

# CS440/ECE448 Lecture 8: Configuration Space

Mark Hasegawa-Johnson, 2/2020

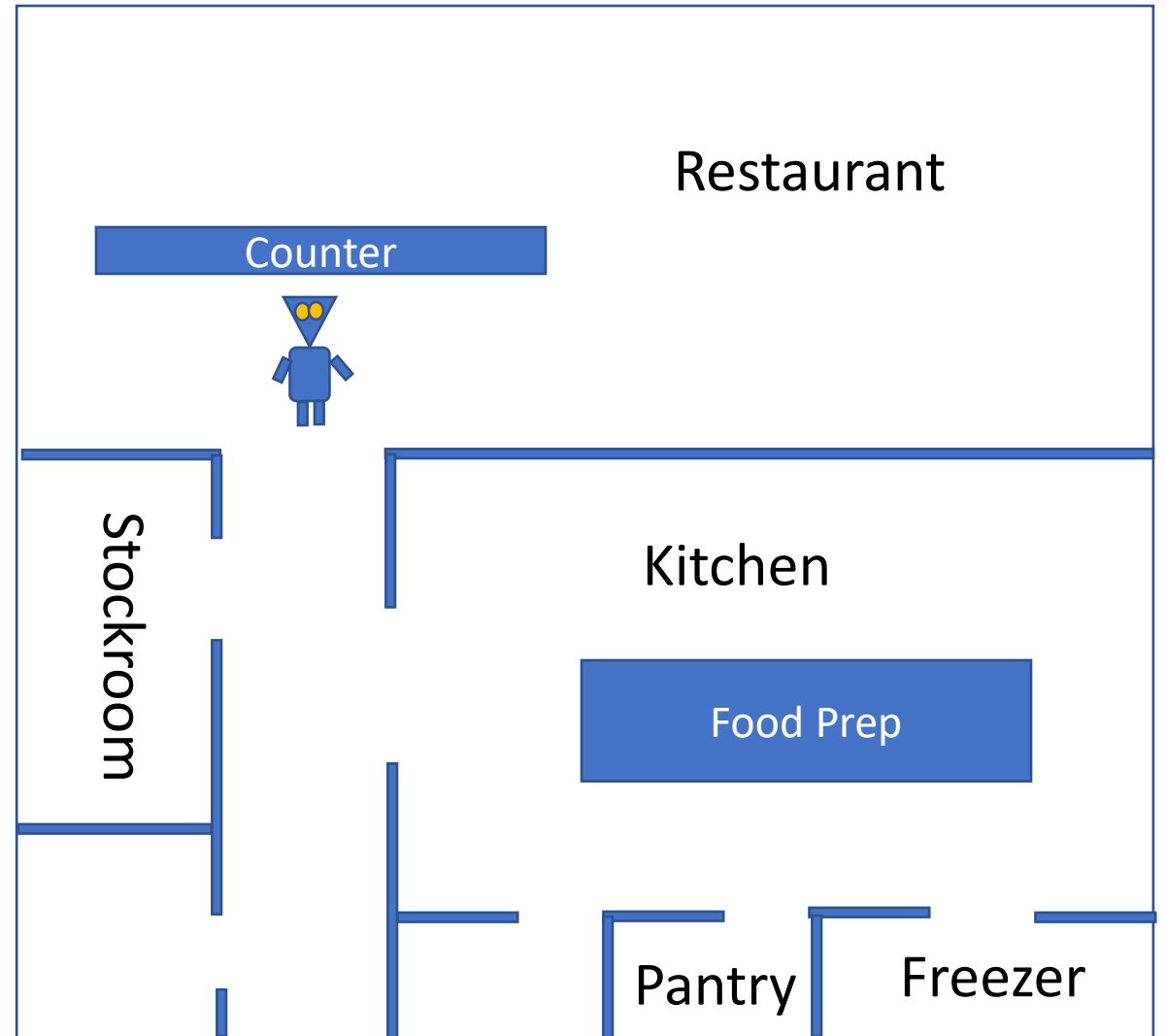
[CC-BY 4.0](#): You are free to: copy and redistribute the material in any medium or format, remix, transform, and build upon the material for any purpose, even commercially, if you give appropriate credit.

# Outline

- Planning = Search
- Poor Robot!
- Configuration Space
- The Robot Arm Problem
- Geometry of the Robot Arm Problem
- Searching for a Solution in Configuration Space

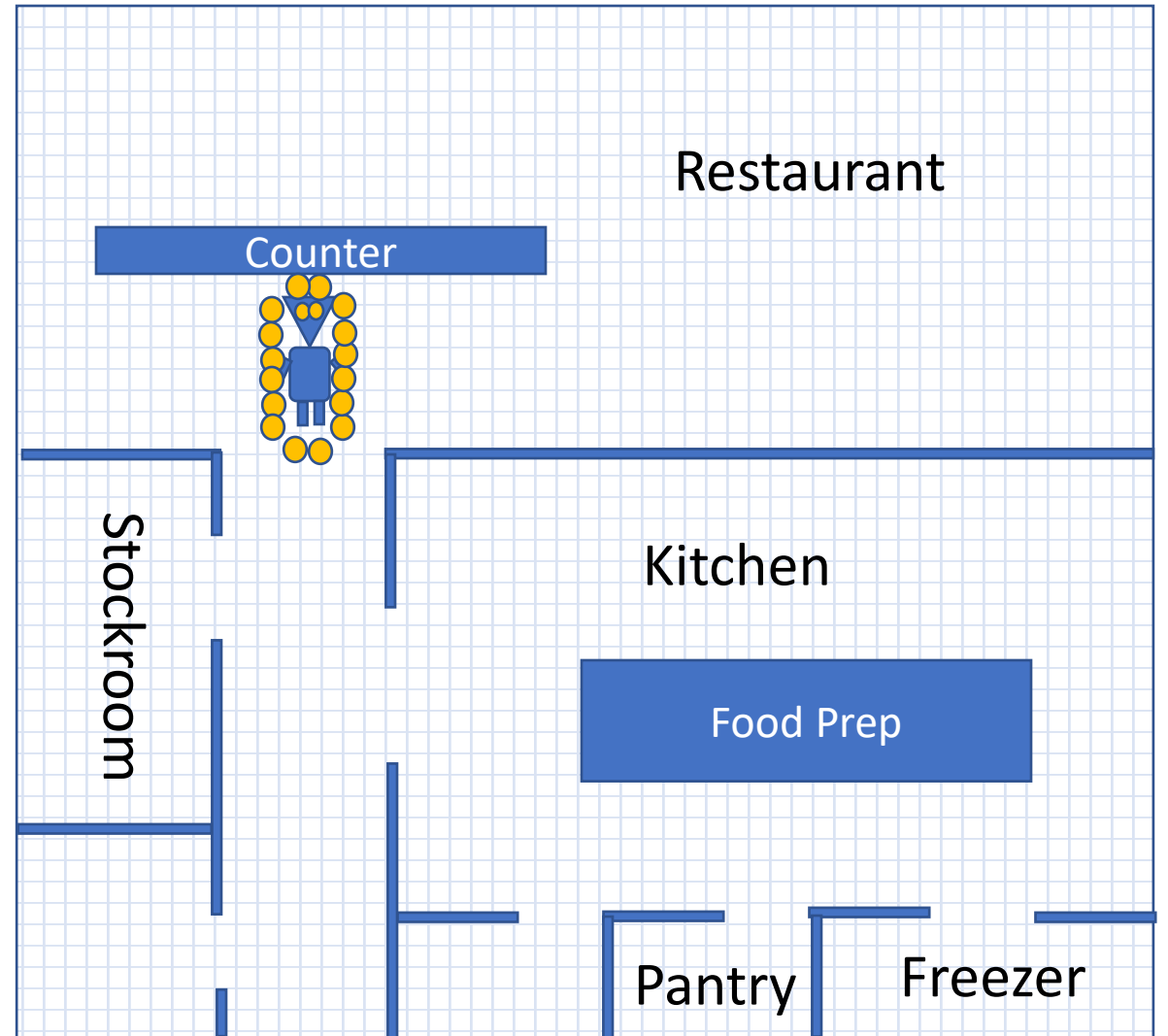
# Planning = search

- The problem: robot needs to get to the stockroom



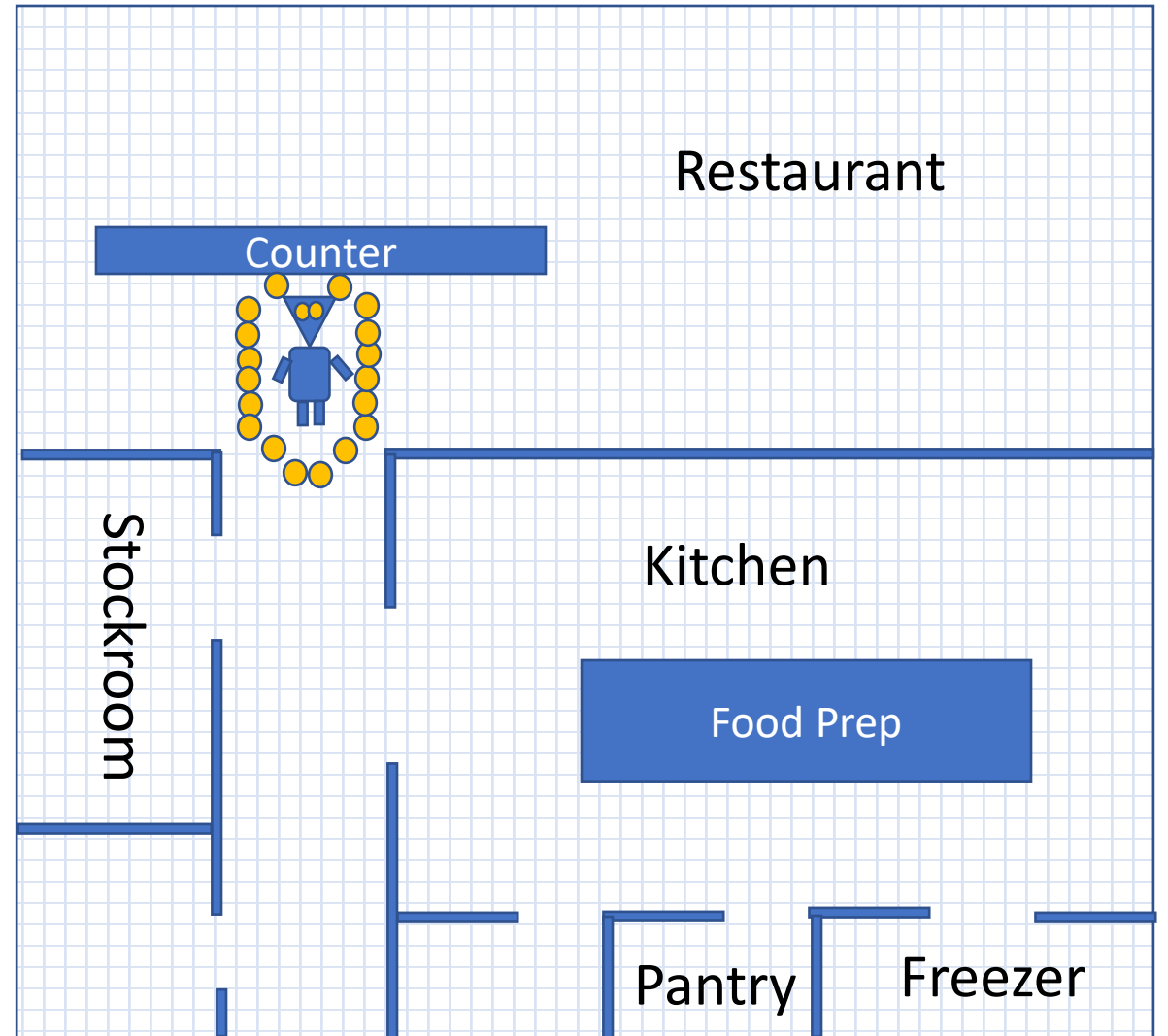
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



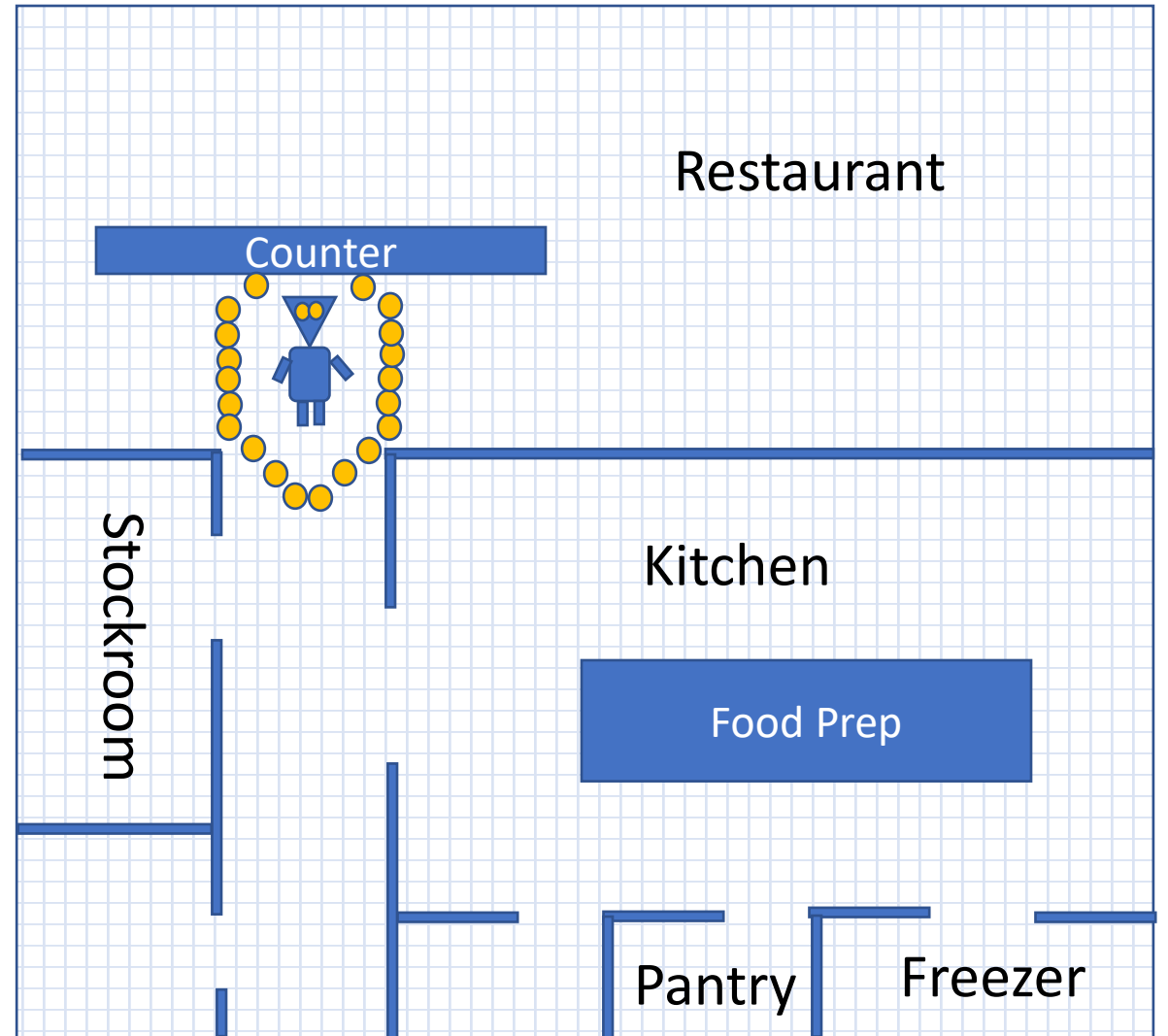
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



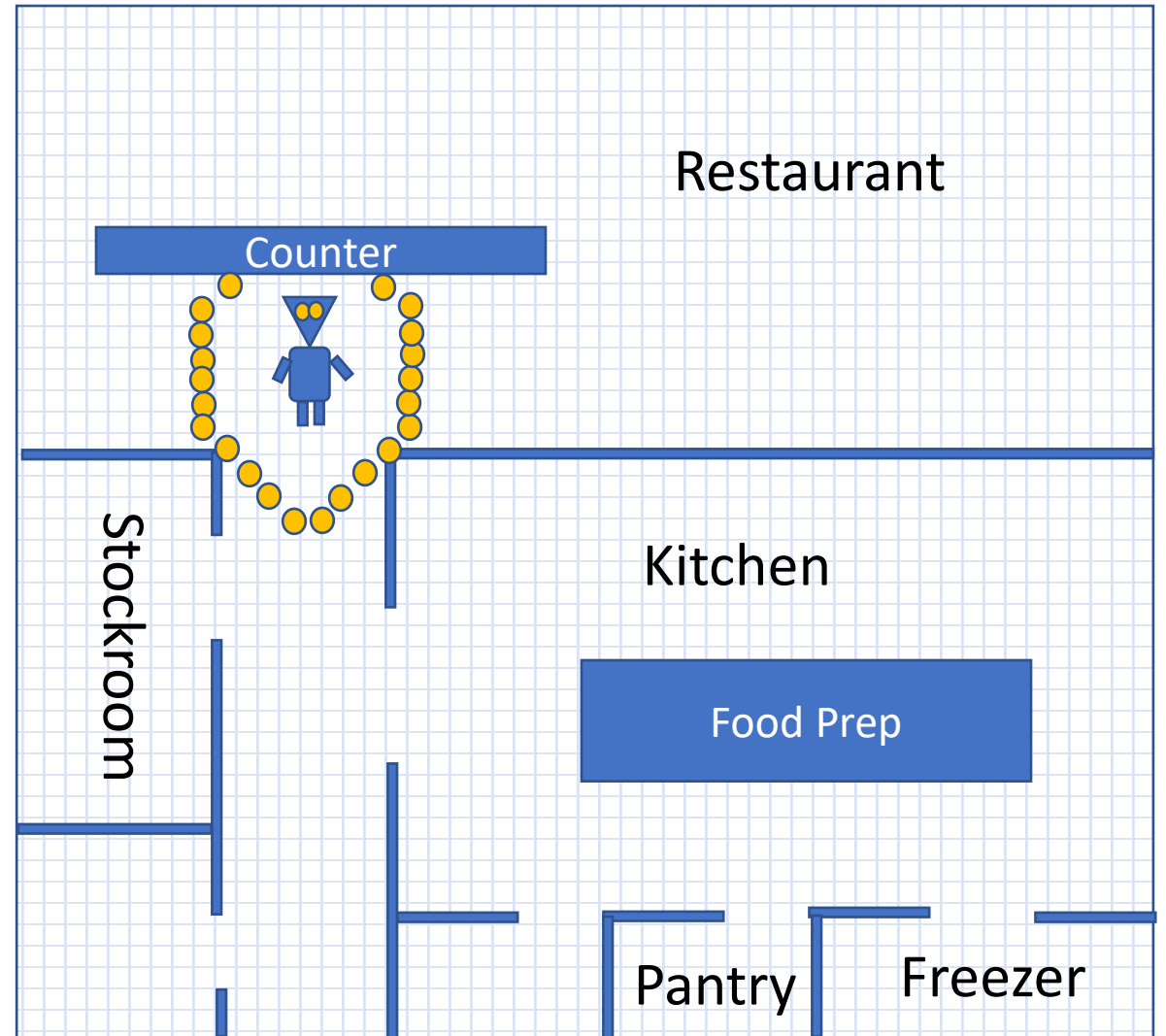
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



