C - Underground Exploration

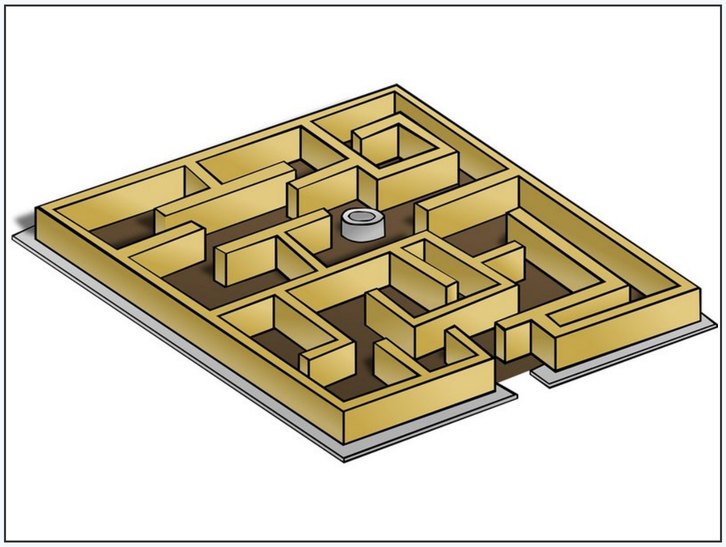
You've discovered an underground system of tunnels under the planet surface, but they are too dangerous to explore! Let's get our robot to explore instead.

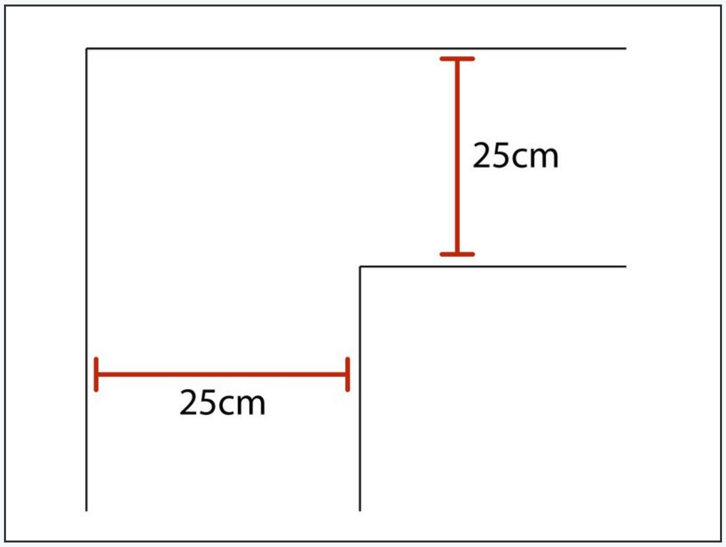
INTRODUCTION

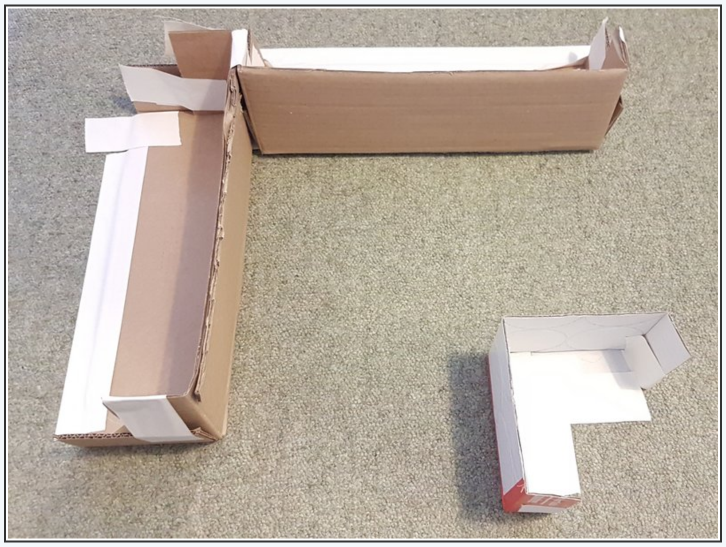
You've discovered an underground system of tunnels under the planet surface, but they are too dangerous to explore! Let's get our robot to explore instead.

