**Student Number:** **Name:**

| |
|---|
| **Write down answers in-between questions. Please answer using short sentences.**<br><br>**The back of each page can be used for practice, but DO NOT write down the answer on the back.**<br><br>**Be sure to write your student number and name on each page.** |

1. (1 pt each) True / False questions:

   1) An object appears red because it absorbs red light and reflects green and blue. ( )
   2) The correct way to transform normal vectors under a model matrix M is to apply the same M directly to them. ( )
   3) According to Euler's rotation theorem, any rotation in 3D space with one point fixed can be represented as a single rotation about some axis through that point. ( )
   4) Rotation matrices are orthogonal matrices with a determinant of 0. ( )
   5) Link transformations in a skeleton are typically time-varying. ( )
   6) Explicit curve representations such as y=f(x) can represent circles and vertical lines. ( )
   7) A Bezier curve always passes through all its control points. ( )
   8) In Hermite curves, the positions and second derivatives at endpoints are required as constraints. ( )
   9) glGenerateMipmap() automatically creates all mipmap levels from the base texture. ( )
   10) Measured BRDFs can be used in rendering to simulate realistic surface appearances. ( )

2. (8 pts) Fill in the blanks:

   Gouraud shading calculates lighting at each ____**(a)**____ and interpolates ____**(b)**____ across the surface.

   Phong shading interpolates ____**(c)**____ and calculates lighting at each ____**(d)**____.

**Student Number:** **Name:**

3. (8 pts) The following figure shows the directions that influence lighting at a point on the surface. Note that **L, N, V,** and **R** are unit vectors.

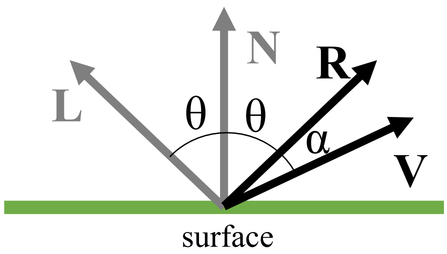
   $\mathbf{l}_d$ : light diffuse color, $\mathbf{m}_d$ : material diffuse color,

   $\mathbf{l}_s$ : light specular color, $\mathbf{m}_s$ : material specular color,

   **L** : light direction, **N** : surface normal,

   **V** : view direction, **R** : reflection direction of light

   n : shininess coefficient




   1) Fill in the blanks (a) and (b) in the formulas below to compute the diffuse and specular colors $\mathbf{I}_d$ and $\mathbf{I}_s$, based on the Phong illumination model. The * symbol represents element-wise multiplication.

   $\mathbf{I}_d = \mathbf{l}_d * \mathbf{m}_d$ ____**(a)**____

   $\mathbf{I}_s = \mathbf{l}_s * \mathbf{m}_s$ ____**(b)**____

   2) To increase the size of the object's specular highlight, which of the symbols above should be adjusted, and how should it be changed?

4. (4 pts) Which of the following correctly describes the role of the ambient term in the Phong illumination model?

   1) It simulates direct light reflected off smooth surfaces.

   2) It adds a constant color to severely approximate indirect light from the environment.

   3) It sharpens the specular highlight for shiny surfaces.

   4) It is used only when normal mapping is applied.

5. (6 pts) Fill in the blanks to complete the SLERP formula that interpolates between two rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$, using a parameter t∈[0,1].

   slerp($\mathbf{R}_1$, $\mathbf{R}_2$,t)= ____**(a)**____ · exp (____**(b)**____ · log(____**(c)**____))

