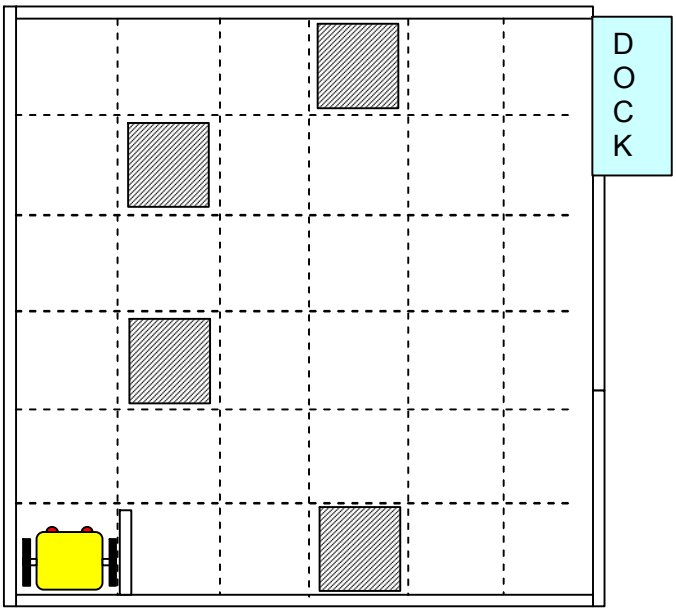
qm queensland museum

Teacher's notes	**Sustainable Futures:** Rover's amazing journey

Challenge

Design and program a lunar rover that is able to avoid obstacles and reach the mother ship docking port.



Toolbox
Procedures:
Program modules that provide instructions that rover will follow.

Solution:
A sequence of procedures that will provide one path to the dock.

Alternative solution:
Suggestions for alternatives that the solver can try.

Getting started

1. **Calibrate lunar rover to move forward:**
   The motion of lunar rover needs to be calibrated so that each forward movement is the same as one unit square on the grid.

   What we want is the command **Fwd1** to move rover forward by one square.

   - Start the program PICAXE Programming Editor then open the file MOTION_MOD.BAS
   - Connect the serial cable between the PC and the rover then switch rover ON.
   - Add the following lines of code into the **Start** routine:

```
'=========================================================
'                  Enter solution gosub procedures here
'=========================================================

Start:  gosub Fwd1              'gosub procedure1
        Wait 2                  'wait time


'=========================================================
```

   - Download the program by clicking on the **Program** button.
   - Remove the serial cable and place rover on the grid pattern.
   - Push the GO button and observe how far rover moves.

Queensland Government

Making adjustments:
To make adjustments to rover's forward motion try the following:

- If rover has travelled **too far** then it will need to be slowed down.
  To slow down rover, go to symbol SPEED in the **init** procedure

  ```
  symbol SPEED = 250
  ```

  Reduce 250 to say 200, then download the program and test again.
  Try different values of SPEED until the desired distance is achieved.

- If rover travelled **less** than one unit square then it will necessary to **increase** the value of SPEED.
  Try different values until rover moves forward by exactly one square unit.

.

2. Calibrate lunar rover to turn right and left:
The motion of lunar rover needs to be calibrated so that each turn movement will be as close as possible to 90º.

To test this connect lunar rover to the PC with the serial cable and turn rover ON.

- Add the following lines of code into the **Start** routine:

```
'============================================================
'                    Enter solution gosub procedures here
'============================================================

Start:  gosub Fwd2              'gosub procedure1
        Wait 2                  'wait time
        gosub Right

'============================================================
```

- Download the program by clicking on the **Program** button.
- Remove the serial cable and place rover on the grid pattern.
- Push the GO button and observe how far rover turns.

