Upper Key Stage 2
Program Solutions Table

UKS2-S1

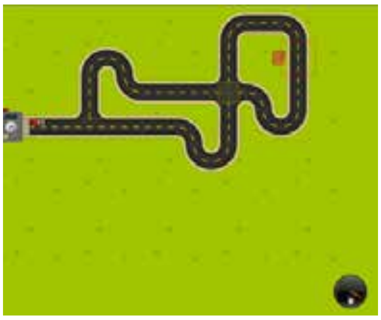
What do we already know? (recap the Blockly commands previously encountered)

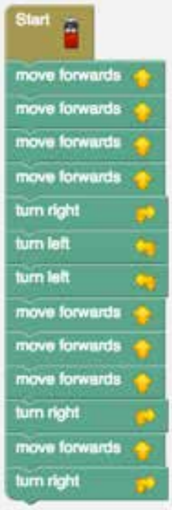
Objectives

- Use the core programming commands appropriately in a visual language
- Understand the **repeat while** command

Limited Blocks

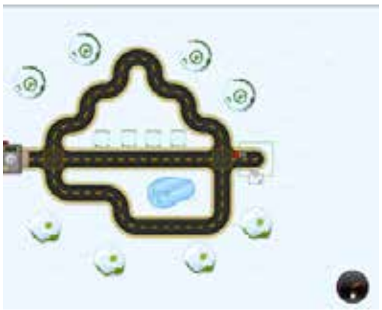
Level 51

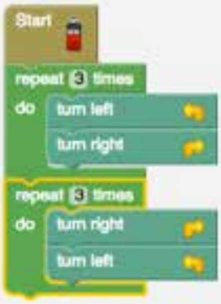




```
Start
move forwards
move forwards
move forwards
move forwards
turn right
turn left
turn left
move forwards
move forwards
move forwards
turn right
move forwards
turn right
```

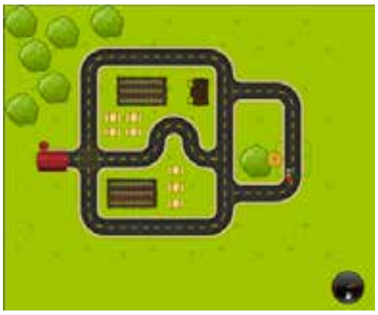
Level 52

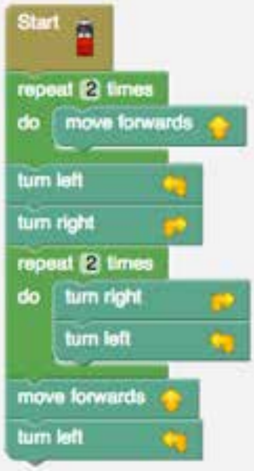




```
Start
repeat 3 times
do  turn left
    turn right
repeat 3 times
do  turn right
    turn left
```

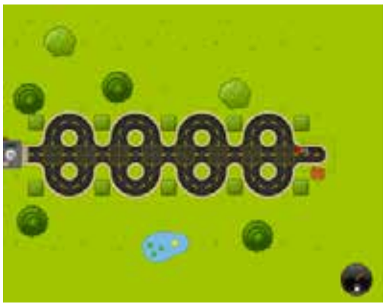
Level 53

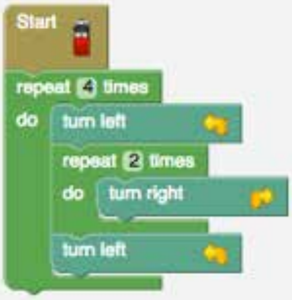




```
Start
repeat 2 times
do  move forwards
turn left
turn right
repeat 2 times
do  turn right
    turn left
move forwards
turn left
```

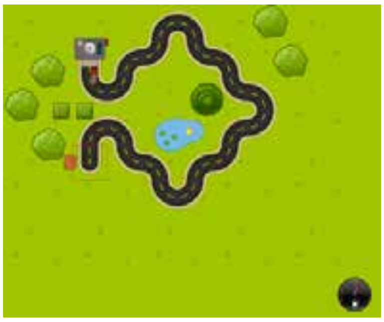
Level 54

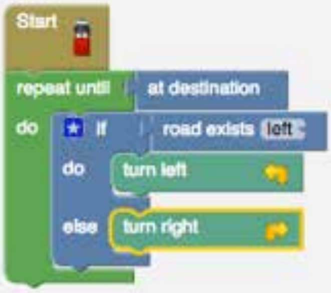




```
Start
repeat 4 times
do  turn left
    repeat 2 times
    do  turn right
    turn left
```

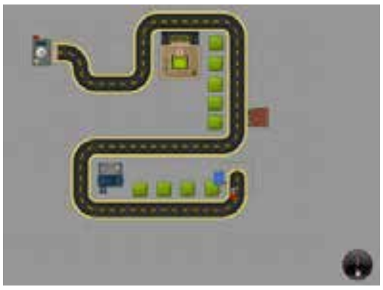
Level 55

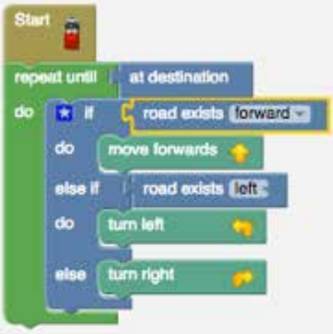




```
Start
repeat until  at destination
do  if  road exists left
    do  turn left
    else  turn right
```

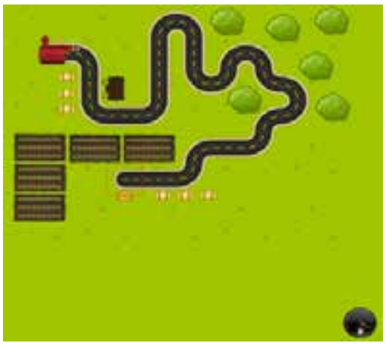
Level 56

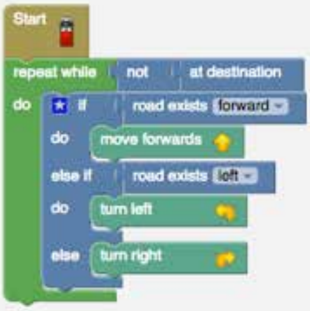




```
Start
repeat until  at destination
do  if  road exists forward
    do  move forwards
    else if  road exists left
    do  turn left
    else  turn right
```

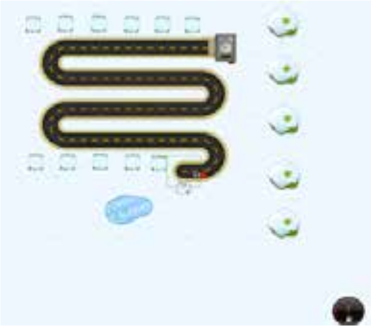
Level 57

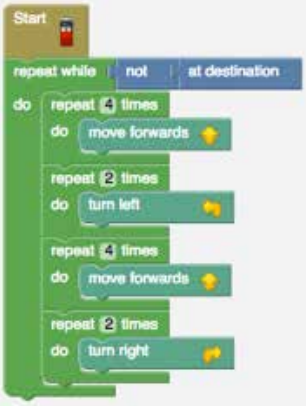




```
Start
repeat while not at destination
do  if road exists forward
    do   move forwards
    else if road exists left
    do   turn left
    else turn right
```

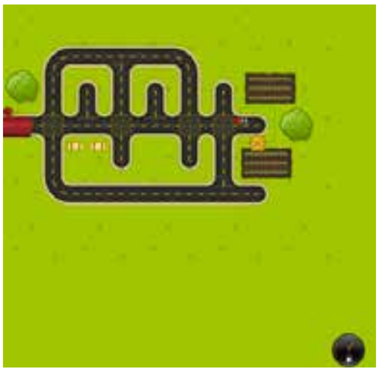
Level 58

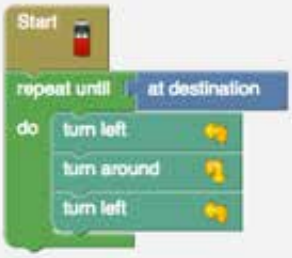




```
Start
repeat while not at destination
do  repeat 4 times
    do  move forwards
    repeat 2 times
    do  turn left
    repeat 4 times
    do  move forwards
    repeat 2 times
    do  turn right
```

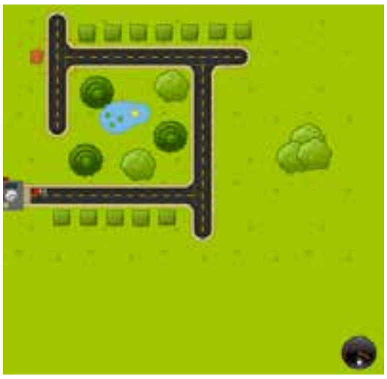
Level 59

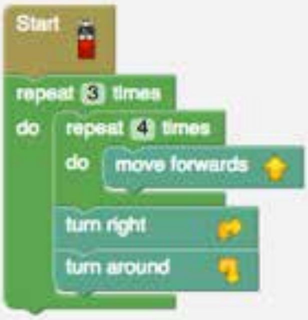




```
Start
repeat until at destination
do  turn left
    turn around
    turn left
```

Level 60





```
Start
repeat 3 times
do  repeat 4 times
    do  move forwards
    turn right
    turn around
```

UKS2-S2

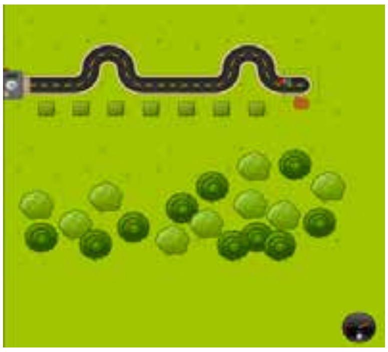
Breaking down the problem into chunks (understanding procedures)

