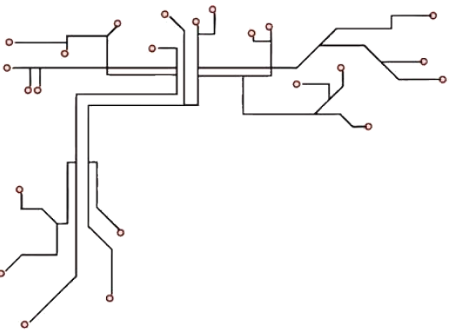
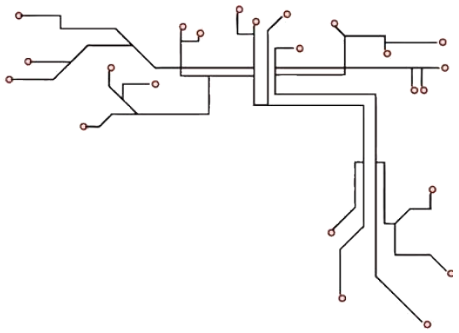
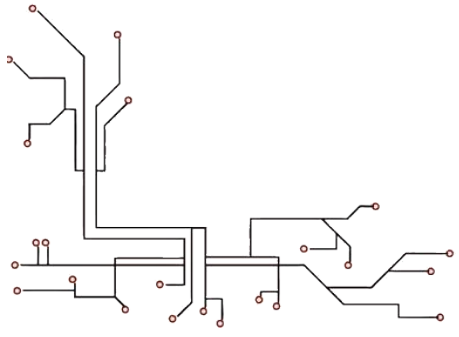
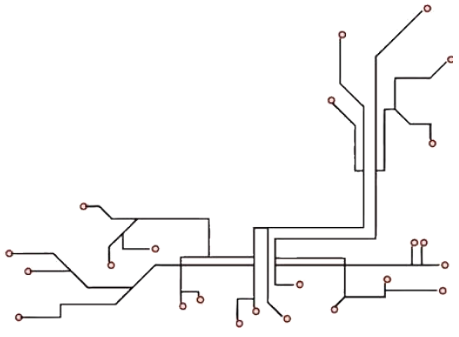



Rescue

GUI & C How-To

A POSTERIORI

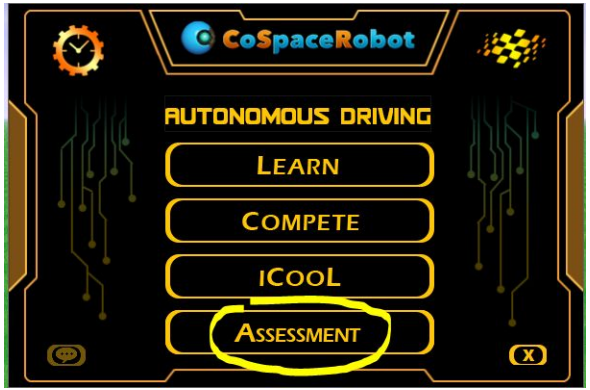
Play · Experience · Learn

NRC ASSESSMENT PLATFORM

A POSTERIORI
Play · Experience · Learn

In the new CoSpace Rescue (CSR 2023) platform, select Assessment, and your choice of coding environment.

We have had our share of woes with CoSpace and Python, so we recommend C.



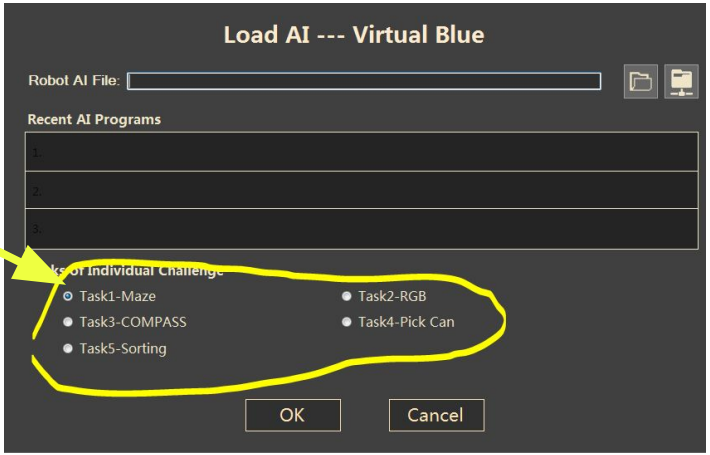
SIMULATOR & TASKS

A POSTERIORI
Play · Experience · Learn

In the new simulator, you will find a single 3D map.

