Divisie Wiskunde en Informatica	Tentamen **Computer Graphics**
Vrije Universiteit	29 – 01 – 2001

Language disclaimer:
You are kindly asked to answer the questions using the English language. However, if it helps clarifying your answers, you may use *a little* Dutch here and there. Doing so, *will not affect* your result.

Allowed material:
For answering the questions, you are allowed to use all kinds of written material like textbooks, printouts of the lecture slides, your own notes, etc.
However, it is **not allowed** to use any electronic equipment or any means of communication.

Wishing you lots of success with the exam!

Normering

Vraag	1		2			3						4			5			6		7		8
	a	b	a	b	c	a	b	c	d	e	f	a	b	c	a	b	c	a	b	a	b	
Punten	4	4	4	4	2	4	3	3	4	6	2	3	6	2	3	3	3	6	6	2	10	6

Totaal: 90 (+10) = 100

1. Synthetic Camera Model

a) Describe how an image is constructed using the synthetic camera model!

b) What is the most important advantage of the synthetic camera model, compared to the pen-plotter model?

2. Callback Functions

a) For which purpose do interactive graphics environments (like OpenGL) use the callback function mechanism? Name four examples of callback functions used by OpenGL!

b) Write a callback function in the C language using OpenGL (the GLUT library) that allows a user to select a range in the active window using the mouse! The user selects a range by moving the mouse to one corner of the range, pressing the left mouse button, moving the mouse (while keeping the button pressed) to the opposite corner of the range, and finally releasing the mouse button. The screen coordinates have to be filled into the following struct variable `range` (see below/next page). (It is not necessary that your callback function visualizes the range while selecting.)

c) How can your OpenGL application ensure that the results of the mouse callback have any effect on the displayed image?

```
typedef struct {
  int x_start, y_start; /* first corner */
  int x_end, y_end;     /* opposite corner */
} range_type;
range_type range;
```

3. Color

a) Explain how different colors are composed in the RGB model! Why can the RGB model create colors that are useful for the human observer?

b) Explain how the CMY color model differs from RGB! What is the main application area for the CMY model?

c) Sketch a color cube for the RGB model in a homogeneous coordinate system defined by the frame (v_1, v_2, v_3, P_0) with:

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad P_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Here, a color c is denoted by its representation $c = P_0 + r \cdot v_1 + g \cdot v_2 + b \cdot v_3, 0 \leq r, g, b \leq 1$ Mark the eight corners of the cube with the respective colors red, green, blue, black, white, cyan, magenta, yellow!

d) Add to your drawing from part c) the three vectors of a second frame

$$u_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad u_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad u_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad Q_0 = P_0 + v_1 + v_2 + v_3$$

(This frame denotes the corresponding color cube for the CMY model.)

e) Identify the matrix M with

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = M \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix}$$

Compute M^T and then $A = \left(M^T\right)^{-1}$!
Hint: A can be computed easily because of its very simple structure; keep in mind that:

$$M^T \left(M^T\right)^{-1} = I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Using A, compute the RGB representation c' of a color from its CMY representation as

