Computer Graphics - CS 4300
Take Home Final Exam - Fall 2014

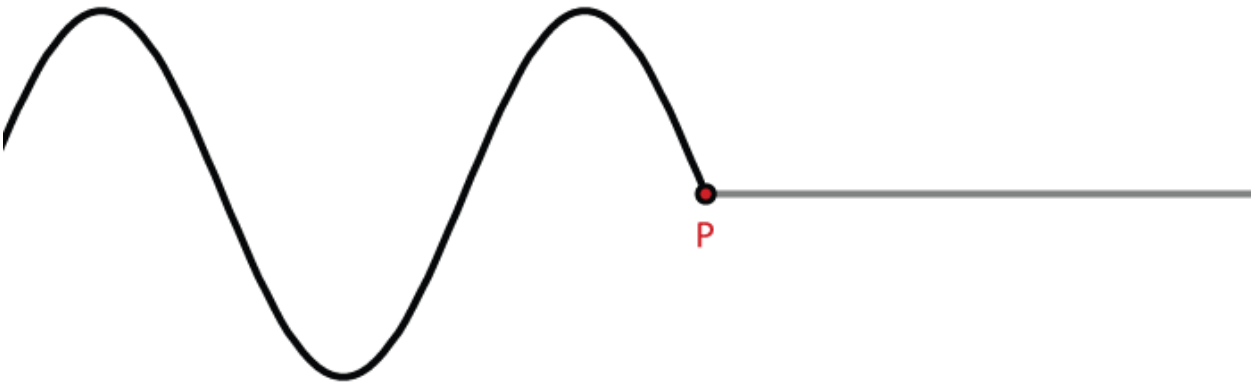
Student Name: Anh Tran
Professor: Nik Bear Brown

Rules:
1. Must do your own work
2. Ask if you don't understand a question.
3. You can discuss general strategies with classmates but any answers and implementation must be your own.
4. Any code, answers not your own must be properly cited.

Due: December 11th, 2014

Short Answer (25 Points)

Q1 (5 Points) What is the continuity of the curve below at the red point? For x<=P it is a sin curve, *y(x)=sin(x)*, and it is a straight line *y(x)=0* for x>P. **C1???**



C0 because the derivative of this curve is not continuous

Q2 (5 Points) Which color is lighter, A or B? (Circle or underline the lighter color)

A	Value	B	Value
Red	#FF0000	Tomato	#***FF6347***
UCLA Gold	***(255,187,54)***	Orange	(255,165,0)
Chartreuse	(0.49,1,0)	Green-Yellow	***(0.69,1, 0.18)***
Boston Red Sox Blue	***#000092***	Navy	#000080
Hot Pink	(255,105,180)	Pokemon Pink	***(255,182,193)***

Q3 (5 Points) Describe Lambert's cosine law. How would you implement this using CgFX?