However, you can change the TASK.

And you will note that the starting point of the robot changes between tasks.

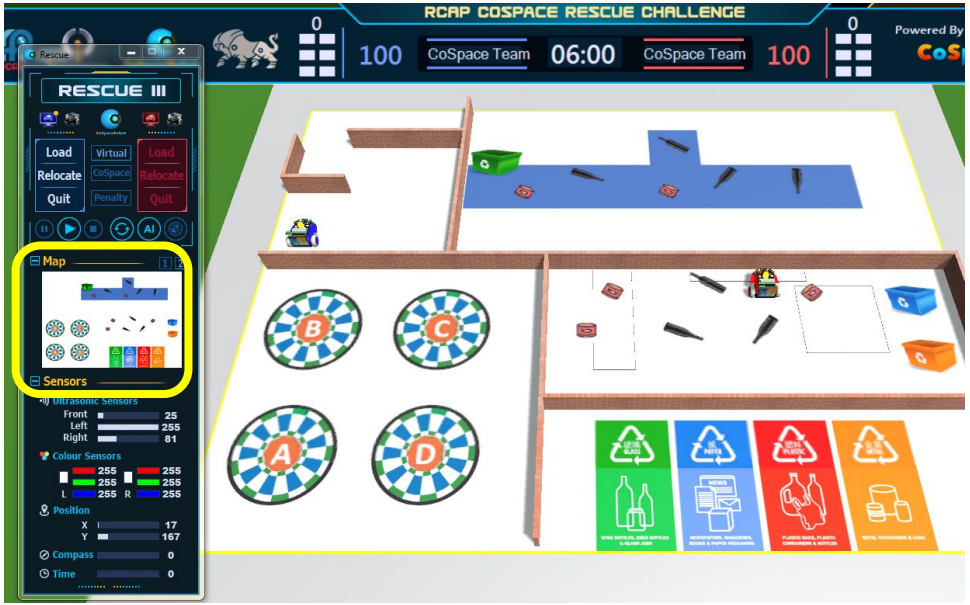


THE ROBOT

Try to manipulate the robot manually within the environment - focus on “Control Panel” window and use the Map widget as well as keyboard arrow keys.

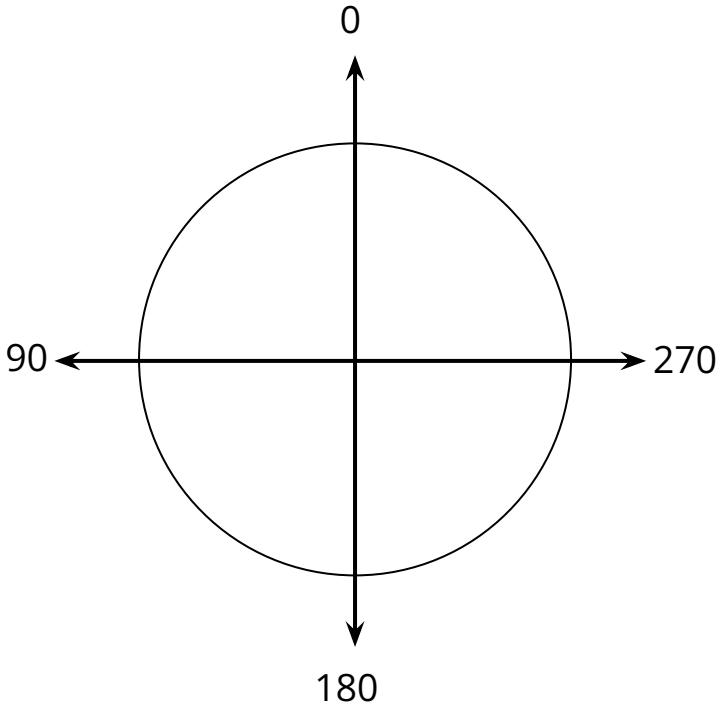
Familiarize yourself with the different RGB color ranges of various objects and field elements.

