Programming for 8-bit Commodore systems using a modern IDE

Byron Stout

byron.stout@gmail.com

While you can still program directly on the original hardware, it sure is nice to have the conveniences of a development environment running on a modern PC. Let's use **CBM PRG Studio** to write BASIC and ML programs for our favorite 8-bit Commodore machines.

About Byron…

Healdton Oklahoma

(population 2,861)

Halfway between everything

That's not me in this picture,

But it happened just like this…

https://external-preview.redd.it/RhfX_IN3oP7orhnRfmOp5XIggMxKjhLGkjnnxhEFMCw.jpg?auto=webp&s=0a86b190e649b0cd2ab130b1c3d36f3b916f0ab6

GREEN TREE

EDGEWATER TECHNOLOGY

Manhattan Toy

USAble Life

Empire BlueCross

WINSTON BENEFITS

MERCK

icms®

MARTEN

usbank

STRATOSPHERE

GRAND CASINO

I was PAID to USE

- DBase IV
- C++
- Borland Pascal
- Borland Delphi
- Visual Basic
- Microsoft Access VBA
- RPG
- COBOL
- Java
- Microsoft ASP
- JavaScript
- VB.NET / ASP.NET
- TSQL
- Microsoft C#
- Allaire ColdFusion
- PHP on LAMP

I learned for FREE

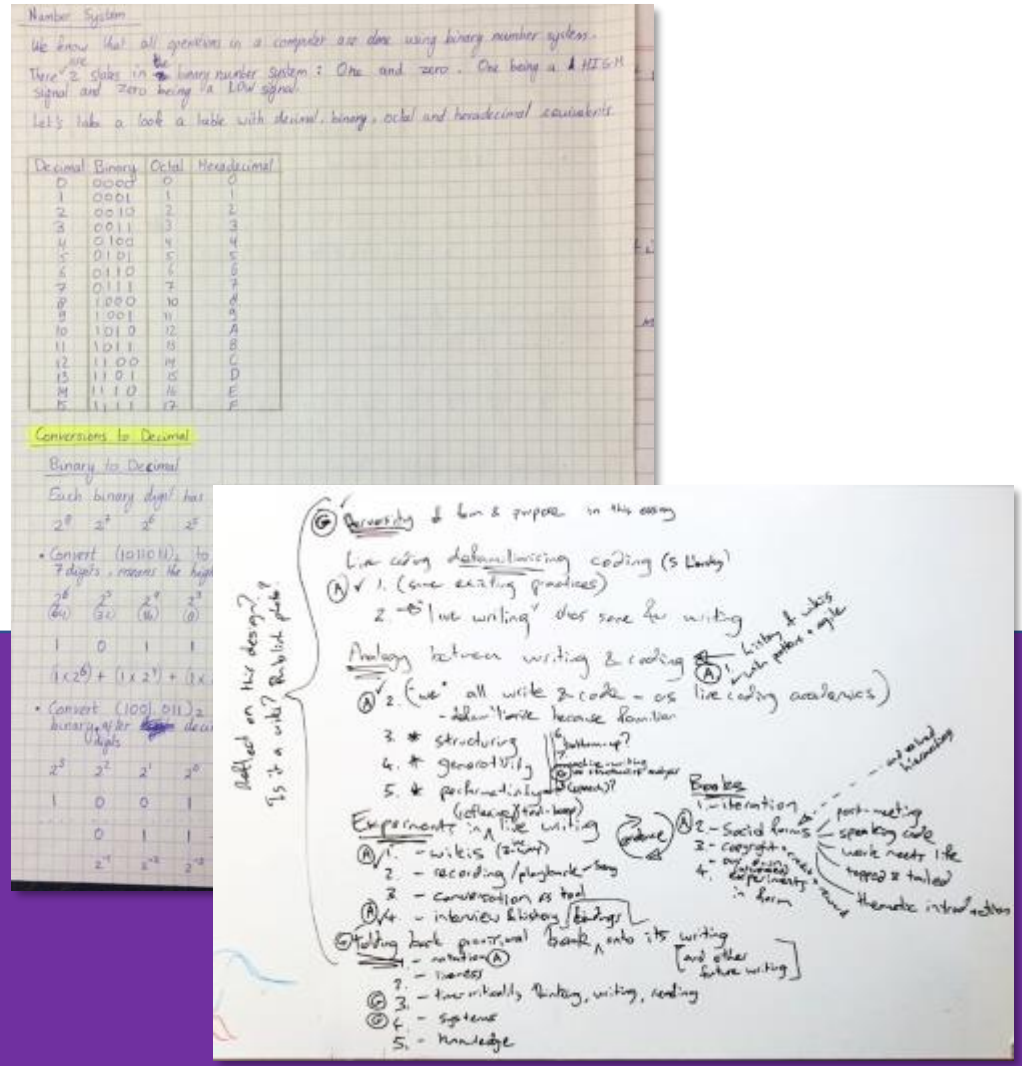
- BASIC
- HyperTalk
- Pilot
- Turbo Pascal
- 8086 Assembly
- 6502 Assembly
- Fortran
- LISP
- Ada
- Clarion
- Perl
- Python
- Ruby

