CS770/870 Assignment 6

- Due: Monday, November 1st, 2021

Ray Casting

For this assignment, you will compute the geometric steps in a ray caster. It will give you experience needed when implementing a ray tracer in upcoming assignments.

The Scene and Camera

Consider a scene with the following two objects:

- there is a sphere with these parameters:
  - center C = (1 1 0)
  - radius R = 2
- there is a triangle with these vertices:
  - A = (0 -2 2)
  - B = (2 2 2)
  - C = (-2 2 2)

There is a perspective camera, with the following parameters:

- eye = (0 0 4)
- lookat = (0 0 0)
- vup = (0 1 0)

The image plane has these parameters:

- (clipLeft, clipRight) = (-2, 2)
- (clipBottom, clipTop) = (-1, 1)
- clipNear = 2

The image plane is divided into

- W = 301 pixels across
- H = 201 pixels up-down

Exercises

Consider a pixel with P_DCS = (x_DCS y_DCS) = (150 100). Perform the following calculations. Then find the object that the ray (which starts at the pixel) hits first.

1. Transform the pixel P_DCS from DCS (device coordinates, also called pixel coordinates) to VCS (view coordinates, also called camera coordinates). Obtain P_VCS, and print it.

2. Obtain the three basis vectors of the VCS system, *X_vcs*, *Y_vcs*, and *Z_vcs*, and print them, on separate lines.

3. Compute M_VCS_to_WCS, the inverse of the view matrix, and print it.

4. Use this matrix to obtain P_WCS, the pixel's world coordinates, and print it.

5. Obtain a unit-length vector V in the direction from the eye to the pixel, and print it.

6. You now have a ray $P = S + tV$, where S is P_WCS. Compute the point *P_s* where the ray hits the sphere. The sphere's equation is

   $$(P - C) \cdot (P - C) = R^2$$

   To do this, substitute the ray's equation P = S + tV in to the sphere's equation. You should get a quadratic equation in t.
   - solve for t
   - check that t is positive
   - if there are 2 positive roots, pick the smaller one
   - plug back into *P_s* = $S + tV$ to get the hit point

   For this question, print each of the following on a separate line:
   - the quadratic's coefficients, a, b, c,
   - the discriminant d,
   - the ray parameter t, and
   - the hit point.

7. Get the surface normal at the hit point: N = *P_s* - C. Normalize this vector, and print it.

8. Now compute the point *P_t* where the ray hits the triangle, as follows.

   - a point *P_t* on the triangle will have barycentric coordinates (u v), and will obey this equation: *P_t* = *A + u(B-A) + v(C-A)*. The barycentric coordinates must obey:
     - 0 ≤ u ≤ 1
     - 0 ≤ v ≤ 1

      - u+v ≤ 1

  - The point P_t will also be on the ray: $P\_t = S + tV$
  - Join the equations: $A + u(B\text{-}A) + v(C\text{-}A) = S + tV$
  - This is actually three equations (one for x, one for y, and one for z). Obtain a 3x3 linear system of equations for t, u, and v.
  - Solve the equations. There is true hit if t is positive, and both u and v obey the barycentric conditions.

  For this question, print each of the following on a separate line:
  - The matrix A,
  - the right-hand side of the matrix equation,
  - the values of t, u, and v, and
  - the hit point.

9. Get the surface normal of the triangle: $N = (B\text{-}A) \times (C\text{-}A)$. Normalize this vector, and print it.
10. Both the sphere's hit point and the triangle's hit point have an associated ray distance t. Chose the point with smaller t, and record its position, and normal vector. Print one of the following:
    - `ray hits triangle first`
    - `ray hits sphere first`

Programming

Write a `C++` program using the `glm` library that writes all the answers to `stdout`. Use the `to_string()` function to make printable versions of your `vec3`, `vec4`, and `mat4` objects. Each answer should have an output line to identify it. For example:

```
std::cout << "Question 1:\n";
std::cout << "Pixel in VCS:  " << glm::to_string(P_vcs) << "\n";
```

Don't submit your program.

Turn Your Work In

When you are finished, go to `mycourses.unh.edu`, find assignment 6, click the "Submit" button, and upload `ray_cast.cpp`.