Look at the way the sensors (Ultrasonic, Compass, Position) respond to various changes.



Compass

Compass refers to the gyro sensing the rotation about the axis that is going through the car from the top to the bottom. Note the angles as you rotate the car 360 degrees.

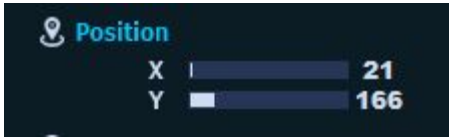


GPS / Position

A POSTERIORI
Play · Experience · Learn

You can use the position information to:

- Stay within certain boundaries
- Decide when to change state (Task 3)
- And more...

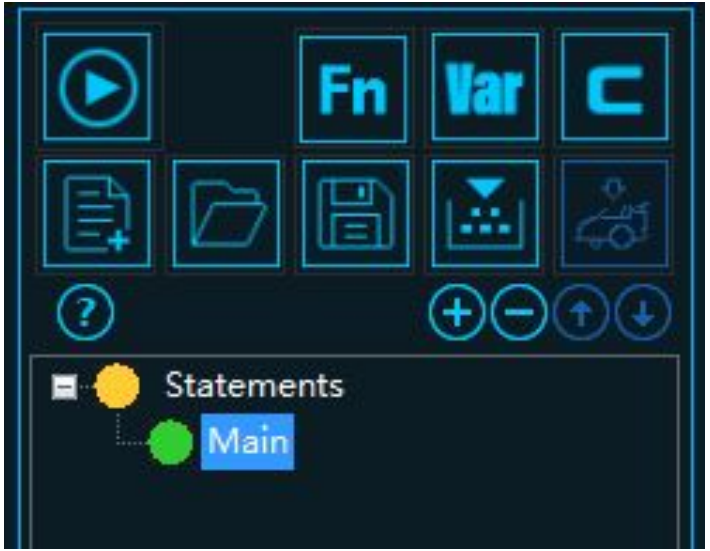


Basic Setup for Full C Programming

A POSTERIORI
Play · Experience · Learn

With the Compiler option greyed out, I recommend to use the following approach to programming in C within the GUI:

1) Open a new AI project and create a single statement



Helper Variables

2) Open the Variable (Var) window and create 3 new variables to mirror the existing actuation variables:

- led (for LED_1)
- wl (for WheelLeft)
- wr (for WheelRight)

C Coding

3) Focus on the one statement you created and open Advanced **Action** window. All your C code will go here.

Add these lines and keep them at the bottom of your action code block:

```
LED_1 = led;
WheelLeft = wl;
WheelRight = wr;
```

C Coding

A POSTERIORI
Play · Experience · Learn

4) The rest of your code can be placed below, mostly as a long if/else conditional statement.

There is no way to create meaningful functions, but you can create variables like **isLeftBrightRed**, etc, as shown.

