First Name: **Signature:**

Last Name:

EPFL Student Number: _____

EPFL - Geometric Computing Laboratory

Exam 2023 (C)

Introduction to Computer Graphics 2023

Procedures:

Leave your **student ID card** on the table during the examination for verification. Write your **first and last name** as well as your **EPFL student number** on the first page and then write your initials on all subsequent pages. Use only black or blue ink pens and write legibly. Exam questions done in pencil or other colored pens will **NOT** be graded. Write your solutions on these sheets. Check if the exam is complete by verifying that no page is missing.

Regulations:

You are allowed to use one A4 sheet with hand-written notes on both sides for the exam. Course notes, textbooks, other books/printed materials, calculators are **NOT** permitted during the examination. The use of electronic devices such as cell phones, laptops, etc. is strictly **PROHIBITED** during the examination.

Students found cheating or violating any of the above regulations will fail the class.

Exercise	*Max. points*	*Earned points*
1. Lighting	6	
2. Raytracing	15	
3. Transformations and Projections	16	
4. Textures	7	
5. GPU Pipeline	10	
6. Procedural Models	15	
7. Affine Combinations	5	
8. Bezier Curves	10	
Total	84	

Initials:

1 Lighting

- **(3 points)**

  Which of the following statements is true for the Phong lighting model? Please check the corresponding box on the right.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

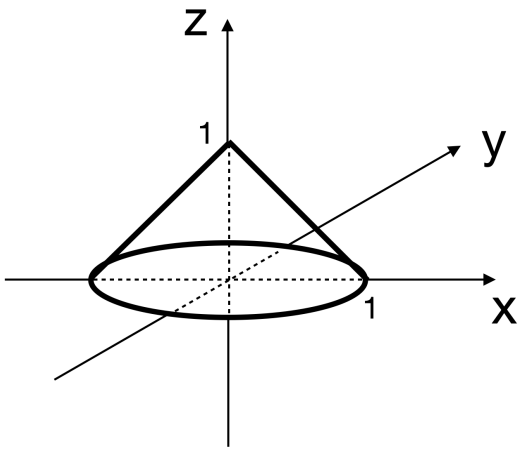
Statement	**True**	**False**
The diffuse component is independent of the position of the camera in the scene.		
The specular component assumes its highest value when the light vector reflected at the surface normal aligns with the eye vector.		
The diffuse component is independent of the position of the light source(s) in the scene.		

- **(3 points)**

  Assume you have a single directional light source and an orthographic camera. Explain why in this scenario, the Blinn-Phong lighting model is more efficient than the Phong lighting model.

Initials:

2 Raytracing



The image above shows a cone aligned with the z-axis. The apex (tip) of the cone is at $(0,0,1)$, the radius at the base in the xy plane is 1. We are also given a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o} = (o_x, o_y, o_z)$ is the origin of the ray and $\mathbf{d} = (d_x, d_y, d_z)$ is the ray direction with $\|\mathbf{d}\| = 1$.

- **(4 points)** Derive the implicit equation of the side surface of the cone (without the base).

Initials:

- **(4 points)** Derive an expression for the intersection of the ray $\mathbf{r}(t)$ with the base of the cone, including possible constraints on t to ensure the intersection is valid.

- **(4 points)** Derive an expression of the intersection of the ray $\mathbf{r}(t)$ with the sides of the cone, including possible constraints on t to ensure the intersection is valid.

Initials:

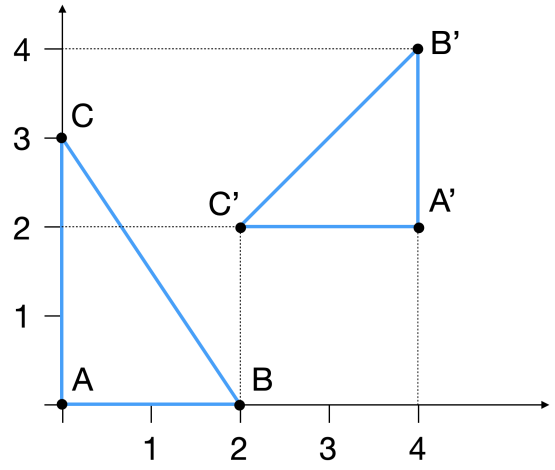
- **(3 points)** Derive the equation of the normal at a point with coordinates (x, y, z) on the surface of the side of the cone.