**Student Number:** **Name:**

6. (6 pts) There are two objects, object1 and object2, and vertex array objects vao1 and vao2 to draw them. vao1 refers to a vertex buffer object that stores vertex positions and vertex colors, while vao2 refers to a vertex buffer object that stores vertex positions and vertex normals. Fill in the blanks (a)~(f) using the shader source code variable names **shader1**, **shader2**, **shader3**, **shader4**, and the shader program variable names **program1**, **program2** below to enable proper rendering using vao1 and vao2. Note that the load_shader() function takes the vertex shader code as the first argument and the fragment shader code as the second argument and returns a shader program object. '...' indicates omitted code, assuming that the relevant parts are appropriately written.

```
shader1 = '''
#version 330 core
layout (location = 0) in vec3
vin_pos;
layout (location = 1) in vec3
vin_normal;
out vec3 vout_surface_pos;
out vec3 vout_normal;
...
void main()
{
    gl_Position = ...
    vout_surface_pos = ...
    vout_normal = ...
}
'''

shader2 = '''
#version 330 core
in vec4 vout_color;
out vec4 FragColor;
void main()
{
    FragColor = vout_color;
}
'''
```

```
shader3 = '''
#version 330 core
in vec3 vout_surface_pos;
in vec3 vout_normal;
out vec4 FragColor;
...
void main()
{
     FragColor = ...
}
'''

shader4 = '''
#version 330 core
layout (location = 0) in vec3
vin_pos;
layout (location = 1) in vec3
vin_color;
out vec4 vout_color;
...
void main()
{
    gl_Position = ...
    vout_color = ...
}
'''
```

```
def main():
    ...
    program1 = load_shader(__(a)__, __(b)__)
    program2 = load_shader(__(c)__, __(d)__)
    ...
    while not glfwWindowShouldClose(window):
        ...
        glUseProgram(__(e)__)
        glBindVertexArray(vao1)
        glDrawArrays(...)

        glUseProgram(__(f)__)
        glBindVertexArray(vao2)
        glDrawArrays(...)
```

**Student Number:** **Name:**

7. (5 pts) Choose ALL correct statements about 3D rotation representations.

   1) Euler angles provide continuous and unique representation, and are ideal for interpolation.

   2) Quaternions support smooth interpolation but are difficult to visualize.

   3) Rotation matrices use minimal parameters and are well suited for interpolation.

   4) Rotation vectors represent rotations with three parameters and can be derived from rotation matrices using logarithmic mapping.

8. (6 pts) The following Node class represents a node in a hierarchical model defined by link and joint transformations and builds a tree structure. Fill in the blank (a), (b), (c) to complete `update_tree_global_transform()` that calculates and updates the global transformation of a node and its all descendants.

```
class Node:
    def __init__( self, parent, link_transform_from_parent,
shape_transform, color):
        self.parent = parent
        self.children = []
        if parent is not None:
            parent.children.append(self)

        self.link_transform_from_parent = link_transform_from_parent
        self.joint_transform = glm.mat4()
        self.global_transform = glm.mat4()
        self.shape_transform = shape_transform
        self.color = color

    def set_joint_transform(self, joint_transform):
        self.joint_transform = joint_transform

    def update_tree_global_transform(self):
        if self.parent is not None:
            self.global_transform = __(a)__
        else:
            self.global_transform = __(b)__

        for child in self.children:
            __(c)__
```

**Student Number:** **Name:**

9. (6 pts) The following snippet represents part of the HIERARCHY section of a BVH file. Assume that the character is in the rest pose. The offset from joint J2 to joint J3 is given as (0.25, 0.5, 0.0). Fill in the blank (a) and (b) so that the global position of joint J3 becomes (0.75, 1.0, 0.0) under the given conditions.

```
HIERARCHY
ROOT J0
{
  OFFSET  0  0  0
  CHANNELS 6 Xposition Yposition Zposition Xrotation Yrotation
Zrotation
  JOINT J1
  {
    OFFSET   0.25 0.5 0.0
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT J2
    {
      OFFSET   _____(a)_____
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT J3
      {
        OFFSET   _____(b)_____
        CHANNELS 3 Zrotation Xrotation Yrotation
        End Site
        {
          OFFSET   0  0  0.1
}}}}}
```

10. (5 pts) Choose **ALL false (incorrect)** statement about computer animation.

1) In key frame animation, animators just specify important events at key frames and computer fill in the remaining frames.

2) In key frame animation, *"Motion"* is controlled by setting keyframes and manipulating various parameter curves.

3) It's very easy and intuitive to create "realistic" and "physically plausible" motions using keyframe animation methods.

4) Motion capture system "captures" movement of people or objects.

