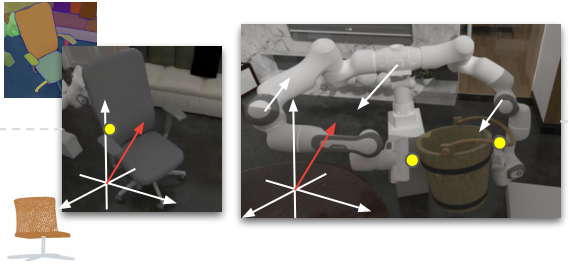

Goal: *push the chair to the marked position*

Environment description

```
class BaseEnv:
  self.chair: ArticulateObject
  self.robot: DualArmPanda
class ArticulateObject:
  self.pose: ObjectPose
  def get_pcd(self) -> np.ndarray[(N,3)]
class DualArmPanda:
  self.ee_coords -> np.ndarray[(4,3)] #
3D positions of 2*2=4 gripper fingers
  ...
```

GPT-4 / Codex ...

Feedback: *keep the chair standing*

User

Dense reward function

```
def compute_reward(self, action):
 ...
# EE approach chair
reward += -dist_ee_to_chair
# Keep chair standing
chair_tilt = np.arccos(z_axis_chair[2])
# Stage reward
if chair_tilt < 0.2 * np.pi:
   if dist_ee_to_chair < 0.1:
     stage_reward += 2
     ...
else:
    stage_reward = -5
return reward
```

RL training

Policy

Rollout

Environment

Expert abstraction