TRIANGULAR μOS 1.35 SDK

for

C= COMMODORE 128

PERSONAL COMPUTER

Programmers Reference Guide

© 2023

Contents:

A. Introduction

Programmer's Reference Guide for TRIANGULAR μOS 1.35 SDK (Software Development Kit) explains aspects of TRIANGULAR μOS, a GUI (graphic user interface) operating system for 8-bit Commodore computers.

Goal of creating this system was to develop GUI for 8-bit Commodore computers with lowest amount of memory: that is Commodore PET with at least 4 KB of memory. Next it was expanded for Commodore VIC-20 with standard 5 KB of memory or later with more amount of expansion RAM was required. And in later versions of μOS was adapted for Commodore 64. This iteration of TRIANGULAR μOS (version 1.35) is designed to run on Commodore 128.

This software was written in Commodore BASIC language (port of Microsoft BASIC) using CBM prg Studio 4.1.0, and is designed to run on Commodore 128 in its standard C128 40-column mode. This version of TRIANGULAR μOS is designed to support BASIC 7.0 and works in color text mode. Commodore BASIC (a runtime interpreted language similar in basic concept to JAVA RTM or C# CLI) is default language used in 8-bit Commodore computers and also functions as their OS and user interface. In similar fashion to early Microsoft Windows (1.0 to 3.11), μOS sits atop of BASIC and KERNAL (Commodore's kernel) and Commodore DOS, which is implemented in every Commodore disk drives or 3rd party solutions in order to load μOS modules, load/save settings and documents, perform operations on floppy disks and communicate with disk drive(s).

Package contains files:

- TRIANGULAR uOS 1.35 for Commodore 128 Programmer's Reference Guide.pdf – this document
- Source Code folder with 4 files: UOS.bas (source code of UOS program), GUI.bas (source code of GUI program), uos-cfg.seq (default configuration file) and TEMPLATE.bas (source code of TEMPLATE program)
- TRIANGULAR uOS 1.35.d81 – empty, preformatted System Disk
- TRIANGULAR uOS 1.35 Documents.d81 – empty, preformatted Documents Disk
- TRIANGULAR uOS 1.35 TEMPLATE.d81 – disk with TEMPLATE program helping in starting creating μOS programs

B. What you need

In order to change and/or compile TRIANGULAR μOS 1.35 from source code, you need to do this using external program like CBM prg Studio 4.1.0 (which was used in development and for compilation of μOS 1.35). Using BASIC 7.0 code on real hardware or emulator is out of question, since source code uses extensive line concatenation (lines up to 255 bytes long). Standard BASIC won't present program lines properly (especially print statements) and its screen/program editor won't be able alter those lines.

Download CBM prg Studio here:
https://www.ajordison.co.uk

For fast creation and modification disk content I recommend DirMaster. I formatted my disk with custom PETSCII characters in Disk name and Disk ID using DirMaster.
Download DirMaster here: https://style64.org/dirmaster

For testing and debugging use real Commodore 128 or emulator (I used freeware VICE emulator). Download VICE emulator here: https://vice-emu.sourceforge.io

Commodore 128 emulator VICE must be configured with enabled disk drive that can read 800KB 3.5¨ diskette (.d81 file): recommended CBM 1581*. Also, you should enable joystick. You can easily configure it as Numpad keys:

- Up (8), Down (2), Left (4), Right (6)
- You can move diagonally e.g., Up-Left (7)
- 0 or right Ctrl: Fire (click/select)

You can also enable printer in in VICE emulator. Do this in:
Settings -> Peripheral devices -> Printers or similar options. You can choose printer as device #4 - #7, although #4 is standard and recommended.

* Using 5.25" disk drive: 1571 (default), alternatively 1541 type drive (1541-II) is possible, but System Disk and Documents disk images first must be converted to .d71 or .d64 file in external program (e.g. DirMaster). Additionally using 1571 disk drive amounts to over twice disks drive speed reduction, while 1541 type drive bring speed to default Commodore 64 levels (~10 times slower than 1581) thus only 1581 type drive type is officially supported.

