The University of Victoria
Department of Computer Science

CSC 305 Section A01

Introduction to Computer Graphics (CRN 20690)

Final Exam April 2019

* * *

Time allowed: 120 minutes.
April 12, 2019, 9:00 am - 11:00 am.
Instructor: Li Ji

- **This Exam has 6 Questions, with 5 pages including this page.**
- **All answers are to be written on UVic's lined answer booklet(s), provided at the time of the final exam.**
- **This is a closed book exam; notes are not permitted. Non-graphical calculators are permitted.**
- **The grade of this midterm is marked out of 100 points in total, and this exam contributes to 40% of your final course grade. <u>You must pass the final to pass this course</u> (scoring at 50 points or above).**
- **Clearly show all your steps of derivation and calculation for mathematical questions. Partial marks will be given based on the correctness of mathematical procedures when there are arithmetics mistakes or incomplete solutions.**
- **Provide succinct answers for non-mathematical questions. Avoid writing long paragraphs; time is limited.**

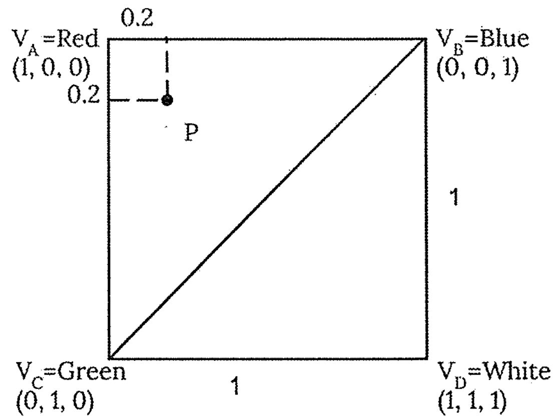
Question 1: Mathematical Foundations (15).

(a) Let vector **v** = (3,5,4),

- (i) What is the length of **v**? **(2)**
- (ii) What is the angle between **v** and the positive Y axis (0,1,0)? **(2)**
- (iii) What is the distance between **v** and the line passing through two points **a**=(1, 2, 4) and **b**=(4, -2, 4)? **(5)**

(b) A square with edges of unit length has its four vertices associated with RGB colours red, green, blue and white (see figure on the right). A point **P** is positioned in the upper-left triangle with distance 0.2 unit to the left edge and 0.2 unit to the top edge. Calculate **P**'s RGB colour using its distances to the edges with:

- (i) Barycentric interpolation inside the triangle $\mathbf{V_A V_B V_C}$ **(3)**
- (ii) Bilinear interpolation within the square $\mathbf{V_A V_B V_C V_D}$**(3)**



Question 1 (b)

Question 2: Ray Tracing and Shading (20).

A ray has been cast from a point of view towards a triangle, and intersect the triangle at point **P**. Calculate the colour of **P** using the Phong reflection model based on the following information:

1. The point of view is positioned at **V**=(-6, 1, -5).
2. A point light source is placed at **L**=(204, 96, 205), with a white color $\mathbf{C_L}$=(1,1,1).
3. The triangle has its three vertices placed at $\mathbf{V_A}$=(0,0,5), $\mathbf{V_B}$=(5, -5, 5) and $\mathbf{V_C}$=(3,-3,2).
4. Point **P** has barycentric coordinates of $\alpha = 0.2, \beta = 0.8$.
5. The ambient and diffuse albedos are $\mathbf{I_a}$= $\mathbf{I_d}$ = (0.8, 0.2, 0.3). The specular albedo equals to full reflection, $\mathbf{I_s}$ = (1,1,1).
6. The ambient coefficient is set to $\mathbf{K_a}$ = 0.2, the diffuse coefficient set to $\mathbf{K_d}$ = 0.6 and the specular coefficient set to $\mathbf{K_s}$ = 0.5, with specular power being α = 2.

You may use either the $\mathbf{R} \cdot \mathbf{V}$ or the $\mathbf{N} \cdot \mathbf{H}$ formation for calculating the specular reflection component. We do not consider backface culling in this question. **(20)**
(Please show all <u>intermediate steps</u> of derivation and calculation. Points will be given even if your solution is not complete.)

