

# T.E.R.O. Level Editor

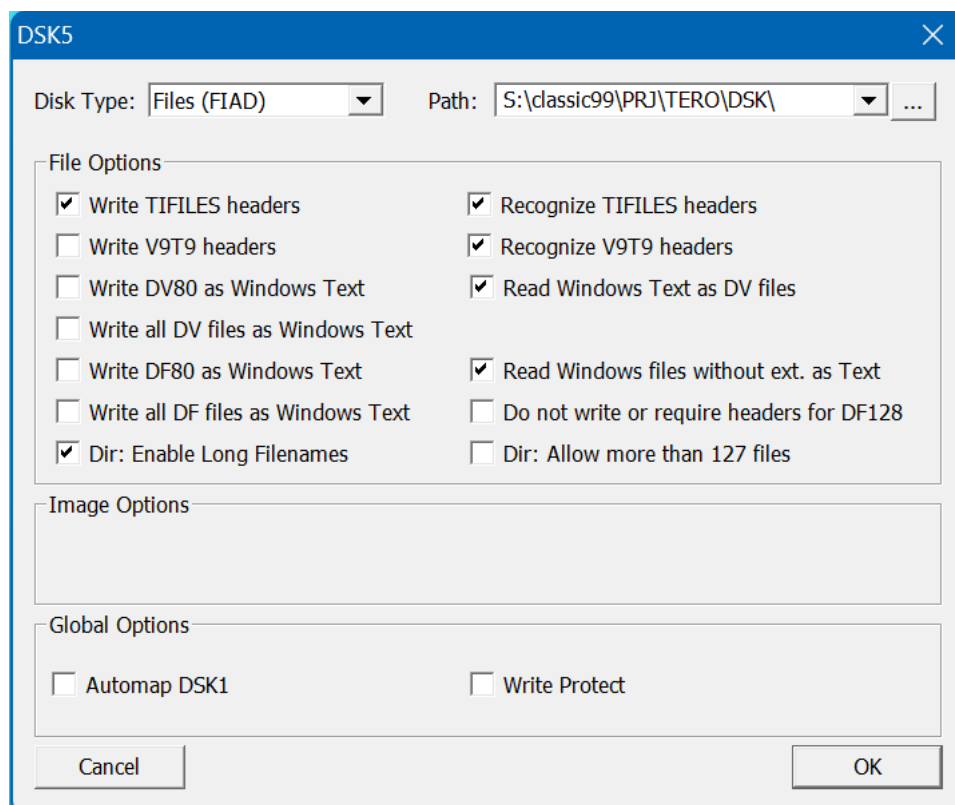
## How to create your own T.E.R.O. game ?

T.E.R.O is a game module for the TI-99/4a. It has some levels included, but can load additional games from disk. This manual describes how to build your own game on Windows, test it in Classic99 and prepare a release.

- Setup your environment
- Create a menu entry
- Use the T.E.R.O. Level Editor
- Test your level(s) in Classic99
- Prepare a release

## Setup your environment

Setup a FIAD disk directory for your game in Classic99. This manual used DSK5:



We will use Windows Text-Files while developing and convert them to a Disk-Image or TIFILES when done.