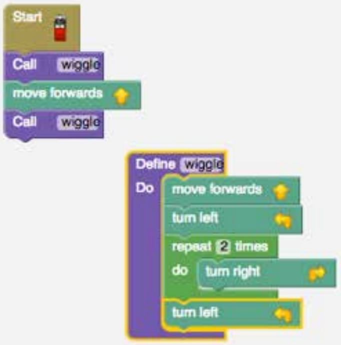
Objectives

- Decompose the programming task into smaller parts
- Identify sections of code which can be used several times and write a procedure for that section
- Use repeat loops within procedures

Procedures

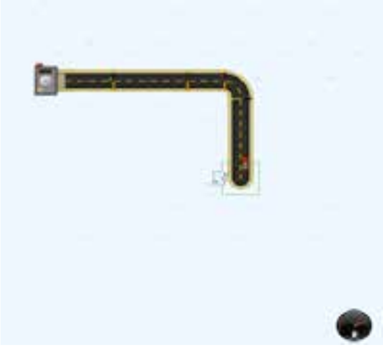
Level 61

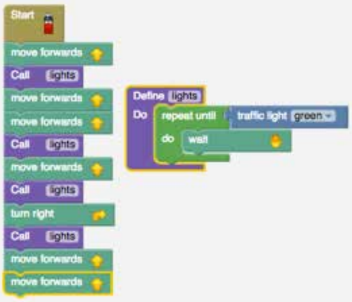




```
Start
Call wiggle
move forwards
Call wiggle

Define wiggle
Do  move forwards
    turn left
    repeat 2 times
    do  turn right
    turn left
```

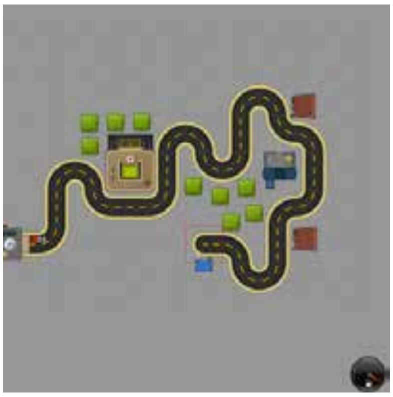
Level 62

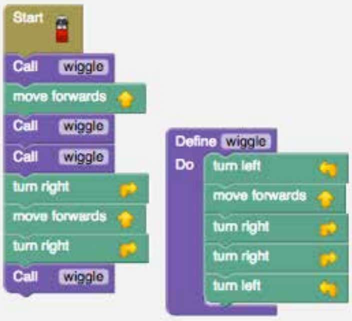




```
Start
move forwards
Call lights
move forwards
move forwards
Call lights
move forwards
Call lights
turn right
Call lights
move forwards
move forwards

Define lights
Do  repeat until traffic light green
    do  wait
```

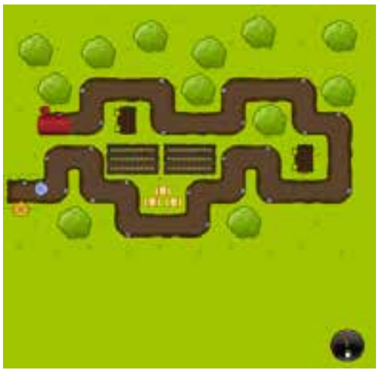
Level 63

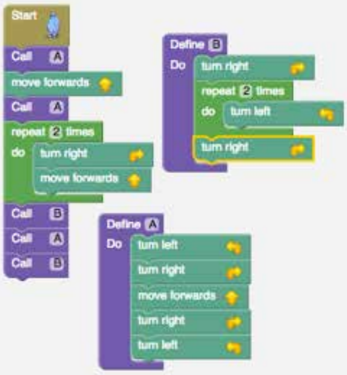




```
Start
Call wiggle
move forwards
Call wiggle
Call wiggle
turn right
move forwards
turn right
Call wiggle

Define wiggle
Do  turn left
    move forwards
    turn right
    turn right
    turn left
```

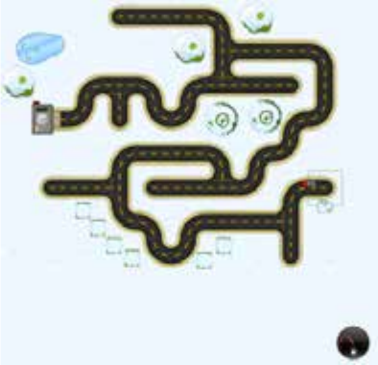
Level 64

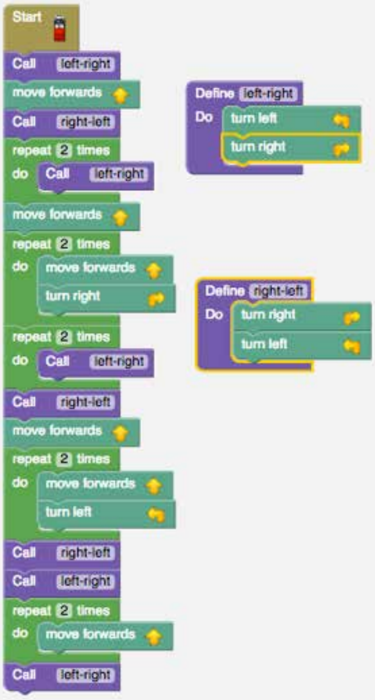




```
Start
Call A
move forwards
Call A
repeat 2 times
do  turn right
    move forwards
Call B
Call A
Call B

Define B
Do  turn right
    repeat 2 times
    do  turn left
    turn right

Define A
Do  turn left
    turn right
    move forwards
    turn right
    turn left
```

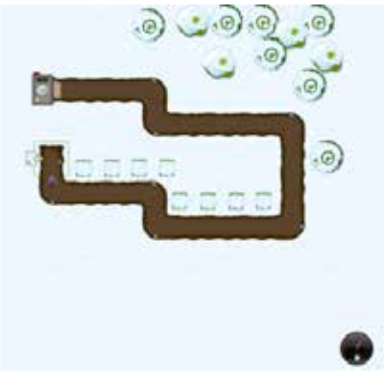
Level 65

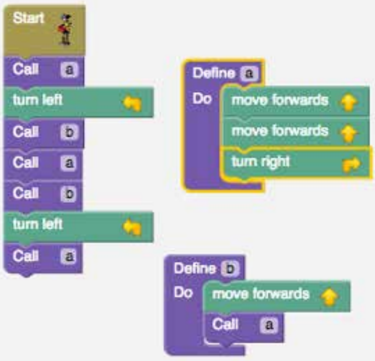




```
Start
Call left-right
move forwards
Call right-left
repeat 2 times
do  Call left-right
move forwards
repeat 2 times
do  move forwards
    turn right
repeat 2 times
do  Call left-right
Call right-left
move forwards
repeat 2 times
do  move forwards
    turn left
Call right-left
Call left-right
repeat 2 times
do  move forwards
Call left-right

Define left-right
Do  turn left
    turn right

Define right-left
Do  turn right
    turn left
```

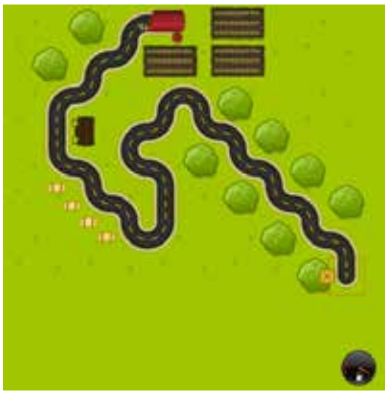
Level 66

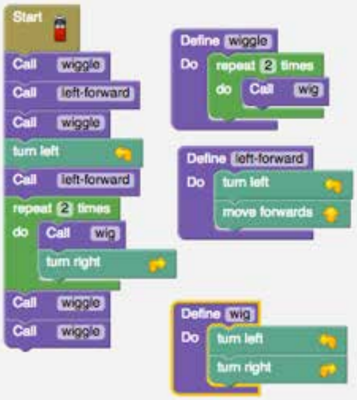




```
Start
Call a
turn left
Call b
Call a
Call b
turn left
Call a

Define a
Do  move forwards
    move forwards
    turn right

Define b
Do  move forwards
    Call a
```

Level 67



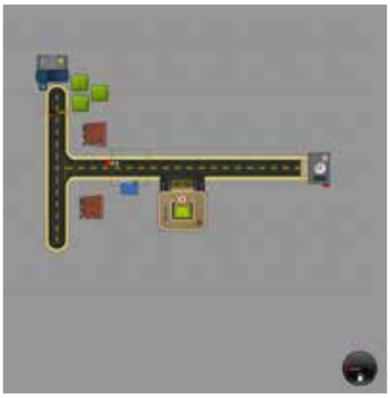


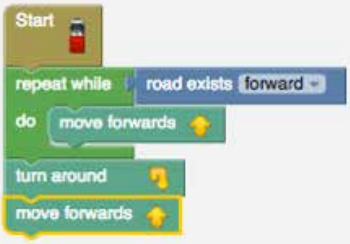
```
Start
Call wiggle
Call left-forward
Call wiggle
turn left
Call left-forward
repeat 2 times
do  Call wig
    turn right
Call wiggle
Call wiggle

Define wiggle
Do  repeat 2 times
    do  Call wig

Define left-forward
Do  turn left
    move forwards

Define wig
Do  turn left
    turn right
```

Blockly Brain Teasers

Note: This section does not have an associated teaching plan, but is a resource to stretch and challenge the more advanced programmers in your class objectives.