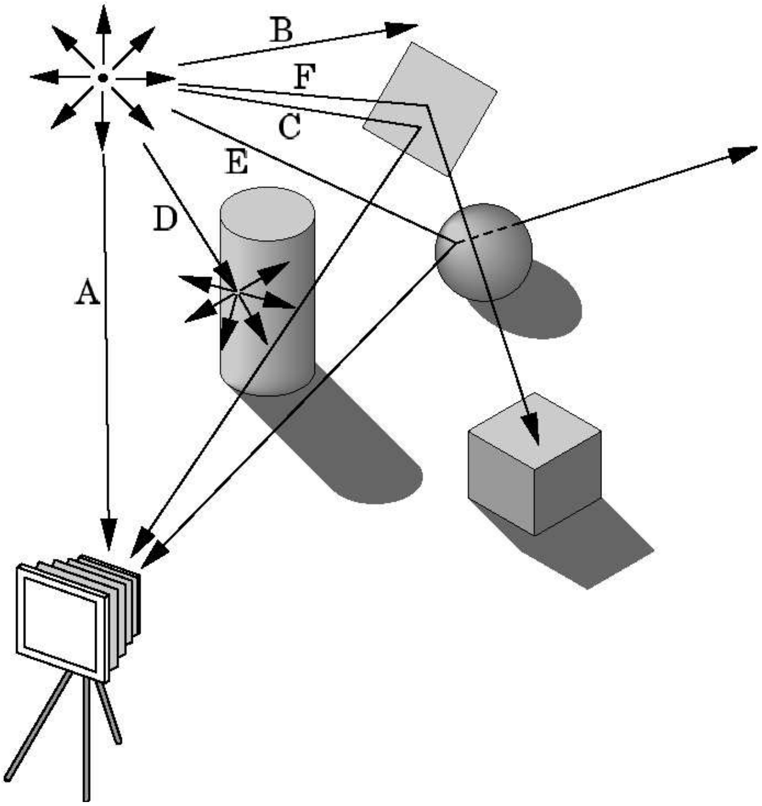
$$c' = Ac = A \begin{bmatrix} c \\ m \\ y \\ 1 \end{bmatrix}$$

f) Using the result from part e), write a C function
`void cmyColor3f(GLfloat c, GLfloat m, GLfloat y)`
that takes a color in the CMY system as parameter, computes the corresponding RGB values, and calls `glColor3f` to set the color in an OpenGL system!

4. Ray Tracing

a) Explain how images are created using the ray tracing technique!

b) Look at the rays denoted *A* to *F* in the following picture! What happens to each ray and how does it contribute to the image created in the camera?



c) What is the most important disadvantage of ray tracing because of which it is not used, for example, by OpenGL?

5. Antialiasing

a) A standard antialiasing technique used in ray tracing is to cast rays not only through the center of each pixel, but also through the four corners of the pixel. What is the increase in necessary computation compared to using only a single ray per pixel?

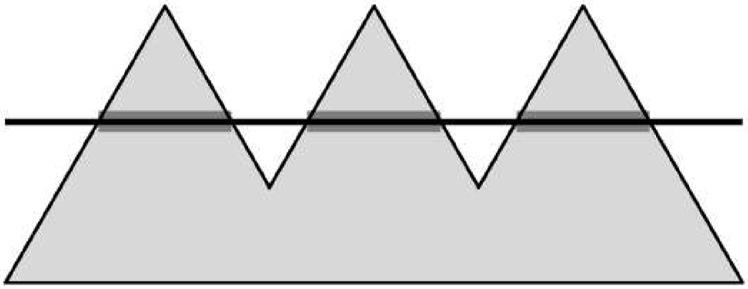
b) Although an ideal pixel is a square of 1 unit per side, most CRT systems generate round pixels that can be approximated as circles of uniform intensity. If a completely full unit square has intensity 1.0, and an empty square has intensity 0.0, how does the intensity of a displayed pixel vary with the radius of the circle? Compute the intensity of a pixel with radius 0.5! Also compute the radius r_1 corresponds to the intensity 1.0! Which side effect has the use of pixels with radius r_1?
(Hint: approximate your calculations with up to 2 decimal digits.)

If the intensity of the screen is the percentage of pixels set, multiplied by the intensity of the individual pixel: does the intensity of the screen also equal 1.0 when using pixels of radius r_1 and setting all pixels?

c) Why is defocusing the beam of a CRT sometimes called "the poor person's antialiasing"?

