The Maze Game

Mini Project 6: Randomize maze attributes

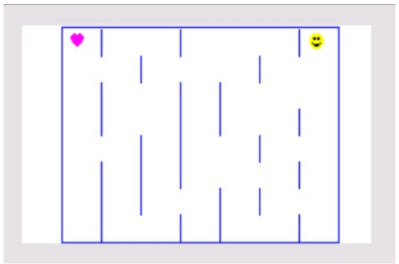
In this sixth mini-project you'll learn how to randomly set doors open and closed in your maze. You'll import the code from mini- project 5 and make a few adjustments.

Objectives:

- Use the randInt() feature to randomly generate numbers
- Learn how to sum the numbers in a List
- Use an IF statement to make decisions.

The MAZE Game Project Overview:

After completing a series of 7 mini-projects, you will have a maze game similar to the one on the right. Projects 1 and 2 will provide skills needed to code movement in the maze game. Projects 3-6 will create code you'll import and use into your final project.



Mini-Project Order:

1. Detect which keys are pressed
2. Use key presses to move string
3. Draw objects using pixels
4. Move objects using keypresses and variables
5. Create a specific Maze
6. **Randomize maze attributes**
7. Create the final maze project.

Teacher Tip:
This file will be imported to project 7.

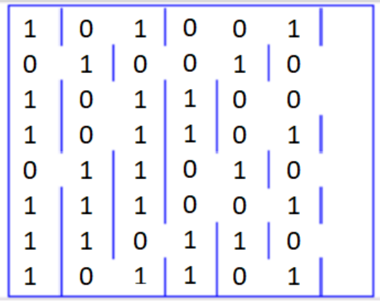
1. You can skip project 6 and go straight to 7, but then your game will be the same each time you open it. In this section, you will learn how to use the randInt() function to randomly create 1s and 0s in the six lists.

   Create a new program named **RMAZE**.

   Code and execute the following

```
:For(A,1,8)
:randInt(0,1) → L1(A)
:Disp L1(A)
:End
```

   Execute your code several times.
   *Approximately what percent of the doors will be open and closed?
   *How will that effect your game?
   *How could you modify the code to change the proportion of 1s and 0s?

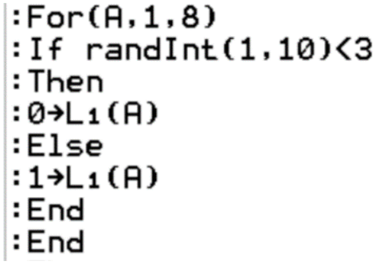


1	0	1	0	0	1
0	1	0	0	1	0
1	0	1	1	0	0
1	0	1	1	0	1
0	1	1	0	1	0
1	1	1	0	0	1
1	1	0	1	1	0
1	0	1	1	0	1

Teacher Tip:
Approximately 50% of the doors should be open and 50% closed. This doesn't make for a very exciting game. Students will need to adjust the proportion.

2. In this step you will modify your existing code. Change the parameters for the random integer function to randInt(1,10). This will generate numbers 1 through 10.

   Modify your code to match the code below. This code will generate a 0 (door) approximately 20% of the time for any given slot.



```
:For(A,1,8)
:If randInt(1,10)<3
:Then
:0→L₁(A)
:Else
:1→L₁(A)
:End
:End
```

```
:ClrDraw
:For(A,1,8)
:If randInt(1,10) < 3
:Then
:0 →L1(A)
:Else
:1 →L1(A)
:End
:End
```

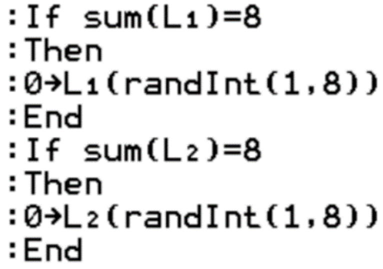
   Create a For loop similar to the one above for each of the 6 Lists.

Teacher Tip:
Students can choose any range of values to generate 1s and 0s. Students could put randInt(1,100) to make more proportions other than those divisible by 10. Students could be challenged to determine algorithms to guarantee there will always be at least two but no more than four doors.

3. If the probability of generating a 1 is 20% for any given element, what is the probability all 8 elements in a row are 1s creating one solid wall without any doors?

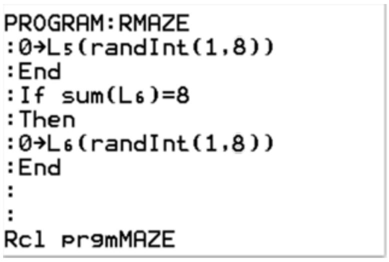
   Since each number is independent of the next, the probability is
   $0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 = 0.8^8 \approx 0.167$.