C. How to compile TRIANGULAR µOS 1.35

Source code of UOS and GUI programs is stored in Source code folder in UOS.bas and GUI.bas files. Segments of programs are commented with simple descriptive caption-like comments: !- characters at the beginning if the line are used to mark comment.

After compiling those files UOS.prg and GUI.prg add them to System Disk. File names on disk should always be UOS and GUI (in upper case/graphic mode) or uos and gui (in lower case mode). Remember to put UOS file first (to properly load system with LOAD "*",8 or DLOAD"*" commands).

Next add uos-cfg.seq file (it should have SEQ property) and place it in the middle of UOS and GUI (that's my convention).

You can use empty, preformatted System Disk file to speed up process (TRIANGULAR uOS 1.35.d81 file).

For more information about SYSTEM DISK check section *F. System Disk Content*.

To create new windowed programs to use in TRIANGULAR µOS either use TEMPLATE.bas (in Source code folder) as a starting point and compile it in CBM prg Studio and add to separate disk or SYSTEM DISK or simply use TEMPLATE program from TEMPLATE.d81 disk. This consist of BASICALLY API, an essential part of TRIANGULAR µOS responsible for servicing and drawing µOS, system desktop, Start Menu and TEMPLATE program (program lines 156 – 173).

TEMPLATE is simple windowed program demonstrating working of such program and is great starting point to start development of new µOS apps. It addresses system calls through BASICALLY JUMP TABLE and is written in a manner that allows to edit on real machine or emulator (while system essentials still require CBM prg Studio).

To run your program, attach disk with it to floppy drive and use DISK or CMD program to navigate to and run it.

Also feel free to modify and experiment with TRIANGULAR µOS code.

D. Troubleshooting

Loading of module of TRIANGULAR μOS can "freeze" in process of inter-loading next μOS module or disk program (a very rare occurrence). This happens when loading screen do not proceed to next module for over 1 minute for μOS. When loading screen is not responsive for longer time, it can mean error in inter-loading procedure, most probably keyboard buffer was not filled with key properly. To see what really happened change color of cursor to blue (press Control + 7) and enter command COLOR 0,1 and hit Return key. This should change background color to white which will show underlying black text of loading sequence message. If computer doesn't change cursor or background color try again. If still there is no effect it might be real freeze. If color change procedure succeeds, try using RUN command to see if program will start or go to top of screen (Home key) and press Return in order try to reload program. If it will loads successfully enter RUN command. If that not work check if load command is correct. It should have format: LOAD "[filename]", [device # (1 or 8 - 11)] like in e.g.: LOAD "GUI", 8. If none of it works then start system anew. To prevent this kind of freeze, try not to use keyboard when inter-loading procedure is performed (it can slip improper key into keyboard buffer, which most often leads to this error).

E. BASICALLY

Below are listed functions of BASICALLY API of TRANGULAR μOS 1.35.
System calls of JUMP TABLE are meant to be permanent and will work on latter μOS versions.

1. Window generator: draws empty window based on data in variable arguments. Before evoking this function assign desired values to those variables:
w1 - window top-left horizontal position
w2 - window top-left vertical position
w3 - length horizontal position
w4 - height bottom-right vertical position
wn$ - window name which will be displayed on title and task bar
ws - window slider (0 – disable / 1 - enable)

Evoke this function with BASICALLY system call gosub 61200

Caution: variable j is used in for=to:next loops.

2. left$(s$,x) – will display x number of spaces (max x = 40)

3. left$(v$,x) – will move cursor down x number of times (max x = 24)

4. left$(h$,x) – will move cursor right x number of times (max x = 40)

BASICALLY JUMP TABLE for TRIANGULAR μOS 1.35

Rudimentary functions

Convert mouse sprite position to window loop objects position:
60000 goto13

Mouse pointer steering:
60001 goto16

Clear SID registers:
60002 goto51

Beep sound:
60003 goto52

Save settings to uos.cfg & store variables in memory system:
60100 goto37

Retrieve memory variables:
60101 goto38

Load program:
60102 goto53

GUI drawing functions

Update color variables:
61000 goto39

1st time VIC-II initialize & sprites off & sprites space clear & sprites shapes:
61001 goto41

