

RECONFIGURING LABOUR

Towards autonomous robots for a moving desk office environment

Aaron Djong with





HANDLING TASKS



GIVE THE TASK TO THE ROBOT



GIVE THE TASK TO THE ROBOT



BETTER OPERATIONS AND USE OF TIME

ROBOTS IN THE WORKPLACE



**ACCURAT
E**



FAST



**CONSISTEN
T**

The background image shows a vast industrial manufacturing facility. Numerous yellow robotic arms, with 'ABB' and 'LEONI' branding visible, are positioned across multiple levels of the factory floor. They are engaged in various tasks, likely assembly or material handling. The environment is filled with complex machinery, conveyor systems, and structural elements of the factory building. The lighting is industrial, and the overall scene conveys a sense of large-scale automated production.

ROBOTIC ARMS



AMAZON WAREHOUSE ROBOTS

A large Amazon warehouse robot, model 2772, is shown in a warehouse setting. The robot is orange and black, with a white circular logo on its side that reads "AMAZON SCIENCE 2772". It is carrying a yellow shelving unit filled with various products, including boxes of "Zing" and "Pelican" brand items. The robot is positioned in front of a large shelving unit with many compartments, each labeled with a number and a QR code. The background shows more warehouse shelving and another robot in the distance.





A wide-angle photograph of a modern office environment. Several employees are seated at white desks, working on computers. The office has large windows in the background, letting in natural light. The image is overlaid with a dark blue filter. The word "BUT" is written in large, white, sans-serif capital letters in the center of the image, with a thin orange horizontal line underneath it.

BUT



PATENT PENDING ©

HARNESS THE BENEFITS OF
FIBRE OPTICS

THE DATA BOOM

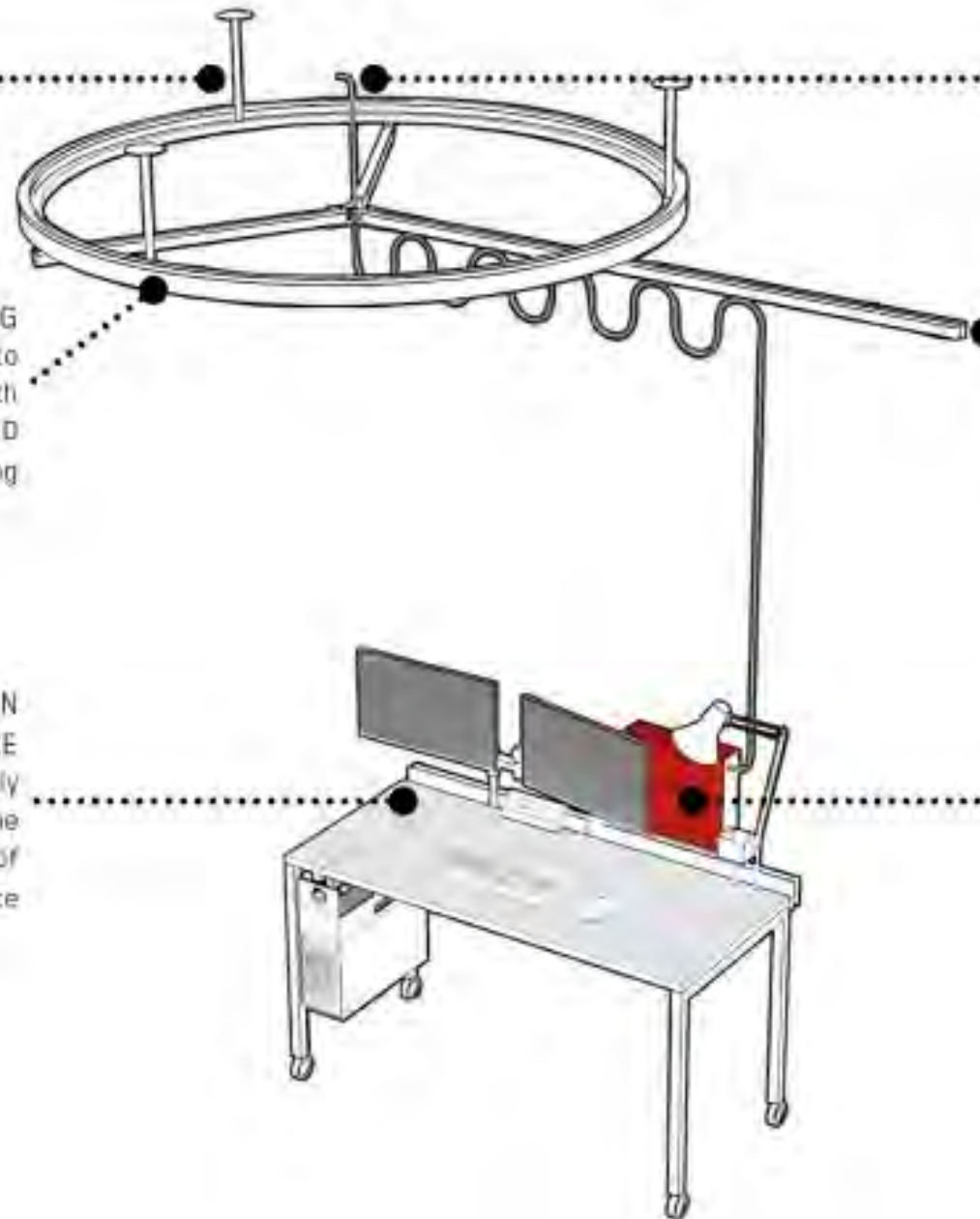
Avoid the constraints of an
umbilical. The 350 degree
rotating boom allows you to
connect your octopus anywhere
within a 2m radius.

INTEGRATED LIGHTING
Avoid glare and add interest to
the floorplate with
integrated and addressable LED
lighting

THE MOBILE WORKSTATION
SYSTEM OF YOUR CHOICE
The Octopus can be easily
configured to interface with the
mobile workstation system of
your preference

THE OCTOPUS

A plug and play system to
allow users to connect up to 8
workstations to a cluster.
No electricians required!



