826 DIGITAL

Pixel Programming: Packet 3

Advanced Pixel Programming

Here's one more set of commands to learn before taking your Pixel Programs to the next level.

1. **call function [FUNCTION]**

   Whenever you see one of these, find the Pixel Program sheet for the new function, and lay it over your current sheet so that the pixel holes line up. And then, just follow the Pixel Programming commands for the new sheet. When you follow a move instruction, make sure to move the sheets together so their holes stay aligned! (You may find it handy to use a couple of paper clips to keep the stack together.)

   a. **pass variables**

      Sometimes, this will accompany the call function command. When it does, it gives you instructions on setting the variables in the new pixel programming function. So, if following the function call, there are lines like

      ```
      call function COUNTDOWN
      +----------------+
      | pass variables |
      | 4 as TIME      |
      +----------------+
      ```

      you need to set 4 as the value of `TIME` before starting the `COUNTDOWN` function.

      Note that a lot of the time, a variable will be passed, something like

      ```
      call function COUNTDOWN
      +----------------+
      | pass variables |
      | NOW as TIME    |
      +----------------+
      ```

      To do this, you'd check on the value of `NOW` in your current function, and then set that as the value for `TIME` in the function that's being called. So, if `NOW = 11:59` at the top of your original Pixel Program, you would set `11:59` as the value for `TIME` in the `COUNTDOWN` function. (You would not set "`NOW`" as the value of `TIME`; you are passing the variable's value, not its name.)

2. **return to last function**

   You'll see one of these at the end of most functions. When you see it, simply remove the sheet of the function you're working on, and resume where you left off in the function before that.

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Challenges

Getting good with functions opens up a lot of doors in programming. It lets you compartmentalize your thinking, so you can focus on one part of your program at a time.

This lets you write high-level code first using functions, even if those functions don't exist yet—this often comes soon after pseudocoding. When all that looks good, then you can worry about those non-existent functions, and create them.

Let's try creating some useful functions.

First, if your teacher hasn't demonstrated them, make sure to try out the functions `JUMP-RIGHT`, `LINE-RIGHT`, and `LINE-UP-RIGHT`.

Challenge 1
Wouldn't it be useful if there were other jump and line functions in other directions? Try to create a few, e.g. `JUMP-LEFT`, `JUMP-UP`, `LINE-DOWN`, `LINE-UP-LEFT`, etc.

Challenge 2
You know what would be even more convenient? What if there was a jump function, or a line function that could go any direction you told it to? We actually have created versions of `JUMP`, `JUMP-DIAGONAL`, and `STRAIGHT-LINE`, but you should try to make your own.

Hint 1: What things would be common between `JUMP-RIGHT` and `JUMP-LEFT`? What things would be different? What might your code use to account for the differences?

Hint 2: Variables.

Hint 3: If you're really stuck, take a look at one of our versions of `JUMP`, `JUMP-DIAGONAL`, and `STRAIGHT-LINE`. Only look at one! Based on that, try and create the other two, or the function `DIAGONAL-LINE`.

Challenge 3
Create a function that calls other functions! Look at `SQUARE-V1`, `SQUARE-V2`, and/or `SQUARE-V3` for inspiration. (V1 uses four very similar functions; V2 uses one function four different ways. V3 is actually just an alternate way of writing V2, a shortcut that looks more like actual computer code.)

The great thing is that once you've created a useful function, it simplifies how you create your code. A lot of the first examples of Pixel Programs you did would be easier to read, write, and execute if they used functions.

Can you create other shapes? Can you create letters, or even write out your name? Can you recreate those earlier Pixel Programs, but with the much cleaner language of functions?

