ECE 66100 Homework #1

by

Adrien Dubois (dubois6@purdue.edu)

October 23, 2025

Problem 1:

The following lemma will be used for the solution to this problem.

Lemma 1:

Let the vector $\vec{\boldsymbol{v}}$ be the homogeneous representation of the physical point $\vec{\boldsymbol{x}}$. It is known that the 3D vector $\vec{\boldsymbol{v}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \in \mathbb{R}^3$ is the homogeneous coordinate representation of a physical 2D point in $(x, y) \in \mathbb{R}^2$.

Lemma 2:

Using lemma 1, we note that every $k \times \vec{\boldsymbol{v}} = k \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, k \in \mathbb{R}, k \neq 0$ is the same physical point $(x, y) \in \mathbb{R}^2$. We can therefore denote a new general representation for the homogeneous coordinate form a physical point in $\mathbb{R}^2$:

$$\vec{\boldsymbol{v}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ where } x = \frac{u}{w} \text{ and } y = \frac{v}{w}$$

Solution:

We can represent the point at origin in the physical space $\mathbb{R}^2$ by:

$$\vec{\boldsymbol{x_0}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Therefore, using lemma 1 we can convert $\vec{\boldsymbol{x_0}}$ into its homogeneous coordinate representation:

$$\vec{\boldsymbol{x_0}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Lastly, by corollary from lemma 2 and the previous result, we can find the generalized version of this point using lemma 2. These are all the points in the homogeneous coordinate system in $\mathbb{R}^3$ that represent the origin the physical space:

$$\vec{\boldsymbol{x_0}} = \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \text{ where } w \in \mathbb{R} \text{ and } w \neq 0$$

Problem 2:

The following definitions will be utilized in the solution to problem 2.

Definition 1: Representation of a line in homogeneous coordinates

In the physical space $\mathbb{R}^2$, we can represent an arbitrary line $\boldsymbol{line_1}$ as follows:

$$\boldsymbol{line_1} : ax + by + c = 0 \text{ for some arbitrary values: } a, b, c \in \mathbb{R}$$

This line will therefore have a slope of $\frac{-a}{b}$ and will intercept the y-axis at point: $(0, \frac{-c}{b})$.
Next, considering the vector representation of the line's parameters:

$$\vec{\boldsymbol{l_1}} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

we can represent the algebraic equation of the line $\boldsymbol{l_1}$ in the homogeneous coordinate system as follows:

$$\boldsymbol{line_1} : \vec{\boldsymbol{l_1}}^T x = 0 \text{ or } \vec{\boldsymbol{l_1}} x^T = 0 \text{ where } x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Lemma 3: The intersection of 2 lines in HC

Given two lines represented in homogeneous coordinates: $l_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix}$ and $l_2 = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix}$, the point of intersection (x) between the two lines is their vector cross product: $x = l_1 \times l_2$.

Solution:

We can define a point at infinity by the point of intersection of two different, parallel lines $\boldsymbol{line_1}$ and $\boldsymbol{line_2}$ parameterized by the following vectors:

$$\vec{\boldsymbol{l_1}} = \begin{bmatrix} a \\ b \\ c_1 \end{bmatrix} \text{ and } \vec{\boldsymbol{l_2}} = \begin{bmatrix} a \\ b \\ c_2 \end{bmatrix}$$

We can confirm that these two lines are different and parallel since they have equal slopes: $m = \frac{-a}{b}$ and different y-intercepts: $(0, \frac{-c_1}{b}$ and $\frac{-c_2}{b})$
Using lemma 3, we can now calculate the point of intersection (x) between $\boldsymbol{line_1}$ and $\boldsymbol{line_2}$:

$$x = \vec{\boldsymbol{l_1}} \times \vec{\boldsymbol{l_2}} = \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix}$$

Now consider two new, different and parallel lines $\boldsymbol{line_3}$ and $\boldsymbol{line_4}$ with the following representations in H.C.:

$$\vec{\boldsymbol{l_3}} = \begin{bmatrix} \alpha \\ \beta \\ c_3 \end{bmatrix} \text{ and } \vec{\boldsymbol{l_4}} = \begin{bmatrix} \alpha \\ \beta \\ c_4 \end{bmatrix} \text{ where } \alpha x + \beta y \neq k(ax + by), k \in \mathbb{R}, k \neq 0$$

