

PROGRAMMING FUNDAMENTALS DAY 3

Andrew Buntine
Technical Director, Hardhat

WHAT ELSE CAN YOU DO?

Use APIs

- *Grab a picture from Reddit.com using their API*
- *Email that picture to a friend using the Mandrill API*



DEMO

<http://picster.hhd.com.au>

Source code:

<http://github.com/buntine/picster>

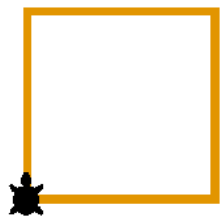
PROGRAMMING CONCEPTS

EXERCISE: TURTLE

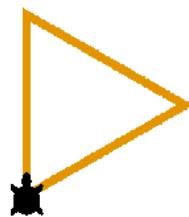
Imagine there was a turtle that understood the instructions (functions) on the right.

Break up into *groups of 2*

Write down a recipe (routine) for each of the pictures below, using the functions available.



Square



Triangle

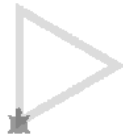


Double Square

Documentation

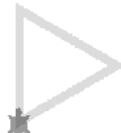
Available turtle functions

- Forward x metres
- Turn right x degrees
- Turn left x degrees
- Repeat x times [
 - *instructions*]





Forward 5 metres
Turn right 90 degrees
Forward 5 metres
Turn right 90 degrees
Forward 5 metres
Turn right 90 degrees
Forward 5 metres
Turn right 90 degrees

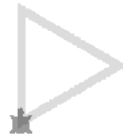




```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```





```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```





```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```





```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

?





```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```

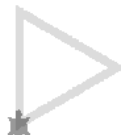




```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```

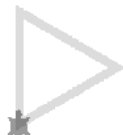




```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```



```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```



```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]  
Turn right 45 degrees
```



```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```



```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]  
Turn right 45 degrees  
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```




```
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees  
Forward 5 metres  
Turn right 90 degrees
```

or

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



```
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees  
Forward 5 metres  
Turn right 120 degrees
```

or

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```



```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]  
Turn right 45 degrees  
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```

or

```
Repeat 2 times [  
  Repeat 4 times [  
    Forward 5 metres  
    Turn right 90 degrees  
  ]  
  Turn right 45 degrees  
]
```

ALGORITHMS

AKA FUNCTIONS, METHODS, ROUTINES



Square

```
Repeat 4 times [  
  Forward 5 metres  
  Turn right 90 degrees  
]
```



Triangle

```
Repeat 3 times [  
  Forward 5 metres  
  Turn right 120 degrees  
]
```



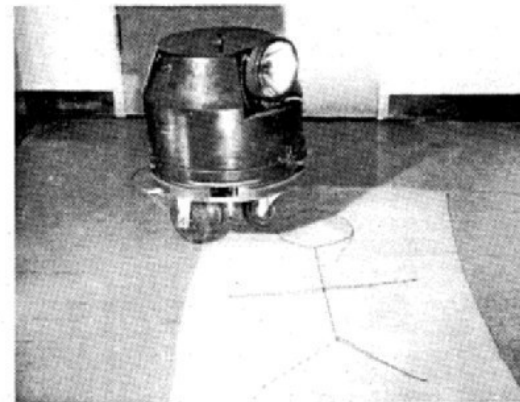
Double Square

```
Repeat 2 times [  
  Repeat 4 times [  
    Forward 5 metres  
    Turn right 90 degrees  
  ]  
  Turn right 45 degrees  
]
```

YOU JUST WROTE LOGO

<http://www.calormen.com/jslogo/>

1. Make a Turtle



The picture shows one of our turtles . . . so-called in honor of a famous species of cybernetic animal made by Grey Walter, an English neurophysiologist. Grey Walter's turtles had life-like behavior patterns built into its wiring diagram. Ours have no behavior except the ability to obey a few simple commands from a computer to which they are attached by a wire that plugs into a control-box that connects to a telephone line that speaks to the computer, which thinks it is talking to a

PERSON

FROM "TWENTY THINGS TO DO WITH A COMPUTER" 1971, SEYMOUR PAPERT & CYNTHIA SOLOMON

```
TO draw :distance
  forward :distance
  back :distance
END
```

```
TO vee :size
  left 50
  draw :size
  right 100
  draw :size
  left 50
```

END

```
TO circle :diameter
  arc 360 :diameter/2
END
```

```
TO person :size
  right 180
  vee :size
  right 180
  forward :size
  vee :size
  forward :size/2
  penup
  forward 50
  pendown
  circle :size
```

END