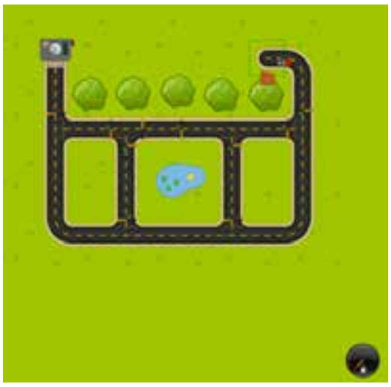
Level 68

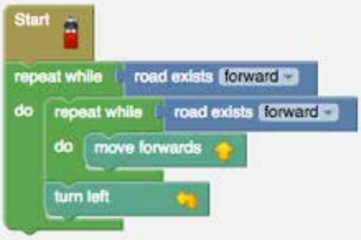




```
Start
repeat while  road exists forward
do  move forwards
turn around
move forwards
```

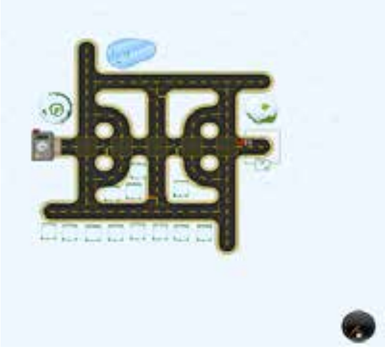
Level 69

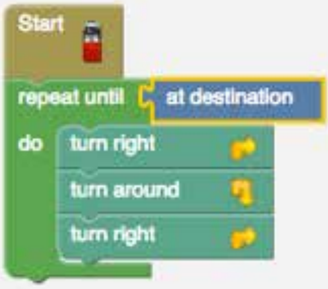




```
Start
repeat while  road exists forward
do  repeat while  road exists forward
    do  move forwards
    turn left
```

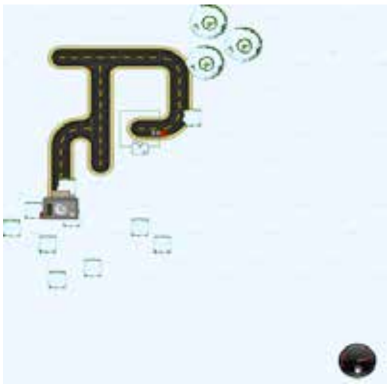
Level 70

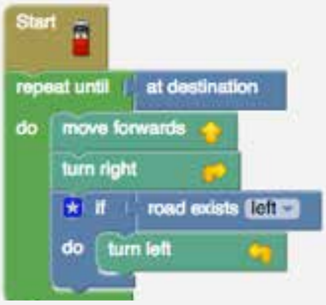




```
Start
repeat until at destination
do  turn right
    turn around
    turn right
```

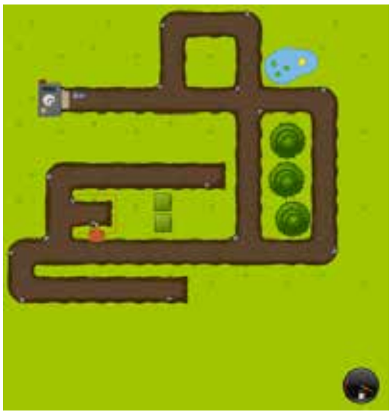
Level 71

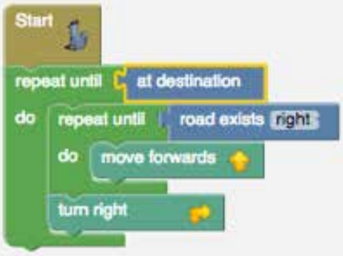




```
Start
repeat until at destination
do  move forwards
    turn right
    if road exists left
    do  turn left
```

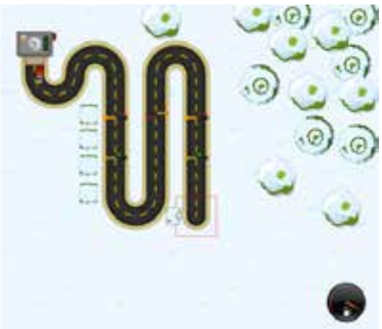
Level 72

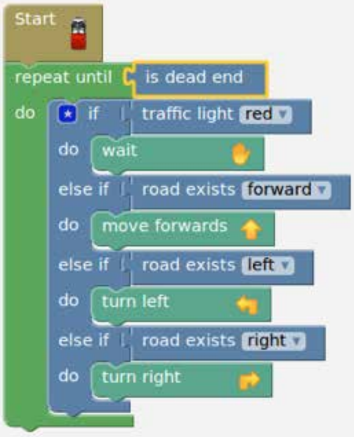




```
Start
repeat until at destination
do  repeat until road exists right
    do  move forwards
    turn right
```

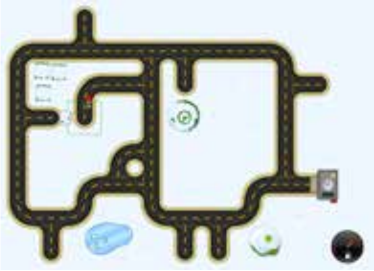
Level 73

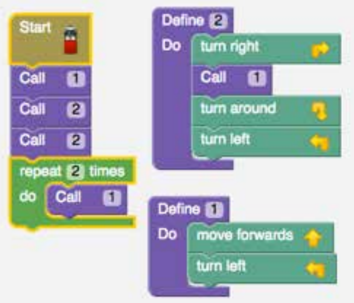




```
Start
repeat until is dead end
do  if traffic light red
    do  wait
    else if road exists forward
    do  move forwards
    else if road exists left
    do  turn left
    else if road exists right
    do  turn right
```

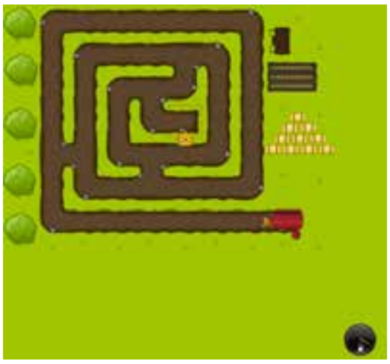
Level 74

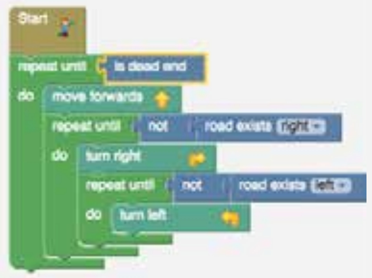




```
Start
Call 1
Call 2
Call 2
repeat 2 times
do  Call 1

Define 2
Do  turn right
    Call 1
    turn around
    turn left

Define 1
Do  move forwards
    turn left
```

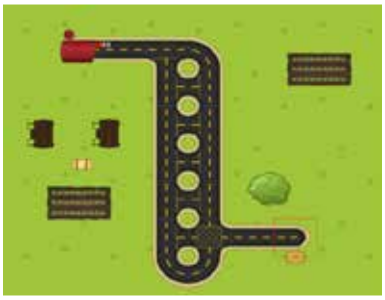
Level 75

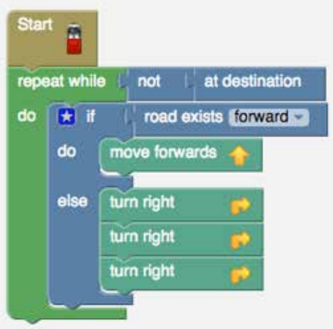




```
Start
repeat until  is dead end
do  move forwards
    repeat until  not  road exists right
    do  turn right
        repeat until  not  road exists left
        do  turn left
```

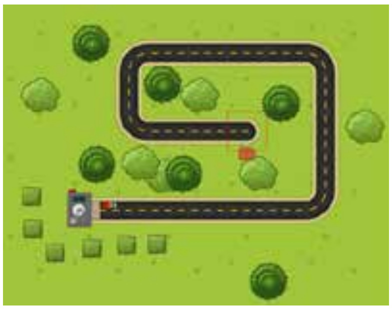
Level 76

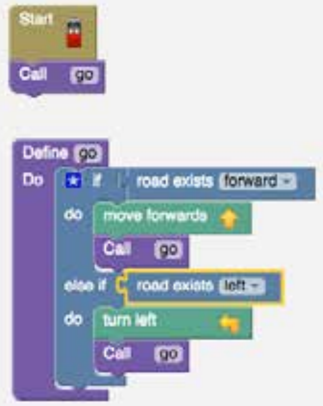




```
Start
repeat while  not  at destination
do  if  road exists forward
    do    move forwards
    else  turn right
          turn right
          turn right
```

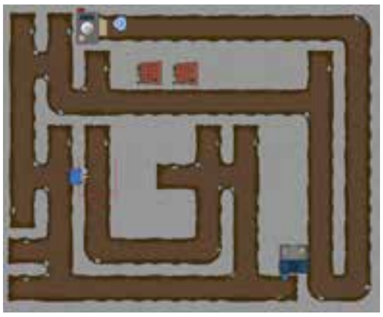
Level 77

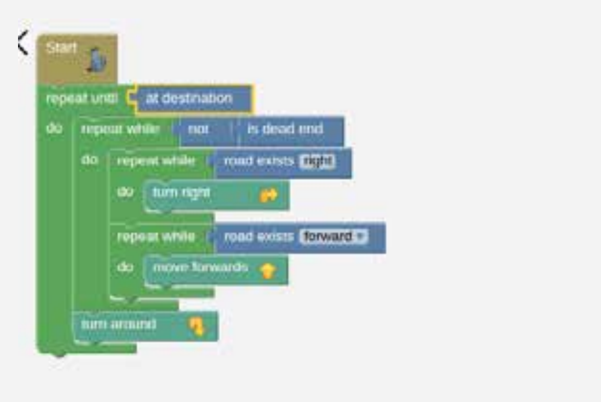




```
Start
Call go

Define go
Do  if  road exists forward
    do  move forwards
        Call go
    else if  road exists left
    do  turn left
        Call go
```

