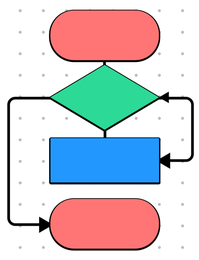


Code flowchart

Understand the flow of code within your system using our code flowchart template. Identify start/stop points, conditions, inputs/outputs and decisions to showcase how a system connects.

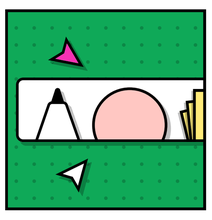
How to use this template

1. Choose your stop and start points.
2. Insert any inputs or outputs and processing points.
3. Insert any decisions.
4. Connect the system with true/false flows.

Quick tips

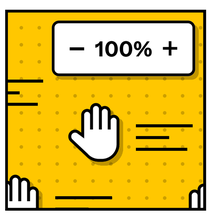
Start with the toolbar

It's at the bottom of your screen, with stickies, stamps, and anything you need.



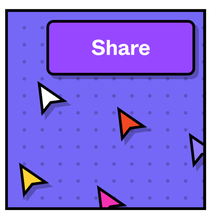
Move and zoom

Use the hand tool to pan around. Zoom controls are in the top right corner.

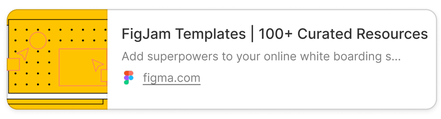


But first: get your team in here

FigJam's better with others. Click the **Share** button above to invite your team.



Explore more templates



PARAM2 = Y coordinate
Read screen memory (Scoreboard) into ZEROPAGE_POINTER_2

Write color ram memory into ZEROPAGE_POINTER_3

Increment through screen memory (Scoreboard)
Add X coordinate
Update into screen memory (Scoreboard)

Increment through color ram
Add X coordinate
Update into color ram memory

Write the screen memory (Scoreboard) to the screen.

If a "0" (zero) is found scanning the byte statements then we are done writing this routine and can exit.

Verify if a line break ("/") character is found, which will write to a new line below.

Write to screen memory (Scoreboard). Get color ram byte (exp: COLOR_WHITE) and write to color ram memory.

Write color ram data to screen (background) in ZEROPAGE_POINTER_1 and continue writing out the rest of the data until complete. PARAM2 (Y position) keeps increasing as data is written down the screen.

START

DisplayText

```
ldx PARAM2
lda SCORE_LINE_OFFSET_TABLE_LO,x
sta ZEROPAGE_POINTER_2
lda SCORE_LINE_OFFSET_TABLE_HI,x
sta ZEROPAGE_POINTER_2 + 1
```

```
lda COLOR_LINE_OFFSET_TABLE_LO,x
sta ZEROPAGE_POINTER_3
lda COLOR_LINE_OFFSET_TABLE_HI,x
sta ZEROPAGE_POINTER_3+ 1
```

```
lda ZEROPAGE_POINTER_2
clc
adc PARAM1
sta ZEROPAGE_POINTER_2
lda ZEROPAGE_POINTER_2 + 1
adc #0
sta ZEROPAGE_POINTER_2 + 1
```

```
lda ZEROPAGE_POINTER_3
clc
adc PARAM1
sta ZEROPAGE_POINTER_3
lda ZEROPAGE_POINTER_3 + 1
adc #0
sta ZEROPAGE_POINTER_3 + 1
```

```
ldy #0
lda (ZEROPAGE_POINTER_1),y
```

cmp #00
beq
@endMarkerReached

cmp #$2F
beq @lineBreak

```
sta (ZEROPAGE_POINTER_2),y
lda PARAM3
sta (ZEROPAGE_POINTER_3),y
iny
```

jmp @inLineLoop

@lineBreak

```
iny
tya
clc
adc ZEROPAGE_POINTER_1
sta ZEROPAGE_POINTER_1
lda #0
adc ZEROPAGE_POINTER_1 + 1
sta ZEROPAGE_POINTER_1 + 1
inc PARAM2
```

jmp DisplayText

@endMarkerReached

RTS

DisplayText byte display

```
SPRITE_CONSOLE_TEXT
        byte ' coll:$   timr:$   spcx:$   spcy:$      /'
        byte ' enco:$   jump:$   mapx:$   mapy:$      /'
        byte '                                        /'
        byte ' enyx:$   wfir:$: frhd:$   actv:$       /'
        byte ' score:$      bullets:$   msbx:$            ',0
```