Making adjustments:
To make adjustments to rover's turning motion try the following:

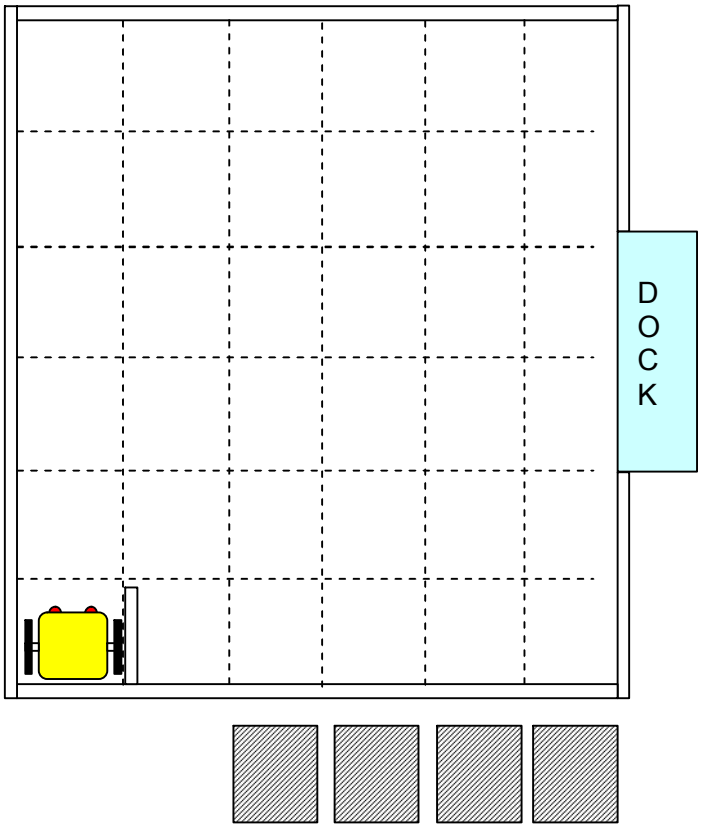
- If rover has turned **too far** then it will need to be slowed down.
  To slow down rover, go to the procedure named **init** and look for the symbol TURN command:

  ```
  symbol TURN = 250
  ```

  Reduce 250 to say 200, then download the program and test again.
  Try different values of TURN until the desired turn is achieved.

- If rover turned **less** than 90º then it will be necessary to **increase** the value of TURN.
  Try different values until rover turns by exactly 90º.
- Check this adjustment works for turning left.

Challenge #1: First steps



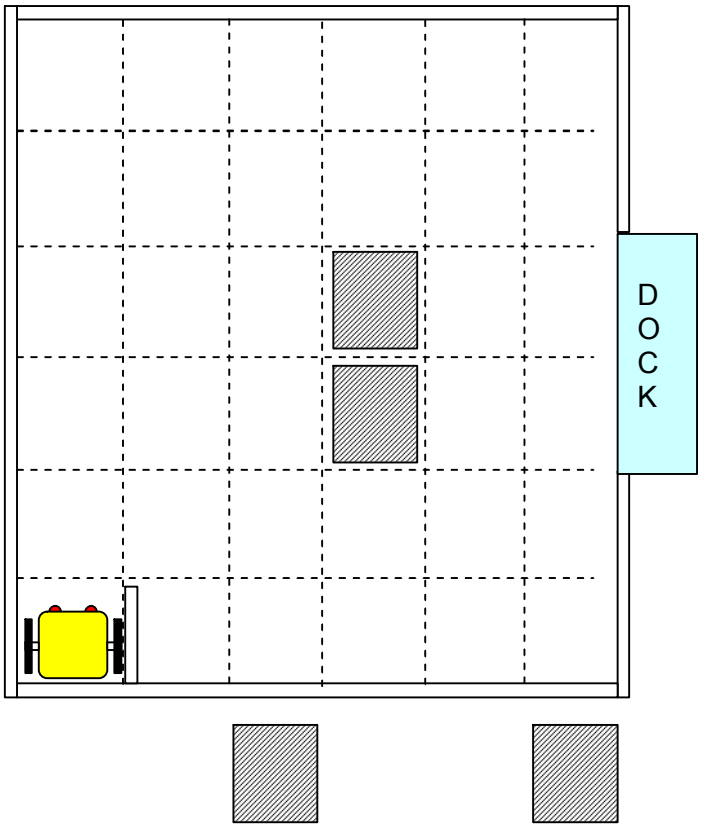
Toolbox:
Procedures:
- Fwd(1-6)
- Right

Solution: (3 moves)
gosub Fwd3
gosub Right
Wait 1

gosub Fwd6
Wait 1
gosub SpinAC

Alternative solution: (? moves)

