
Adventuron-2-PAW User Guide



Figure 1. By Bill Bertram - Own work, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=170050>

A guide to porting Adventuron games (using a limited set of features) to run on the [ZX Spectrum](#)¹ 48K / 128K microcomputer by translating Adventuron source code to InPaws source code and using the [Professional Adventure Writer](#)² engine, © Gilsoft - 1986.

1. Introduction

Adventuron now has "early access" functionality for exporting certain adventures to be played on ZX Spectrum, using the PAW adventure engine (1986).

This functionality is still a little bit rough around the edges, and will be improved in future, but the takeaway should be that if you code a simple adventure game, that you can mostly just run the game on the Spectrum (assuming it doesn't break size constraints).

¹ https://en.wikipedia.org/wiki/ZX_Spectrum

² https://en.wikipedia.org/wiki/Professional_Adventure_Writer

It is important to keep testing the adventure as it is in development with "Adventuron 2 PAW" (via copy paste between browser tabs) to make sure that you are not using an unmapped feature, or are not greatly exceeding PAW's memory constraints. A future version of Adventuron may integrate these checks, depending on how popular this feature is.

Adventuron 2 PAW is available here : <https://eeyo.io/adventuron2paw>

1.1. Words of Warning



Adventure 2 PAW has a similar panel layout to Adventuron, but is a completely different application in a different browser tab.



Adventuron 2 PAW currently has no state, so code edits will not be saved if you edit on the left hand side, and will be lost when the browser tab closes.



It is absolutely important that you don't use a new function in Adventuron without checking that it works in Adventuron 2 PAW if you plan to port the game.



The ported game is not GUARANTEED to work, and this application is best-efforts right now. As a rule of thumb, most of the functions of the Adventuron Classroom tutorial are ported, except for doors.



PAW is copyright of Gilsoft and if you wish to distribute a game with PAW, then a license is required. Follow @Timbucus (Tim Gilberts on Twitter for more information).



Adventuron 2 PAW does not redistribute the PAW utility and Adventuron Software Limited does not accept any legal liability for misuse of this tool.



Please note that PAW is more capable than the mapped features suggest. The mapped features are the simple overlap.

2. Pre-requisite knowledge required

You should be familiar with the file system on Windows computers, and how to edit text files, and how to type commands in a console window.

If you are not familiar with these activities, then a simplified process is forthcoming later in the year.

3. Required Downloads

This procedure is only currently followable on Microsoft Windows PCs, and requires the following downloads. All of these tools have non Windows alternatives, but this tutorial only focuses on Windows.

A PAW license is required if you are planning to release a game commercially. Please ask @TimBucus on Twitter (Tim Gilberts) for more information.

- InPaws (Windows only) - This is a PAW compiler for Windows.
- A licensed copy of PAW (available to download [here](#)³ - but read the conditions at the bottom of the page), especially if you plan to release the game commercially. PAW is still copyright protected commercial software.
- A licensed copy of [Spectaculator](#)⁴ (Windows only) - this is a spectrum emulator.
- [Visual Studio Code](#)⁵.

4. Preparing The Environment (Windows PCs Only)

- Step A. Decompress the InPaws zipfile to a folder (such as c:\inpaws or d:\inpaws)
- Step B. Download Spectaculator, and install from the msi file (trial license available for 28 days then full license is reqable to download [here](#)⁶ - but read the conditions at the bottom of the page), especially if yuired). Other emulators will work too, but this tutorial assumes Spectaculator.

³ <https://8-bit.info/the-gilsoft-adventure-systems>

⁴ <http://www.spectaculator.com/downloads/>

⁵ <https://code.visualstudio.com/download>

⁶ <https://8-bit.info/the-gilsoft-adventure-systems>

- Step C. Download VSCode and install. After installation, load VSCode, then press control + shift + x to search for extension to install, then type "inPaws", then click the "Install" button to install.
- Step D. Load "Paw_A17C_A.tap" file into spectaculator (double clicking the tap file in windows should load it in spectaculator). MAKE SURE YOU HAVE LICENSE IF YOU WILL BE REDISTRIBUTING GAMES PRODUCED WITH PAW.