It’s hard to visualize code inside 40 columns and 25 rows

```
60 PRINT"****** ******"
65 PRINT"7 OR 11 WINS":PRINT"3, 6, OR 12
 LOSES"
67 FORTD=1TO1000:NEXT
70 DI=D1+D2:IFDI=7ORDI=11THEN90
75 IFDI=3ORDI=6ORDI=12THEN110
80 PRINT:PRINT"NO WINNING COMBINATION, T
RY AGAIN."
85 FORTD=1TO1500:NEXTTD:PRINTCHR$(147):G
OTO30
90 FORTD=1TO100:NEXT:PRINTCHR$(147)
95 FORP=1TO10:PRINT"  YOU WON !!!!",:NEX
T
100 PRINT:PRINT"YOU WON $"BT:MN=MN+BT
105 FORTD=1TO1000:NEXT:PRINTCHR$(147):GO
TO20
110 PRINTCHR$(147);"TOO BAD,"
115 PRINT"YOU LOST $"BT
120 IFMN-BT<=0THEN130
125 MN=MN-BT:FORTD=1TO1000:NEXT:PRINTCHR
$(147):GOTO20
130 PRINT"YOU RAN OUT OF MONEY !!!!!!"
135 END
READY.
```

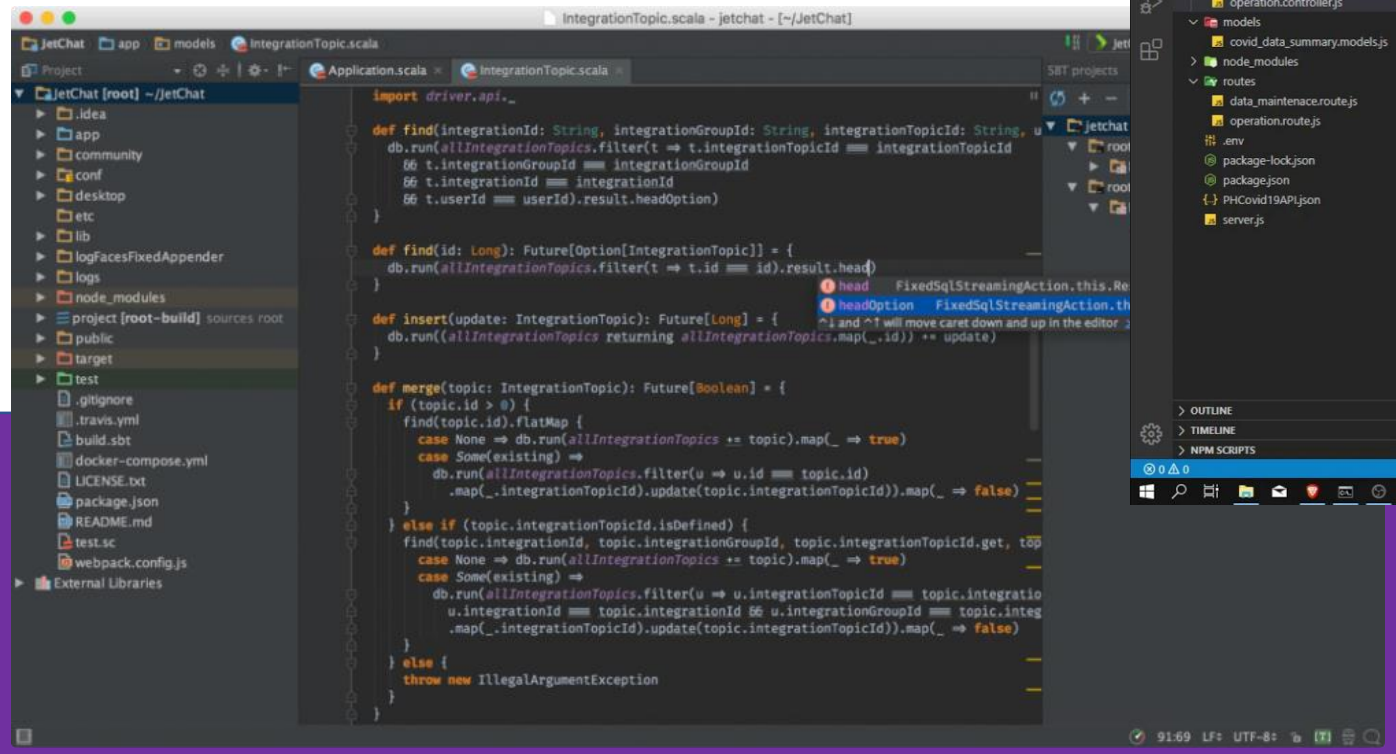
An incredible 320 X 200 pixels!

