

## Contents

Scratch Advanced: “Tick” technique and Clones .....	1
The tick-technique! .....	1
Part 1: The Game Time Loop.....	1
Part 2: The setup.....	2
Part 3: The sprites react to each game tick .....	2
The Spinning Shape Illusion .....	2
The Amazing virtual 3D Printer .....	5

## Scratch Advanced: “Tick” technique and Clones

This week at CDAtHenry scratch advanced we looked at a handy technique to use when working with clones. I call it the “tick” technique, but am not sure if it’s got a real name!!

This method is an efficient and easy way to coordinate the movement of several sprites, and is a handy way to keep things organized in games with plenty of sprites.

We created a few different effects and animations using this technique as applied to clones.

### The tick-technique!

There are 3 parts to the technique

1. Create a game time loop
2. Initialize each sprite when things start
3. Write some code to run every time the clock ticks

I’ll explain these in detail below.

### Part 1: The Game Time Loop

The game loop runs in one sprite, or in the stage. It is a simple loop that broadcasts a message constantly and allows time to tick up. An example of this loop is below:

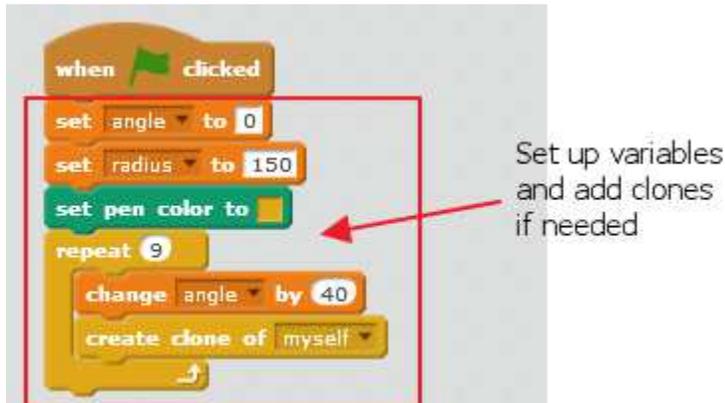


This is a central clock that all the sprites react to.

## Part 2: The setup

The setup phase is where you set the sprites up as you want them. If the setup is taking too long for your taste, you might want to put it into a block called “setup” and select run without screen refresh.

We saw that if you are creating variables “for this sprite only” they need to be different for each sprite



## Part 3: The sprites react to each game tick

Each time the clock “ticks”, the sprites decide where they are supposed to be or what they are doing. What you want to do here is up to you!



Put the cool stuff here..!

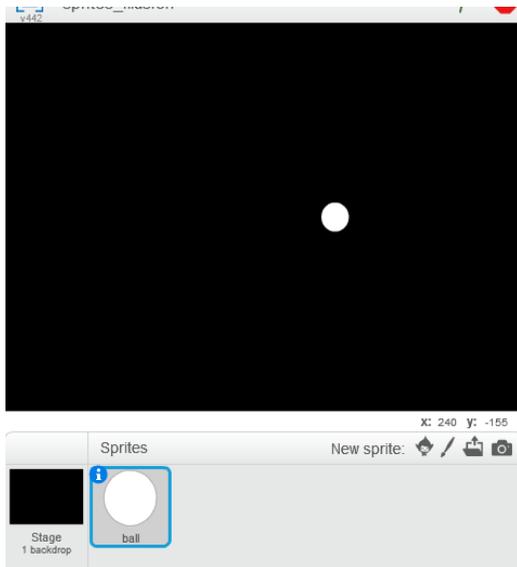
Doing things this way means that each sprite calculates its position at exactly the same time. This can make things a lot easier to manage when it comes to figuring out if they are touching or whatever.

Okay.. that’s the technique.. on to a couple of examples of using it!

## The Spinning Shape Illusion

This shows the use of the tick technique for moving several shapes together, and how clones can be used to do it with very little code.