Challenge #2: Next steps



Toolbox
Procedures:
- Fwd(1-6)
- Right
- Left

Solution: (7 moves)
gosub Fwd4
gosub Right
Wait 1

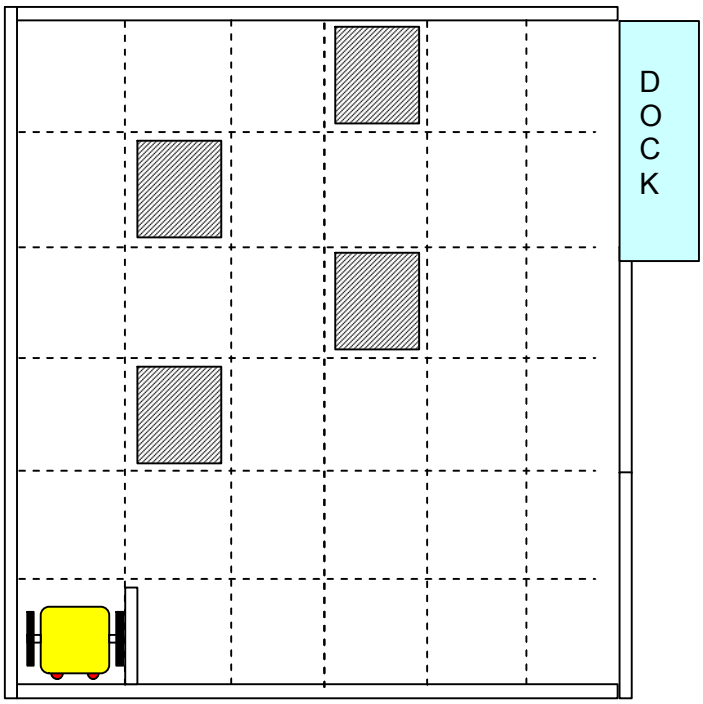
gosub Fwd5
gosub Right
Wait 1

gosub Fwd1
gosub Left
Wait 1

gosub Fwd2
Wait 1
gosub SpinAC

Alternative solution: (? moves)

