CSci 5607 **Second Exam (take-home portion)** **Spring 2023**

Due Weds May 3, 6:30pm

Name:____ **ID#:___________**

*There are **12** multi-part questions in this part of the test for a total of **120** points, plus 5 points of optional extra credit.*

You may use any available primary resource, including: lecture slides, lecture videos, textbooks (hardcopy or online), blogs, online tutorials, or any class notes you have prepared yourself. You are on your honor not to attempt to seek direct assistance on any of the question on this exam from other people, such as colleagues or strangers on the internet.

To earn full credit, you must show all of your work.

Reflection and Refraction

1. Consider a sphere ***S*** made of solid glass (η = 1.5) that has radius r = 3 and is centered at the location s = (2, 2, 10) in a vaccum (η = 1.0). If a ray emanating from the point ***e*** = (0, 0, 0) intersects ***S*** at a point ***p*** = (1, 4, 8):

a. (2 points) What is the angle of incidence θ_i ?

b. (1 points) What is the angle of reflection θ_r ?

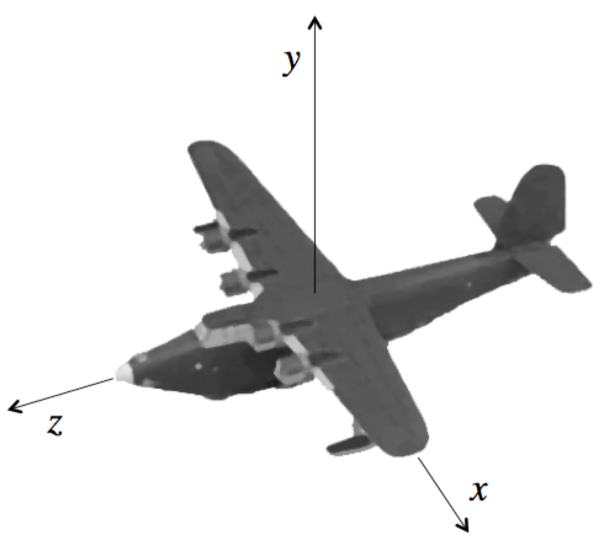
c. (3 points) What is the direction of the reflected ray?

d. (3 points) What is the angle of transmission θ_t ?

e. (4 points) What is the direction of the transmitted ray?

Geometric Transformations

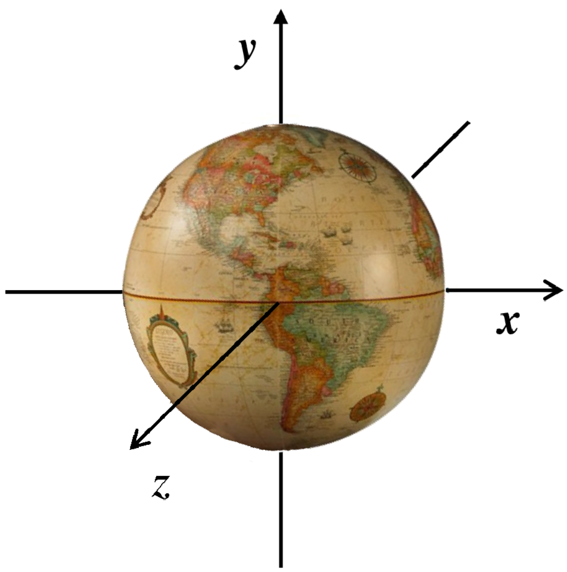
2. (8 points) Consider the airplane model below, defined in object coordinates with its center at (0, 0, 0), its wings aligned with the ±x axis, its tail pointing upwards in the +y direction and its nose facing in the +z direction. Derive a sequence of model transformation matrices that can be applied to the vertices of the airplane to position it in space at the location ***p*** = (4, 4, 7), with a direction of flight w = (2, 1, –2) and the wings aligned with the direction d = (–2, 2, –1).