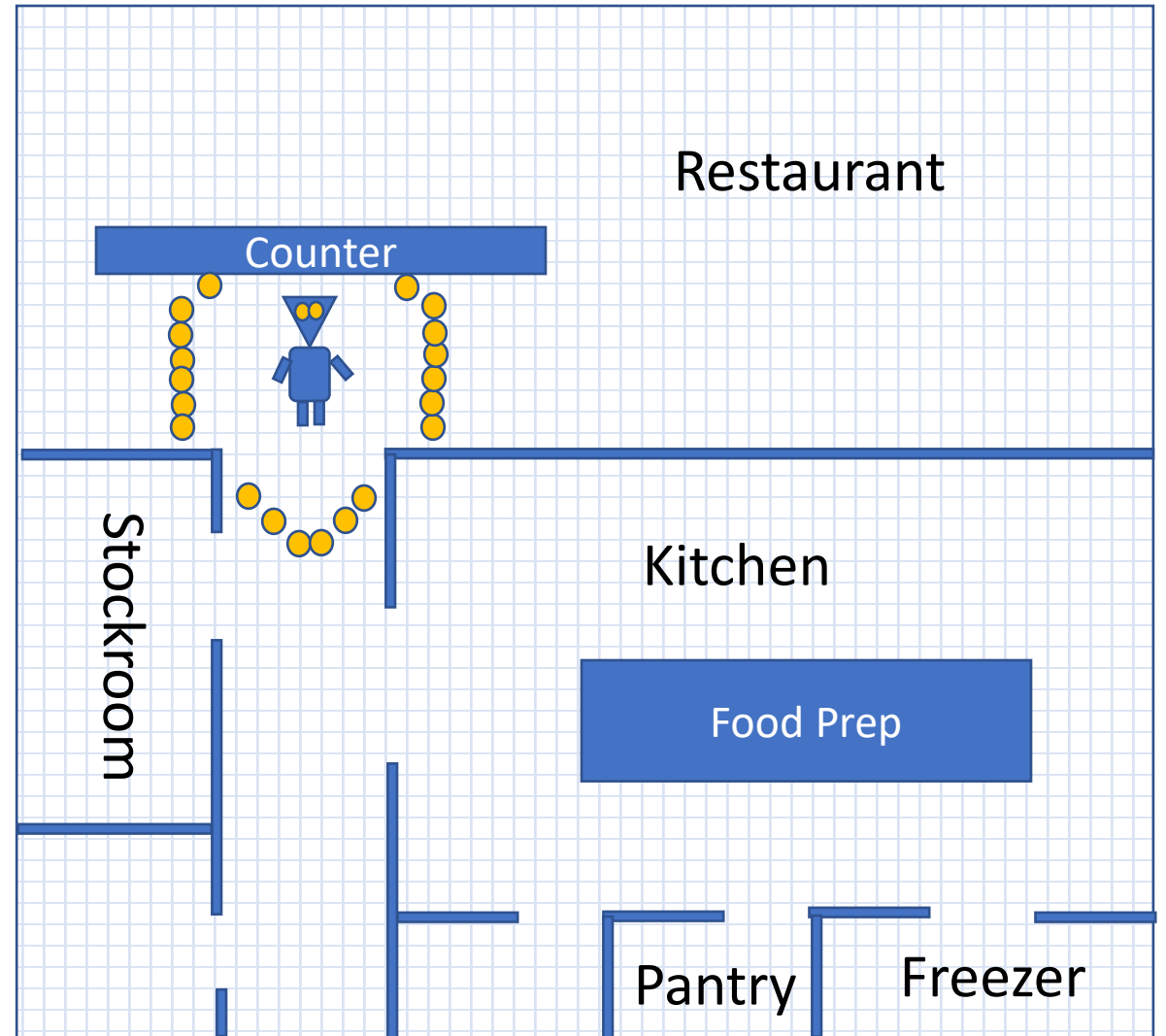
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



# Planning = search

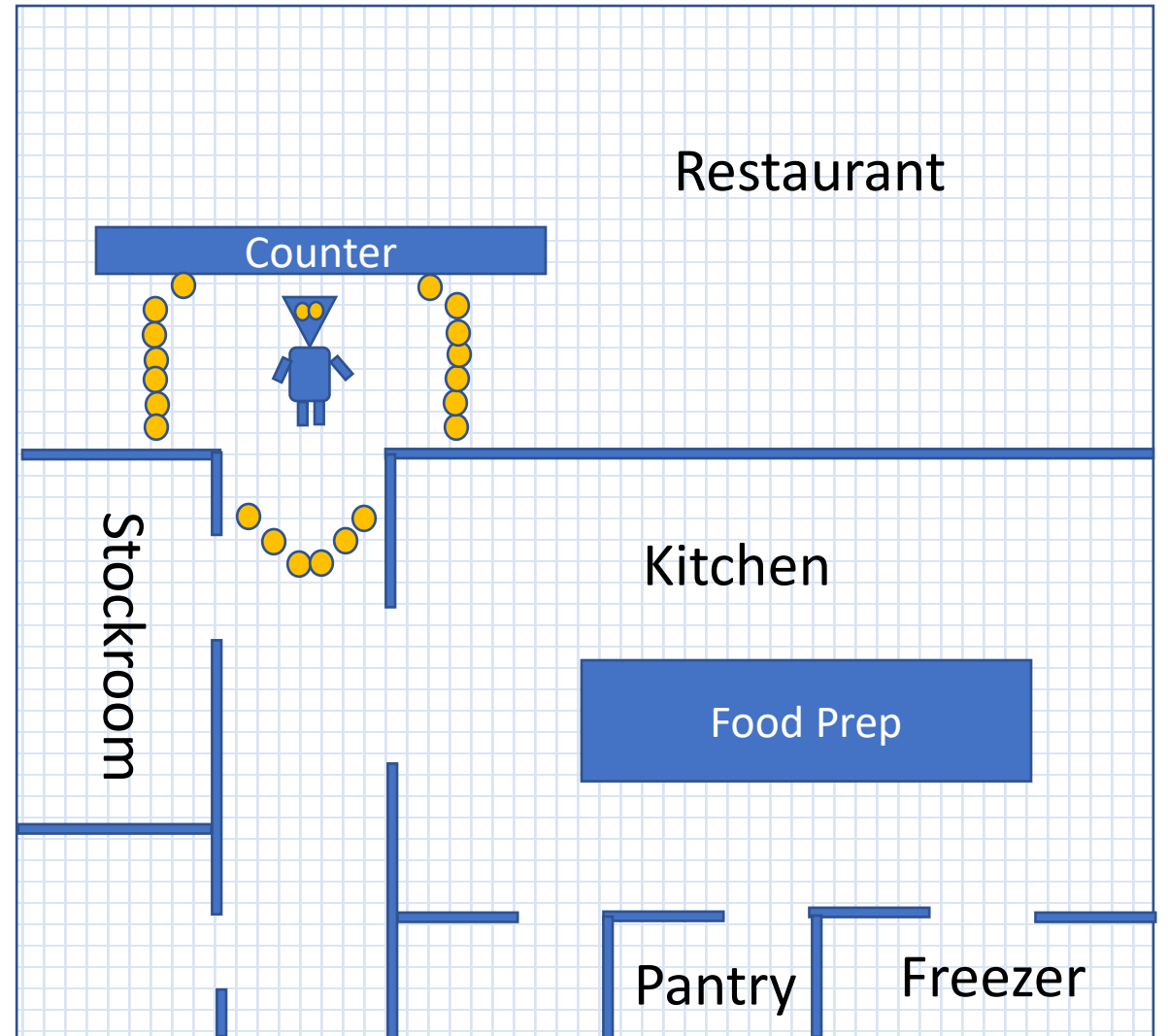
- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.





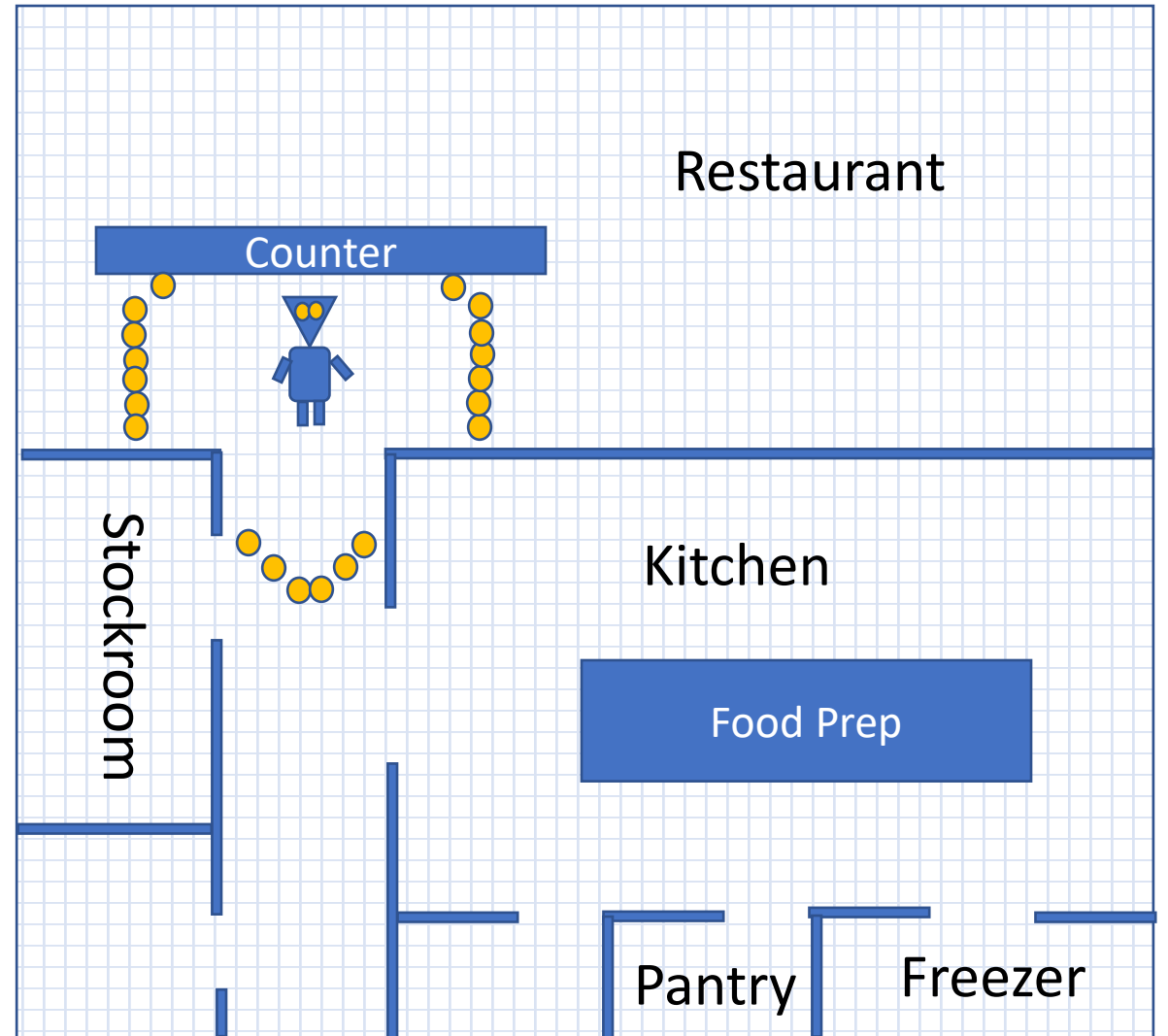
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



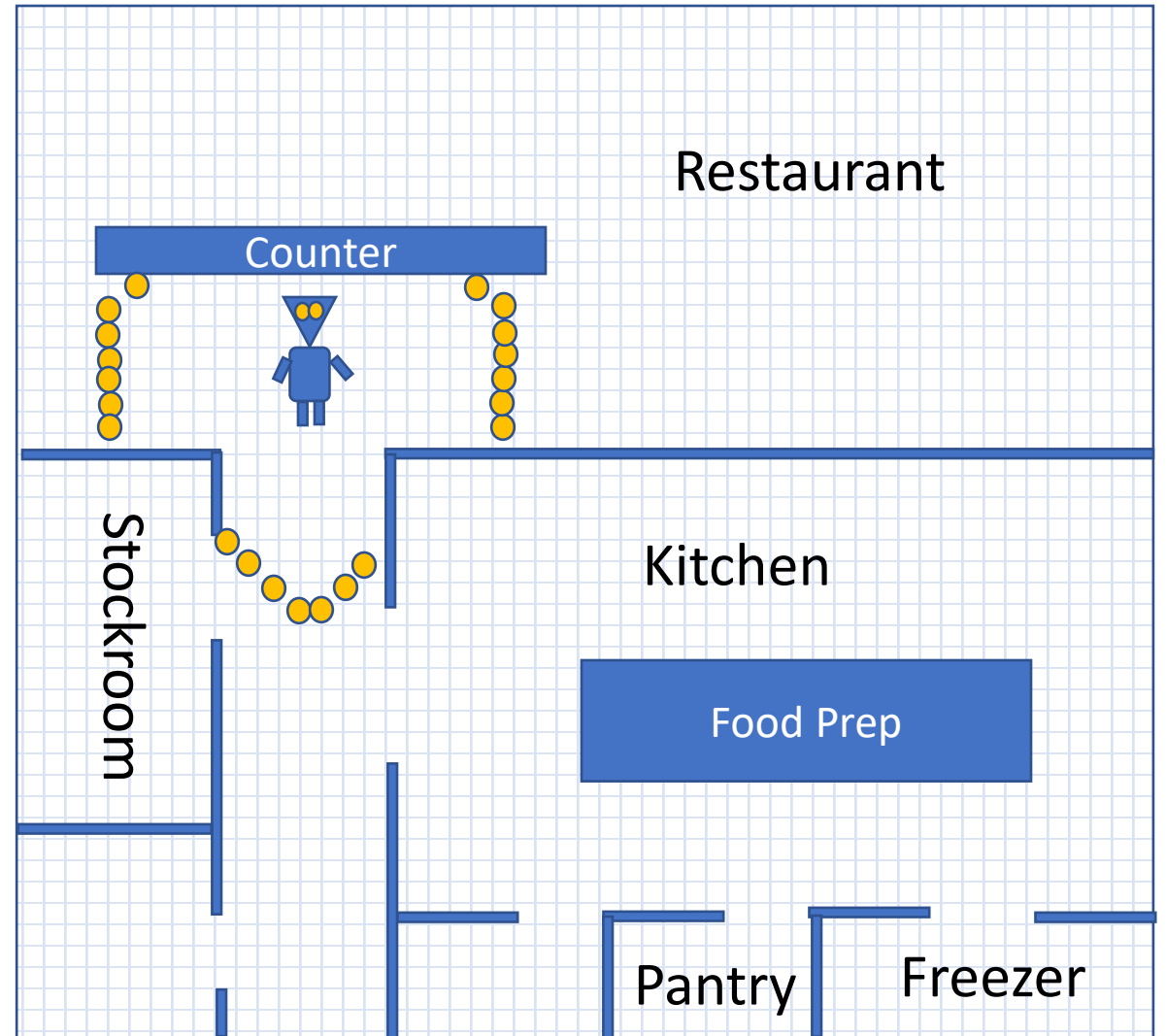
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



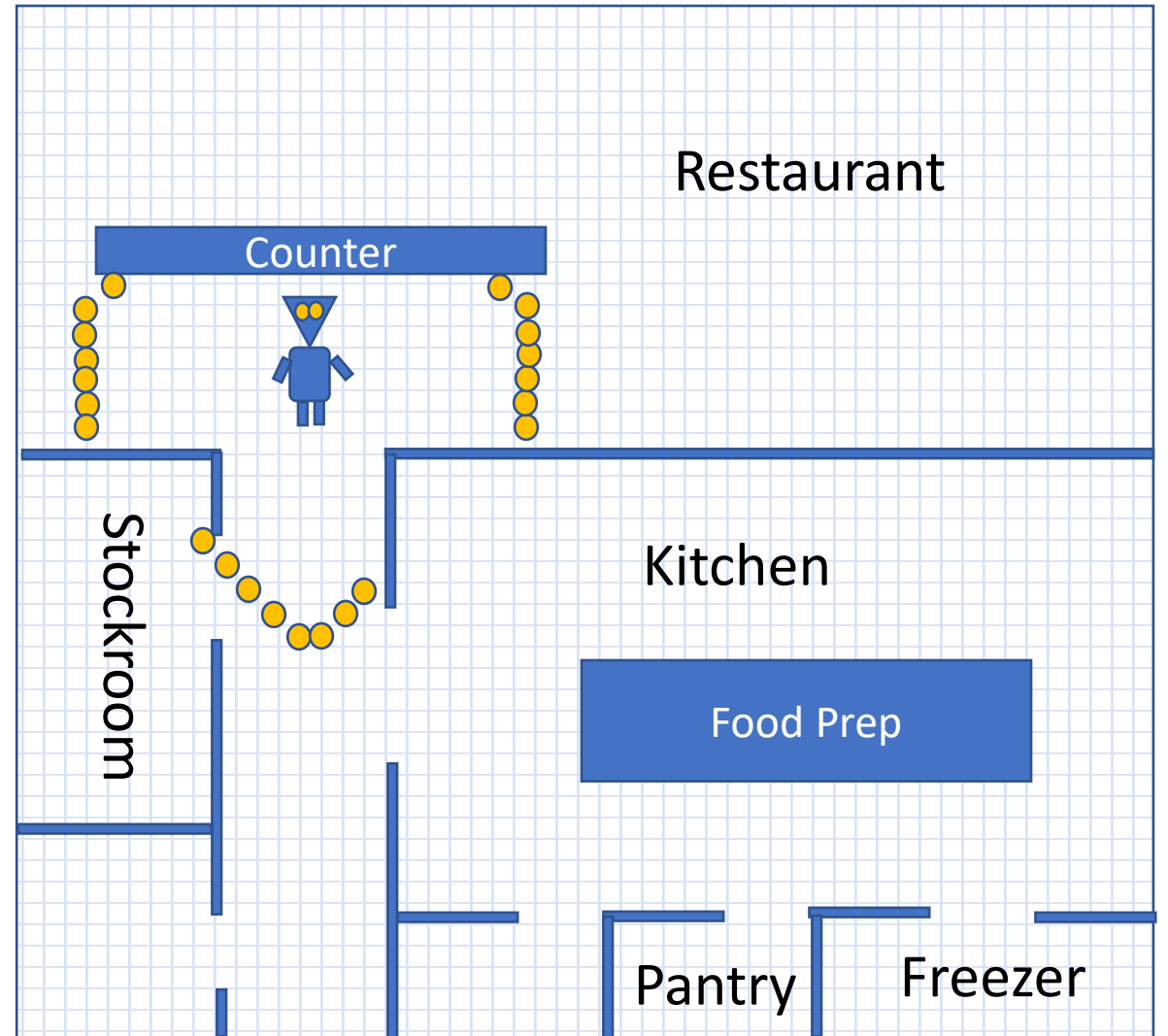
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