Challenge #3: Big steps

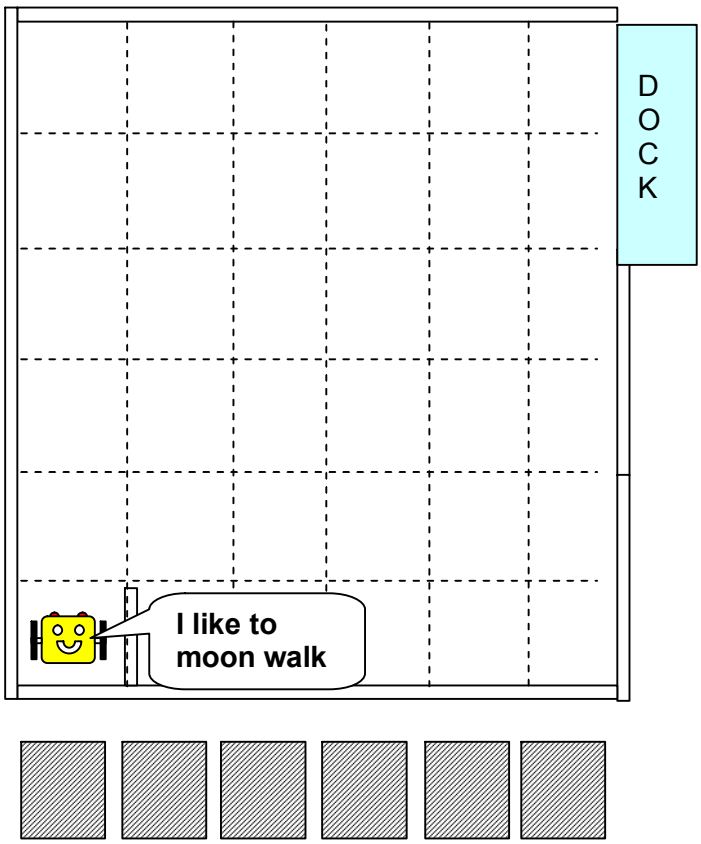


Procedures:
- Fwd(1-6)
- Right
- Left
- Rev

Solution: (? moves)

- Include all procedures in solution
- Reduce to smallest number of moves

Challenge #4: Set your own



Design your own challenge by placing the blocks in any of the grid squares.

Procedures:

Use any available

Solution:

Use smallest number of moves
Modify a procedure to make it more efficient

Extra challenge:
Design a routine so that your lunar rover can negotiate any arrangement of obstacles.

Hint: Use the distance detector on pin7:

```
if pin7 = 1 then Avoid
   gosub Fwd
```

Rover commands

Summary of commands available for use from the BASIC code included in MOTION_MOD.BAS:

Want	How to send command	What I do
Forward	gosub FWD1 gosub FWD2 gosub FWD3 gosub FWD4 gosub FWD5 gosub FWD6	Move forward the number of unit squares
Back	gosub Rev	Reverse or go backwards
Turn	gosub Right gosub Left	right turn left turn
Turn forward or backward	gosub FwdR gosub FwdL gosub RevR gosub RevL	forward right forward left reverse right reverse left
Rest time	Wait 1 Wait 10 Wait 0.5	Wait 1 second Wait 10 seconds Wait 0.5 seconds
Spin	gosub SpinC gosub SpinCRand gosub SpinAC	spin clockwise spin random clockwise spin anticlockwise
Stop	gosub StopAll	stop everything
Slow	gosub FwdSlow gosub RevSlow	forward slow at half speed reverse slow at half speed

Additional commands can easily be added by creating new sub procedures such as Rev2, etc. eg

```
'============================================================
'                    Reverse 2 units straight at SPEED
'============================================================

Rev2:
        for b0 = 1 to 2              'Repeat 2 times
            gosub Rev                'Forward 1 unit
        next b0
        return                       'End

'============================================================
```

More sophisticated procedures can be developed that include a combination of simpler procedures. eg. ***Avoid*** would require a sequence of stop, turn or spin, and then go forward again.

New commands

Case: Avoidance

Challenge:
Design a routine so that your lunar rover can negotiate any arrangement of obstacles.

Define:
Specify an acceptable performance standard.

eg. Set a time limit or maximum number of moves to complete the task.

Solution: Use the distance detector on pin7:

```
if pin7 = 1 then Avoid          ' if object is detected then avoid it
        gosub Fwd1              ' otherwise go forward
```

Avoid:
What to do if object detected:

1. Stop
2. Turn either right or left
3. Go forward

Sub procedure to be added:

```
'================================================================
'                          Avoid routine
'================================================================
Avoid:
        gosub StopAll           ' stop
        gosub Right             ' turn right
        gosub Fwd1              ' forward 1 square
        return                  ' go back to call point
'================================================================
```

Test:
Try the new rover commands and note any limitations of the procedure.

Refine:
Make changes to the avoidance procedure based on the test results.
Continue to test and refine the procedure to meet the required performance standard.

Evaluate:
How well was the performance standard met?
Specify another performance standard to take the solution to an improved level.