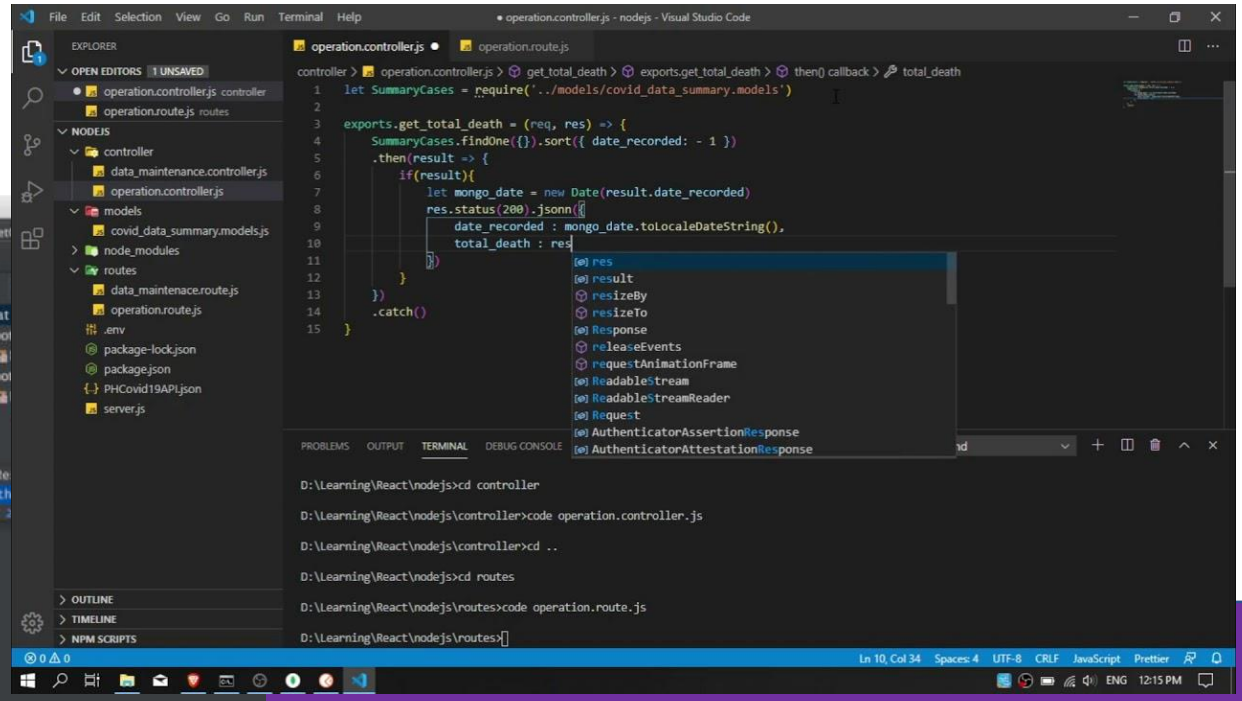
You have to keep a lot of notes…



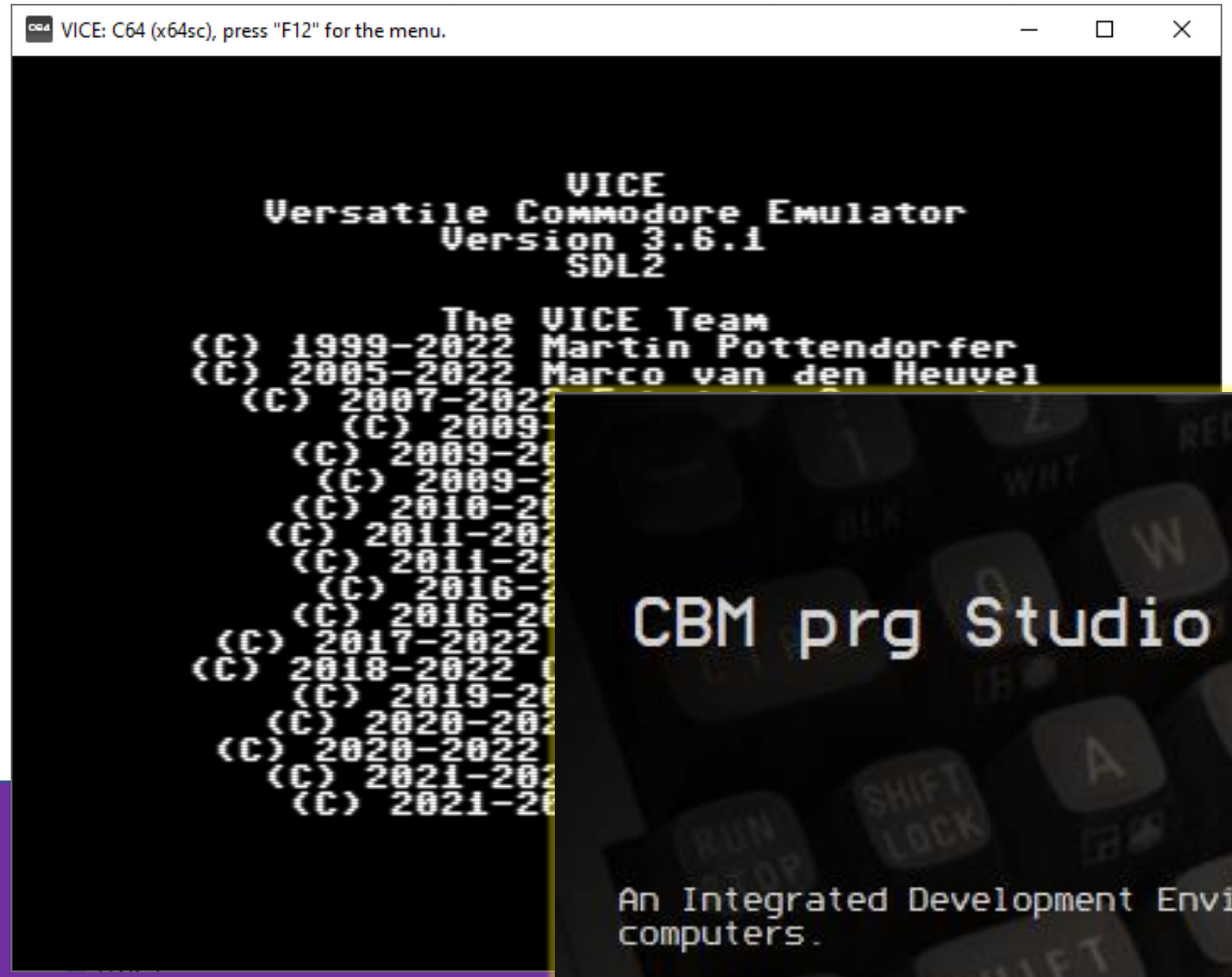