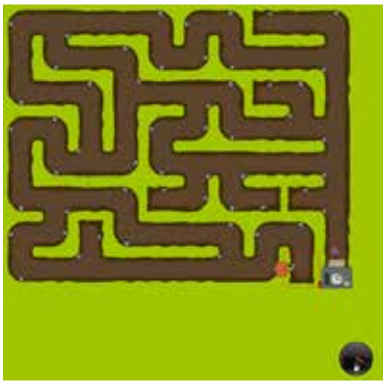
Level 78

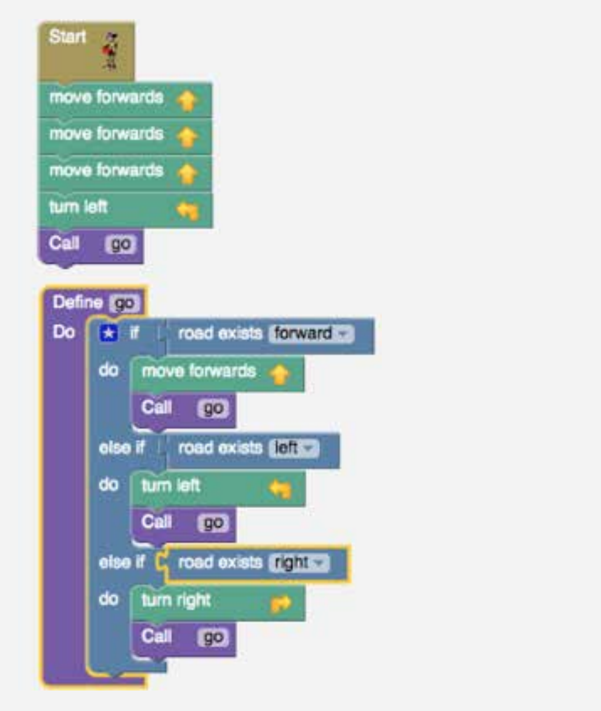




```
Start
repeat until  at destination
do  repeat while  not  is dead end
    do  repeat while  road exists  right
        do  turn right
        repeat while  road exists  forward
        do  move forwards
    turn around
```

Level 79





```
Start
move forwards
move forwards
move forwards
turn left
Call  go

Define  go
Do  if  road exists  forward
    do  move forwards
        Call  go
    else if  road exists  left
    do  turn left
        Call  go
    else if  road exists  right
    do  turn right
        Call  go
```

UKS2-S3

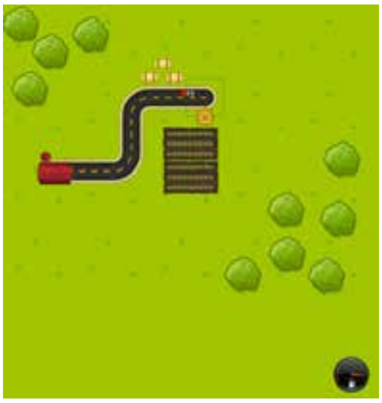
Switching from Blockly to Python (visual to text language)

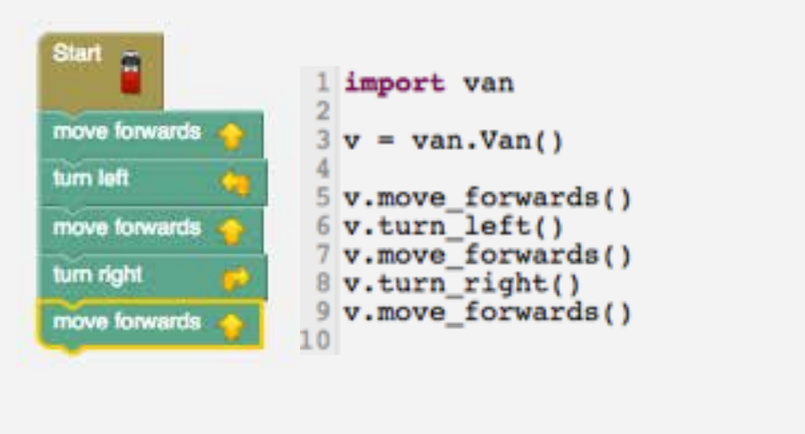
Objectives

- Develop an initial understanding of Python as a text based language
- Understand that Python has precise syntax
- Identify characteristics of Python, compare this with Blockly
- Use and understand the movement instructions in Python code
- Use and understand repeat loops in Python (for count in range (n))

Introduction to Python

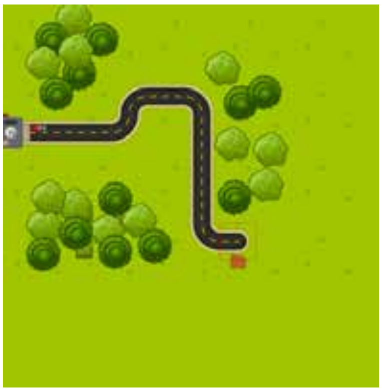
Level 80

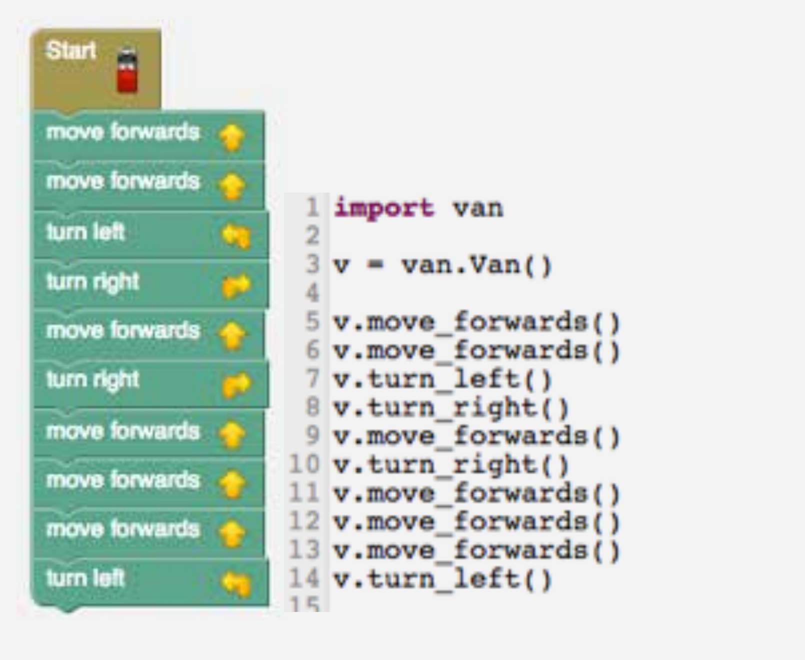




```
import van

v = van.Van()

v.move_forwards()
v.turn_left()
v.move_forwards()
v.turn_right()
v.move_forwards()
```

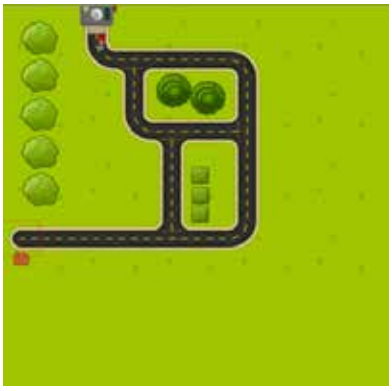
Level 81

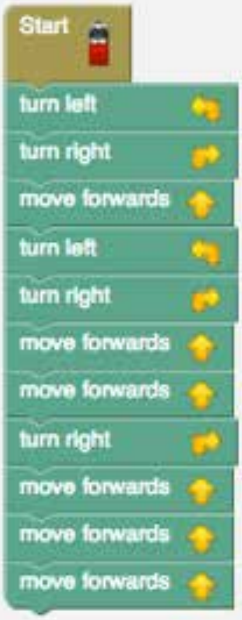




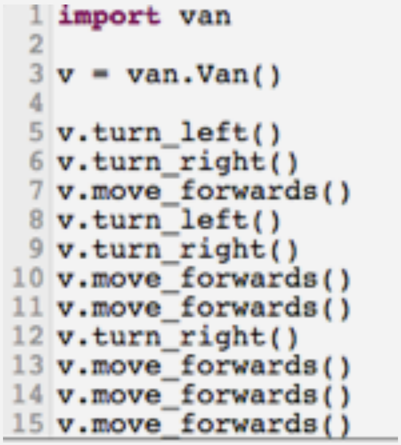
```
import van

v = van.Van()

v.move_forwards()
v.move_forwards()
v.turn_left()
v.turn_right()
v.move_forwards()
v.turn_right()
v.move_forwards()
v.move_forwards()
v.move_forwards()
v.turn_left()
```

Level 82



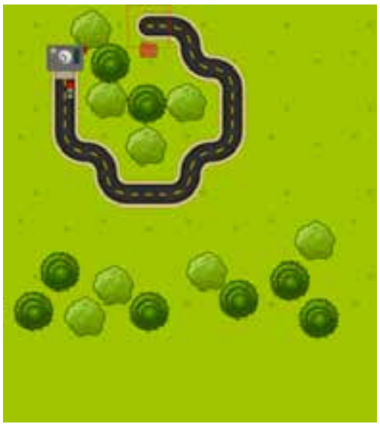


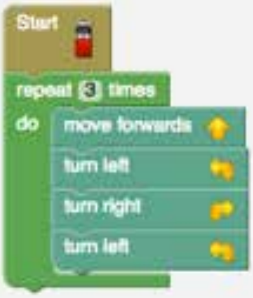
```
Start
turn left
turn right
move forwards
turn left
turn right
move forwards
move forwards
turn right
move forwards
move forwards
move forwards
```



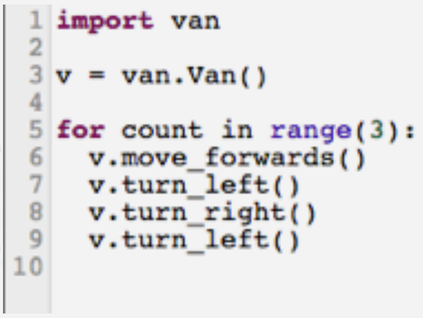
```
import van

v = van.Van()

v.turn_left()
v.turn_right()
v.move_forwards()
v.turn_left()
v.turn_right()
v.move_forwards()
v.move_forwards()
v.turn_right()
v.move_forwards()
v.move_forwards()
v.move_forwards()
```

Level 83



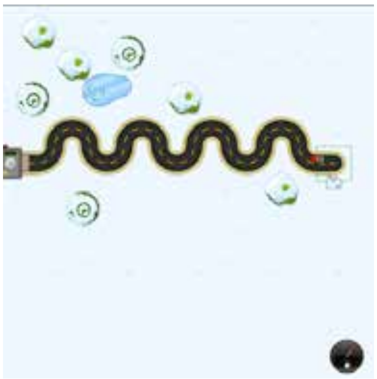


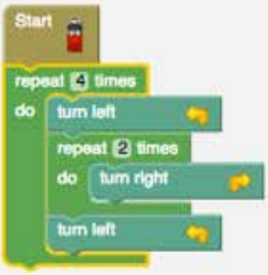
```
Start
repeat 3 times
do  move forwards
    turn left
    turn right
    turn left
```



