Random Numbers

When a programmer asks for a a series of random numbers, he typically doesn't want *truly* random numbers. What he generally wants is numbers that are evenly distributed over a given range, supplied in an unexpected sequence.

Plain English's random number generator is a hybrid of English and Intel x86 machine code (for speed). It uses a "seed" that is initialized to the system's tickcount when a program starts running. The seed can be set by the programmer to a specific value if he wants the same sequence of numbers to be generated with each run. This is the routine:

```
To pick a random number between a min number and a max number:
Put the seed's whereabouts into eax.
Intel $8BC8. \ mov ecx,eax \ calculate zero based max
Intel $8B8510000000. \ mov eax,[ebp+16]
Intel $8B00. \ mov eax,[eax] \ dereference
Intel $8B9D0C000000. \ mov ebx,[ebp+12]
Intel $2B03. \ sub eax,[ebx]
Intel $40. \ inc eax \ adjust randseed
Intel $691105840808. \ imul edx,[ecx],134775813
Intel $42. \ inc edx
Intel $8911. \ mov [ecx],edx
Intel $F7E2. \ mul edx
Intel $0313. \ add edx,[ebx] the min
Intel $8B9D08000000. \ mov ebx,[ebp+08]
Intel $8913. \ mov [ebx],edx
Put the random number into the context's number.
```

Everything following a backslash to the end of the line is a comment. Comments appear in a different color than "code" in our editor so they can be easily ignored. (The colors in this article are reversed.) The algorithm is essentially the same as the *linear congruential generator*that was used in Borland's Turbo Pascal.

Now here is a program that illustrates exactly how uniform a distribution that routine generates:

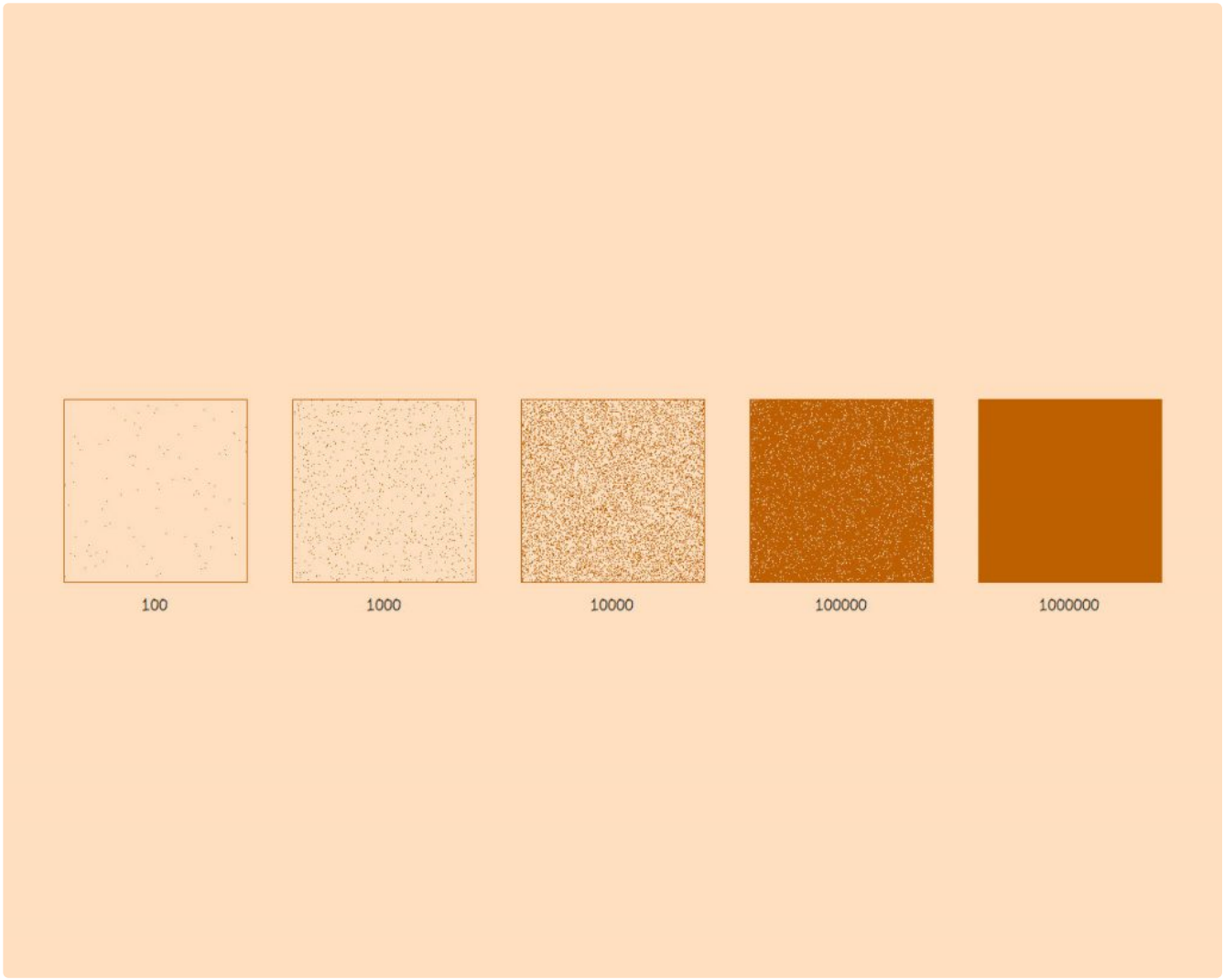
```
To run:
Start up.
Clear the screen using the tan color.
Make a box 2 inches by 2 inches.
Center the box on the screen. Move the box left 5 inches.
Fill and label the box with random spots stopping at 100.
Move the box right 2-1/2 inches.
Fill and label the box with random spots stopping at 1000.
```

```
Move the box right 2-1/2 inches.
Fill and label the box with random spots stopping at 10000.
Move the box right 2-1/2 inches.
Fill and label the box with random spots stopping at 100000.
Move the box right 2-1/2 inches.
Fill and label the box with random spots stopping at 1000000.
Refresh the screen.
Wait for the escape key.
Shut down.

To fill and label a box with random spots stopping at a number:
Privatize the number.
Draw the box with the brown pen.
Loop.
Pick a spot anywhere in the box.
Draw the spot with the brown pen.
If a counter is past the number, break.
Repeat.
Write the number under the box.
Refresh the screen.

To pick a spot anywhere in a box:
Pick the spot's x between the box's left and the box's right.
Pick the spot's y between the box's top and the box's bottom.
```

And here is what we get when we run it:



You can see that the distribution is reasonably uniform in each case. Plain English uses a 96 pixel-per-inch resolution, so a 2-inch square contains 36,864 spots. It takes many more iterations than that to completely fill the box, however, because the random number generator sometimes returns a number more than once. It returns fewer duplicates, of course, when the range (the size of the box, in this case) is larger. Just 10,000,000 iterations, for example, was enough to almost completely fill my entire, 1,310,720-pixel screen.