

Exam – Computer Graphics

17 august 2007, 8-13

- 1 (a) What is the most efficient way to calculate the inverse of a rotation matrix? (0.3)
(b) What is the most important advantage of representing transforms as matrices? (0.4)
(c) How is this advantage utilized in the design of a renderer? (0.3)
- 2 (a) Explain how rasterization of triangle is done. (0.5)
(b) Explain how the shading of a rasterized triangle is done? (0.5)
- 3 (a) What is the visual difference between *flat shading* and *smooth shading*. (0.3)
(b) How do their implementations differ? (0.7)
4. (a) What is a *cube map* and what is it used for? (0.2)
(b) How do you do lookup in a *cube map*? (0.8)
- 5 (a) Describe what is drawn on the screen after a call to the function *draw()* below. (0.8)

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

    glColor(0, 1, 0)
    glTranslate(2, 3, 0)
    glPushMatrix()
    glRotate(90, 0,0,-1)
    glPushMatrix()
    glRotate(180, 0,0,1)
    glTranslate(-2, -1, 0)
    glPopMatrix()
    glPushMatrix()
    glScale(2, 2, 1)
    drawSquare()

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

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

- (b) OpenGL has a concept called *display list*? What is it and which problem does it solve? (0.2)

- 6 (a) Which relationships does the *rendering equation* express? (0.6)
(b) Write down the equation and explain what each of its constituents means. Answer as detailed as you can (0.4)

THE END!