MAT 7003 : Mathematical Foundations

(for Software Engineering)

J Paul Gibson, A207

paul.gibson@it-sudparis.eu

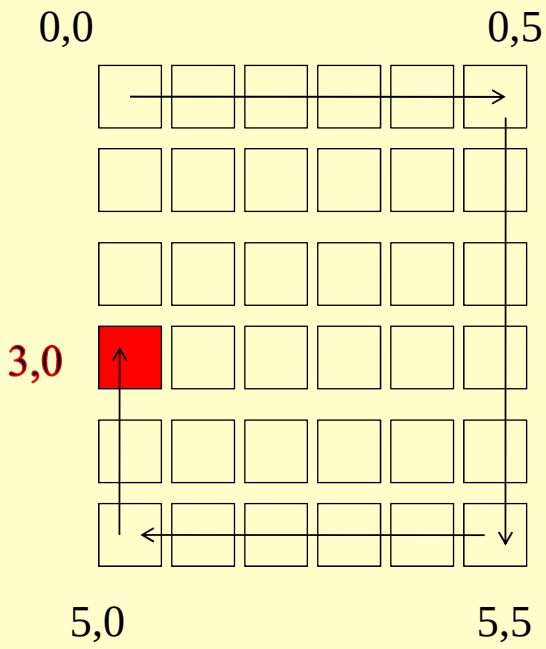
http://www-public.it-sudparis.eu/~gibson/Teaching/MAT7003/

L0-Introduction-SpiralWalkSolution

/~gibson/Teaching/MAT7003/L0-Introduction-SpiralWalkSolution.pdf

Note: all code and documentation can be downloaded from web site

REMINDER - Robot Walker: step1



You are to specify a function, f, that:

Takes as input the:

•size of a square grid

Calculates the x,y co-ordinates of the robot after it has walked *half way* around the grid following a spiral walk, starting at 0,0 and moving clockwise.

In the example the function Calculates
f(6) = (3,0)

An operational specification – as a computer program – can solve the problem

For example, this is output when you compile and execute `SpiralWalkerTest.java`

```
The size of the grid is set, by default, to 6.
You can override this default value by passing an integer value as a
main argument parameter, if you so wish.

**********************************************************************
SpiralWalker Test (Size 6). Execution Date/Time 2010/10/05 17:32:49
**********************************************************************

Grid size of 6 means walker must go through 18 steps
The current position is column 0, row 2
The current direction is up
++++++
-----+
+----+
+----+
+----+
++++++
```

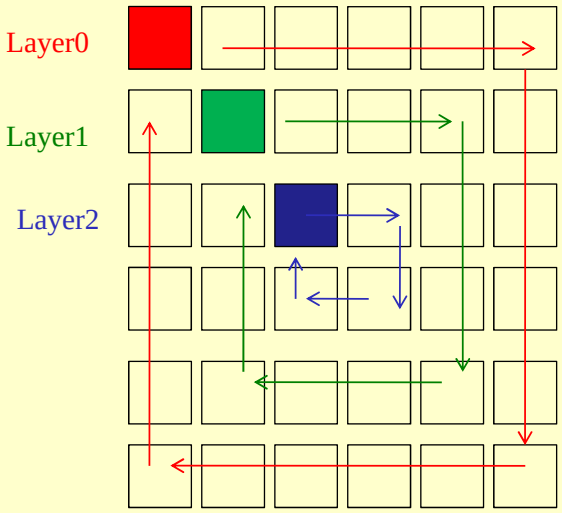
Note: this result is 1 position out from the specified test on the earlier slide as it counts the *steps between locations* and not the locations. (But this is acceptable.)

We can use this code to generate a range of results, that might help us identify a (mathematical) pattern