Modern PCs and Development Environments

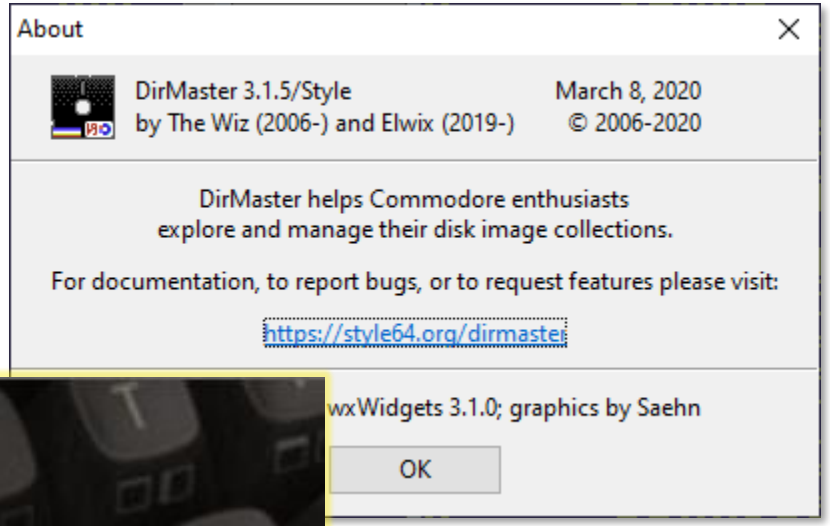
Ridiculous amount of screen space……





