

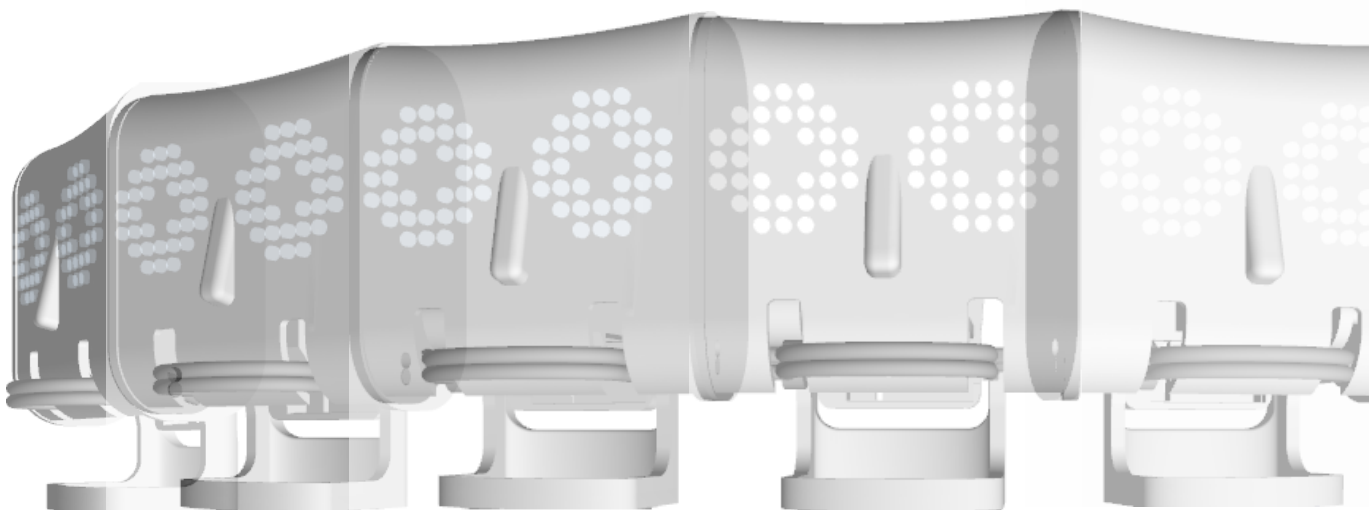
Move like a
human 3

Program your robot
to make life like
movements

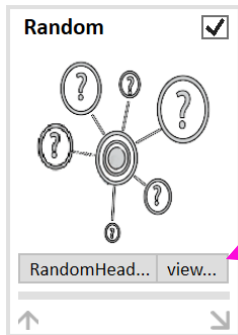
Programming your robot to move like a human is hard!

One of the major challenges is that humans move all the time. If you were to try and program all of the small movements we make then your scripting area would quickly fill up with code.

But if you choose a few different actions (or states), and program those separately, then you can transition between them in a cycle that simulates our constant movement.



The **Random** multiblock tells the robot to make a series of random movements.



Click view on the multiblock to see the code it contains

Making your robot move in this random way brings it to life and provides you with your first state – *awake*. Simply copy the code from the **Random** multiblock and paste it into your scripting area.


If we're going to use *awake* as one of our states then it makes sense to use *asleep* as another.

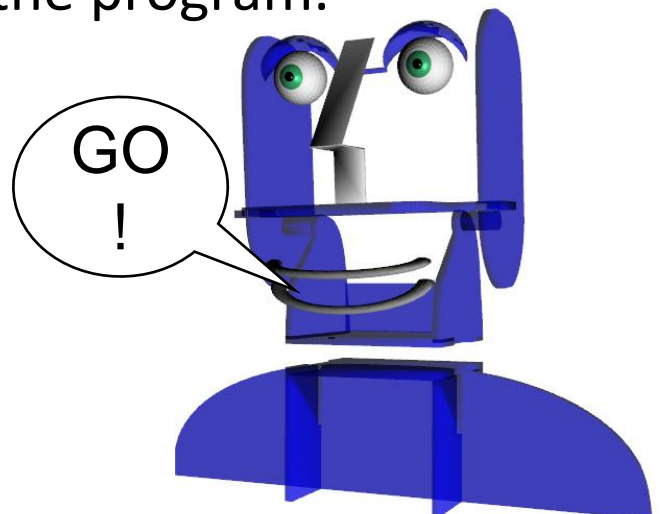
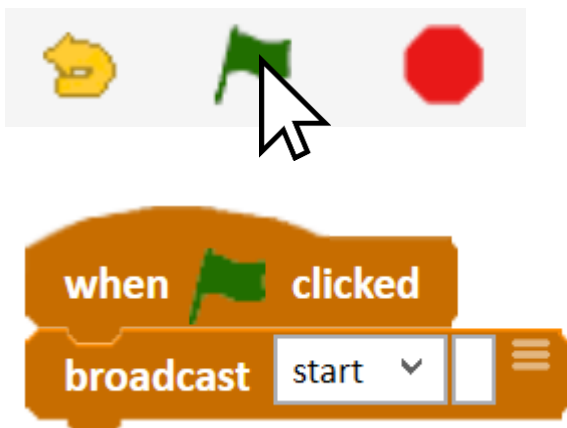
How would you make your robot appear to be asleep? Try writing a short program for the *asleep* state. Use the same scripting area as the *awake* state but start a new chain of blocks.

You should now have two separate states for your robot to transition between. In our example we will be using three states but stick to two if you prefer.

Our third state will be *tired*. If you want to include a third state, arrange your blocks for it now.