VIC-II initialize & sprites off & sprites space clear & sprites shapes:
61002 goto43

Turn off, reset and move all sprites to bottom-right corner:
61003 goto47

Move all sprites to bottom-right corner:
61004 goto48

Create mouse pointer:
61100 goto49

Full size window area:
61101 goto50

Draw load:
61102 goto54

Load sequence:
61103 goto55

Restart system:
61104 goto56

Shut down:
61105 goto57

Go back to uOS:
61106 goto80

Draw Background & task panel:
61150 goto15

Draw DESKTOP icons:
61151 goto75

Go to desktop:
61152 goto61

Go to desktop loop:
61153 goto63

Go to Start Menu:
61154 goto99

Move window:
61160 goto93

Window generator:
61200 goto58

F. System Disk Content

#	Name	Type	Size B	Size KB	Disk Blocks	Size on Disk KB
1	UOS	PRG	4,654	4.54	19	4.75
2	UOS.CFG	SEQ	39	0.04	1	0.25
3	GUI	PRG	41,092	40.13	162	40.50
		TOTAL:	45,785	44.71	182	45.50

G. Memory Map

Keyboard buffer		
Keyboard	**Buffer**	**Buffer size**
C128	842-851	208

Tape buffer:

Commodore 128:	2816-3071 $0B00-$0BFF

Address [DEC / $]	Variable name	Values [DEC]	Function
3071 / $0BFF	Commodore computer line	0	Unknown
		1	PET 1.0
		2	PET 2.0-4.1
		3	VIC-20
		4	C128: C64 Mode
		5	C64
		6	C128
3070 / $0BFE	Screen width	Default: 40	Screen width
3069 / $0BFD	Memory size	0-255	Size in KB
3068 / $0BFC	Desktop background pattern	<>0	0: Default [223]
3067 / $0BFB	Desktop background reverse	0	Not reversed
		1	Reversed
3066 / $0BFA	Title bar color	0-255	4-bit
3065 / $0BF9	Desktop pattern color	0-255	4-bit
3064 / $0BF8	Mouse pointer horizontal position (H0) [40 col]	0-38	Default: [20]
3063 / $0BF7	Mouse pointer vertical position (V0) [40 col]	0-23	Default: [10]
3062 / $0BF6	GUI Program mode	0	None
		1	DESKTOP
		2	THIS PC
		3	SETTINGS
		4	APPS
		5	GAMES
		6	COLORS
		7	DISK
		8	MATH
		9	CMD
		10	External program
3061 / $0BF5	Printer device #	0 or 4-7	Default: 0 [None]

Address	Description	Value	Meaning
3060 / $0BF4	Datasette availability	0	No
		1	Yes - #1 [Default]
3051-3057 / $0BEB-$0BF0	Sequence of values in keyboard buffer for inter-loading module	-	-
3050 / $0BEA	TRIANGULAR µOS version	0-255	135 for version 1.35
3000 / $0BB8	Boot drive # (BD)	8-11	Device #
3001 / $0BB9	Work drive # (WD)	8-11	Device #
3011 / $0BC3	Device #8 detected	0	No
		1	Yes
3012 / $0BC4	Device #9 detected	0	No
		1	Yes
3013 / $0BC5	Device #10 detected	0	No
		1	Yes
3014 / $0BC6	Device #11 detected	0	No
		1	Yes
3015 / $0BC7	Device #8 code	0-255	Ref: Drive codes
3016 / $0BC8	Device #9 code	0-255	Ref: Drive codes
3017 / $0BC9	Device #10 code	0-255	Ref: Drive codes
3018 / $0BCA	Device #11 code	0-255	Ref: Drive codes

Drive codes

#	Drive name	Also used by / Notes
0	Unknown	
7	2031	
16	2040	
32	3040	
169	4040	
170	1541	1540 on VIC-20
76	1541-II	8050, 8250, SFD-1001 D9060/D9090
255	1551	
173	1571	1570
108	1581	
48	SD2IEC	* Experimental