You need to copy the files to the

- TERO-CAT

- HERO-L01
- EDCAT

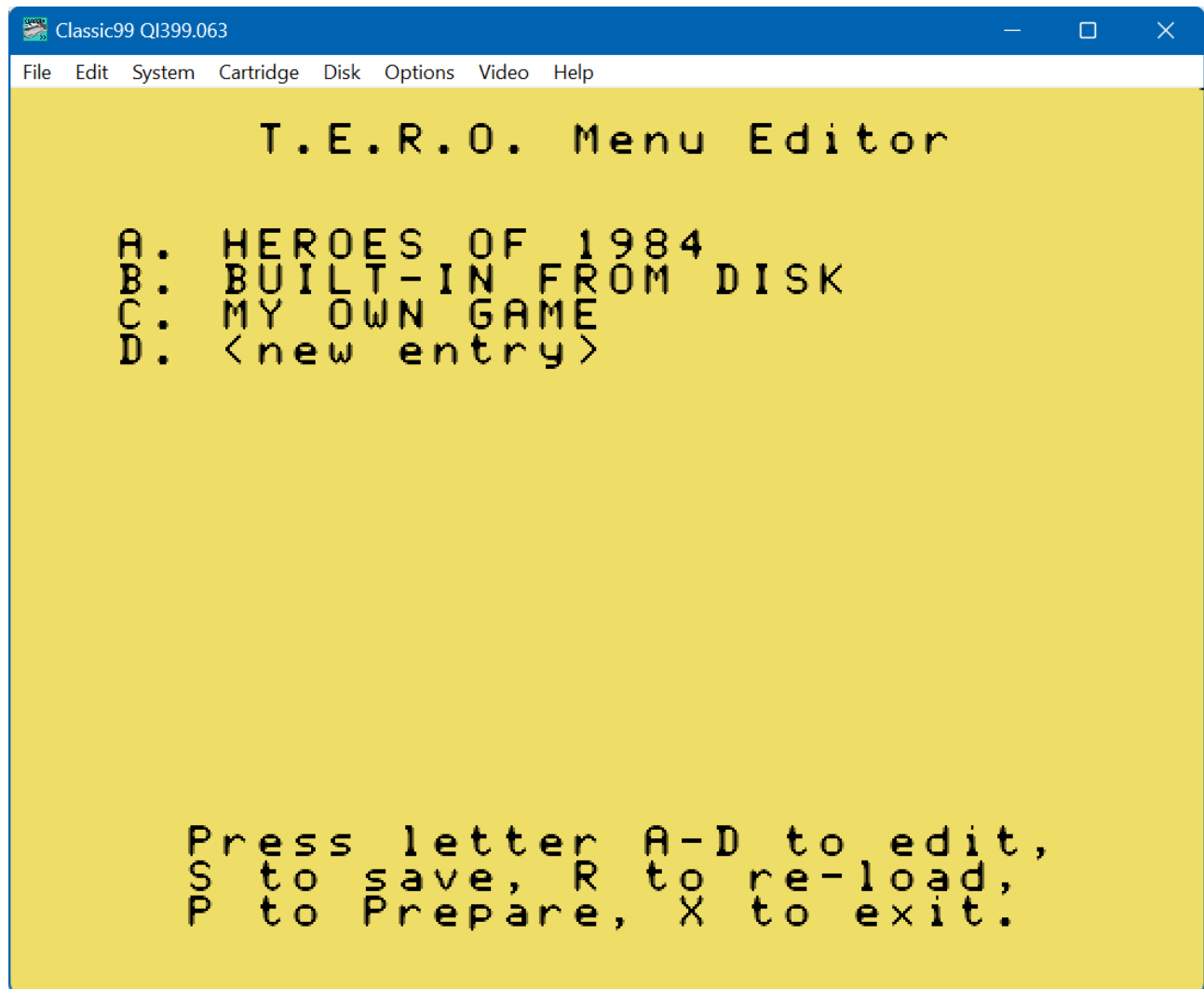
Rename HERO-L01 to the name you will use for your game, keep the suffix -L01, i.e. STEVE-L01

## Create a Menu-Entry for your Game

Start any Extended BASIC and load the Catalog Editor EDCAT:

RUN “DSK5.EDCAT”

