

Solamente Gráficos

1987 Seymour Papert

Algunas personas ven Logo como "solamente gráficos. Creen que para programar en serio, uno debería aprender BASIC o PASCAL. Otros recuerdan que hay una parte de Logo -que no es solamente gráficos - existe algo llamado procesamiento de listas. Entonces, primero uno hace gráficos como la parte introductoria de Logo, y luego pasa a otra cosa: programación sería con listas.

En este artículo les voy a mostrar algunas cosas que he disfrutado haciendo "solamente gráficos".

1. Fuera del Ascensor ...

Una puerta de ascensor se abre y sale un hombrecito. (Por supuesto, esto es solamente gráficos). Hay tres figuras en el programa: la puerta abierta, la puerta cerrada y el hombre. Pero hace falta pulir un poco esta animación; la puerta se abre muy bruscamente. Debemos mejorarla un poco haciendo otra figura: la puerta medio abierta.



Figuras para el elevador.

```
para elevador
  rg
  dile :puerta
  et
  sp
  fpos :ascensor
  ffig :puerta.cerrada mt
  dile :hombre
  et
  sp
  fpos :ascensor
  ffig :forma.hombre frumbo 90
  aparezca
fin
```

```
para aparezca
  dile :puerta
  ábrase
  dile :hombre
  mt
  espera 15
  camine
fin
```

```
para ábrase
  ffig :puerta.cerrada
  espera 15
  ffig :abriéndose
  espera 15
  ffig :puerta.abierta
fin
```

Entonces la puerta se abre, el hombre aparece, y camina hacia la derecha en la pantalla. Qué tan lejos camina? Bueno, las personas no son totalmente predecibles; caminan hacia la derecha en la pantalla. Qué tan lejos camina? Bueno, las personas no son totalmente predecibles; caminan diferentes distancias. Algunos caminan diez pasos, otros ocho, algunos nueve, algunos seis, y así sucesivamente. Una manera de decidir cuántos pasos da cada uno es tirando un dado -que es muy fácil de hacer en Logo con **azar**.

```

para tire.dado
  reporta 1 + azar 6
fin

```

```

para camine
  ad 20 * tire.dado
  cp
  estampa
fin

```

Entonces el primer hombre aparece y camina *ad 20 * tire.dado*. Ahora, el próximo hombre sale de la puerta y también camina *ad 20 * tire.dado*.



Figura: Hombre saliendo del elevador

```

para salir
  dile :puerta
  ábrase
  dile :hombre
  mt
  espera 15
  camine
  sp
  et
  fpos :ascensor
  salir
fin

```

en vez de, *aparezca* en *elevador*.

Después de un rato, uno de los hombres llegará a un lugar que ya está ocupado. ¿Qué vamos a hacer al respecto? ¿Cómo vamos a ver dónde cada hombre se detuvo?

Hay varias maneras de hacer esto con listas - pero estamos haciendo "solamente gráficos", entonces queremos mantenerlo simple y elemental. El hombre tiene que ver si otro hombre ya está ahí. Si el espacio está ocupado, simplemente sube y se para en la cabeza del otro hombre. Entonces usa *colordebajo* para ver si alguien está ahí: *si colordebajo = 7 [iz 90 ad 10]*.

```

para suba
  si colordebajo = cfondo
    [cp
     estampa
     alto]
  ad 18
  suba
fin

```

Y el *iz 90* tiene que llevarse a cabo antes de que el hombre suba.

```

para caminar
  frumbo 90
  ad 20 * tire.dado
  iz 90
  suba
fin

```

en vez de *camine* en *salir*.

Ahora *salir*, muestra el lugar en donde cada hombre se detiene, y vemos diferentes alturas en cada lugar.

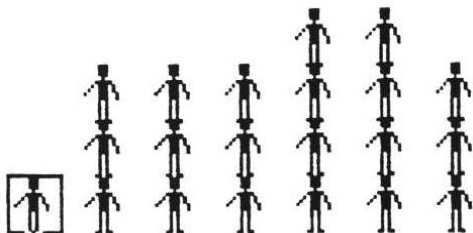


Figura: Hombre subiendo después de emerger del elevador

Si jugamos con esto por un rato, nos damos cuenta de que tal vez la gente verdaderamente no se comporte de esta manera. Esto es, ellos no decidirán de antemano qué tan lejos van a llegar. En cambio, pueden decidir después de cada paso si se detienen o continúan. ¿Hará esto alguna diferencia? Suponga que en lugar de ir hacia adelante *tire.dado* uno va *adelante 20* y entonces tiene un procedimiento *decide*.

```
para decide
  si 1 = azar 5
    [iz 90
     suba
     alto]
  espera 15
  ad 20
  decide
fin
```

en vez de *caminar en salir*.

```
para salir
  frumbo 90
  ad 20
  decide
fin
```

Nuevamente, la gente caminará diferentes distancias. ¿Qué obtenemos esta vez? Hagamos los gráficos (después de todo, estamos interesados en hacer gráficos) y ver qué patrón obtenemos.

