

# Pixel Programming: Packet 2

## Intermediate Commands, Part 2

We aren't introducing any new commands, just a new way to use a very powerful command. Last time you were pixel programming, you learned about the repeat statement, which told you to repeat all the commands inside the bracket. You can stick pretty much any code inside a bracket, including more repeat commands. In other words, you can use loops with loops! These are called nested loops.

Let's look at an example. These three chunks of Pixel Programming code are identical.

<pre>repeat 2 times ----   repeat 3 times   ----     MARK     move RIGHT   ----   move DOWN ----</pre>	<pre>repeat 2 times ----   MARK   move RIGHT   MARK   move RIGHT   MARK   move RIGHT   move DOWN ----</pre>	<pre>MARK move RIGHT MARK move RIGHT MARK move RIGHT move DOWN MARK move RIGHT MARK move RIGHT move DOWN MARK move RIGHT MARK move RIGHT move DOWN</pre>
--	---	--

---

# Pixel Programming: Packet 2

## Intermediate Commands, Part 2

The code on the left is the original nested loop. In the middle, we've unpacked the inner loop, so the `MARK` and `move RIGHT` bit is now repeated 3 times inside the outer loop. But the outer loop needs to be unpacked too, so we repeat everything there 2 times.

This concept can be a bit tough; it needs a bit of practice. Just think like a computer, and take things one instruction at a time. If you follow the instructions the way you're supposed to, the way you learned last time, you'll be fine; remember, you aren't learning any new commands, just new ways to put them together.

In addition to trying out the new demo Pixel Programs, the ones with nested loops, try creating your own if you haven't already. Experimenting with sticking a loop within a loop is one of the best ways to really get a feel for how this works.

---

# Pixel Programming: Packet 2

## Pixel Programming Codename BROWN

=====

current pixel

list of variables:

COLOR == \_\_\_\_\_

NUMZIGS == \_\_\_\_\_

ZIGLEN == \_\_\_\_\_

=====

ask user to set NUMZIGS

ask user to set ZIGLEN

set COLOR to BLACK

repeat NUMZIGS times

----

| repeat ZIGLEN times

| ----

| | move UP

| | move RIGHT

| | MARK

| ----

|

| repeat ZIGLEN times

| ----

| | move DOWN

| | move RIGHT

| | MARK

| ----

----

# Pixel Programming: Packet 2

## Pixel Programming Codename ARIGATO

=====

current pixel

list of variables:

COLOR == \_\_\_\_\_

NUMBER == \_\_\_\_\_

SIZE == \_\_\_\_\_

TEMP == \_\_\_\_\_

=====

ask user to set NUMBER	set TEMP to (SIZE*3)	repeat NUMBER times
ask user to set SIZE	repeat TEMP times	----
set COLOR to BLACK	move UP	repeat SIZE times
set TEMP to (NUMBER*SIZE*2)	MARK	----
repeat TEMP times	----	move LEFT
----	repeat NUMBER times	move UP
move RIGHT	repeat NUMBER times	MARK
MARK	----	repeat SIZE times
----	repeat SIZE times	----
set TEMP to (SIZE*3)	----	move LEFT
repeat TEMP times	repeat SIZE times	move DOWN
----	----	MARK
move DOWN	move RIGHT	----
MARK	move UP	
set TEMP to (NUMBER*SIZE*2)	MARK	
repeat TEMP times	----	
----	----	
move LEFT	set TEMP to (SIZE*3)	
mark	repeat TEMP times	
----	----	
	move DOWN	
	----	

# Pixel Programming: Packet 2

## Pixel Programming Codename TUT

=====

current pixel

list of variables:

COLOR == \_\_\_\_\_

SIZE == \_\_\_\_\_

STEP == \_\_\_\_\_

=====

ask user to set SIZE

set COLOR to YELLOW

set STEP to 0

set SIZE to (SIZE-1)

repeat SIZE times

repeat SIZE times

----

----

| set STEP to (STEP+1)

| set STEP o (STEP-1)

| repeat STEP times

|

| ----

| repeat STEP times

| | MARK

| ----

| | move UP

| | MARK

| ----

| | move UP

| repeat STEP times

| ----

| ----

| repeat STEP times

| | move DOWN

| ----

| ----

| | move DOWN

|

| ----

| move RIGHT

|

----

| move RIGHT

----