Test 1

CS 5610/6610

Interactive Computer Graphics

Spring 2015

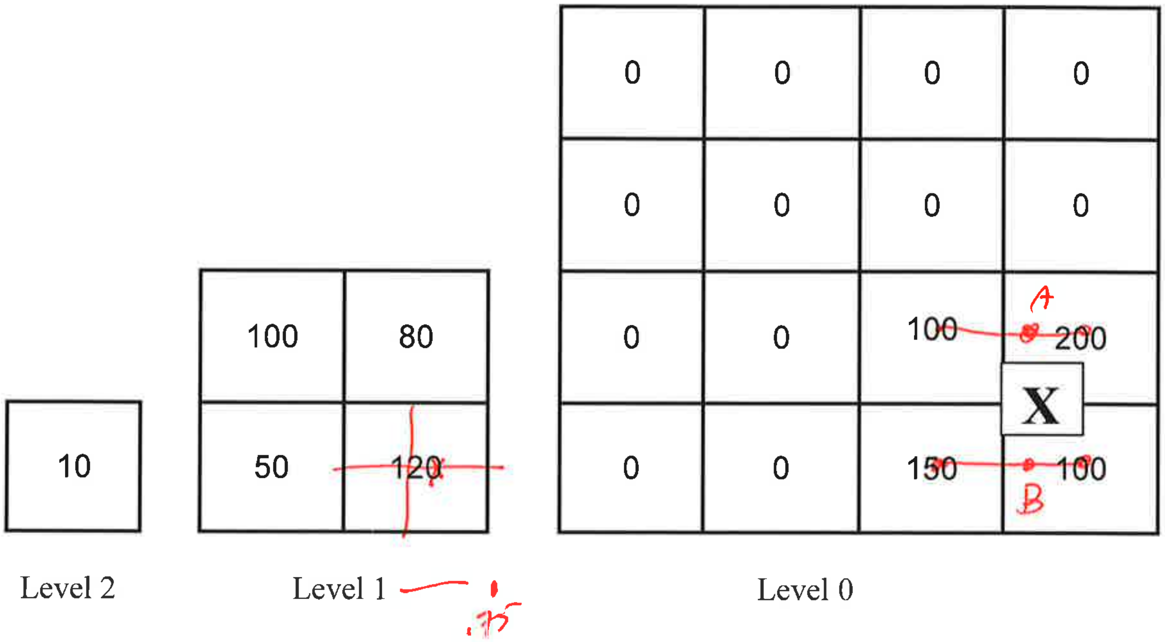
Name: Solution

Student ID: ______________________

Rules:

1. CS 5610 students, answer any 4 questions (no extra credit)
2. CS 6610 students, answer all 6 questions

Open Book/notes



Given the above luminance texture.
Counting from zero at **Level 0**,
a fragment's center falls at the X (75% away from the left-most texel).
The texel values shown are at the center of the texels (as shown)

It's projection is **d=0.75**.

1a. [10 pts] What is the assigned value of a fragment with:
minification filter set at: GL_LINEAR
magnification filter set at: GL_LINEAR

143.75

A: $.25(100) + .75(200)$
175
B: $.25(150) + .75(100)$
112.5
$X = .5(175) + .5(112.5)$
143.75

1b. [10 pts] What is the final luminance value:
minification filter set at: GL_LINEAR_MIPMAP_NEAREST
magnification filter set at: GL_LINEAR

