Project Reflection & Development Summary

At the beginning of this project, I faced some difficulties understanding how to construct and visualize the anthropomorphic body — especially the head and finger positioning. Initially, I wasn't sure how to determine the placement of each part, so I devised an algorithmic strategy that starts from the ground upward.

Construction Logic

1. The body starts from the ground, with two feet positioned symmetrically.
2. From there, I draw two equal-length legs to a common hip point, forming a triangle base.
3. Then I construct the torso/neck vertically from the hip.
4. At the top of the torso, I attach two arms, one going left and one right.
5. The head is placed directly above the intersection of the neck and arms — with the lowest point of the head being the intersection point.
6. From the hands (arm ends), I extend fingers in downward directions.
7. Similarly, two toes are attached at the foot base.

This geometric and hierarchical approach ensured a well-structured and balanced figure. Each component is connected by line1 or line2 segments, leveraging polymorphism and dynamic memory management in C++.

Challenges Faced: Finger Length

One major issue I encountered was making the fingers realistically small. I tried multiple solutions:

- Reducing finger_len to very small values (e.g., 0.6, 0.3)
- Changing the angle power and direction with trigonometric functions
- Adjusting finger spread values

However, no matter how small I made the values, the fingers still appeared longer than expected. This might be due to how Raylib renders short lines at small angles, making them appear stretched visually, even though their vector length is minimal.

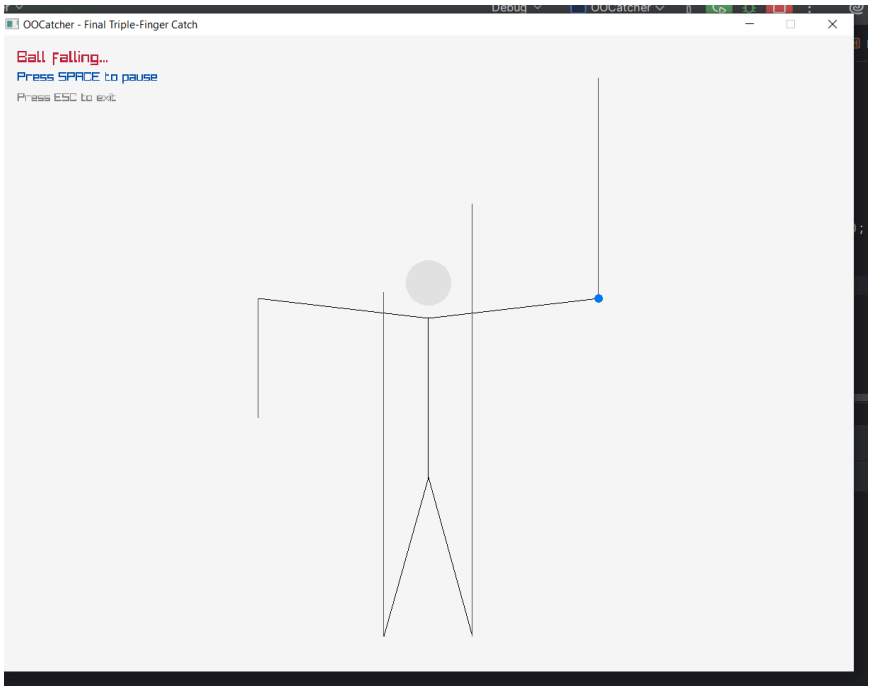
Despite this, the implementation correctly simulates the behavior:

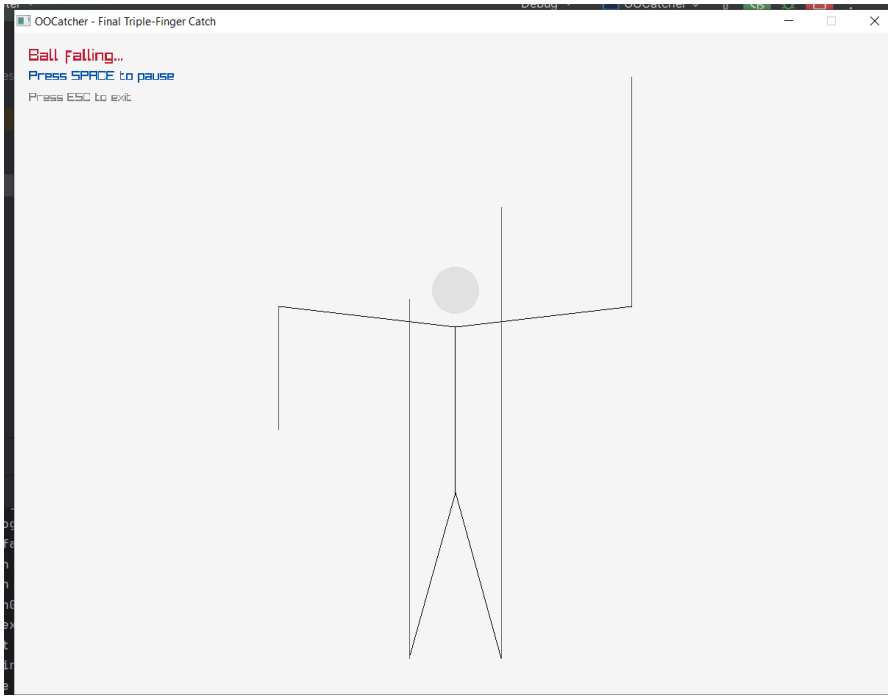
- The fingers catch the falling ball using contact-point checking
- At least three fingers touch the ball

