CS 418: Interactive Computer Graphics Spring 2018
In-class Worksheet: Compositing

Alpha Blending

The Over Operator

Compositing of two colors with alpha values can be accomplished using the over operator:

$$C_{A \text{ over } B} = \alpha_A C_A + (1 - \alpha_A)\, \alpha_B C_B$$
$$\alpha_{A \text{ over } B} = \alpha_A + (1 - \alpha_A)\, \alpha_B$$

1. Blending

Suppose C_A=(0.5, 0.5, 1.0, 0.25)and C_B=(0.25, 0.25, 0.25, 0.75)

Compute C_A over C_B assuming NON-pre-multiplied alpha.

2. Pre-multiplied Alpha

a. If CA=(0.125, 0.125, 0.5, 0.5) uses pre-multiplied alpha, what is the equivalent color using non-pre-multiplied alpha?

b. In WebGL, which gl.blendFunc parameters correspond to using pre-multiplied alpha in your blending? You can consult the table on the back for a list of blending parameters.

3. Alpha Blending in WebGL

Alpha blending in WebGL most commonly uses the function call: `gl.blendFunc(GLenum sfactor, GLenum dfactor)` where the final color is generated by as Cs_over_d = sfactor*Cs + dfactor*Cd

Possible values for sfactor and dfactor are:

FUNCTION	RGB BLEND FACTORS	ALPHA BLEND FACTOR
gl.ZERO	(0, 0, 0)	0
gl.ONE	(1, 1, 1)	1
gl.SRC_COLOR	(R_s, G_s, B_s)	A_s
gl.ONE_MINUS_SRC_COLOR	$(1, 1, 1) - (R_s, G_s, B_s)$	$1 - A_s$
gl.DST_COLOR	(R_d, G_d, B_d)	A_d
gl.ONE_MINUS_DST_COLOR	$(1, 1, 1) - (R_d, G_d, B_d)$	$1 - A_d$
gl.SRC_ALPHA	(A_s, A_s, A_s)	A_s
gl.ONE_MINUS_SRC_ALPHA	$(1, 1, 1) - (A_s, A_s, A_s)$	$1 - A_s$
gl.DST_ALPHA	(A_d, A_d, A_d)	Ad
gl.ONE_MINUS_DST_ALPHA	$(1, 1, 1) - (A_d, A_d, A_d)$	$1 - A_d$

Suppose alpha blending is enabled using the over operator and the following fragments are processed in the given order for the same pixel location. What is the final color?
Assume:

1. Framebuffer is set to (0,0,0,1) at all pixel locations.
2. Depth testing is on
3. We are using pre-multiplied alpha

Fragment 1: (0.5, 0.5, 0.5, 0.25) with a depth value of 0.25
Fragment 2: (0.25, 0.25, 0.25, 0.25) with a depth value of 0.5
Fragment 3: (0.125, 0.125, 0.125, 0.25) with a depth value of 0.9
(Hint: WebGL depth buffer assumes smaller being closer to the viewer)