Harry Langford
hjel2@cam.ac.uk

Paper 3 Question 4

(a) Modelling transformations transform an object from object coordinates (where the object is at the origin and has an arbitrary size) into world coordinates with the rest of the scene.

Viewing transformations change an object from world coordinates into viewing coordinates such that the camera is at the origin and is pointing down the positive z-axis.

Projection transformations project all objects in a scene onto a given plane. This is usually $z = d$ for some constant d. They usually project z to $\frac{1}{z}$ rather than d so that this can be used to calculate the distance to the object.

(b) (i) If an object was moving then the modelling transformation would change.

(ii) If an object in the scene was changing shape then the modelling transformation would change.

(iii) Light bending due to the extreme gravity of a black hole would affect all shapes in the same way – assuming that the shapes were near to each other. In this case we would change the viewing transformation.

(c) We would compute the following in the GPU:

```
in vec2 xy;
GLposition = vec3(xy, f(xy.x, xy.y))
```

The return from this could be taken as a vec3[].

```
in vec3[] xyz;
out xy;
int min = vec3(0, 0, float("inf"));
for each in xyz:
        if each.z < min.z:
                min = each;
xy = min.xy;
```

The z-buffer would essentially be used to compute $f(x, y)$ in parallel on the GPU and then we could find the minimum in series (finding the minimum could be parallelised using a recursive algorithm to $\Theta(\lg n)$ however that is not relevant to the z-buffer).

(d) Bump mapping finds the normal at a point and offsets it slightly according to a "bump map". Which is a 2d array indicating whether the location is raised or inset. This allows the user to map a texture onto an object making it appear to have an uneven surface.

Some people may prefer displacement mapping because displacement mapping changes the **shape** of the object rather than just appearing to add insets and raised portions. For example bump mapping could not be used to model a sphere with a hole cut out of it – while displacement mapping could.

(e) we triangularise 3D objects because it removes all ambiguity about where lines on any non-coplanar points may go.

Triangles allow easy baryocentric interpolation of colours in the vertex shader.

GPU's are optimised to deal with trianges (as opposed to arbitrary-sided n-gons).

Triangles are simpler shapes and it requires far simpler maths to do operations on trianlges than on n-gons.