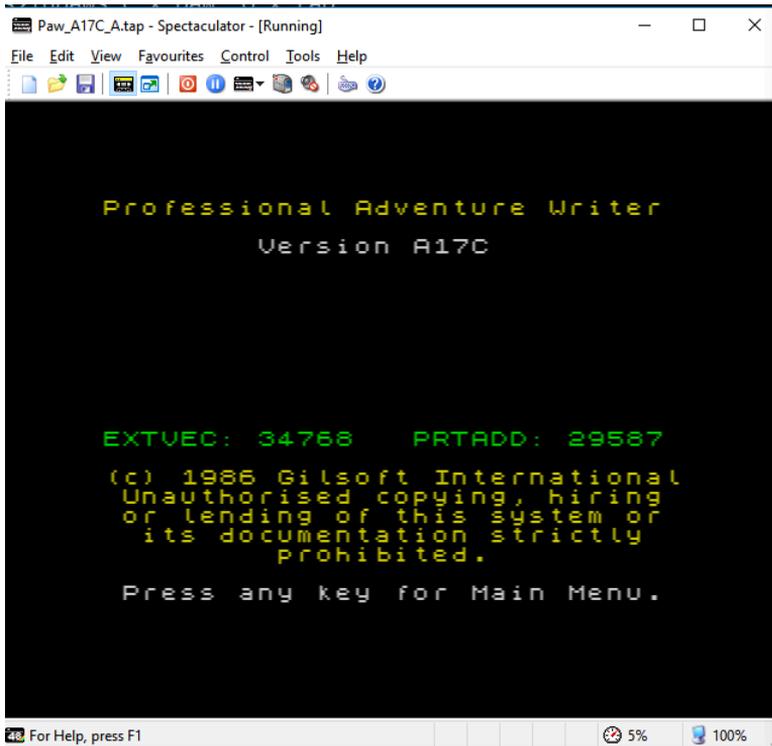


Figure 2. PAW A17C loaded into Spectaculator

Press *SPACEBAR*, then *E*, then *ENTER*, then *J*, then *ENTER*, then *X* , if you followed the instructions exactly, then you will see the screen below....

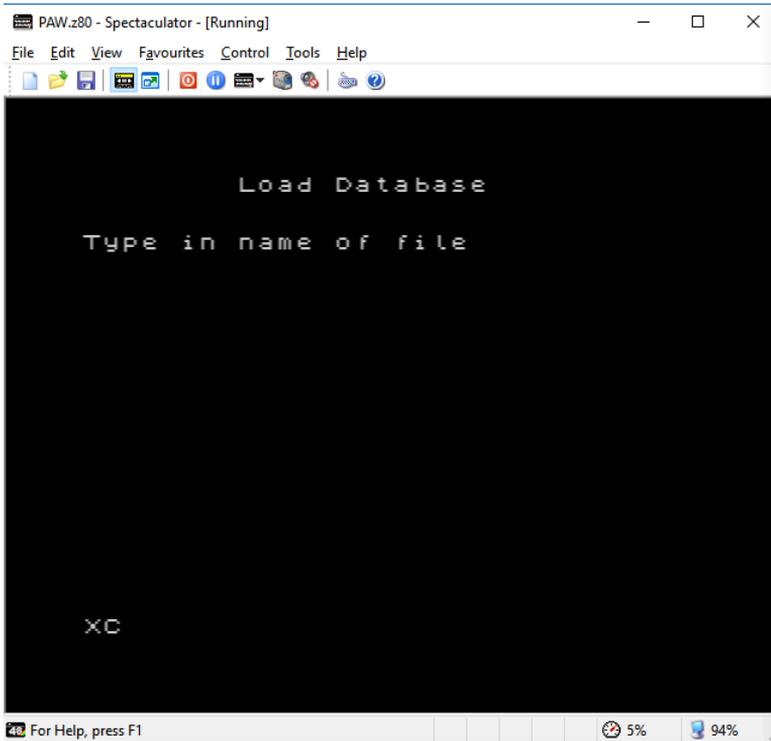


Figure 3. Ready to load data file X from tape (we will create a snapshot here)

Next, go to the "File / Save As" menu option.