Initials:

3 Transformations and Projections

- **(3 points)**

  A triangle ABC is transformed by a transformation T to a triangle $A'B'C'$ as shown in the image below, i.e. A goes to A', B to B', and C to C'. Derive the transformation matrix for T in homogenous coordinates.



Initials:

- **(4 points)**

  Let A be a 2D transformation matrix given in homogenous coordinates as

$$A = \frac{1}{2}\begin{bmatrix} \sqrt{3} & 1 & 2 \\ -1 & \sqrt{3} & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

  Which of the following statements is true? Please check the corresponding box on the right.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

Statement	**True**	**False**
A linear combination of rotation matrices defines a rotation.		
The 2D transformation represented by A preserves area.		
Rotations are associative.		
The 2D transformation represented by A preserves angles.		

- **(3 points)** Which of the following matrices given in homogenous coordinates defines a projection? Please check the corresponding box below.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

Does the matrix define a projection?			
	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{d} \end{bmatrix}$
Yes			
No			

Initials:

- **(6 points)**

  Is the following statement true?

  Every 2D rotation can be expressed as a sequence of shear in x, followed by a shear in y, followed by a non-uniform scaling.

  Prove your answer!

Initials:

4 Textures

- **(7 points)**

  Which of the following statements is true? Please check the corresponding box on the right.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

Statement	**True**	**False**
Barycentric coordinates are preserved under perspective projection.		
Mipmapping reduces high-frequency noise but adds low-frequency blur.		
Textures coordinates are typically assigned at vertices and interpolated to the interior of a triangle.		
Bump mapping requires multi-pass rendering.		
Mipmapping doubles the required texture memory.		
Mapping a non-planar 3D surface to the plane always induces distortion in either angles or areas as measured on the surface.		
Environment maps produce the same reflections as raytracing.		

Initials:

5 GPU Pipeline

Vertex shader

```
// TODO: declarations missing

attribute vec3 position;
attribute vec3 normal;
attribute vec2 texcoord;

void main() {
    surface_position = (mat_mv * vec4(position,
        1.0)).xyz;
    surface_normal = mat_n_to_view * normal;
    uv = texcoord;

    gl_Position = mat_mvp * vec4(position, 1.0);
}
```

Fragment shader

```
precision mediump float;

// TODO: declarations missing

void main() {

    vec3 n = normalize_or_not_1(surface_normal);
    vec3 sp = normalize_or_not_2(surface_position);
    vec3 lp = normalize_or_not_3(light_position);

    vec3 light_direction = normalize(lp - sp);
    float diffuse_intensity = max(dot(n,
        light_direction), 0.0);

    vec2 tc = normalize_or_not_4(uv);
    vec3 surface_color = texture2D(tex_color, tc).rgb;

    vec3 color = surface_color * diffuse_intensity *
        light_color;
    gl_FragColor = vec4(color, 1.0);
}
```

Here is the pipeline shading a mesh with a single point light and a diffuse lighting model without distance attenuation.

- **(6 points) Declarations.** Complete the pipeline by filling in the declarations of variables in code blocks which will be included in the shaders above. Please write the declarations as actual code, for example make sure to specify the types of all variables. The type for textures is `sampler2D`.