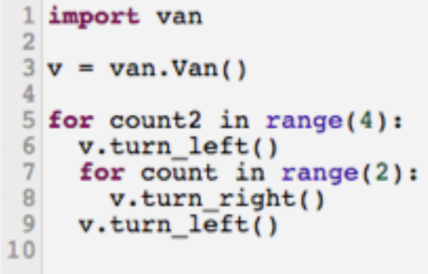
```
import van

v = van.Van()

for count in range(3):
  v.move_forwards()
  v.turn_left()
  v.turn_right()
  v.turn_left()

```

Level 84





```
Start
repeat 4 times
do  turn left
    repeat 2 times
    do  turn right
    turn left
```



```
import van

v = van.Van()

for count2 in range(4):
  v.turn_left()
  for count in range(2):
    v.turn_right()
  v.turn_left()

```

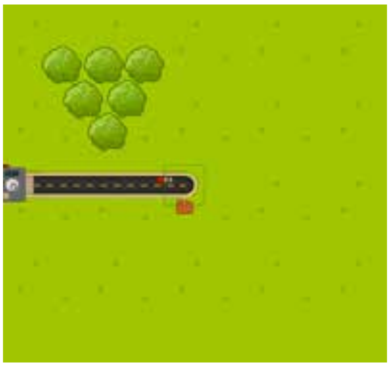
UKS2-S4

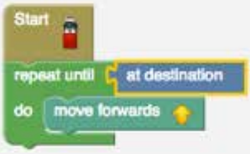
Understanding more Python commands (**while**, **if**.. **elif**..**else**)

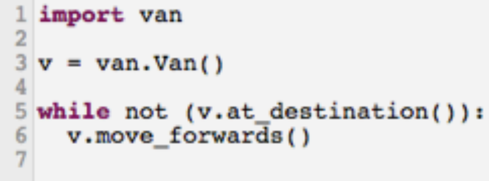
Objectives

- Create the core program in visual Blockly and convert it to Python
- Understand how the syntax of selection statements works in Python
- Understand the Python **while**, **if** , **elif** , **else** commands
- Analyse how **procedures** work in Python (extension)

Level 85

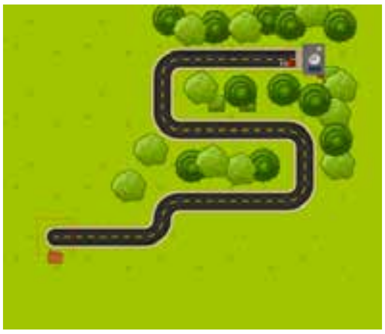


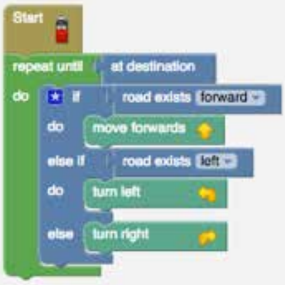


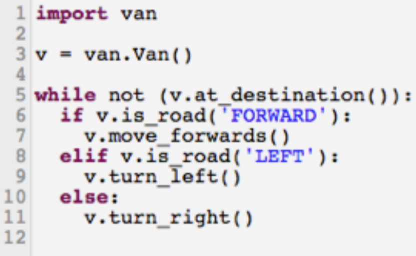


```
import van

v = van.Van()

while not (v.at_destination()):
  v.move_forwards()

```

Level 86

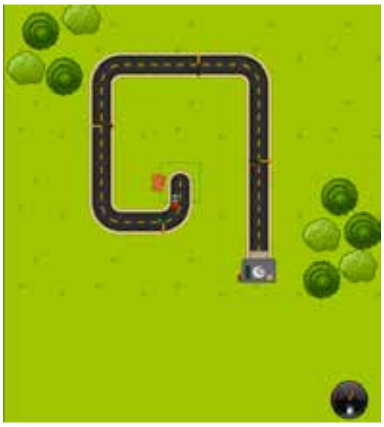


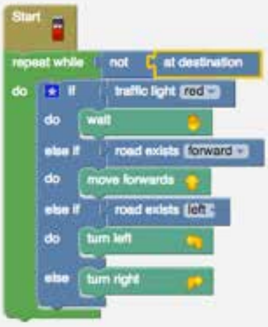




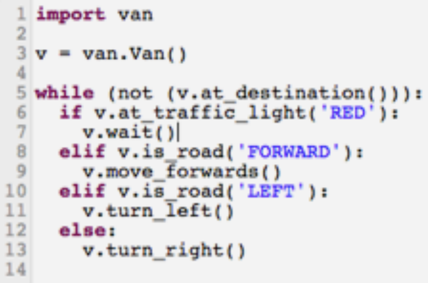
```
import van

v = van.Van()

while not (v.at_destination()):
  if v.is_road('FORWARD'):
    v.move_forwards()
  elif v.is_road('LEFT'):
    v.turn_left()
  else:
    v.turn_right()

```

Level 89



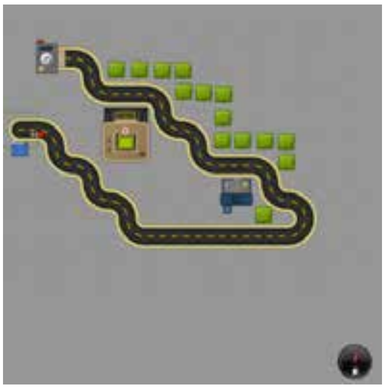


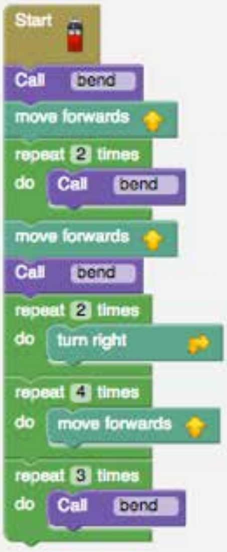
```
Start
repeat while  not  at destination
do  if  traffic light  red
    do  wait
    else if  road exists  forward
    do  move forwards
    else if  road exists  left
    do  turn left
    else  turn right
```



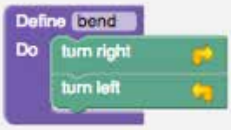
```python
import van

v = van.Van()

while (not (v.at_destination())):
  if v.at_traffic_light('RED'):
    v.wait()
  elif v.is_road('FORWARD'):
    v.move_forwards()
  elif v.is_road('LEFT'):
    v.turn_left()
  else:
    v.turn_right()

```

Level 90

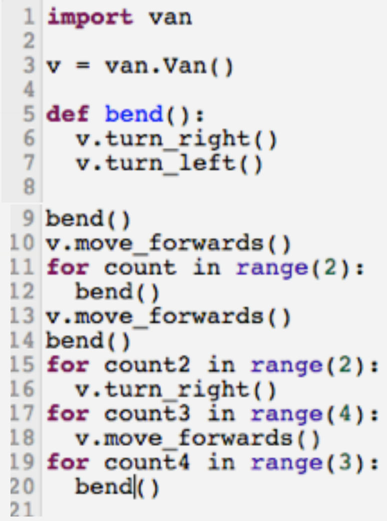




```
Start
Call  bend
move forwards
repeat 2 times
do  Call  bend
move forwards
Call  bend
repeat 2 times
do  turn right
repeat 4 times
do  move forwards
repeat 3 times
do  Call  bend
```

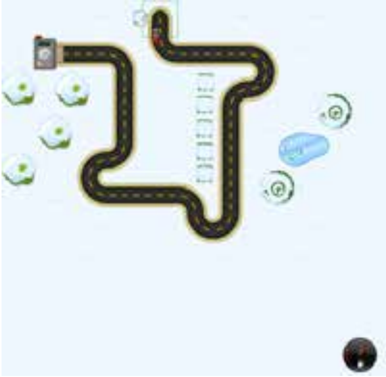


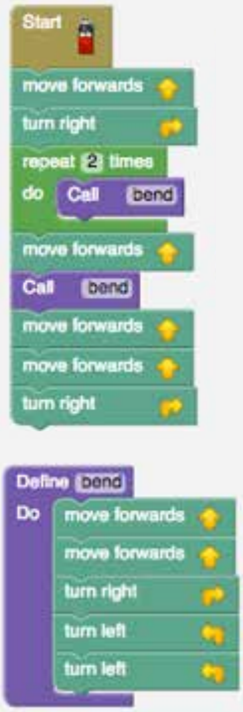
```
Define  bend
Do  turn right
    turn left
```



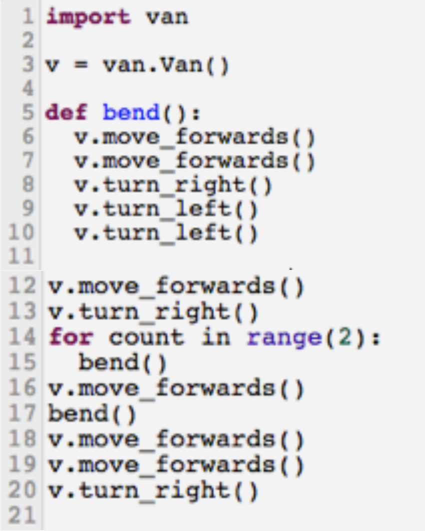
```python
import van

v = van.Van()

def bend():
  v.turn_right()
  v.turn_left()

bend()
v.move_forwards()
for count in range(2):
  bend()
v.move_forwards()
bend()
for count2 in range(2):
  v.turn_right()
for count3 in range(4):
  v.move_forwards()
for count4 in range(3):
  bend()

