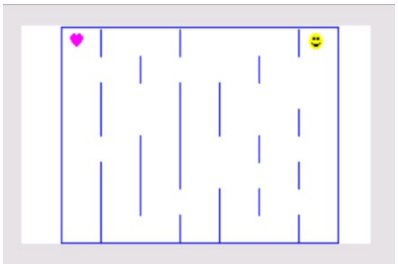
The Maze Game	Mini Project 4: Move objects using keypresses and variables
In this fourth mini-project, you will import your player character into a new file. You'll modify the code so the arrow keys will move each pixel in your character.	**Objectives:** • Use getKey to determine key press event • Use variables to store horizontal and vertical shifts • Use a While loop to repeat code • Use If then statements to make selections • Use the Pxl-On and Pxl-Off command to create characters

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:
Make sure students create a new project. Having a saved copy of the original makes it easier to start over if necessary.

1. Moving your player object is similar to moving your name in Project 2.

   You'll need two variables to shift your picture vertically and horizontally.
   We'll use **V** and **H** to shift our picture. We'll use **K** to represent the button push value.

   We'll work from a copy of your original picture. This will allow you to start over if you need without creating the object from scratch.

   Create a new program named **PLAYER.**
   In this project you will:
   Draw the initial heart
   - Use a loop to detect key press value
   - If an arrow key is pressed
     - Turn all the heart pixels off
     - Redraw the heart in the new location

2. Type the following code

```
:ClrDraw
:0 →K
:0 →V
:0 →H
```

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

rcl (2nd sto→)
prgm
EXEC
Select the name of your file
Press the enter key

```
NORMAL FLOAT AUTO REAL RADIAN MP
EDIT MENU: [α]pha] [f5]
PROGRAM:PLAYER
:ClrDraw
:AxesOff
:0→B
:0→H
:0→V
:
:
:
Rcl prgmHEART
```

Teacher Tip:
Recall imports the code from the requested project. Any adjustments made to the code in the new file are not reflected in the original.

3. Moving your object is similar to moving your name like you did in Project 2.

You will use:
*a while loop around all of your pixel code that executes until the clear button is pressed.
*the **getKey** command to determine when arrow keys are pressed.
*IF statements to change the vertical and horizontal variables **V** and **H**.

A.) Add a WHILE statement at the end of code. The statement should continue while the user doesn't press the clear button.

B.) Use the getKey command to store the keys pressed in the value **K**

C.) Write an If statement that will happen if an arrow key is pushed.

4. Your code should look something similar to the code on the right.

```
:Pxl-On(21,35,MAGENTA)
:Pxl-On(21,36,MAGENTA)
:
:While K≠45
:getKey→K
:If K=24 or K=25 or K=26 o
r K=34
:Then
:
```

Next you'll write the code to hide the old shape.

5. Add a comment line so it will be easier to debug your code.

   :"HIDE OLD

   Then import the heart code

   :rcl prgmHeart

```
:While K≠45
:getKey→K
:If K=24 or K=25 or K=26 o
r K=34
:Then
:"HIDE OLD
:
Rcl prgmHEART
```

   Delete the ClrDraw from the beginning of the heart.
   If you leave the ClrDraw in here, it will delete your maze and your goal when we import this code into the maze game.

```
:While K≠45
:getKey→K
:If K=24 or K=25 or K=26 o
r K=34
:Then
:"HIDE OLD
:ClrDraw
:
:
```

6. You need to modify every pixel line of the form

   Pxl-On(row number, column number, color)

   To

   Pxl-Off(row number **+ V**, column number **+ H**)

   **Remember V is the vertical shift and H is the horizontal shift.*

   This will turn off the old pixel from the old shifted location.

   Add an End statement to the last line to end the If for the key press.

   :End

```
:
:For(C,34,37)
:Pxl-Off(20+V,C+H)
:End
:
:Pxl-Off(21+V,35+H)
:Pxl-Off(21+V,36+H)
:End
:
```

Teacher Tip:
Students need to press 2nd del to insert.

7. Now you'll change the vertical and horizontal shift based on the key pressed.

   If the key is the left arrow key subtract 30 from H.

   If the key is the right arrow key add 30 to H.

   If the key is the up arrow key subtract 20 from V.

   If the key is the down arrow key add 20 to V.

```
:End
:If K=24
:Then
:H-30→H
:End
:If K=26
:Then
:H+30→H
:End
```

```
:If K = 24
:Then
:H-30→H
:End
……
```

Add the code for the right, up and down key

Teacher Tip:

```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha] [f5]
PROGRAM:PLAYER
:If K=24
:Then
:H-30→H
:End
:If K=26
:Then
:H+30→H
:End
:If K=25
:Then
:V-20→V
:End
:If K=34
:Then
:V+20→V
:End
:
```

8. If an arrow key was pressed, you need to redraw the shape after the shift.

   Insert the following:

   :If K=24 or K=25 or K=26 or K=34
   :Then
   :"DRAW NEW

```
:H+30→H
:End
:
:If K=24 or K=25 or K=26 o
r K=34
:Then
:"DRAW NEW
:
Rcl prgmHEART
```

   Now import the heart file one more time.