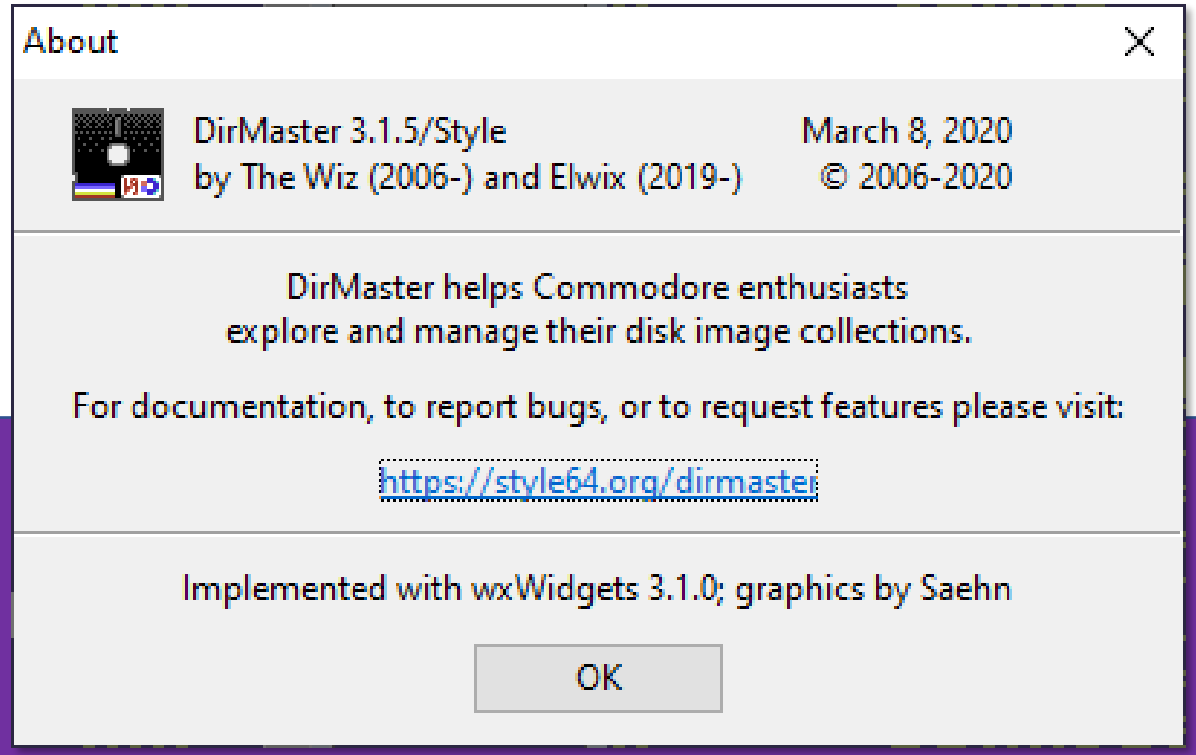
Easily 60 rows and 120 columns (or more for young eyes)





