# Solutions, CSci 490, Spring 2005, Midterm

**1.** A raster graphics system is one based on images rendered in discrete units called pixels, whereas a vector graphics system is based on continuous primitives, such as lines and circles. Raster graphics systems include nearly all modern computer monitors and printers; and most popular image representations use it (GIF, JPEG, PNG, among others). Vector graphics systems include plotters; many programming APIs; the Postscript language; most modern font representations.

**2.** The homogeneous coordinate system adds a fourth coordinate to the regular coordinate system, whose value will typically be 1, but which may be something else to represent a factor by which to scale the other coordinates. They are useful in the graphics pipeline for two reasons: They allow translations to be represented in matrix form with other common transformations; and they allow perspective transformations to end up representing the needed scaling based on depth. [Either of these reasons would be OK.]

**3.**
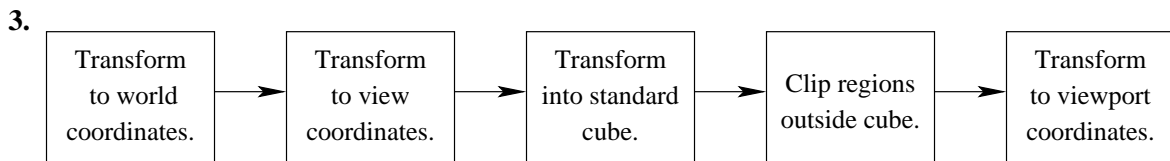


**4.**

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**5.** Curved surfaces are most easily represented in most systems with a polygon approximation, but rendering the polygon surfaces shows sharp shading differences if the polygons are relatively large, destroying the illusion of curves. Gouraud shading strives to smooth out the shading by interpolating the color of each pixel based on the colors computed for the various vertices of each polygon.

**6.** The idea of Bresenham's algorithm is to avoid the floating-point arithmetic involved in maintaining `error`. We can do this easily by scaling `error` by a factor of 2`W`.

```
public static void drawLine(int x0, int y0, int x1, int y1) {
    int W = x1 - x0;
    int H = y1 - y0;
    int error = 0;
    int y = y0;
    for(int x = x0; x <= x1; x++) {
        drawPixel(x, y);
        error += 2 * H;
        if(error > W) {
            y++;
            error -= 2 * W;
        }
    }
}
```

[Traditionally, Bresenham's algorithm also involves translating `error` downward by `W` units, so that the `if` statement involves evaluating whether an integer is positive, which is easier than comparing it to another integer. This change, though, is less significant.]

**7.** Bump mapping is the usage of a texture for affecting the shading seen for a surface. The texture is used to permute the surface normal as it would normally be computed. It is useful for bumpy surfaces such as brick walls and orange peels.

**8.** We saw four techniques in class:

- Projecting all polygons onto the floor's plane. This is simple and quick, but it projects shadows only onto the floor, not onto other surfaces.
- Computing texture maps for surfaces based on shadows cast by other objects. Again, this is relatively quick, but it projects shadows only onto surfaces for which the texture map is computed.
- Casting rays from each intersection point to each light source and seeing whether any polygons fall on the ray. This works correctly, but it is quite slow.
- Pre-computing *shadow maps* for each light source representing the depth for each pixel, and using this to determine whether each intersection point is blocked by another polygon. This works correctly most of the time, but it raises issues with the resolution of the shadow map (rays will not hit a point in the map precisely) and with the resolution of the computed depths.

**9.** Diffuse reflection is light that is reflected off from a surface in all directions; since most objects absorb some light frequencies and reflect others, we generally undersand the diffuse reflection to represent the object's "color." Specular reflection is a more direct reflection of light off the surface, so that the object acts as a very dull mirror; specular reflections generally reflect the light's color more than the object's color, and they are more concentrated.

**10.** In the real world, surfaces are lit not only by light coming directly from light sources, but also by light reflected off of other surfaces (both diffusely and specularly). Ambient light is a crude approximation to these secondary reflections of light, by simply adding a constant amount of light to every surface in the scene.