.120

Table 4.3 Source and Destination Blending Factors

Constant	RGB Blend Factor	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)	A_d
GL_ONE_MINUS_DST_ALPHA	$(1, 1, 1) - (A_d, A_d, A_d)$	$1 - A_d$
GL_CONSTANT_COLOR	(R_c, G_c, B_c)	A_c
GL_ONE_MINUS_CONSTANT_COLOR	$(1, 1, 1) - (R_c, G_c, B_c)$	$1 - A_c$
GL_CONSTANT_ALPHA	(A_c, A_c, A_c)	A_c
GL_ONE_MINUS_CONSTANT_ALPHA	$(1, 1, 1) - (A_c, A_c, A_c)$	$1 - A_c$
GL_SRC_ALPHA_SATURATE	$(f, f, f), f = \min(A_s, 1 - A_d)$	1
GL_SRC1_COLOR	(R_{s1}, G_{s1}, B_{s1})	A_{s1}
GL_ONE_MINUS_SRC1_COLOR	$(1, 1, 1) - (R_{s1}, G_{s1}, B_{s1})$	$1 - A_{s1}$
GL_SRC1_ALPHA	(A_{s1}, A_{s1}, A_{s1})	A_{s1}
GL_ONE_MINUS_SRC1_ALPHA	$(1, 1, 1) - (A_{s1}, A_{s1}, A_{s1})$	$1 - A_{s1}$

Given the following:

Fragment: (R,G,B,A)=(1.0, 0.0, 1.0, 0.5)
Framebuffer initial values: (0.8, 1.0, 0.8, 0.75)

Assume blending is enabled and the state is correctly setup.
What is the result of the following:

2a. [10 pts] RGBA blend with:
GlBlendFunc(GL_ONE_MINUS_DST_COLOR, GL_SRC_ALPHA)?

Frag * 1-FBcolor   FB * Frag α

SRC ~~1.0 0 1.0~~   Dst

R	G	B	α
(1.0)(.2)	0.0(0.0)	1.0 (.2)	.5(.25)
.2	0.0	.2	.125

R	G	B	α
.8(.5)	1.0(.5)	.8(.5)	.75(.5)
.4	.5	.4	.375

+

.6	.5	.6	.5
R	G	B	α

2b. [10 pts] RGBA blend with:
GlBlendFunc(GL_DST_COLOR, GL_SRC_ALPHA)?

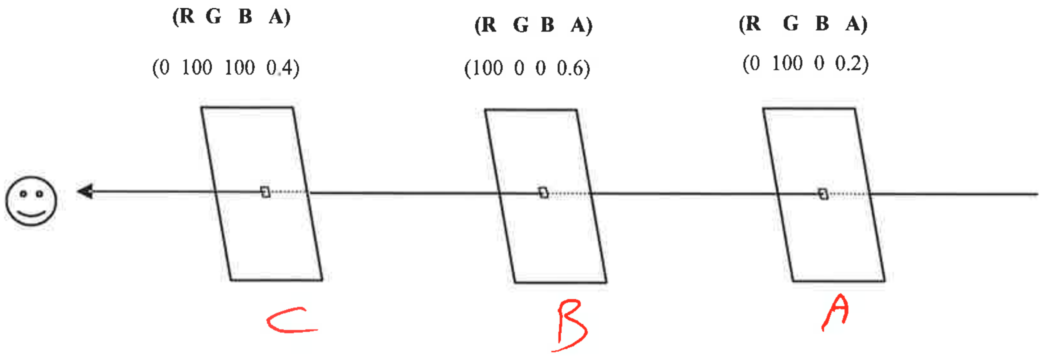
Frag * FBcolor   FB * Frag α

R	G	B	α
(1.0)(.8)	0.0(1.0)	1.0(.8)	.5(.75)
.8	0	.8	.375

R	G	B	α
.4	.5	.4	.375

+

R	G	B	α
1.0	.5	1.0	.75



The back-to-front compositing equation is:

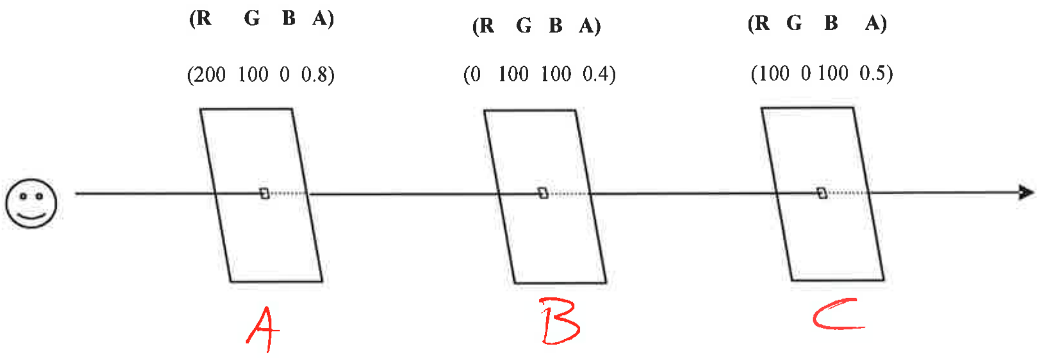
$$C_{out} = (1-\alpha_c)C_{in} + \alpha_c C_c$$

3a. (10pts) For the given pixel, what is the result of back-to-front compositing? (show all steps):

· +

	C_{in}	α_c	C_c	$\alpha_c C_c$	$1-\alpha_c$	$1-\alpha_c C_{in}$	C_{out}
A	0 0 0	.2	0 100 0	0,20,0	.8	0,0,0	0 20 0
B	0 20 0	.6	100 0 0	60,0,0	.4	0 8 0	60 8 0
C	60 8 0	.4	0,100,100	0, 40, 40	.6	36 4.8 0	36 ~~[illegible]~~ ~~100~~ 40 44.8

+



The front-to-back compositing equation is:

$$C_{out} = C_{in} + (1 - \alpha_{in})\alpha_c C_c$$

$$\alpha_{out} = \alpha_{in} + \alpha_c(1 - \alpha_{in})$$

3b. (10pts) For the given pixel, what is the result of front-to-back compositing? (show all steps):

	~~Cout~~ C_{in}	$1-\alpha_{in}$	$\alpha_c C_c$	α_{in}	$\alpha_c(1-\alpha_{in})$	α_{out}
A	0 0 0	1	160, 80, 0	0	.8	,8
B	160 80 0	.2	0 40 40	,8	.08	,88
C	160 88 8	,5	50 0 50	.88	,06	,94

C_{out}

$$\begin{array}{rrr} 160 & 80 & 0 \\ +\ 0 & 8 & 8 \\ 160 & 88 & 8 \\ +25 & 0 & 25 \\ \hline 185 & 88 & 33 \end{array}$$

$$\boxed{185 \quad 88 \quad 33}$$

3.

a. [10] What are the GlBlendFunc() parameters for implementing back-to-front compositing?

$S = C_c$ $D = C_{in}$

SRC.α (Src_{color}) + 1 − SRC.α (Dst_{color}) ~~B~~

glBlendFunc (~~B~~GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)

b. [10] What are the GlBlendFunc() parameters for implementing front-to-back compositing?

$\text{Result}_c = \text{DST}_{color} + \text{SRC}_c * \text{SRC}_\alpha * (1 - \text{DST}_\alpha)$

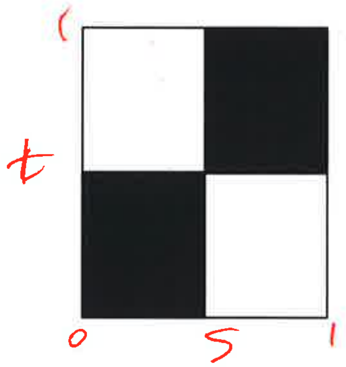
↕

$\text{Result}_\alpha = \text{DST}_\alpha + \text{SRC}_\alpha (1 - \text{DST}_\alpha)$

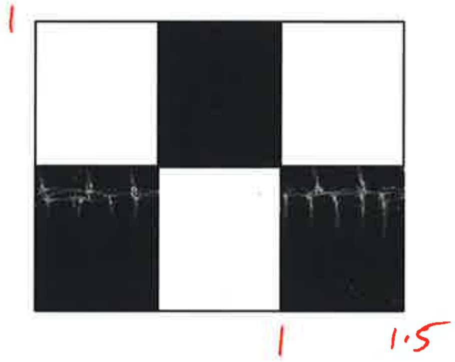
(GL_ONE_~~mius~~ MINUS_DST_ALPHA, GL_ONE)

But, must premultiply $\text{SRC}_c * \text{SRC}_\alpha$ in the fragment shader!