H. BASIC Variables

#	Variable	Type	Description	Value	Memory cell	Notes
1	K$	String	Default key char variable	Any	-	
2	I	Float	Temporary variable (used e.g., in FOR)	Any	-	
3	J	Float	Temporary variable (used e.g., in FOR)	Any	-	
4	K	Float	Temporary variable (used e.g., in FOR)	Any	-	
5	T$	String	Time in TI$ form	"000000" - "235959"	-	
6	T1$	String	Hours	"0" - "23"	-	
7	T2$	String	Minutes	"0" - "59"	-	
8	T3$	String	Seconds	"0" - "59"	-	
9	Q$	String	Quotation mark	CHR$(34) / ["]	-	
10	N$	String	String variable with file name	Any	-	
11	C$	String	Temporary string variable	Any	-	
12	C1$	String	Secondary temporary string variable	Any	-	
13	L$(X)	String array	Loading text and CBM DOS commands	Any	-	Used with subscript 1 or 2 only
14	L	Float	Cut string by L value (RIGHT$ or LET$)	Any	-	Used only in CMD & MONITOR
15	CL	Float	Command length	Any	-	Used only in CMD & MONITOR
16	FM	Float	BASIC System memory in KB	0-255	3069 $0BFD	
17	VT	Float	Version number	135	-	135 for version 1.35
18	BD	Float	Boot drive #	8-11	3000 $0BB8	
19	BS	Float	Temporary boot drive #	8-11	*3000 $0BB8*	Used only in UOS
20	WD	Float	Work drive #	8-11	3001 $0BB9	
21	DR	Float	Active drive #	8-11	-	
22	DD	Float	Datasette availability	0-1	3060 $0BF4	
23	PP	Float	Printer device #	0 or 4-7	3061 $0BF5	
24	AC	Float	GUI Program mode	0-8	3062 $0BF6	
25	BP	Float	Desktop background pattern	<>0 (0 = Default [223])	3068 $0BFC	
26	BR	Float	Desktop background reverse	0-1	3067 $0BFB	
27	BC	Float	Desktop pattern color	0-255	3065 $0BF9	

28	TC	Float	Title bar color	0-255	3066 $0BFA	
29	TC$	String	Title bar color in string format	Any	-	
30	V	Float	Mouse pointer vertical position	0-23	3063 $0BF7	
31	H	Float	Mouse pointer horizontal position	0-38	3064 $0BF8	
32	V0	Float	Mouse sprite position vertical	0-255	-	
33	H0	Float	Mouse sprite position horizontal	0-255	-	
34	G0	Float	Border color register	Constant	53280 $900F	G0+1 for Background color
35	G1	Float	Start of screen memory	0-255	-	PEEK(2616)*256
36	GC	Float	Start of color memory	0-255	-	((PEEK(648)AND3)+148)*256
37	C1	Float	Cursor position memory cell	Any	-	(V*SW+H0)+G1
38	CV	Float	Original char ASCII code under cursor	0-255	-	
39	S	Float	Start of SID sound memory	0-255	37151 $911F	Used only in GUI programs
40	J	Float	Read Joystick register	0-255	56320 $DC00	Used only in GUI programs
41	S0	Float	Start of voice registers	54272	36873 $9009	
42	VL	Float	Volume level register	0-16	36878 $900E	
43	R	Float	Return from Start Menu	0-2	-	
44	V$	String	String cursor down	{down*23} - Constant	-	Used only in GUI programs
45	H$	String	String cursor right	{right*39} - Constant	-	Used only in GUI programs
46	S$	String	String space	{space*39} - Constant	-	Used only in GUI programs
47	W1	Float	Window creator top-left V0 position	Any	-	Used only in GUI programs
48	W2	Float	Window creator top-left H0 position	Any	-	Used only in GUI programs
49	W3	Float	Window creator bottom-right V0 position	Any	-	Used only in GUI programs
50	W4	Float	Window creator bottom-right H0 position	Any	-	Used only in GUI programs
51	WN$	String	Window creator window name	Any	-	Used only in GUI programs
52	WS	Float	Window creator display slider	0-1	-	Used only in GUI programs
53	FH$	String	Store disk header	Any	-	Used only in DISK & CMD
54	FI	Float	Number of disk entries	Any	-	Used only in DISK & CMD
55	FI$(X)	String array	String array storing disk content list	Any	-	Used only in DISK & CMD (X=1004)
56	PS$	String	String clearing disk header	Constant	-	Used only in DISK

I. Support

For more information, to report bug or to get help go to links listed below.

