Exam: **Computer Graphics II** **Mock exam**	Examiner:	Date/Time:
Semester:	Duration: **120 minutes**	Max. number of points: **120**
Allowed auxiliaries: **Drawing utensils, ruler, non-programmable calculator**		

PLEASE NOTE:

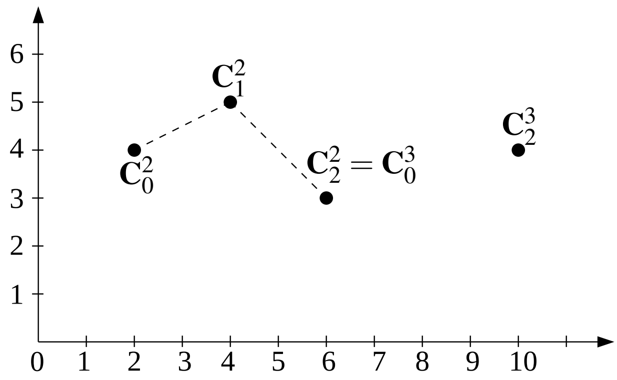
Please note that the topics of the real exam can possibly differ from this mock exam. **Basically all topics that were discussed in the lecture are relevant for the exam.** The discussion of the mock exam will take place in the lecture on July 19, 2018. Please note that we will only discuss assignments (or parts of it) that you have specific questions for. Work through the mock exam and send questions **at least 2 days before the discussion date** to markus.kluge@uni-siegen.de or to dmitri.presnov@uni-siegen.de.

Assignment 1 Splines (36 points)

1.1. Given two *quadratic Bézier* segments defined by control points

$$\mathbf{C}_0^2 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \mathbf{C}_1^2 = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \mathbf{C}_2^2 = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$$

and $\mathbf{C}_0^3$, $\mathbf{C}_1^3$, $\mathbf{C}_2^3$, whereby only $\mathbf{C}_0^3 = \mathbf{C}_2^2$ and $\mathbf{C}_2^3 = \begin{pmatrix} 10 \\ 4 \end{pmatrix}$ are known.



a) Calculate the missing control point $\mathbf{C}_1^3$ so that the resulting Bézier spline between two segments is C^1 continuous.

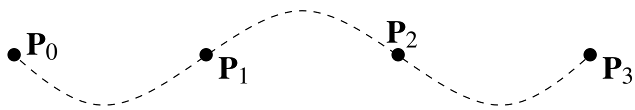
b) Proof *by calculation* using the de Boor algorithm that the *uniform, quadratic B-spline* curve $\mathbf{D}(u)$ defined by the de Boor points

$$\mathbf{D}_0 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \mathbf{D}_1 = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \mathbf{D}_2 = \begin{pmatrix} 8 \\ 1 \end{pmatrix}, \mathbf{D}_3 = \begin{pmatrix} 12 \\ 7 \end{pmatrix}$$

interpolates the point $\mathbf{C}_0^2$. Hint: Choose u so that $\mathbf{D}(u) = \mathbf{C}_0^2$ is the starting point of the B-spline curve. The weights $\alpha_i^j(u)$, which are needed in the algorithm, can be calculated by means of the following formula:

$$\alpha_i^j(u) = \frac{u - t_i}{t_{i-j+3} - t_i} \text{ mit } i = j, \ldots, 2 \text{ und } j = 1, 2$$

1.2. Given arbitrary, pairwise different points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ in 3D that should be interpolated by a curve (see sketch).



a) Given the first de Boor point $\mathbf{D}_0$ of a uniform, C^1 continuous *quadratic* B-spline curve $\mathbf{D}$. Determine the four next de Boor points $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4$ of the B-spline curve $\mathbf{D}$ as a function of the given points so that $\mathbf{P}_0, \ldots, \mathbf{P}_3$ are interpolated by the B-spline curve $\mathbf{D}$.

b) An interpolating *cubic* Bézier spline curve can be defined by specifying control points of each curve segment by means of the Catmull-Rom approach. The resulting Bézier spline curve is C^1 continuous at the interpolated control points.
Determine Bézier control points $\mathbf{C}_0^1, \mathbf{C}_1^1, \mathbf{C}_2^1, \mathbf{C}_3^1$ of the curve segment $\mathbf{C}^1(u)$ between $\mathbf{P}_1$ and $\mathbf{P}_2$ as a function of the points $\mathbf{P}_0, \ldots, \mathbf{P}_3$.

c) Find a representation

$$\mathbf{C}^1(u) = \sum_{k=0}^{3} \mathbf{P}_k \mathbf{X}_k(u)$$

of the curve segment $\mathbf{C}^1(u) = \sum_{k=0}^{3} \mathbf{C}_k^1 \mathbf{B}_k^3(u)$ that only depends on the points $\mathbf{P}_0, \ldots, \mathbf{P}_3$ and Bernstein-Bézier polynomials $\mathbf{B}_0^3, \mathbf{B}_1^3, \mathbf{B}_2^3, \mathbf{B}_3^3$. For this purpose determine the polynomials $\mathbf{X}_0(u), \mathbf{X}_1(u), \mathbf{X}_2(u), \mathbf{X}_3(u)$ as a function of the Bernstein-Bézier polynomials.

