FIT 100 Project 3 – Conway's Game of Life

- A kind of cellular automaton
- Proposed by the mathematician John Conway in 1970.
- Complex behavior emerges from simple rules.
- The Game of Life itself is not a realistic simulation of any actual phenomenon
- However, researchers have used cellular automata to model the spread of weeds, fire, urban sprawl, and other phenomena

FIT 100 Project 3 – Purpose

- Experience with one kind of simulation
- Experience using arrays (including 2 dimensional arrays)
- Experience modifying an existing program

FIT 100 The Rules of the Game

- A game in the sense of a simple set of rules that can give rise to complex behavior, rather than in the sense of several players competing.
- The game is played on a 2 dimensional array of cells.
- Each cell is either alive or dead. We are given some starting configuration of live and dead cells.
- At each step, we compute the next state of the array.
- The game continues indefinitely, although obviously you'll want to stop it at some point.

FIT 100 Computing the Next State

- For each cell, calculate how many live neighbors it has. Each cell has 8 neighbors:

1	2	3
4		5
6	7	8

FIT 100 Computing the Next State (2)

- For each cell, calculate how many live neighbors it has out of its 8 neighbors.
- If the cell is alive:
  - If it has 2 or 3 live neighbors, it remains alive.
  - If it has 0 or 1 live neighbors, it does of loneliness
  - If it has 4 or more live neighbors, it dies of overcrowding.
- If the cell is dead:
  - If it has 3 live neighbors, it comes alive
  - Otherwise it remains dead.

FIT 100 Computing the Next State (3)

- The next state for each cell is computed using the current states of its neighbors. (You'll get wrong results if you update a cell and then use the updated state of that cell when computing the state of its neighbors.)

FIT 100 One and Two Dimensional Arrays

- ooops ….. Visual Basic does allow you to declare both the lower and upper bounds of an array:
  Dim a(1 To 100) as Integer
- You can have two dimensional arrays as well as one dimensional ones. This is much more natural for the Game of Life

```
Dim cells (0 To 31, 0 To 31) As Integer
Dim i As Integer
cells(2,3) = 1
i = cells(4,5)
```


FIT 100 Visualizing a 2-D Array

- Dim a (1 To 3, 1 To 3) As Integer

We'll regard the first subscript as the row, and the second as the column

a(1,1)	a(1,2)	a(1,3)
a(2,1)	a(2,2)	a(2,3)
a(3,1)	a(3,2)	a(3,3)

FIT 100 More on Declaring Arrays

- Unless you do something special, the bounds of an array in the declaration must be constants:
  Dim a(1 To 100) as Integer
- We can declare constants though for easier modification:

```
Const nRows As Integer = 30
Const nCols As Integer = 30
Dim newCells (1 To nRows, 1 To nCols) as Integer
```


FIT 100 For Loops

- Do while is going to get pretty tedious for the Game of Life, because we iterate through arrays all the time
- Equivalent constructs:

```
Dim i As Integer
i = 1
Do While i<=10
  Print i
  i = i+1
Loop
```

```
Dim i As Integer
For i = 1 To 10
  Print i
Next i
```


FIT 100 Mini-Exercise

- What does this print?

```
Dim i As Integer
For i = 3 To 6
   Print i
Next i
```

- Write a for loop that prints the integers between 0 and 10 inclusive.

FIT 100 Mini-Exercise – Answer

- What does this print?

```
Dim i As Integer
For i = 3 To 6
   Print i
Next i
```

```
3
4
5
6
```

- Write a for loop that prints the integers between 0 and 10 inclusive.

```
Dim i As Integer
For i = 0 To 10
   Print i
Next i
```

FIT 100 Mini-Exercise

- What does this print?

```
Dim i As Integer
For i = 3 To 2
   Print i
Next i
```


FIT 100 Mini-Exercise -- Answer

- What does this print?

```
Dim i As Integer
For i = 3 To 2
   Print i
Next i
```

Nothing!

FIT 100 Nested For Loops

- Just as with Do While loops, we can nest one for loop inside another. This is very common when dealing with 2-dimensional arrays

```
Dim i As Integer, j As Integer
For i = 1 To 3
  For j = 1 To 2
    Print i, j
  Next j
Next i
```

Result:
1 1
1 2
2 1
2 2
3 1
3 2

FIT 100 Mini-Exercise

- What does this print?

```
Dim i As Integer, j As Integer
For i = 0 To 4
  For j = 1 To 2
    Print i, j
  Next j
Next i
```


FIT 100 Mini-Exercise – Answer

- What does this print?

```
Dim i As Integer, j As Integer
For i = 0 To 4
  For j = 1 To 2
    Print i, j
  Next j
Next i
```

0 1
0 2
1 1
1 2
2 1
2 2
3 1
3 2
4 1
4 2

FIT 100 Mini-Exercise

- What does this print?

```
Dim i As Integer, j As Integer
For i = 0 To 4
  For j = 1 To 2
    Print i, j
  Next j
  Print "hi there"
Next i
```

FIT 100 Mini-Exercise – Answer

- What does this print?

```
Dim i As Integer, j As Integer
For i = 0 To 4
  For j = 1 To 2
    Print i, j
  Next j
  Print "hi there"
Next i
```

```
0  1
0  2
hi there
1  1
1  2
hi there
2  1
2  2
hi there
3  1
3  2
hi there
4  1
4  2
hi there
```