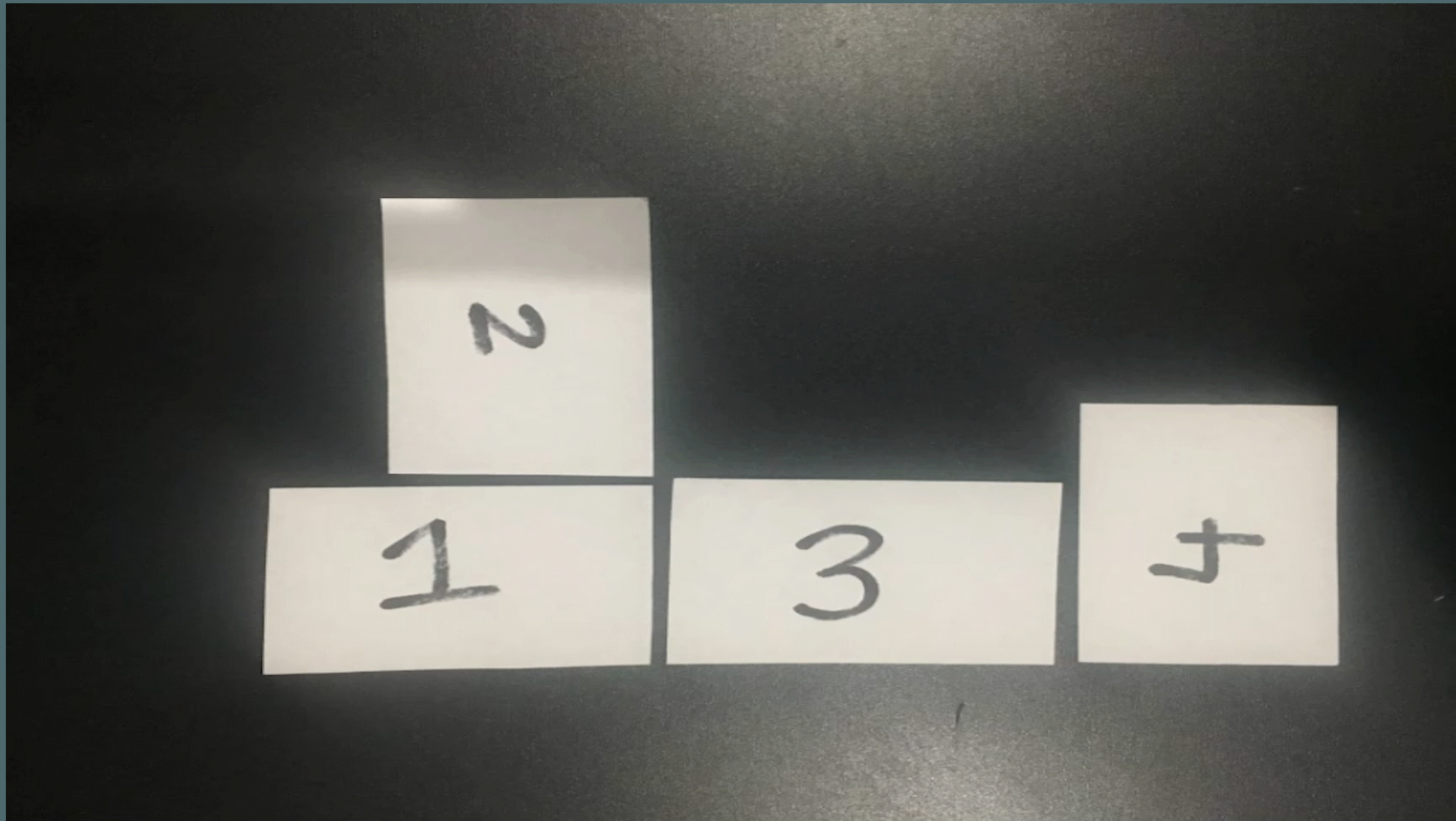
THE PROBLEM



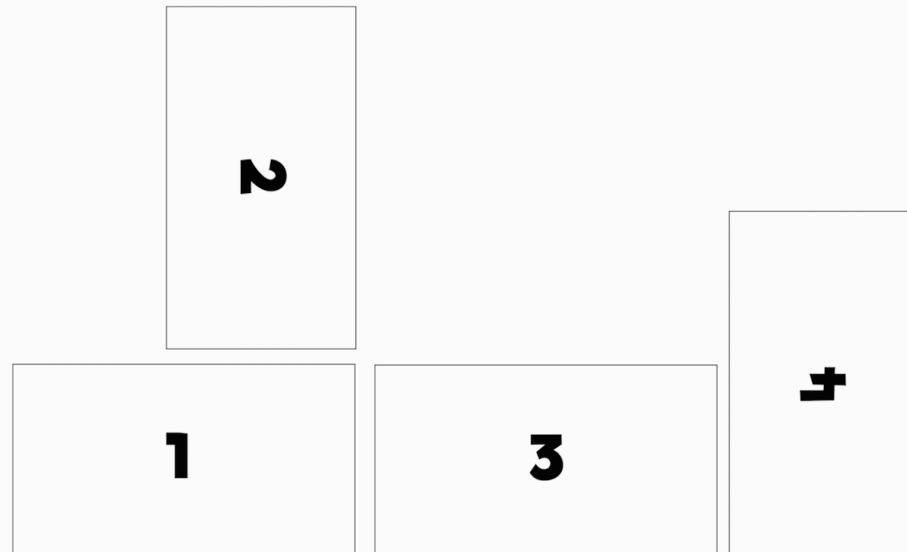


THE QUESTION

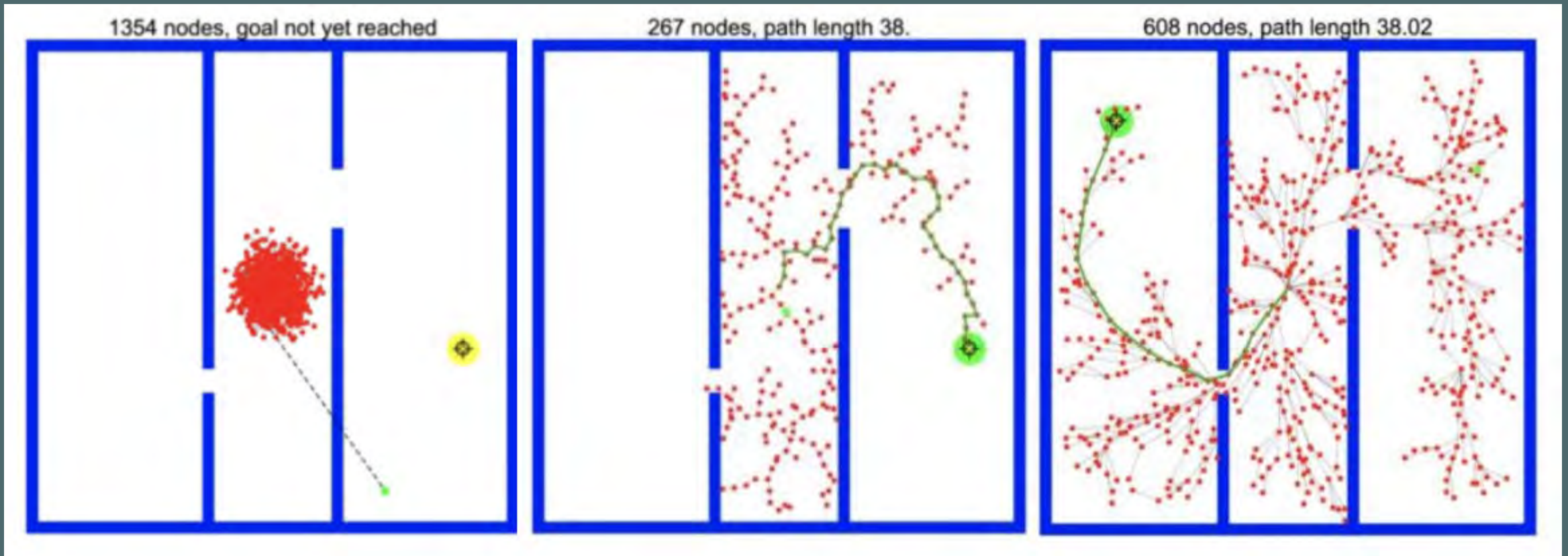
STOP MOTION MOVEMENT



DIGITAL RECREATION MOVEMENT



MOTION PLANNING ALGORITHMS



Random Tree (RT)

Rapidly Exploring
Random Tree (RRT)

Rapidly Exploring
Random Tree* (RRT*)