Step 1

Make the Underground Tunnels







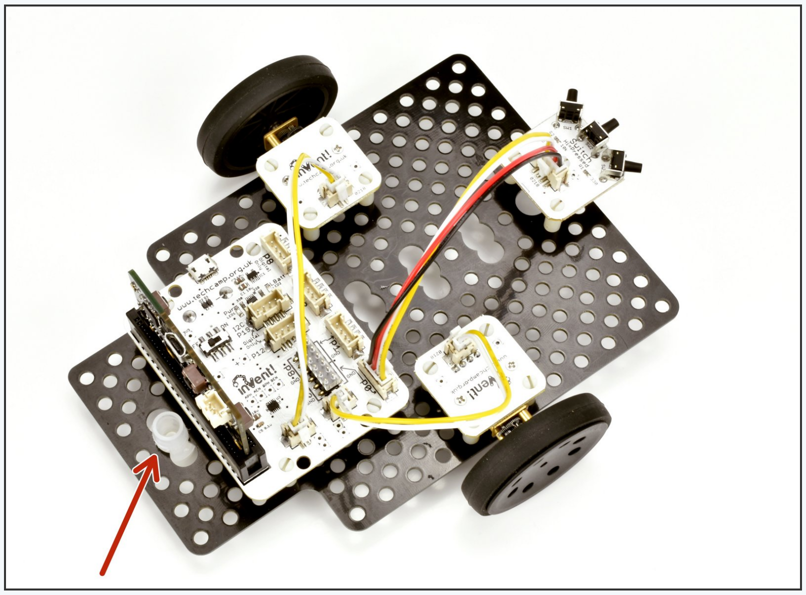
- First, we need to make a system of tunnels to **test** the robot exploration program with. You can do this **individually**, or in **groups.**
- The tunnel system under the planet is made of **straight walls** all at **90 degrees** to each other, like the maze in the picture.
- For now, make a **small section** of maze like the second picture - just a **simple right turn.**
- You can use books, cardboard and tape or anything else sensible you can think of! **Make sure the walls are taller than your robot.**

⚠ The walls must be **at least 25cm apart** so your robot has room to turn - this is **very important!**

Step 2

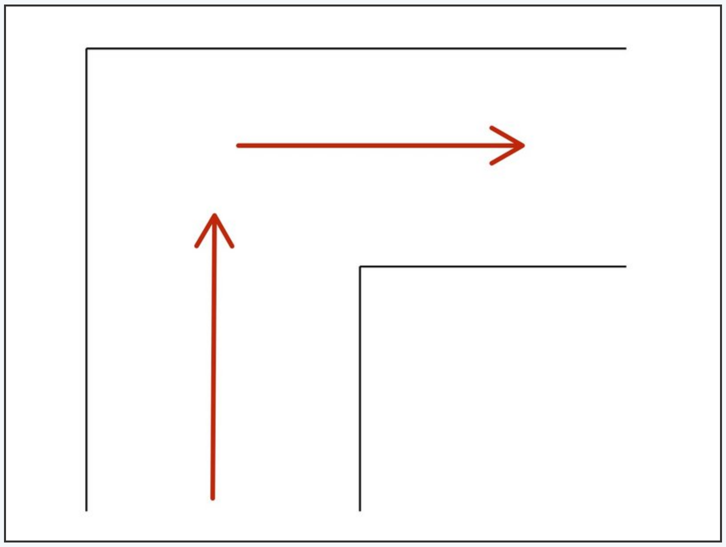
Setup the Robot

- Setup your robot like the picture - make sure everything is in **exactly the right place** or your robot won't **balance properly.**
- The left motor should be in **M1**, the right motor in **M2,** and the switch in **P0**.
- The **trackball** goes at the back to keep the robot stable.



