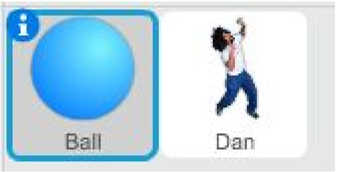
When computers learn by trial and error, we call it 'artificial intelligence', or 'machine learning'. Just this week, a computer used this method and beat a human Othello champion using strategies the human player had never discovered.

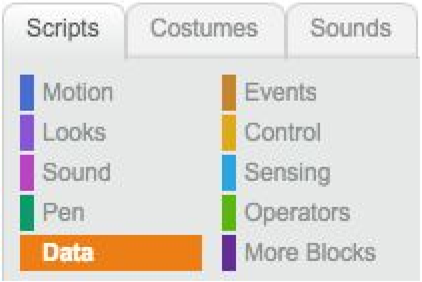
In our Scratch program, the computer will get better at playing dodge ball. The program will randomly select angles to throw the dodgeball at the person. When the ball fails to hit the person, the program will remember that angle by storing it in a list, and never use it again. In time, the program will only have good angles to choose from.

Getting Started

1.Right-click to delete the cat Sprite. Create 2 new Sprites. Choose 'Ball' and 'Dan' from the Library.



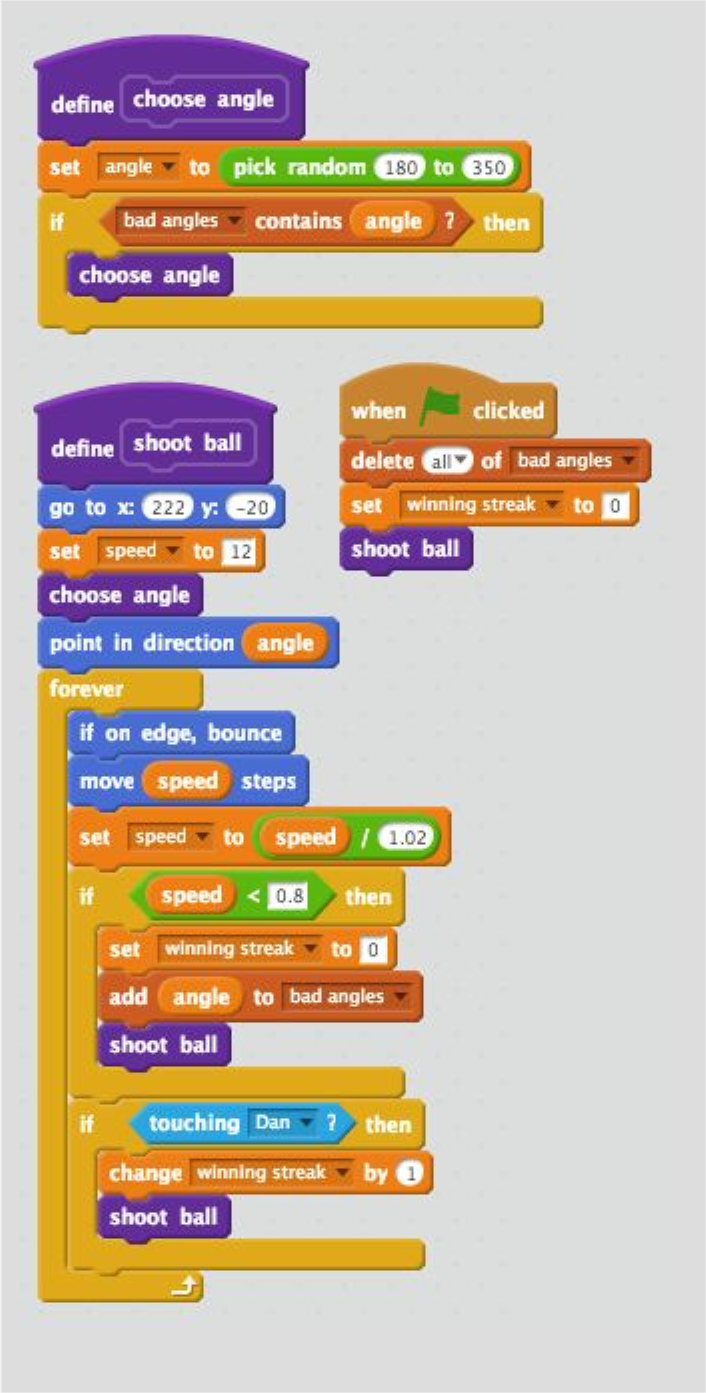
2. Make sure 'Ball' is selected (blue rectangle). Under the 'Scripts' tab, select the 'Data' button. Make variables for 'speed', 'angle', and 'winning streak'. Make a list named 'bad angles'. Uncheck (to hide) all but the 'winning streak' variable'.



3. Select the 🟪 More Blocks button. Make blocks called 'chose angle', and 'shoot ball'. These blocks store lists of instructions just like JavaScript functions do. Like functions, the instructions are not run when they are defined, only when they are called.

4. Drag the blocks into place for your scripts. You can find each block you will need by clicking the button associated with the same color. In come cases, you may need to change a value on the block by clicking the little rectangle or changing a number.

PUTTING IT ALL TOGETHER

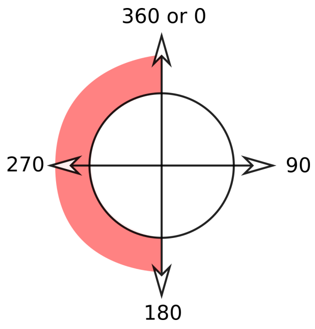


```
define choose angle
set angle to pick random 180 to 350
if bad angles contains angle ? then
    choose angle
```

```
define shoot ball
go to x: 222 y: -20
set speed to 12
choose angle
point in direction angle
forever
    if on edge, bounce
    move speed steps
    set speed to speed / 1.02
    if speed < 0.8 then
        set winning streak to 0
        add angle to bad angles
        shoot ball
    if touching Dan ? then
        change winning streak by 1
        shoot ball
```

```
when ⚑ clicked
delete all of bad angles
set winning streak to 0
shoot ball
```

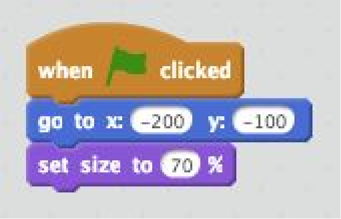
The choose angle block is where the learning happens. Every time it is called, it randomly selects an angle of direction between 180° and 360°, and remembers it with the variable 'angle'. Before it's done it checks our bad angles list to make sure the angle it's picked is not one that has caused the ball to miss hitting the player in the past. If the angle is on the list, the block calls itself, trying again until it finds an angle not on the list. Eventually all the bad angles will be collected by our program and it will become impossible to miss a shot!



Scratch Directional Angles

The shoot ball block points the ball in the direction of angle at the force of our speed variable. If you study our code, the speed starts out at 12 but gets divided by 1.02 every time the screen redraws. In other words, just like in real life, the ball loses speed as it travels. The two 'if' statements check to see if the ball has nearly stopped moving or if it is touching the guy. If the ball stops moving without touching the guy, we save the bad angle to our list and it won't be tried again.

Finally, select our 'Dan' sprite. This code will shrink him down to size and move him where we want him - in the bottom left. Click the flag to start!



```
when ⚑ clicked
go to x: -200 y: -100
set size to 70 %
```

Watch the winning streak variable. For the first few minutes it will be unlikely to go above 1 or 2 successful hits in a row. After several minutes you will notice the computer is getting better. Run it for long enough and it will reach perfection!