Department of Computer Science	Exam **Computer Graphics**
Vrije Universiteit	04 – 04 – 2005

This is a *"closed book"* exam.
No printed materials or electronic devices are admitted for use during the exam.
You are supposed to answer the questions **in English**.

Wishing you lots of success with the exam!

Points per question (maximum)

Q	1		2			3				4			5			6		7			8		
	a	b	a	b	c	a	b	c	d	a	b	c	a	b	c	a	b	a	b	c	a	b	c
P	3	3	4	4	4	4	4	5	4	5	3	3	10	4	4	3	3	3	3	3	4	4	3

Total: 90 (+10 bonus) = 100

1. Human Visual System

a) Explain the following properties of the human visual system as far as they are relevant for the perception of images generated by computer graphics!

- human color perception
- CIE standard observer curve
- lateral inhibition

b) What can a computer-graphics application do to work around the following properties of the human visual system

- standard observer curve,
- lateral inhibition ?

2. Input Devices

a) What are the *measure* and *trigger* of an input device? What are the *measure* and *trigger* of a *mouse*? How can this information be derived from the physical mouse device?

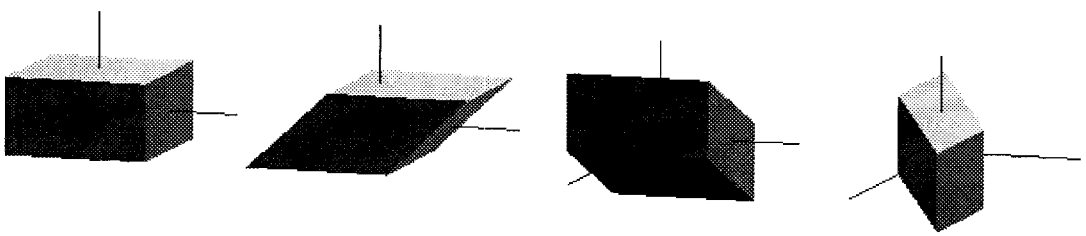
b) Explain the different input modes: request mode, sample mode, event mode! Give examples for each mode! Which input mode is implemented in OpenGL's callback functions? And what is the advantage of doing so?

c) 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.)

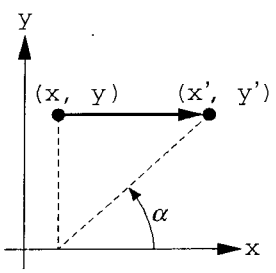
```
typedef struct {
  int x_start, y_start; /* first corner */
  int x_end, y_end;     /* opposite corner */
} range_type;
range_type range;
```

3. Shear

The following pictures show (from left to right) a cube, and the same cube sheared along the X axis, the Y axis, and the Z axis.



The following drawing (left) shows how the shear along the X axis can be represented depending on a shearing angle α. H_x is the respective transformation matrix.



$$H_x(\alpha) = \begin{bmatrix} 1 & \cot\alpha & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a) Make similar drawings, one for the shear along the Y axis (angle β), and one for the shear along the Z axis (angle γ), according to the above picture!

b) Determine the shear matrices $H_y(\beta)$, and $H_z(\gamma)$, corresponding to part a!

c) Implement a C function `void glShearX(GLfloat alpha)` that works like `glTranslate` or `glRotate` (affecting the currently active transformation matrix by multiplying $H_x(\alpha)$ to it! You can assume that a function cot() is available.

d) Determine a shear matrix $H(\alpha, \beta, \gamma)$ that combines the effects of $H_x(\alpha)$, $H_y(\beta)$, and $H_z(\gamma)$!

Assume you have implemented the function glShearX, and also (analogously) glShearY and glShearZ. Is it possible to implement a function `glShearXYZ(alpha,beta,gamma)` by a combination of calls to glShearX, glShearY, and glShearZ? Give arguments for your answer!

4. Antialiasing

a) 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 that corresponds to the intensity 1.0! Which side effect has the use of pixels with radius r_1?
(Hints: approximate your calculations with 1 or 2 decimal digits, $\sqrt{\pi} \approx 1.8$)

If the intensity of the whole 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?

b) Defocusing the beam of a CRT is sometimes called "the poor man's antialiasing." Can you explain why?

c) 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?

5. Scene Graphs

Consider the class definition `Node` for scene graphs and its two subclasses `Translation` and `Polygon` below. All specific classes of nodes (geometric objects, transformations, lights, material properties, etc.) are supposed to be subclasses of `Node`. The `Render` method shall "render" the given object, like drawing a polygon or applying a transformation. The scene graph shall be organized as a left-child, right-sibling tree. Rendering is supposed to be triggered by invoking `Traverse` on the root node of a scene graph, which in turn will invoke `Render` on all nodes of the tree. In the following, use OpenGL calls where necessary.

```
class Node{
  public:
    Node();
    virtual ~Node();
    void AddChild(Node *);
    void Traverse();
    virtual void Render();
  private:
    Node *LeftChild;
    Node *RightSibling;
};

class Translation: public Node{
  public:
    Translation(float x, float y, float z);
    virtual ~Translation();
    virtual void Render();
  private:
    float dx,dy,dz;
};

class Polygon: public Node{
  public:
    Polygon(float v[][3], int n);
    virtual ~Polygon();
    virtual void Render();
  private:
    float vertices[1000][3];
    int size;
};
```

a) Implement the constructor of class `Node`, and its methods `AddChild` and `Traverse`!

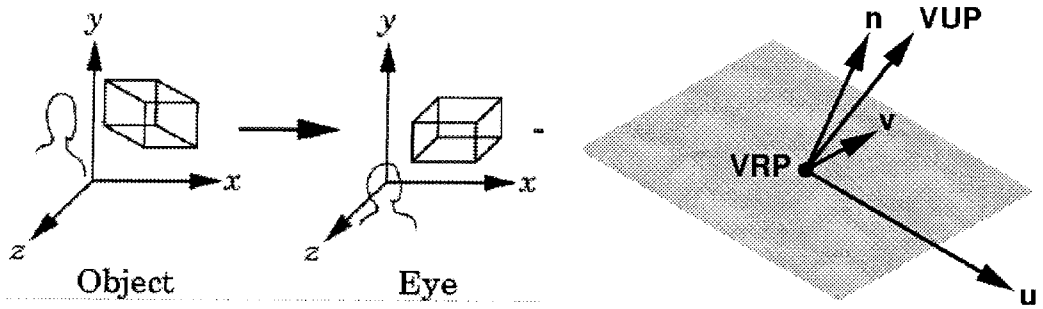
b) Implement the method `Render` of class `Translation`!

c) Implement the method `Render` of class `Polygon`!

6. Viewing

a) Explain (briefly) how the transformation from object (world) coordinates to eye (camera) coordinates contributes to a viewing API!

b) What are the components of a u-v-n viewing coordinate system?



7. Polygon Shading

a) Explain the basic idea of the *Phong reflection* model! Draw a simple figure that shows the vectors involved in computing the shade of a given point on the surface of an object!

b) Explain how *flat shading* works, for example for a polygonal mesh! What are the advantage and the disadvantage of flat shading?

c) Explain *Phong shading* and how it improves over the disadvantage of flat shading!

8. Curves and Surfaces

a) For a parametric curve, explain the degrees of continuity C^0, C^1, and C^2!

b) A curve segment of a Bezier curve is shown in the figure (right). Are Bezier curves C^0, C^1, or C^2? Explain why!

c) For a 512×640 pixel window, what is the maximum number of subdivisions that are needed to render a cubic polynomial surface?