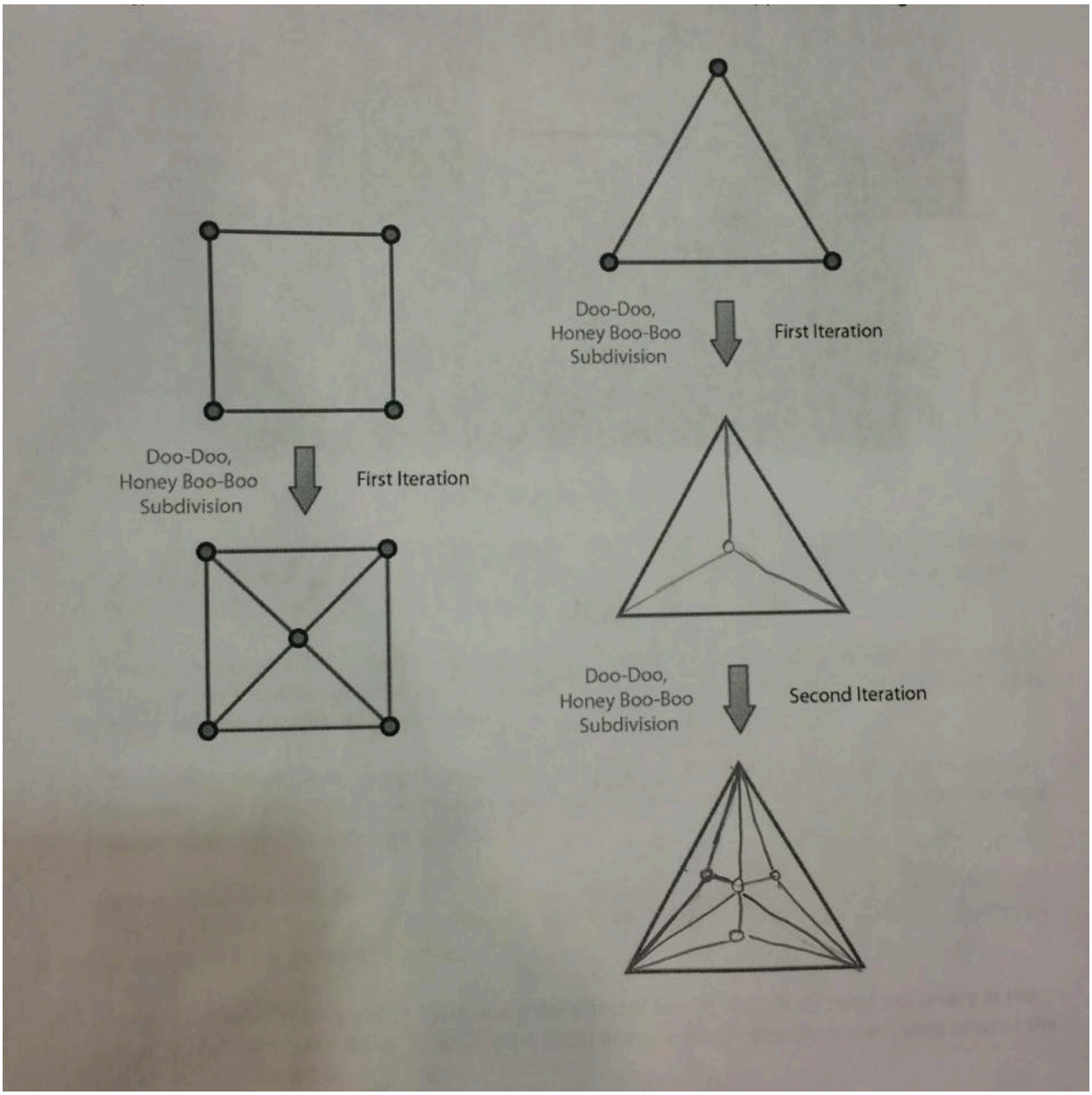
The luminous intensity from a diffusible reflecting surface is directly proportional to cosine of the angle between the observer's line of sight and surface normal

float3 lambert = saturate(dot(lightDir, worldNormal));.

Q4 (5 Points) The *Doo-Doo, Honey Boo-Boo Subdivision Scheme* works in the following manner:

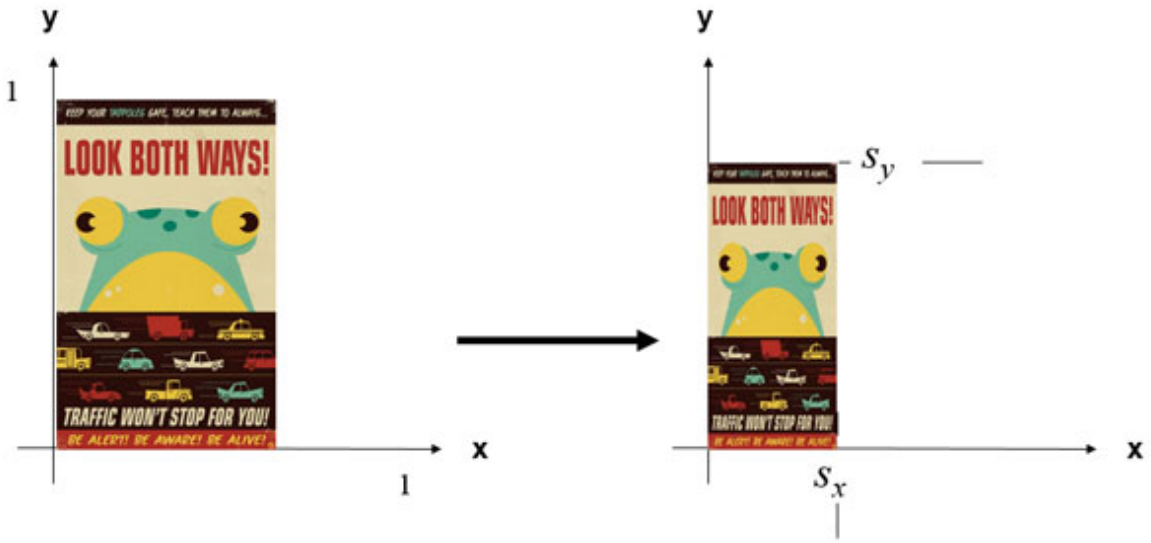
1. A new vertex is created at the centroid (center of mass) of each polygon of the mesh.
2. All of the existing vertices in each polygon is then connected with edges to the new centroid vertex (generating n new faces).