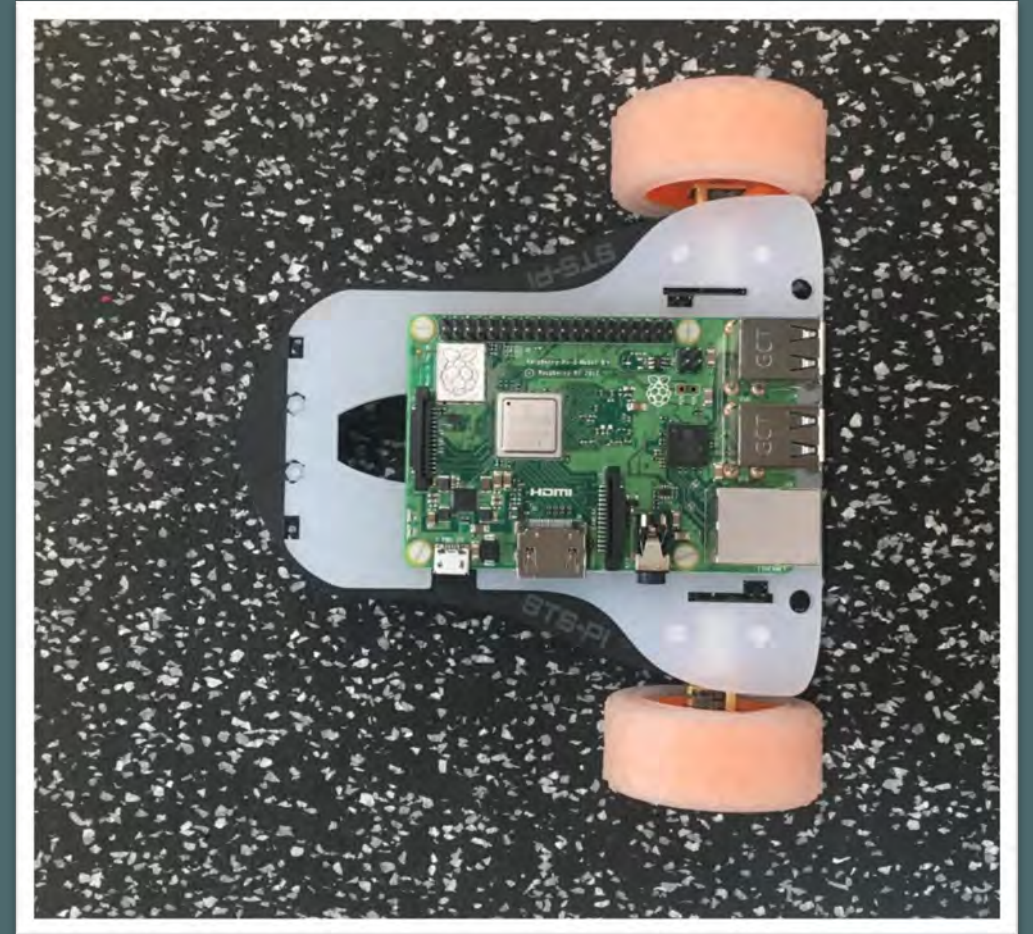
MOTION PLANNING ALGORITHMS

A*

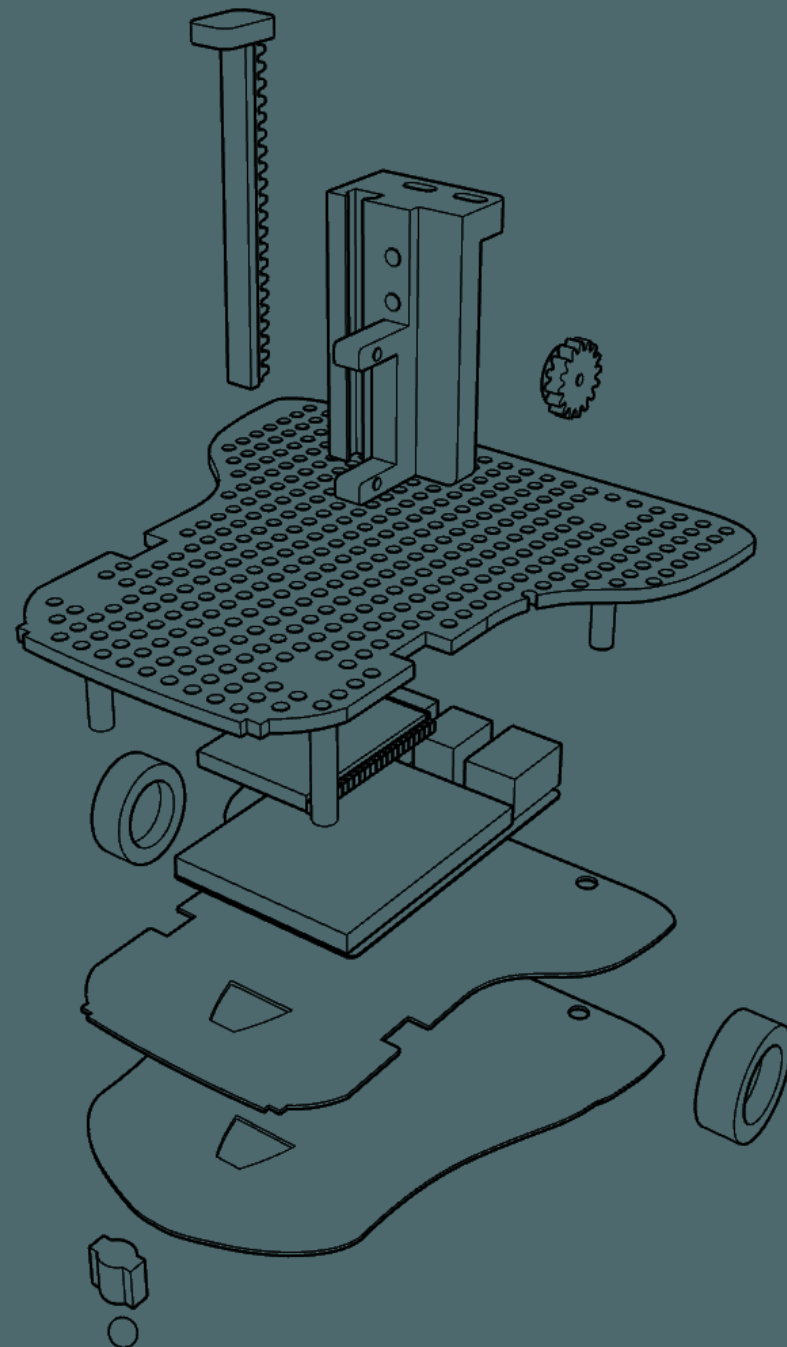
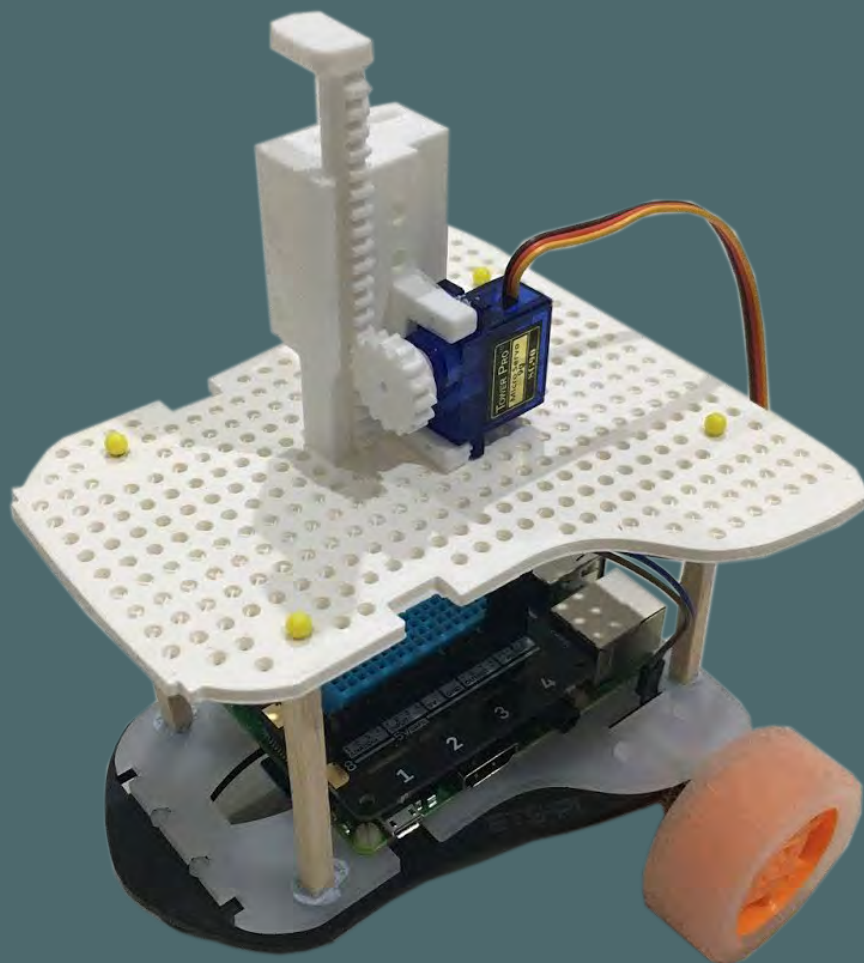


