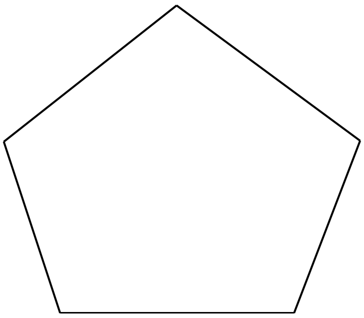
Assignment in Computer Graphics II

– Assignment 7 –

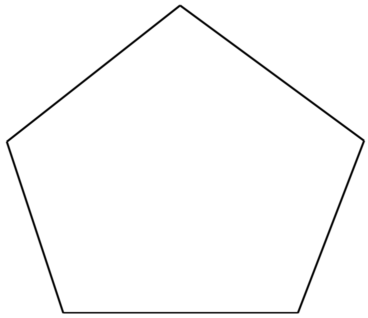
Computer Graphics and Multimedia Systems Group
David Bulczak, Christoph Schikora

Assignment 1 [1 Point] Subdivision Curves

Perform one step of a subdivision procedure for each of the pentagons below. Chaikin (left) and the 4-point method (right) by drawing the new polygon and its vertices (the exact position of the vertices are not relevant).



Chaikin



4-Point

Assignment 2 [2 Points] Solid Modeling

To achieve boolean operations with polygonal b-reps, efficient computations for polygon intersections are necessary.

Given two planar polygons with vertices $\{\mathbf{P}_1, \dots, \mathbf{P}_k\}$ respectively $\{\mathbf{Q}_1, \dots, \mathbf{Q}_l\}$ and two planes E_P, E_Q containing the polygons.

E_P, E_Q are given in Hesse normal form:

$$E_P: \quad \hat{\mathbf{n}}_P \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + d_P = 0, \quad \hat{\mathbf{n}}_P = \begin{pmatrix} a_P \\ b_P \\ c_P \end{pmatrix}$$

$$E_Q: \quad \hat{\mathbf{n}}_Q \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + d_Q = 0, \quad \hat{\mathbf{n}}_Q = \begin{pmatrix} a_Q \\ b_Q \\ c_Q \end{pmatrix}$$

1. Show that for arbitrary vectors

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

the following holds:

$$\mathbf{b} \times \mathbf{a} = -\mathbf{a} \times \mathbf{b}, \tag{1}$$

$$\mathbf{a} \times \mathbf{a} = 0, \tag{2}$$

$$\mathbf{a} \cdot (\mathbf{a} \times \mathbf{b}) = 0, \tag{3}$$

$$\mathbf{a} \times (\beta \mathbf{b} + \gamma \mathbf{c}) = \beta(\mathbf{a} \times \mathbf{b}) + \gamma(\mathbf{a} \times \mathbf{c}) \tag{4}$$

with scalar factors $\beta, \gamma \in \mathbb{R}$.

2. To implement the intersection computation we need a function `intersectPlane`, that intersects two planes E_P, E_Q and returns a line G in parametric form.
   Proof that

   $$G : \mathbf{P} + \alpha \vec{\mathbf{l}}, \alpha \in \mathbb{R} \text{ with } \mathbf{P} = \frac{(d_Q \hat{\mathbf{n}}_P - d_P \hat{\mathbf{n}}_Q) \times (\hat{\mathbf{n}}_P \times \hat{\mathbf{n}}_Q)}{\|\hat{\mathbf{n}}_P \times \hat{\mathbf{n}}_Q\|^2} \text{ and } \vec{\mathbf{l}} = \hat{\mathbf{n}}_P \times \hat{\mathbf{n}}_Q$$

   solves the problem!

   Hint:

   1. **Think about the properties of G w.r.t. the planes.**
   2. $\hat{\mathbf{a}} \cdot (\hat{\mathbf{b}} \times (\hat{\mathbf{c}} \times \hat{\mathbf{d}})) = (\hat{\mathbf{a}} \times \hat{\mathbf{b}}) \cdot (\hat{\mathbf{c}} \times \hat{\mathbf{d}})$

Assignment 3 [2 Points] L-Systems

Given the alphabet $V = \{\Delta, s, +, -, [,]\}$, the axiom $\omega_0 = \Delta$ and the rule

$$p(\Delta) = s[\Delta][+\Delta][-\Delta], \quad \text{else } p(x) = x, \forall x \neq \Delta$$

The geometrical interpretation of a word is implemented in an OpenGL program in the following way:

character		**OpenGL-Code**
Δ	$\longrightarrow$	`glBegin(GL_TRIANGLES);`
		`glVertex2f(0.0, 0.0);`
		`glVertex2f(1.0, 0.0);`
		`glVertex2f(0.5, sqrt(3.0)/2.0);`
		`glEnd();`
s	$\longrightarrow$	`glScalef(0.5, 0.5, 1.0);`
+	$\longrightarrow$	`glTranslatef(1.0, 0.0, 0.0);`
-	$\longrightarrow$	`glTranslatef(0.5, sqrt(3.0)/2.0, 0.0);`
[	$\longrightarrow$	`glPushMatrix();`
]	$\longrightarrow$	`glPopMatrix();`

Sketch the figures you get after 0-, 1-, 2- and 3- iterations.

Hand in: 30.11.2015, at beginning of the lecture or until 12:00 in the mailbox of the chair (next to room H-A 7107)