Below is an example of the *Doo-Doo, Honey Boo-Boo Subdivision Scheme* applied to a quad for one iteration. Show (draw) the Doo-Doo Honey Boo-Boo Subdivision Scheme applied to a triangle for two iterations.



Q5 (5 Points) Specify a transformation matrix for the image transformation below:

Scale matrix



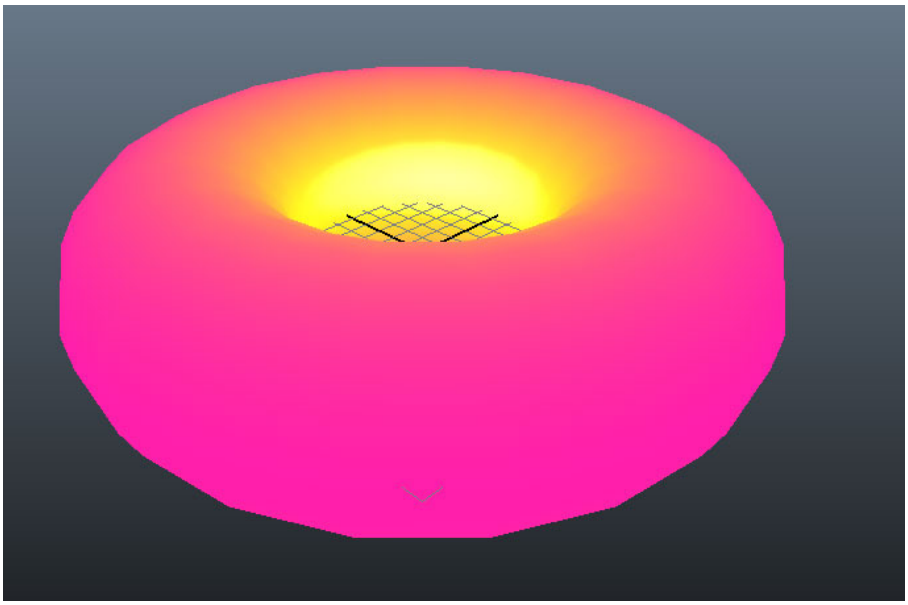
$$\begin{bmatrix} sx & 0 \\ 0 & sy \end{bmatrix}$$

Coding & Analysis (75 Points)

Any relevant code and analysis should be zipped together with your final.

Q6 (25 Points)

Use the attached "hoochy" CgFX shader code (hoochy.zip) to answer the questions below.



"hoochy" CgFX shader applied to a torus

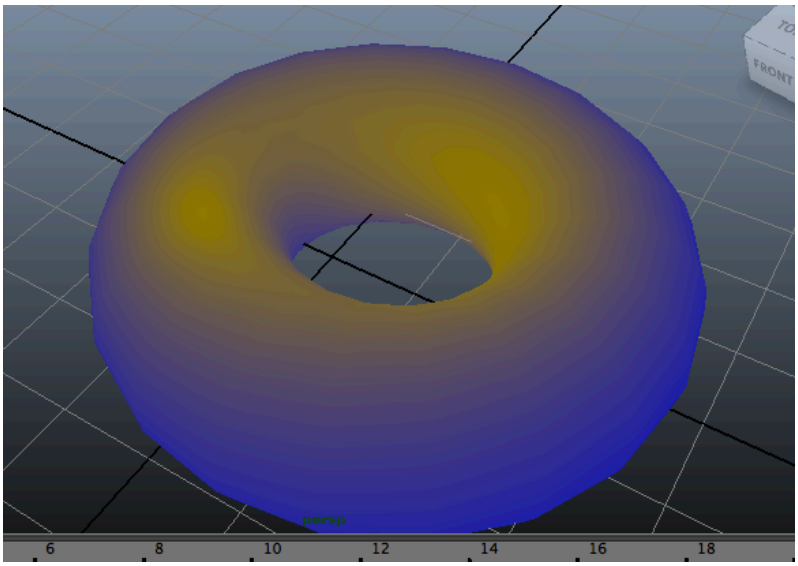
A. (5 Points) Does the hoochy shader have an ambient component? (If yes, you MUST point out where in the "hoochy" CgFX shader code the effect exists (if it does) and you MUST provide a screenshot of what the shader would look like without the effect.)