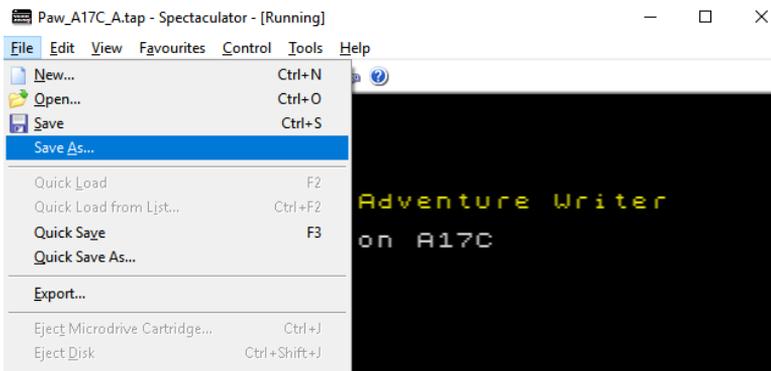


Figure 4. File / Save As

Now, navigate to the inpaws folder (from step A), select Save As Type .z80, then type filename PAW.z80 and then click the save button.

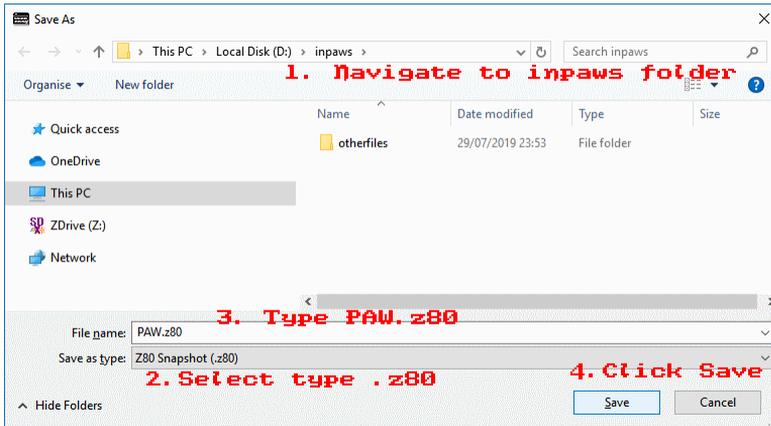


Figure 5. Saving PAW.z80

- Step E. Open A Command Prompt Window (press windows button then type 'cmd' then press enter) and navigate into the InPaws folder (leave this window open for use later).

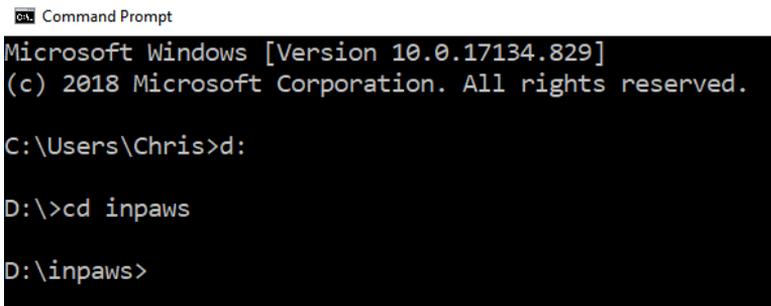


Figure 6. Navigating to the inpaws folder

- Step F. In VSCode, create a new text file, and save as **x.paw**. (the 'x' part is arbitrary but we will use it later).

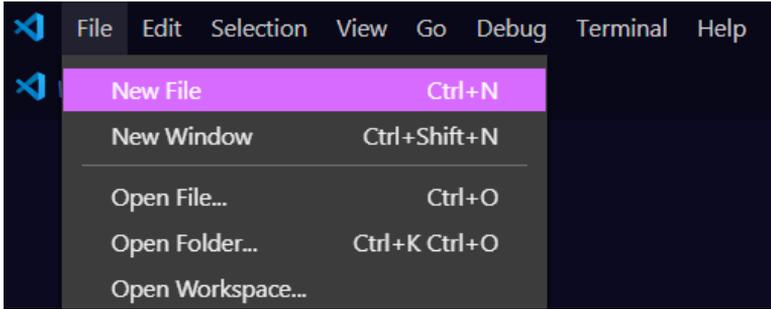


Figure 7. Creating A New File (VS Code)