Dijkstra's

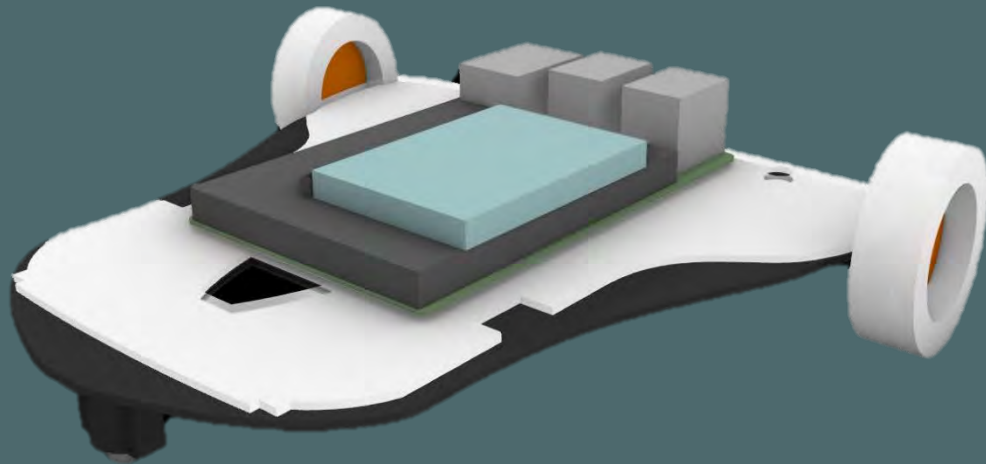
THE BASE ROBOT KIT: STS-PI



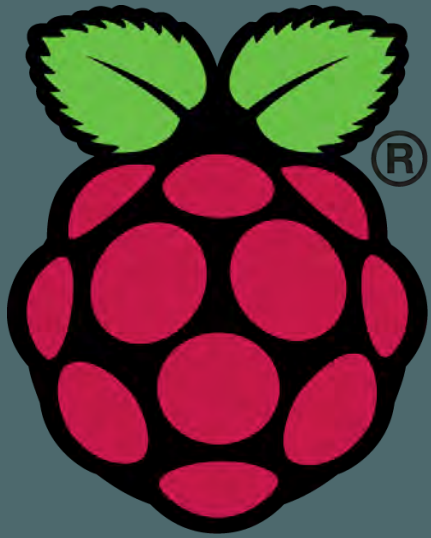
CUSTOMISED



DIGITAL MODEL



PROGRAMMING



PROGRAMMING

```
remotecontrol.py ×
remotecontrol.py
11
12 def readchar():
13     fd = sys.stdin.fileno()
14     old_settings = termios.tcgetattr(fd)
15     try:
16         tty.setraw(sys.stdin.fileno())
17         ch = sys.stdin.read(1)
18     finally:
19         termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
20     if ch == '\x03':
21         raise KeyboardInterrupt
22     return ch
23
24 def readkey(getchar_fn=None):
25     getchar = getchar_fn or readchar
26     c1 = getchar()
27     if ord(c1) != 0x1b:
28         return c1
29     c2 = getchar()
30     if ord(c2) != 0x5b:
31         return c1
32     c3 = getchar()
33     return ord(c3) - 65 # 0=Up, 1=Down, 2=Right, 3=Left arrows
34
35 # End of the functions that read your keyboard
```

Keyboard Key Reader

```
remotecontrol.py • motor.py ×
robot > pi2go > motor.py
49 # Main body of code - this detects your key
50 try:
51     while True:
52         keyp = readkey()
53         if keyp == 'w' or keyp == UP:
54             pi2go.forward(speed)
55             print 'Forward', speed
56         elif keyp == 's' or keyp == DOWN:
57             pi2go.reverse(speed)
58             print 'Backward', speed
59         elif keyp == 'd' or keyp == RIGHT:
60             pi2go.spinRight(speed)
61             print 'Spin Right', speed
62         elif keyp == 'a' or keyp == LEFT:
63             pi2go.spinLeft(speed)
64             print 'Spin Left', speed
65
66         elif keyp == '.' or keyp == '>':
67             speed = min(100, speed+10)
68             print 'Speed+', speed
69         elif keyp == ',' or keyp == '<':
70             speed = max(0, speed-10)
71             print 'Speed-', speed
```

Motor Controls

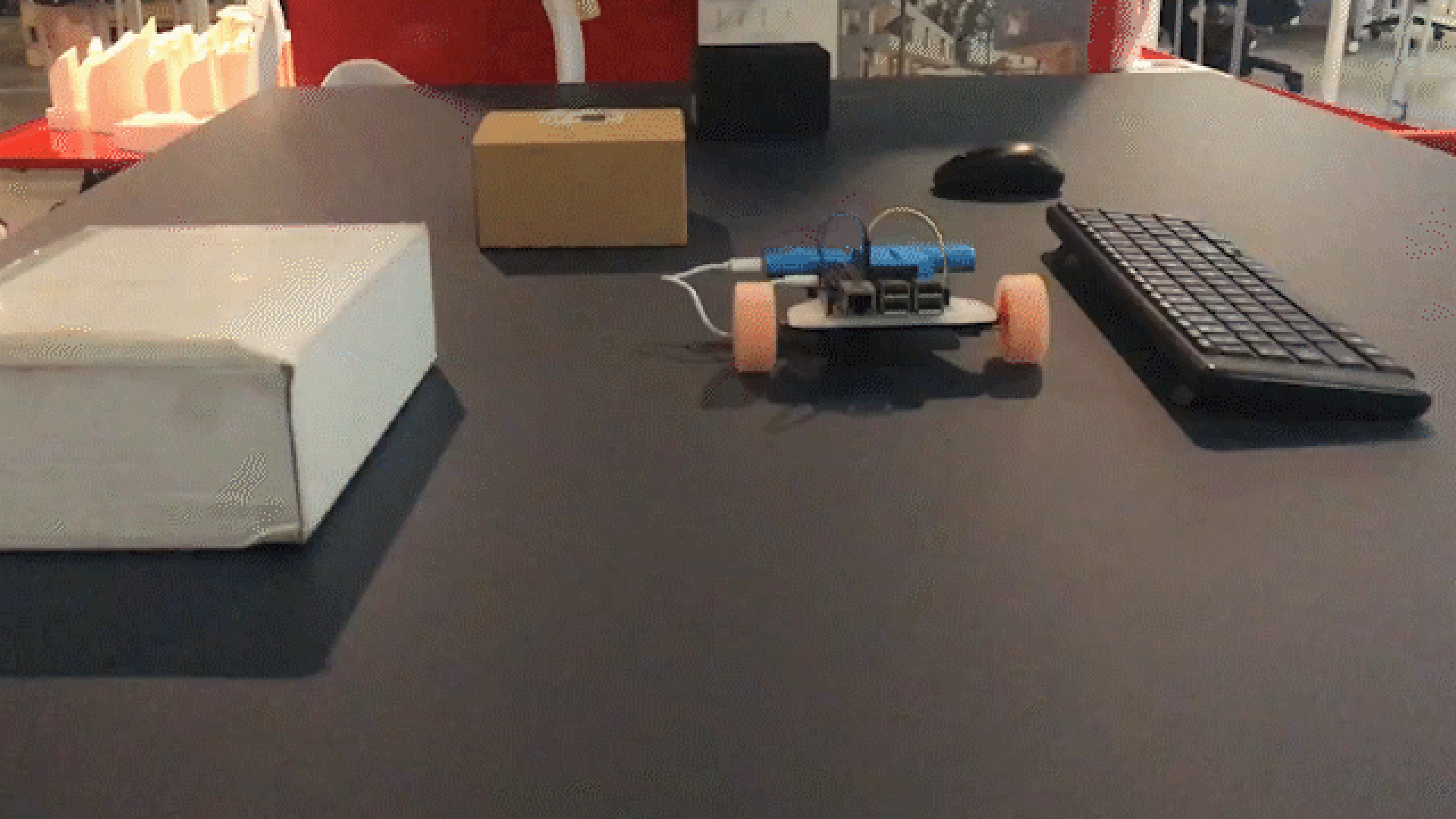
PROGRAMMING

```
remotecontrol.py • motor.py X
robot > pi2go > motor.py
49 # Main body of code - this detects your key
50 try:
51     while True:
52         keyp = readkey()
53         if keyp == 'w' or keyp == UP:
54             pi2go.forward(speed)
55             print 'Forward', speed
56         elif keyp == 's' or keyp == DOWN:
57             pi2go.reverse(speed)
58             print 'Backward', speed
59         elif keyp == 'd' or keyp == RIGHT:
60             pi2go.spinRight(speed)
61             print 'Spin Right', speed
62         elif keyp == 'a' or keyp == LEFT:
63             pi2go.spinLeft(speed)
64             print 'Spin Left', speed
65
66         elif keyp == '.' or keyp == '>':
67             speed = min(100, speed+10)
68             print 'Speed+', speed
69         elif keyp == ',' or keyp == '<':
70             speed = max (0, speed-10)
71             print 'Speed-', speed
```