Basic Code

```
'BASIC motion program modules:
'Motion_mod.bas
'Updated on 16/6/2009
'Allan Morrison

‘============================================================
‘ Program provides a template for basic motion of a robot.
‘ Modules provide the following movements:
‘        Fwd ……. forward in units 1-6
‘        Rev ……. reverses
‘        Right ……. right turn
‘        Left ……. left turn
‘        FwdR ……. forward right
‘        FwdL …….. forward left
‘        RevR …….. reverse right
‘        RevL …….reverse left
‘        SpinC ……. spin clockwise
‘        SpinCRand … spin random clockwise
‘        SpinAC ……...spin anticlockwise
‘        FwdSlow …… forward slow
‘        RevSlow …….reverse slow

'============================================================
'                          Main program control
'============================================================
init:

        symbol SPEED = 250              ‘Sets forward distance unit
        symbol TURN = 300               ‘Sets turn angle
        symbol SPIN = TURN * 4          ‘Sets spin angle

main:

WaitStart:                              'Wait for the start
                if pin6 = 1 then Start  'Push GO button to start
                        goto WaitStart  'Keep waiting
                end

'============================================================
'                  Enter solution gosub procedures here
'============================================================

Start:
                                        ‘Test procedures to be removed
                gosub Fwd2              'gosub procedure1
                Wait 1                  'wait time

                gosub Right             'gosub procedure2
                Wait 1                  'wait time




'============================================================
```

```
                                        '
                gosub StopAll           'Do Procedure STOP all motors
                goto WaitStart          'Do Procedure WaitStart again or
                'goto Start             'Do Procedure Start to continue loop
                end                     'end start

end                                     'end main


'============================================================
'                       Stop all motors
'============================================================

StopAll:
                low 0                   'Motor A Reverse    OFF
                low 1                   'Motor A Forward    OFF
                low 2                   'Motor B Reverse    OFF
                low 3                   'Motor B Forward    OFF
                return                  'End


'============================================================
'                       Forward straight
'============================================================

Fwd:
                low 0                   'Motor A Reverse    OFF
                high 1                  'Motor A Forward    ON
                low 2                   'Motor B Reverse    OFF
                high 3                  'Motor B Forward    ON

                pause SPEED             'Wait SPEED second command

                gosub StopAll           'Stop all motors
                return                  'End


'============================================================
'                       Forward 1 unit straight at SPEED
'============================================================

Fwd1:
                gosub Fwd               'Forward 1 unit
                return                  'End


'============================================================
'                       Forward 2 units straight at SPEED
'============================================================

Fwd2:
                for b0 = 1 to 2                 'Repeat 2 times
                        gosub Fwd               'Forward 1 unit
                next b0
                return                          'End
'============================================================
```

```
'                       Forward 3 units straight at SPEED
'=============================================================

Fwd3:
            for b0 = 1 to 3             'Repeat 3 times
                  gosub Fwd             'Forward 1 unit
            next b0
            return                      'End

'=============================================================
'                       Forward 4 units straight at SPEED
'=============================================================

Fwd4:
            for b0 = 1 to 4             'Repeat 4 times
                  gosub Fwd             'Forward 1 unit
            next b0
            return                      'End

'=============================================================
'                       Forward 5 units straight at SPEED
'=============================================================

Fwd5:
            for b0 = 1 to 5             'Repeat 5 times
                  gosub Fwd             'Forward 1 unit
            next b0
            return                      'End


'=============================================================
'                       Forward 6 units straight at SPEED
'=============================================================

Fwd6:
            for b0 = 1 to 6             'Repeat 6 times
                  gosub Fwd             'Forward 1 unit
            next b0
            return                      'End
```

```
'============================================================
'                        Forward at half SPEED
'============================================================

FwdSlow:

        for b0 = 1 to 50


                low 0           'Motor A Reverse    OFF
                high 1          'Motor A Forward    ON
                low 2           'Motor B Reverse    OFF
                high 3          'Motor B Forward    ON

                pause 5         'Wait 0.01 second command

                gosub StopAll   'Stop all motors

                pause 5         'Wait 0.005 second command