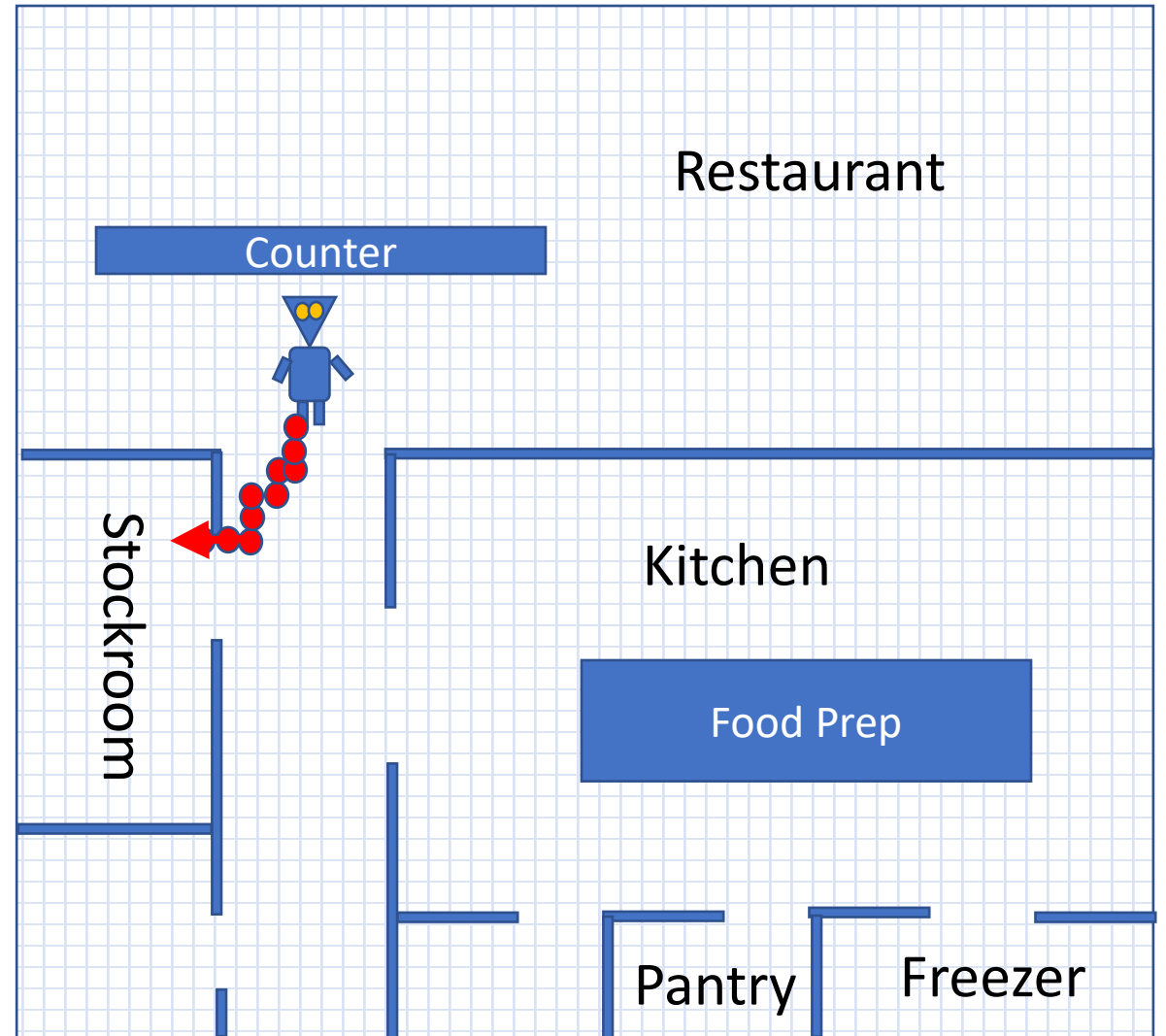
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



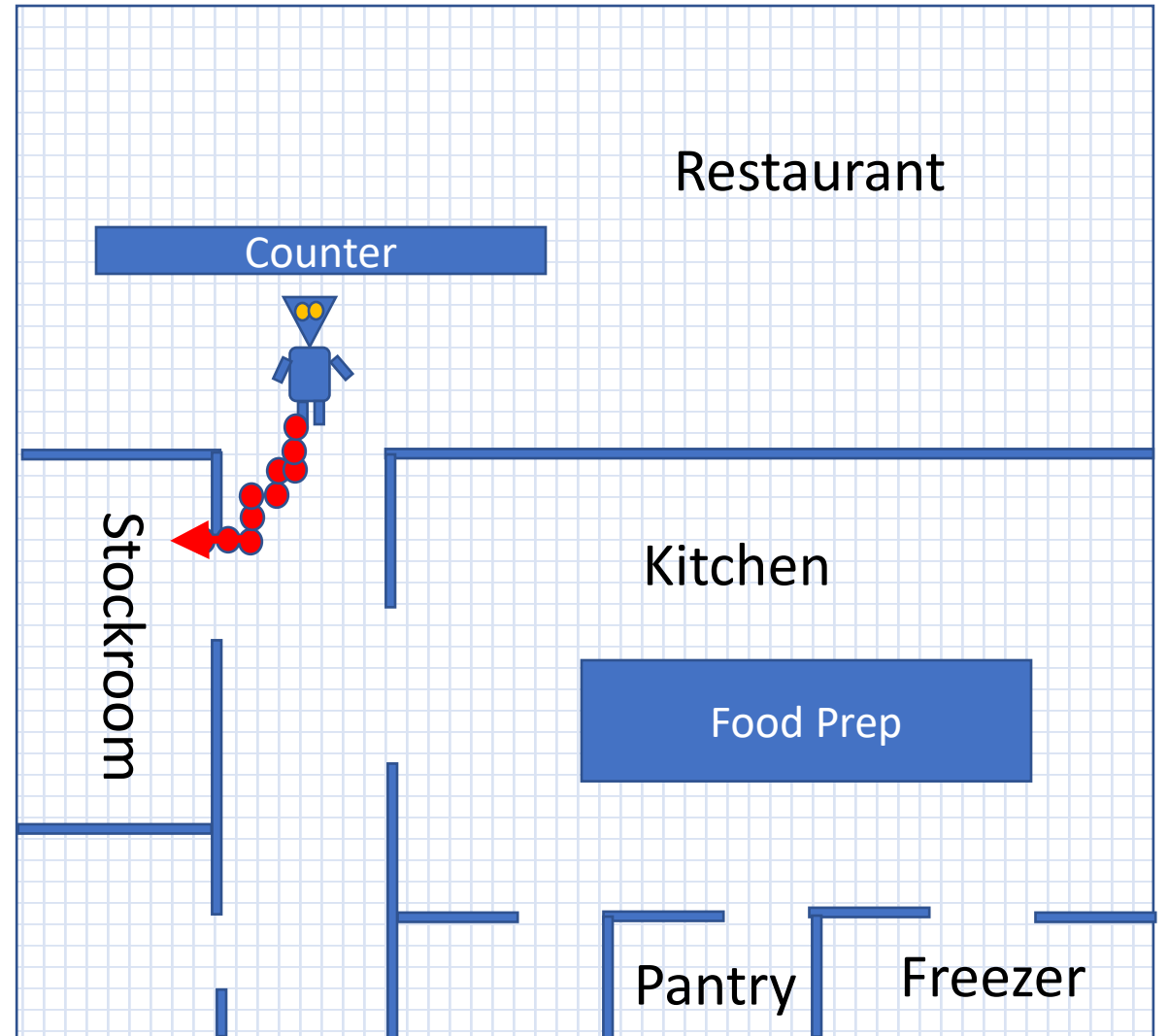
# Planning = search

- Step #1: Robot consults his internal map and uses BFS or A\* to find the best path.



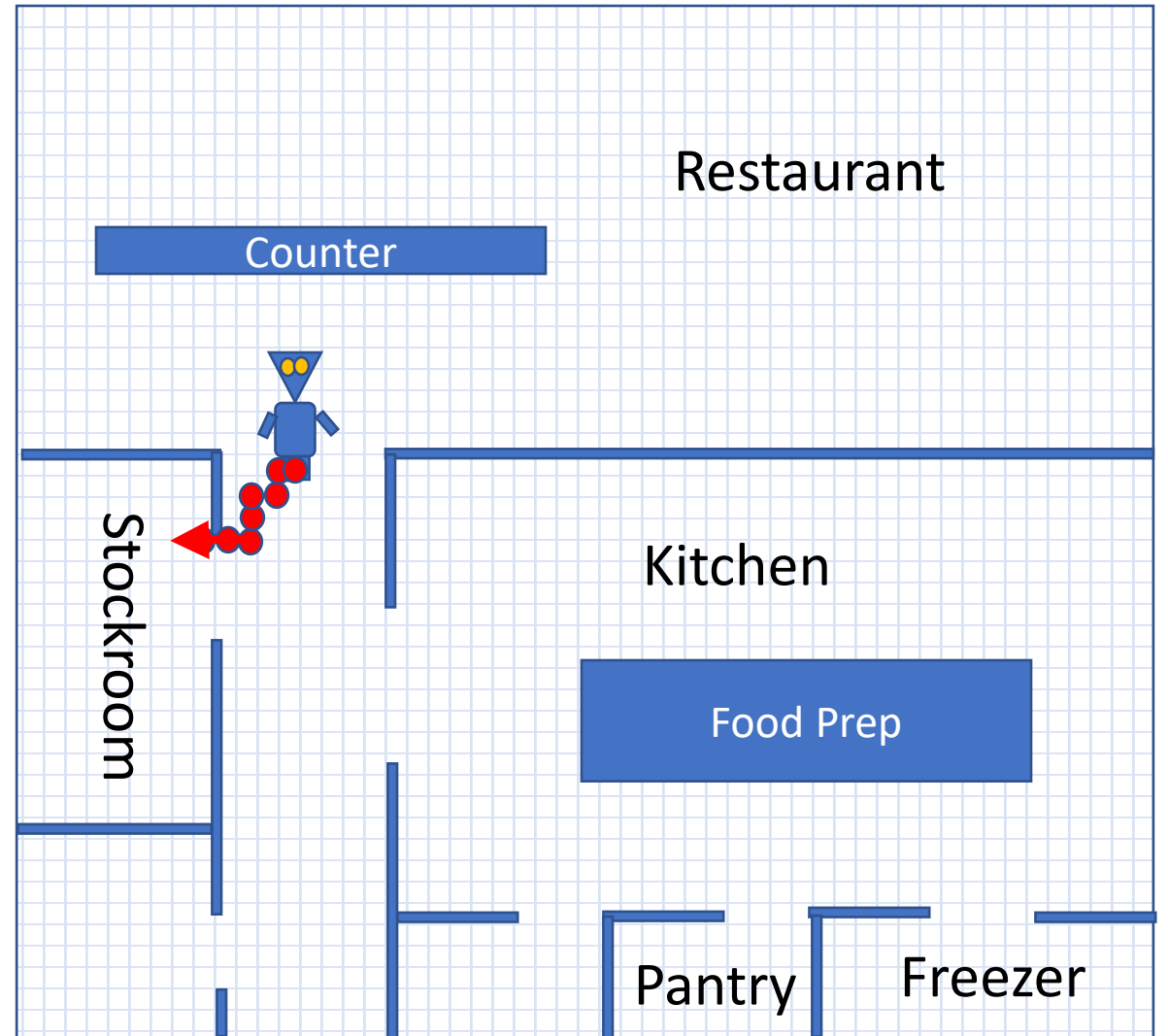
# Planning = search

- Step #2: Robot attempts to follow that path.



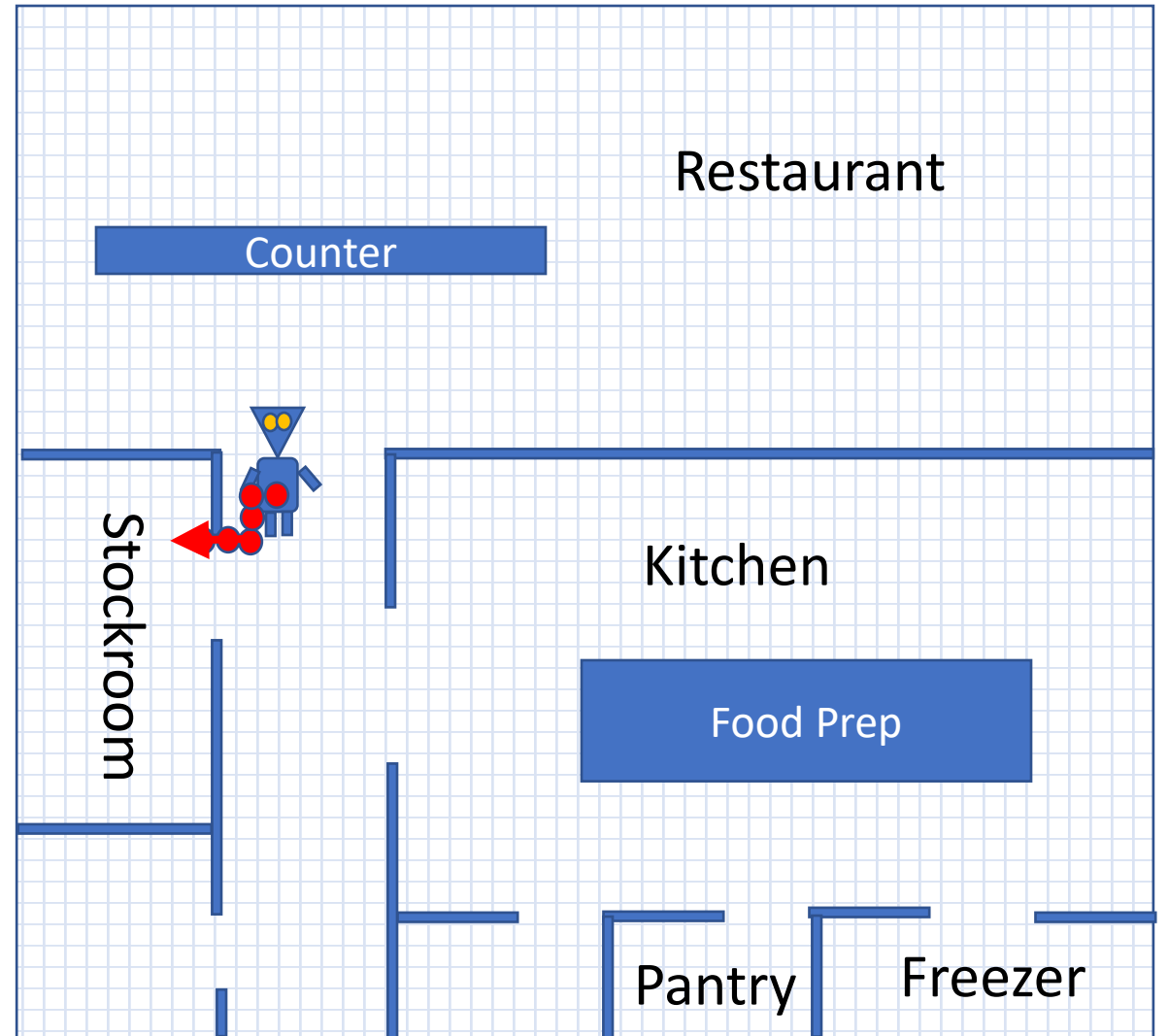
# Planning = search

- Step #2: Robot attempts to follow that path.



# Planning = search

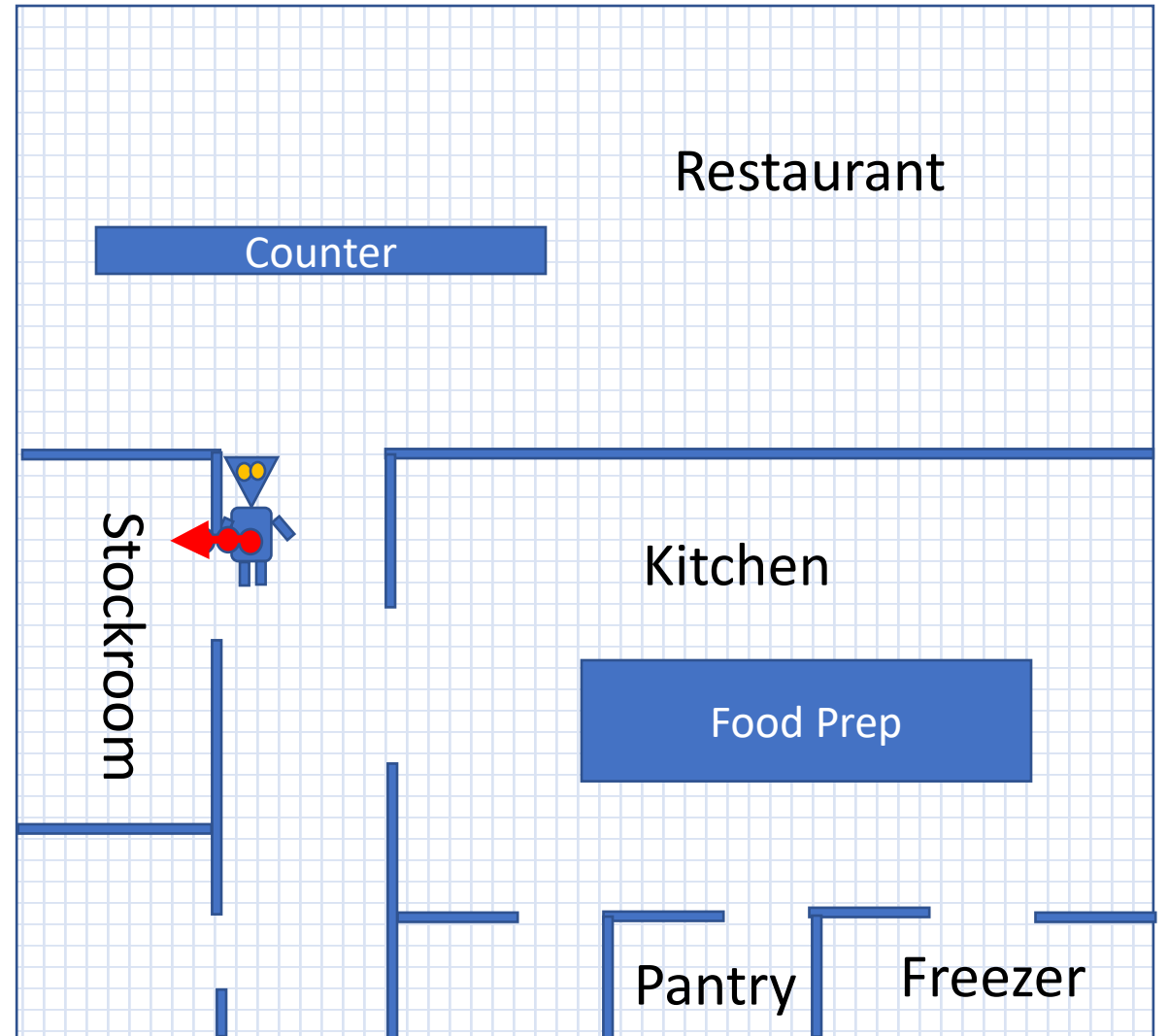
- Step #2: Robot attempts to follow that path.