Assignment 2 Frenet Frame (6 points)

Given the curve $\mathbf{C}(u) = \left(\sin(u), u^2, 0\right)^T$. Determine the Frenet frame in $u = 0$.

Assignment 3 Polygon Meshes (28 points)

3.1. Given a *closed*, 2-manifold polygon mesh with *genus* 0 so that the *Eulers formula*

$$N_V - N_E + N_F = 2$$

is valid, whereby N_V, N_E and N_F are the number of vertices, edges and faces respectively.

a) Proof with the help of the properties of 2-manifold polygon meshes that generally is valid:

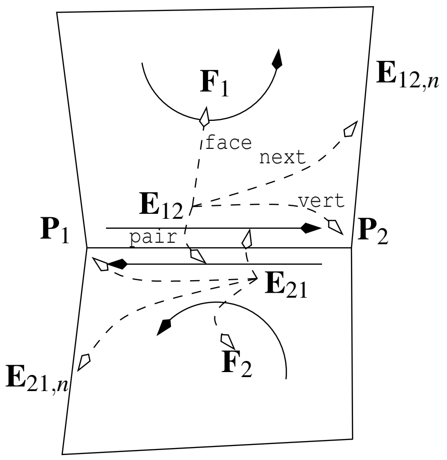
$$N_V \geq 4 \text{ und } N_F \geq 4$$

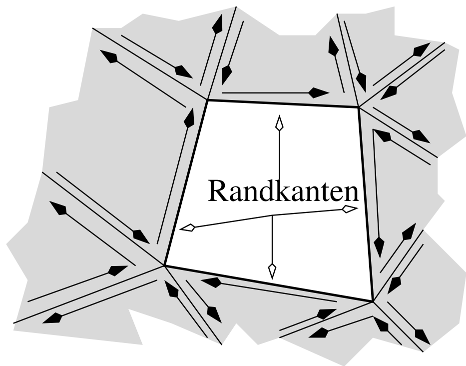
b) For which polygon meshes with the above properties holds exactly:

$$N_V = 4 \text{ und } N_F = 4$$

c) Deduce from the (maximum) lower limits from the assignment part a) a maximum lower limit for N_E!

3.2. The *Half-Edge* data structure mainly works with references to edges. Additionally for each vertex `V` is given a reference to an outgoing half-edge `V.edge` and for each face `F` a reference to an arbitrary associated half-edge `F.edge`.





Develop a pseudo-code for the following tasks using the names of references:

a) Given a polygon, all half-edges to the polygon should be found.

b) Given a vertex, all half-edges around this vertex should be found.

c) Given a vertex on a border polygonal path, all half-edges of the border polygonal path should be found.

Assignment 4 Quaternions (14 points)

Rotate the point $\mathbf{P} = (-2, 1, 0)$ twice by means of **concatenation of quaternions**: first by angle Φ_1 about the axis $\hat{\mathbf{v}}_1$ and then by angle Φ_2 about the axis $\hat{\mathbf{v}}_2$.

$$\Phi_1 = \frac{\pi}{2}, \Phi_2 = \pi, \hat{\mathbf{v}}_1 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \hat{\mathbf{v}}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

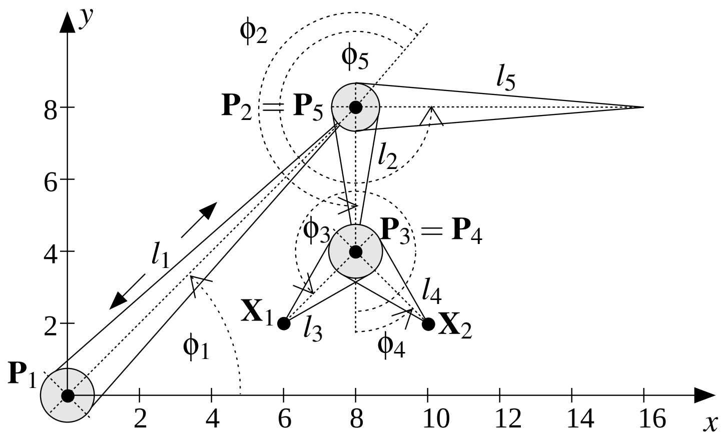
a) For this purpose, first determine all quaternions ($\underline{\mathbf{q}}$, $\underline{\overline{\mathbf{q}}}$, $\underline{\mathbf{q}}_P$) required for the concatenated rotation.

b) Then apply the rotation and calculate the rotated point $\mathbf{P}'$.

c) What are the resulting rotation axis $\hat{\mathbf{v}}_{1,2}$ and angle $\Phi_{1,2}$?

Hint: observe the formula for quaternion multiplication in the annex.