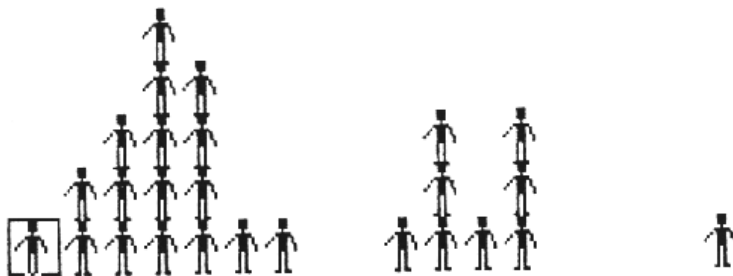


Figura: Hombre emerge y decide

Esta vez, el patrón es muy diferente. El primero era un tanto plano. (uno puede tener que caminar muchos homrecitos antes de que el patrón se torne claro). El nuevo hace una clase de curva muy diferente.

¿Podemos pensar en otra manera de decidir qué tan lejos llegan los homrecitos? Podríamos utilizar más de un dado posiblemente 2 dados de seis lados o 3 dados de cuatro-lados.

```
para cualquier.dado :caras
  reporta 1 + azar :caras
fin
```

```
para dados :número :caras
  si :número = 1
```

```

[reporta cualquier.dado :caras]
reporta (cualquier.dado :caras) + (dados :número - 1 :caras)
fin

```

Ahora podamos usar tantos dados como queramos en *caminar.lejos*.

```

para caminar.lejos
  frumbo 90
  ad 12 * dados 3 6
  iz 90
  suba
fin

```

Podemos hacer los hombrecitos más pequeños para poder ver más de ellos. Hasta podemos estrujarlos tan planos que todo lo que vemos son sus cabezas y pies. Aquí hay 150 hombrecitos tirando varias clases de dados cuando salen del ascensor.

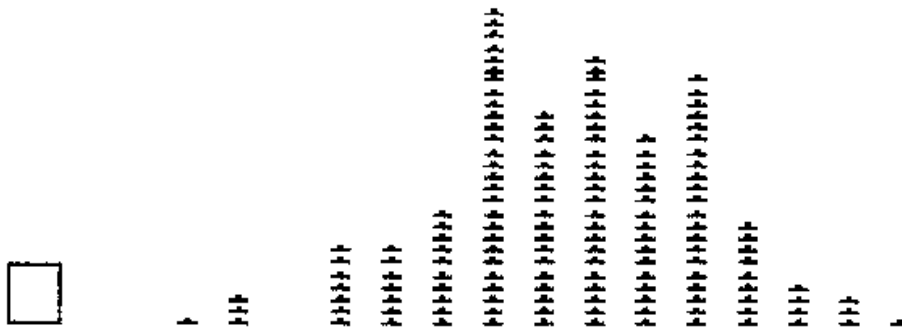


Figura: Tirada de dados, tres hombres aplastados, para *caminar.lejos*

Y aquí hay 100 hombres decidiendo aleatoriamente en cada momento, si detenerse o continuar.

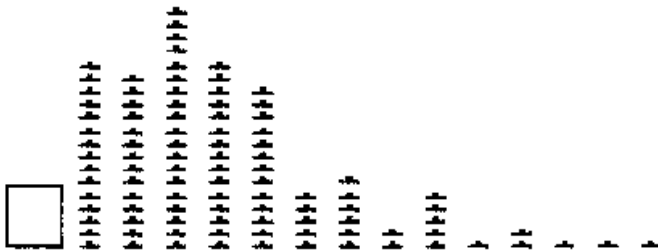


Figura: Hombres aplastados *emergen y deciden*

Ahora, algunas personas podrían no estar de acuerdo, y decir que esto no es hacer gráficos del todo. Pueden decir que es la teoría de probabilidad o estadística -- pero eso no es cierto. No estoy haciendo estadística, estoy haciendo *solamente gráficos*.