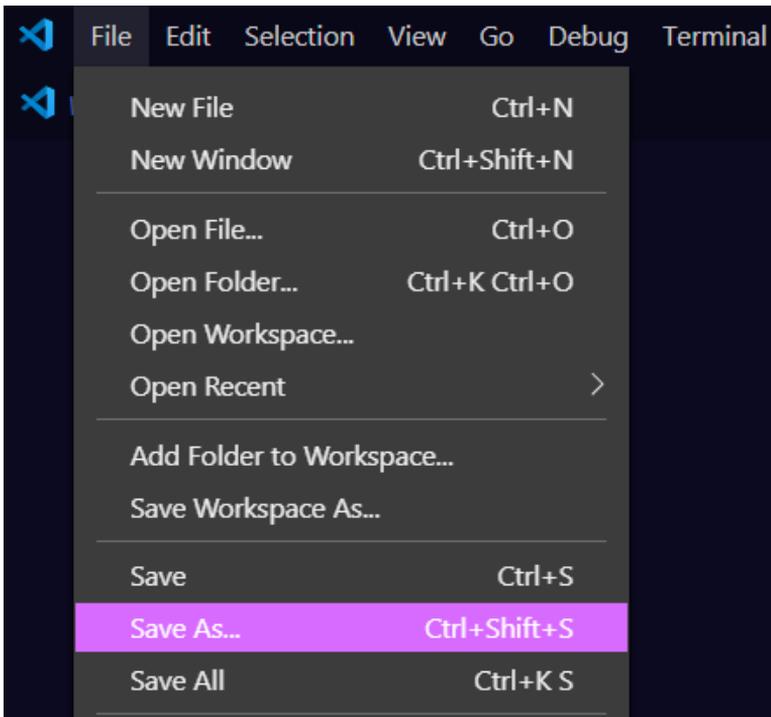


Figure 8. File / Save As (VS Code)

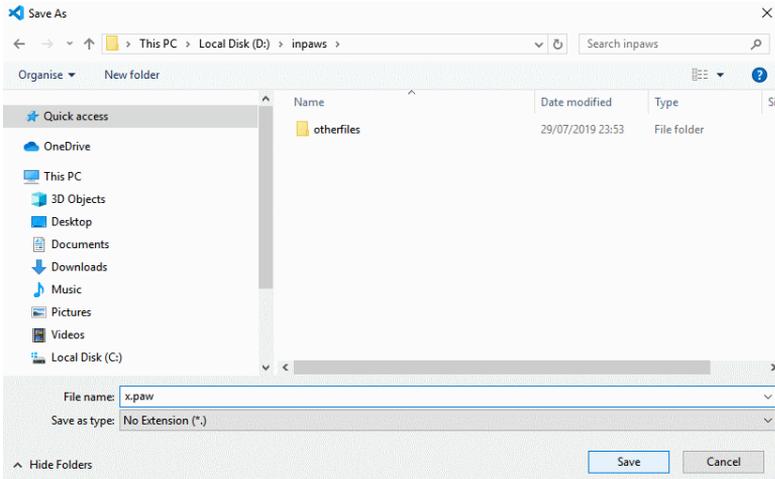
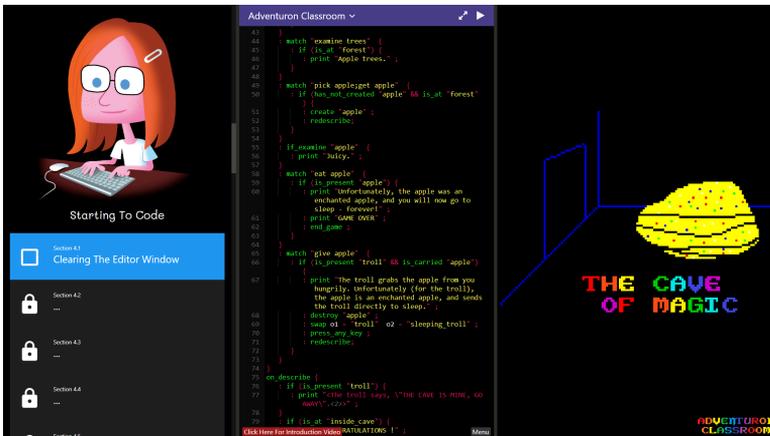


