mBot Default Program Part 3 - Line Following

Time required: 90 minutes
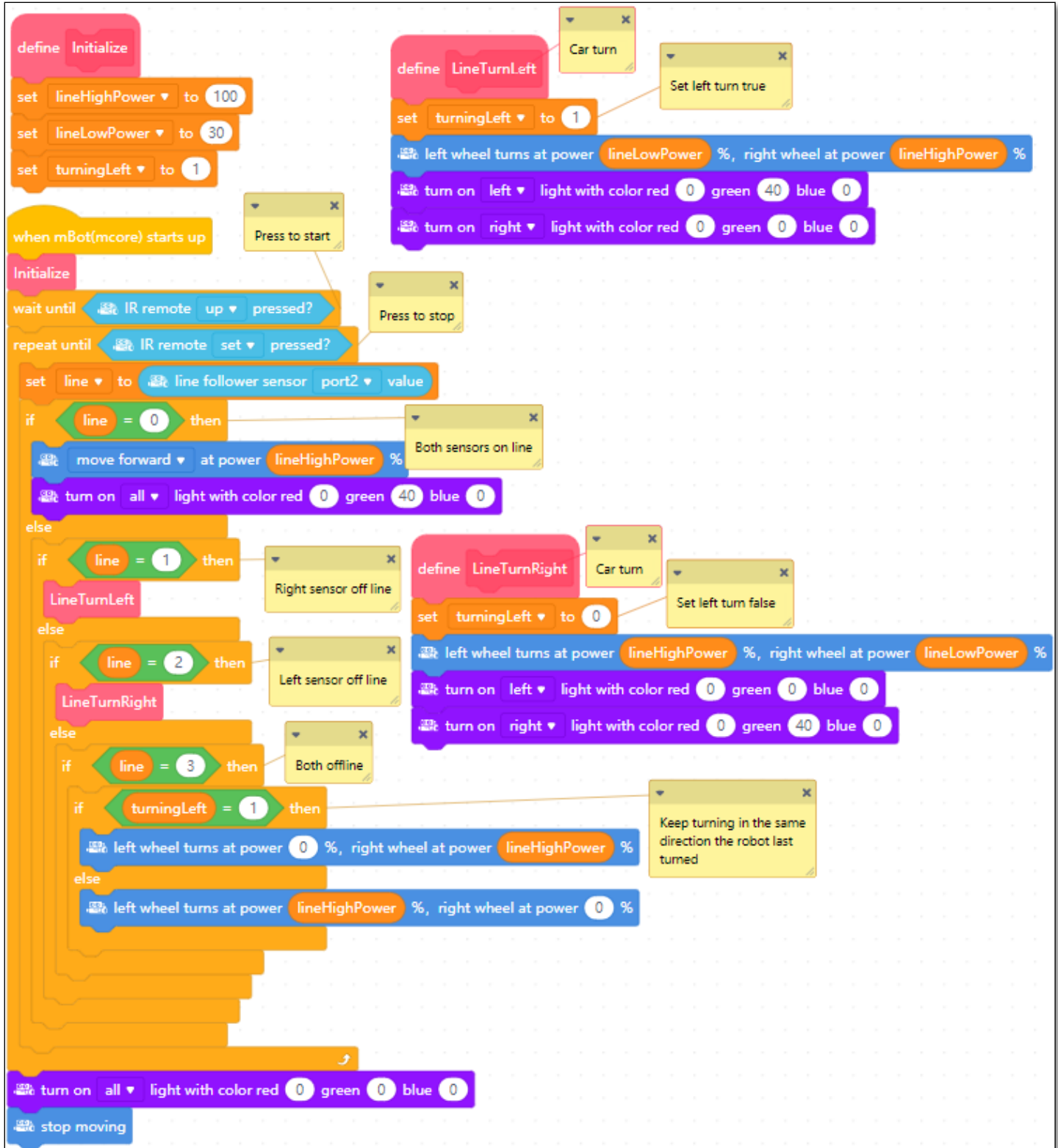
This program adds line following features to our default program.

Tutorial Assignment

1. Start mBlock. Open **mBot Default Program Part 2 Obstacle Avoidance**.

2. Save as **mBot Default Program Part 3 Line Following**.

3. Complete and test the program with the requirements listed.

Requirements

- Integrate the Smart Line Following assignment with your current mBot Default Program.

    - From the Smart Line Following program: remove the Initialize block, wait until, repeat until, and run forward at speed 0.

- Create separate turn functions for the line following. Experiment with the turns to speed up and smooth out the line following. Use factors to change leftPower and rightPower. By doing this, you can press A, use the Set_Speed block to change the speed, then press C and the Line Following will work at the new speed.

- Button A starts remote control mode. The modeFlag = 0

- Button B starts obstacle avoidance mode. The modeFlag = 1

- Button C starts line following mode. The modeFlag = 2

- Create and test the program.

```
define Initialize
set lineHighPower ▾ to 100
set lineLowPower ▾ to 30
set turningLeft ▾ to 1

when mBot(mcore) starts up          [Press to start]
Initialize
wait until < IR remote up ▾ pressed? >
repeat until < IR remote set ▾ pressed? >          [Press to stop]
    set line ▾ to ( line follower sensor port2 ▾ value )
    if < line = 0 > then          [Both sensors on line]
        move forward ▾ at power (lineHighPower) %
        turn on all ▾ light with color red (0) green (40) blue (0)
    else
        if < line = 1 > then          [Right sensor off line]
            LineTurnLeft
        else
            if < line = 2 > then          [Left sensor off line]
                LineTurnRight
            else
                if < line = 3 > then          [Both offline]
                    if < turningLeft = 1 > then          [Keep turning in the same direction the robot last turned]
                        left wheel turns at power (0) %, right wheel at power (lineHighPower) %
                    else
                        left wheel turns at power (lineHighPower) %, right wheel at power (0) %

turn on all ▾ light with color red (0) green (0) blue (0)
stop moving
```

```
                                        [Car turn]
define LineTurnLeft                     [Set left turn true]
set turningLeft ▾ to 1
left wheel turns at power (lineLowPower) %, right wheel at power (lineHighPower) %
turn on left ▾ light with color red (0) green (40) blue (0)
turn on right ▾ light with color red (0) green (0) blue (0)
```

```
                                        [Car turn]
define LineTurnRight                    [Set left turn false]
set turningLeft ▾ to 0
left wheel turns at power (lineHighPower) %, right wheel at power (lineLowPower) %
turn on left ▾ light with color red (0) green (0) blue (0)
turn on right ▾ light with color red (0) green (40) blue (0)
```

Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.

- **In class assignment submission** → Demonstrate in person.

- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.