   Currently, the code will generate a column without any open spaces approximately 16.7% of the time. To fix this problem, we will check to see if the sum of all the items in the list equal 8. If the sum equals 8, all items in the list equal 1. If this happens, code the project to randomly change one element to 0.



```
:If sum(L₁)=8
:Then
:0→L₁(randInt(1,8))
:End
:If sum(L₂)=8
:Then
:0→L₂(randInt(1,8))
:End
```

```
: If sum(L1) = 8
:Then
:0 → L1(randInt(1,8))
:End
```

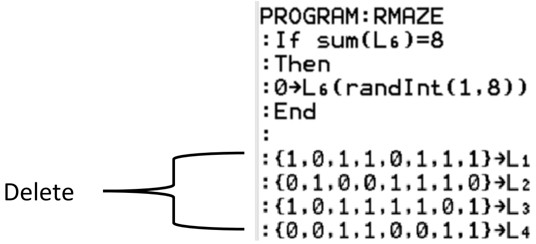


```
PROGRAM:RMAZE
:0→L₅(randInt(1,8))
:End
:If sum(L₆)=8
:Then
:0→L₆(randInt(1,8))
:End
:
:
Rcl prgmMAZE
```

4. Now that the code randomly generates values for the lists, you need to import the drawing code from the basic maze. After the import, delete the 6 lines that create the basic 6 lists from before. (Otherwise, you'll overwrite your random mazes.)

   Use recall to import the code from image you'd like to be the player.

   rcl (2nd sto→)
   prgm
   EXEC
   Select the MAZE file
   Press the enter key



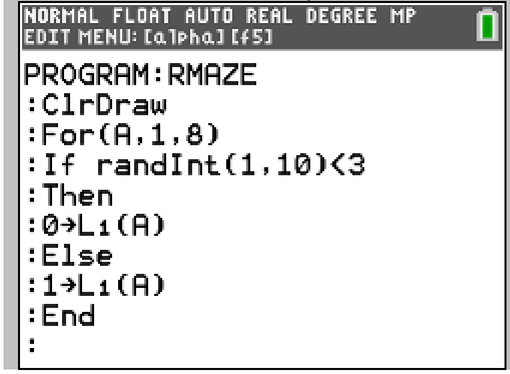
```
PROGRAM:RMAZE
:If sum(L₆)=8
:Then
:0→L₆(randInt(1,8))
:End
:
:{1,0,1,1,0,1,1,1}→L₁
:{0,1,0,0,1,1,1,0}→L₂
:{1,0,1,1,1,1,0,1}→L₃
:{0,0,1,1,0,0,1,1}→L₄
```

5. Execute your code several times.

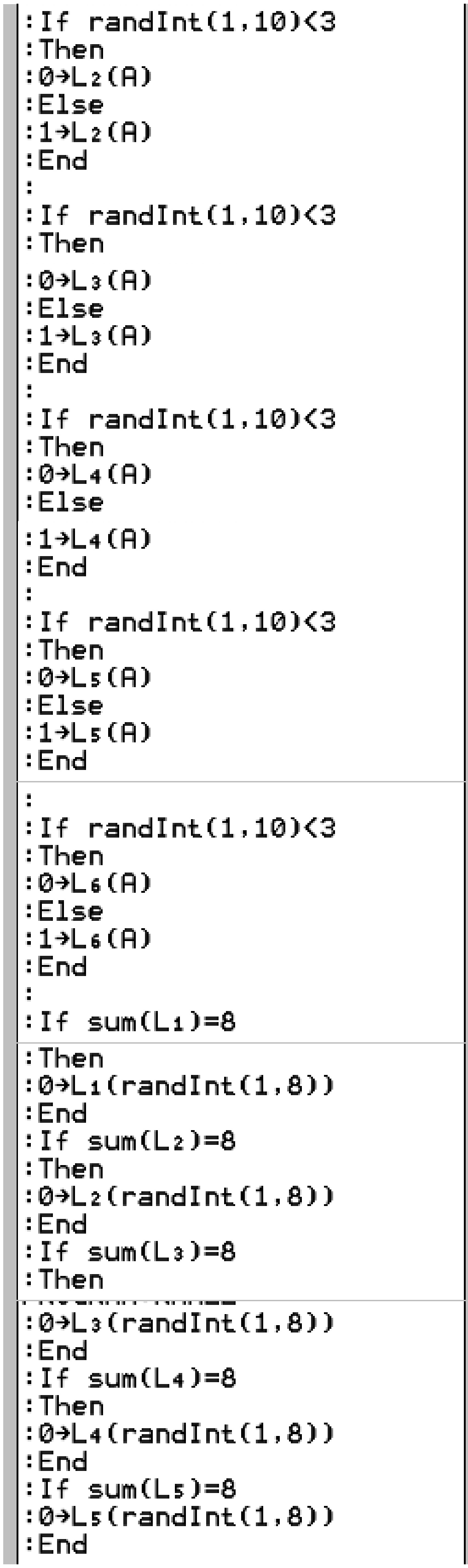
   Is each maze unique?

   Are you satisfied with the proportion of 1s and 0s? If not, how can you change the proportion?