5. [10pts] Given the following texture:



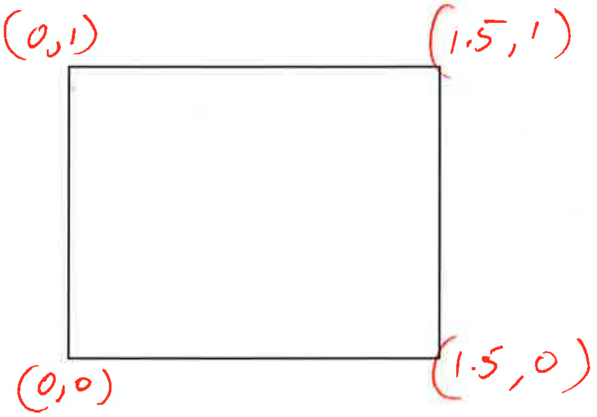
What texture coordinates would you assign to a quad to get this result?:



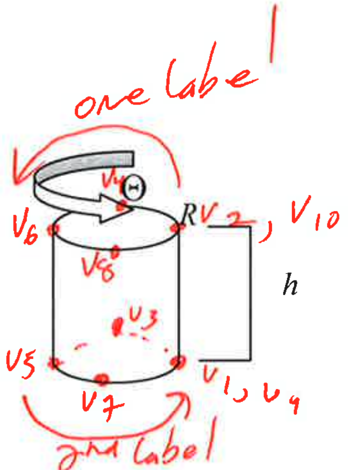
s wraps
not clamps

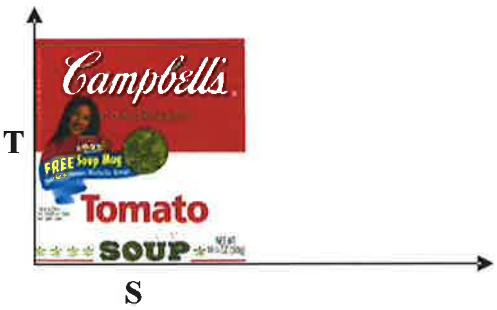
Quad (write the texture coordinates at the corner vertices):

(s, t)



$x = R\cos(\Theta)$
$y = R\sin(\Theta)$
$z = h$





6. **(10pts) You are given the opportunity for perform the modified inverse Warhol transform, $W^{(-1)}$, which takes the Campbell's Soup label and wraps it around a geometric can so that the label appears twice (once on each side). What vertices are required (give them and indicate them on the figure)? What are the texture coordinates required (give them below)? Be sure to describe any assumptions you make. Although the image to be textured does not appear square, you can assume the soup label image is square.**

S ~~goes~~ goes to 2

Assumptions:

	X	Y	Z	S	T
V1	R cos(0)	R sin(0)	0	0	0
V2	R cos(0)	R sin(0)	h	0	1
V3	R cos(90)	R sin(90)	0	.5	0
V4	R cos(90)	R sin(90)	h	.5	1
V5	R cos(180)	R sin(180)	0	1	0
V6	R cos(180)	R sin(180)	h	1	1
V7	R cos(270)	R sin(270)	0	1.5	0
V8	R cos(270)	R sin(270)	h	1.5	1
V9	R cos(360)	R sin(360)	0	2	0
V10	R cos(360)	R sin(360)	h	2	1

S wraps not clamps