1. We set the background to all black and created a sprite that is a simple ball shape, as shown below:

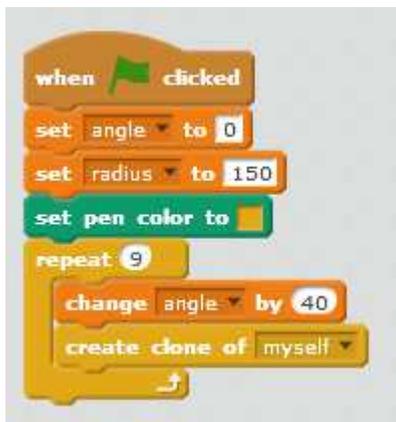


2. Start off by creating the tick-technique game timer loop:

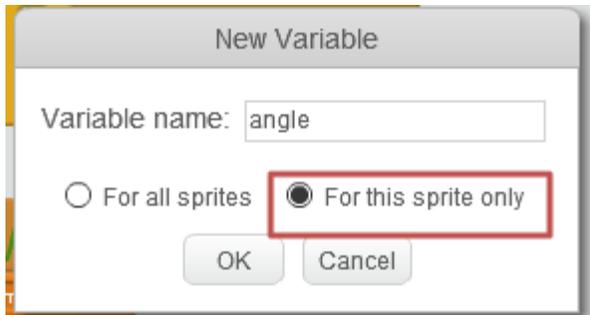


3. Next, set up the sprites. We decided to use 9 balls evenly distributed in a circle. We revised our circles and figures that this means we need a ball every 40 degrees. This comes from:

$360 \text{ degrees in a full circle} / 9 \text{ balls} = 40 \text{ degrees}$

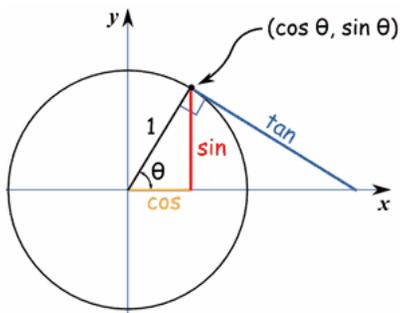


The trick here is to make sure that the variable "angle" is set "for this sprite only"



Now each of the 9 clones has a different “angle” variable, one for every 40 degrees in the circle.

4. In the first version we just drew the balls in a circle, with no animation. To get the coordinates we remembered the unit circle and what the Sin and Cos functions do:

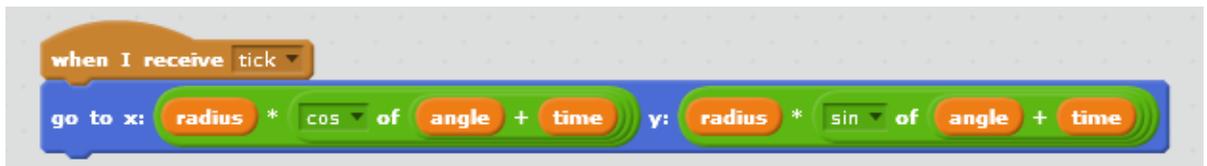


We remembered that the radius \* sin(angle) will give the Y value and radius\*cos(angle) will give the x-value, if you want x and y to be on the edge of a circle.



This spreads the balls evenly around in a circle. But no movement yet!

5. To animate it, change this block to increase the angle over time – the easiest way is by adding time to the angle! Change the tick block as shown.



This way, the balls rotate around in a circle.

6. We played with some cool patterns by changing the “sin” and “cos” formulas to others in the operators area, or even removing them. We go some 3d effects by only having time on

