CPSC 314
Final Exam

April 13, 2018

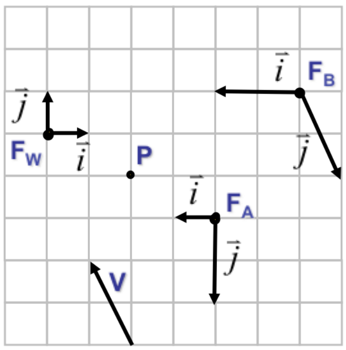
Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: Solutions

Student Number: ______

Question 1	/ 12
Question 2	/ 11
Question 3	/ 10
Question 4	/ 7
Question 5	/ 14
Question 6	/ 13
TOTAL	/ 67

1. Coordinate Frames



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_W = \begin{bmatrix} -1 & 0 & 4 \\ 0 & -2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_A$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_B = \begin{bmatrix} 0.5 & -0.5 & 1.75 \\ 0 & 1 & 1.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_A$$

(a) (3 points) Express point P in each of the three coordinate frames.

$P_W(2,-1)$ $P_A(2,-0.5)$ $P_B(2.5,1)$

(b) (3 points) Express vector V in each of the three coordinate frames.

$V_W(-1,2)$ $V_A(1,-1)$ $V_B(0,-1)$

(c) (2 points) Find the 3×3 transformation matrix which takes a point from F_A and expresses it in terms of F_W. I.e., determine M, where $P_w = MP_A$. Write your answer in the appropriate space to the right of the diagram.

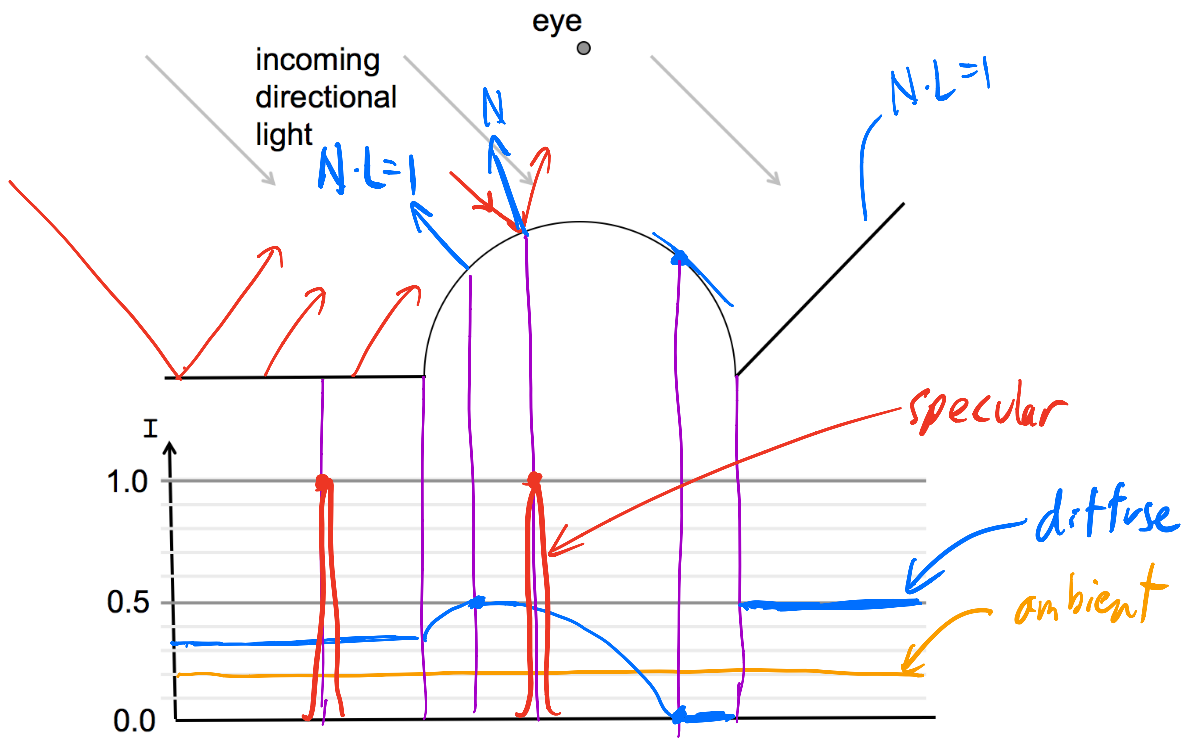
(d) (2 points) Find the 3×3 transformation matrix which takes a point from F_A and expresses it in terms of F_B. I.e., determine M, where $P_B = MP_A$. Write your answer in the appropriate space to the right of the diagram.