Figure 9. Save as x.paw in the inpaws folder

5. Porting Procedure

5.1. Create An Adventure Game

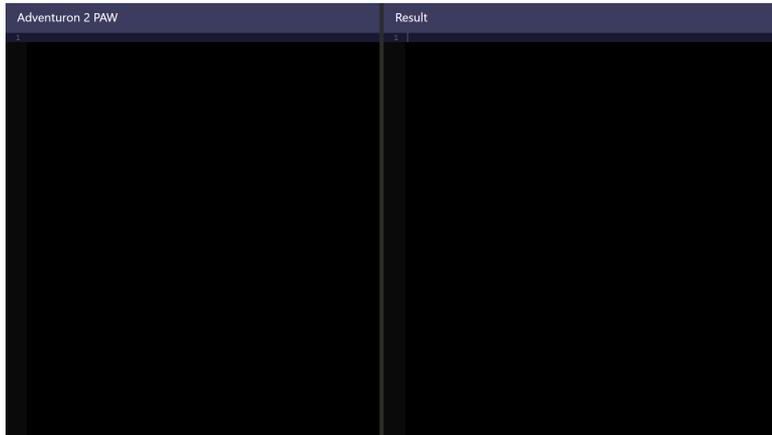
Step 1 - Start writing Adventuron game (editor at <https://www.adventuron.io/classroom>) using only the compatible functions shown later in the document.



Regularly step through steps 2 to 4 as you are developing (a PAW compatibility mode will be added to Adventuron in the future).

5.2. Creating The PAW Database

Step 2 - In a new browser tab, navigate to <https://eeyo.io/adventuron2paw> , and copy the adventuron source code from Adventuron into the left hand side editor in Adventuron 2 PAW.



Step 3 - Look for error messages on right hand side, and if present, then resolve inside Adventuron tab, then go back to step 2.

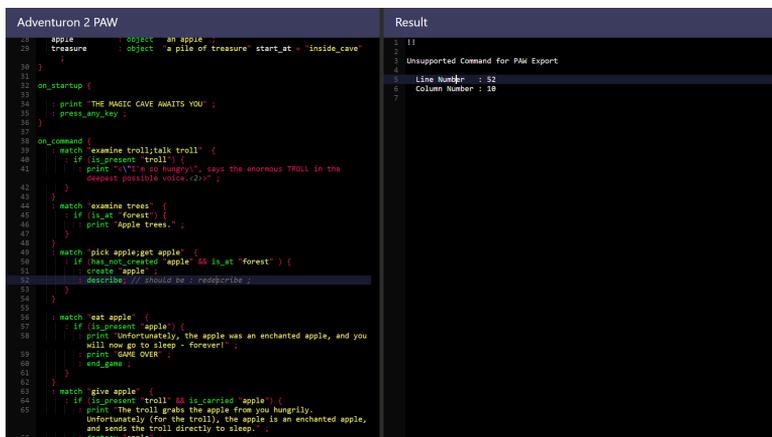


Figure 10. With Error On Line 52

If no error messages are show, then go to step 4.

Adventuron 2 PAW	Result
<pre> 1 start_at = lakeside 2 loading_screen = loading_screen 3 locations { 4 forest location "You are on the forest path.\nTall trees tower over you on both sides." ; 5 outside_cave location "You are standing outside the CAVE OF MAGIC(5)." ; 6 inside_cave location "You are inside the CAVE OF MAGIC(5)." ; 7 lakeside location "You are by the side of a beautiful lake." ; 8 } 9 connections { 10 from_direction to = { 11 lakeside north forest; 12 forest north outside_cave; 13 outside_cave north inside_cave; 14 } 15 } 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 </pre>	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 </pre>

Figure 11. No Errors

Step 4 - With the InPaws source code showing on the right hand side, then paste that code into the VSCode tab representing the **x.paw** file in the inpaws folder(as created in preparation step F).

Step 5 - Using the command line window (opened in preparation step E), type the following command .

```
inpaws c x.paw -o x.tap
```



If you have already typed this command before, press the up cursor button on the keyboard, to cycle through previously typed commands, then press enter to resubmit.

```
Command Prompt

D:\inpaws>inpaws c x.paw -o x.tap
Compilation successfully finished
Resulting file: x.tap
Free memory after compile: 9744
-----
23      locations
48      objects
143     messages
55      system messages
25      processes
1       character sets
Saved by compression: 8118
-----

D:\inpaws>
```