For example, this is output when you compile and execute `SpiralWalkStatistics.java`

```
Calculating the results of the spiral walk function for input values 1..20
Grid size: 1. Position of walk half-way round spiral, column 0 row 0
Grid size: 2. Position of walk half-way round spiral, column 1 row 1
Grid size: 3. Position of walk half-way round spiral, column 2 row 2
Grid size: 4. Position of walk half-way round spiral, column 1 row 3
Grid size: 5. Position of walk half-way round spiral, column 0 row 4
Grid size: 6. Position of walk half-way round spiral, column 0 row 2
Grid size: 7. Position of walk half-way round spiral, column 1 row 1
Grid size: 8. Position of walk half-way round spiral, column 5 row 1
Grid size: 9. Position of walk half-way round spiral, column 7 row 3
Grid size: 10. Position of walk half-way round spiral, column 8 row 8
Grid size: 11. Position of walk half-way round spiral, column 5 row 9
Grid size: 12. Position of walk half-way round spiral, column 1 row 9
Grid size: 13. Position of walk half-way round spiral, column 1 row 5
Grid size: 14. Position of walk half-way round spiral, column 4 row 2
Grid size: 15. Position of walk half-way round spiral, column 10 row 2
Grid size: 16. Position of walk half-way round spiral, column 13 row 7
Grid size: 17. Position of walk half-way round spiral, column 14 row 14
Grid size: 18. Position of walk half-way round spiral, column 7 row 15
Grid size: 19. Position of walk half-way round spiral, column 2 row 14
Grid size: 20. Position of walk half-way round spiral, column 2 row 6
```

Now some layer analysis and mathematical modelling



I propose the following:

Layer Theorem:

The layer of the halfway point of square grid size*size is given by the formula:

LayerMidPoint(size) = size/7

So assuming the theorem to be correct, for *size=6*, the midpoint is in *layer 0*.
We can validate, through observation, that this is the case.

Now some layer analysis and mathematical modelling

Layer Theorem:
The layer of the halfway point of square grid size*size is given by the formula:

LayerMidPoint(size) = size/7

Calculating the results of the spiral walk function for input values 1..20 –

```
Grid size: 1. Position of walk half-way round spiral, column 0 row 0
Grid size: 2. Position of walk half-way round spiral, column 1 row 1
Grid size: 3. Position of walk half-way round spiral, column 2 row 2
Grid size: 4. Position of walk half-way round spiral, column 1 row 3
Grid size: 5. Position of walk half-way round spiral, column 0 row 4
Grid size: 6. Position of walk half-way round spiral, column 0 row 2
Grid size: 7. Position of walk half-way round spiral, column 1 row 1
Grid size: 8. Position of walk half-way round spiral, column 5 row 1
Grid size: 9. Position of walk half-way round spiral, column 7 row 3
Grid size: 10. Position of walk half-way round spiral, column 8 row 8
Grid size: 11. Position of walk half-way round spiral, column 5 row 9
Grid size: 12. Position of walk half-way round spiral, column 1 row 9
Grid size: 13. Position of walk half-way round spiral, column 1 row 5
Grid size: 14. Position of walk half-way round spiral, column 4 row 2
Grid size: 15. Position of walk half-way round spiral, column 10 row 2
```

Now some layer analysis and mathematical modelling

Layer Theorem: *check* the change of layer from size 6 to 7

```
Grid size: 6. Position of walk half-way round spiral, column 0 row 2
Grid size: 7. Position of walk half-way round spiral, column 1 row 1
```

Execute the following commands:

`java SpiralWalkTest 6`

```
Grid size of 6 means walker must go
through 18 steps
The current position is column 0,
row 2
The current direction is up
++++++
-----+
+----+
+----+
+----+
++++++
```

`java SpiralWalkTest 7`

```
Grid size of 7 means walker must
go through 24 steps
The current position is column 1,
row 1
The current direction is right
+++++++
++----+
+-----+
+-----+
+-----+
+-----+
+++++++
```


Now some layer analysis and mathematical modelling

Let's see if the layer analysis gives the right answer for another test case, eg `size = 22.`

By layer theorem, the layer of the mid point is 22/7 = 3

The distance travelled until we move from layer 2 to layer 3 =

(size-1)*4 + (size-3)*4 + (size-5)*4 = 4*(3*size-9) = 228

The total walk is 22*22/2 = 242 so we have 14 steps to make after we arrive at the correct layer

The start co-ordinates of layer 3 are (3,3) so the midpoint is **(17,3)**

Now some layer analysis and mathematical modelling

Let's see if this is coherent with our java code .

```
java SpiralWalkTest 22

Grid size of 22 means walker must go through 242 steps
The current position is column 17, row 3
The current direction is right
++++++++++++++++++++++
++++++++++++++++++++++
++++++++++++++++++++++
++++++++++++++++++-+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
+++----------------+++
++++++++++++++++++++++
++++++++++++++++++++++
++++++++++++++++++++++
```

TO DO: try to prove/understand the **Layer Theorem**