# Planning = search

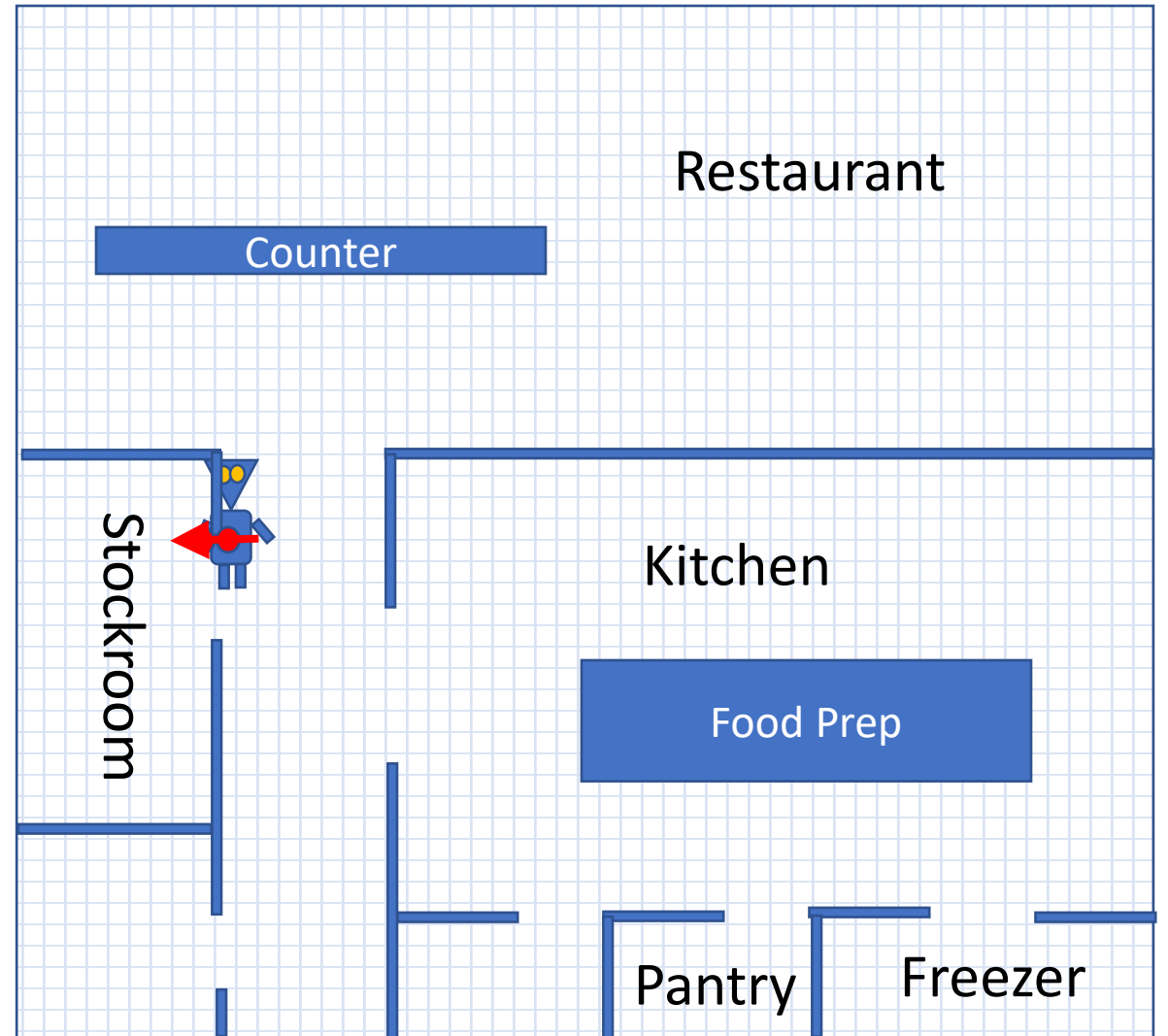
- Step #2: Robot attempts to follow that path.



# Planning = search

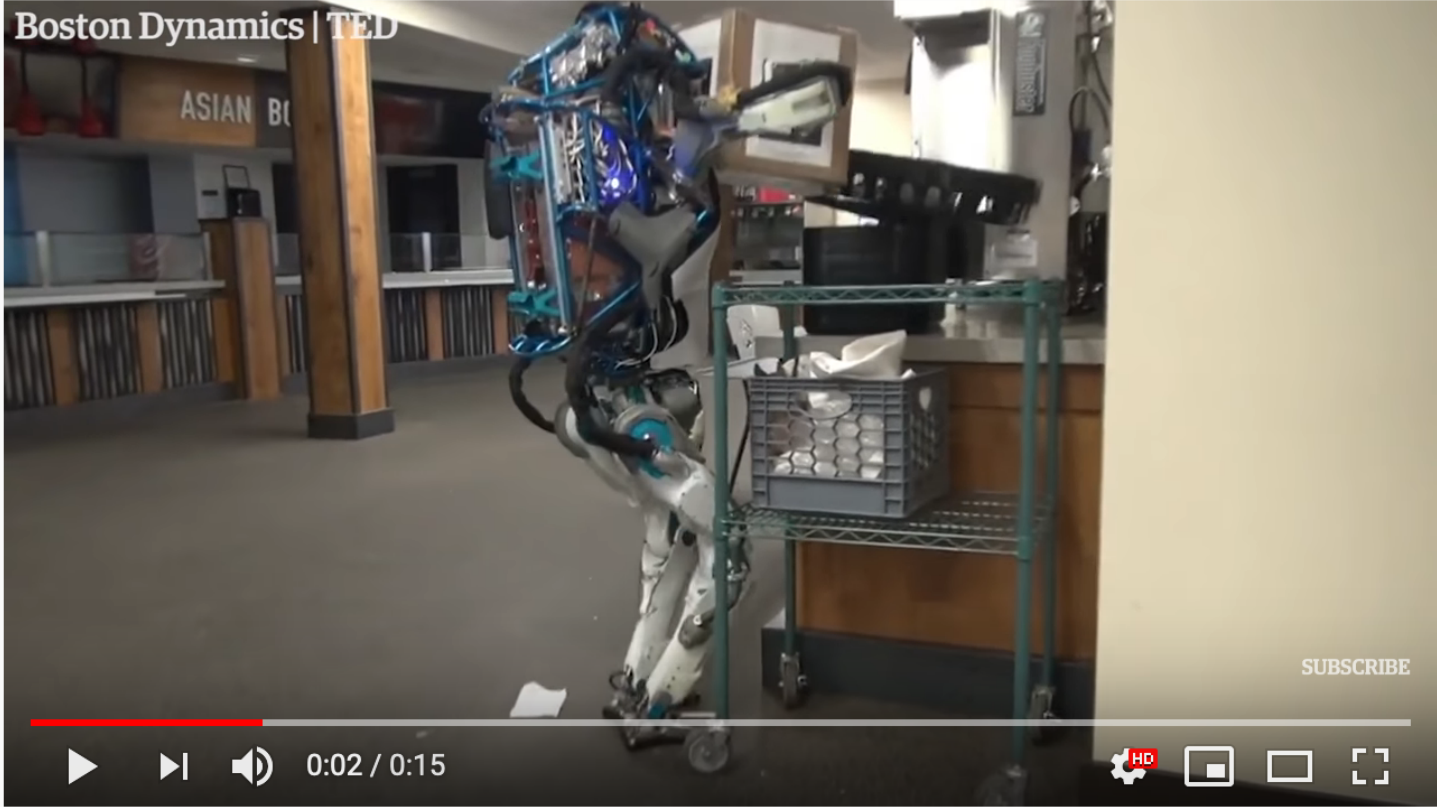
- Step #3: Robot bumps into the doorframe.

Poor robot.



# Poor robot

[https://www.youtube.com/watch?v=JzlsvFN\\_5HI](https://www.youtube.com/watch?v=JzlsvFN_5HI)



The video player shows a Boston Dynamics robot in a kitchen-like setting. The robot is attempting to stack a metal shelf on top of another one. It has successfully placed the shelf on the second shelf, but it is struggling to balance it. The robot is white with blue accents. The background shows a kitchen counter with a sink and a sign that says "ASIAN B". The video player interface includes a search bar with "robot fail", a Premium logo, a play button, a volume icon, a progress bar at 0:02 / 0:15, and a "SUBSCRIBE" button.

Boston Dynamics | TED

ASIAN B

SUBSCRIBE

0:02 / 0:15

HD

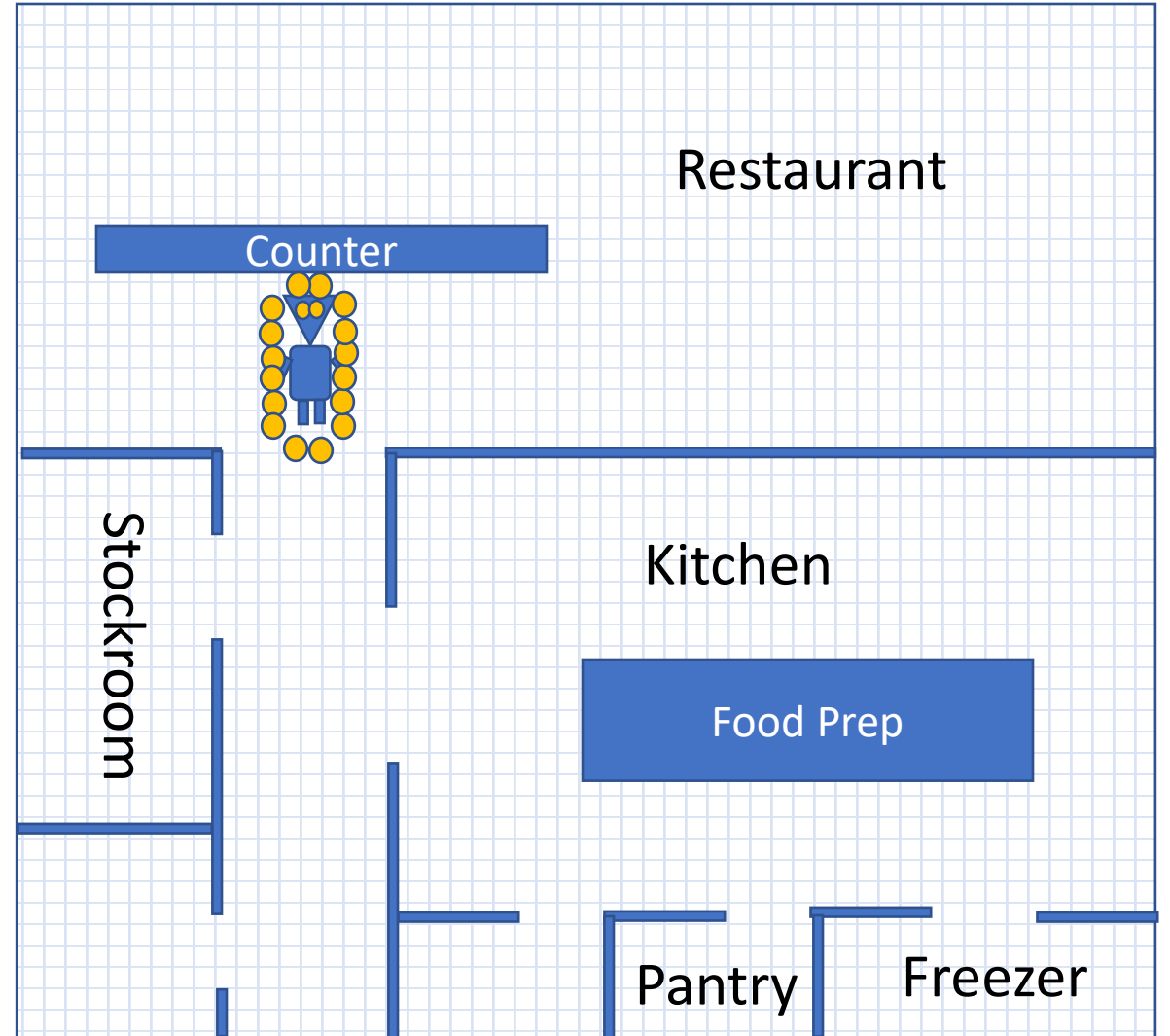
Hapless Boston Dynamics robot in shelf-stacking fail

267.980 views • Aug 15, 2017

663 22 SHARE SAVE

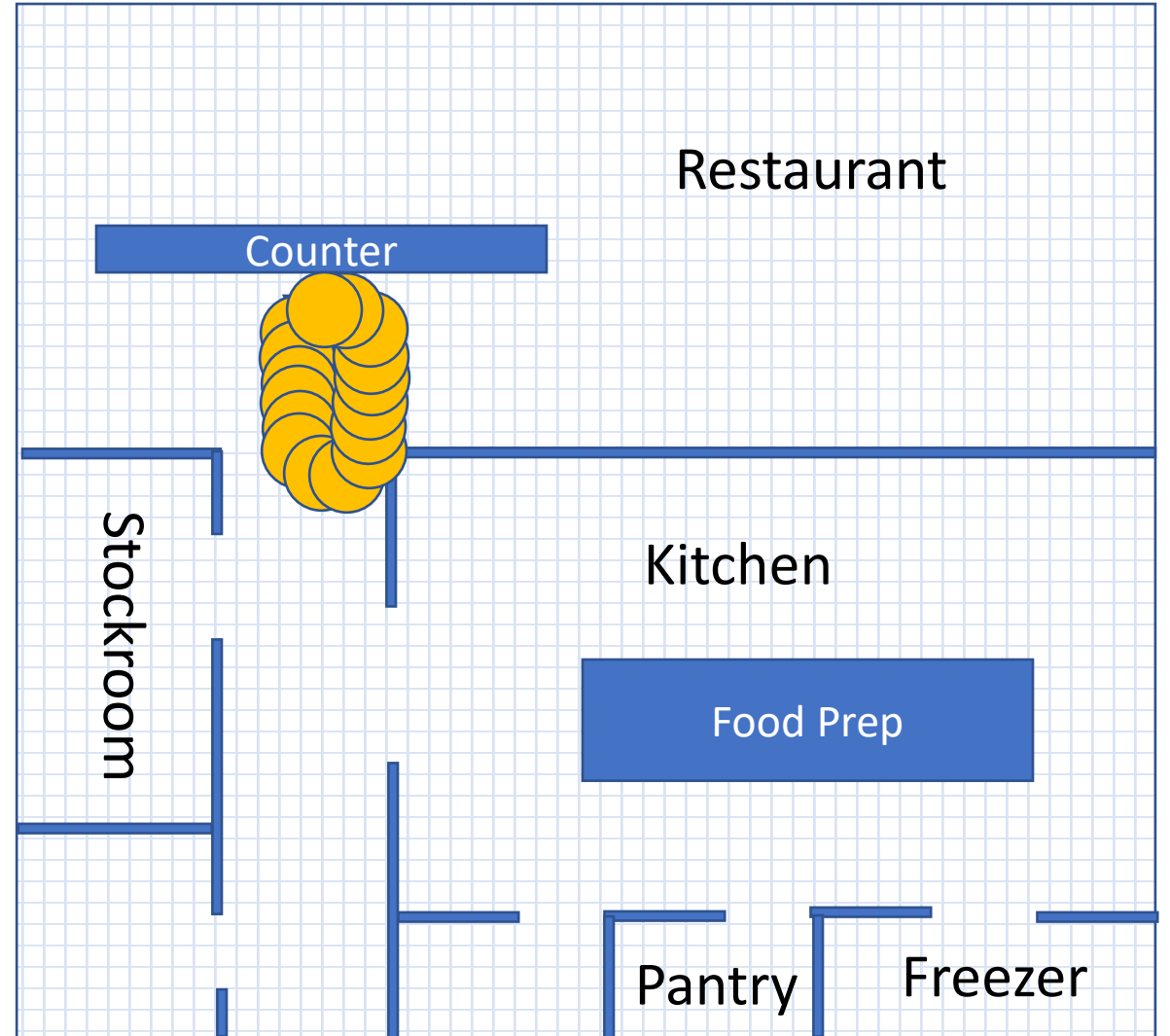
# How can we help the robot?

- Let's give it more information.
- Let's tell it how wide it is.