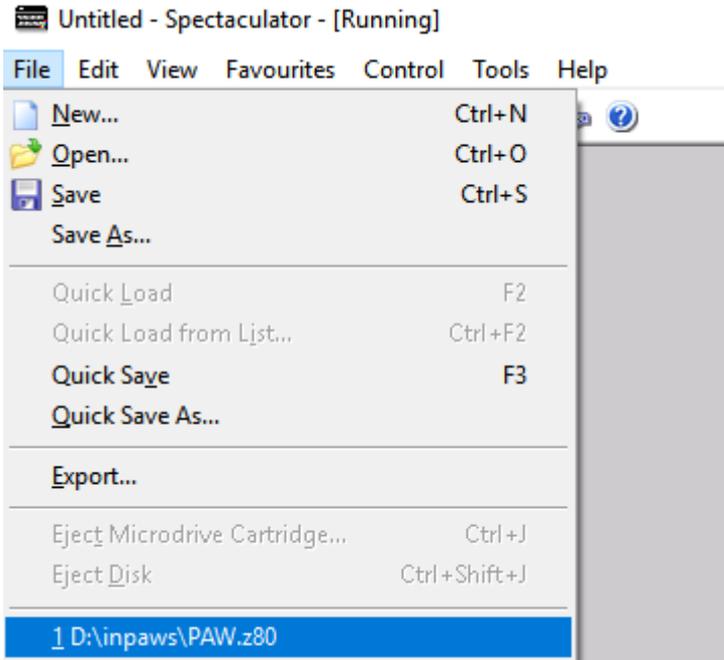
It is quite possible to see an error here, if there is a bug, please report it to info@adventuron.io⁷.

5.3. Exporting As A Standalone .tap file

Step 6 - Loading The Database

Load Spectaculator (using start menu), and in the recent file list (under the file menu), select PAW.z80 (created in step D) - or navigate to PAW.z80 in the inpaws folder.

⁷ <mailto:info@adventuron.io>



We now have PAW loaded at the point at which it will be looking for a PAW database on tape, we now need to insert a virtual tape into the virtual ZX Spectrum.

Either:

(6.1) If have not loaded x.tap before, then go to the "File / Open" menu option, and navigate to the x.tap file in the inpaws folder, and select the file

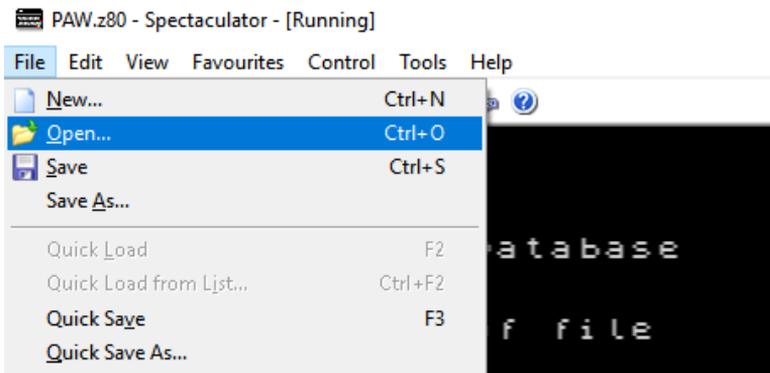


Figure 12. Selecting the x.tap file for the first time

Or (6.2) If you have previously loaded the x.tap file into the emulator, just select it from the recent files list.

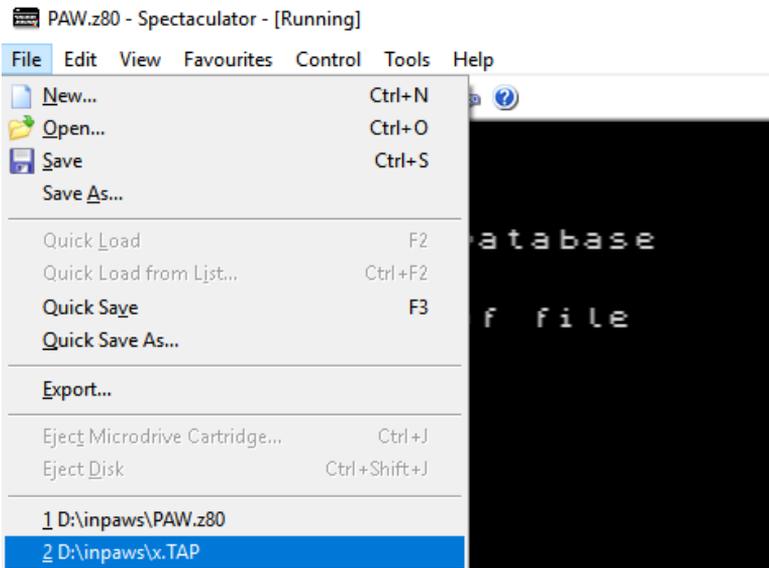
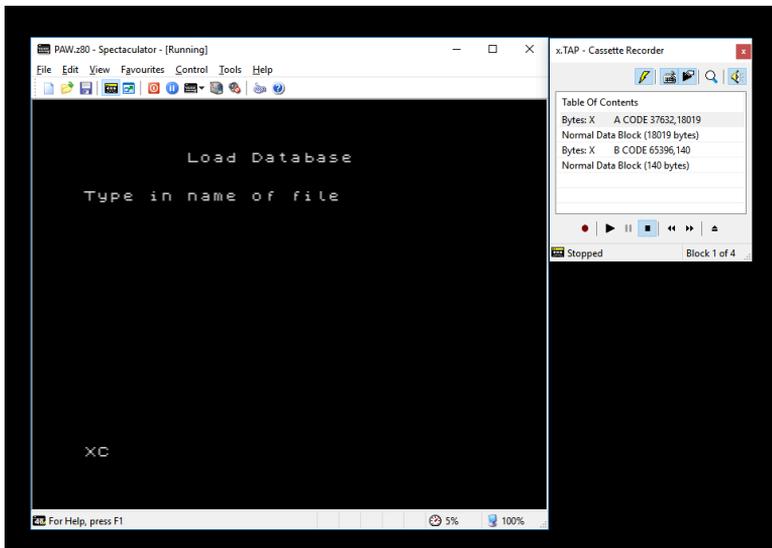