Original Script

```
remotecontrol.py •
remotecontrol.py
40 # and changes direction depending on it
41 try:
42     while True:
43         keyp = readkey()
44         if keyp == 'w' or keyp == UP:
45             explorerhat.motor.forward(speed)
46             print 'Forward', speed
47         elif keyp == 's' or keyp == DOWN:
48             explorerhat.motor.backwards(speed)
49             print 'Backward', speed
50         elif keyp == 'd' or keyp == RIGHT:
51             explorerhat.motor.one.backwards(speed)
52             explorerhat.motor.two.forwards(speed)
53             print 'Spin Right', speed
54         elif keyp == 'a' or keyp == LEFT:
55             explorerhat.motor.two.backwards(speed)
56             explorerhat.motor.one.forwards(speed)
57             print 'Spin Left', speed
58         elif keyp == '.' or keyp == '>':
59             speed = min(100, speed+10)
60             explorerhat.motor.forward(speed)
61             print 'Speed+', speed
62         elif keyp == ',' or keyp == '<':
63             speed = max (0, speed-10)
64             explorerhat.motor.forward(speed)
65             print 'Speed-', speed
```

Fixed Script



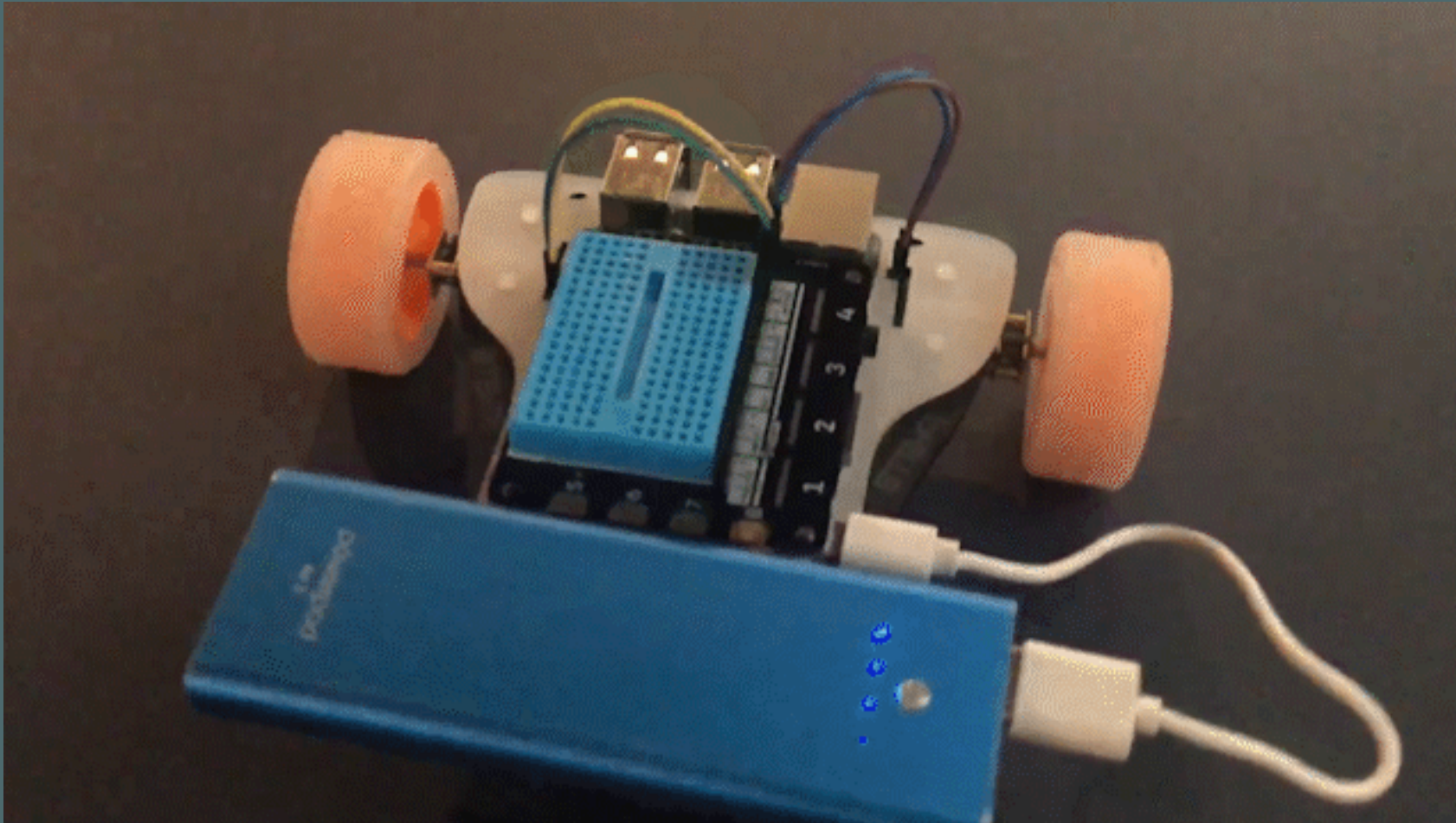
OUTCOMES

**CENTER POINT
OF ROTATION**

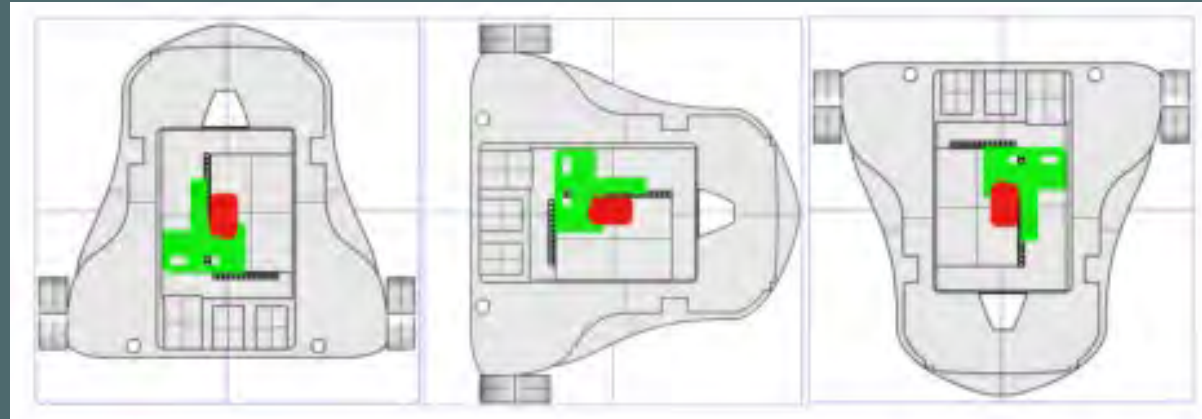
**WEIGHT
DISTRIBUTION**

**ENTRY AND
EXIT**

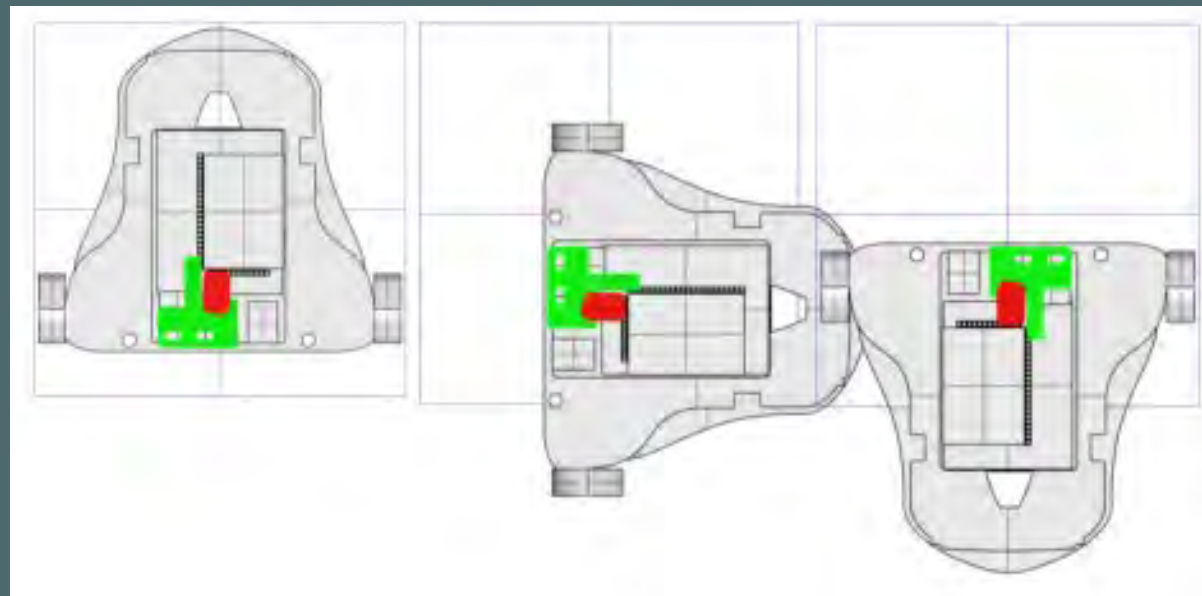
CENTER POINT OF ROTATION



CENTER POINT OF ROTATION

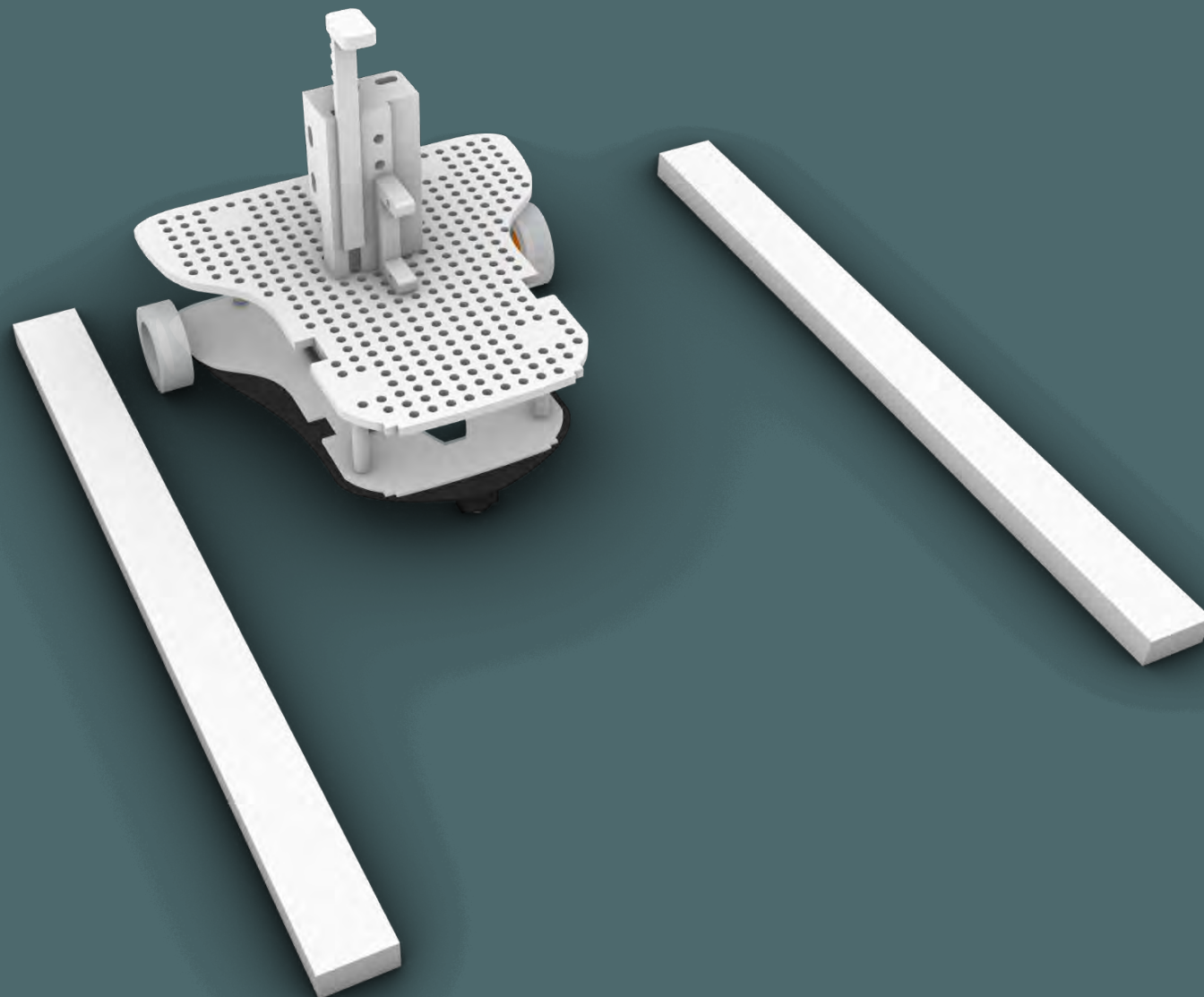


Wanted Rotation

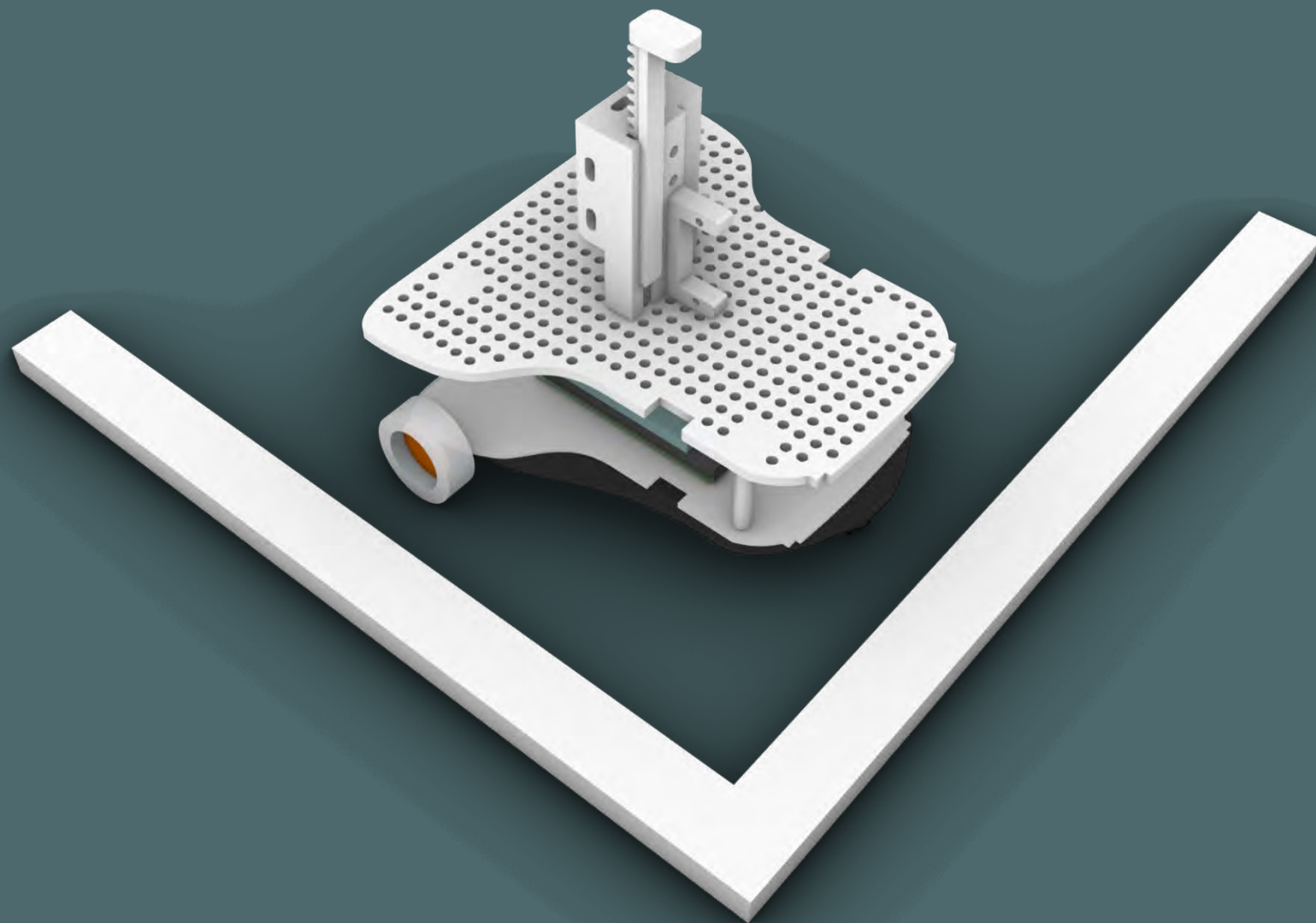


Achieving that Rotation

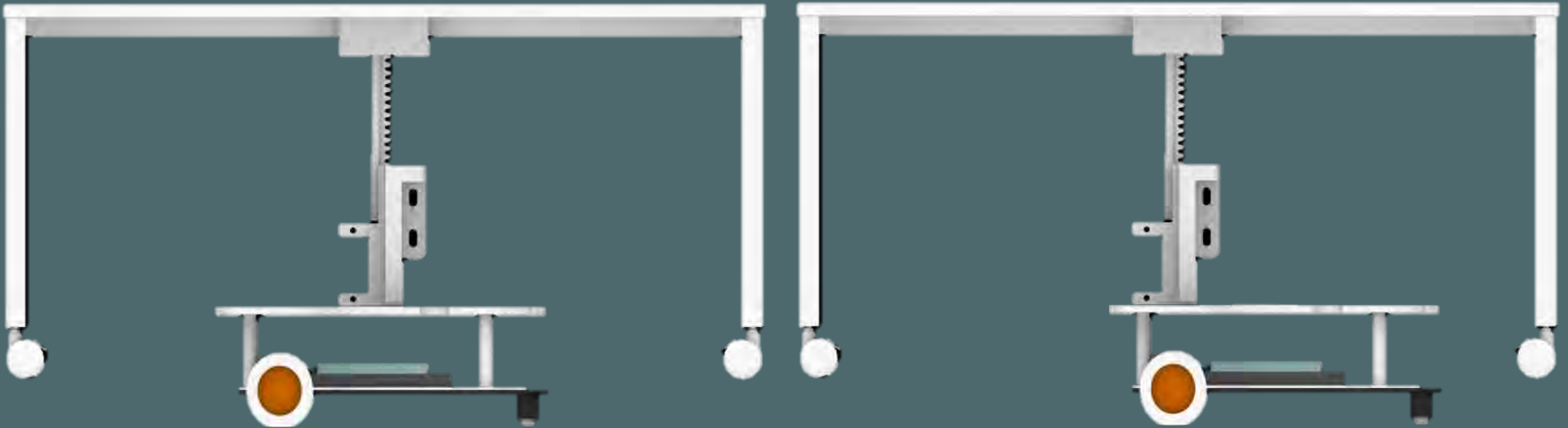
CENTER POINT OF ROTATION



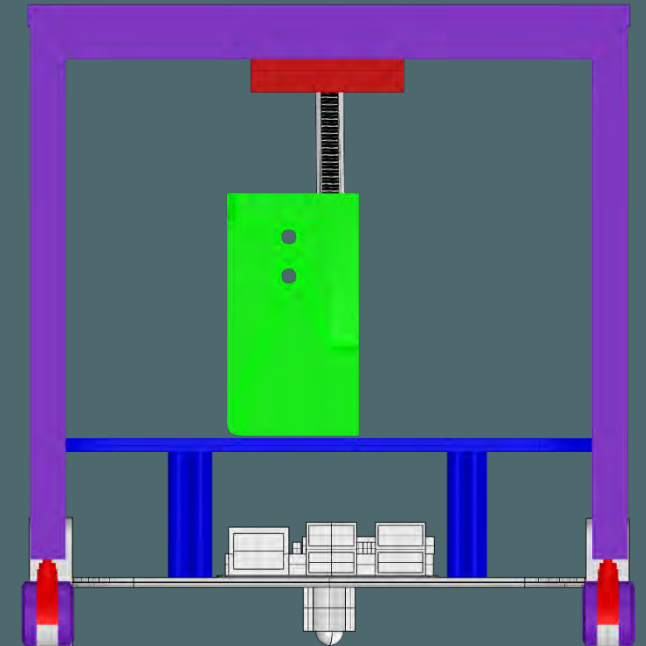
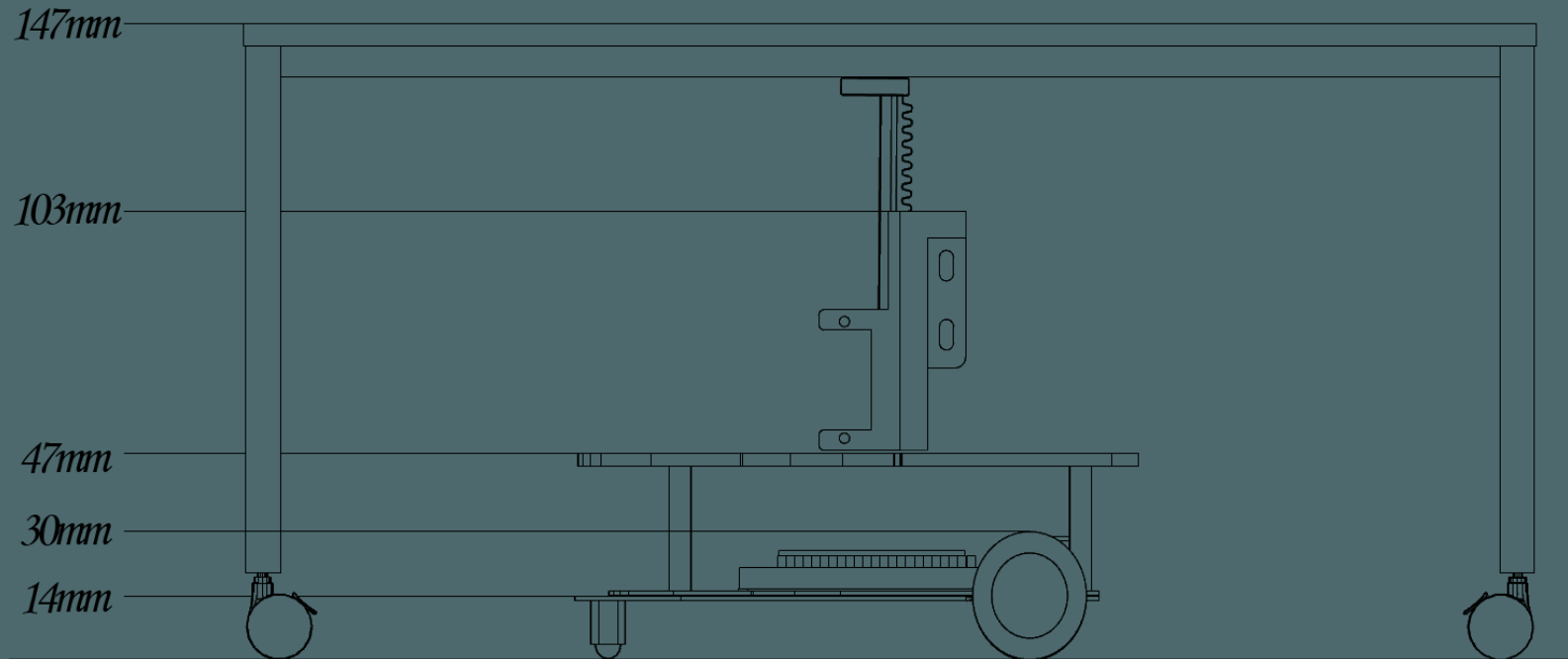
CENTER POINT OF ROTATION