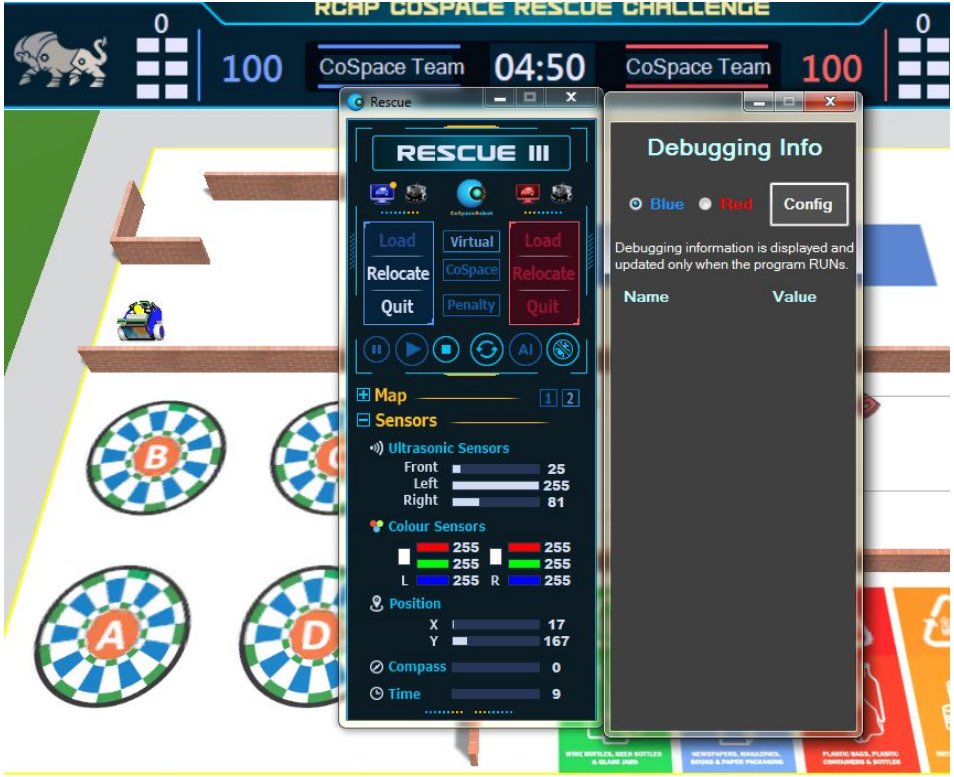
You should add a Duration condition as shown, if you want to use Duration in your code (sleep).

```
int isLeftBrightRed = CSLeft_B < 10 && CSLeft_G < 10 &&
                      CSLeft_R > 200;

if (Duration > 0) {
  // Do Nothing
}
else if (TaskList[0].TaskID == 0) {
 if (Time < 85) {
  printf("right\n");
  wl = 100;
  wr = 55;
 }
 else {
  printf("left\n");
  wl = 35;
  wr = 100;
 }
}
else {
  printf("fwd\n");
  wl = 100;
  wr = 100;
  led = 0;
}

LED_1 = led;
WheelLeft = wl;
WheelRight = wr;
```

DEBUGGING

Inevitably, you will face problems with your car's behaviors.

One good way to debug its behavior is to look at the various sensor and actuator variables during a run.

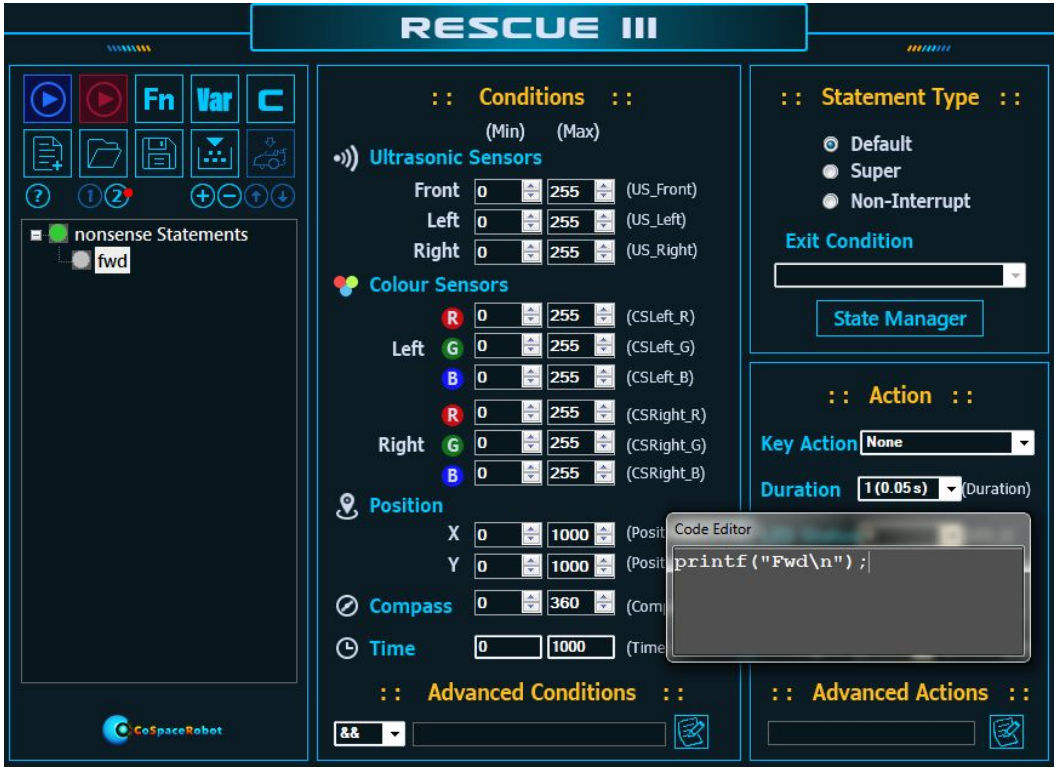


DEBUGGING

If you are using traditional GUI statements, you can add **printf()** into every **Advanced Action**, as shown.