Assignment 5 2D skeleton animation (24 Punkte)



Given the figure and the following values for a two-dimensional, two-tier skeleton model.

$$\mathbf{P}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \phi_1 = \frac{\pi}{4}, \quad \phi_2 = \pi + \frac{\pi}{4}, \quad \phi_3 = \frac{3\pi}{2} + \frac{\pi}{4},$$

$$l_1 = 8\sqrt{2}, \quad l_2 = 4, \quad l_3 = 2\sqrt{2}, \quad l_4 = 2\sqrt{2}$$

7.1. Calculate the end-effector $\mathbf{X}_1$ by successively calculating the intermediate points $\mathbf{P}_2$ und $\mathbf{P}_3$ *in global coordinates*.

7.2. For the given model, give a *hierarchy of transformations* so that the local coordinate system of each segment can be computed.
Please note: Use the following terms:
$T(x, y)$: Translation
$R(\phi)$: Rotation about the origin

7.3. Specify the *workspace* of the end effector $\mathbf{X}_1$ for the given lengths and angles. Explain your claim briefly.

Assignment 6 Spline-based animation (12 Punkte)

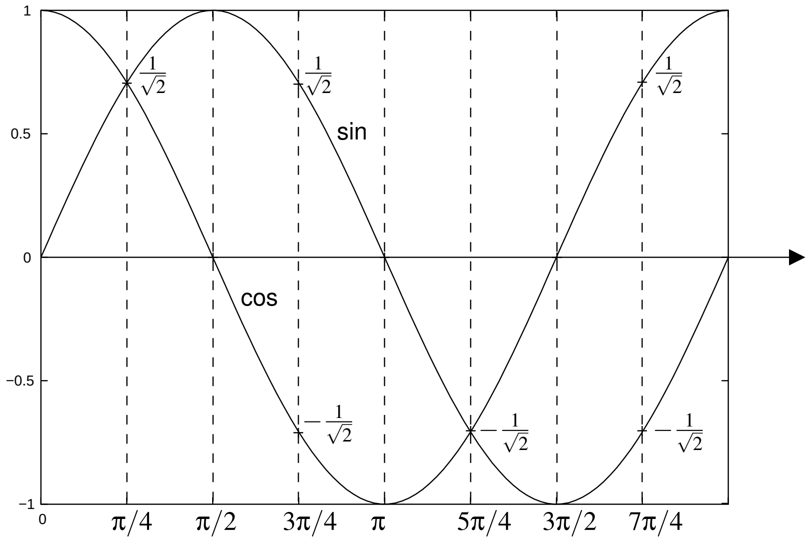
The curve points of a quadratic Bézier curve for $u_0 = 0$, $u_1 = 0.2$, $u_2 = 0.4$, $u_3 = 0.6$, $u_4 = 0.8$, $u_5 = 1$ are

$$\mathbf{C}(u_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{C}(u_1) = \begin{pmatrix} 9 \\ 10 \end{pmatrix}, \mathbf{C}(u_2) = \begin{pmatrix} 16 \\ 20 \end{pmatrix}, \mathbf{C}(u_3) = \begin{pmatrix} 21 \\ 30 \end{pmatrix}, \mathbf{C}(u_4) = \begin{pmatrix} 24 \\ 40 \end{pmatrix}, \mathbf{C}(u_5) = \begin{pmatrix} 25 \\ 50 \end{pmatrix}.$$

Determine the parameters u_i^* $(i = 0, ..., 5)$ so that the distance between two consecutive curve points $\mathbf{C}(u_i)$ is equal for all points. Therefore, **calculate** all u_i $(i = 0, ..., 5)$ approximatively for arc length l_i between curve points $\mathbf{C}(u_0)$ and $\mathbf{C}(u_i)$ only with the help of the given curve points. Then, divide the total curve length into five equidistant parts l_i^* and determine the corresponding parameter u_i^* for each arc length l_i^* using linear interpolation at the corresponding intervals.

Annex

Sine and cosine:



The notation for a quaternion $q \in \mathbb{H}$ is the following:

$$q = (s, \vec{\mathbf{v}}) = (s, (v_1, v_2, v_3)) \mathrel{\widehat{=}} s + v_1 i + v_2 j + v_3 k$$

Multiplication of the imaginary units:

$$i^2 = j^2 = k^2 = -1, \quad ij = k,\ jk = i,\ ki = j \text{ and } ji = -k,\ kj = -i,\ ik = -j$$

Multiplication of two quaternions: $q_1 = (s_1, \vec{\mathbf{v}}_1)$ and $q_2 = (s_2, \vec{\mathbf{v}}_2)$:

$$\underline{\mathbf{q}}_1 \underline{\mathbf{q}}_2 = (s_1 s_2 - (\vec{\mathbf{v}}_1 \cdot \vec{\mathbf{v}}_2),\ s_1 \vec{\mathbf{v}}_2 + s_2 \vec{\mathbf{v}}_1 + \vec{\mathbf{v}}_1 \times \vec{\mathbf{v}}_2)$$

Please note: The multiplication of two quaternions is not commutative!