        next b0                 'next loop

        return                  'End


'============================================================
'                        Turn right
'============================================================

Right:
        low 0           'Motor A Reverse    OFF
        high 1          'Motor A Forward    ON
        high 2          'Motor B Reverse    ON
        low 3           'Motor B Forward    OFF

        pause TURN      'Wait TURN second command

        gosub StopAll   'Stop all motors
        return          'End


'============================================================
'                        Turn left
'============================================================

Left:
        high 0          'Motor A Reverse    OFF
        low 1           'Motor A Forward    ON
        low 2           'Motor B Reverse    ON
        high 3          'Motor B Forward    OFF

        pause TURN      'Wait TURN second command

        gosub StopAll   'Stop all motors
        return          'End
```

```
'=============================================================
'                          Spin clockwise
'=============================================================

SpinC:
        low 0               'Motor A Reverse    OFF
        high 1              'Motor A Forward    ON
        high 2              'Motor B Reverse    ON
        low 3               'Motor B Forward    OFF

        pause SPIN          'Wait 1 second command

        gosub StopAll       'Stop all motors
        return              'End


'=============================================================
'                          Spin clockwise random interval
'=============================================================

SpinCRand:

        random w0           'random word b0,b1

        low 0               'Motor A Reverse    OFF
        high 1              'Motor A Forward    ON
        high 2              'Motor B Reverse    ON
        low 3               'Motor B Forward    OFF

        pause b1            'Wait random time command up to 0.25 second

        gosub StopAll       'Stop all motors
        return              'End


'=============================================================
'                          Spin anti-clockwise
'=============================================================

SpinAC:
        high 0              'Motor A Reverse    ON
        low 1               'Motor A Forward    OFF
        low 2               'Motor B Reverse    OFF
        high 3              'Motor B Forward    ON

        pause SPIN          'Wait 1 second command

        gosub StopAll       'Stop all motors
        return              'End
```

```
'=========================================================
'                        Reverse to the left
'=========================================================

RevL:
        low 0           'Motor A Reverse    OFF
        low 1           'Motor A Forward    OFF
        high 2          'Motor B Reverse    ON
        low 3           'Motor B Forward    OFF

        pause SPEED     'Wait 1 second command

        gosub StopAll   'Stop all motors
        return          'End



'=========================================================
'                        Reverse to the right
'=========================================================

RevR:
        high 0          'Motor A Reverse    ON
        low 1           'Motor A Forward    OFF
        low 2           'Motor B Reverse    OFF
        low 3           'Motor B Forward    OFF

        pause SPEED     'Wait SPEED second command

        gosub StopAll   'Stop all motors
        return          'End



'=========================================================
'                        Reverse straight back
'=========================================================

Rev:
        high 0          'Motor A Reverse    ON
        low 1           'Motor A Forward    OFF
        high 2          'Motor B Reverse    ON
        low 3           'Motor B Forward    OFF

        pause SPEED     'Wait 1 second command

        gosub StopAll   'Stop all motors
        return          End
```

```
'============================================================
'                        Reverse at half SPEED
'============================================================

RevSlow:

        for b0 = 1 to 50            'Repeat 100 times 2 x .01 seconds


            high 0                  'Motor A Reverse    ON
            low 1                   'Motor A Forward    OFF
            high 2                  'Motor B Reverse    ON
            low 3                   'Motor B Forward    OFF

            pause 5                 'Wait 0.01 second command

            gosub StopAll           'Stop all motors
            pause 5                 'Wait 0.01 second command

        next b0                     'next loop

        return                      'End

'============================================================
'                        Forward to left
'============================================================

FwdL:
        low 0               'Motor A Reverse    OFF
        low 1               'Motor A Forward    OFF
        low 2               'Motor B Reverse    OFF
        high 3              'Motor B Forward    ON

        pause SPEED         'Wait 1 second command

        gosub StopAll       Stop all motors
        return              'End

'============================================================
'                        Forward to right
'============================================================

FwdR:
        low 0               'Motor A Reverse    OFF
        high 1              'Motor A Forward    ON
        low 2               'Motor B Reverse    OFF
        low 3               'Motor B Forward    OFF

        pause SPEED         'Wait 1 second command

        gosub StopAll       'Stop all motors
        return              'End


'============================================================
```