Upper Key Stage 2
Program Solutions Table

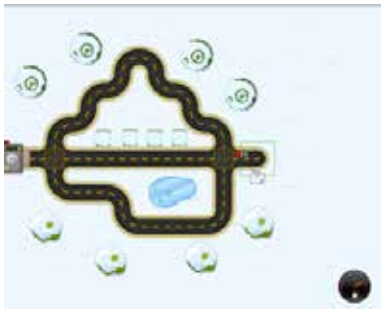
UKS2-S1

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

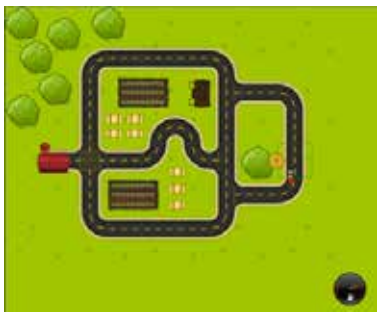
Objectives

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

Limited Blocks

Level 51		
Level 52		

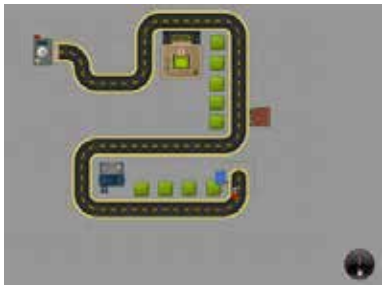
Level 53



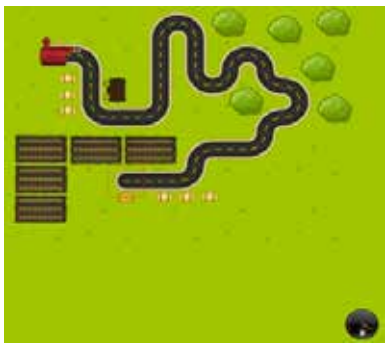
Level 54

Level 55

Level 56

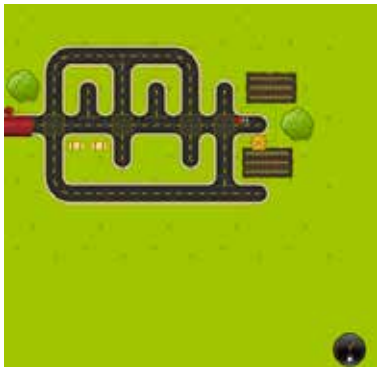


Level 57



Level 58

Level 59



Level 60

UKS2-S2

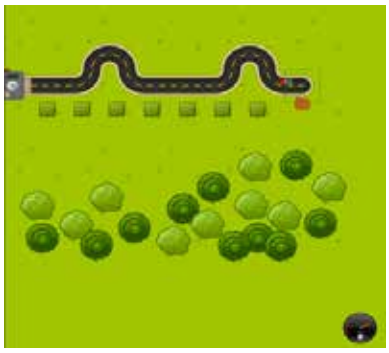
Breaking down the problem into chunks (understanding procedures)

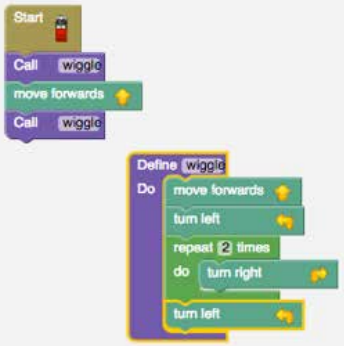
Objectives

- 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

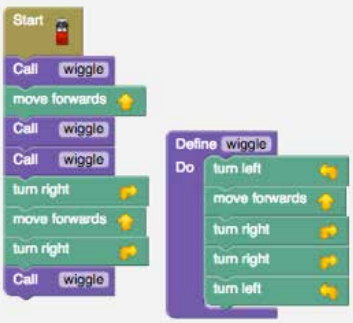
Level 61



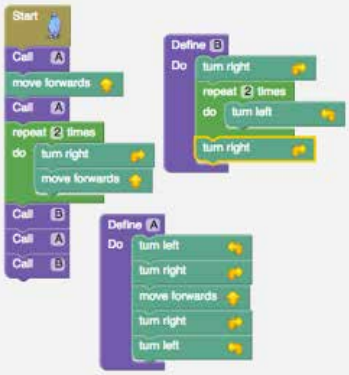


Level 62

Level 63

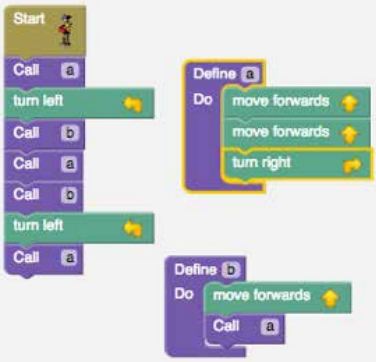


Level 64



Level 65

Level 66

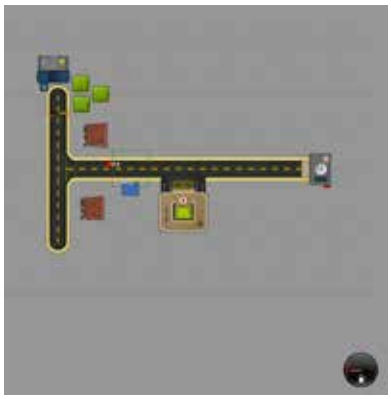


Level 67

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.