Position a **when I receive _** block at the start of each state and a **broadcast _** block at the end.

Drag a **when  clicked** and **broadcast _** block into your scripting area too. These will be used to start the program.



This example program simulates the states of being *awake*, *tired* and *asleep*. The program uses a chain of probability to transition between the different states. When the start message is received, this program will run and output one of the three states.

```
when I receive start
  set HeadTurn to 5
  set HeadNod to 5
  set LidBlink both to 10
  set state to item random of options
  if state = tired then
    broadcast tired
  if state = awake then
    broadcast awake
  if state = asleep then
    broadcast asleep
```

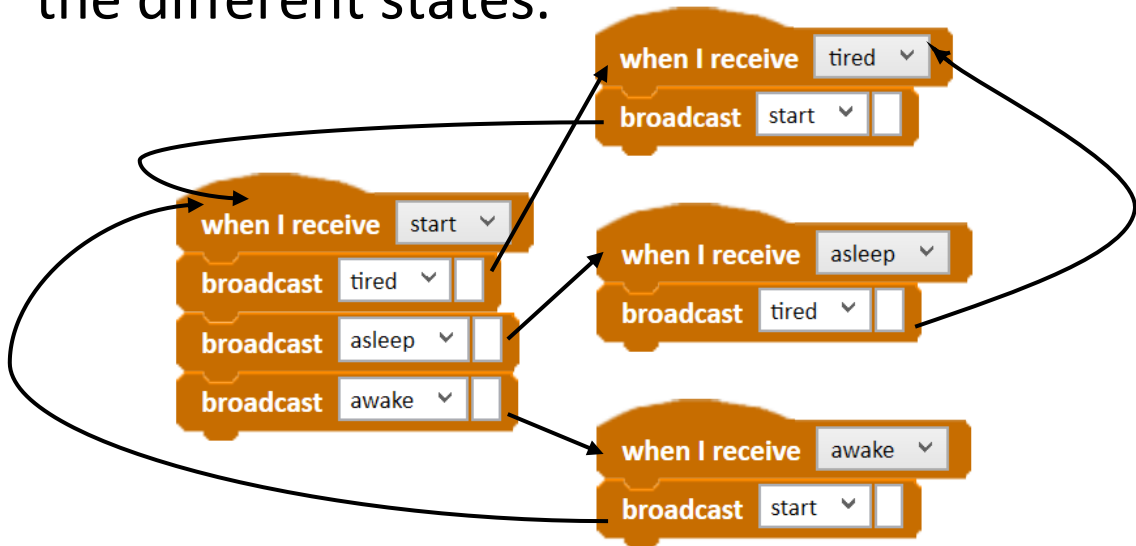
These motion blocks reset the robot between states

Use an add_to_list block to create your options list

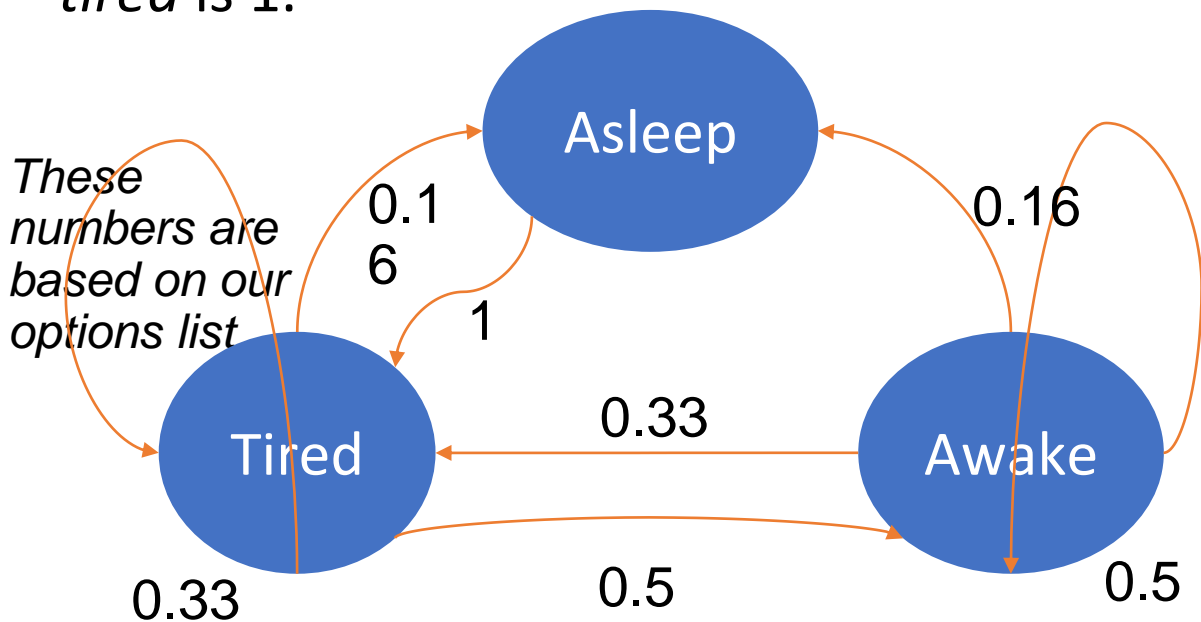
```
options: tired|awake|asleep|awake|awake|tired
```

The number of times the state appears in the options list affects the probability of it being chosen by random.

We have used **broadcast** blocks to trigger the different states.



At the end of the *asleep* state we have chosen to broadcast the *tired* state rather than the Start message. This means that the probability of transitioning from *asleep* to *tired* is 1.



Using the techniques we've just explored, try making some other movement states to transition between and alter the options list to change the probabilities.