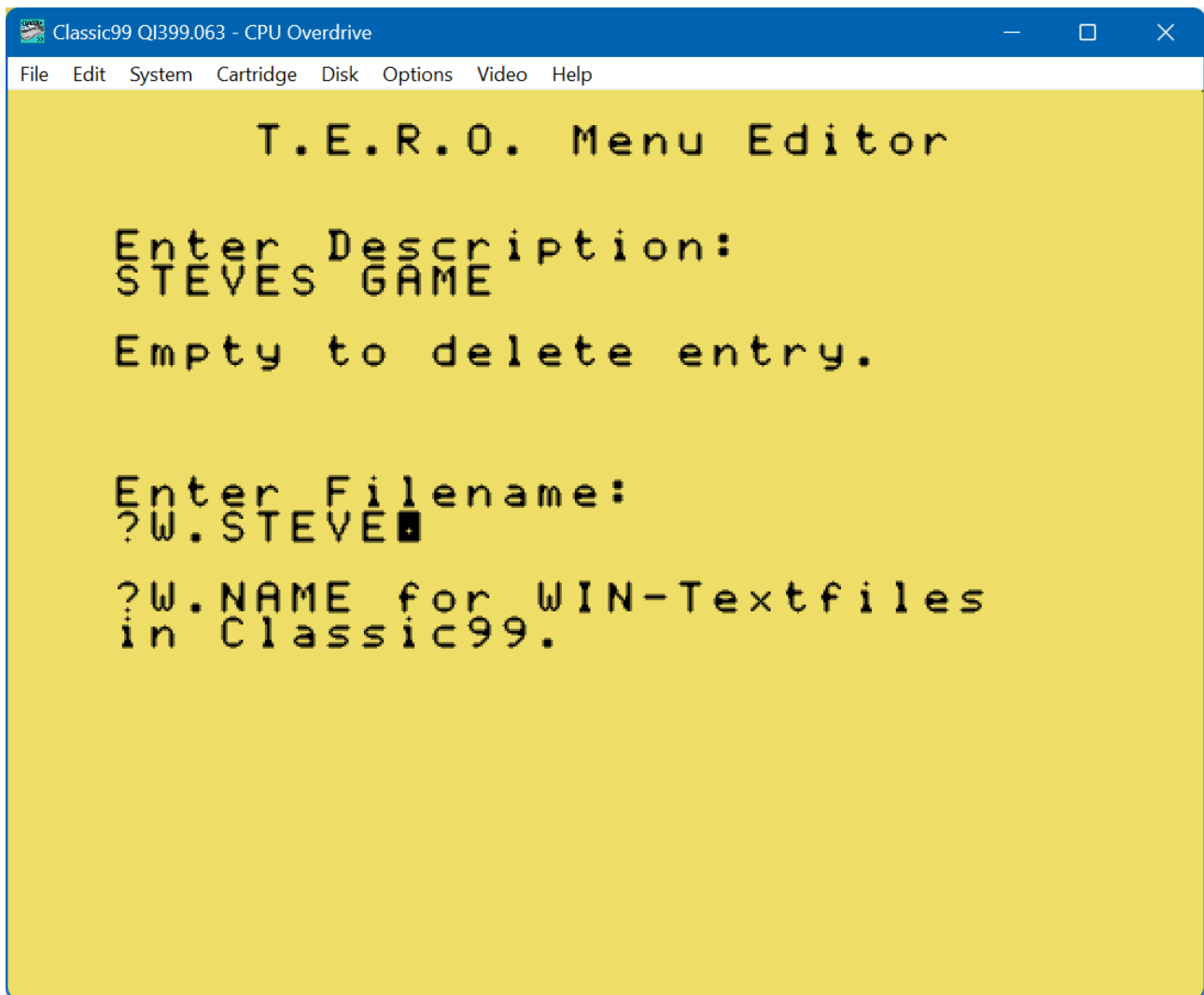
Select the disk you use for your game, default is DSK5



Change an existing entry or chose the letter with the label <new entry>.

The game C “MY OWN GAME” is already created for first experiments and is the default of the Level Editor.

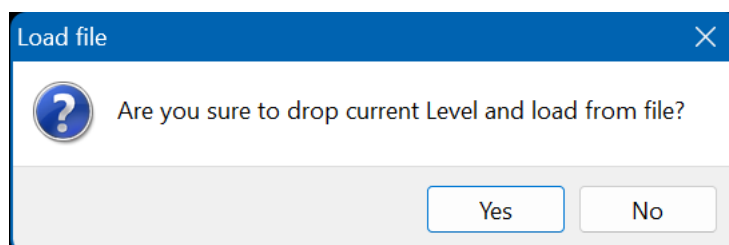
First enter the description in the menu, then the filename without the suffix -L0x