Tools for using a Windows PC for 8-bit Work

https://style64.org/dirmaster
TheWiz and Elwix



- First Released 2006
- Freeware
- View Disk Contents
- Copy, Delete, Move Files
- Add Separators
- Support all sorts of CBM Disk and Tape formats

https://www.ajordison.co.uk
Arthur Jordison

- First Released 2011
- Freeware
- Not Open Source
- Built on Microsoft.NET Framework
- Support for all Commodore 8-bit Machines

Why Use CBM PRG Studio?

- Git integration
- Sprite Editor
- Character Editor
- Screen Designer
- SID tool
- Screen Code Builder
- Memory Viewer
- Assembler/Disassembler

- Program Import/Export
- Code Formatting
- Renumbering Tool
- Auto Code Formatting
- BASIC Constants
- Renumbering
- Multiple Source Files
- MDI Interface

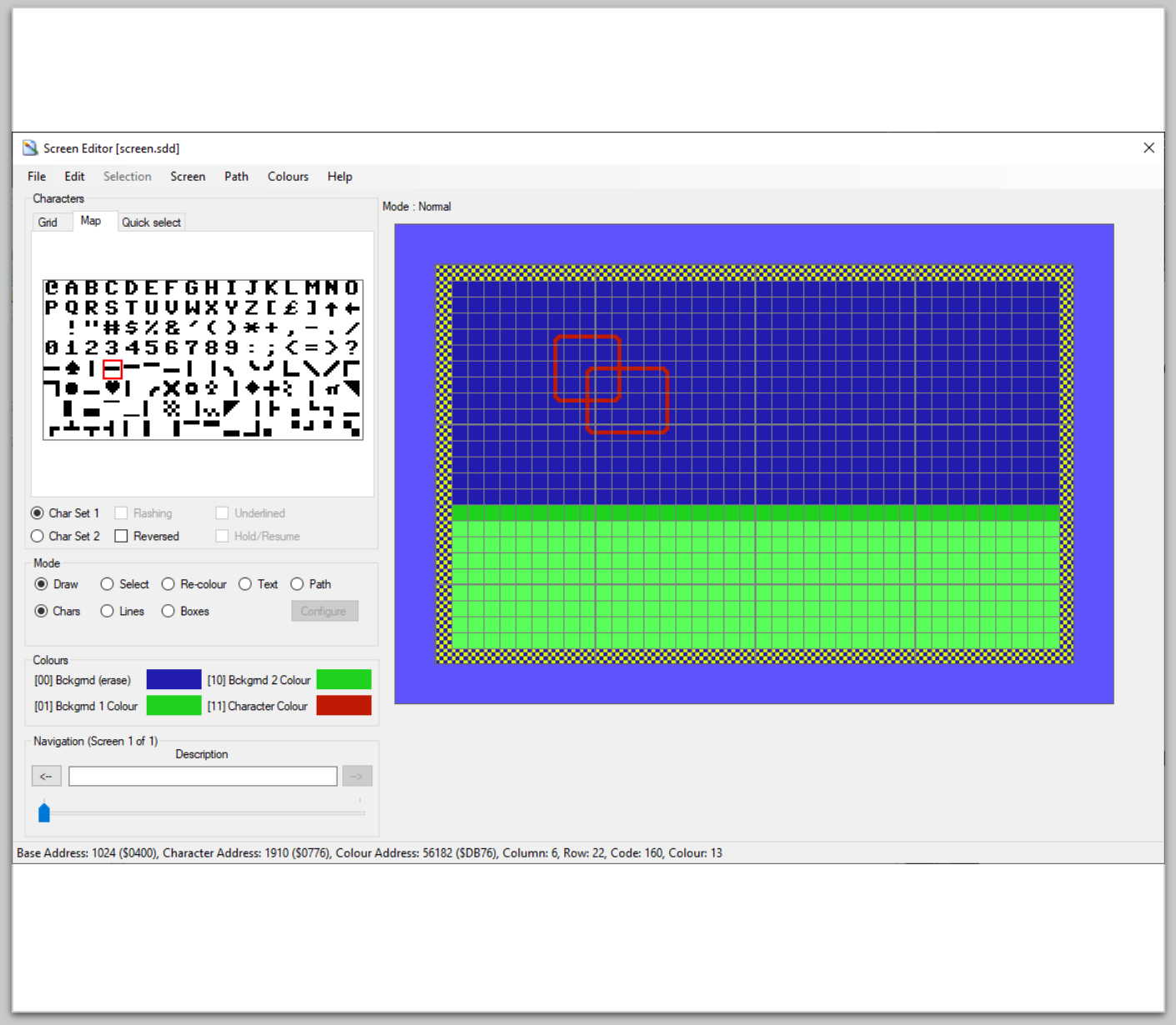
Code Editor Features

- Auto Numbering
- Auto Complete
- Closing Quotes and Parenthesis
- White space
- Block Comment / Indent
- Listing Comments / Comment Blocks
- Regions
- Screen Code Builder
- Renumbering tool
- Multiple Files

```
5 rem BASIC test program
10 print chr$(147)
15 print "{white}"
20 print "hello vcf west"
30 print

#region !- waiting to press a key
50   print "press <any key> to continue"
60   get a$: if a$ = "" then 60
70   print "{clear}"
#endregion#endregion

!-======================
!-Print some characters
!-======================
100 for i = 65 to 80
105     print tab(-64+i);
110     print chr$(i)
120 next i

130 end
```