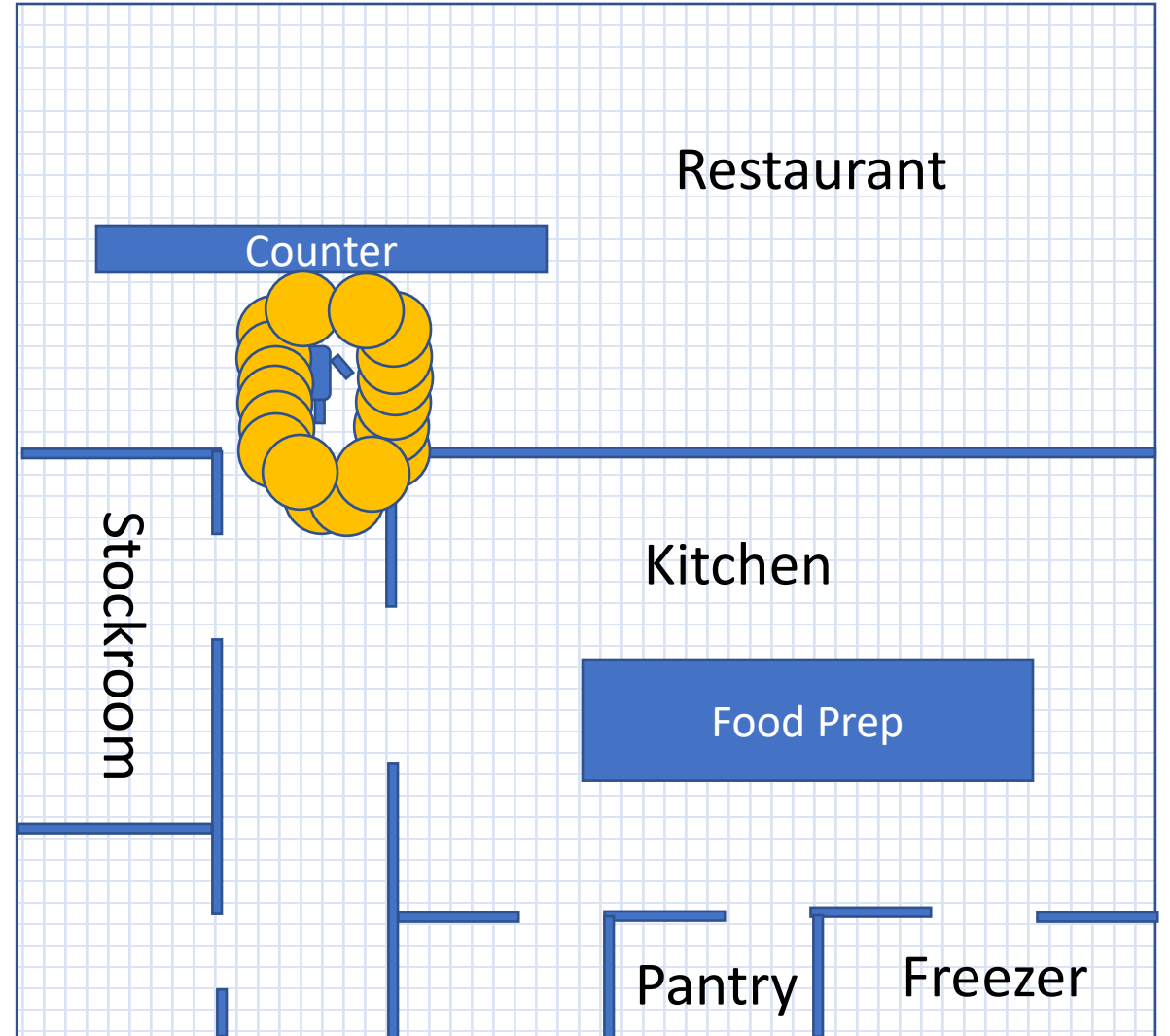
# How can we help the robot?

- Option #1: every node in the search tree carries information about the size of the robot.



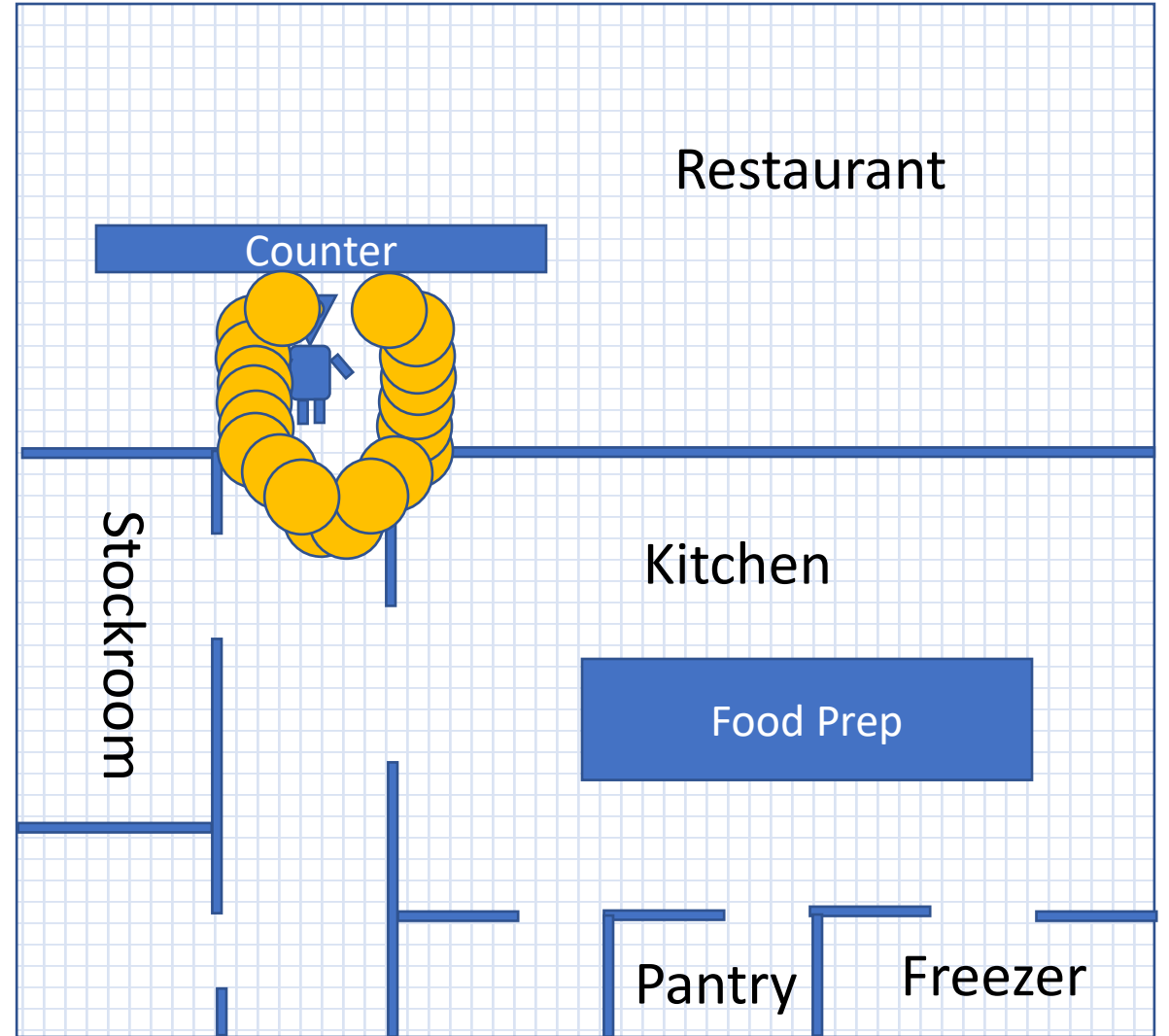
# How can we help the robot?

- Option #1: every node in the search tree carries information about the size of the robot.



# How can we help the robot?

- Option #1: every node in the search tree carries information about the size of the robot.

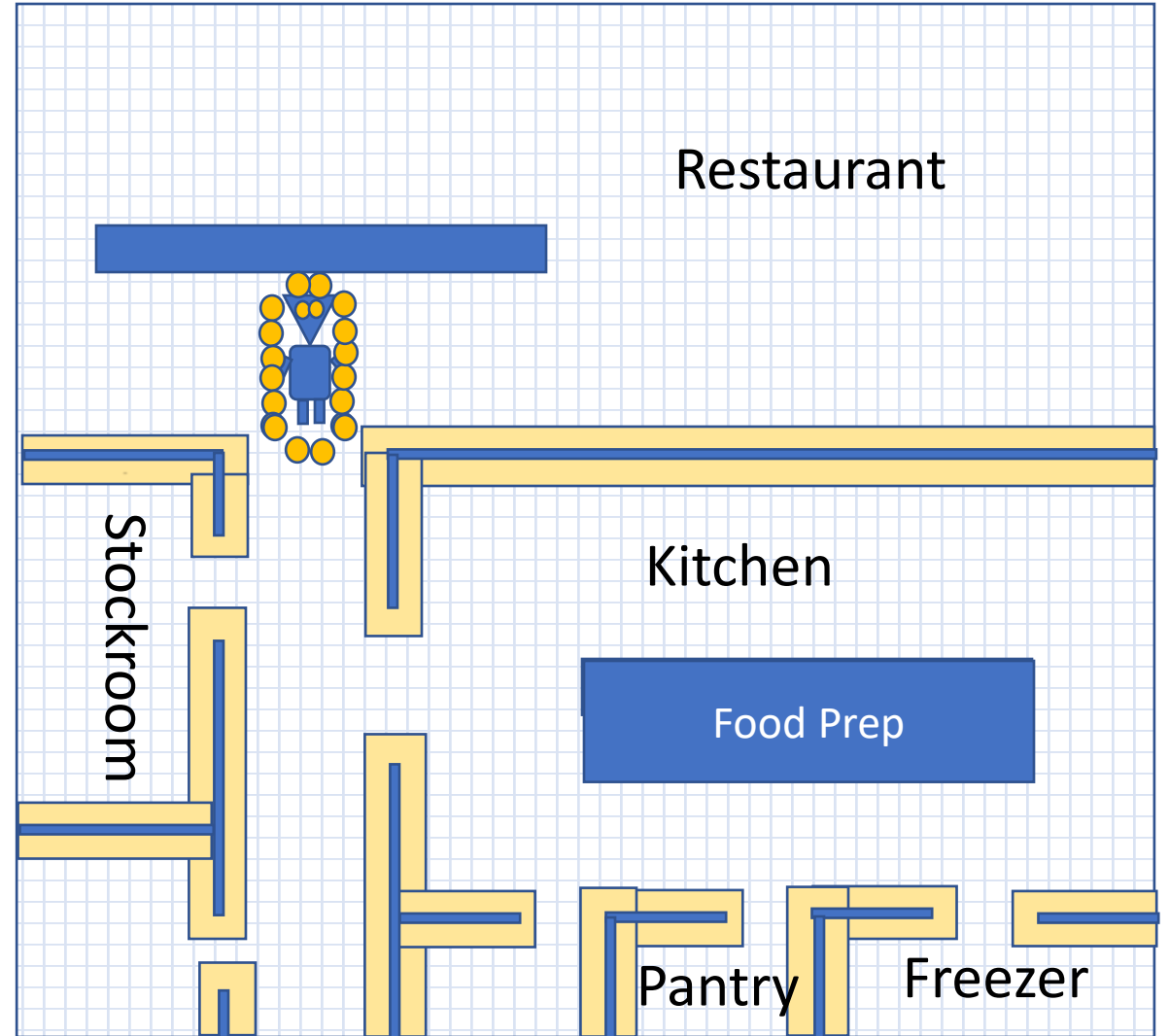


OK, that's a little unwieldy...



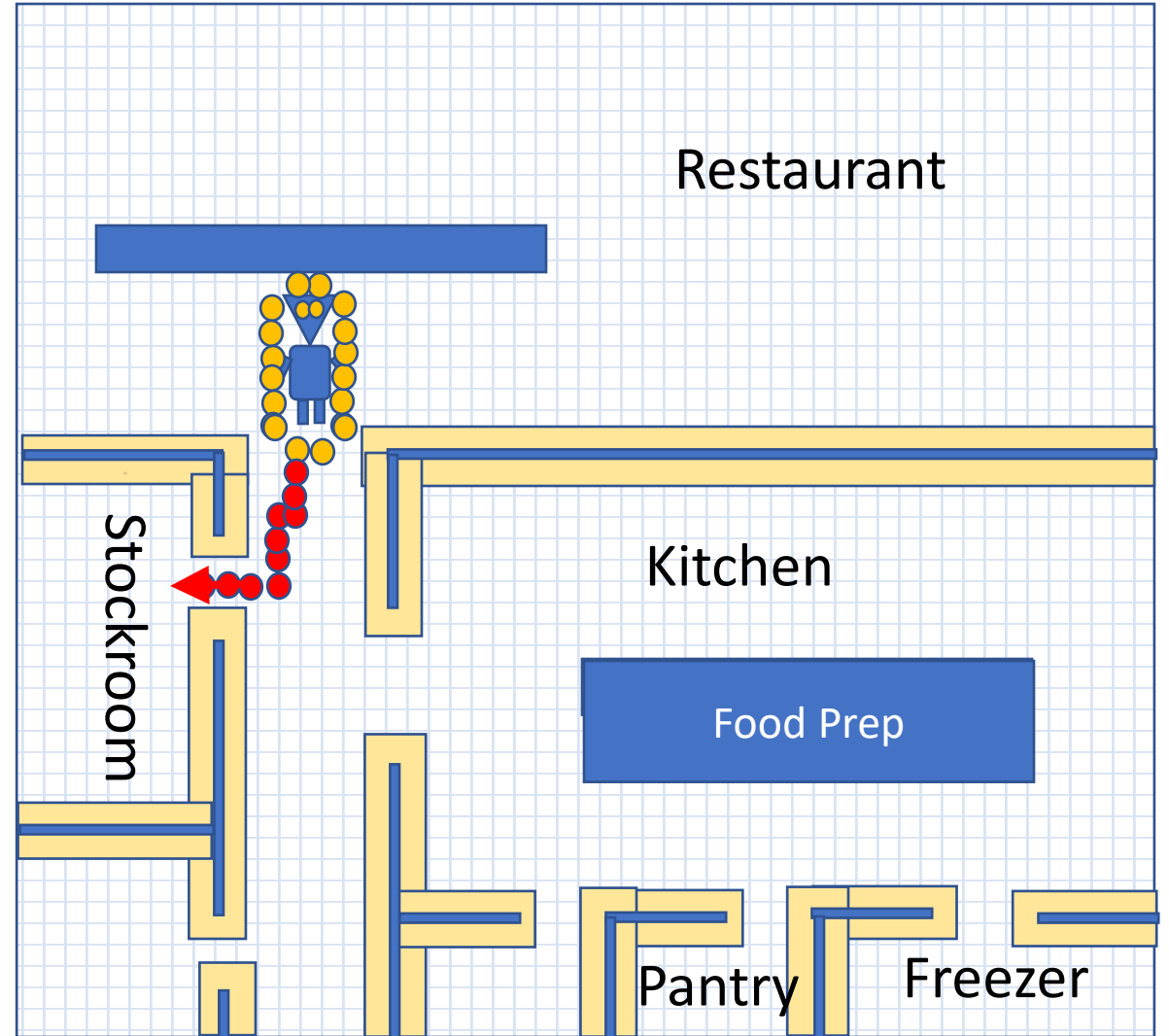
# How can we help the robot?

- Option #2: the map tells the robot how wide it is.



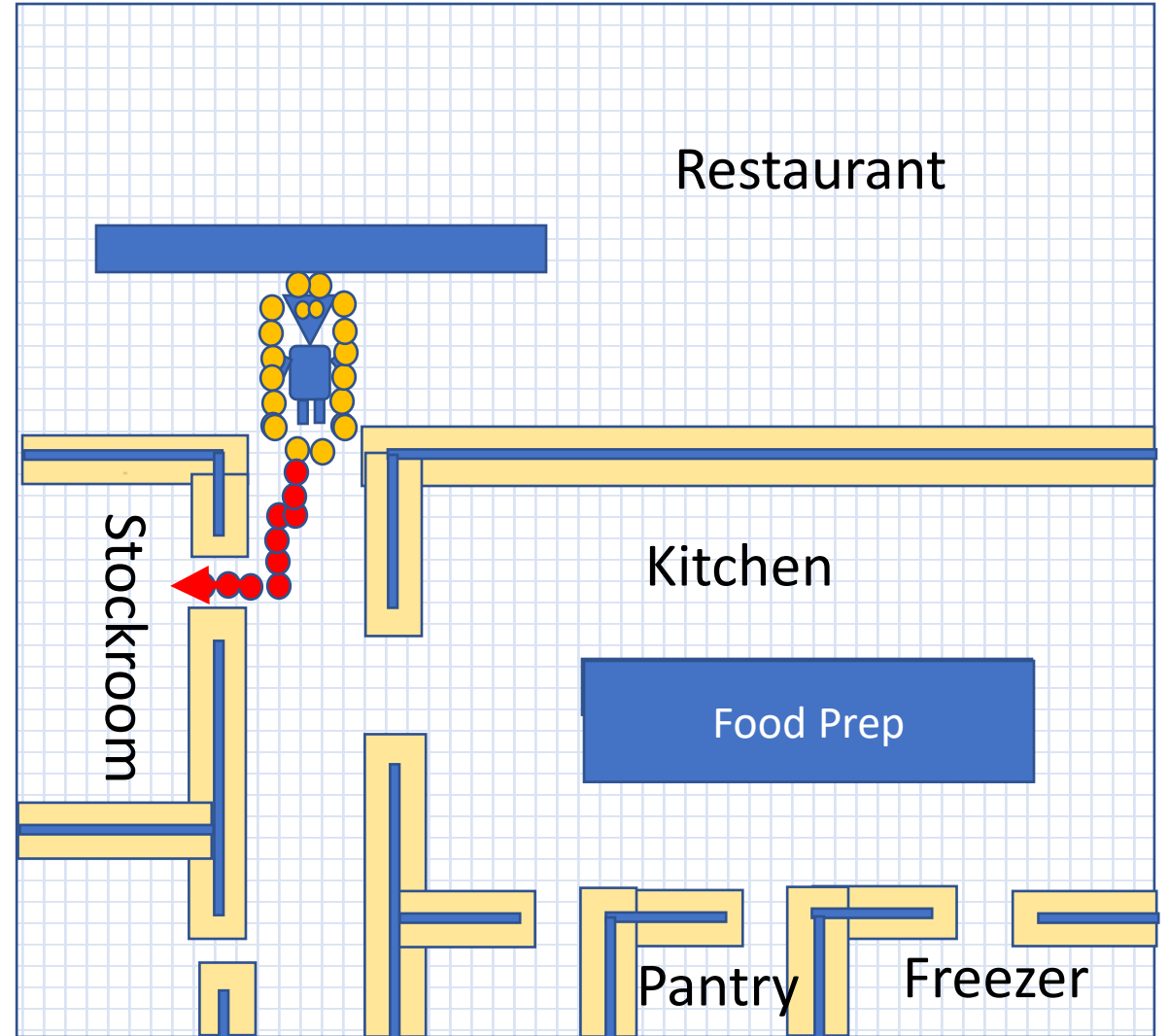
# How can we help the robot?

- Option #2: the map tells the robot how wide it is.
- Now, any optimal path that the robot finds is a path that it can actually use.



# Configuration Space

- This new search space is called a **configuration space**.
- It specifies which configurations are possible.