```

Level 91





```
Start
move forwards
turn right
repeat 2 times
do  Call bend
move forwards
Call bend
move forwards
move forwards
turn right

Define bend
Do  move forwards
    move forwards
    turn right
    turn left
    turn left
```



```
import van

v = van.Van()

def bend():
   v.move_forwards()
   v.move_forwards()
   v.turn_right()
   v.turn_left()
   v.turn_left()

v.move_forwards()
v.turn_right()
for count in range(2):
    bend()
v.move_forwards()
bend()
v.move_forwards()
v.move_forwards()
v.turn_right()

```

UKS2-S5

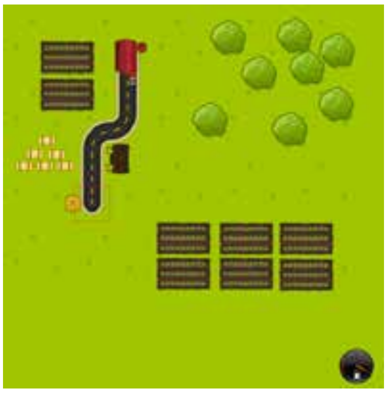
Writing basic code directly in Python (**forwards**, **turn**, **print**, **repetition**)

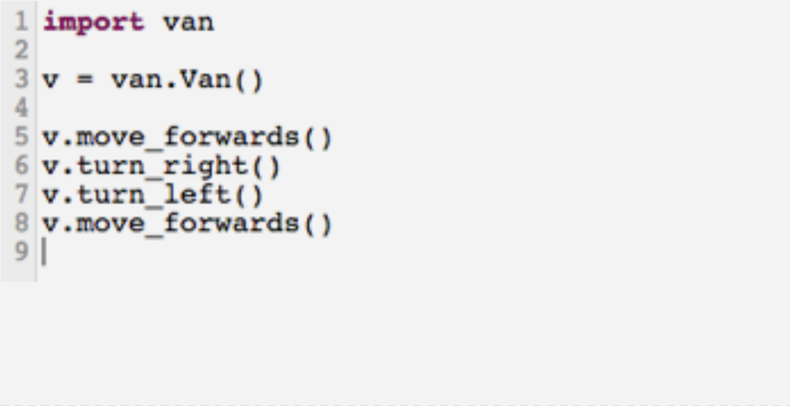
Objectives

- Write code in Python without the support of Blockly
- Write simple programs in Python using code for simple movement e.g. **v.move_forwards()**
- Use the print command in Python (not available in Blockly)
- Debug their Python programs, demonstrating an understanding of the appropriate syntax
- Use indents correctly in Python
- Use the Repeat loop … **for** count **in range (n)**:

Python

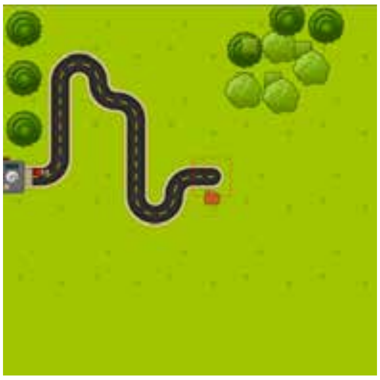
Level 92

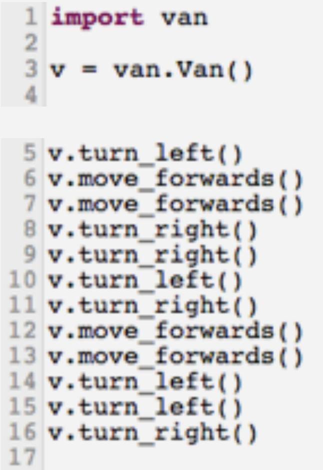




```
import van

v = van.Van()

v.move_forwards()
v.turn_right()
v.turn_left()
v.move_forwards()

```

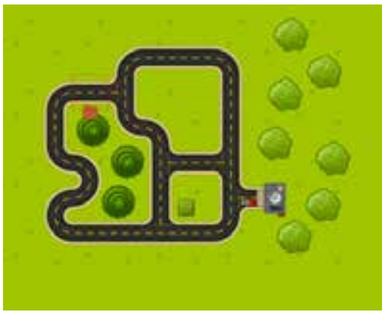
Level 93

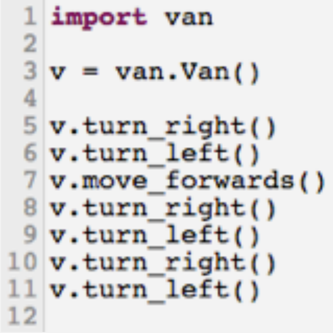




```
import van

v = van.Van()


v.turn_left()
v.move_forwards()
v.move_forwards()
v.turn_right()
v.turn_right()
v.turn_left()
v.turn_right()
v.move_forwards()
v.move_forwards()
v.turn_left()
v.turn_left()
v.turn_right()
```

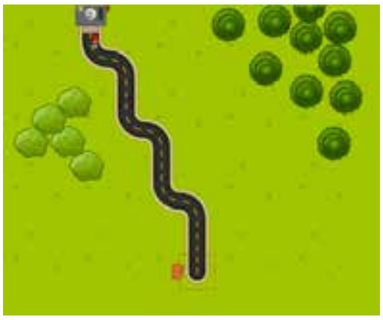
Level 94

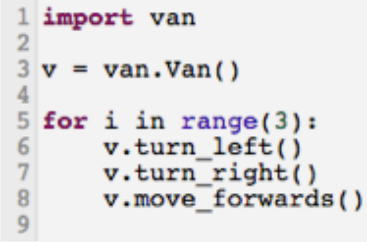




```
import van

v = van.Van()

v.turn_right()
v.turn_left()
v.move_forwards()
v.turn_right()
v.turn_left()
v.turn_right()
v.turn_left()
```

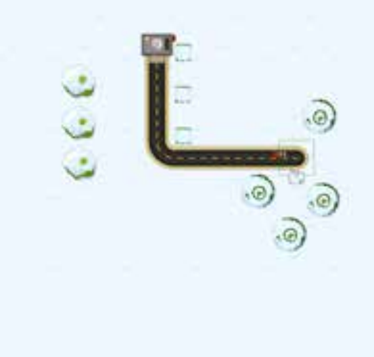
Level 95

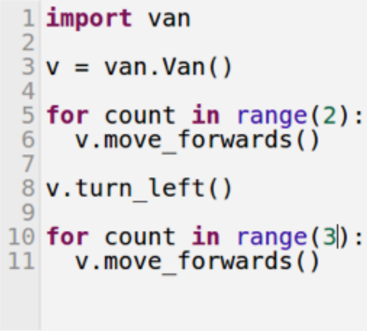




```
import van

v = van.Van()

for i in range(3):
    v.turn_left()
    v.turn_right()
    v.move_forwards()
```

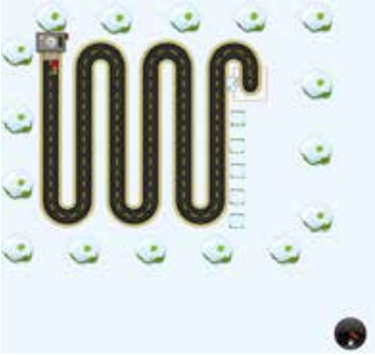
Level 96

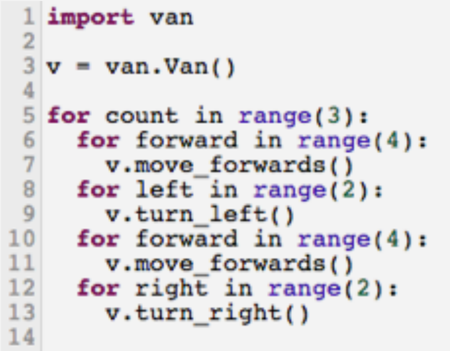




```
import van

v = van.Van()

for count in range(2):
  v.move_forwards()

v.turn_left()

for count in range(3):
  v.move_forwards()
```

Level 97





```
import van

v = van.Van()

for count in range(3):
  for forward in range(4):
    v.move_forwards()
  for left in range(2):
    v.turn_left()
  for forward in range(4):
    v.move_forwards()
  for right in range(2):
    v.turn_right()

```

UKS2-S6

Flying solo with Python! (programming independently using **repetition** and **selection**, **extension** to using **procedures** - several lessons)

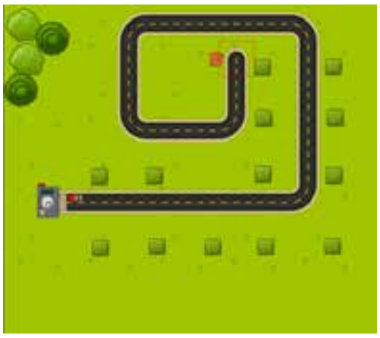
Objectives

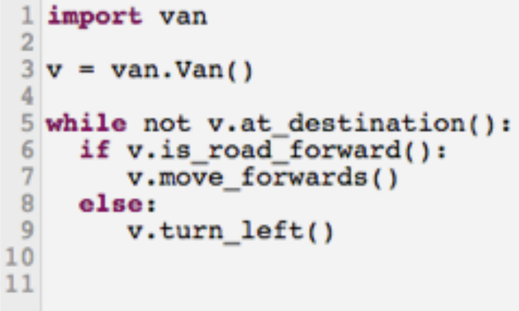
- Design and write programs independently in Python using **repetition** and **selection**: **for count in range (n):** and **while** , **if** , **elif**, **else**
- Debug Python programs, demonstrating an understanding of the appropriate syntax
- Use indents correctly in Python
- Use **comments** in Python to explain how the program works

Extension Objectives

- Defining new **procedures** in Python (also called **functions**)

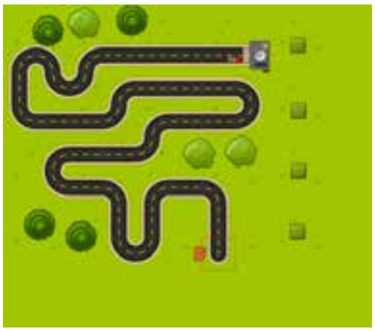
Level 98

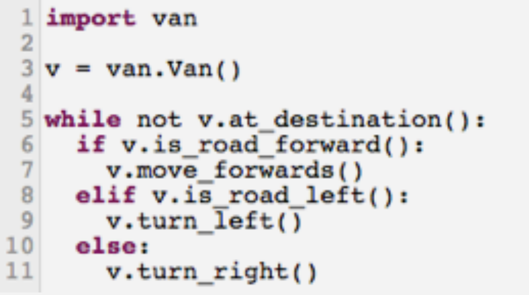




```
import van

v = van.Van()

while not v.at_destination():
  if v.is_road_forward():
     v.move_forwards()
  else:
     v.turn_left()