   Delete the ClrDraw

9. Lastly, you need to modify every pixel line of the form

   Pxl-On(row number, column number, color)
   and
   Pxl-Off(row number, column number)

   Since the variables V and H hold the shift variables, the modifications are

   Pxl-On(row number **+ V**, column number **+ H**, color)
   and
   Pxl-Off(row number **+ V**, column number **+ H**)

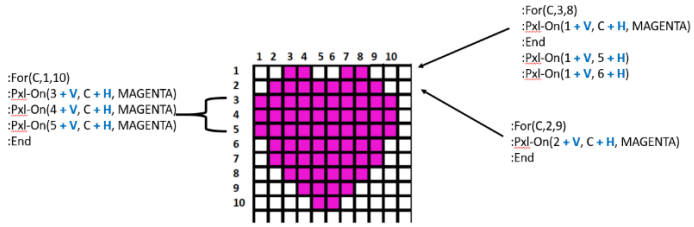
```
:Pxl-On(20+V,C+H,MAGENTA)
:End
:
:Pxl-On(21+V,35+H,MAGENTA)

:Pxl-On(21+V,36+H,MAGENTA)

:End
:End
```

Add an End for the If.

Add an End for the While Loop



Teacher Tip:

Students need to hit 2nd del to insert

Sample Player Code:

```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha] [f5]
PROGRAM:PLAYER
:ClrDraw
:0→K
:0→V
:0→H
:
:ClrDraw
:AxesOff
:
:For(A,36,41)
:Pxl-On(12,A,MAGENTA)
:End
:Pxl-Off(12,38)
:Pxl-Off(12,39)
:
:For(A,35,42)
:Pxl-On(13,A,MAGENTA)
:End
:
:For(R,14,16)
:For(C,34,43)
:Pxl-On(R,C,MAGENTA)
:End
:End
:
:For(R,17,18)
:For(C,35,42)
:Pxl-On(R,C,MAGENTA)
:End
:End
:
:For(C,36,41)
:Pxl-On(19,C,MAGENTA)
:End
:
:For(C,37,40)
:Pxl-On(20,C,MAGENTA)
:End
:
:Pxl-On(21,38,MAGENTA)
:Pxl-On(21,39,MAGENTA)
:
:While K≠45
:getKey→K
:If K=24 or K=25 or K=26 o
r K=34
```

```
:Then
:"HIDE OLD
:
:For(A,36,41)
:Pxl-Off(12+V,A+H)
:End
:
:For(A,35,42)
:Pxl-Off(13+V,A+H)
:End
:
:For(R,14,16)
:For(C,34,43)
:Pxl-Off(R+V,C+H)
:End
:End
:
:For(R,17,18)
:For(C,35,42)
:Pxl-Off(R+V,C+H)
:End
:End
:
:For(C,36,41)
:Pxl-Off(19+V,C+H)
:End
:
:For(C,37,40)
:Pxl-Off(20+V,C+H)
:End
:
:Pxl-Off(21+V,38+H)
:Pxl-Off(21+V,39+H)
:End
:If K=24
:Then
:H-30→H
:End
:If K=26
:Then
:H+30→H
:End
:If K=25
:Then
:V-20→V
:End
:If K=34
:Then
:V+20→V
:End
:
:If K=24 or K=25 or K=26 o
r K=34
:Then
```

```
:"DRAW NEW
:
:For(A,36,41)
:Pxl-On(12+V,A+H,MAGENTA)
:End
:Pxl-Off(12+V,38+H)
:Pxl-Off(12+V,39+H)
:
:For(A,35,42)
:Pxl-On(13+V,A+H,MAGENTA)
:End
:
:For(R,14,16)
:For(C,34,43)
:Pxl-On(R+V,C+H,MAGENTA)
:End
:End
:
:For(R,17,18)
:For(C,35,42)
:Pxl-On(R+V,C+H,MAGENTA)
:End
:End
:
:For(C,36,41)
:Pxl-On(19+V,C+H,MAGENTA)
:End
:
:For(C,37,40)
:Pxl-On(20+V,C+H,MAGENTA)
:End
:
:Pxl-On(21+V,38+H,MAGENTA)

:Pxl-On(21+V,39+H,MAGENTA)


:End
:End
```