\nis the character for NewLine. If you don't add that your next printf() wi ll be printed just after the last printed character.

The text goes into **AI window.**



DEBUGGING

Another way to debug your AI code is to insert **printf()** statements into every conditional path, as shown.

\n is the character for NewLine. If you don't add that your next printf() will be printed just after the last printed character.

The text goes into **AI window.**

```
if (Duration > 0) {
  // do nothing
}
else if (task == 1 && isOrange) {
  printf("Orange, so stop\n");
  wl = 0;
  wr = 0;
  led = 1;
  Duration = 80;
}
else if (task == 2 && isOrange && waypoint < 3) {
  printf("Orange, go slow`\n");
  wl = 10;
  wr = 10;
}
else if (task == 2 && isOrange) {
  printf("Orange, so stop\n");
  wl = 0;
  wr = 0;
  led = 1;
  Duration = 80;
}
else if (task == 2 && state == 1 && RotationZ > 135) {
  printf("first green, keep turning\n");
  wl = 30;
  wr = 10;
}
else if (task == 2 && state == 1 && RotationZ < 135) {
  printf("first green, finish turn\n");
```

DEBUGGING

A POSTERIORI
Play · Experience · Learn

If you wish to print variable values you will need to add placeholders.
Lookup printf() function in C reference.

%d is for numerical values.

For instance:

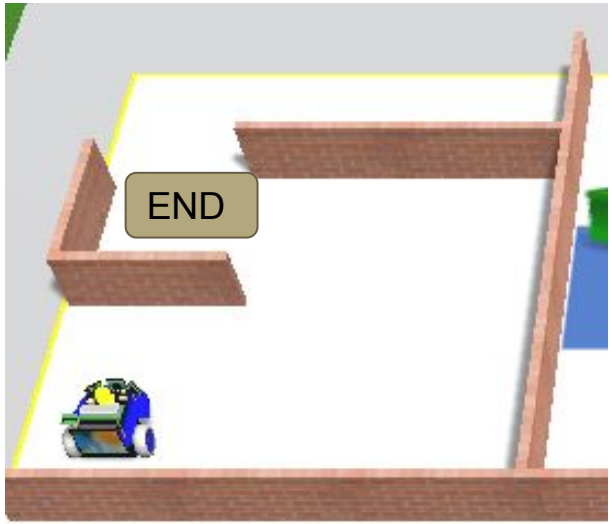
```
printf("Compass = %d\n", Compass);

or

Printf("X = %d, Y = %d\n", PositionX, PositionY);
```

Task 1

Get robot to the END.

Strategies:

- Wall-Following
- Time-based Turns (Right, Left, Left)
- Time or State-based Dead Reckoning



Task 2

A POSTERIORI
Play · Experience · Learn

Recognize colors, drive & flash accordingly.

Utilise:

- Color Sensor
- LED

Task 2 – Color Detection

U19

Task:

Program your robot to recognize different recycling stations.

Orange (Metal) – LED Flash, Speed 10
Red (Plastic) – LED On, Speed 20
Blue (Paper) – LED Flash, Speed 30
Green (Glass) – LED Flash, Stop, 3 seconds