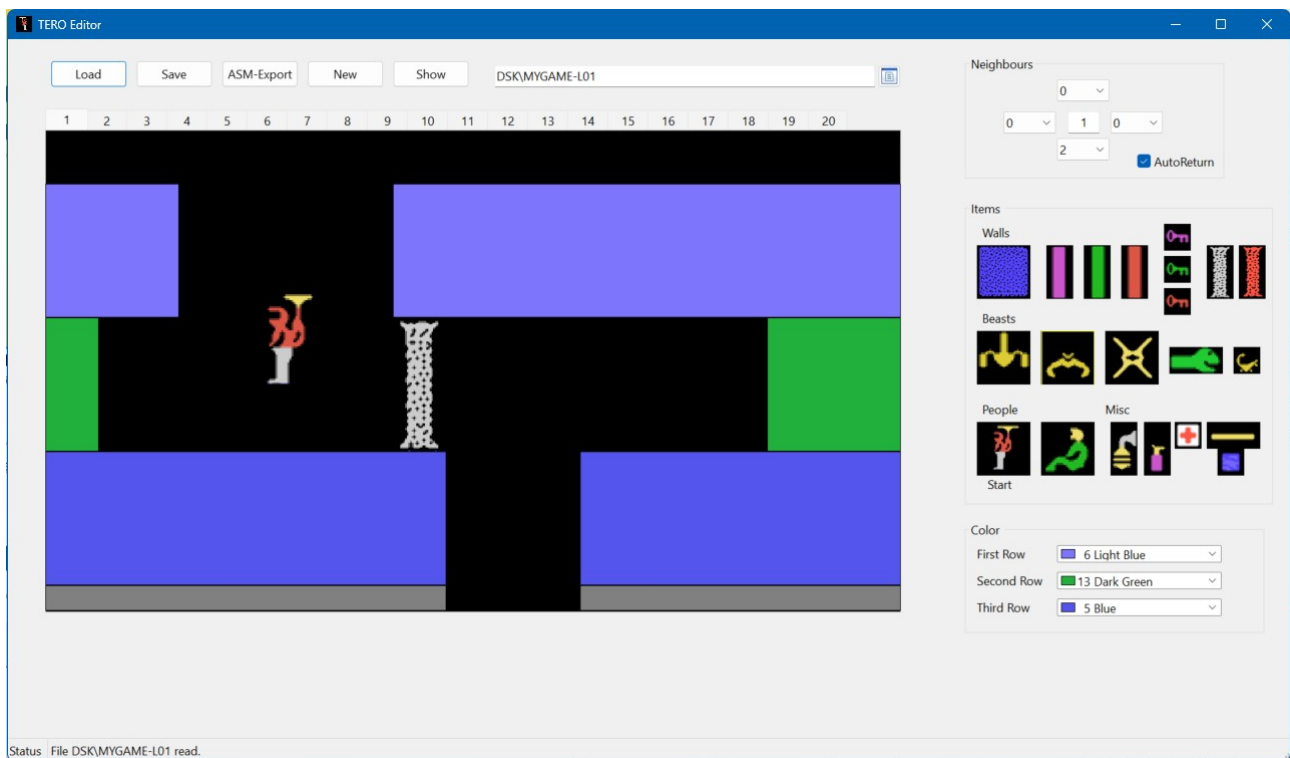
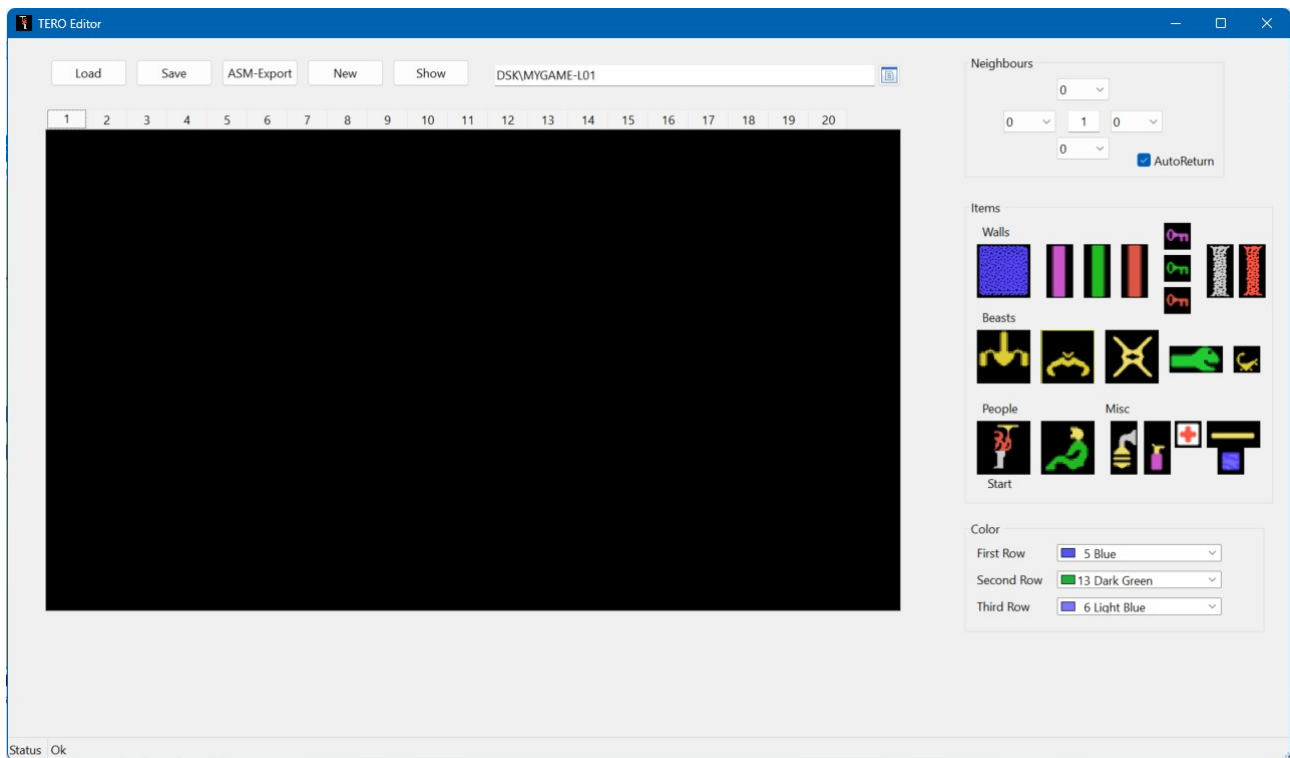


Make sure you start with “?W.”, which will make Classic99 read the windows file as a Display/Variable file. We will see later, how to convert the Windows-files to TI-FILES.