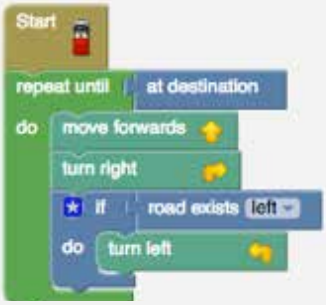
Level 68



Level 69

Level 70

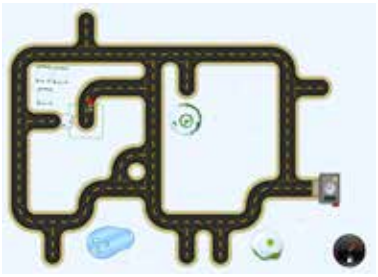
Level 71

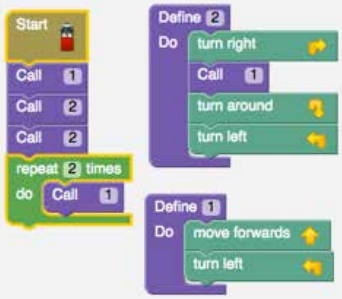


Level 72

Level 73

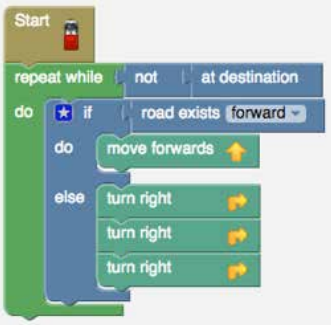
Level 74



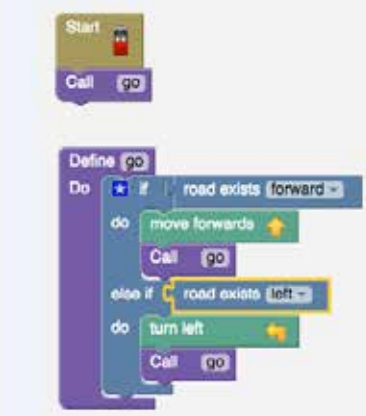


Level 75

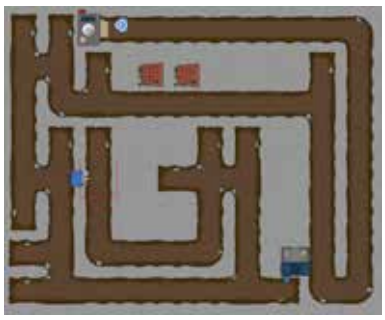
Level 76



Level 77



Level 78



Level 79

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

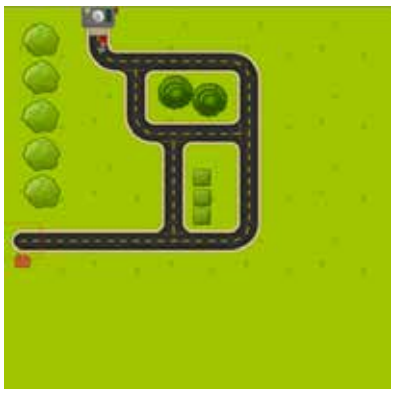
Level 80


```
import van

v = van.Van()

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

Level 81

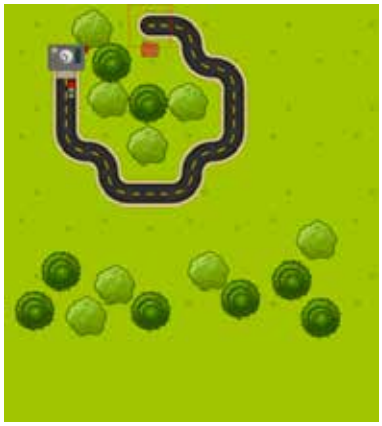

```
import van

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




```
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




```
import van

v = van.Van()

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

Level 84

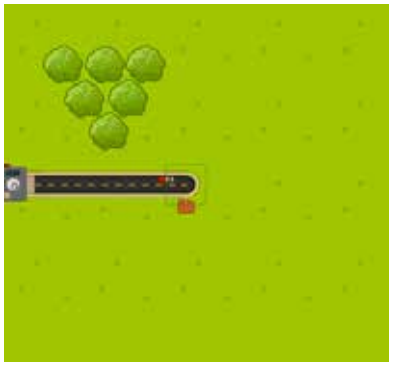

```
import van

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**)

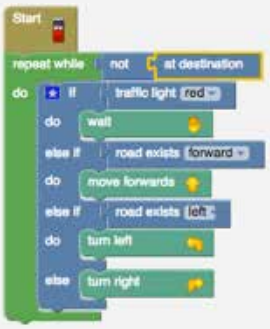
Objectives

- 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




```
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


```
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


```
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

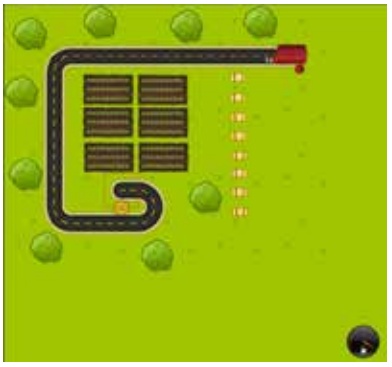
Level 106

```
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


```