Following the previous reasoning, these lines will also meet at infinity. This point of intersection will be:

$$x' = \vec{\boldsymbol{l_3}} \times \vec{\boldsymbol{l_4}} = \begin{bmatrix} \beta \\ -\alpha \\ 0 \end{bmatrix} \neq \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix}$$

Hence, it has been showed that not all points at infinity in the physical plane $\mathbb{R}^2$ are the same. Instead, these points all lie on the same line at infinity in the homogeneous coordinate system, and the direction at which they approach infinity is controlled by the slope of the line parameterized by their coordinate in HC.

Problem 3:

Definition 2: General Conic in HC

It is known that the representation of a conic in the physical space $\mathbb{R}^2$ is:

$$\textbf{Equation 1: } ax^2 + bxy + cy^2 + dx + ey + f = 0$$

Following the procedure in **Definition 1**, we can express the previous equation in homogeneous coordinates by representing the point (x, y) as $\begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix}$ where $x = \frac{x_1}{x_3}$ and $y = \frac{x_2}{x_3}$. Using this representation, we can rewrite **Equation 1** as follows:

$$\textbf{Equation 2: } ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

Which is the same as the following vector-matrix product:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{2}{2} & f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

Therefore, every conic must satisfy the equation: $\vec{\boldsymbol{x}}^T \boldsymbol{C} \vec{\boldsymbol{x}} = 0$ where C is the non-singular, symmetrical matrix shown above.

Definition 3: Degenerate Conic in HC

A degenerate conic is a subset of the general conic that occurs when the slice a double cone with a plane that passes through its axis. This will result in two lines which we will denote: l and m

Since every point on the degenerate conic C is on l or lm, the definition of C in HC is given by:

$$\boldsymbol{C} = \vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T + \vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T$$

Lemma 4: Rank of the sum of two matrices

It is known that if **A** and **B** are two $m \times n$ matrices, then rank(**A** + **B**) $\leq$ rank(**A**) + rank(**B**).

Solution

From **Definition 2**, we know that the general definition of a conic C is: $\vec{\boldsymbol{x}}^T \boldsymbol{C} \vec{\boldsymbol{x}} = 0$

Using definition 3, we know that the degenerate conic C can be defined by $\boldsymbol{C} = \vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T + \vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T$, where:

$$\vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T = \begin{bmatrix} l_1m_1 & l_1m_2 & l_1m_3 \\ l_2m_1 & l_2m_2 & l_2m_3 \\ l_3m_1 & l_3m_2 & l_3m_3 \end{bmatrix} \text{ and } \vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T = \begin{bmatrix} l_1m_1 & l_2m_1 & l_3m_1 \\ l_1m_2 & l_2m_2 & l_3m_2 \\ l_1m_3 & l_2m_3 & l_3m_3 \end{bmatrix}$$

It is trivial to see that the rows and columns of both $\vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T$ and $\vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T$ are linearly dependent. Therefore, by the definition of the rank function:

$$Rank(\vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T) = 1 \text{ and } Rank(\vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T) = 1$$

.

Lastly, by **Lemma 4**, we can determine that:

$$Rank(C) = Rank(\vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T + \vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T) \leq Rank(\vec{\boldsymbol{l}}\vec{\boldsymbol{m}}^T) + Rank(\vec{\boldsymbol{m}}\vec{\boldsymbol{l}}^T) = 2$$

Problem 4:

As stated in **Equation 2**, a conic can be defined as follows:

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

We have also shown in **Definition 2** that the previous equation can be represented as such:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

From problem 4, we know that a conic will have: $0 < Rank(C) \leq 3$, $Rank(C) \in \mathbb{R}$. Therefore, we can determine that at least one of a, c or f will be non-zero. So, it can be factored out of the equation without affecting the conic since:

$$\vec{\boldsymbol{x}}^T(k\boldsymbol{C})\vec{\boldsymbol{x}} = 0$$

$$k(\vec{\boldsymbol{x}}^T\boldsymbol{C}\vec{\boldsymbol{x}}) = 0$$

$$\vec{\boldsymbol{x}}^T\boldsymbol{C}\vec{\boldsymbol{x}} = \frac{0}{k} = 0 \text{ for } k \in \mathbb{R}, k \neq 0$$

