

Assignment 3, Due Thu. 26 November 2009, 9.30am. Worth 33% of CA. Demonstrations of software execution, see below, must be completed by Thu. 26 November 2009, 11.00am

Answers should be in your own words. For numerical answers, where appropriate, show your working; a numerical answer written down with no explanation of working is likely to get zero marks.

Late assignments. 10% of marks will be deducted for each day late; assignments seven days late will receive 0%.

Computing Department Rules: 1. Assignments must be neatly typed and printed; 2. A cover sheet and declaration must be included.

Relevant sections of code (with comments as necessary) should be submitted.

1. GLSL shaders. Use my sample program `gls1.cpp` and shaders `toonf2.vert` and `toonf2.frag` as your framework. Copy them to `myprog.vert` and `myprog.frag` respectively. If you find my notes a bit light, have a look at the OpenGL SuperBible and the shader examples that come with that. 50% of marks for demonstration of some parts of this question, at which you should be prepared to discuss your program.

(a) Revise `myprog.vert` to scale vertex coordinates as follows: `x, 1.5, y, 2.0, and z, 0.5`. In the corresponding fragment shader, use the following, i.e. the display is in monochrome:

```
color = vec4(intensity,intensity,intensity,1.0);  
//color = 1.0 -color; // comment out
```

[20 + 20 = 40 marks]

(b) Revise `myprog.frag` to render the scene in *sepia* monochrome. You will have to find out the RGB makeup of sepia.

```
vec4 sepia = vec4(?, ?, ?, 1.0);
```

[20 + 20 = 40 marks]

(c) Explain in detail what `gl_Position = ftransform();` does. Rewrite it in two different ways, each explicitly mentioning `gl_vertex`.

[20 marks]

2. Articulated figures (Chapter 14). 50% of marks for demonstration of some parts of this question, at which you should be prepared to discuss your program.

(a) Add a single finger to one of the lower arms in `articRobot.cpp`; remember to change `ncomp` as needed. You should be able to demonstrate rotating the finger.

[30 + 30 = 60 marks]

(b) Add eyes (black spheres) to `articRobot.cpp`; remember to change `ncomp` as needed.

[30 + 30 = 60 marks]

(c) Draw the *tree* corresponding to (b).

[20 marks]

- (d) Give a detailed explanation of how `traverse` works; in particular, you should mention the role of `glPushMatrix()` and `glPopMatrix()` and concatenation of transformations in:

```
glPushMatrix();
    traverse(t, it);
// restore transformation of higher node
glPopMatrix();
```

Hint: it may help to refer to the tree corresponding to the figure.

[20 marks]

3. Animated articulated figures. 50% of marks for demonstration of some parts of this question, at which you should be prepared to discuss your program. I have extended the software in Chapter 14 to incorporate animation, see directory `animArticRobot`; based these answers on that.

- (a) Based on the swinging arm animation in `animArticRobot.cpp`, add animations to the other arm and to the legs (upper legs) to create a *walking* animation.

[30 + 30 = 60 marks]

- (b) In (a) the lower limbs do not rotate independently of the upper limbs. Add animation to the lower limbs to produce a more realistic walking animation. Get someone to walk in a stylised manner and see if you can come up with the animation angles. I think the animations need to be synchronised, i.e. the lower limb animations should follow the same time cycle as the upper limbs — but not the same set of angles.

[30 + 30 = 60 marks]

4. Animated articulated figures. 50% of marks for demonstration of some parts of this question, at which you should be prepared to discuss your program. Based on `animArticRobot.cpp`, create an animated articulated figure of your own. For example, simple four legged animal shape, simple vehicle, etc.

[50 + 50 = 100 marks]