6. XOR

a) Show how you can use the XOR operation to implement the odd-even polygon fill algorithm! Assume the simple case in which 0 is the background color and 1 is the edge and fill color. Which writing mode do you use to modify the image?
(Hint: The edges are already drawn before the polygon will be filled.)



b) Reconsider the range selection from question 2.b)
Suppose you wish to visualize the currently selected rectangle, while the mouse button is being pressed. Assume that the mouse callback may not only get invoked when a mouse button is pressed or released, but also when the mouse is moved while a button is being held down. In that case, assume the button state to be GLUT_HOLD. Also assume that you have a function that allows you to draw a rectangle in XOR writing mode. How would you implement the visualization of the range selection?

7. Graphics Objects

a) Explain the notion and use of “instance transformation” for graphics objects!

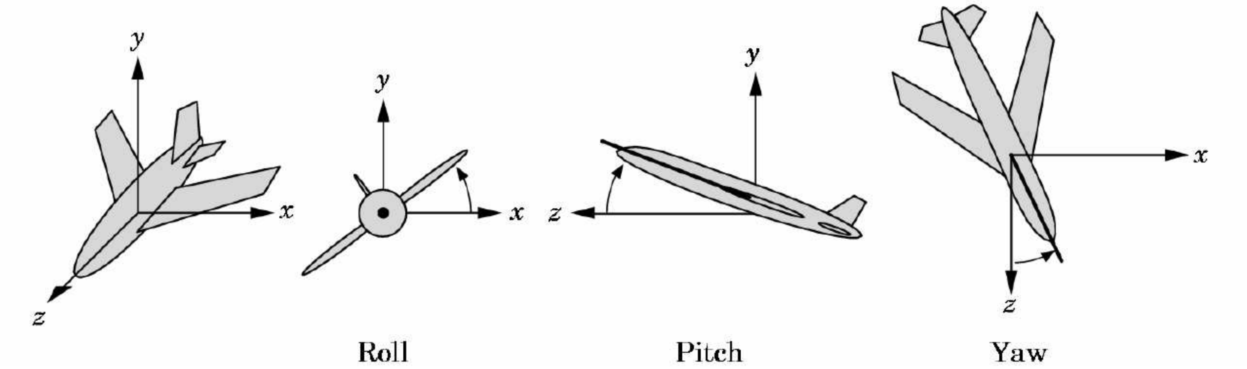
b) Suppose you want to animate the movement of the parts of a complex machine. For the individual parts, you can use graphical objects from a library. Your animation shall show the transition of the machine from one position to a second position, expressed via the rotation angles between the parts of the machine. Assume the description of the machine already exists in form of a tree of nodes of type node, representing the hierarchical structure of the library objects forming the machine.

Write a C function void animate(node *tree, int step, int total_steps) that produces one image of the machine! (Your function shall use OpenGL to produce the image.) The transition from the first to the second position of the machine shall happen in total_steps steps; your function should just display the step step $\in 0 \ldots$ total_steps $- 1$. The struct element positive gives the direction of the movement; if positive == TRUE, then the angle shall be increased from angle1 to angle2, and decreased otherwise. The function attrib_function allows to set atributes like color or material properties; it is optional and shall only be called if the pointer is not NULL;

```
#define FALSE 0
#define TRUE  1
typedef struct node {
  /* relative to the parent node in the tree:          */
  GLfloat tx,ty,tz;           /* translation           */
  GLfloat sx,sy,sz;           /* scaling               */
  GLfloat rx,ry,rz;           /* rotation axis         */
  GLfloat angle1,angle2;      /* angle of rotation,    */
                              /* position 1 and 2      */
  int     positive;           /* direction of movement */
                              /* TRUE or FALSE         */
  void (*attrib_function)();  /* setting attributes,   */
                              /* if not NULL           */
  void (*drawing_function)(); /* draw the object       */
  struct node *sibling;       /* list of sibling nodes */
  struct node *child;         /* list child nodes      */
} node;
```

8. Viewing

Consider an airplane whose position is specified by the roll, pitch, and yaw, and by the distance from an object. Find a model-view matrix in terms of these parameters!
(Hint: It is not required to manually multiply individual matrix elements!)



y
x
z
y
x
Roll
y
z
Pitch
x
z
Yaw