Applying this to the algebraic form of the conic C, we get:

$$a(x_1^2 + \frac{b}{a}x_1x_2 + \frac{c}{a}x_2^2 + \frac{d}{a}x_1x_3 + \frac{e}{a}x_2x_3 + \frac{f}{a}x_3^2) = 0$$

$$x_1^2 + \frac{b}{a}x_1x_2 + \frac{c}{a}x_2^2 + \frac{d}{a}x_1x_3 + \frac{e}{a}x_2x_3 + \frac{f}{a}x_3^2 = 0$$

This is an equation in 5 variables, so we would need 5 points to solve it. Therefore, at least 5 points are required to define a conic.

Problem 5:

Step 1: Deriving the HC representation for line 1

Since it is known that $\boldsymbol{l_1}$ passes through the points (0,0) and (3,4), we can find the definition of the line in homogeneous coordinates through the vector cross product of points 1 and 2:

$$\boldsymbol{l_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \\ 0 \end{bmatrix}$$

Step 2: Deriving the HC representation for line 2

Following the same procedure, we can represent line 2 through the following forms:

$$\boldsymbol{l_2} = \begin{bmatrix} -1 \\ 4 \\ 1 \end{bmatrix} \times \begin{bmatrix} \frac{3}{2} \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5/2 \\ -5 \end{bmatrix}$$

Step 3: Find the intersection between both lines

Lastly, the intersection point between $\boldsymbol{l_1}$ and $\boldsymbol{l_2}$, $\vec{\boldsymbol{x}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$, can be calculated through the vector cross product of both lines:

$$\vec{\boldsymbol{x}} = \begin{bmatrix} -4 \\ 3 \\ 0 \end{bmatrix} \times \begin{bmatrix} 5 \\ \frac{5}{2} \\ -5 \end{bmatrix} = \begin{bmatrix} -15 \\ -20 \\ -25 \end{bmatrix}$$

Part 2:

If we change the definition of the first line to instead pass through: (-1, 2) and (1, 2), we would only need 2 steps to find their intersection:

Step 1: Definition of line 1

$$\boldsymbol{l_1} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix}$$

Since we are operating in HC, multiplying this line by the scalar multiple $k = \frac{5}{4}$ would not change the line itself. Therefore, we can say that

$$\boldsymbol{l_1} = \begin{bmatrix} 5 \\ 5/2 \\ 0 \end{bmatrix}$$

Step 2: Definition of line 2

From the previous solution, we know that the HC representation of line 2 is:

$$\boldsymbol{l_2} = \begin{bmatrix} 5 \\ 5/2 \\ -5 \end{bmatrix}$$

So, it is clear that this new $\boldsymbol{l_1}$ will be parallel to $\boldsymbol{l_2}$. Therefore, we can determine the point at which they intersect will be:

$$\vec{\boldsymbol{x}} = \begin{bmatrix} \frac{5}{2} \\ -5 \\ 0 \end{bmatrix}$$

Problem 6:

Definition 4:

It is known that the equation of a standard ellipse is given by:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$

where

- (h,k) is the center of the ellipse
- a is the length of the semi-major axis
- b is the length of the semi-minor axis

Solution:

Using **Definition 4**, we can determine that the algebraic equation of the given ellipse is:

$$\frac{(x-1)^2}{1} + \frac{(y-4)^2}{4} = 1$$

Expanding this out, we get the following algebraic equation, and its corresponding matrix in HC following the process outlined in **Equation 2**:

$$x^2 - 2x + \frac{y^2}{4} - 2y + 4 = 0$$

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & \frac{1}{4} & -1 \\ -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

Next, we can convert the point $\boldsymbol{p}$ from the physical space $\boldsymbol{p} = (0,0)$ to HC $\boldsymbol{p} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

So, using the information in the prompt, we can calculate the polar line for $\boldsymbol{C}$ at $\boldsymbol{p}$:

$$\text{Polar line } = \boldsymbol{Cp} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & \frac{1}{4} & -1 \\ -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 4 \end{bmatrix}$$