```

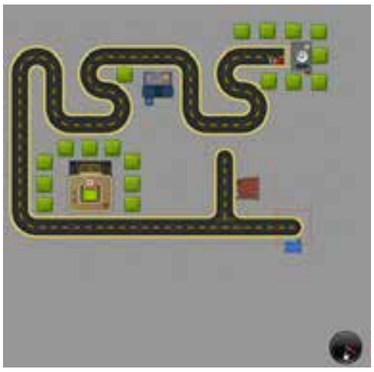
Level 99

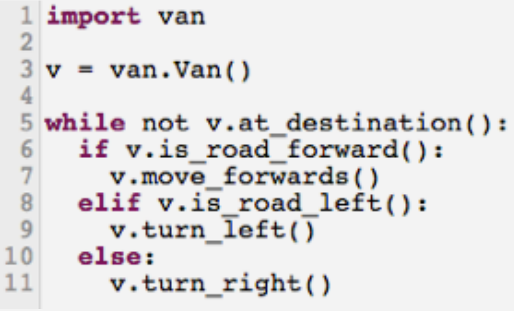




```
import van

v = van.Van()

while not v.at_destination():
  if v.is_road_forward():
    v.move_forwards()
  elif v.is_road_left():
    v.turn_left()
  else:
    v.turn_right()
```

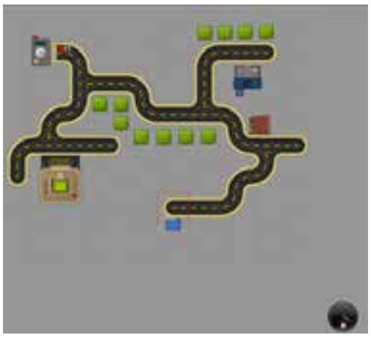
Level 100

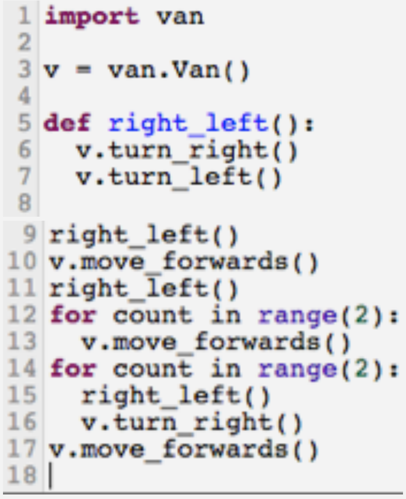




```
import van

v = van.Van()

while not v.at_destination():
  if v.is_road_forward():
    v.move_forwards()
  elif v.is_road_left():
    v.turn_left()
  else:
    v.turn_right()
```

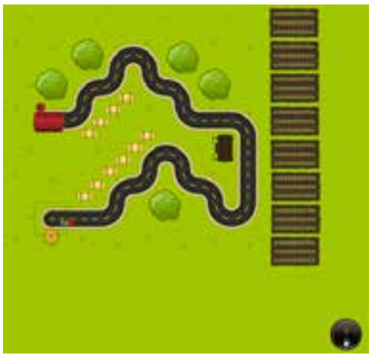
Level 101

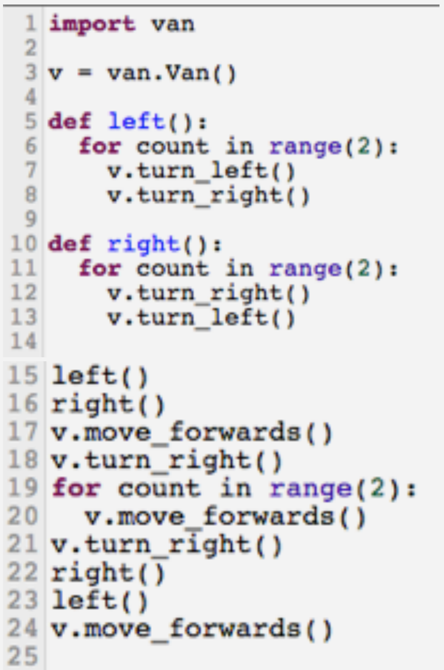




```
import van

v = van.Van()

def right_left():
  v.turn_right()
  v.turn_left()

right_left()
v.move_forwards()
right_left()
for count in range(2):
  v.move_forwards()
for count in range(2):
  right_left()
  v.turn_right()
v.move_forwards()
```

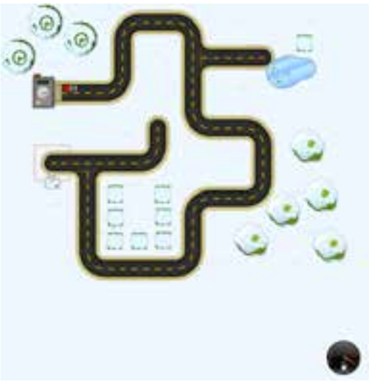
Level 102

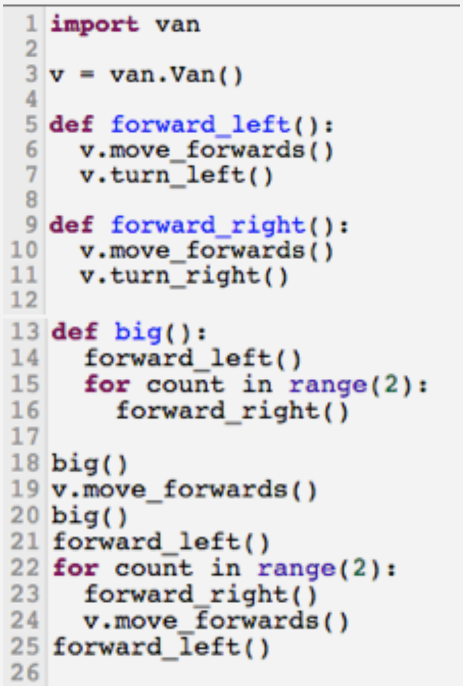




```
import van

v = van.Van()

def left():
  for count in range(2):
    v.turn_left()
    v.turn_right()

def right():
  for count in range(2):
    v.turn_right()
    v.turn_left()

left()
right()
v.move_forwards()
v.turn_right()
for count in range(2):
  v.move_forwards()
v.turn_right()
right()
left()
v.move_forwards()
```

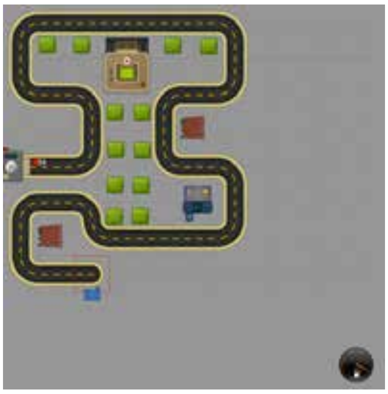
Level 103

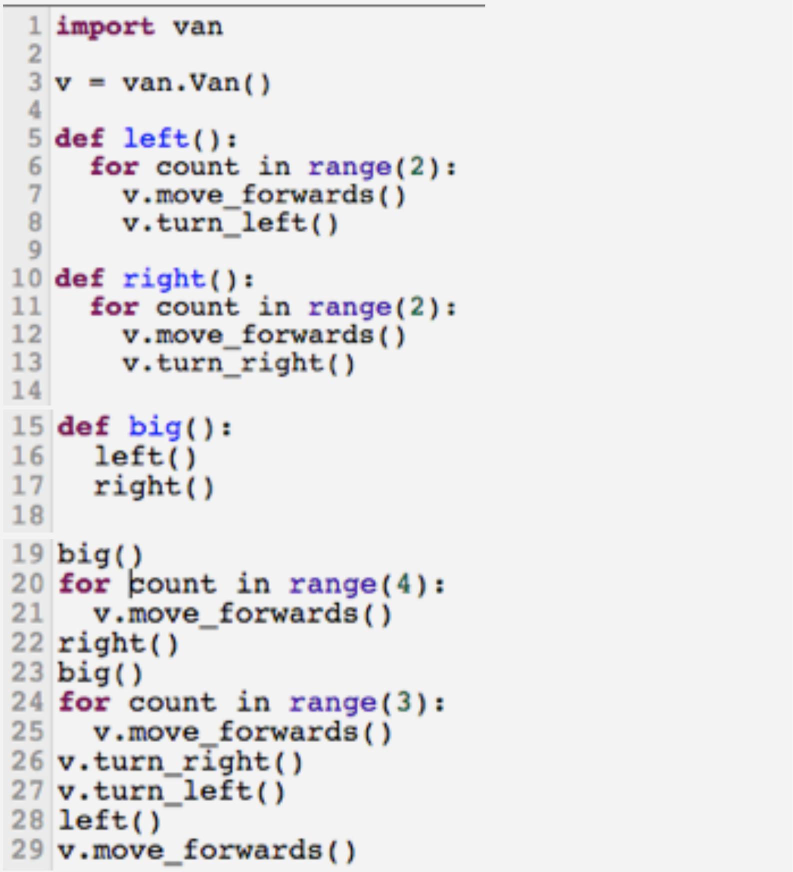




```
import van

v = van.Van()

def forward_left():
  v.move_forwards()
  v.turn_left()

def forward_right():
  v.move_forwards()
  v.turn_right()

def big():
  forward_left()
  for count in range(2):
    forward_right()

big()
v.move_forwards()
big()
forward_left()
for count in range(2):
  forward_right()
  v.move_forwards()
forward_left()
```