## Use the T.E.R.O. Level Editor

Start `TERO_Edit.exe` in Windows, select the level file in your game directory, then press „Load“





You could also start without loading a file, just enter a filename and press “Save”.

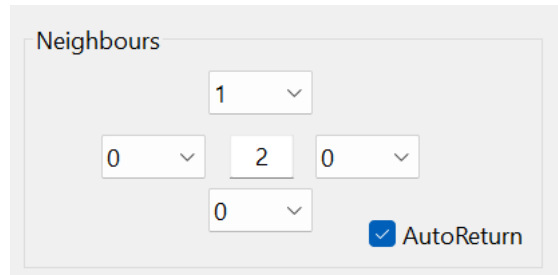
A T.E.R.O. Game consists of multiple Levels. Each level is stored in a separate file, starting with the suffix -L01 up to the last level, obviously -L99 is the highest possible number.

## Navigating the screens

Each Level has up to 20 screens. In the Level Editor you may use the tabs above the screen to switch between screens. You may also navigate between screens in the “Neighbors” section on the

top left of the screen. In the middle, the current screen number is shown. You may use the drop-down menus to specify the neighbor screens, up/down/left/right. A zero stands for no neighbor is this direction. In above's example, the is only one adjacent room: Below is number 2.

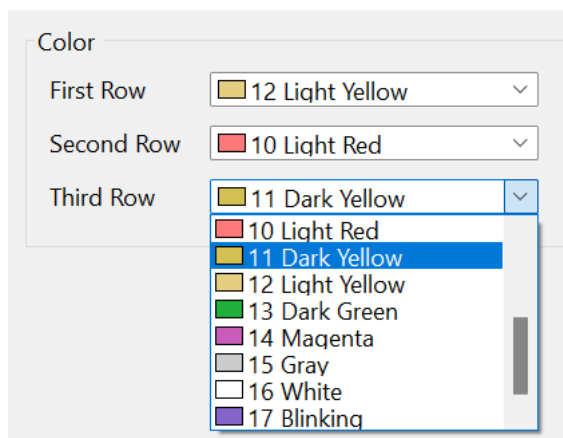
You may click the tab "2" or simply right-click on the drop-down menu with the "2". If the "AutoReturn" flag is set, the way back is automatically included. In this case, in screen "2" will be a "1" in the "Up" field.



The 'Neighbours' panel contains four directional drop-down menus arranged in a cross shape. The 'Up' menu is set to '1', the 'Down' menu is set to '2', the 'Left' menu is set to '0', and the 'Right' menu is set to '0'. To the right of these menus is a checked checkbox labeled 'AutoReturn'.

**Leave no gaps** between screens, use the numbers 1 to n for n screens.

Each screen has three rows. You may select the colors in the lower right of the screen.



The 'Color' panel shows three rows of color selection. The 'First Row' is set to '12 Light Yellow', the 'Second Row' is set to '10 Light Red', and the 'Third Row' is set to '11 Dark Yellow'. A list of color options is shown for the 'Third Row', including '11 Dark Yellow', '10 Light Red', '12 Light Yellow', '13 Dark Green', '14 Magenta', '15 Gray', '16 White', and '17 Blinking'.

Color 17 is not actually a color, but makes the walls blink red and become deadly when touched.

You will notice a gray bar above and/or below the three rows, indicating the rows directly above or below in the adjacent room. When playing, you can only leave the screen when there is no wall in the adjacent screen, but this is not visible. It is your decision to paint a corresponding wall or leave the player guessing the actual exit.

## Designing a level

You can **add items** by clicking them, they will appear in the middle of the screen. From there you can drag them wherever you like. You may click several items to stack them first before placing them on the screen.

You can **delete items** by dragging them to the bottom of the black playfield.

Some items can be used multiple times, others just once per screen or level.

The last action will always be reported on the status line.