TRANGULAR µOS for Commodore 128 on Commodore 128 forum:
c-128.freeforums.net/thread/1102/triangular-c128-commodore-128-basic

TRANGULAR µOS for Commodore 128 on Lemon64 forum:
www.lemon64.com/forum/viewtopic.php?t=80716

TRIANGULAR YouTube channel:
www.youtube.com/@triangular_uos

Contact info:

 Michael Goral
 @-mail: michaelgoral@gmail.com

J. Changelog

Changelog for TRIANGULAR µOS 1.35 for Commodore 128 [23-06-2023]:

- BIOS improved and bug fixed
- Movable windows by clicking on title bar
- Mouse pointer change when in moving window mode
- Mouse pointer change to hourglass when waiting
- Desktop icons layout rearrangement
- Start menu orb from sprite
- MATH bug fixed
- GUI cleaned up and improved
- Sprites without flickering
- Updated loader graphic using sprite stripes
- Border COLOR reinforced
- Code refactored
- CMD and MONITOR text area handled by WINDOW command
- CMD info properly display amount of free memory from both memory banks
- MONITOR function calling machine program is much simpler
- Drive detection database is held in DATA statements
- CRAB IN NEW YORK error in which 1 collision with cars or trains takes 2 lives is bug fixed plus minor improvements
- STAR WARS: X-Wing vs TIE Fighter game added
- DO...WHILE...LOOP...UNTIT...EXIT and BEIN...BEND commands implemented
- FAST and SLOW utilized to speed up drawing PETSCII elements of inter-loading, in BIOS and GAMES
- BASICALLY API Window creation function expanded
- BASICALLY API Jump Table added
- Improvements and bugfixes
- SDK adds template windowed program with its source files
- Updated and augmented documentation and SDK documentation now in single PDF file

Changelog for TRIANGULAR µOS 1.34 for Commodore 128 [28-05-2023]:

- Windows loops changes to relative windows position
- Mouse routine improved and mouse position changed
- More BASIC 7.0 commands added
- SIMCITY bug fixed
- CRAB IN NEW YORK (a 3rd game) added
- Bugfixes

Changelog for TRIANGULAR µOS 1.33 for Commodore 128 [17-05-2023]:

- Mouse routine reworked and improved
- BASICALLY Window creator reworked with window displaying mechanism
- TREASURE CHAMBER, game by Fabrizio Caruso added
- More BASIC 7.0 commands added (IF…THEN…ELSE, SLEEP) and RESTORE command expanded
- Bugfixes

Changelog for TRIANGULAR µOS 1.32/C128 for Commodore 128 [11-02-2023]:

- BASIC 7.0 WINDOW command utilized
- Enlarged DESKTOP area
- Change versioning scheme and version held as numeric value in memory
- Change file system from filename>ext (extension) to filename.ext (extension) eg.: uos>cfg to uos.cfg
- BASIC 7.0 sound commands implemented
- Minor improvements and bugfixes

Changelog for TRIANGULAR µOS 1.31/C128 for Commodore 128 [15-01-2023]:

- Mouse pointer routines redesigned which resulted in twice faster movements
- Some additional graphic operations converted to BASIC 7.0 syntax
- Minor improvements and bugfixes

Changelog for TRIANGULAR µOS 1.30/C128 for Commodore 128 [12-01-2023]:

- Commodore 128 in its standard 40 column VIC-II C128 mode is supported
- Only 3.5" 1581 type disk drive is supported
- Loading and saving is up to 10x faster due to faster C128 1581 disk drive handling
- Color theme changed from C64 blue to more resembling C128
- BASIC 7.0 handles sprites and few needed instructions added
- GAMES folder contains only SIMCITY, other 3 games removed
- Minor improvements and bugfixes
- Changelog is revised and integrated back into User's Manual

Changelog for TRIANGULAR µOS 1.25/C64 for Commodore 64 [20-10-2022]:

- Code of TRIANGULAR µOS is reviewed, improved, cleaned and bug fixed
- Slightly updated loader module screen (LOADING/RESTARTING/SHUT DOWN)
- Multi-characters handled by special function amounted for saving 1 KB of code
- BIOS: Some minor visual changes (mostly colors)
- BIOS: Detection of C64 or C128 in C64 mode
- GUI: Mouse pointer uses sprite instead of PETSCII character
- GUI: Time separator : (in right bottom corner) is blinking as seconds progress
- GUI: Windows repositioned and stretched with minor visual changes
- GUI: Windows can activate elements outside their loops in more advanced manner
- GUI: First element of µOS API named BASICCALLY added: Window creator that draws window based on given arguments
- GUI: Windows refreshes faster and without blinking
- SETTINGS: System settings and color settings merged into one SETTINGS window, divided by 2 tabs into SYSTEM and GRAPHICS (color options)
- SETTINGS: GRAPHICS tab now supports 16 colors and 2x more background patterns
- DISK: Program window slightly enlarged and improved few minor mechanisms
- CMD: Small visual changes and improved few minor mechanisms
- WORDS: Commands panel slightly redesigned with new options added
- WORDS: Repositioned and expanded text area
- WORDS: > symbol (indicating line to write) is removed and text area is 1 character wider
- WORDS: Faster typing-in mechanism
- WORDS: Clear Screen [Shift + Home] erases currently written line
- WORDS: F6 Copy line and F7 Paste line added
- WORDS: Clipboard shared with MATH calculator (F7 will retrieve memorized number in MATH)
- MATH: Window repositioned and minimal color theme update
- MATH: Clipboard shared with WORDS word processor (memory contains number copied inside WORDS)
- MONITOR: Show memory area function is reworked and enlarged
- SYNTH: Piano keyboard expanded and new voices added
- SIMCITY: Mechanism that selects land plot to activate is more random and equally covers available zoning area
- GAMES: CHUCK'S CHALLENGE & LIFE RAFT RESCUE - issues with sprites are corrected
- Changelog added as separate document

Changelog for TRIANGULAR µOS 1.20/C64 for Commodore 64 [31-08-2022]:

- Support for Commodore 64 with its 40 column VIC-II text mode and SID chip is added
- BIOS added indicator reminding to type time (<- TYPE NEW TIME) after pressing T
- WORDS – by pressing Home key you can reach first line of document
- SYNTH – reworked with simplified keyboard and limited to 1 voice channel which now can play sounds in normal, lower and higher octave ranges
- Only SIMCITY is carried over from TRIANGULAR µOS 1.15/VIC for Commodore VIC-20
- In place of old games 3 new games from Roman Werner added (TAXI TAXI, CHUCK'S CHALLENGE and LIFE RAFT RESCUE)
- All games merged into GUI program file
- Bugfixes and improvements

Changelog for TRIANGULAR µOS 1.15/VIC for Commodore VIC-20 [19-08-2022]:

- TRIANGULAR µOS 1.15/VIC won't start on VIC-20 with less than 29 KB of RAM (24 KB RAM Expansion is needed or higher)
- GUI merged with MONITOR, WORDS and SYNTH
- MONITOR and WORDS errors messages accompanied by beep sound
- Besides LOADING screen there is added RESTARING (with yellow TRIANGULAR logo) and SHUT DOWN (with red logo)
- Bugfixes and improvements

Changelog for TRIANGULAR µOS 1.14/VIC for Commodore VIC-20 [14-08-2022]:

- TRIANGULAR µOS won't start on VIC-20 with less than 21 KB of RAM (16 KB RAM Expansion is needed or higher)
- GUI merged with MATH and CMD
- MATH keys assigned for basic functions (+, -, *, /, %) and ON
- CMD beep sound added while displaying error messages

Changelog for TRIANGULAR µOS 1.13/VIC for Commodore VIC-20 [12-08-2022]:

- TRIANGULAR µOS won't start on VIC-20 with less than 13 KB of RAM (8 KB RAM Expansion is needed or higher)
- GUI merged with COLORS & DISK
- GUI streamlined
- DISK received minor improvement of disk content handling mechanism