```
person 100
person 10
```

3. Make a Turtle



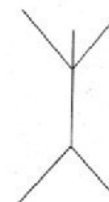
The picture shows one of our turtles . . . so-called in honor of a famous species of cybernetic animal made by Grey Walter, an English neurophysiologist. Grey Walter's turtles had life-like behavior patterns built into its wiring diagram. Ours have no behavior except the ability to obey a few simple commands from a computer to which they are attached by a wire that plugs into a control-box that connects to a telephone line that speaks to the computer, which thinks it is talking to a

Now

```
TO MAN :SIZE
1 VEE :SIZE
2 RIGHT 180
3 FORWARD :SIZE
4 VEE :SIZE
5 FORWARD :SIZE/2
```

We now use the previously defined command in making our new command. In other words TO DRAW was a sub-procedure of TO VEE; TO VEE is a sub-procedure of TO MAN.

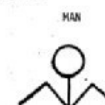
MAN 100 will draw



MAN 10 will draw

X

Here are some other drawings the fifth grade kids made the turtle draw.



MAN

MAN



A CREATIVE PROCESS

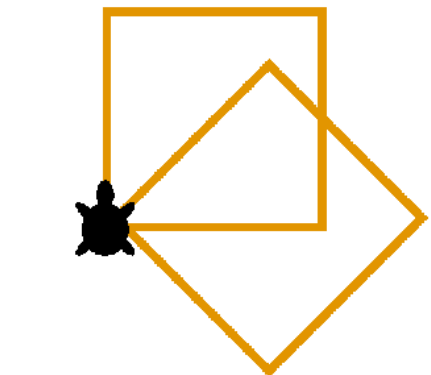
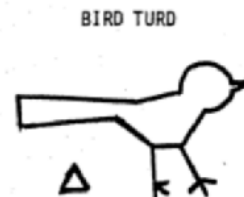
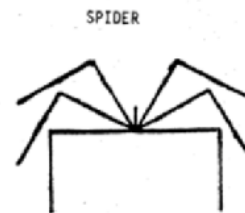
Understanding the problem

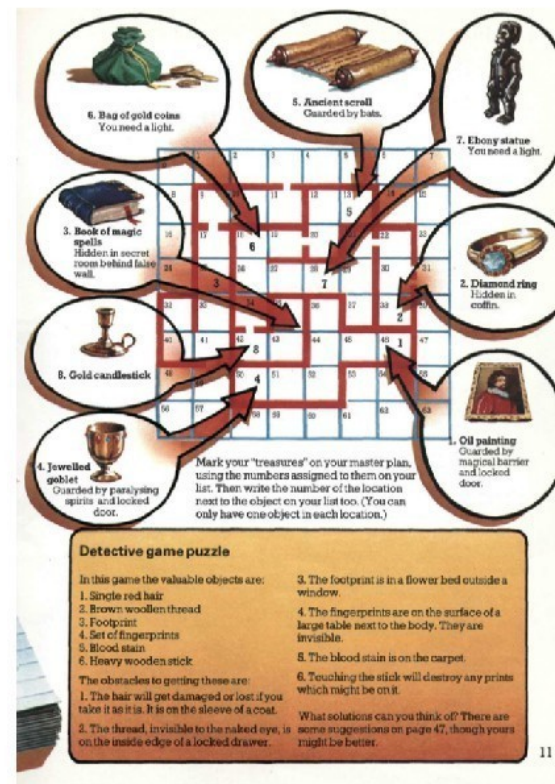
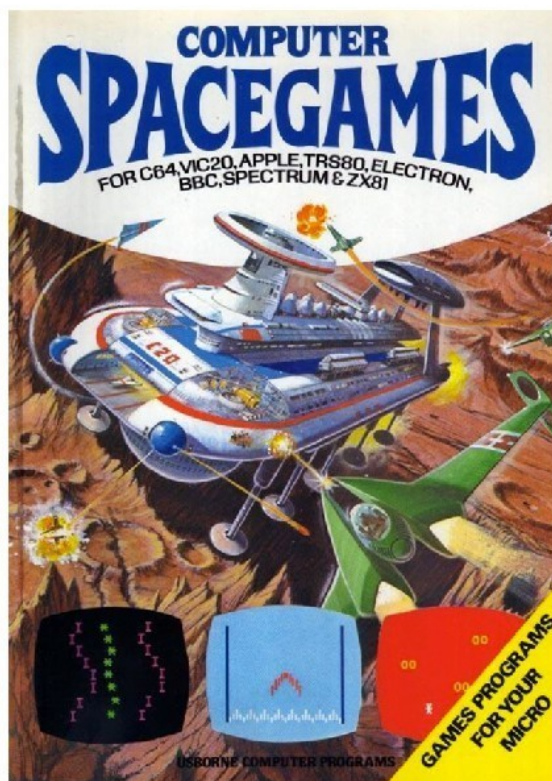
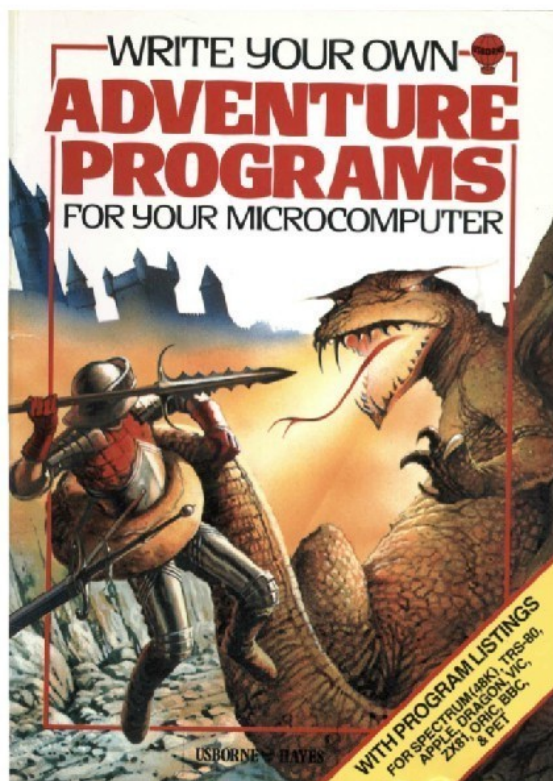
Exploring solutions

Pattern recognition

Abstraction

Sharing ideas





IDEAS JUST COVERED

Functions and libraries

Pre-built library functions such as Forward

Looping

Repeat things using repeat

Abstraction

Created our own functions such as square

Parameters

Allowed for adjustments of numbers such as length

ABSTRACTION

Problem



Solution