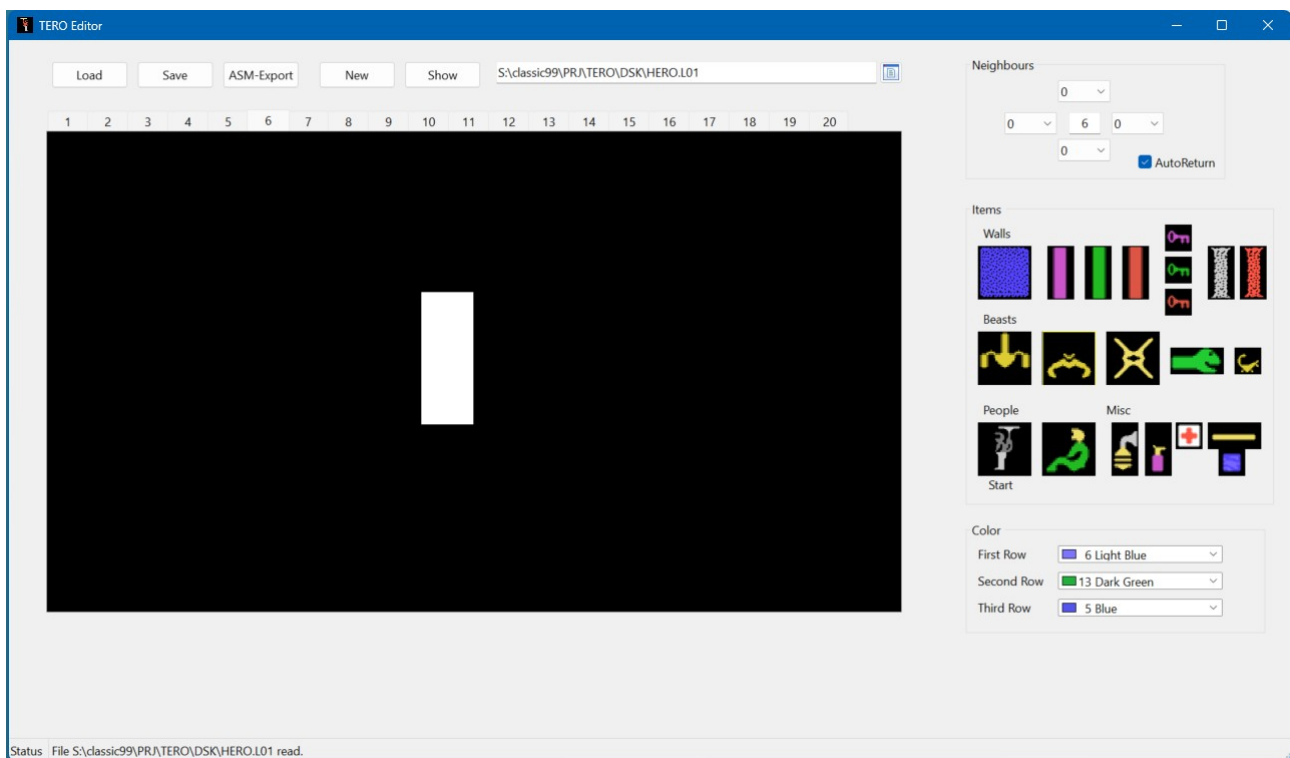
## Walls



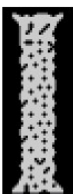
Walls are the main element of a screen. They come in three rows and each row may have a separate color, selected on the bottom-right.

A new wall will be created as a white block. Drag it to the desired place, where it snaps into position and gets its assigned color. The minimum width of a wall is two characters. You may right-mouse-click on a wall and drag it to the right or left to enlarge or reduce the width.

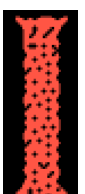
Plan the path of the player. Make sure that any open row to the left or the right is matched in the adjacent room. There is no check that you can actually leave the room there. In the up/down direction, there is a check. You can only leave the room vertically, if there is no wall in the adjacent room. You may use this to let the player search for the invisible exit. There are only shown in the editor for the support of the design.



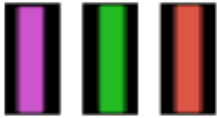
## Barriers



There are two types of barriers, gray and red. Both can be eliminated with a grenade. The red barrier is deadly for the player.



## Doors and Keys



There are three types of doors with three corresponding keys to open them.



## Animals



- Spider
- Bat – the yellow box indicates area of movement
- Moth – the yellow box indicates area of movement, only one per screen allowed
- Snake
- Little Scorpion

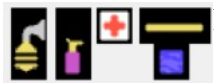
All animals are deadly when touched.

## People



Only on the first screen the player can be positioned. There may be multiple victims to be rescued, but the first one found ends the level.