NO

B. (5 Points) Does the hoochy shader have a specular component? (If yes, you MUST point out where in the "hoochy" CgFX shader code the effect exists (if it does) and you MUST provide a screenshot of what the shader would look like without the effect.)

```
   float3 specContrib = spec * Ks * LampColor;


float gKs <
   string UIWidget = "slider";
   float UIMin = 0.0;
   float UIMax = 1.0;
   float UIStep = 0.05;
   string UIName =  "Specular";
> = 0.4;

float gSpecExpon <
   string UIWidget = "slider";
```

```
    float UIMin = 1.0;
    float UIMax = 128.0;
    float UIStep = 1.0;
    string UIName =  "Specular Exponent";
> = 5.0;
```

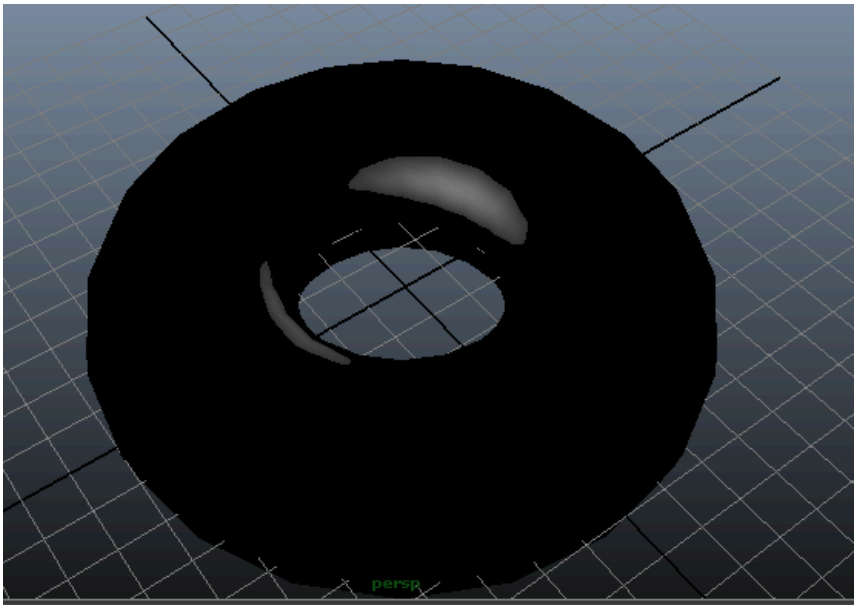


YES

C. (5 Points) Does the hoochy shader have a diffuse component? (If yes, you MUST point out where in the "hoochy" CgFX shader code the effect exists (if it does) and you MUST provide a screenshot of what the shader would look like without the effect.)

YES

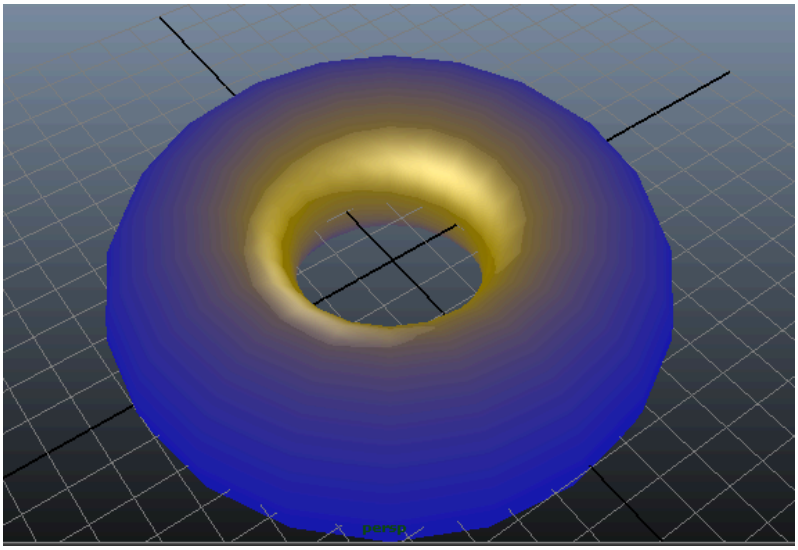
```
float3 diffuseColor = SurfaceColor;

/ Hoochy Light
float3 gHoochyColor <
    string UIName =  "Hoochy Light";
    string UIWidget = "Color";
> = {0.07f,0.07f,0.07f};

float3 gWarmColor <
    string UIName =  "Hooch Warm Tone";
    string UIWidget = "Color";
> = {0.5f, 0.4f, 0.05f};

float3 gCoolColor <
    string UIName =  "Hooch Cool Tone";
    string UIWidget = "Color";
> = {0.05f, 0.05f, 0.6f};
```