(e) (2 points) Construct the matrix M that computes $P_w = MP_A$ (as in part (c) above), using translates, rotates, and scales. Express M as a product of such matrices. Use any order of matrices that suits you.

$M = \text{Trans}(4,-2,0)\ \text{Rotate}(z,180^\circ)\ \text{Scale}(1,2,1)$

2. Local Illumination

(a) (6 points) Sketch the ambient, diffuse, and specular components for the scene below, as would be computed by the Phong illumination model, i.e., $I = I_a k_a + I_L k_d (N \cdot L) + I_L k_s (R \cdot V)^n$. Assume that the eye is directly above the scene at a large distance, and use the values $I_a = I_L = 1$, $k_a = 0.2$, $k_d = 0.5$, $k_s = 1$, $n = 200$.



(b) (2 points) Which terms in the Phong illumination model need to have their values clamped, i.e., bounded in value, and why?

N·L and R·V; do not want "negative light"

(c) (1 point) Which terms in the Phong illumination model are physically based?

diffuse term: N·L

(d) (2 points) Give the reflected RGB light color for the following combination of light colors and surface colors. Note: magenta is an additive combination of red and blue. Assume RGB∈ [0, 1].

white light, blue surface: (R,G,B) = (0,0,1) — (1,1,1) * (0,0,1)

red light, blue surface: (R,G,B) = (0,0,0) — (1,0,0) * (0,0,1)

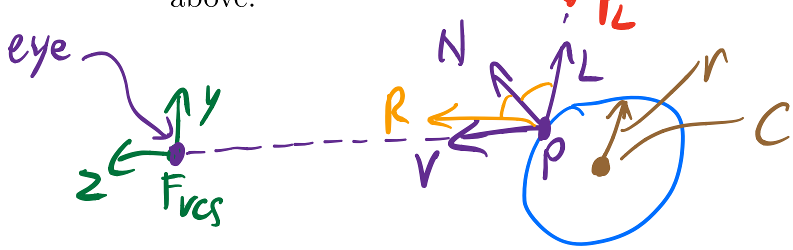
yellow light, green surface: (R,G,B) = (0,1,0) — (1,1,0) * (0,1,0)

yellow light, magenta surface: (R,G,B) = (1,0,0) — (1,1,0) * (1,0,1)

3. Fragment shaders

Give a GLSL implementation of the fragment shader for the specular term of the Phong shading model, i.e., $I = I_L k_s (R \cdot V)^n$ for a sphere. As input, assume that the fragment shader has access to: I_L, k_s, both as `vec3`, the point P on the surface of the sphere, passed in as a `varying vec3` variable, and that the other known quantities are: the sphere center, C, and radius, r; and the light position, P_L. Assume that all these quantities are given in VCS coordinates, and that other quantities need to be derived. You can use the `reflect(I,N)` function. In your shader, declare all non-local variables as being `varying, attribute, or uniform`.

(a) (2 points) Provide a diagram that illustrates all the needed geometry, as described above.