Converting back to algebraic form, we get the following equation of the polar line:

$$-x - y + 4 = 0$$

Plugging in $x = 0$ and $y = 0$ to the previous equation, we can find the following x and y intercepts in the physical space $\mathbb{R}^2$ for the polar line:

$$\text{y-intercept} = (0,4), \text{x-intercept} = (4,0)$$

Problem 7:

Lemma 5

It is known that given the angle of a line to the x-axis $\boldsymbol{\alpha}$, the slope of that line can be calculated as follows:

$$\text{slope } = tan(\boldsymbol{\alpha})$$

Solution

1. Convert the laser to a line in HC

Using **lemma 5**, we can get the slope of the line created by the laser pointer:

$$\text{Slope } = \frac{-a}{b} = tan(45°) = 1$$

So, since we know that the laser pointer originates from the origin in the physical space, we can say that the HC representation of the line is:

$$\boldsymbol{l_1} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

2. Convert 2 triangle edge to HC

Next, to know if the laser passes through the triangle, it is sufficient to check whether it crosses any 2 edges within the point-to-point (end-point inclusive) domain. To accomplish this, we first need use **definition 1** to calculate the HC representation of 2 lines from the triangle:

$$\text{Edge Line 1}: y = \left(\frac{3-5}{5-3}\right)x + \left(5 - \left(\frac{3-5}{5-3}\right)3\right) = -1x + 8 \text{ or, in HC: } e_{tri1} = \begin{bmatrix} 1 \\ 1 \\ -8 \end{bmatrix}$$

$$\text{Edge line 2}: y = \left(\frac{5-3}{7-5}\right)x + \left(5 - \left(\frac{5-3}{7-5}\right)3\right) = x - 2 \text{ or, in HC: } e_{tri2} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$

3. Find the intersection between the laser and corresponding edges

Following the procedure in problem 5, we can determine that the laser will intersect with edge 1 at the point:

$$p_4 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -8 \end{bmatrix} = \begin{bmatrix} -8 \\ -8 \\ -2 \end{bmatrix} \text{ which corresponds to: } \left(\frac{-8}{-2}, \frac{-8}{-2}\right) = (4, 4) \text{ in the physical space.}$$

and the laser will intersect edge 2 at:

$$p_5 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} \text{ which corresponds to a point at infinity in the physical space.}$$

Ensure that the intersections occur within the domain of the triangle

Since we know that p_4 and p_5 intersect edges 1 and 2 respectively, we can create a bounding box around the triangle to determine whether the intersections occur within this area. Considering $p_1 = (3, 5), p_2 = (5, 3)$ and $p_3 = (7, 5)$, this will require in four comparisons for each intersection. Using point 4 as an example:

$$min(x_1, x_2) < x_4 < max(x_1, x_2) \text{ and } min(y_1, y_2) < y_4 < max(y_1, y_2)$$

Therefore, since $3 < 4 < 5$ and $3 < 4 < 5$, we can determine that the laser intersects with the triangle by splitting the first edge created by the line between points 1 and 2.

To help visualize this I have included the following graphic with the relevant points (labeled p1, p2, p3, and p4), triangle edges (in solid black), the laser (dotted red line) and the bounding box (dotted blue line) for comparison:

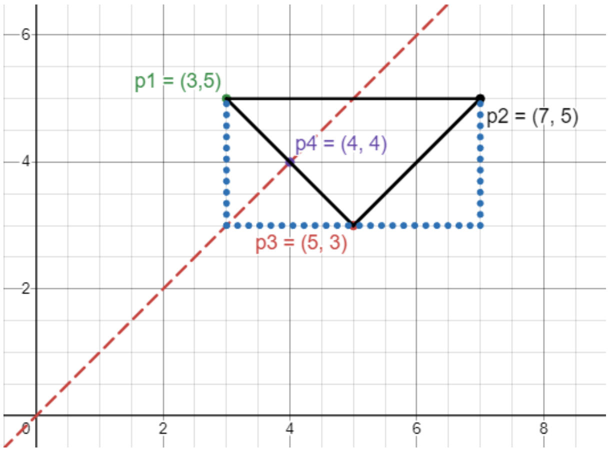

Figure 1: Visual representation of the laser aim game, with relevant points, lines and edges

