



Find wooden chair near table.

Time



LLM Agent

Observation: User is looking for a **wooden chair** near a **table**.

Reasoning: "Need to look for **wooden chairs** within close vicinity of a **table**."

Plan:

1. Call **Target Finder** with **wooden chair** and call **Landmark Finder** with **table**.
2. Compare candidates by their closeness to the landmark **table**.
3. Ask user for further clarification if needed.
4. Repeat until the correct object is found.

Let me check for **wooden chairs** near a **table**.



LLM Agent

### Target Finder

Name: **Chair**  
Attribute: **Wooden**

OpenScene / LERF

Candidate 1

Candidate 2



### Landmark Finder

Name: **Table**  
Relationship: Near

OpenScene / LERF



I think target #1 is what you want.



LLM Agent

Observation: **Target Finder** and **Landmark Finder** returned me coordinates. I also object volume and distance to landmarks for each candidate.

Reasoning: Need to aggregate the information and decide.

Plan:

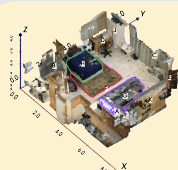
1. Compare **chair** candidates by their closeness to the landmark **table**.
2. Examine the object volume and filter out any object too small.
3. Score, rank and filter each candidate holistically.
4. Make a grounding decision.

Self-critique: I should make a decision based on the holistic situation, taking account all factors into consideration.

Replace



GPT-4o



Observation: User is looking for a **wooden chair** near a **table**.

Reasoning: "Need to look for **wooden chairs** within close vicinity of a **table**."

Plan:

1. Call **Target Finder** with **wooden chair** and call **Landmark Finder** with **table**.
2. Compare candidates by their closeness to the landmark **table**.
3. Ask user for further clarification if needed.
4. Repeat until the correct object is found.