# 6. Lighting and Transparency

COMP3421 Computer Graphics • KC Notes

## 6.1 Colouring

* Implement colour by having separate red, green and blue components for
  + Light intensities
  + Reflection coefficients
  + Treat intensities as vectors and multiplication becomes **component-wise**, i.e.
* Still add all components up
  + Using too much light will cause colour to become 1 – becomes saturated

## 6.2 Transparency

* Let light through from the object behind it – use rgba (alpha – 1 is opaque, 0 is transparent)
* **Alpha blending**: blend colours according to their alpha value
  + Linear interpolation, where **image is the new item’s colour**:

|  |  |
| --- | --- |
| // Alpha blending is disabled by default. To turn it on in init()  gl.glEnable(GL.GL\_BLEND);  gl.glBlendFunc(GL.GL\_SRC\_ALPHA, GL.GL\_ONE\_MINUS\_SRC\_ALPHA);  // other blend functions are also available | |
|  |  |

* Transparency doesn’t work in terms of the depth buffer, e.g.
  + Ordering is still important
  + Draw transparent front item, then draw opaque back item – because of the depth buffer, item will not draw on top and it will not show up behind the transparent item
  + **Turn off depth buffer temporarily** gl.glDepthMask(false)