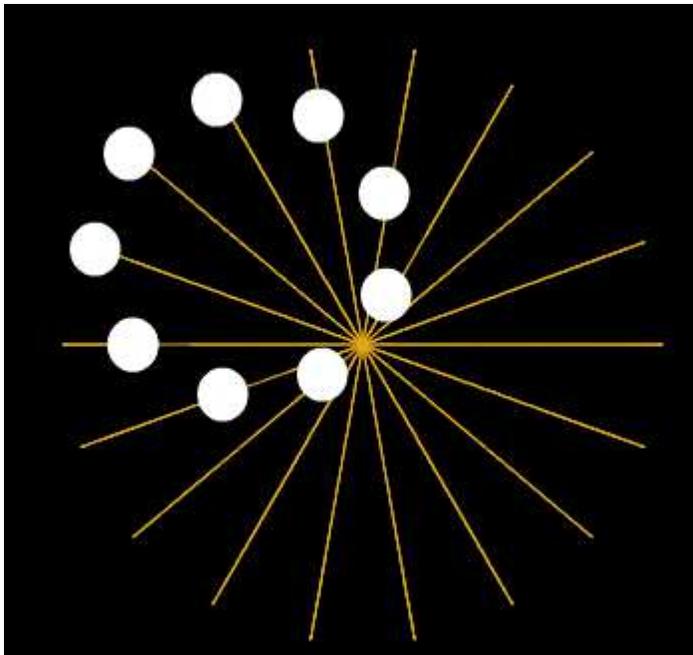
one side, and some neat flying effects by changing to use “tan” in places.

7. Next, we did the amazing rotating ball illusion! We did this by changing the formula to the following one:

```
when I receive tick
go to x: radius * sin of time + angle * cos of angle y: radius * sin of time + angle * sin of angle
if key space pressed? then
  pen down
else
  pen up
  clear
```

Note we also added a little code to let us draw the path the globes are following by pressing “space”.

The neat thing here was that even though the balls are spinning around in a circle, which itself is spinning around.. surprisingly everything is moving in straight lines. Weird that you can see so much circular motion with just straight line movement!



Again – we played around with using different values, formulas and even increasing decreasing radius’s and created some brilliant movement patterns that we could use in games for explosions, bad guys and lots of other things.

## The Amazing virtual 3D Printer

We next took another look at using the “tick” technique to build a 3D printer in scratch.

We chatted for a few minutes about 3D and figured out a few things that make things look 3D:

1. More distant objects are smaller
2. More distant objects tend to move to the center of the screen (the vanishing point)
3. More distant objects are more faded in colour

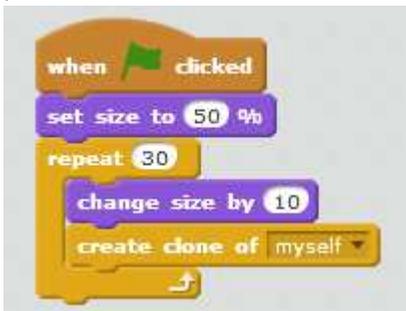
We also explained why this is by talking about how a camera works, and how we really see the world upside-down, but the brain makes it look the right way up – wowsers!

The Amazing virtual 3D printer is a technique to make 3D objects by building up sprites in layers, just like a 3D printer does!

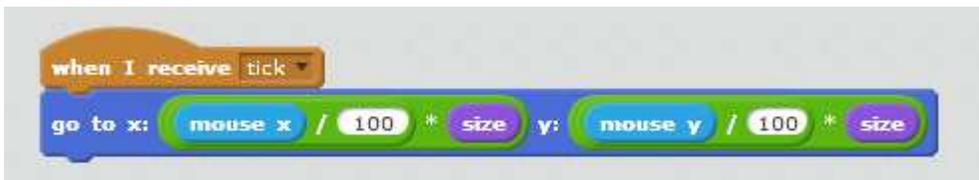
1. Start a normal scratch program – we just used the cat
2. The Timer Game Loop: Create the “tick” technique Game Timer Loop just as before



3. The Setup: Build up the 3D image by creating many clones of cats, each a bit bigger than the previous:



4. Set things moving! Each time the timer ticks, we move the images toward the mouse pointer. The trick here is that the bigger (and hence the close) the image is, then the more we move it. This simple trick has the effect of making the more distant things be closer to the “vanishing point”.



Play around with colors, effects, changing costumes, etc to see what you can produce here! You can do awesome 3D effects!