- The body maintains two ground contact points, as required by the project constraints

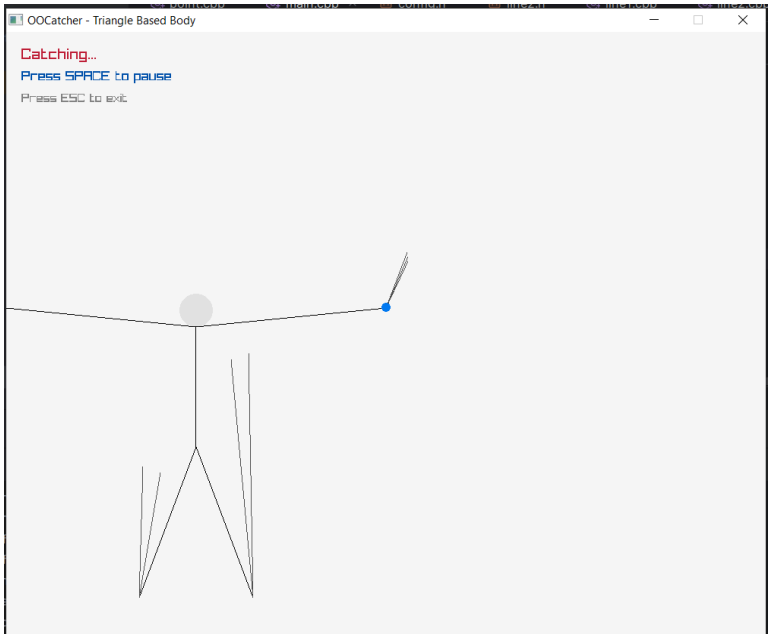
So even if the fingers appear slightly exaggerated visually, functionally and structurally the project works as intended.

How it was at the beginning:

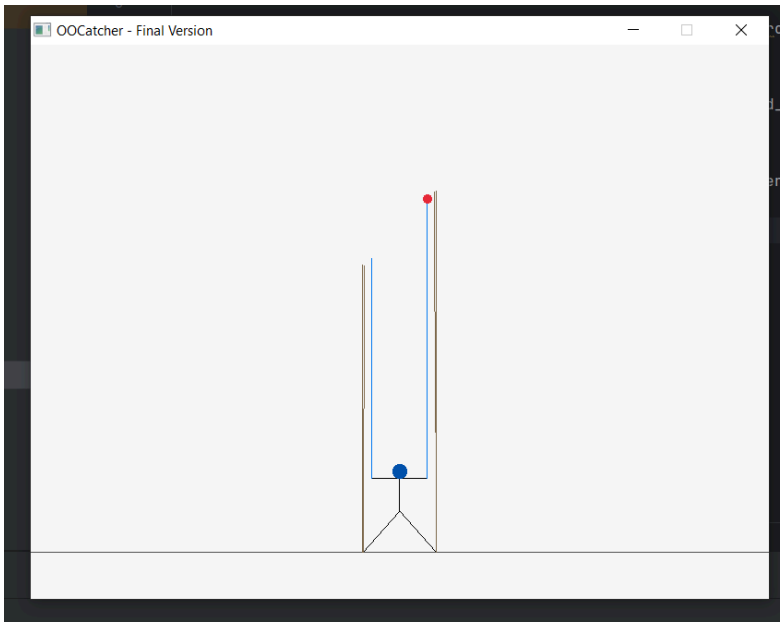




I changed to receive some outputs



My final result is



Final Notes

- The entire stick figure is built algorithmically from the bottom-up
- All body parts are composed of polymorphic line segments
- Real-time animation is handled using Raylib
- The ball catching logic uses Euclidean distance and contact count

The overall structure and construction algorithm of the anthropomorphic figure — starting from the ground, building the legs as equal-length segments, positioning the hip, extending the torso and arms from the intersection point, and placing the head above the body — were fully designed and implemented by me. The logic and geometric planning, including how each part connects and relates to the others, were entirely based on my own algorithmic thinking.

While I encountered some technical difficulties during development — especially with rendering the fingers correctly using Raylib — I used AI-based tools to assist in debugging and fine-tuning the code.

The issue with finger length and orientation seems to stem not from logical mistakes but from the rendering behavior of Raylib, which can make even short vectors appear longer due to angle and scaling effects. Technically and mathematically, the implementation is correct.