## Other Items



From left to right:

- Lamp – Easily destroyed, it gets dark when touched. But the player has a night-vision that gets activated in the dark
- Additional grenades
- Medi-Kit for additional energy points
- Raft – only one per screen - moves horizontally between the left and right bounds. Bounds are evaluated on startup of the screen and remain unchanged, even when removed by the player. The raft starts moving towards the middle of the screen.
- Water – Under the third row the level can be terminated by water, which is deadly to touch

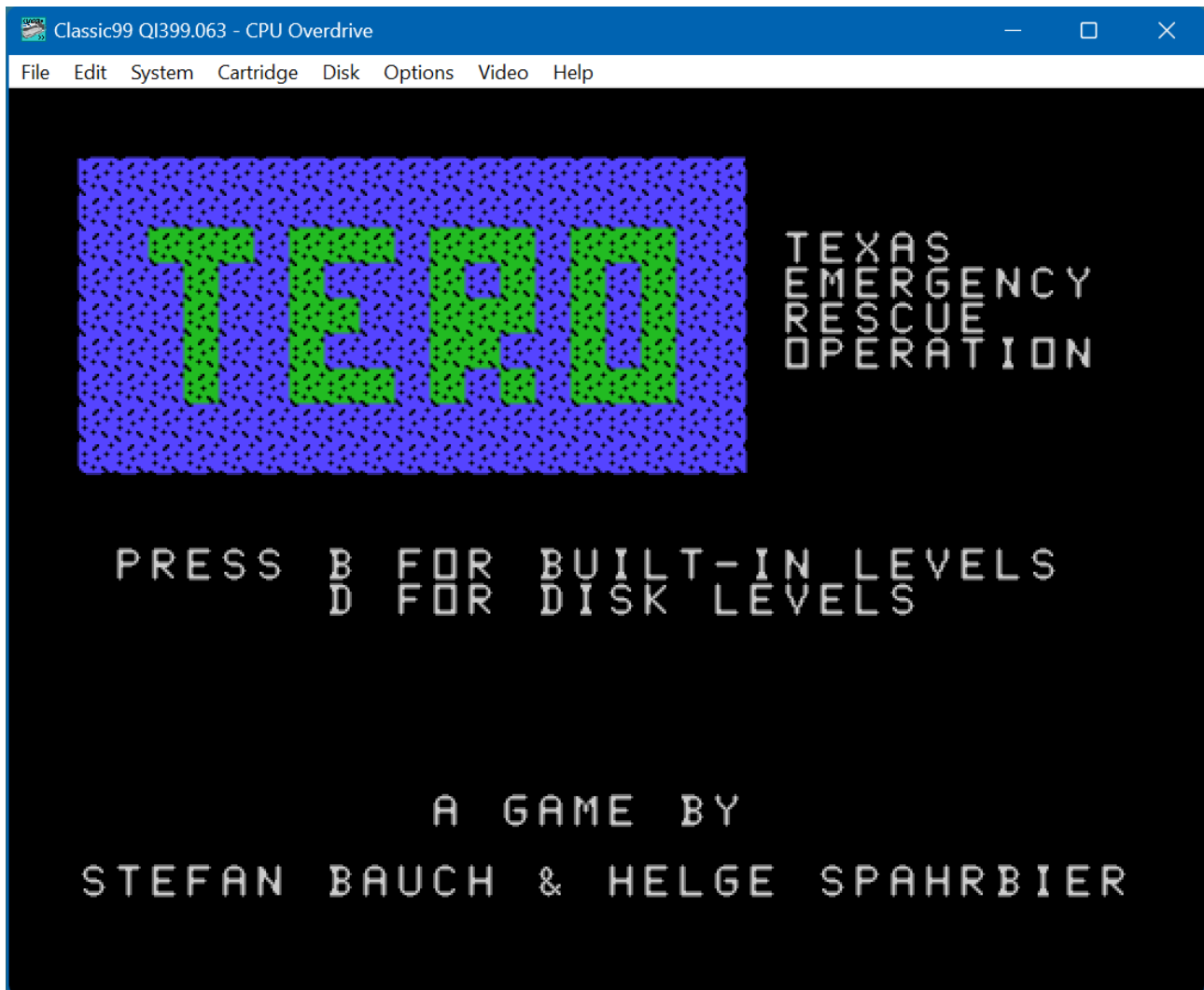
## Hints on designing a T.E.R.O game

- Each item takes up memory ... avoid too many items. The game might crash when running out of memory
- Mind the 10 character filename size when working with disk images. Classic99 is more tolerant with FIAD files.

## Testing your game

The level editor writes regular windows text-files. Classic99 is able to load them directly when using the prefix ?W. in the filename. Specify this in the menu using EDCAT.

Insert the module file TERO-DEV.BIN into Classic99 and start the module:



Press D for DISK LEVELS and select the drive with your files. This opens the catalog file TERO.CAT.