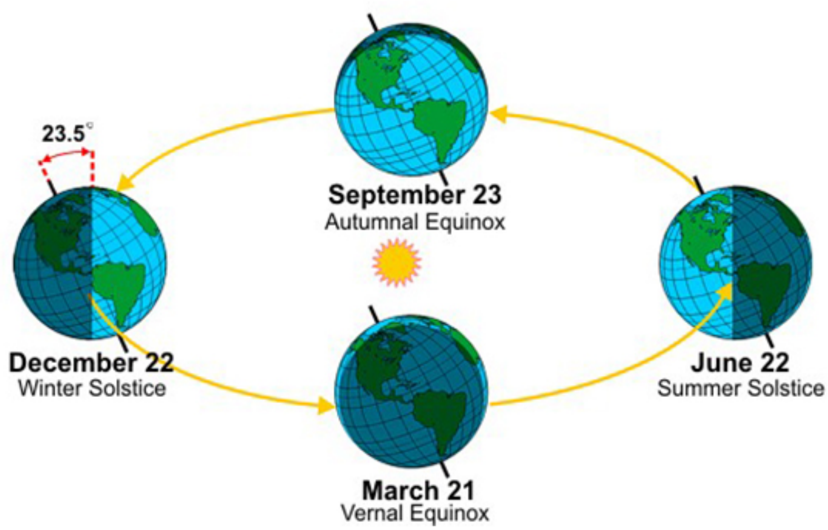
3. Consider the earth model shown below, which is defined in object coordinates with its center at (0, 0, 0), the vertical axis through the north pole aligned with the direction (0, 1, 0), and a horizontal plane through the equator that is spanned by the axes (1, 0, 0) and (0, 0, 1).

a. (3 points) What model transformation matrix could you use to tilt the vertical axis of the globe by 23.5˚ away from (0, 1, 0), to achieve the pose shown in the image on the right?

b. (5 points) What series of rotation matrices could you apply to the globe model to make it spin about its tilted axis of rotation, as suggested in the image on the right?



c. [5 points extra credit] What series of rotation matrices could you use to send the tilted, spinning globe model on a circular orbit of radius r around the point (0, 0, 0) within the xz plane, as illustrated below?



The Camera/Viewing Transformation

4. Consider the viewing transformation matrix **V** that enables all of the vertices in a scene to be expressed in terms of a coordinate system in which the eye is located at (0, 0, 0), the viewing direction ($-n$) is aligned with the $-z$ axis (0, 0, –1), and the camera's 'up' direction (which controls the roll of the view) is aligned with the y axis (0, 1, 0).

a. (4 points) When the eye is located at $\boldsymbol{e}$ = (2, 3, 5), the camera is pointing in the direction (1, –1, –1), and the camera's 'up' direction is (0, 1, 0), what are the entries in **V**?

b. (2 points) How will this matrix change if the eye moves forward in the direction of view? [which elements in V will stay the same? which elements will change and in what way?]

c. (2 points) How will this matrix change if the viewing direction spins in the clockwise direction around the camera's 'up' direction? [which elements in V will stay the same? which elements will change and in what way?]

d. (2 points) How will this matrix change if the viewing direction rotates directly upward, within the plane defined by the viewing and 'up' directions? [which elements in V will stay the same? which elements will change and in what way?]

5. Suppose a viewer located at the point (0, 0, 0) is looking in the $-z$ direction, with no roll ['up' = (0, 1 ,0)], towards a cube of width 2, centered at the point (0, 0, –5), whose sides are colored: red at the plane $x = 1$, cyan at the plane $x = -1$, green at the plane $y = 1$, magenta at the plane $y = -1$, blue at the plane $z = -4$, and yellow at the plane $z = -6$.

a. (1 point) What is the color of the cube face that the user sees?

b. (3 points) Because the eye is at the origin, looking down the $-z$ axis with 'up' = (0,1,0), the viewing transformation matrix **V** in this case is the identity **I**. What is the model matrix **M** that you could use to rotate the cube so that when the image is rendered, it shows the red side of the cube?

c. (4 points) Suppose now that you want to leave the model matrix **M** as the identity. What is the viewing matrix **V** that you would need to use to render an image of the scene from a re-defined camera configuration so that when the scene is rendered, it shows the red side of the cube? Where is the eye in this case and in what direction is the camera looking?

The Projection Transformation

6. Consider a cube of width $2\sqrt{3}$ centered at the point $(0, 0, -3\sqrt{3})$, whose faces are colored light grey on the top and bottom ($y = \pm\sqrt{3}$), dark grey on the front and back ($z = -2\sqrt{3}$ and $z = -4\sqrt{3}$), red on the right ($x = \sqrt{3}$), and green on the left ($x = -\sqrt{3}$).

a. Show how you could project the vertices of this cube to the plane $z = 0$ using an orthographic parallel projection:

i) (2 points) Where will the six vertex locations be after such a projection, omitting the normalization step?

ii) (1 points) Sketch the result, being as accurate as possible and labeling the colors of each of the visible faces.

iii) (2 points) Show how you could achieve this transformation using one or more matrix multiplication operations. Specify the matrix entries you would use, and, if using multiple matrices, the order in which they would be multiplied.

b. Show how you could project the vertices of this cube to the plane $z = 0$ using an oblique parallel projection in the direction $d = (1, 0, \sqrt{3})$:

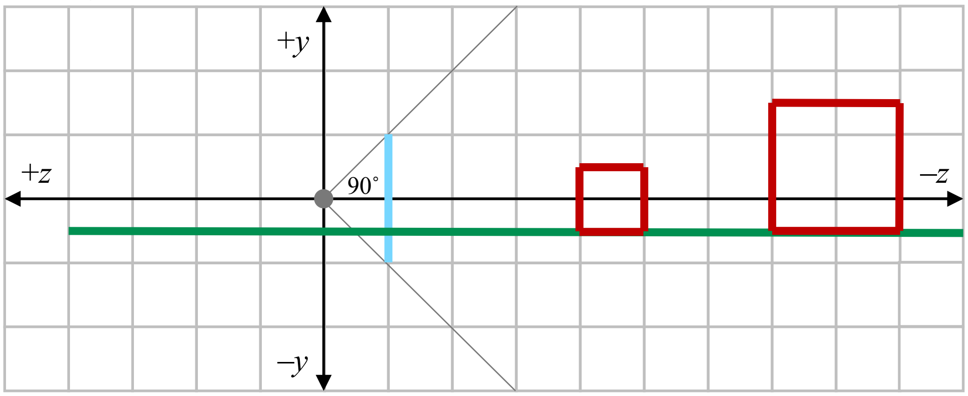
i) (3 points) Where will the six vertex locations be after such a projection, omitting the normalization step?

ii) (2 points) Sketch the result, being as accurate as possible and labeling the colors of each of the visible faces.

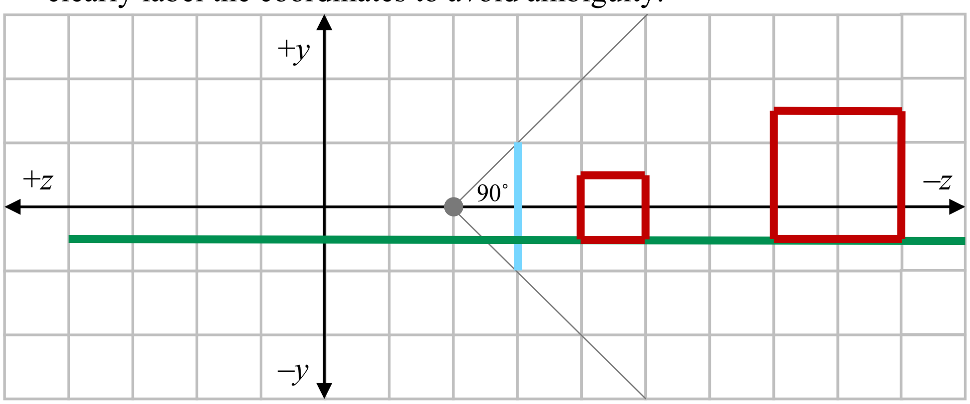
iii) (4 points) Show how you could achieve this transformation using one or more matrix multiplication operations. Specify the matrix entries you would use, and, if using multiple matrices, the order in which they would be multiplied.

7. Consider the simple scene shown in the image below, where two cubes, one of height 1 and one of height 2, are both resting on a horizontal groundplane ($y = -\frac{1}{2}$), with the smaller cube's front face aligned with $z = -4$ and the larger cube's front face aligned with $z = -7$.

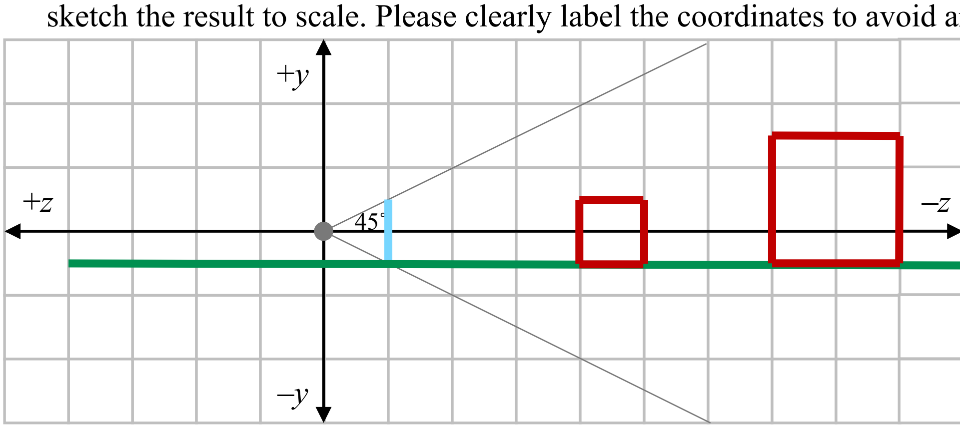
a. (5 points) Let the camera location be (0, 0, 0), looking down the $-z$ axis, with the field of view set at 90˚. Determine the points, in the image plane, to which each of the cube vertices will be projected and sketch the result to scale. Please clearly label the coordinates to avoid ambiguity.



b. (4 points) How would the image change if the camera were moved forward by 2 units, leaving all of the other parameter settings the same? Determine the points, in the image plane, to which each of the cube vertices would be projected in this case and sketch the result to scale. Please clearly label the coordinates to avoid ambiguity.



c. (4 points) How would the image change if, instead of moving the camera, the field of view were reduced by half, to 45˚, leaving all of the other parameter settings the same? Determine the points, in the image plane, to which each of the cube vertices would be projected and sketch the result to scale. Please clearly label the coordinates to avoid ambiguity.



d. (2 points) Briefly describe what you notice. When looking at two cube faces that are equal sizes in reality (e.g. front and back) does one appear smaller than the other when one is more distant from the camera than the other? When looking at two objects that are resting on a common horizontal groundplane, does the groundplane appear to be tiled in the image, so that the objects that are farther away appear to be resting on a base that is higher as their distance from the camera increases? What changes do you observe in the relative heights, in the image, of the smaller and larger cubes as the camera position changes? Is there a point at which the camera could be so close to the smaller cube (but not touching it) that the larger cube would be completely obscured in the camera's image? Based on these insights, what can you say about the idea to create an illusion of "getting closer" to an object in a photographed scene by zooming in on the image and cropping it so that the object looks bigger?

8. Consider the perspective projection-normalization matrix **P** which maps the contents of the viewing frustum into a cube that extends from –1 to 1 in *x*, *y z* (called normalized device coordinates):

$$\mathbf{P} = \begin{bmatrix} \frac{2\cdot near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2\cdot near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -\frac{2\cdot far\cdot near}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Suppose you want to define a square, symmetric viewing frustum with a near clipping plane located 0.5 units in front of the camera, a far clipping plane located 20 units from the front of the camera, a 60˚ vertical field of view, and a 60˚ horizontal field of view.

a. (2 points) What are the entries in **P**?
b. (3 points) How should be matrix **P** be re-defined if the viewing window is re-sized to be twice as tall as it is wide?
c. (3 points) What are the new horizontal and vertical fields of view after this change has been made?

When the viewing frustum is known to be symmetric, we will have *left* = –*right* and *bottom* = –*top*. In that case, an alternative definition can be used for the perspective projection matrix where instead of defining parameters *left*, *right*, *top*, *bottom*, the programmer instead specifies a vertical field of view angle and the aspect ratio of the viewing frustum.

$$\mathbf{P}_{alt} = \begin{bmatrix} \left(\frac{height}{width}\right)\cot\left(\frac{\theta_v}{2}\right) & 0 & 0 & 0 \\ 0 & \cot\left(\frac{\theta_v}{2}\right) & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -\frac{2\cdot near\cdot far}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

d. (1 point) What are the entries in $\mathbf{P}_{\mathbf{alt}}$ when the viewing frustum is defined by: a near clipping plane located 0.5 units in front of the camera, a far clipping plane located 20 units from the front of the camera, a 60˚ vertical field of view, and a square aspect ratio?

- **e.** (1 points) Suppose the viewing window is re-sized to be twice as wide as it is tall. How might you re-define the entries in $\mathbf{P}_{\mathbf{alt}}$?
- **f.** (2 points) What would the new horizontal and vertical fields of view be after this change has been made? How would the image contents differ from when the window was square?
- **g.** (1 points) Suppose the viewing window is re-sized to be twice as tall as it is wide. How might you re-define the entries in $\mathbf{P}_{\mathbf{alt}}$?
- **h.** (2 points) What would the new horizontal and vertical fields of view be after this change has been made? How would the image contents differ from when the window was square?
- **i.** (1 points) Suppose you wanted the user to be able to see more of the scene in the vertical direction as the window is made taller. How would you need to adjust $\mathbf{P}_{\mathbf{alt}}$ to achieve that result?

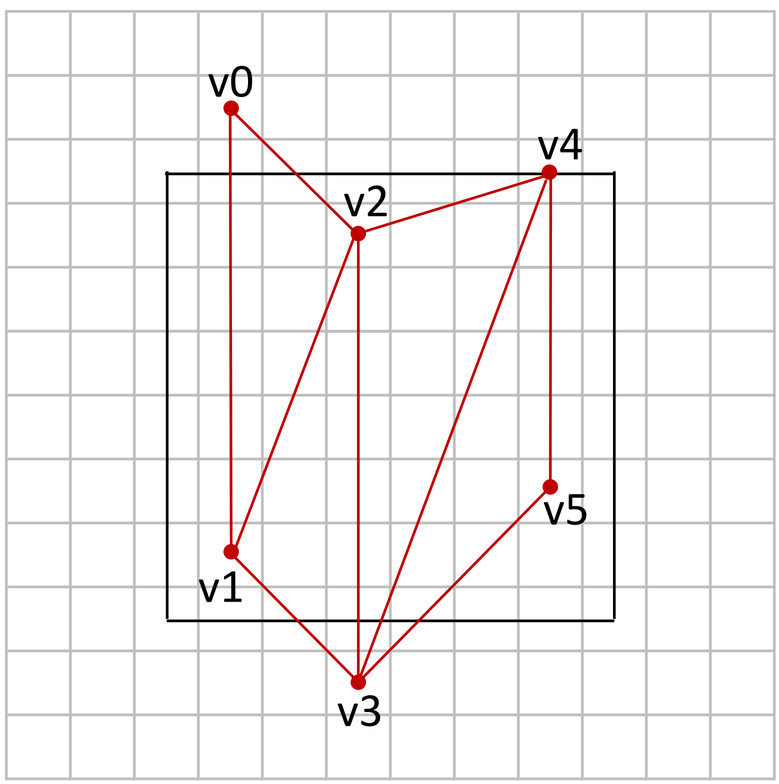
Clipping

9. Consider the triangle whose vertex positions, after the viewport transformation, lie in the centers of the pixels: $p_0 = (3, 3)$, $p_1 = (9, 5)$, $p_2 = (11, 11)$.

- **a.** (6 points) Define the edge equations and tests that would be applied, during the rasterization process, to each pixel (x, y) within the bounding rectangle $3 \leq x \leq 11$, $3 \leq y \leq 11$ to determine if that pixel is inside the triangle or not.
- **b.** (3 points) Consider the three pixels $p_4 = (6, 4)$, $p_5 = (7, 7)$, and $p_6 = (10, 8)$. Which of these would be considered to lie inside the triangle, according to the methods taught in class?

10. When a model contains many triangles that form a smoothly curving surface patch, it can be inefficient to separately represent each triangle in the patch independently as a set of three vertices because memory is wasted when the same vertex location has to be specified multiple times. A triangle strip offers a memory-efficient method for representing connected 'strips' of triangles. For example, in the diagram below, the six vertices v0 .. v5 define four adjacent triangles: (v0, v1, v2), (v2, v1, v3), (v2, v3, v4), (v4, v3, v5). [Notice that the vertex order is switched in every other triangle to maintain a consistent counter-clockwise orientation.] Ordinarily one would need to pass 12 vertex locations to the GPU to represent this surface patch (three vertices for each triangle), but when the patch is encoded as a triangle strip, only the six vertices need to be sent and the geometry they represent will be interpreted using the correspondence pattern just described.

(5 points) When triangle strips are clipped, however, things can get complicated. Consider the short triangle strip shown below in the context of a clipping cube. After the six vertices v0 .. v5 are sent to be clipped, what will the vertex list be after clipping process has finished? How can this new result be expressed as a triangle strip? (Try to be as efficient as possible) How many triangles will be encoded in the clipped triangle strip?



Ray Tracing vs Scan Conversion

11. (8 points) List the essential steps in the scan-conversion (raster graphics) rendering pipeline, starting with vertex processing and ending with the assignment of a color to a pixel in a displayed image. For each step briefly describe, in your own words, what is accomplished and how. You do not need to include steps that we did not discuss in class, such as tessellation (subdividing an input triangle into multiple subtriangles), instancing (creating new geometric primitives from existing input vertices), but you should not omit any steps that are essential to the process of generating an image of a provided list of triangles.

12. (6 points) Compare and contrast the process of generating an image of a scene using ray tracing versus scan conversion. Include a discussion of outcomes that can be achieved using a ray tracing approach but not using a scan-conversion approach, or vice versa, and explain the reasons why and why not.