Product Requirements Document (PRD)

Project Title:

Rotating Rectangle Python Program

Objective:

Develop a Python program that displays a 2x4 rectangle rotating around its center at the origin. The rectangle lines are blue, corner points red, axes are equal, and the plot title is "Rotate Rectangle" in red. The rotation is performed incrementally using a rotation matrix.

Requirements:

Functional Requirements:

1. Define a rectangle with **width = 4 units** and **height = 2 units**, centered at origin `(0,0)`.
2. Store the rectangle corner points in a vector (NumPy array) in order, and copy the first point as the last to close the rectangle.
3. Rotate the rectangle in **200 steps**, each by **π/128 radians**.
4. For each step:
5. Compute the **new points vector** by multiplying a 4x4 rotation matrix with the current points vector.
6. Update the plot to show the rectangle.
7. Visual representation:
8. Rectangle **lines in blue**.
9. Rectangle **corner points in red**.
10. Coordinate axes have equal scale.
11. Title of the figure: **"Rotate Rectangle"** in red.
12. Introduce a **10 milliseconds delay** between steps to create smooth animation.
13. The 4x4 **rotation matrix** should represent rotation by **π/128 radians** in 2D homogeneous coordinates.

Non-Functional Requirements:

1. Use Python 3.x with **NumPy** and **Matplotlib**.
2. Code should be readable with descriptive variable names.
3. The animation should be smooth and visually clear.

Implementation Steps:

1. Import `numpy` and `matplotlib.pyplot`.
2. Define rectangle dimensions: width=4, height=2.
3. Calculate initial corner points, include first point as last to close rectangle.
4. Define the 4x4 rotation matrix for rotation about the origin by π/128 radians.
5. Set up the plot: axes equal, title in red.
6. Use a loop of 200 steps:
7. Multiply rotation matrix with current points vector.

8. Update rectangle lines and corner points in the plot.
9. Use `plt.pause(0.01)` to create delay.
10. Display final rotated rectangle.

Notes:

- The rectangle is axis-aligned initially.
- The rotation matrix should be in **homogeneous coordinates** to allow matrix multiplication with point vectors.
- Ensure the rectangle remains centered at the origin throughout the animation.