Step 3

Right Turn



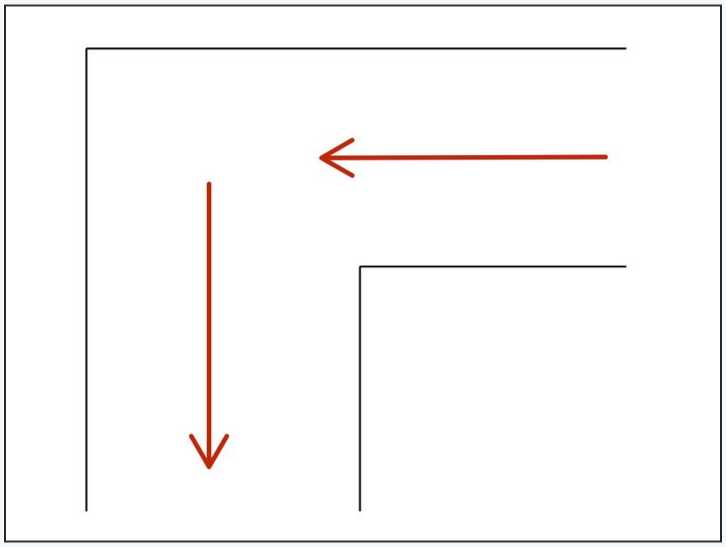
- Let's write a simple program to make the robot navigate the **right turn.** Your program should:
    - Drive **forwards**
    - **If** the switch is pressed, **reverse slightly,** then **turn right 90 degrees**
    - Drive **forwards** again
- You should just need **1** IF statement to complete this - your code should be quite similar to the obstacle avoider in the last lesson!

Be sure to **test it properly** on your maze section until it works reliably!

If you are having trouble getting your turns to be accurate, you can always try **slowing down your motors.**

Step 4

Left Turn

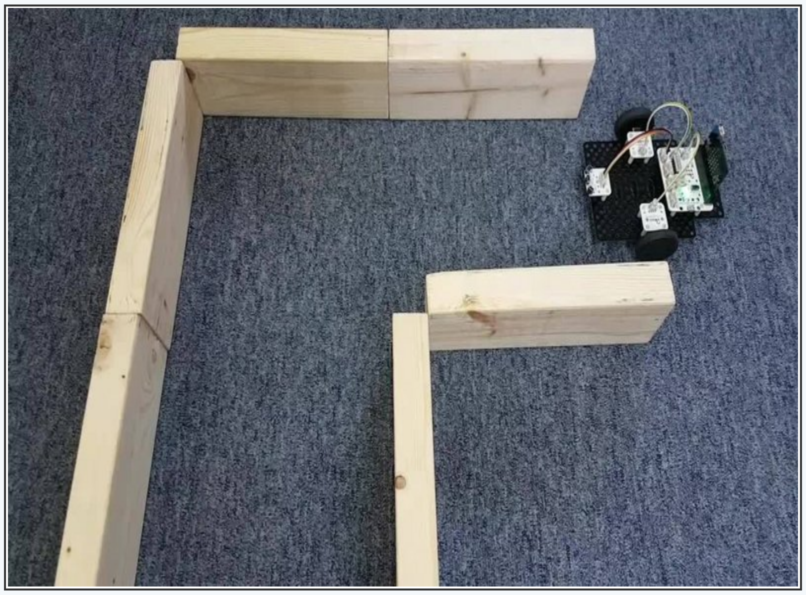


- When your robot is able to make the right turn correctly, try running it through the maze section from the **other direction**, to try a **left turn.**
- Did it behave how you expected?
- Your robot probably **turned right** 90 degrees, **hit** the other wall, **turned right** 90 degrees again and went **back** where it came from!
- This is no good - the robot will never make it through the tunnels! **Can you think how to fix it?** Have a think then move on to the next step.