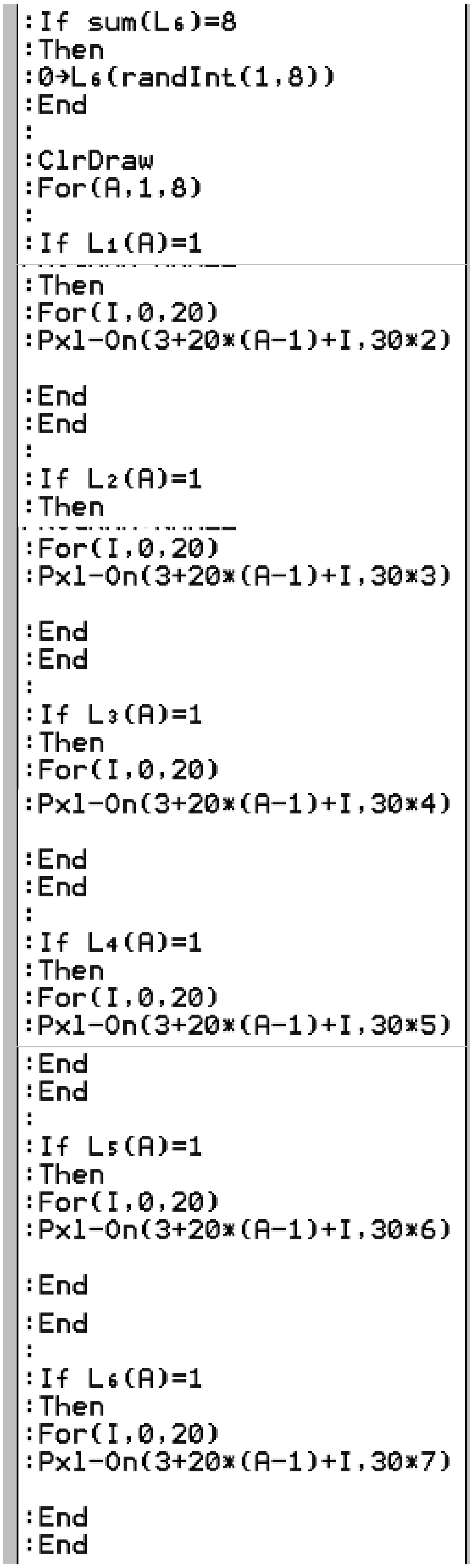
Teacher Tip:
If the mazes are not unique, make sure the student deleted the 6 lines creating L1-L6 from the previous file.



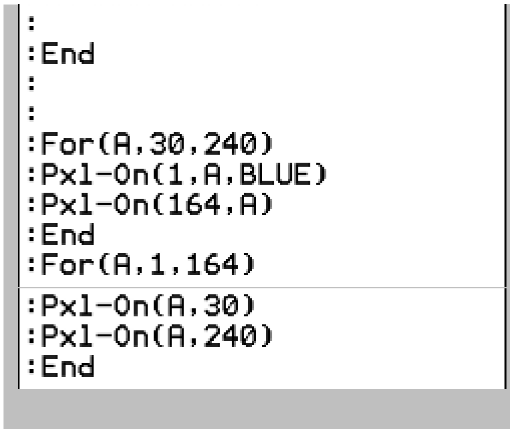
```
PROGRAM:RMAZE
:ClrDraw
:For(A,1,8)
:If randInt(1,10)<3
:Then
:0→L₁(A)
:Else
:1→L₁(A)
:End
:
```



```
:If randInt(1,10)<3
:Then
:0→L₂(A)
:Else
:1→L₂(A)
:End
:
:If randInt(1,10)<3
:Then
:0→L₃(A)
:Else
:1→L₃(A)
:End
:
:If randInt(1,10)<3
:Then
:0→L₄(A)
:Else
:1→L₄(A)
:End
:
:If randInt(1,10)<3
:Then
:0→L₅(A)
:Else
:1→L₅(A)
:End
:
:If randInt(1,10)<3
:Then
:0→L₆(A)
:Else
:1→L₆(A)
:End
:
:If sum(L₁)=8
:Then
:0→L₁(randInt(1,8))
:End
:If sum(L₂)=8
:Then
:0→L₂(randInt(1,8))
:End
:If sum(L₃)=8
:Then
:0→L₃(randInt(1,8))
:End
:If sum(L₄)=8
:Then
:0→L₄(randInt(1,8))
:End
:If sum(L₅)=8
:0→L₅(randInt(1,8))
:End
```



```
:If sum(L₆)=8
:Then
:0→L₆(randInt(1,8))
:End
:
:ClrDraw
:For(A,1,8)
:
:If L₁(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*2)

:End
:End
:
:If L₂(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*3)

:End
:End
:
:If L₃(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*4)

:End
:End
:
:If L₄(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*5)
:End
:End
:
:If L₅(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*6)

:End
:End
:
:If L₆(A)=1
:Then
:For(I,0,20)
:Pxl-On(3+20*(A-1)+I,30*7)

:End
:End
```



```
:
:End
:
:
:For(A,30,240)
:Pxl-On(1,A,BLUE)
:Pxl-On(164,A)
:End
:For(A,1,164)
:Pxl-On(A,30)
:Pxl-On(A,240)
:End
```