The Maze Game — Mini Project 2: Use key presses to move string

In this second mini-project, you will learn how to store your name as a string. You will use the **getKey** command from the first project to move your name on the screen. This project will teach you how to move objects similarly to how you will move your shape for the maze project.

Objectives:

- Use getKey and If statements to make a selection
- Use the Output command to display string
- Use variables to control where to display
- Use a While loop to repeat code

MAZE 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 project teaches coding skills and debugging skills. It is important students experience debugging in a small controlled environment before they try to debug long lines of code. This project shows a common error that occurs when drawing off screen. This bug also happens frequently in the final maze project.

1. In this mini-project you will use key presses to move text on the screen. Your screen is broken into multiple rows that go down the screen and multiple columns that go across the screen.

   Type the code below, insert your name.
   *Str1 can be found under Vars →String.

```
PROGRAM:MOVE
:ClrHome
:
:3→R
:1→C
:"JANE"→Str1
:Output(R,C,Str1)
```

```
:3 →R
:1 →C
:"NAME" →Str1
:Output(R,C,Str1)
```

   Execute your code.
   What row and column contains your first initial?

```
PROGRAM:MOVE
:ClrHome
:0→K
:1→R
:1→C
:"JANE"→Str1
:
:While K≠45
:Output(R,C,Str1)
:getKey→K
```

2. You will now modify your code to get a key press value. You will use this value to change either the row or column variable which will change where your name is displayed.

   Use the variable K to store the value from the key press.

   Use a while loop to continually get key press values until the clear key is pressed.

   You will need 4 If statements, one for each arrow key.

   Do you remember the arrow key numbers from Project 1?
   If not, rerun Project 1 and determine the number for each arrow key.

   What do you think the If statements will look like to control the movement of the string?

Teacher Tip:
Make sure students use the ≠ sign. If students use the = sign the while condition isn't met and the code in the loop won't be executed at all.
Keys: Up- 25, Down- 34, Left- 24, Right- 26

```
:If K=34
:Then
:R+1→R
:ClrHome
:End
```

3. The down key gives value of 34.
   To move the text down the screen, check to see **If** key34 was pressed.
   If it was pressed, increase the row by 1 and clear the old text off the screen

   Add this code inside your loop so it is repeated each time the program loops through and checks for a key press.

```
:If K = 34
:Then
:R+1 →R
:ClrHome
:End
```

```
ERROR: DOMAIN
1:Quit
2:Goto

Value entered is not
  allowed in the function
  or command.
Ex: RADIAN MODE tan(π/2)
    DEGREE MODE e^(π/2*i)
```

   Be careful when you execute the code.
   If **R** the number of rows goes off the screen you will get the following error.

   How times can you press the down arrow before you get an error?
   How many rows of text fit on the screen?

Teacher Tip: It is important students experience this error in project 2 with only a few lines of code. In project 4 students move approximately 100 pixels in a similar fashion but with many more lines of code.

If students select Goto when the error occurs it will take them to the place in the code where the error occurs. This makes it easier to debug. Students often choose Quit by default but then have trouble finding the location to debug.

4. Code the Left, Right and Up arrow keys.

   Check your code with the answers on the next slide

5.

Up	Left	Right
:If K=25 :Then :R-1→R :ClrHome :End	:If K=24 :Then :C-1→C :ClrHome :End	:If K=26 :Then :C+1→C :ClrHome :End

Can you error trap your code so the Domain Error doesn't occur when the user moves the name off the screen?

6. Error Trapping:
   The row and column variables have to be larger than 0.

```
:If R = 0
:Then
:1 →R
:End
:If C= 0
:Then
:1 →C
:End
```

   There are 10 rows.

```
:If R = 11
:Then
:10 →R
:End
```

   There are 26 columns.
   Count the number of letters in your name. Subtract from 27 to determine the maximum value for C.

```
:If C = 24
:Then
:23 →C
:End
```

Teacher Tip:

Project code:

```
PROGRAM:MOVE
:ClrHome
:0→K
:1→R
:1→C
:"JANE"→Str1
:
:While K≠45
:Output(R,C,Str1)
:getKey→K
:
:If K=25
:Then
:R-1→R
:ClrHome
:End
:If K=34
:Then
:R+1→R
:ClrHome
:End
:
:If K=24
:Then
:C-1→C
:ClrHome
:End
:If K=26
:Then
:C+1→C
:ClrHome
:End
:
:If R=0
:Then
:1→R
:End
:If C=0
:Then
:1→C
:End
:If R=11
:Then
:10→R
:End
:If C=24
:Then
:23→C
:End
:End
```