OpenGL/VRML Materials

These numbers come from the OpenGL teapots.c demo, ï¿½ Silicon Graphics, Inc., ï¿½ 1994, Mark J. Kilgard. See also [1], [2], and [3].

The numbers

Name	Ambient			Diffuse			Specular		
emerald	0.0215	0.1745	0.0215	0.07568	0.61424	0.07568	0.633	0.727811	0.633
jade	0.135	0.2225	0.1575	0.54	0.89	0.63	0.316228	0.316228	0.316228
obsidian	0.05375	0.05	0.06625	0.18275	0.17	0.22525	0.332741	0.328634	0.346435
pearl	0.25	0.20725	0.20725	1	0.829	0.829	0.296648	0.296648	0.296648
ruby	0.1745	0.01175	0.01175	0.61424	0.04136	0.04136	0.727811	0.626959	0.626959
turquoise	0.1	0.18725	0.1745	0.396	0.74151	0.69102	0.297254	0.30829	0.306678
brass	0.329412	0.223529	0.027451	0.780392	0.568627	0.113725	0.992157	0.941176	0.807843
bronze	0.2125	0.1275	0.054	0.714	0.4284	0.18144	0.393548	0.271906	0.166721
chrome	0.25	0.25	0.25	0.4	0.4	0.4	0.774597	0.774597	0.774597
copper	0.19125	0.0735	0.0225	0.7038	0.27048	0.0828	0.256777	0.137622	0.086014
gold	0.24725	0.1995	0.0745	0.75164	0.60648	0.22648	0.628281	0.555802	0.366065
silver	0.19225	0.19225	0.19225	0.50754	0.50754	0.50754	0.508273	0.508273	0.508273
black plastic	0.0	0.0	0.0	0.01	0.01	0.01	0.50	0.50	0.50
cyan plastic	0.0	0.1	0.06	0.0	0.50980392	0.50980392	0.50196078	0.50196078	0.501960
green plastic	0.0	0.0	0.0	0.1	0.35	0.1	0.45	0.55	0.45
red plastic	0.0	0.0	0.0	0.5	0.0	0.0	0.7	0.6	0.6
white plastic	0.0	0.0	0.0	0.55	0.55	0.55	0.70	0.70	0.70
yellow plastic	0.0	0.0	0.0	0.5	0.5	0.0	0.60	0.60	0.50
black rubber	0.02	0.02	0.02	0.01	0.01	0.01	0.4	0.4	0.4
cyan rubber	0.0	0.05	0.05	0.4	0.5	0.5	0.04	0.7	0.7
green rubber	0.0	0.05	0.0	0.4	0.5	0.4	0.04	0.7	0.04
red rubber	0.05	0.0	0.0	0.5	0.4	0.4	0.7	0.04	0.04

white rubber	0.05	0.05	0.05	0.5	0.5	0.5	0.7	0.7	0.7
yellow rubber	0.05	0.05	0.0	0.5	0.5	0.4	0.7	0.7	0.04

How to use it

OpenGL

Multiply the shininess by 128!

```
  mat[0] = ambr;
  mat[1] = ambg;
  mat[2] = ambb;
  mat[3] = 1.0;
  glMaterialfv(GL_FRONT, GL_AMBIENT, mat);
  mat[0] = difr;
  mat[1] = difg;
  mat[2] = difb;
  glMaterialfv(GL_FRONT, GL_DIFFUSE, mat);
  mat[0] = specr;
  mat[1] = specg;
  mat[2] = specb;
  glMaterialfv(GL_FRONT, GL_SPECULAR, mat);
  glMaterialf(GL_FRONT, GL_SHININESS, shine * 128.0);
```

VRML97

Compute ambientIntensity as (0.212671*ambr + 0.715160*ambg + 0.072169*ambb)/(0.212671*difr + 0.715160*difg + 0.072169*difb)

```
Material {
  ambientIntensity  amb
  diffuseColor      difr digg difb
  specularColor     specr specg specb
  shininess         shine
}
```