```
repeat 3 [  
  forward 100  
  right 120  
]
```

Abstraction

```
T0 triangle  
  repeat 3 [  
    forward 100  
    right 120  
  ]  
END
```

Re-Use

ABSTRACTION

Problem



Solution

```
repeat 3 [  
  forward 100  
  right 120  
]
```

Abstraction

```
T0 triangle  
  repeat 3 [  
    forward 100  
    right 120  
  ]  
END
```

Re-Use

```
left 90  
repeat 2 [  
  triangle  
  forward 25  
]
```



WHY ABSTRACT YOUR CODE?

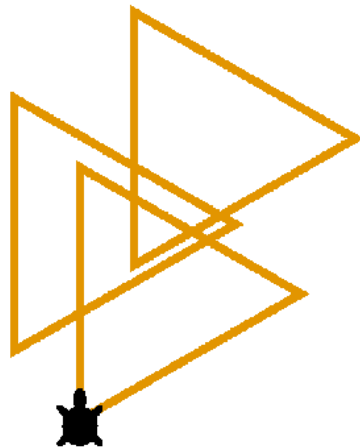
- Simplifies your code
- Allows yourself (and others) to re-use your work
- Libraries and frameworks provide pre-built abstractions



PARAMETERS

Without parameters

```
T0 triangle  
  repeat 3 [  
    forward 100  
    right 120  
  ]  
END
```

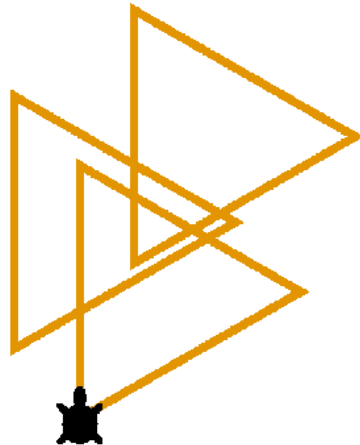


With parameters

PARAMETERS

Without parameters

```
T0 triangle  
  repeat 3 [  
    forward 100  
    right 120  
  ]  
END
```



With parameters

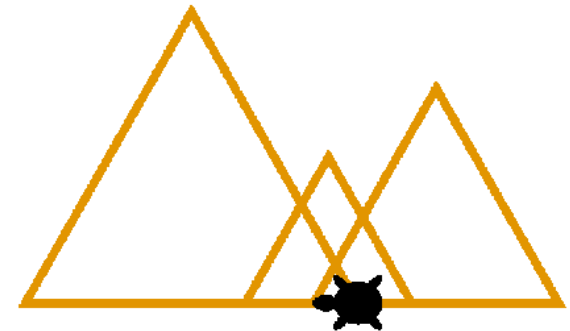
```
T0 triangle :size  
  repeat 3 [  
    forward :size  
    right 120  
  ]  
END
```



WHY USE PARAMETERS?

Benefits:

- Makes your code easier to read
- Makes your code more re-usable and generic



```
T0 triangle :size      left 90
  repeat 3 [           triangle 80
    forward :size      forward 50
    right 120          triangle 50
  ]                   forward 20
END                   triangle 100
```