# Configuration Space

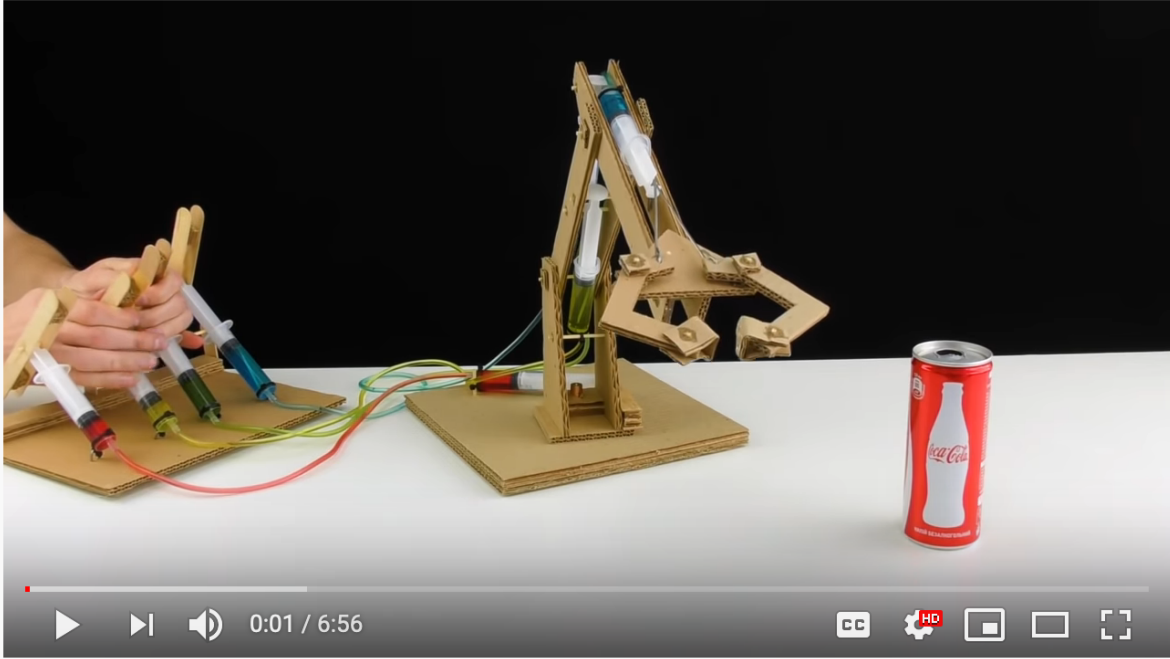
In classical mechanics,

- the parameters that define the configuration of a system are called **generalized coordinates**, and
- the vector space defined by these coordinates is called the **configuration space**.

[https://en.wikipedia.org/wiki/Configuration\\_space\\_\(physics\)](https://en.wikipedia.org/wiki/Configuration_space_(physics))

# Configuration Space Example: Robot Arm

<https://www.youtube.com/watch?v=P2r9U4wkjcc>

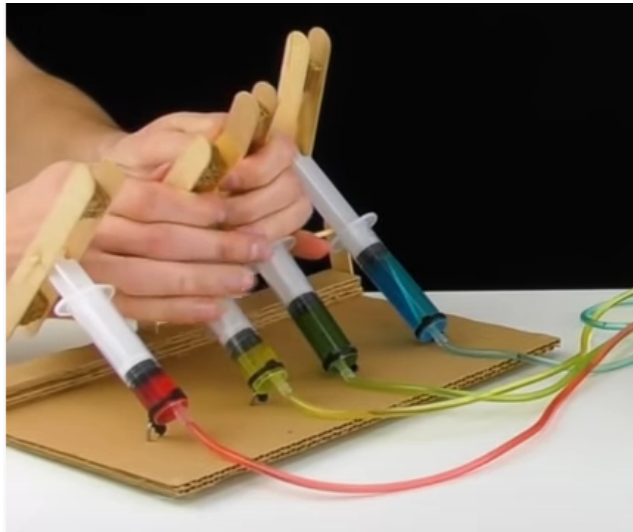


The image shows a YouTube video player interface. At the top, there is a navigation bar with a menu icon, a 'Premium' badge, a search bar containing the text 'robot arm', and icons for video recording, a grid, a notification bell, and a profile picture. The main video frame displays a hydraulic-powered robotic arm constructed from cardboard. A person's hands are visible on the left, adjusting the arm's components. The arm is positioned over a red Coca-Cola can on a white surface. The video player controls at the bottom show a play button, a progress bar at 0:01 / 6:56, and icons for closed captions, HD quality, and full screen.

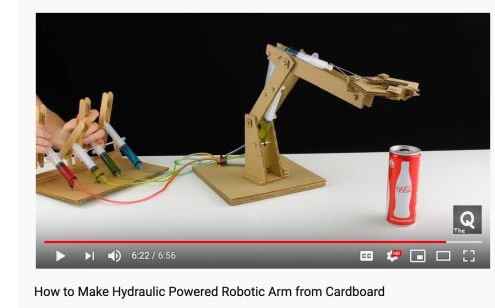
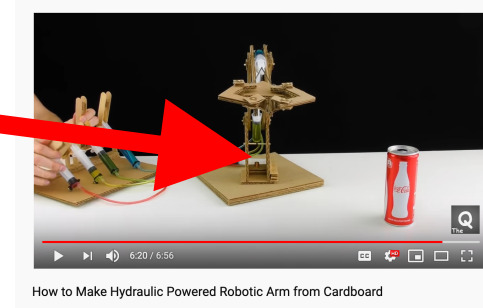
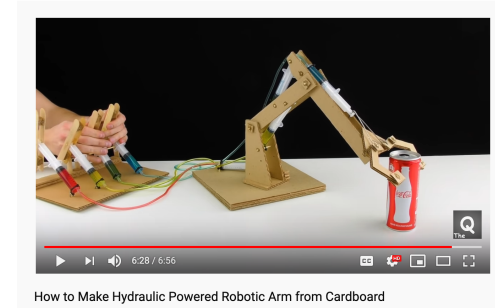
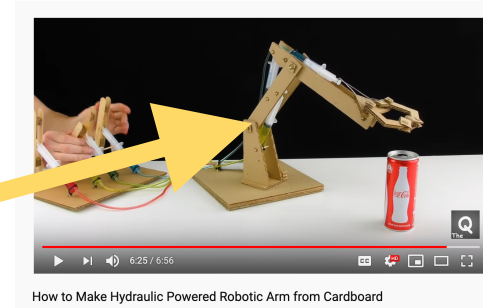
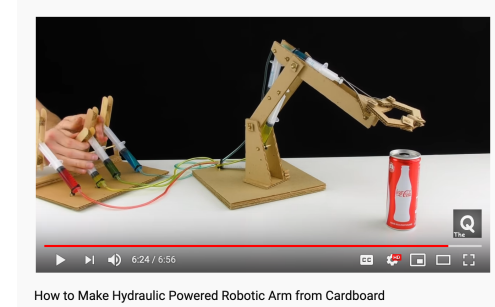
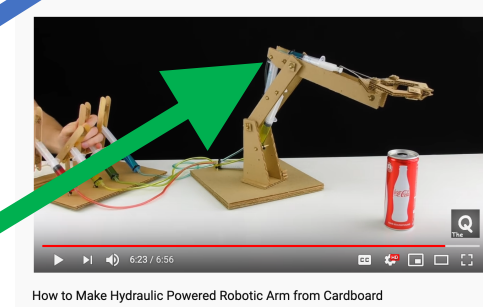
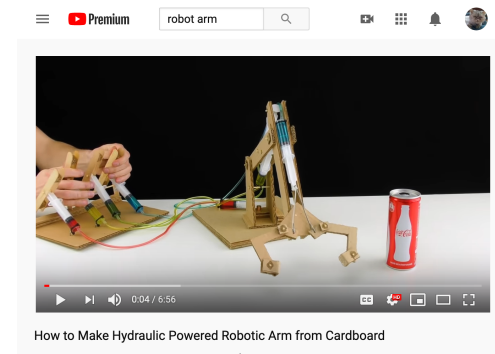
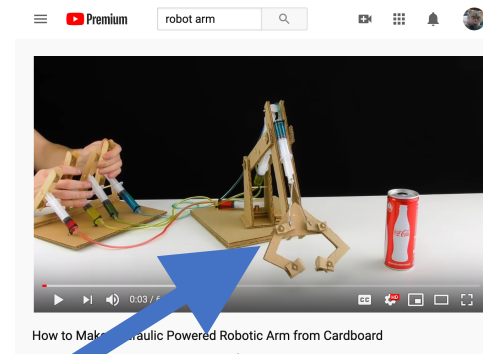
How to Make Hydraulic Powered Robotic Arm from Cardboard

# Configuration Space Example

Configuration space:  
4 coordinates

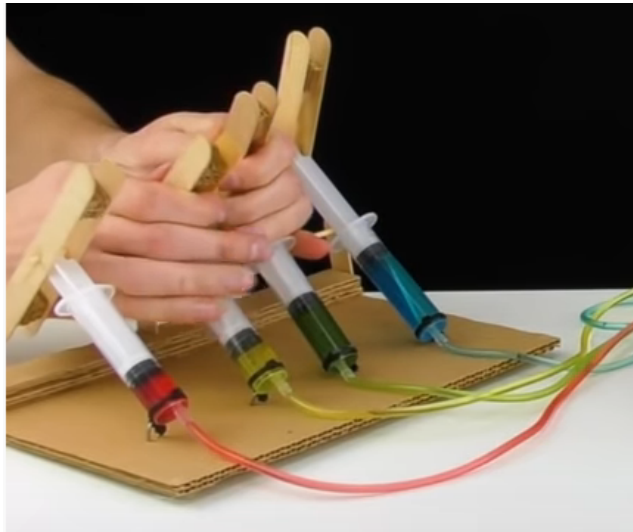


1. Grip
2. Elbow
3. Shoulder
4. Rotation

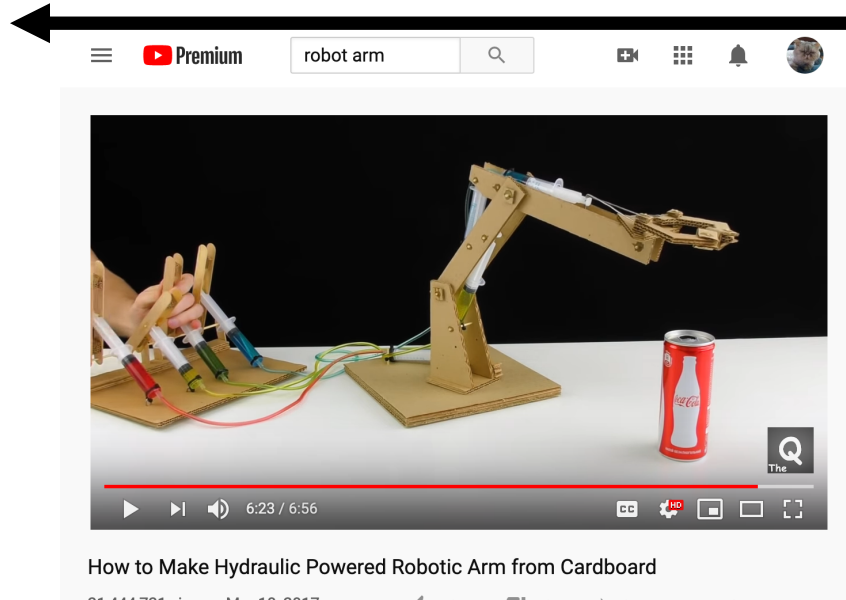


The MP2  
Configuration Space:

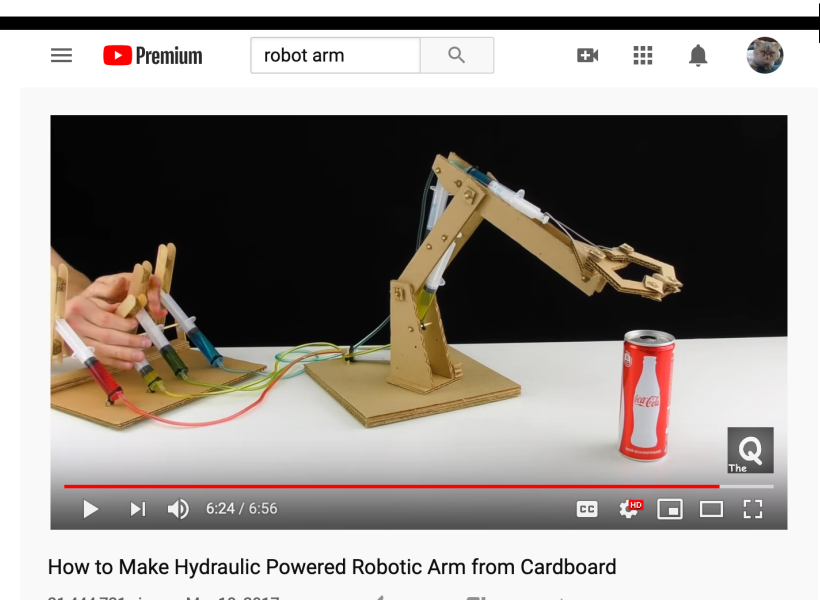
Just 2 coordinates



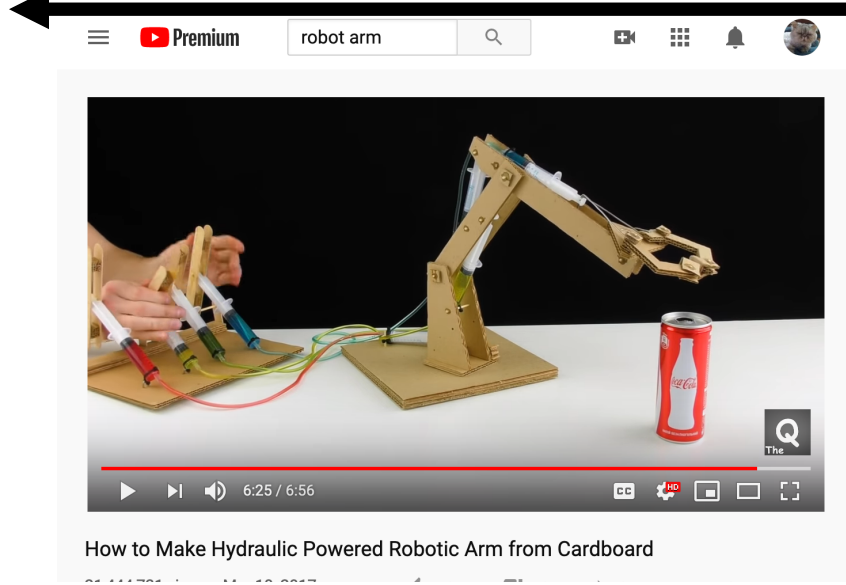
Elbow  $\approx 120^\circ$



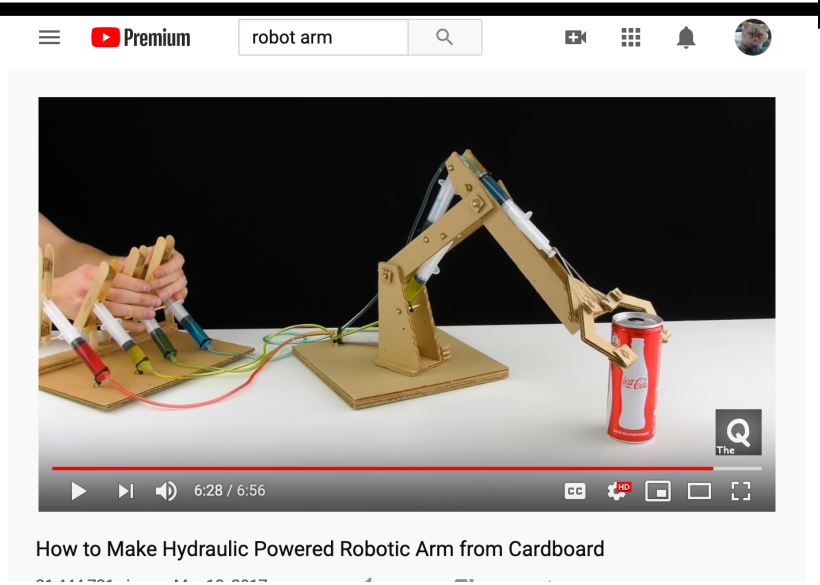
Elbow  $\approx 90^\circ$



Shoulder  $\approx 60^\circ$



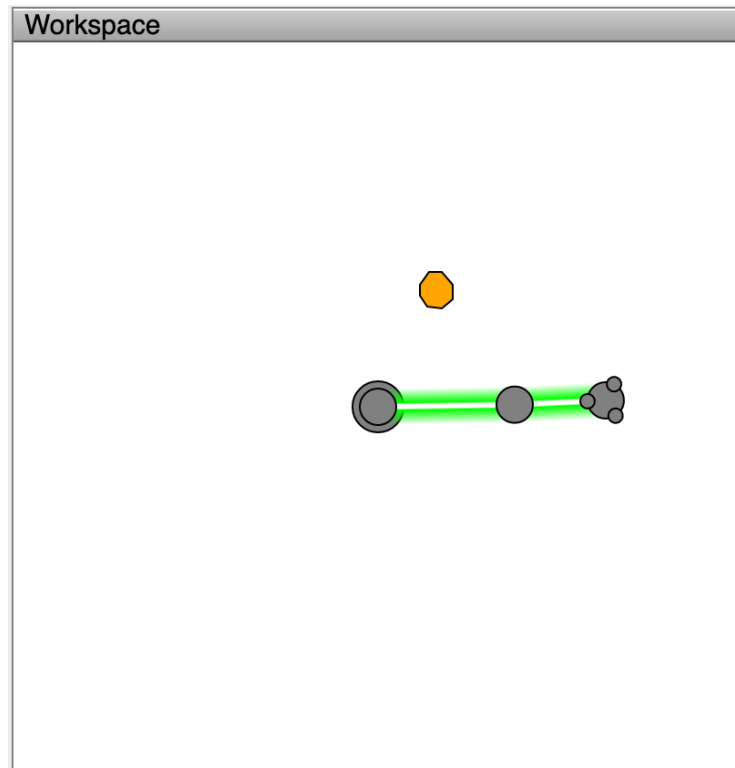
Shoulder  $\approx 45^\circ$



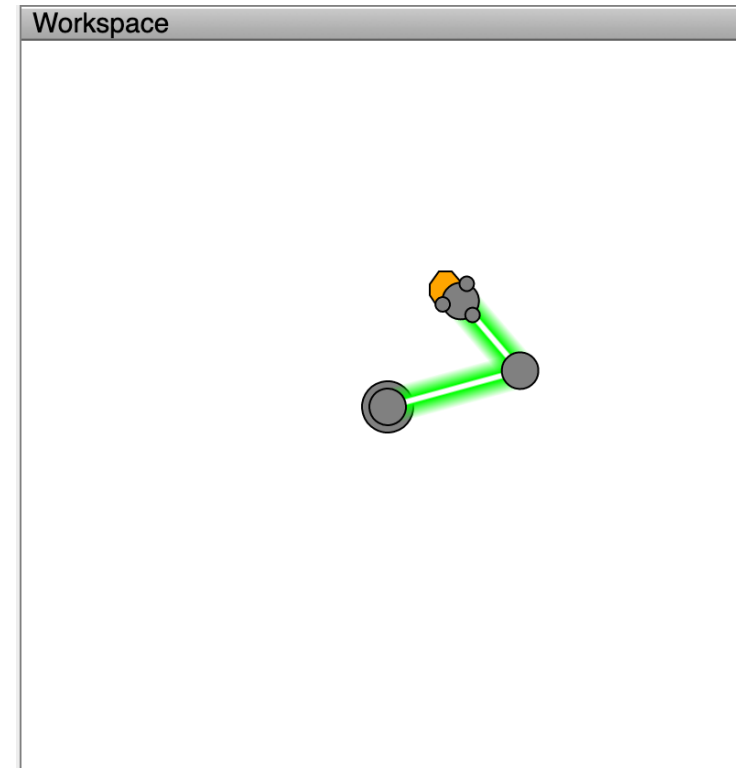
# The Robot Arm Reaching Problem

Jeff Ichnowski, University of North Carolina, <https://www.cs.unc.edu/~jeffi/c-space/robot.shtml>

- Given a robot arm in START,
- how should I adjust ELBOW and SHOULDER to most quickly reach GOAL?



