Fields and Attractors
-> the use of turtles (agents) as particles

'other ways of drawing circles'

As part of the "new epistemology", (the idea that there are new ways of knowing things) Resnic cites the "mathematical" idea of a circle and Seymour Papert's Logo method.

In conventional terms you can get a circle by knowing that a point on the circumference of a circle centred on ORIGIN of radius R is given by

```
Xcirc = originX + R cos (angle)
Ycirc = originY + R sin (angle)
```

However, using the logo turtle drawing method we can say

```
To circle
      Repeat 36
            Forward 1
            Turn Left 10
      End repeat
End circle
```

The first example uses mathematical knowledge without understanding, the second uses an algorithm to drive a little turtle round in a circle, by moving forwards and turning 10 degrees 36 times. As well as being easier to understand (it requires only English and a familiarity with walking) the turtle drawing method is particularly suited to computer implementation since computers are good at doing things over and over again, and the transparency of the process makes it easy to adapt to make different turtle drawings.

This example is borrowed from Michel Resnick's book 'Turtles, Termites and Traffic Jams', with a debt to the related but subtly different circles program in the StarLogo distribution.

Our example today is one step further from Papert's because we are now using parallel computation to make a whole lot of turtles draw a circle.

The program shows how turtles can decide where to go on the basis of other turtles, and is interesting because it shows some interesting dynamics arising from the parallel interactions of the turtles.

Each turtle sets for the center of the 'universe' (0,0), when outside an observer-set radius to the center. Otherwise, in other words when inside that radius, the turtles retreat away from the center. This makes a circular arrangement of circles, but it is a bit uneven. To spread the turtles out we should get the turtles to make the distances between themselves more equal.

How to do this?

Measure the circumference and somehow move the turtles on circular arcs more or less...? One simple way (which is an example of how thinking parallelly is a bit counter intuitive) is to just get the turtles to back away from each other by asking each turtle to check the distance between itself and the nearest turtle and to back off. Because they are all doing this at the same time they will end up smoothing out the bumps as it were.

```
;******************set-up procedure*****************
turtles-own[ closest-turtle ]

to setup
      ca
      ask patches [set pcolor white]
      crt density
      ask turtles
      [
            setxy random-int-or-float screen-size-x _
            random-int-or-float screen-size-y
            set color black
            set shape "circle"
      ]
end

;******************main procedures*****************
to attract
      ask turtles
      [
            set heading  towardsxy 0 0
            ifelse ((distancexy 0 0 ) < radius)
                  [bk 1]
                  [fd 1]
      ]
end

to repel
      ask turtles
      [
            set closest-turtle min-one-of turtles with _
            [self != myself] [distance myself]
            set heading towards closest-turtle
            bk repel-strength
]
end
```

The project therefore has three procedures:

1 the standard setup procedure to create turtles and sprinkle them about

2 the attract procedure which makes the circle

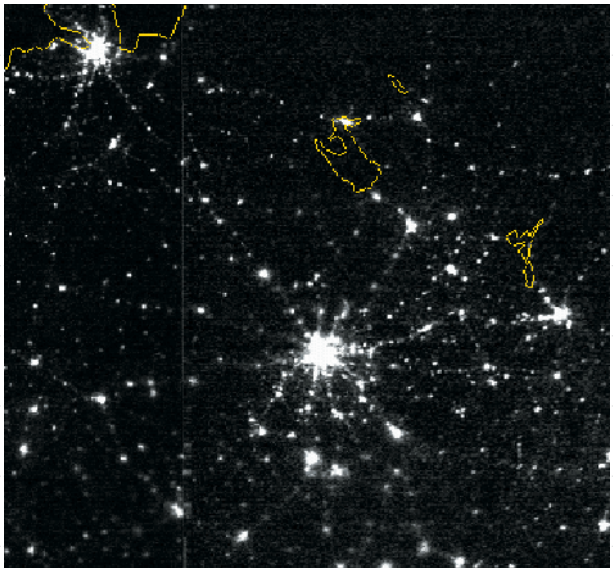
3 the repel procedure to smooth it out

If you turn on the attract button then the agents form a circle, though it is a bit uneven.
Play with this project by altering the repel strength using the slider. What do you observe?

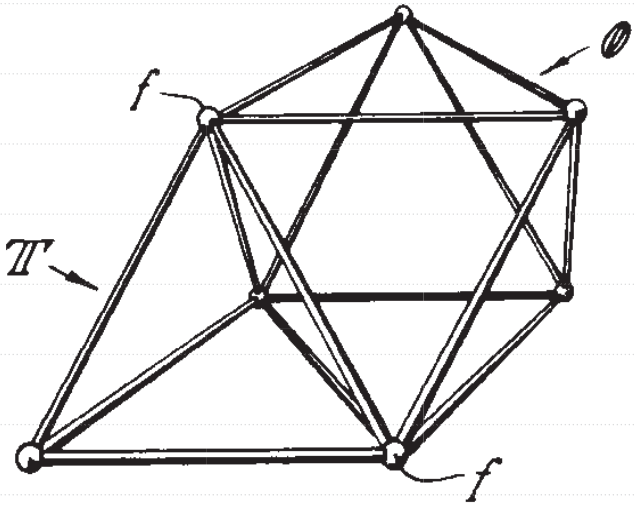
TASK 1

See if you can modify the program by altering the position that the turtles use as the centre of the circle to be the position of the mouse.
Where in the program is this represented?

Look in the help to find out how the position of the mouse is defined.



Moscow/ St Petersburg



Buckminster Fuller's
Octet truss (octahedron/
tetrahedron packing)

Emergent tessellations

Taking the repel procedure from the project we have looked at so far we can look at the more general case where all the turtles are repelled by each other but not attracted to anything. Now we are not trying to draw a circle but saying "if all the turtles are told to back off each others nearest neighbour what will happen?"

The answer is that the agents settle down (depending on the repel strength) into a place where they are equally near to all their nearest neighbours, which as Buckminster Fuller told us 50 years ago is a triangular tessellation, the least energy configuration for regularly tessellating the plane.

TASK 2

How would you work it so that one of the turtles has more "push" than the others? So that instead of the tessellation being evenly triangulated you get a bigger gap somewhere?

Hints

- you can use WHO to select a particular turtle

- varying the repel distance is obviously a good idea

- you might need to check both that any particular turtle is a big pusher, and that for any ordinary turtle it's nearest neighbour is a big pusher as well.

- it takes 2 lines of code so don't go mad