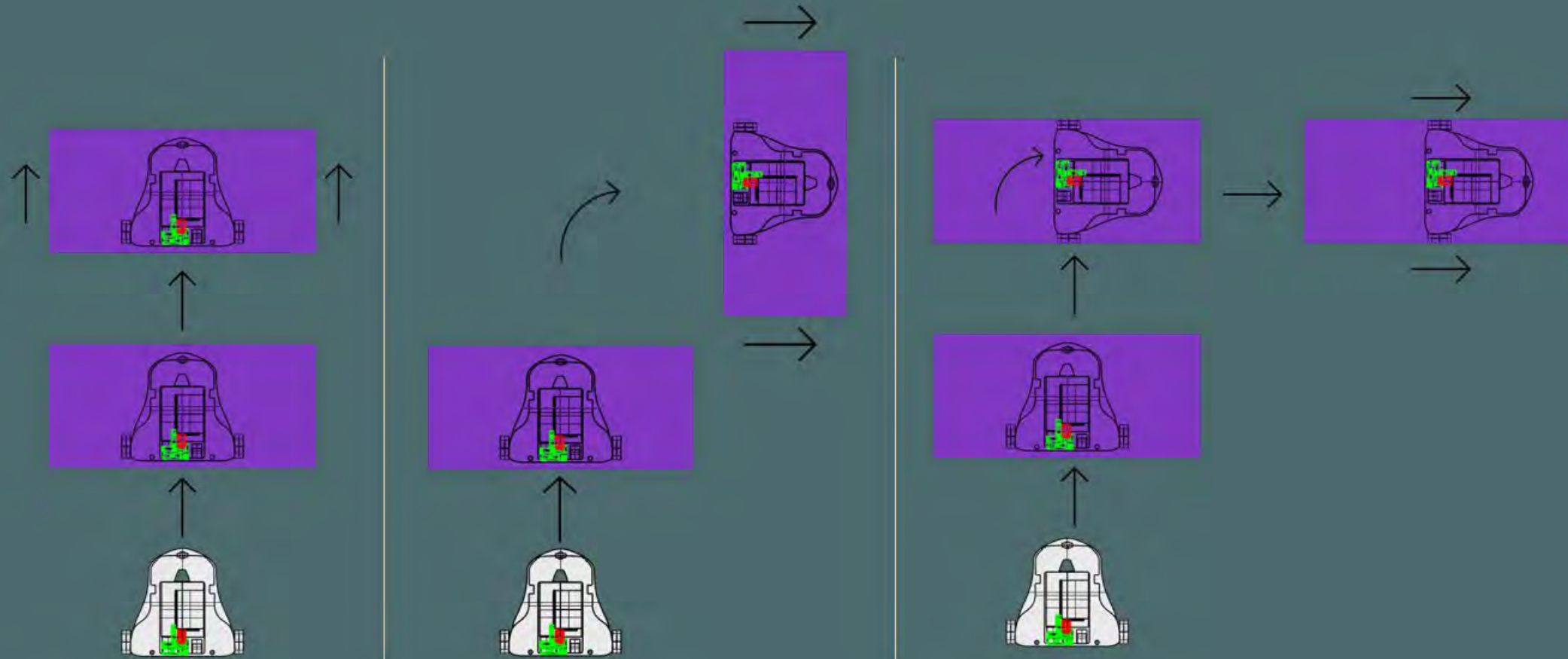
WEIGHT DISTRIBUTION



APPROACHING THE DESK (ENTRY AND EXIT)



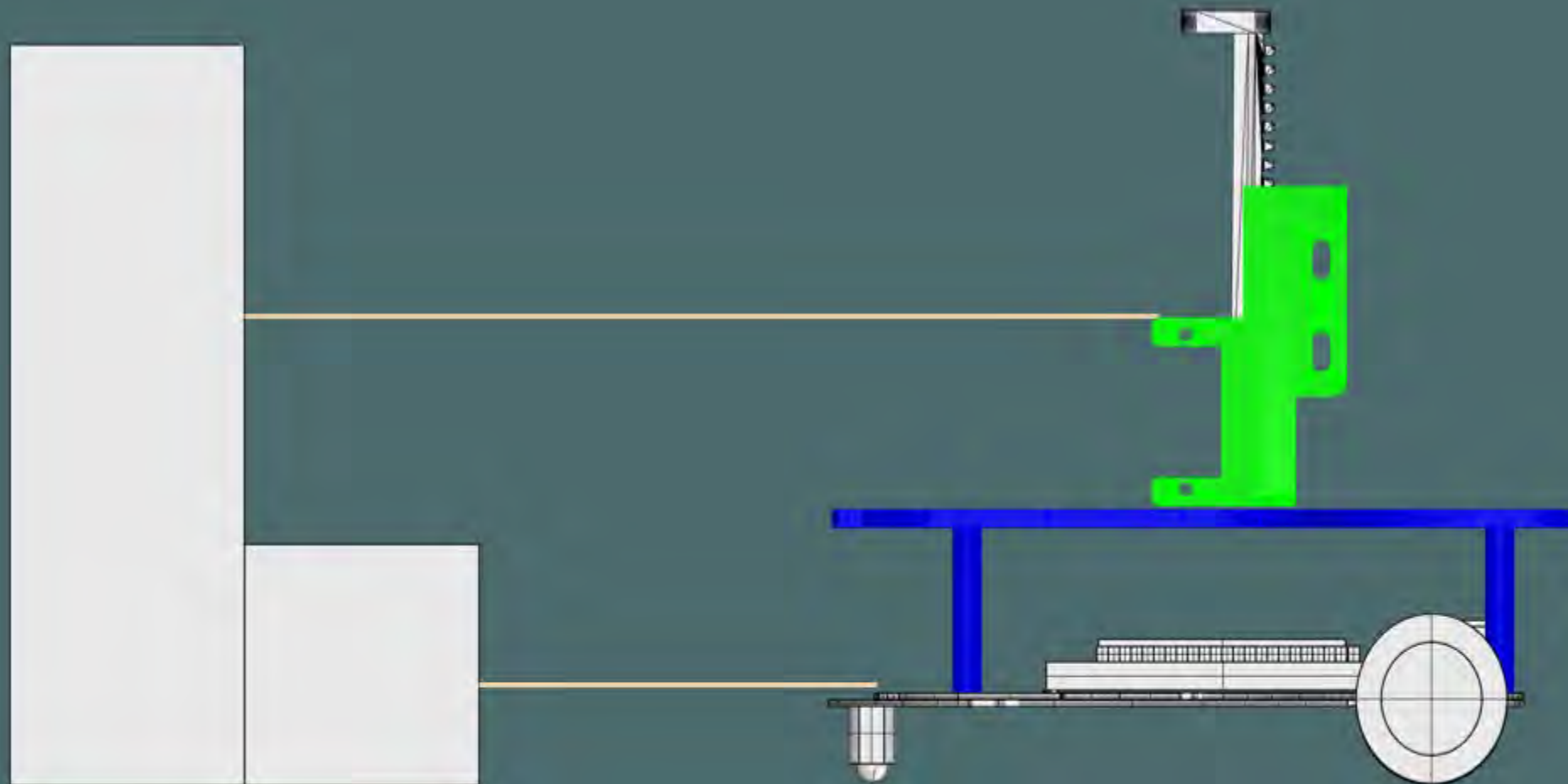
APPROACHING THE DESK (ENTRY AND EXIT)





FUTURE DEVELOPMENTS AND CONCLUSION

FUTURE DEVELOPMENTS: AWARENESS



FUTURE DEVELOPMENTS: THE LIFT



[illegible]