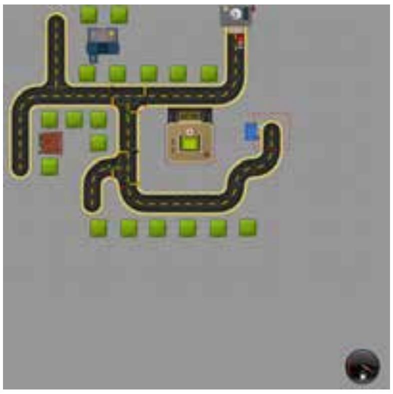
Level 104

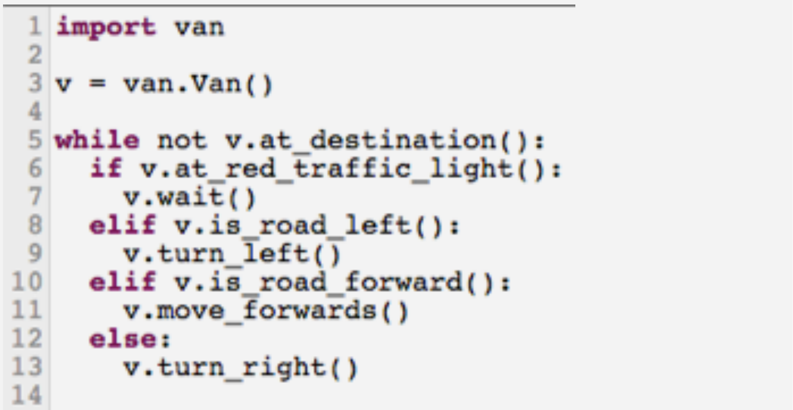




```
import van

v = van.Van()

def left():
   for count in range(2):
      v.move_forwards()
      v.turn_left()

def right():
   for count in range(2):
      v.move_forwards()
      v.turn_right()

def big():
   left()
   right()

big()
for count in range(4):
   v.move_forwards()
right()
big()
for count in range(3):
   v.move_forwards()
v.turn_right()
v.turn_left()
left()
v.move_forwards()
```

Level 105





```
import van

v = van.Van()

while not v.at_destination():
   if v.at_red_traffic_light():
      v.wait()
   elif v.is_road_left():
      v.turn_left()
   elif v.is_road_forward():
      v.move_forwards()
   else:
      v.turn_right()
```

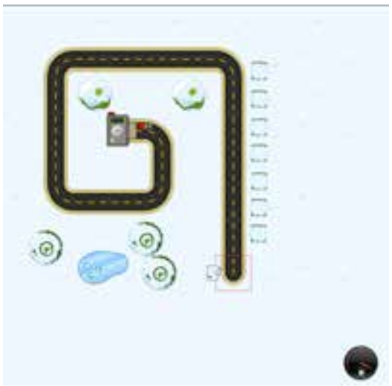
UKS2-S7

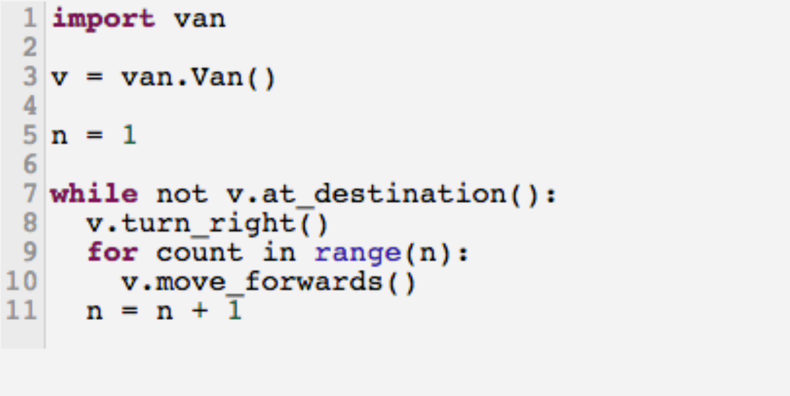
Creating new Python variables, incrementing variables

Objectives

- Design and write programs independently in Python using **repetition** and **selection**: **for count in range (n): and while** , **if** , **elif**, **else**
- Debug Python programs, demonstrating an understanding of the appropriate syntax
- Use indents correctly in Python
- Creating and increment **variables**
- Use **comments** in Python to explain programming

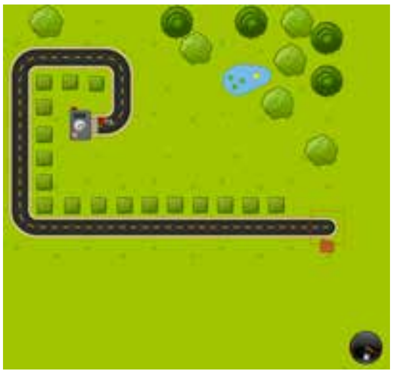
Level 106

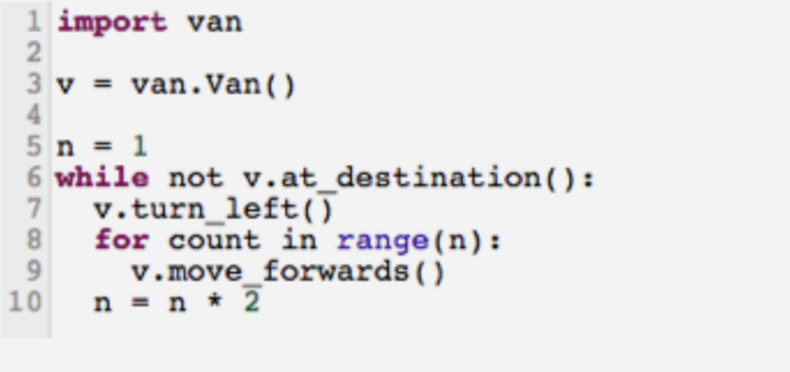




```
import van

v = van.Van()

n = 1

while not v.at_destination():
  v.turn_right()
  for count in range(n):
    v.move_forwards()
  n = n + 1
```

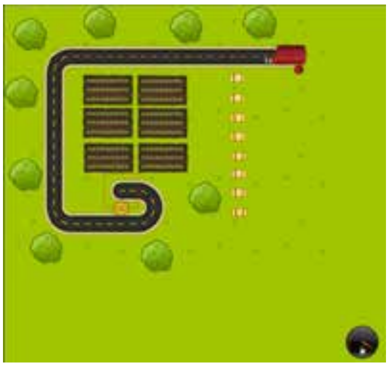
Level 107

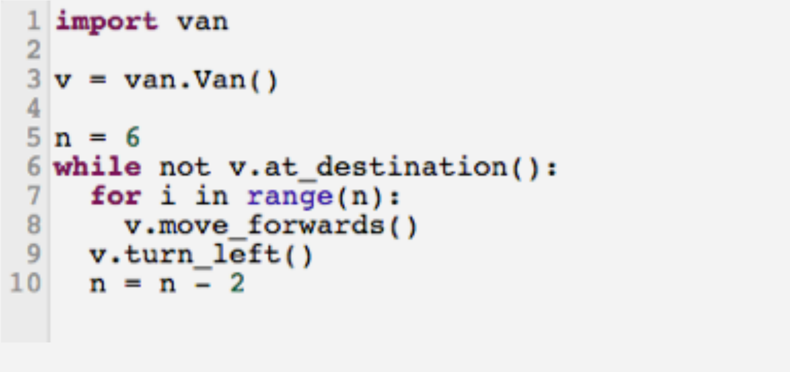




```
import van

v = van.Van()

n = 1
while not v.at_destination():
  v.turn_left()
  for count in range(n):
    v.move_forwards()
  n = n * 2
```

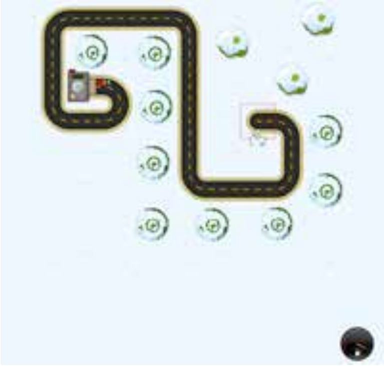
Level 108

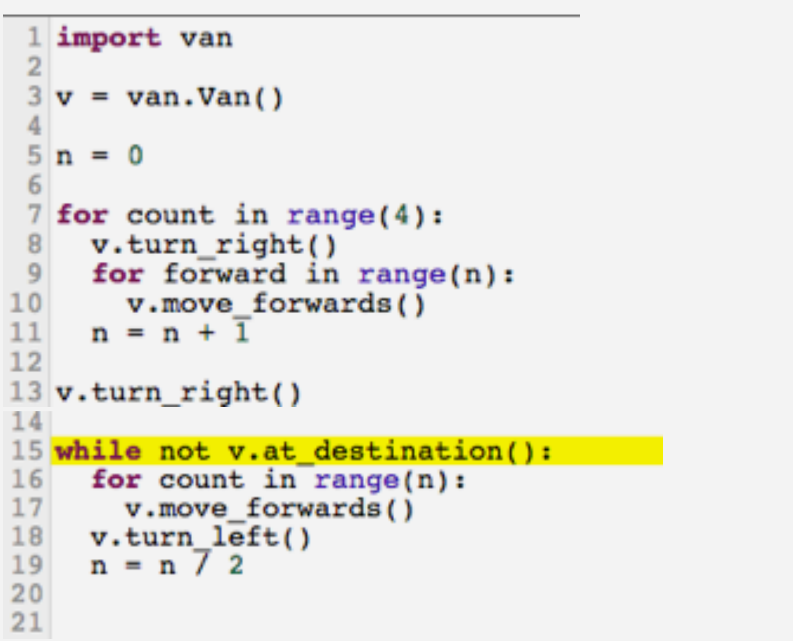




```
import van

v = van.Van()

n = 6
while not v.at_destination():
  for i in range(n):
    v.move_forwards()
  v.turn_left()
  n = n - 2
```

Level 109





```
import van

v = van.Van()

n = 0

for count in range(4):
  v.turn_right()
  for forward in range(n):
    v.move_forwards()
  n = n + 1

v.turn_right()

while not v.at_destination():
  for count in range(n):
    v.move_forwards()
  v.turn_left()
  n = n / 2
```