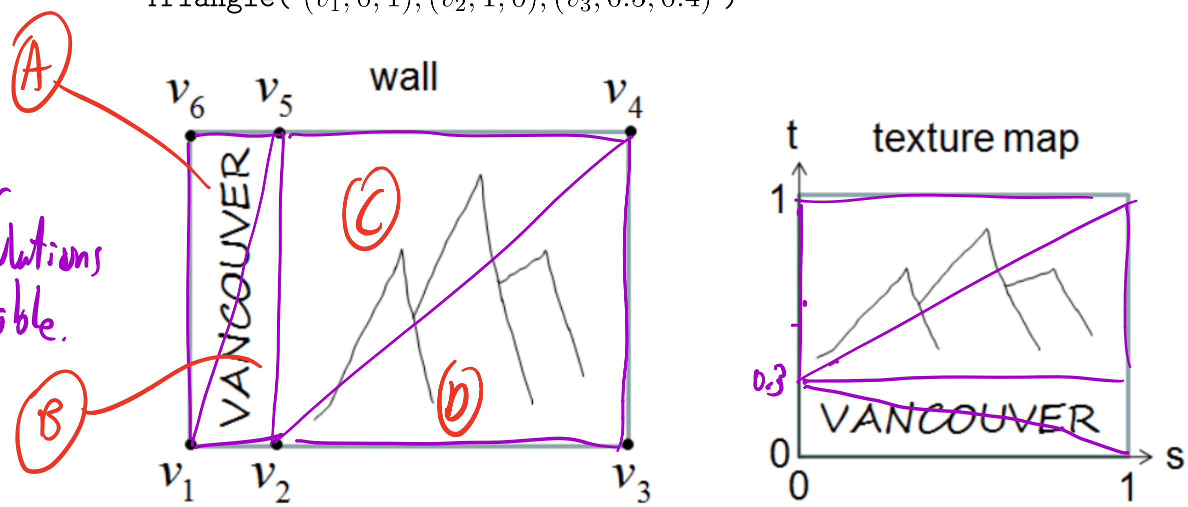
(b) (8 points) Give your GLSL code

```
varying vec3 P;          uniform vec3 I_L, k_s;
uniform vec3 P_L, C;     uniform float n;
uniform float r;
main() {
   vec3 Peye = vec3(0,0,0);
   N = normalize(P-C);
   L = normalize(P_L-P);
   V = normalize(Peye-P);
   R = reflect(-L,N)
   dp = max(0, dot(R,V))
   gl_FragColor = I_L * k_s * pow(dp,n);
}
```

4. Texture mapping

   (a) (4 points) You wish to create a texture-mapped wall, as shown on the left, with the help of the texture map shown on the right. Sketch the four triangles needed to do this. Then give a detailed specification for each triangle, which consists of three vertices, each having a vertex and its associated s, t texture coordinate, e.g., (v, s, t):
   `Triangle(` $(v_1, 0, 1), (v_2, 1, 0), (v_3, 0.5, 0.4)$ `)`

other triangulations possible.



(A) Triangle($(V_1, 0, 0.3), (V_5, 1, 0), (V_6, 1, 0.3)$)
(B) Triangle($(V_1, 0, 0.3), (V_2, 0, 0), (V_5, 1, 0)$)
(C) Triangle($(V_2, 0, 0.3), (V_4, 1, 1), (V_5, 0, 1)$)
(D) Triangle($(V_2, 0, 0.3), (V_3, 1, 0.3), (V_4, 1, 1)$)

   (b) (2 points) Give one advantage and one disadvantage of procedural textures, as compared to regular texture maps.

+ computation instead of memory, i.e., less memory
− could be slow to compute; difficult to prefilter, i.e., MIPMAP!

   (c) (1 point) Give a usage for animated texture coordinates.

animated flames, water, moving clouds

5. Short Answer

(a) (1 point) Reproducing the patterns of light caused by the lensing effect of surfaces is best done using the following rendering method: photon mapping

(b) (1 point) Shader variables that are interpolated across triangles are called varying variables.

(c) (1 point) Viewing and projection matrices would typically be uniform variables in a vertex shader.

(d) (1 point) the vertex positions, as modeled in object coordinates, are defined as attribute variables in the vertex shader.

(e) (1 point) `T or F` A triangle can always be culled if both of its vertices lie outside the view volume.

(f) (1 point) `T or F` Testing the z-buffer for visibility before the fragment shader, i.e., an early Z-buffer test, can lead to more efficient rendering.

(g) (1 point) `T or F` Texture coordinates are typically stored as **`attributes`** and then turned into **`varyings`**.

(h) (1 point) `T or F` For the barycentric coordinates of an arbitrary point on a plane, either inside or outside the given triangle, at least one of the barycentric coordinates will always be positive.

(i) (1 point) `T or F` Backface culling can be done in either VCS or NDCS.

(j) (1 point) Express the point (x,y,z,h)= (3,1,2,2) in cartesian coordinates: $\left(\frac{3}{2}, \frac{1}{2}, 1\right)$.

(k) (1 point) Given a pixel intensity $i \in [0, 1]$, give an expression for computing a new intensity i' for a "toon"-shaded version of the image, as in the assignment.

$i' = n * \text{floor}(i * n + 0.5)$

(l) (3 points) For each of the following types of light paths, list which types of rendering methods (L=local Phong shading, RT=ray-tracing, P=path tracing, R=radiosity) could be used to model that kind of light path. In the descriptions below, S = specular surface, D = diffuse surface.