Changelog for TRIANGULAR µOS 1.12/VIC for Commodore VIC-20 [09-08-2022]:

- GUI merged with APPS, GAMES and SETTINGS
- COLORS can retrieve default color theme with DEAULT button
- DISK & CMD directory of disk content is retrieving file list at once
- CMD other updates, D> (duplicate) command added and other command syntax changes
- MATH improved, various functions added and few additional keys mapped
- MONITOR command syntax overhauled into 1-line commands
- Bugfixes and other minor improvements

Changelog for TRIANGULAR µOS 1.11/VIC for Commodore VIC-20 [31-07-2022]:

- TRIANGULAR µOS won't start on unexpanded VIC-20 5KB. VIC-20 with 8KB of RAM (3KB RAM Expansion) is needed (or higher)
- UOS and BIOS merged into single UOS program and improved error messages system
- CMD merged with its help file CMD>HLP and further improved
- CMD syntax of R> and C> operations changed to more intuitive [original file]=[new file]
- STAR WARS merged with its game engine file STAR WARS>ENG

Changelog for TRIANGULAR µOS 1.10/VIC for Commodore VIC-20 [28-07-2022]:

- Support for Commodore VIC-20 with at least 5KB and it's 22 columns, 8 color text mode added
- New colorful loader for inter-loading operations
- BIOS Setup menu offers option to enable/disable Datasette
- BIOS Setup supports separate Work disk drive for storing system apps documents
- BIOS improvements and bugfixes
- GUI cursor is joystick operated and can move diagonally
- GUI windowed environment uses custom color background and title bar
- Up arrow button added to task bar (placed right of clock) to go back to main desktop screen
- SETTINGS is split into two apps: SETTINGS which can change time, work disk and printer options and COLORS which can change colors of GUI elements.
- DISK is now windowed and cursor operated. Changed disk content display mechanism.
- CMD introduced improved mechanism for displaying success or error of performed operation. Minor review of command syntax (I> and V> instead of I and V). Bugfixes.
- APPS folder instead of OFFICE, contains WORDS, MATH, MONITOR, SYNTH
- MATH calculator revamped, simplified, windowed and cursor operated
- MONITOR have blue background. Command SHOW displays 2 hex digits instead of 4. HELP is consolidated. Minor bugfixes.
- 4 new games: SIMCITY, STAR WARS X-Wing vs TIE-Fighter (new version), BREAKOUT (new version), NEED4VIC
- Various other bugfixes and improvements
- Empty and formatted disk image called TRIANGULAR uOS 1.10-VIC Documents in .d64 and archived .zip file formats added for use as Work disk

Changelog for TRIANGULAR µOS 1.05/PET for Commodore PET [29-06-2022]:

- Config file contains system key
- UOS/BIOS error messages system improved
- DISK text program is placed in THIS PC and it shows disk content and runs programs
- SETTINGS can properly cancel changes and other bugfixes
- Improvements, bugfixes and cleaned code from redundant parts in all programs produced very stable version

Changelog for TRIANGULAR µOS 1.04/PET for Commodore PET [21-06-2022]:

- OS name changed to TRIANGULAR µOS
- Launching program and config file names changed
- 8 KB version removed (since it is actually slower than 4 KB version)
- GUI: windows have black close buttons
- Taskbar window name moved to left side of TRIANGULAR logo orb
- Click/select key changed to 0 (zero)
- DESKTOP renamed to GUI
- THIS PC window renamed to THIS PET and adds DISK icon which loads program form disk
- SETTINGS now have SAVE button for saving settings
- Some icons updated
- CMD is greatly overhauled with commands syntax similar to DOS Wedge/JiffyDOS and added listing directory of disk content function
- MONITOR improved
- WORDS instead of WORD – this is completely new word processor
- SIMCITY game added in place of LUNAR LAND
- Other games have slightly different menu keys
- Games from 8 KB version removed
- Improvements and bugfixes

Changelog for TRIANGULAR OS 1.03 for Commodore PET [27-02-2022]:

