Driving School

Time required: 120 minutes

NOTE: The mBot is not an accurate robot. As the batteries discharge and the conditions change, it will behave differently. We can change power and time. Just try to get close.

We can accurately move and turn. We will combine the movement and turning programs into one. There isn't an example program, it is up to you to figure it out.

Charge your batteries. Calibrate your robot.

It is time to put your mBot through its paces. Can you pass the driving tests?

Assignment

- Complete each shape in a separate code block.

- Assign each block/shape to a different remote-control button. Use Simple Remote Control as an example.
  Hint: Use if else to stack your remote-control keys.

- Open **Calibrate Distance and Square** and save the program as **Driving School**.

Requirements

1. **Square** - your robot will trace the path of a square that is 1-foot square. It will start and end in the same place and the same orientation.

2. **Rectangle** - your robot will trace the path of a rectangle that is 1-foot x 2-foot. It will start and end in the same place and the same orientation.

3. **Sentry** - your robot will trace a 1-foot square around an object. Start the square one way, then turn around and go back the other way. Return to the beginning point and orientation.
   Sentry Demo Video

4. **Retrace** - move in a 1-foot square forward, and then move in reverse to retrace that same square backwards to the beginning point and orientation. One solution would be to build a Reverse block that uses negative numbers for motor movement.
   Retrace Demo Video

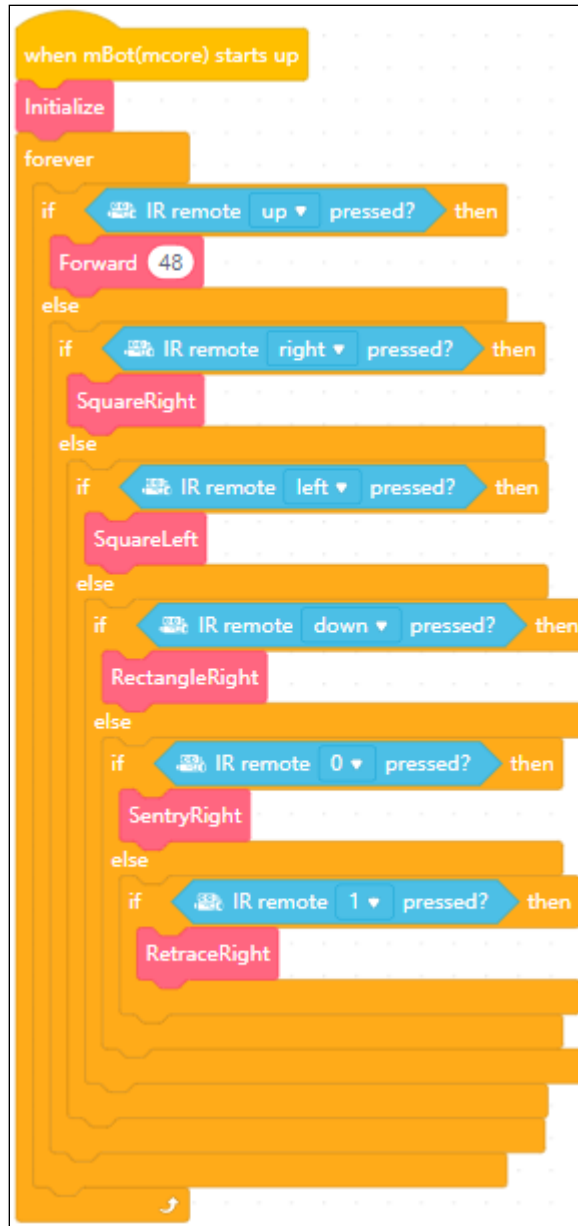It is a good idea to sketch out on a piece of paper the turns and straight traveling you have to do.
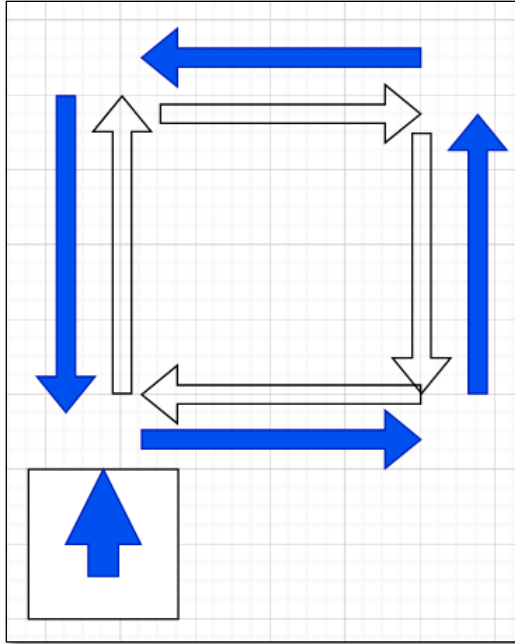
Pseudocode Example Square

```
block loop 4 times
    forward 100
    right 90
```

This is what the main part of your program should look like. Each shape will have its own separate block. Try to build from the parts you already have to build the separate shape blocks.

```
when mBot(mcore) starts up
Initialize
forever
    if  [IR remote up pressed?] then
        Forward 48
    else
        if  [IR remote right pressed?] then
            SquareRight
        else
            if  [IR remote left pressed?] then
                SquareLeft
            else
                if  [IR remote down pressed?] then
                    RectangleRight
                else
                    if  [IR remote 0 pressed?] then
                        SentryRight
                    else
                        if  [IR remote 1 pressed?] then
                            RetraceRight
```

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.