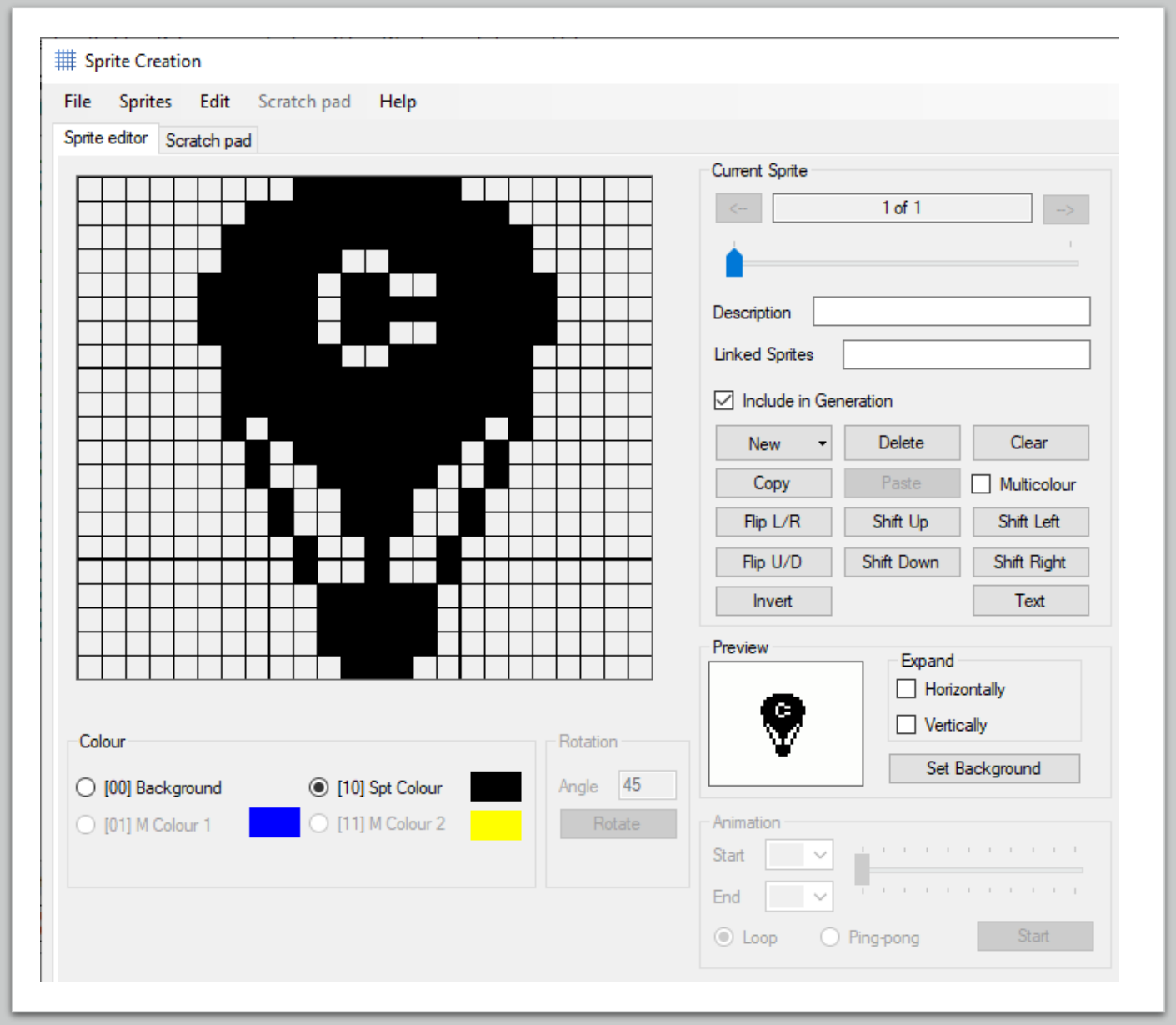
Screen Editor Features

- Draw Mode
- Background / Character Colors
- Line and Box Mode
- Text Mode
- Grid / Map / Quick Select
- Export to Code



Sprite Editor Features

- Import From Source
- Flip Left / Right
- Flip Up / Down
- Shift Left / Right
- Shift Up / Down
- Scratch Pad
- Export To Source



Assembly Features

- 3-pass Assembler
- Labels
- Variables
- Conditionals
- Macros
- Generate a BASIC Loader
- Generate SYS Call

```
1
2  *=$0810
3
4  clear    JSR      $E544
5
6  start    LDA      #02
7           STA      $d020
8           STA      53281
9
10
11          LDX      #65      ; ASCII A
12
13 LOOP     TXA
14          JSR      $E716
15          INX
16          CPX      #91
17          BNE      LOOP
18
19 exit      RTS
```

What did we See? What's Left?

- Git integration
- Sprite Editor
- Character Editor
- Screen Designer
- SID tool
- Screen Code Builder
- Memory Viewer
- Assembler/Disassembler

- Program Import/Export
- Code Formatting
- Renumbering Tool
- Auto Code Formatting
- BASIC Constants
- Renumbering
- Multiple Source Files
- MDI Interface

Next Steps on Your Journey

DOWNLOAD THE TOOLS

CHECK OUT THE HELP FILE

CHECK OUT THE VIDEOS FROM OLDSKOOLCODER

EXPERIMENT AND

HAVE FUN

Thank You for Listening

Byron Stout

byron.stout@gmail.com