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Function **JUMP-RIGHT**

==========

☐ current pixel

list of variables:

JUMPDIST == _____

==========

repeat JUMPDIST times

| move RIGHT

---

return to last function

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Function **LINE-RIGHT**

```
=====================================================================
                                        list of variables:
   current pixel                        COLOR == _______________
                                        LINELEN == _______________
=====================================================================
```

```
set LINELINE to
     (LINELEN-1)

MARK

repeat LINELINE times
----
| move RIGHT
| MARK
----

return to last function
```

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Function **LINE-UP-RIGHT**

```
=====================================================================
                                          list of variables:
   current pixel                          COLOR == _______________
                                          LINELEN == _____________
=====================================================================

set LINELEN to
  (LINELEN-1)

MARK

repeat LINELEN times
----
| move UP
| move RIGHT
| MARK
----

return to last function
```

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Function **STRAIGHT-JUMP**

```
=====================================================================
                                        list of variables:
   current pixel                        JUMPDIST == _______________
                                        JUMPDIR == _______________
=====================================================================

repeat JUMPDIST times
----
| move JUMPDIR
----

return to last function
```

Pixel Programming: Packet 3

Pixel Programming Function **DIAGONAL-JUMP**

```
=====================================================================

                                        list of variables:
   current pixel                        JUMPDIST == _______________
                                        HORIZDIR == _______________
                                        VERTDIR == _______________
=====================================================================

repeat JUMPDIST times
----
| move HORIZDIR
| move VERTDIR
----

return to last function
```

826 DIGITAL

Pixel Programming: Packet 3

Pixel Programming Function **STRAIGHT-LINE**

```
=====================================================================

                                        list of variables:
   current pixel                        COLOR == _______________
                                        LINELEN == _______________
                                        LINEDIR == _______________

=====================================================================

set LINELEN to
  (LINELEN-1)

MARK

repeat LINELEN times
----
| move LINEDIR
| MARK
----

return to last function
```

Pixel Programming: Packet 3

Pixel Programming Function **SQUARE-V1**

```
=========================================================================
                                          list of variables:
   current pixel                          SQCOLOR == _______________
                                          SQSIZE == _______________
=========================================================================
```

```
call function
  LINE-RIGHT
+---------------------+
| pass variables      |
| SQCOLOR as COLOR    |
| SQSIZE as LINELEN   |
+---------------------+

call function
  LINE-DOWN
+---------------------+
| pass variables      |
| SQCOLOR as COLOR    |
| SQSIZE as LINELEN   |
+---------------------+

call function
  LINE-LEFT
+---------------------+
| pass variables      |
| SQCOLOR as COLOR    |
| SQSIZE as LINELEN   |
+---------------------+

call function
  LINE-LEFT
+---------------------+
| pass variables      |
| SQCOLOR as COLOR    |
| SQSIZE as LINELEN   |
+---------------------+
```

Pixel Programming: Packet 3

Pixel Programming Function **SQUARE-V2**

```
=========================================================================
                                          list of variables:
   ☐ current pixel                        SQCOLOR == _______________
                                          SQSIZE == _______________
=========================================================================
```

```
call function STRAIGHT-LINE
+----------------------+
| pass variables       |
| SQCOLOR as COLOR     |
| SQSIZE as LINELEN    |
| RIGHT as LINEDIR     |
+----------------------+

call function STRAIGHT-LINE
+----------------------+
| pass variables       |
| SQCOLOR as COLOR     |
| SQSIZE as LINELEN    |
| DOWN as LINEDIR      |
+----------------------+

call function STRAIGHT-LINE
+----------------------+
| pass variables       |
| SQCOLOR as COLOR     |
| SQSIZE as LINELEN    |
| LEFT as LINEDIR      |
+----------------------+

call function STRAIGHT-LINE
+----------------------+
| pass variables       |
| SQCOLOR as COLOR     |
| SQSIZE as LINELEN    |
| UP as LINEDIR        |
+----------------------+
```

Pixel Programming: Packet 3

Pixel Programming Function **SQUARE-V3**

```
=====================================================================

                                          list of variables:
   current pixel                          SQCOLOR == _______________
                                          SQSIZE == _______________

=====================================================================
```

```
call function STRAIGHT-LINE
  (pass: SQCOLOR, SQSIZE, RIGHT)
call function STRAIGHT-LINE
  (pass: SQCOLOR, SQSIZE, DOWN)
call function STRAIGHT-LINE
  (pass: SQCOLOR, SQSIZE, UP)
call function STRAIGHT-LINE
  (pass: SQCOLOR, SQSIZE, LEFT)
```