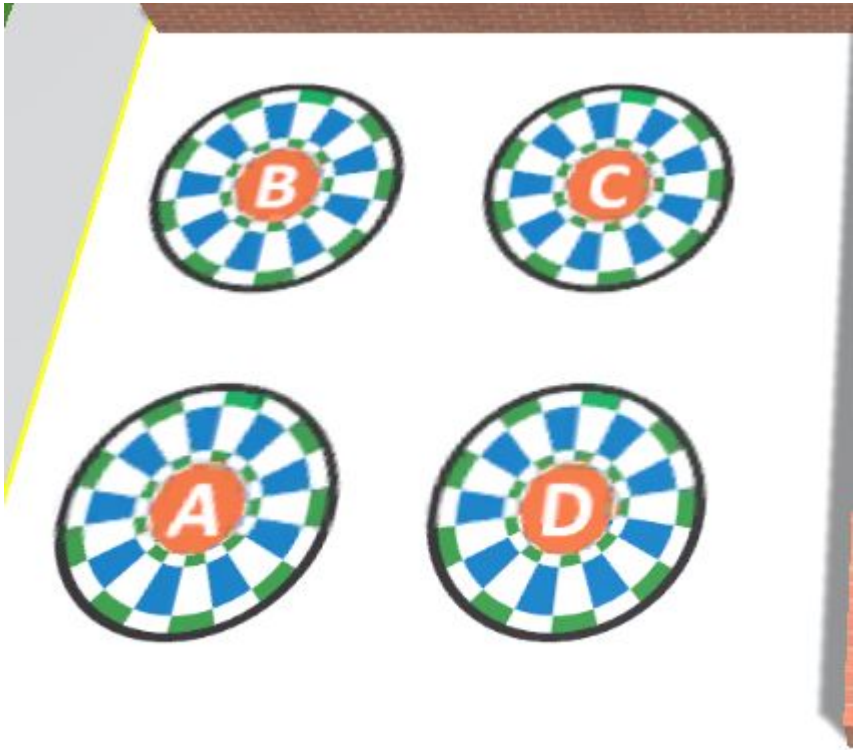
Task 3

Go To: A->B->C->D

Flash LED at each Point

Utilise:

- GPS
- Compass/TurnTo
- State Variable?



A POSTERIORI
Play · Experience · Learn

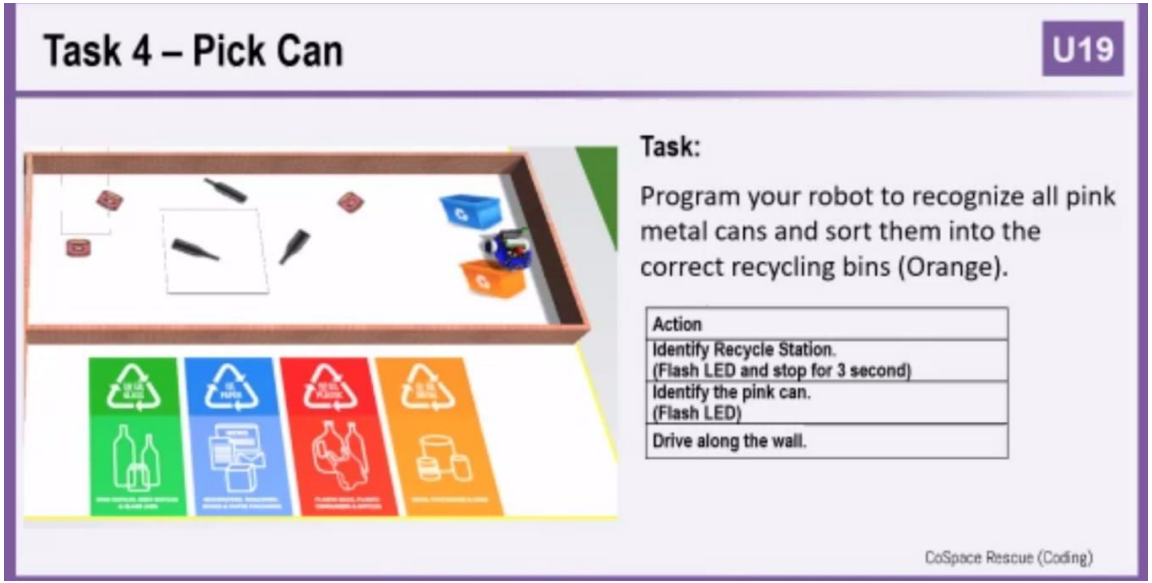
Task 4

CoSpace Rescue Lite

Utilise:

- Pickup Pink Cans
- Deposit in Orange Bin

NOTE: Says only flash LED to pickup...



Task 5

A POSTERIORI
Play · Experience · Learn

Special Zone Algorithm

Utilise:

- Color Sensor & State Variable to stay in Blue Area
- This time pick up Black objects...