Step 5

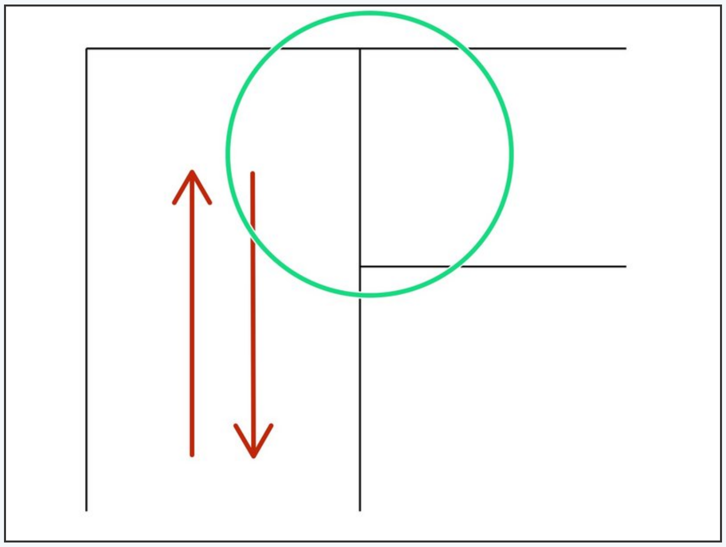
Fixing the Left Turn

- To fix this, we need to write a program that can **work out** whether we need to turn **left** or **right.**
- If you think you know how to do this, great - **try it out!** If not, here is a way that might work:
  - If the switch is pressed, **always turn right 90 degrees.**
  - Move forward a **small amount**, and if the switch is pressed **again,** we must be at a **left turn!**
  - **Spin 180 degrees**, then continue driving forwards
- **Check out the video** for how your robot should handle the left turn if you're still not sure!
- Here's a hint - you will need to put an If statement **<u>inside</u> another IF statement.**



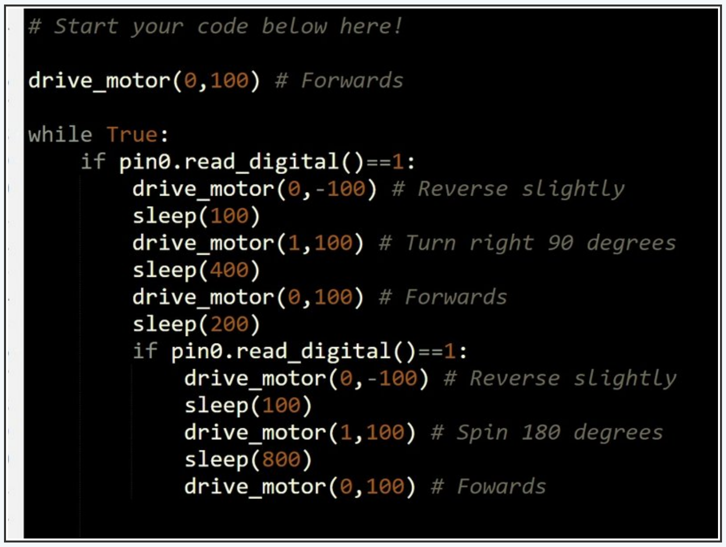
Step 6

Dead Ends



- Now our robot can handle almost anything underground, but what about a **dead end?**
- **Add another IF statement** to your program to check if the switch is being pressed after the 180 degree turn, and if it is, **turn back!**
- There's some **example code** for a program that can deal with left and right turns in the second image if you are stuck.
- **Add another wall** to your test maze so you can properly test your code!

⚠ Now things are getting complicated, don't forget to **comment** your code properly.



```
# Start your code below here!

drive_motor(0,100) # Forwards

while True:
    if pin0.read_digital()==1:
        drive_motor(0,-100) # Reverse slightly
        sleep(100)
        drive_motor(1,100) # Turn right 90 degrees
        sleep(400)
        drive_motor(0,100) # Forwards
        sleep(200)
        if pin0.read_digital()==1:
            drive_motor(0,-100) # Reverse slightly
            sleep(100)
            drive_motor(1,100) # Spin 180 degrees
            sleep(800)
            drive_motor(0,100) # Fowards
```

This document was generated on 2022-01-01 09:47:18 PM (MST).

Step 7

Test the Full Maze!

- Time for a real test of your program!
- As a group, **combine** all the small maze pieces into **one large maze,** with at least one left turn, one right turn and a dead end. The bigger the better!
- **Time** each other's robots and see who can get through the maze the fastest.
- You will probably want to do some **test runs** first so you can adjust your program so it is as fast as possible.

Step 8

Two Switch Sensors

- For the super advanced explorers, you could try using **two switch modules** like in the obstacle avoidance challenge, to see if you can do the maze any faster.
- You could also experiment with the **positioning** of the wheels, switches and trackball on your robot to see which positions work the best.
- In short, try experimenting with **anything** you think might **improve the performance** of you robot!