

Exam – Computer Graphics

19 october 2009, 8-13

Answers may be given in either Swedish or English
Electronic calculator allowed

- 1 (a) How is a *similarity transform* different from a *rigid body transform*. Give an as exact answer as you can. (0.7)
(b) Explain why and when *transformation of normals* requires special attention. Also describe how it is to be done correctly. (0.3)
- 2 Explain the following interpolation concepts:
 - Bilinear interpolation (0.3)
 - Perspective correct interpolation (0.3)
 - Catmull-Rom interpolation (0.4)
- 3 (a) Give the formula for *Phong shading* and give definitions for all terms used. (0.4)
(b) In what respects is Phong shading an approximation to the *rendering equation*. (0.4)
(c) Explain how *Blinn specular* can be more efficient than *Phong specular*. (0.2)
- 4 (a) What is *radiosity*? (0.3).
(b) What is *Perlin noise*? What can it be used for? (0.3)
(c) Explain the term *skeleton animation*. (0.4)
5. (a) What is drawn on the screen after a call to the function *draw()* below? (1.0)

```
def draw():
    glColor(1, 0, 0)
    glPushMatrix()
    glTranslate(0, -4, 0)
    glScale(1, 2, 1)
    drawSquare()

    glColor(0, 1, 0)
    glTranslate(2, 1, 0)
    glPushMatrix()
    glRotate(180, 0,0,-1)
    glPushMatrix()
    glTranslate(-2, -1, 0)
    glPopMatrix()
    glPushMatrix()
    glRotate(90, 0,0,-1)
    glScale(2, 2, 1)
    drawSquare()

    glColor(0, 0, 1)
    glPopMatrix()
    glPushMatrix()
    glRotate(180, 0,0,1)
    glTranslate(-2, 2, 0)
    drawSquare()

def drawSquare():
    glBegin(GL_QUADS)
    glVertex(0,0,0)
    glVertex(0,1,0)
    glVertex(1,1,0)
    glVertex(1,0,0)
    glEnd()
```

- 6 (a) A point has coordinates $(-4.0, 80.0, -16.0)$ in the coordinate system of the camera. What are the corresponding normalized device coordinates if the camera uses perspective projection and the distance to the image plane (the zoom factor) is 3.0? (0.6)
- (b) Is the point visible on a screen with a size of $(640,480)$ pixels assuming that device positions are mapped according to the smallest dimension of the screen.? If so, where? (0.4)

THE END!