Bonus Problem:

For the solution to the bonus problem, I have first included some example graphs generated by my code, and then I have included a printout of the code used for this solution afterwards. The demonstrations used the following inputs:

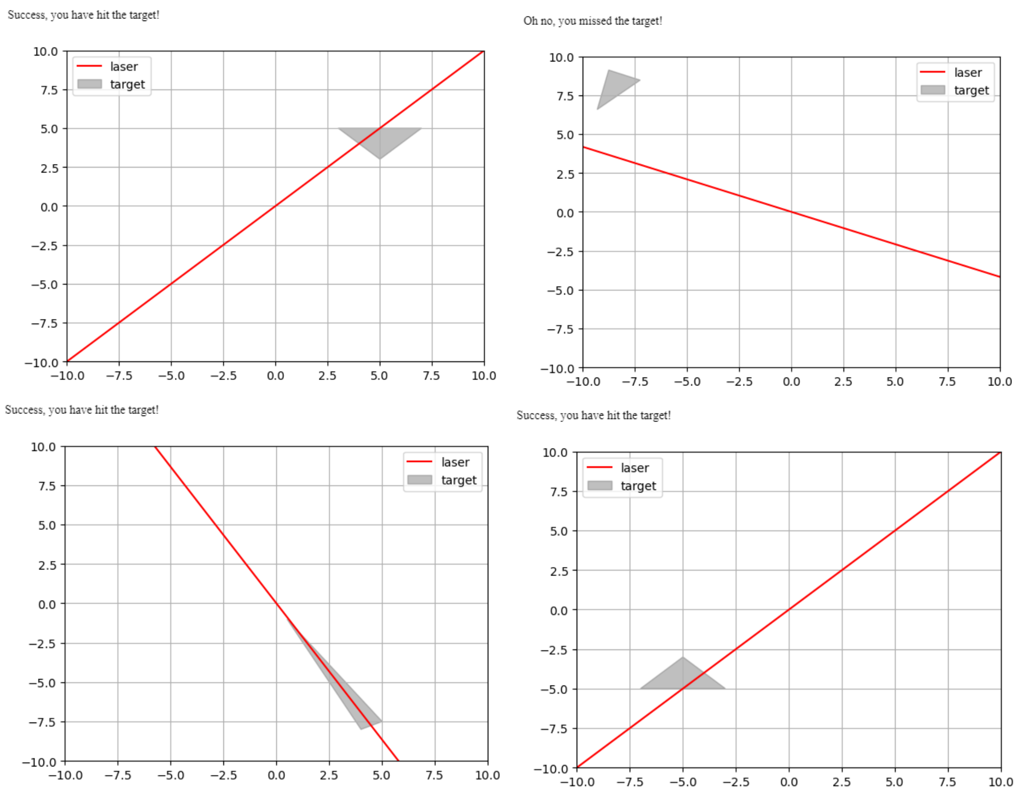

Figure 2: Python plots of the triangle and relevant laser, with a message informing the user if they hit or missed the target

