und bauen Sie so schrittweise Ihren Baum auf. Leere Teilbäume brauchen nicht angegeben zu werden.



Aufgabe 14 (3 Punkte)

Geben sie die Farbe **Gelb** als Tripel in dem jeweiligen Farb-Modell an. Der zulässige Wertebereich ist links angegeben:

RGB	$([0.0, 1.0], [0.0, 1.0], [0.0, 1.0])$	$(1, 1, 0)$
CMY	$([0.0, 1.0], [0.0, 1.0], [0.0, 1.0])$	$(0, 0, 1)$
HSV	$([0^\circ, 360^\circ], [0\%, 100\%], [0\%, 100\%])$	$(60^\circ, 100\%, 100\%)$

Aufgabe 15 (5 Punkte)

Sie möchten eine Fläche Ihrer dreidimensionalen Szene beleuchten. Berechnen Sie den spekularen Beleuchtungsanteil für einen Punkt einer Fläche. Der zu beleuchtende Punkt liegt bei den homogenen Koordinaten $(1, 1, 3, 1)$. Sie haben den Vektor vom Punkt zur Lichtquelle bereits an der Normalen gespiegelt und den Vektor $(0, 1, 0)$ erhalten. Der Augenpunkt befindet sich bei den Koordinaten $(4, 5, 3, 1)$.

Die Formel für den spekularen Lichtanteil aus der Vorlesung lautet $C_s = k_s \cdot I_e \cdot \cos(R, A)^{O_e}$. Nehmen Sie an, dass $k_s = 1$, $I_e = 0,5$ und $O_e = 2$. Berechnen Sie C_s mit den angegebenen Informationen.

Hinweis: Eine Skizze der Situation kann helfen.

Berechnet werden muss der $\cos(R, A)$. $R = (0, 1, 0)$, $A = \text{Augenpunkt} - \text{PunktDerFläche} = (4, 5, 3) - (1, 1, 3) = (3, 4, 0)$. Es gilt $\cos(R, A) = \frac{R \cdot A}{|R| \cdot |A|} = \frac{4}{5}$. Die Gleichung ergibt also insgesamt: $C_s = k_s \cdot I_e \cdot \cos(R, A)^{O_e} = 1 \cdot 0,5 \cdot \left(\frac{4}{5}\right)^2 = \frac{32}{100}$.