Vertex shader

Fragment shader

Initials:

- **(4 points) Normalize or not?** Which of the `normalize_or_not` calls in the fragment shader should actually perform normalization? Consider that any normals or direction vectors passed from the CPU side have already been normalized. Unnecessary operations should be avoided for performance reasons.
  For each of the instances below, please write **Yes** if it should be normalized or **No** if the value should be unchanged; please give a short **reason** for whether the normalization should be performed or not.

```
vec3 n = normalize_or_not_1(surface_normal);
```

```
vec3 sp = normalize_or_not_2(surface_position);
```

```
vec3 lp = normalize_or_not_3(light_position);
```

```
vec2 tc = normalize_or_not_4(uv);
```

Initials:

6 Procedural Models

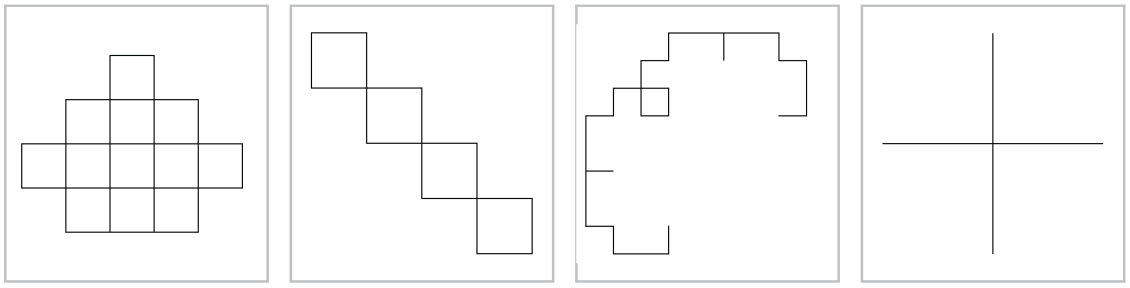
- **Fractals**: Assume we have a fractal in the plane constructed by the following recursive procedure: Take a unit square, split it into a grid of 3x3 equal squares and keep only the three squares on the diagonal from bottom left to top right. Repeat this process on each generated square.

  a) **(3 points)** What is the Haussdorff dimension of this fractal?

  b) **(4 points)** Give formulas for the area and boundary curve length of the above fractal as a function of recursion level k, where $k = 0$ is the single starting square. What do these expressions converge to as $k \to \infty$?

Initials:

- **L-Systems (8 points)**



A B C D

The figure above shows the turtle graphic output (scaled to fit into the frame) at recursion level 5 of four different L-Systems that all start with the same axiom F using a 90 degree turning angle for the $+$ and $-$ symbols. In the table below, assign the correct letter (A, B, C, D) to the rule defined in the left column.

Rule	**Figure Label**
$F \rightarrow -F+F-$	
$F \rightarrow -F++F$	
$F \rightarrow F+F-$	
$F \rightarrow -F+F+F$	

Initials:

7 Affine Combinations

- **(5 points)**

  Let **A**, **B**, **C** be points in 3D. The scalar t can assume any value in $[0,1]$. In affine geometry, which of the following expressions is valid, i.e., uniquely defines a point independent of the specific coordinate system chosen? Please check the corresponding box on the right.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

Expression	**Valid**	**Not Valid**
$\mathbf{A} - 2\mathbf{B}$		
$\cos(t)\mathbf{A} + \sin(t)\mathbf{B}$		
$2\mathbf{A} - \mathbf{B}$		
$\mathbf{A} - 2\mathbf{B} + \mathbf{C}$		
$(\mathbf{A} + \mathbf{B})/\mathbf{C}$		

Initials:

8 Bezier Curves

- **(4 points)**

  Which of the following statements is true? Please check the corresponding box on the right.

  (correct answer = 1 point, no answer = 0 points, wrong answer = -0.5 points)

Statement	**True**	**False**
Bezier curves interpolate all their control points.		
If two control points of a Bezier curve coincide, then the curve must have a sharp corner.		
If a Bezier curve has no self-intersections, then its control polygon has no self-intersections.		
If all control points of a Bezier curve lie on a line, then the Bezier curve is injective.		

Initials:

- **(6 points)** We want to represent the 2D curve of the function $y = x^2 - x + 1$ in the interval [0,1] with a Bézier curve. Determine the **exact** coordinates of the control points.

  $b_0 =$

  $b_1 =$

  $b_2 =$

Initials:

Initials: