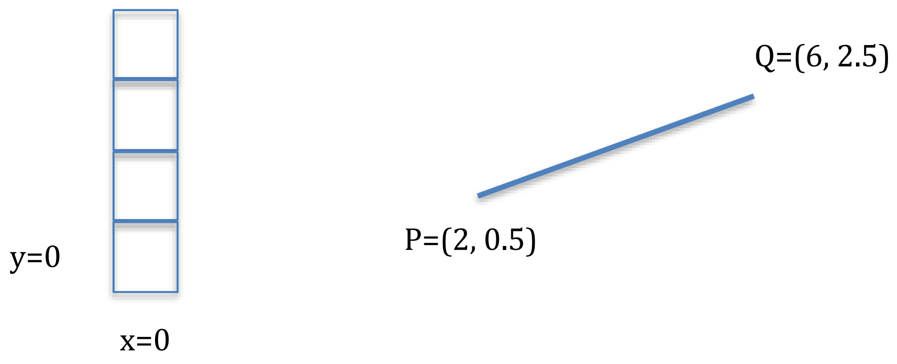
CS 419: Production Rendering Fall 2018
In-class Worksheet 0

Ray-Tracing in Flatland

1. Complexity is hard. So let's imagine we live in a 2D world with only two colors…let's say magenta and cyan. We want to create a digital image of an imaginary scene populated by line segments. How can we write code to do that?



We place an imaginary view plane in our imaginary scene. Suppose we shoot rays from the center of each pixel in the direction orthogonal to the view plane.

a. What is a parametric equation for the rays?

r(t) = o + dt where **o** is the pixel center and **d** is the vector (1,0)

b. What is a parametric equation for the line segment PQ?

l(t)=(1-t)P+(t)Q=P+t(Q-P)

c. How can we check if a ray intersects a line segment?

We expand the parametric equations out into separate expression for the x and y coordinates. Then, solve a system of two linear equations:

$x_o+x_d t_r=x_p+t_l(x_q-x_p)$
$y_o+y_d t_r=y_p+t_l(y_q-y_p)$

d. Perform this test for the pixel at (0,0)…what is the result?

From the second equation:
$1/2+0t_r=1/2 + t_l$ (2)
$t_l=0$

So the ray hits (grazes?) the bottom of the line
We'll call this a hit

e. Assume the line segment is a Magenta colored surface and the scene background is Cyan. What image gets produced?

(0,3)=cyan
(0,2)=magenta
(0,1)=magenta
(0,0)=magenta

2. How would you change this system to render using perspective projection?

The ray origins would be a single pint behind the viewplane