D. (5 Points) Does the hoochy shader have any fresnel effects? (If yes, you MUST point out where in the "hoochy" CgFX shader code the effect exists (if it does) and you MUST provide a screenshot of what the shader would look like without the effect.)

YES

*spec *= glossy_drop(spec,GlossTop,(GlossTop-GlossEdge),GlossDrop);*



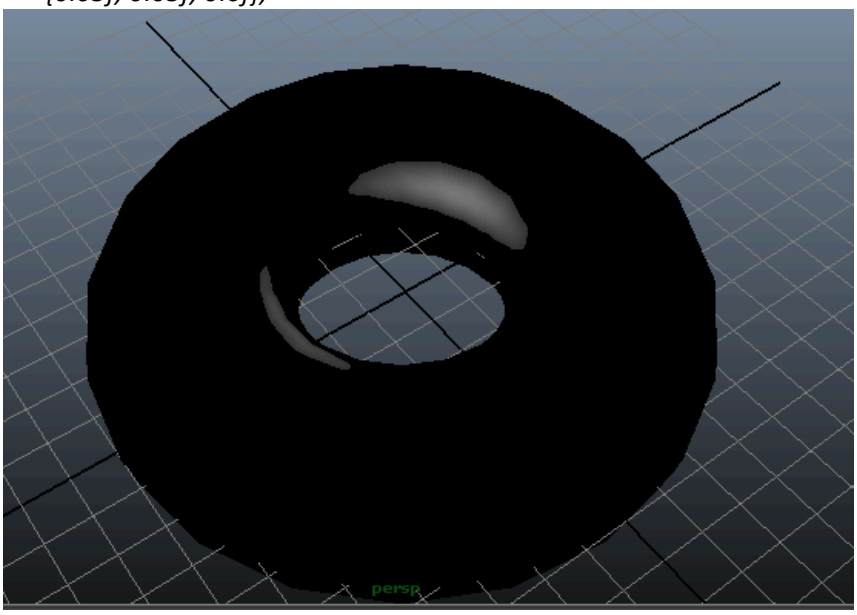
E. (5 Points) Does the hoochy shader have any color effects? (If yes, you MUST point out where in the "hoochy" CgFX shader code the effect exists (if it does) and you MUST provide a screenshot of what the shader would look like without the effect.)

YES

float3 toneColor = lerp(CoolColor,WarmColor,hoochFactor);

/ Hoochy Light

```
float3 gHoochyColor <
   string UIName =  "Hoochy Light";
   string UIWidget = "Color";
> = {0.07f,0.07f,0.07f};

float3 gWarmColor <
   string UIName =  "Hooch Warm Tone";
   string UIWidget = "Color";
> = {0.5f, 0.4f, 0.05f};

float3 gCoolColor <
   string UIName =  "Hooch Cool Tone";
   string UIWidget = "Color";
> = {0.05f, 0.05f, 0.6f};
```



Q7 (50 Points)

Write code (in any language) to represent the data and functionality for a simple 2D particle system.
I used the processing's particle example as a reference
The particle system must:

i. (10 Points) Have a class for the particle system that creates, and destroys particles.
ii. (5 Points) The initial position, initial velocity, initial size, initial color, initial transparency, lifetime, and mass can be set at creation.
iii. (5 Points) The particle system must limit the lifetime and total number of particles.
iv. (10 Points) Have a class for the particles that include all of the properties that a minimal particle system should have.
v. (10 Points) Collisions must detected.
vi. (10 Points) The particles must obey Newton's laws of motion.
vii. Any software used as a basis to start your code must be cited.