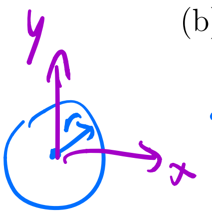
light → S → eye L, RT, P
light → D → eye L, RT, P, R
light → S → S → eye RT, P
light → D → S → eye RT, P
light → D → D → eye P, R
light → D → D → S → eye P

6. Miscellaneous

   (a) (4 points) Sketch the graphics pipeline, i.e., the series of transformations that takes a vertex from object coordinates, i.e., OCS, through to device or display coordinates, i.e., DCS. Label the tranformation matrices or other transformations involved, and the coordinate systems.

$$\text{OCS} \rightarrow \boxed{M_{model}} \xrightarrow{\text{WCS}} \boxed{M_{view}} \xrightarrow{\text{VCS}} \boxed{M_{proj}} \xrightarrow{\text{CCS}} \boxed{/h} \xrightarrow{\text{NDCS}} \boxed{\text{viewport transform}} \xrightarrow{\text{DCS}}$$

   (b) (3 points) Give explicit, implicit, and parametric equations for a circle of radius 1, centred at the origin.

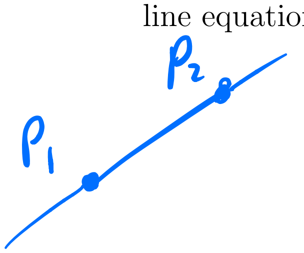


explicit $y = \pm\sqrt{1-x^2}$

implicit $0 = 1 - x^2 - y^2$

parametric $x = \cos(t)$ $\quad t \in [0, 2\pi)$
$y = \sin(t)$

   (c) (2 points) Given a line defined by points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, show how to compute an implicit line equation for the line passing through those points. Your line equation should be valid for vertical lines as well as horizontal lines.



$y = mx + b \Rightarrow 0 = Ax + By + C$

$$y = y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1)$$

$$(y - y_1)(x_2 - x_1) = (y_2 - y_1)(x - x_1)$$

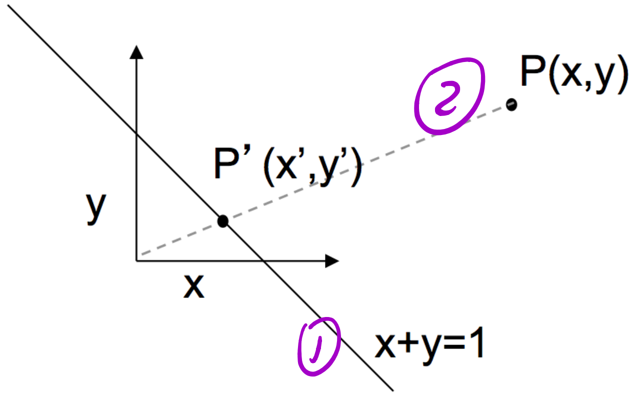
$$0 = -(y - y_1)(x_2 - x_1) + (y_2 - y_1)(x - x_1)$$

$$0 = (y_2 - y_1)x + (x_1 - x_2)y + y_1(x_2 - x_1) - x_1(y_2 - y_1)$$

$$0 = \underbrace{(y_2 - y_1)}_{A}x + \underbrace{(x_1 - x_2)}_{B}y + \underbrace{y_1x_2 - x_1y_2}_{C}$$

(d) (4 points) Challenge question

Develop a 2D projection matrix (3×3) that when followed by a divide-by-h will project 2D points $P(x, y)$ onto the image plane defined by $x + y = 1$, as shown below.



line 1 $x' + y' = 1$

line 2 $y' = \left(\frac{y}{x}\right) x'$

$\Rightarrow \quad x' + \left(\frac{y}{x}\right) x' = 1$

$x'\left(1 + \frac{y}{x}\right) = 1$

$x'\left(\frac{x+y}{x}\right) = 1$

$\boxed{x' = \frac{x}{x+y}}$ $\quad y' = 1 - x' = 1 - \frac{x}{x+y} = \frac{(x+y) - x}{x+y}$

$\boxed{y' = \frac{y}{x+y}}$

$$\begin{bmatrix} x \\ y \\ x+y \end{bmatrix} = \begin{bmatrix} 1 & & 0 \\ & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\xrightarrow{/h} = \begin{bmatrix} x/x+y \\ y/x+y \\ 1 \end{bmatrix}$$