Code printout:

```python
#################################################################
# Author: Adrien Dubois
# Date: 08/28/24
# Subject: ECE661 HW1
#################################################################

import numpy as np
import sys
import math
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon

class Point:
    def __init__(self, x, y):
        """Defines a point using its physical space coordinates"""
        self.x = x
        self.y = y
        self.hc = self.get_hc()
    @classmethod
    def from_hc(cls, hc):
        """Defines a point from its representation in homogeneous coordinates"""
        # If the point is at infinity hc[3] == 0, then don't do the normalization
        if np.isclose(hc[2],0):
            x = hc[0]
            y = hc[1]
        else:
            x = hc[0] / hc[2]
            y = hc[1] / hc[2]
        return cls(x, y)
    def get_hc(self):
        """Returns the point in homogeneous coordinates"""
        return np.array([self.x, self.y, 1])
    def __str__(self):
        """To string method for debugging"""
        return f"Point(x={self.x}, y={self.y}, hc={self.hc})"

class Line():
    def __init__(self, point1, point2):
        """Defines a line that passes through 2 points in the physical space"""
        assert isinstance(point1, Point) and isinstance(point2, Point), "A line should be
created by 2 Points, or by its angle to the x-axis"
        self.hc = self.get_hc(point1, point2)
    @classmethod
    def from_angle(cls, angle, y_int):
        """Defines a line by its angle to the x-axis and y intercept"""
        intercept = Point(0, y_int) # Point1 is defined by the y-intercept
        point2 = Point(1, y_int + math.tan(math.radians(angle))) # Point 2 is the point on the
line at x=1
        return cls(intercept, point2)
    def get_hc(self, point1, point2):
        """Returns the line in homogeneous coordinates"""
        slope = (point2.y - point1.y) / (point2.x - point1.x)
        c = point2.y - slope * point2.x
        return np.array([-slope, 1, -c])
    def __str__(self):
        return f"Line(algebraic={self.hc[1]} * y = {-self.hc[0]} * x + {-self.hc[2]}, hc=
{self.hc})"

def create_printout(laser, triangle_points):
    # The following visualization was inspired by the resource included below:
    # https://stackoverflow.com/questions/44397105/how-to-draw-a-triangle-using-matplotlib-
pyplot-based-on-3-dots-x-y-in-2d

    # Generate values for the laser
    x_values = np.linspace(-10, 10, 400)
    y_values = -laser.hc[0] * x_values - laser.hc[2]
```

```
    # Define the vertices of the triangle
    triangle_vertices = np.array([triangle_points[0].hc[:2], triangle_points[1].hc[:2],
triangle_points[2].hc[:2]])

    # Generate the plot with the laser and triangle
    fig, ax = plt.subplots()
    ax.plot(x_values, y_values, "r", label="laser")
    triangle = Polygon(triangle_vertices, closed=True, color="grey", alpha=0.5,
label="target")
    ax.add_patch(triangle)

    # Organize visualization
    ax.set_xlim(-10, 10)
    ax.set_ylim(-10, 10)
    ax.grid(True)
    ax.legend()
    plt.show()

if __name__ == "__main__":
    assert len(sys.argv) == 5, 'The inputs should be as follows: "(x1,y1)" "(x2,y2)" "(x3,y3)"
angle_of_laser_in_degrees.'

    # Read arguments for triangle points & laser angle
    triangle_points = []
    for i in range(1,4):
        point_string = sys.argv[i].strip("()")
        x_str, y_str = point_string.split(",")
        triangle_points.append(Point(float(x_str), float(y_str)))
    aiming_angle = float(sys.argv[4])

    # Create laser line in hc
    laser = Line.from_angle(aiming_angle, 0)

    # Convert 2 triangle edges in hc
    edge1 = Line(triangle_points[0], triangle_points[1])
    edge2 = Line(triangle_points[1], triangle_points[2])

    # Find the intersections:
    intersect1_hc = np.cross(laser.hc, edge1.hc)
    intersect2_hc = np.cross(laser.hc, edge2.hc)
    p_of_int1 = Point.from_hc(intersect1_hc)
    p_of_int2 = Point.from_hc(intersect2_hc)

    # Check that the intersection fits within the domain defined by the triangle
    # If the point of intersection fits, then print and exit, else keep look at the other edge
    if p_of_int1.x < max(triangle_points[0].x, triangle_points[1].x) and p_of_int1.x >
min(triangle_points[0].x, triangle_points[1].x):
        if p_of_int1.y < max(triangle_points[0].y, triangle_points[1].y) and p_of_int1.y >
min(triangle_points[0].y, triangle_points[1].y):
            print("Success, you have hit the target!")
            create_printout(laser, triangle_points)
            exit()
        else:
            pass
    if p_of_int2.x < max(triangle_points[1].x, triangle_points[2].x) and p_of_int2.x >
min(triangle_points[1].x, triangle_points[2].x):
        if p_of_int2.y < max(triangle_points[1].y, triangle_points[2].y) and p_of_int2.y >
min(triangle_points[1].y, triangle_points[2].y):
            print("Success, you have hit the target!")
            create_printout(laser, triangle_points)
            exit()
        else:
            print("Oh no, you missed the target!")
    else:
        print("Oh no, you missed the target!")
    create_printout(laser, triangle_points)
```