- System now have 4KB and 8KB modes – launcher will choose which one to boot into
- 8KB mode has consolidated code of GUI, STAR WARS into 1 program, as well as BIOS and TRIANGULAR OS launcher, CMD and its HELP, Monitor and its HELP
- Fixed bug in disk detection system
- Few minor bugfixes
- 2 new games (RATRUN & MAD BOMBER) only in 8KB mode (instead of SNAKES and LUNAR LAND)

Changelog for TRIANGULAR OS 1.02 for Commodore PET [6-02-2022]:

- Minor visual changes across the board (mostly highlighted key letters)
- Launch program renamed to TRIANGULAR OS
- TRIANGULARS OS/BIOS – has improved disk drive detection system. Drive database expanded (include SD2PET (experimental)). Loads OS>CFG file with wallpaper settings.
- BIOS Setup Menu – option is added to exit to BASIC
- GUI OS – SETTINGS saves wallpaper settings in file OS>CFG
- CMD – fixed drive # change mechanism. Minor bugfixes.
- OFFICE apps visual revision and many bugfixes
- GAMES minor visual changes and bugfixes

Changelog for TRIANGULAR OS 1.01 for Commodore PET [16-01-2022]:

- First version to have manual
- Various minor improvements done in launching TRIANGULAR program
- BIOS – fixed launching logo position
- GUI – memorize cursor position in-between of loading modules plus minor bugfixes
- CMD – many bugfixes
- MONITOR – fixed serious bug preventing user from running machine language programs
- OFFICE – apps can now properly load and save data on disk
- STAR WARS – added music in intro and outro. Game engine now don't reset system clock
- SNAKES – AI opponent fixed and minor esthetic changes
- LUNAR LAND – received minor esthetic changes

Changelog for TRIANGULAR OS 1.00 for Commodore PET [24-12-2021]:
- Starting procedure changed: TRIANGULAR disk detecting program -> BIOS (Launching screen combined with Setup Menu) -> DESKTOP (GUI)
- System utilizes memory to store status of TRIANGULAR OS
- BIOS detects if there are disk drives #8 - #11, detecting mechanism is improved and functioning drive type detection added
- BIOS Setup Menu displays drives and can change BOOT drive and restart system
- DESKTOP is streamlined and icons redesigned
- START Menu is placed on center of task bar and is displayed just as TRIANGULAR logo and have SETTINGS, RESTART and SHUT DOWN options
- SETTINGS (renamed CONTROL PANEL) can change desktop wallpaper from 5 patterns
- MY COMPUTER is renamed THIS PC and disk icon now open CMD program
- CMD (renamed DOS) can change operating disk (#8 - #11) plus some improvements
- OFFICE – apps have minor improvements
- MONITOR – heavily reworked and improved, operates on HEX values.
- GAMES icon in place of STAR WARS icon opens folder with 4 games: STAR WARS, SNAKES, LUNAR LAND and BREAKOUT

Changelog for TRIANGULAR OS 1.00 BETA for Commodore PET [2016 to 24-10-2021]:

- System supports 1 cassette recorder: device #1 and only 1 disk drive: device #8
- BIOS have implemented simple PET type detection and it detects if there is disk drive #8
- BIOS Setup Menu is accessed with DEL key where you can change or reset system time
- Starting procedure: BIOS -> TRIANGULAR DOS -> Launch screen -> DESKTOP (GUI)
- DESKTOP (GUI) contains wallpaper, task bar on which is located clock (right bottom), START Menu with TRIANGULAR logo (left bottom) and 4 icons: MY COMPUTER, OFFICE, STAR WARS and MONITOR
- START Menu has CONTROL PANEL, RESTART, EXIT TO DOS and QUIT TO BASIC options
- CONTROL PANEL can change time and reset system clock
- MY COMPUTER contains cassette and disk icon which can load first encountered program (LOAD for cassette icon and LOAD "*",8 for disk icon)
- OFFICE contains 4 office suite programs: WORD a word processor, CALC spreadsheet, CONTACTS contact manager and MATH calculator
- STAR WARS brings fabulous STAR WARS X-Wing vs TIE-Fighter game
- MONITOR a very simple memory monitor program, operates on decimal numbers