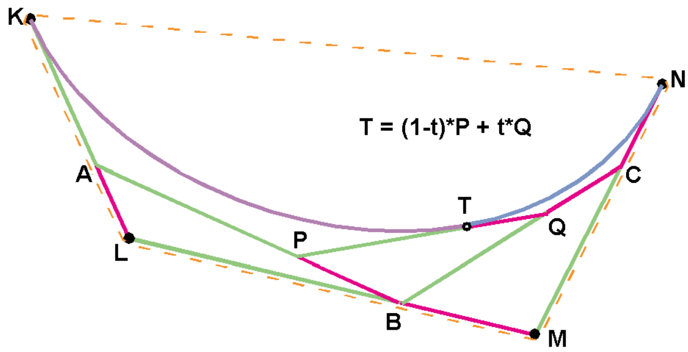
5) In motion capture data, *"skeleton"* is typically created manually.

6) Physically-based simulation can be used to generate high-quality computer animation because physical reality plays a key role in animation quality.

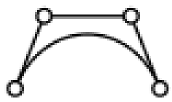
**Student Number:** **Name:**

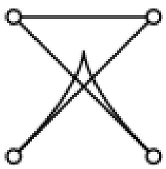
11. (6 pts) Consider a cubic polynomial curve defined on a 1D space: $p(t)=at^3+bt^2+ct+d$.
Suppose the following constraints are given: $p(0)=1$, $p(1)=4$, $p'(0)=0$, $p'(1)=1$.
Compute the values of the coefficients a, b, c, and d that satisfy the given constraints.

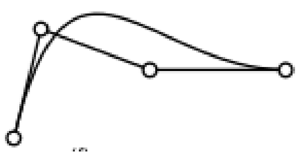
12. (6 pts) The right figure illustrates how the de Casteljau's Algorithm generates a Bezier curve, describing the relationship between a point T on the curve at time t and points P and Q. Write the equation that describes the relationship between the point T and points A, B, and C.

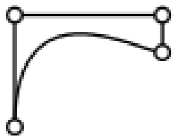



13. (5 pts) Below are four curves and their "control points/polygon." Some of the control points are the Bezier control points for the curve drawn with it (i.e., the curve in this case is a Bezier curve); the others are not. Choose **ALL non-Bezier** curve & control polygon. Assume that none of the control points overlap or are repeated.

a. 

b. 

c. 

d. 

**Student Number:** **Name:**

14. (8 pts) Choose the correct options for each blank.
OpenGL provides several texture wrapping options to determine how UV coordinates outside the [0,1] range are handled. The default is **__(a)__** which simply repeats the image. An alternative, **__(b)__**, mirrors the image at each boundary. To use two textures (e.g., diffuse and specular), we assign each sampler2D a texture unit using **__(c)__**, and activate it with **__(d)__** before binding the texture object.

(a) ☐ GL_CLAMP_TO_EDGE ☐ GL_REPEAT ☐ GL_MIRRORED_REPEAT
(b) ☐ GL_CLAMP_TO_BORDER ☐ GL_MIRRORED_REPEAT ☐ GL_LINEAR
(c) ☐ glActiveTexture ☐ glTexParameteri ☐ glUniform1i
(d) ☐ glUniform1i ☐ glActiveTexture ☐ glBindTexture

15. (6 pts) At a point p on a sphere, the surface normal is aligned with the w-axis of the local tangent space. A tangent-space normal map is used to encode the surface normals of the sphere, and the RGB channels of the normal map represent the u, v, and w components of the normal vector, respectively. Each channel stores values in the range [0, 255], corresponding to a vector component range of [-1, 1]. What should be the RGB color value stored in the normal map at the UV coordinate corresponding to point p?

16. (5 pts) Choose **ALL false (incorrect)** about local and global illumination models.

1) Local illumination models only consider light that comes directly from light sources, while global illumination models simulate indirect light interactions between object surfaces.

2) Local illumination models simulate all light interactions within a limited area, while global illumination models consider the entire scene.

3) Local illumination models are usually faster to compute than global illumination models.

4) Local illumination models only consider diffuse reflection, while global illumination models consider both diffuse and specular reflection.