START



GOAL



# The Robot Arm Reaching Problem

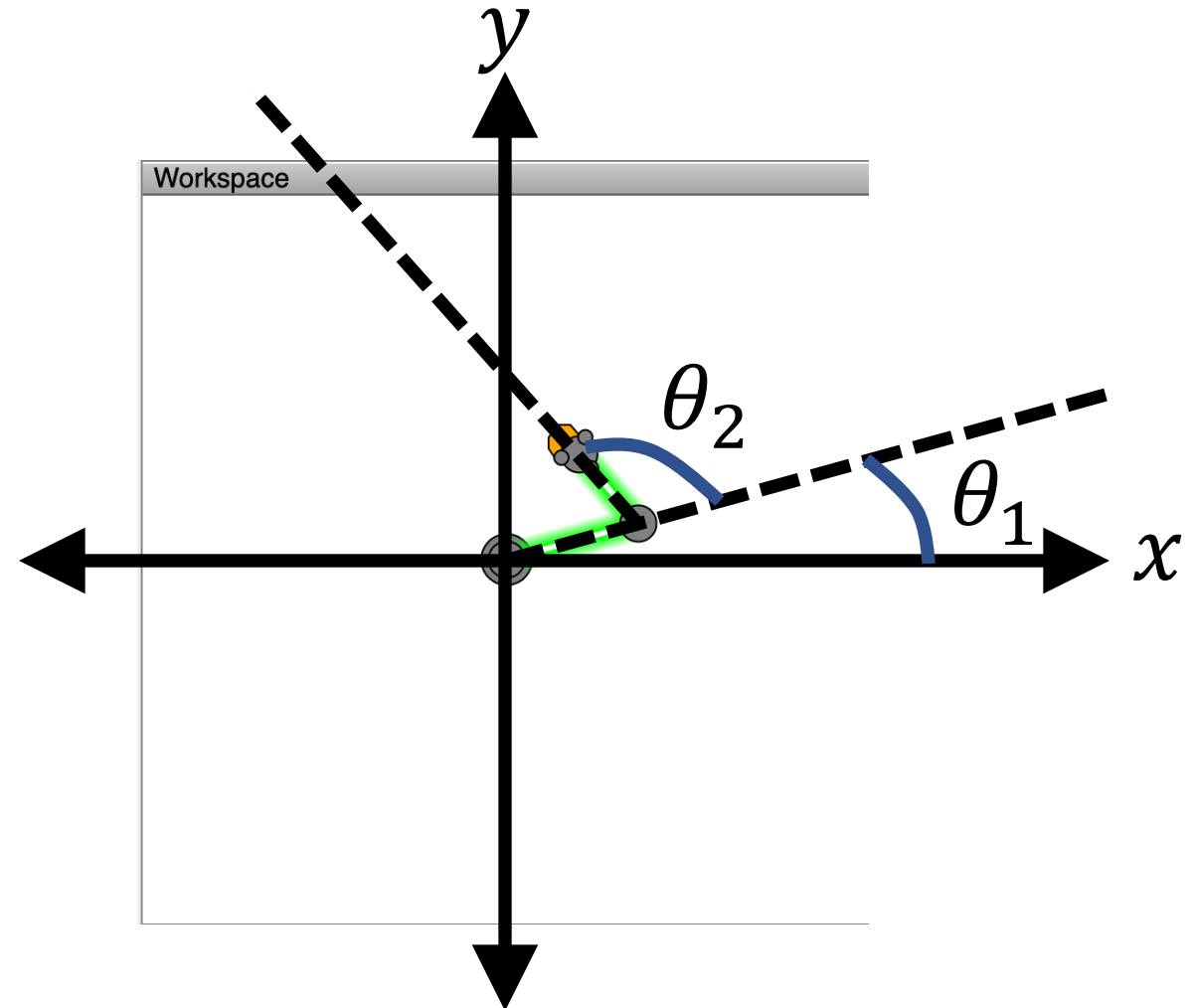
Jeff Ichnowski, University of North Carolina, <https://www.cs.unc.edu/~jeffi/c-space/robot.shtml>

Define some variables:

- $\theta_1$  = shoulder angle
- $L_1$  = length of upper arm
- $\theta_2$  = elbow angle
- $L_2$  = length of lower arm

Then

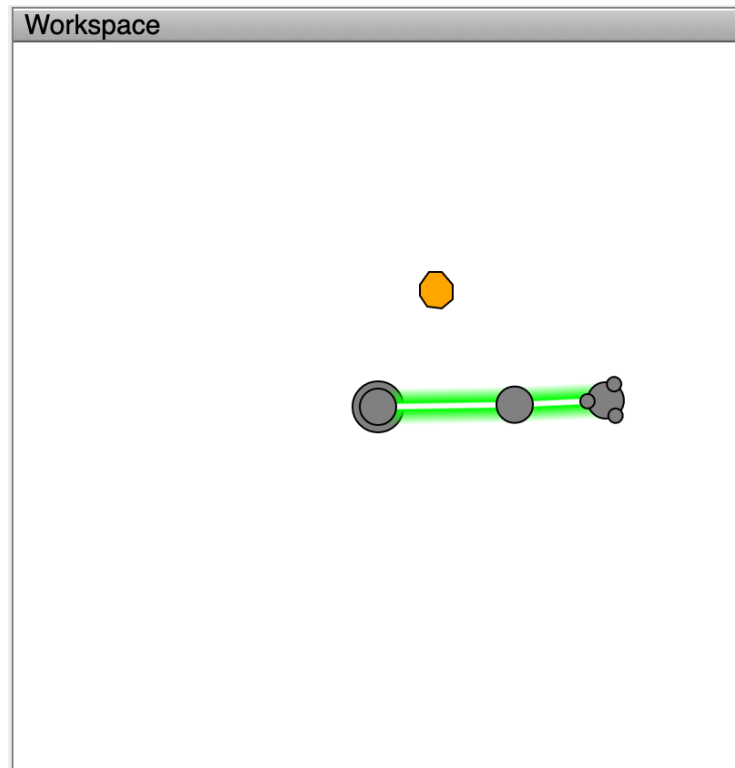
$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$



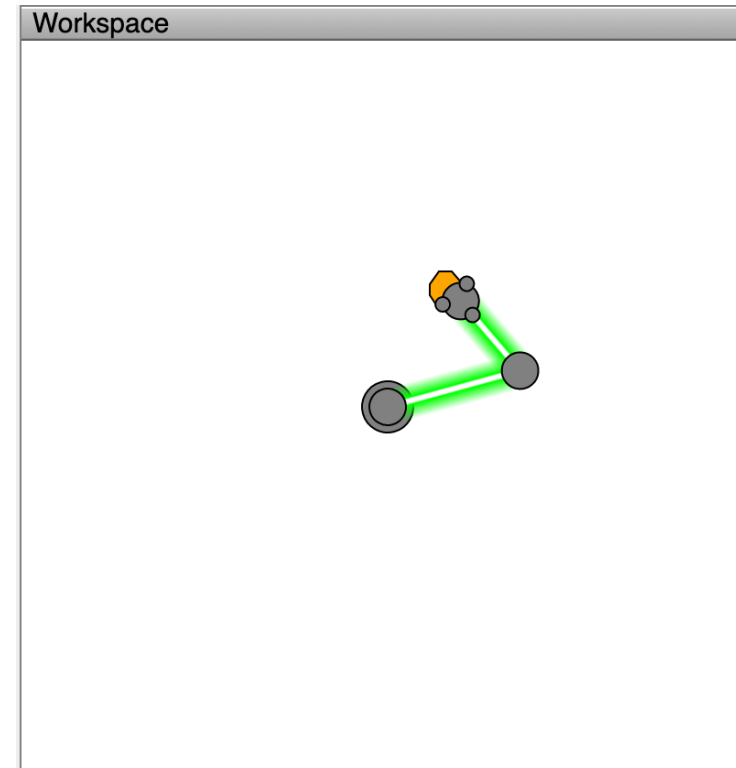
# The Robot Arm Reaching Problem

Jeff Ichnowski, University of North Carolina, <https://www.cs.unc.edu/~jeffi/c-space/robot.shtml>

- Given a robot arm in STARTING VALUES OF  $(\theta_1, \theta_2)$ ,
- how should I adjust  $(\theta_1, \theta_2)$  to most quickly reach GOAL VALUES OF  $(x, y)$ ?

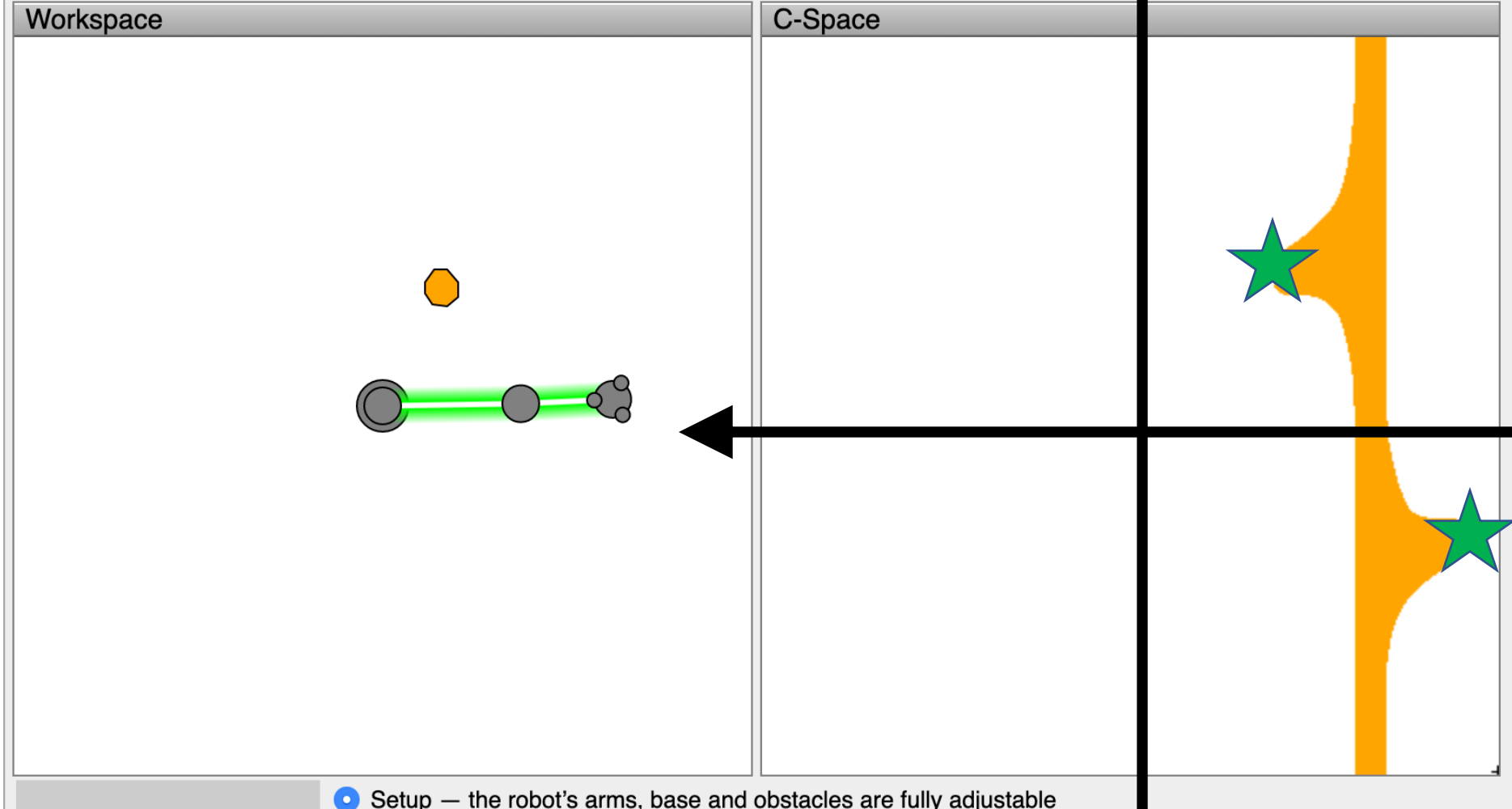


START

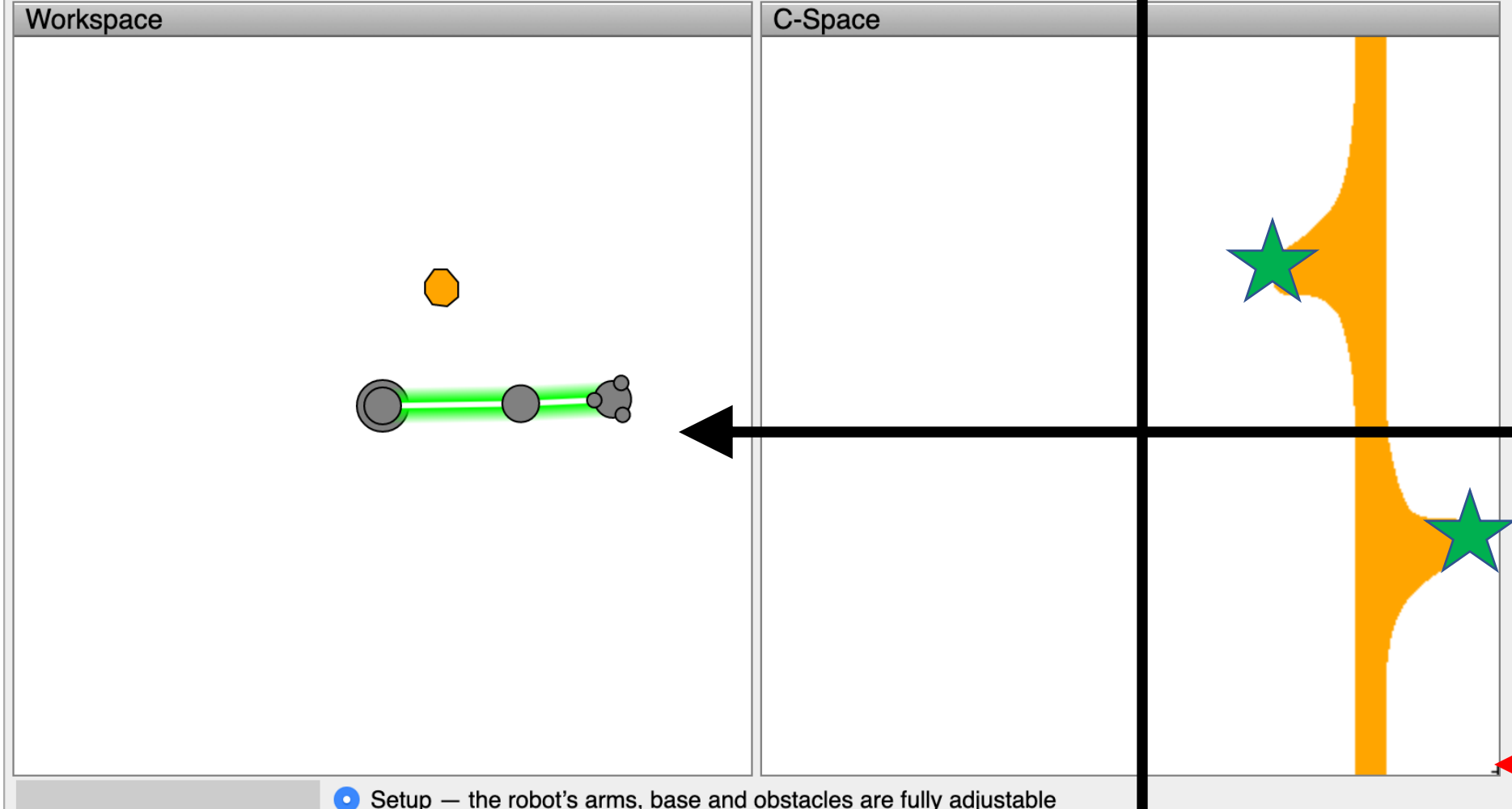


GOAL

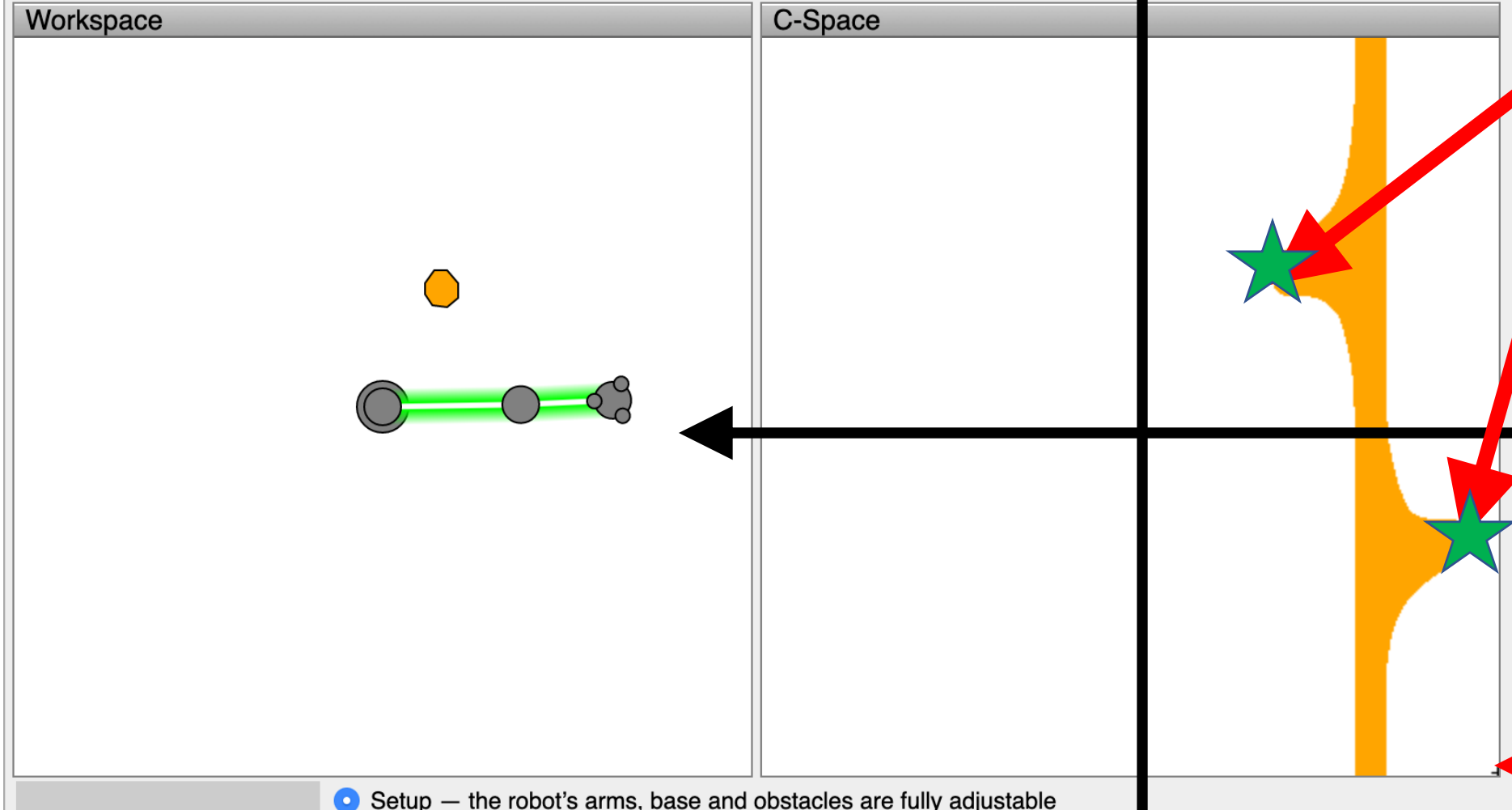
# Configuration Space Visualization of 2-D Robotic Manipulator