You may now select your game from the menu. There is now error-checking, configured file with suffix -L01 must exist.

The DEV developer version let you now select a level to test.

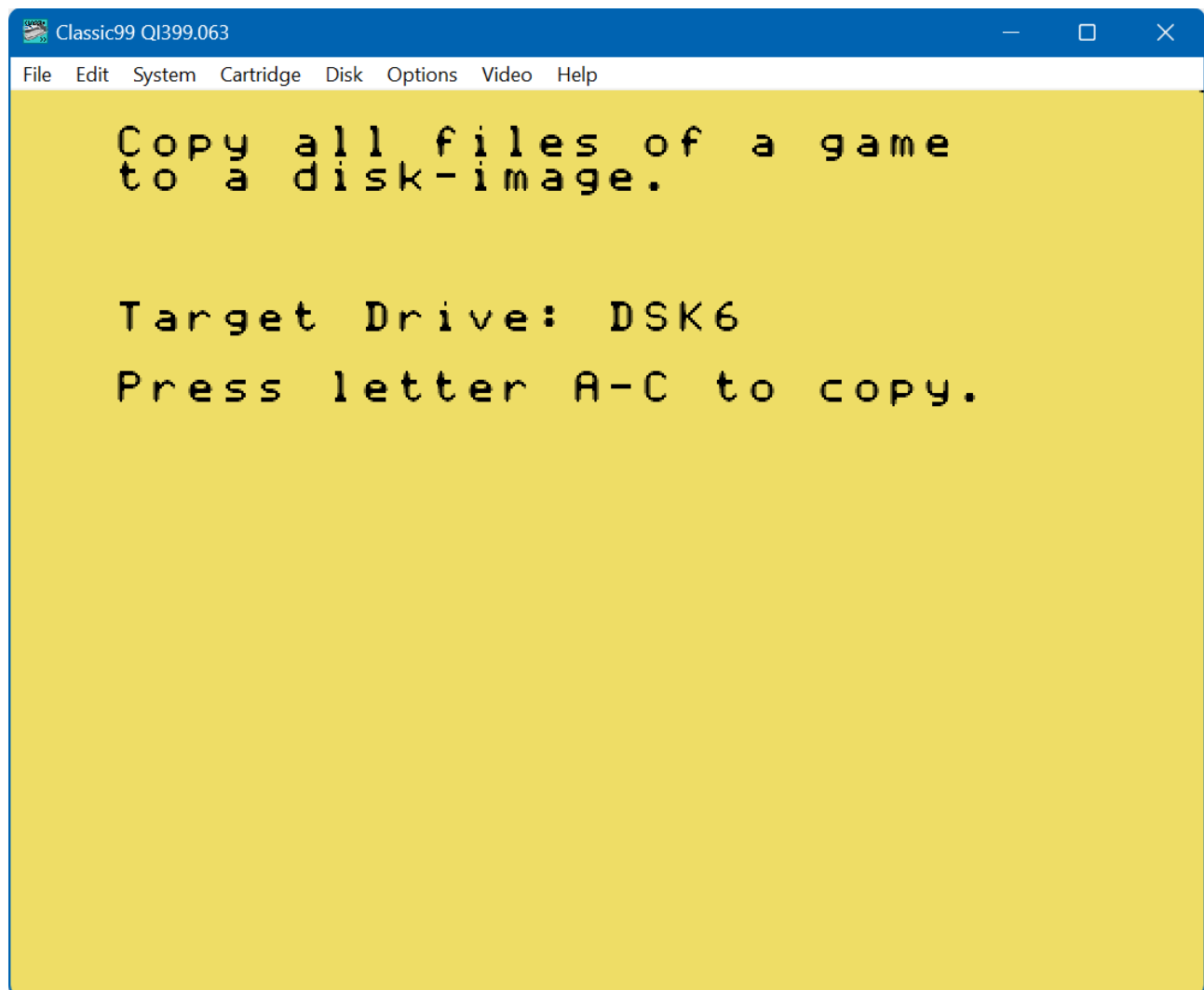
The game ends when the next level file can not be found.

## Prepare for release

When you tested your levels and want to convert the Windows files to TI-FILES or a Disk-Image, use the “P to Prepare” option of EDCAT. You need to remember the letter in the menu before your finished game.

Prepare a second drive in Classic99, i.e. DSK6 with either a FIAD or a disk-image target to copy the files to. Specify TI-FILES header for a FIAD drive.

Then press “P” in the menu.



After selecting the target drive, select the game by entering the letter of the game in your menu. The files are now copied to the target and can now be used without the ?W in the filename, on the real hardware or other emulators.