Question 3: Raster Graphics (12).
The following is a code excerpt from a Unity vertex-fragment shader pair.

```
1. #pragma vertex vert
2. #pragma fragment frag
3.    struct appdata
4.    {
5.      float4 vertex : POSITION;
6.      float2 uv : TEXCOORD0;
7.    }
8.
9.    sampler2D _MainTex;
10.   float4 _light_color;
11.
12.   struct v2f
13.   {
14.     float4 vertex : SV_POSITION;
15.     float2 uv : TEXCOORD0;
16.   }
17.
18.   v2f vert (appdata v)
19.   {
20.      v2f o;
21.      o.vertex = UnityObjectToClipPos(v.vertex);
22.      o.uv = v.uv;
23.      return o;
24.   }
25.
26.   float4 frag (v2f i) : SV_Target
27.   {
28.      float4 albedo = tex2D(_MainTex, i.uv);
29.      return albedo * _light_color;
30.   }
```

Based on your experience of assignment 2 and 3, <u>briefly</u> explain the following:

(a) What kind of multiplication is executed after the return statement (line 29) of the fragment shader (`albedo * _light_color`)? **(2)**

(b) In the v2f struct (line 12), how is the `i.uv` input parameter (line 26) of the fragment shader calculated from the `o.uv` component returned from the vertex shader (line 23)? **(3)**

(c) What is the difference between the `appdata` input struct (line 3) and the variables listed individually (line 9), such as `_MainTex` and `_light_color`? **(2)**

(d) Assuming we rendered one cube that consists of 12 triangles and 36 vertices with this shader. The screen resolution is 1024x768 pixels; the rendered cube occupy 200 pixels on the screen and there are no other objects occluding the cube's rendering. How many times are `vert` and `frag` executed on the GPU? <u>Briefly</u> explain your answer. **(5)**
(Exact numbers depend on specific hardware architecture, and your answer will be marked full points as long as your overall reasoning is correct.)

Question 4: Matrix Based Transformations (13).
Given a perspective transformation matrix M_{persp}:

$$M_{persp} = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -2fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) Show the process of applying this matrix to transform a given 3D Cartesian coordinate (x, y, z) and obtaining the resultant Cartesian coordinate (x', y', z'). Derive x', y' and z' in terms of x, y, z, f and n. **(8)**
(b) Show that z' preserves the order of z, *i.e.* if $z_1 > z_2$ then $z'_1 > z'_2$ and *vice versa.* **(5)**

Question 5: Parametric Curves (20).
The cubic Hermit spline can be written in basis function form as:

$$f(t) = (2t^3 - 3t^2 + 1)\mathbf{P}_0 + (-2t^3 + 3t^2)\mathbf{P}_3 + (t^3 - 2t^2 + t)\mathbf{T}_0 + (t^3 - t^2)\mathbf{T}_3$$

where $\mathbf{P}_0$ is the position of the knot at the beginning of the curve, $\mathbf{P}_3$ is the position of the knot at the end of the curve, $\mathbf{T}_0$ is the tangent vector at the beginning of the curve, and $\mathbf{T}_3$ is the tangent vector at the end of the curve.
It is also known that the cubic Bézier curve uses the four cubic Bernstein polynomials as basis functions.

(a) Write down the expression for the cubic Bézier curve in its basis function form, with a sequence of control points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$. **(3)**
(b) Prove that the cubic Bézier curve is the same as the cubic Hermit spline if we use $T_0 = 3(P_2 - P_1)$ and $T_3 = 3(P_4 - P_3)$ as tangent vectors for knots P_0 and P_1. **(8)**
(c) Given four control points of a two dimensional cubic Bézier curve: (-1, -3), (-1, 5), (11, 5), (11, -3). Calculate the 8 control points that define the two cubic Bézier segments created by segmenting the original curve using the de Casteljau subdivision algorithm at parameter t = 0.25. **(6)**
(d) Calculate the tangent vector of the original cubic Bézier curve at the point of segmentation in (c). **(3)**

Question 6: General Knowledge (20).

(a) Suppose an artist specified the following keyframes for an object's surface colour and position on the **x** axis. The artist also specified that linear interpolation is to be used to set the object's x-position offset in every frame.

Time	0:00	0:10	0:12	0:14
Surface Colour	Red	Orange	Yellow	Blue
x-Position	1	2	4	5

(i) During which part of this animation is the object moving fastest along the x-axis? **(2)**
(ii) Assuming that the artist wants the object's surface to stay with constant brightness and colour saturation, describe a method that could be used to

interpolate the object's surface colour between keyframes. We do not consider lighting in this question. **(5)**

(iii) During which time does the hue of the object's surface change most rapidly if your method is used? **(3)**

(b) Briefly describe the reasons that we need a dedicated Graphics Processing Unit (GPU) in our computers instead of just faster CPUs or more CPU chips. **(3)**

(c) What are the three major behavioural rules for simulating boids? According to your experiment in assignment 3, briefly explain how did you adjust the weight and priority between these three rules. **(4)**

(d) A straight line segment between two points **A** and **B** in the three-dimensional space can be expressed simply as a linear interpolation:

$f(t) = (1 - t)\mathbf{A} + t\mathbf{B}, t \in [0, 1]$.

Show that above is also an arc-length parametrization for the line segment. **(3)**

* * *

END OF EXAM