# Configuration Space Visualization of 2-D Robotic Manipulator



# Configuration Space Visualization of 2-D Robotic Manipulator

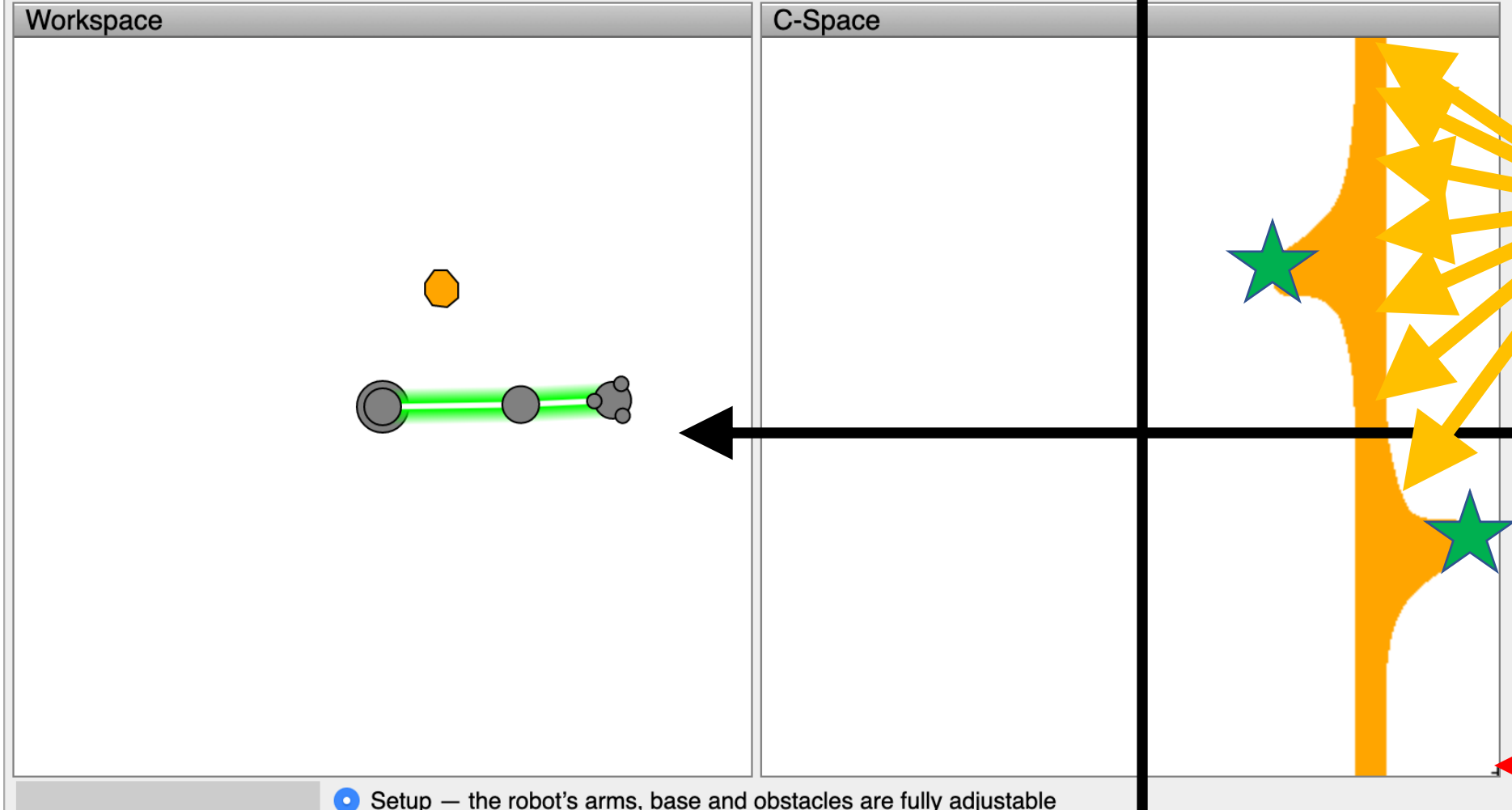


THESE VALUES  
OF  $(\theta_1, \theta_2)$   
CORRESPOND  
TO REACHING  
THE GOAL

STARTING  
VALUES OF  
 $(\theta_1, \theta_2)$

• Setup — the robot's arms, base and obstacles are fully adjustable

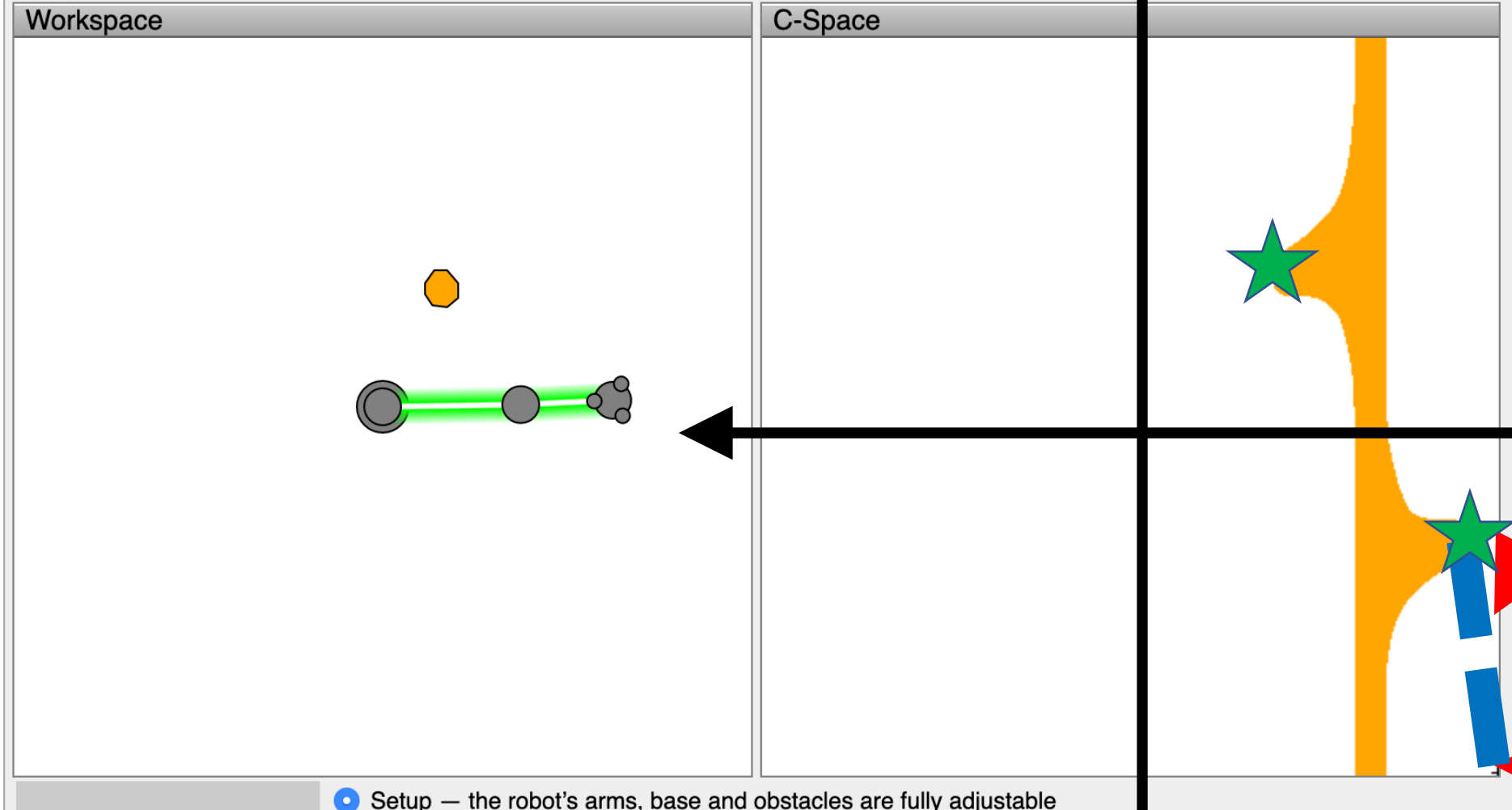
# Configuration Space Visualization of 2-D Robotic Manipulator



Bumping anywhere else on the orange bar means that the upper or lower arm collides, not the hand. Call these regions OBSTACLES.

STARTING VALUES OF  $(\theta_1, \theta_2)$

# Configuration Space Visualization of 2-D Robotic Manipulator

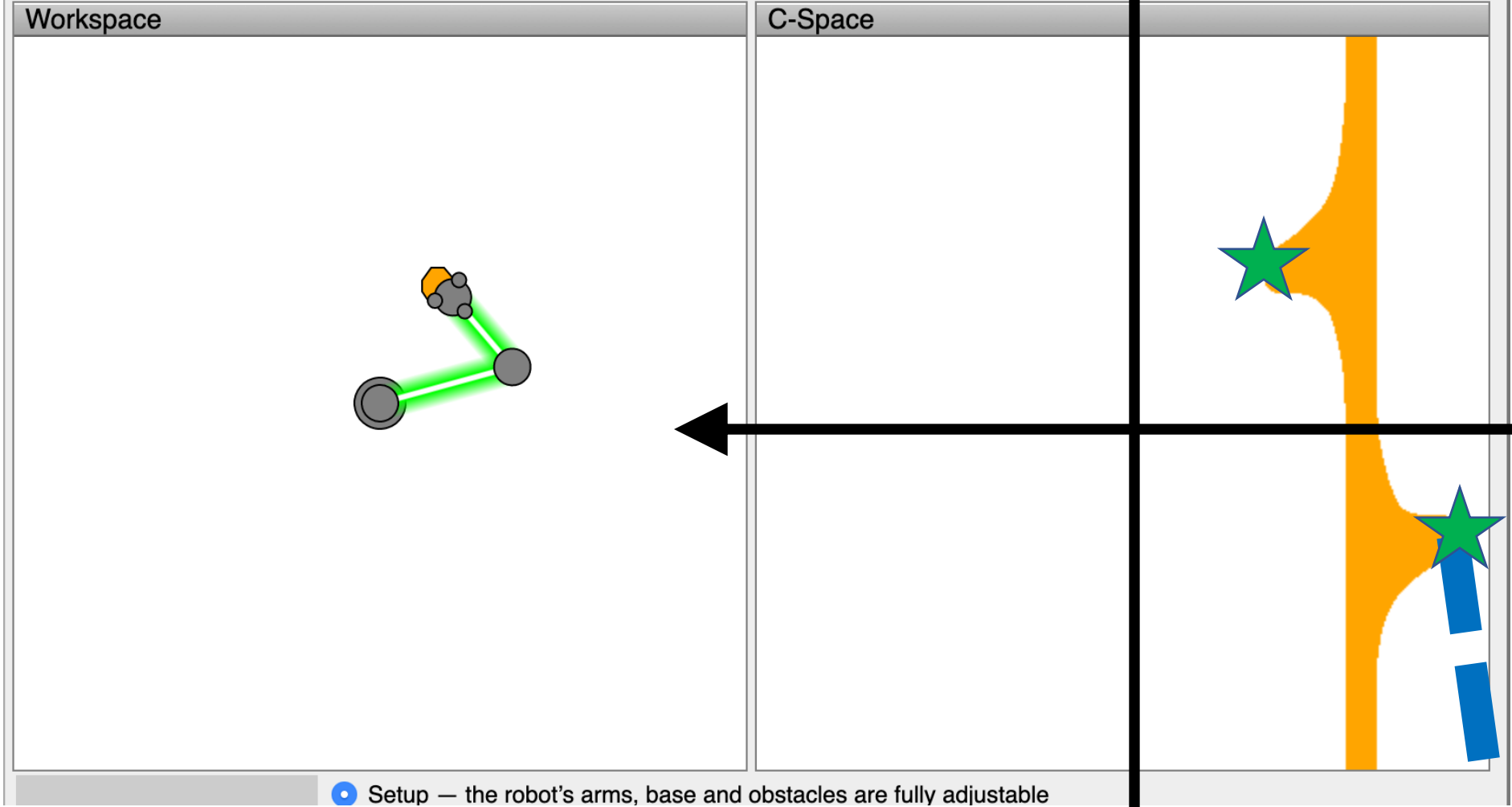


• Setup — the robot's arms, base and obstacles are fully adjustable

This is the shortest path from START to GOAL avoiding all OBSTACLES

STARTING VALUES OF  $(\theta_1, \theta_2)$

# Configuration Space Visualization of 2-D Robotic Manipulator



DONE!

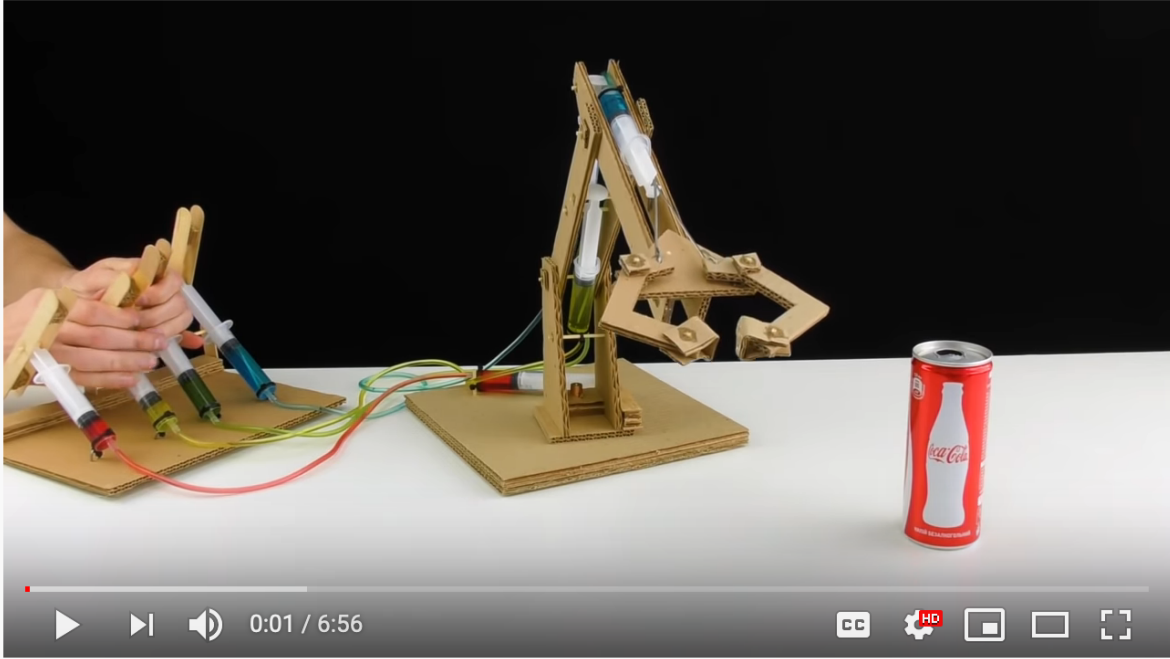


# How to solve the Robot Arm problem

1. Create a configuration space (a space whose coordinates are the set of all configuration parameters for the robot. Two dimensions, for the MP; in the real world, 3d or 4d is more common).
2. Label the START
3. Label the GOAL (there might be more than one set of configuration parameters that is an acceptable way to reach the GOAL).
4. Label the OBSTACLES (convert them from  $(x, y)$  to  $(\theta_1, \theta_2)$ ).
5. Use BFS or A\* to find the shortest path from START to GOAL, avoiding all OBSTACLES.

BTW, this person is really a maestro.  
Watch it again, if you want to.

<https://www.youtube.com/watch?v=P2r9U4wkjcc>



The image shows a YouTube video player interface. At the top, there is a navigation bar with a menu icon, a 'Premium' badge, a search bar containing the text 'robot arm', and icons for video, grid, notifications, and a profile picture. The main video frame displays a hydraulic powered robotic arm constructed from cardboard. A hand is shown adjusting the arm's components. To the right of the arm is a red Coca-Cola can. The video player controls at the bottom include a play button, a progress bar showing 0:01 / 6:56, and icons for closed captions, HD quality, and full screen.

How to Make Hydraulic Powered Robotic Arm from Cardboard