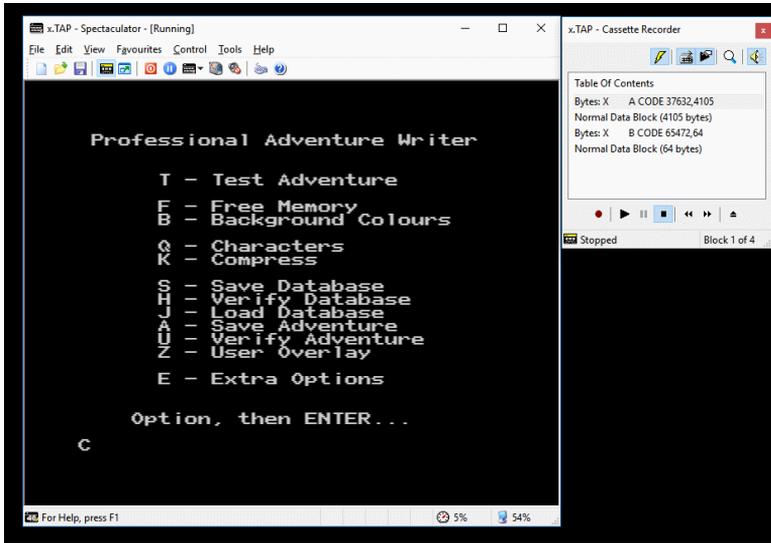
Figure 13. Selecting the x.tap file for the first time

ONCE THE TAPFILE IS LOADED, YOU SHOULD SEE SOMETHING LIKE THIS:



The "cassette recorder" window will show four lines, and this is the adventure database ready to load into the PAW utility.

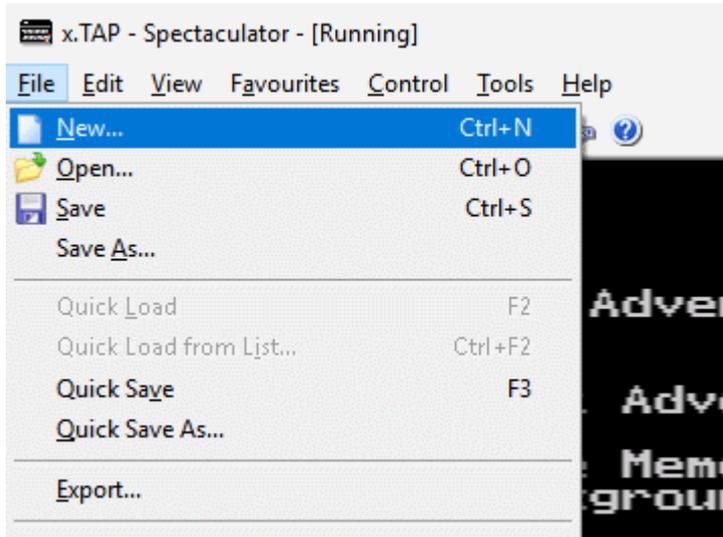
Now press ENTER to load the game - and upon loading you should see the following screen, and typically, the font will change to the font of the adventure game (in this screenshot, the adventure is using the Plotter Bold font by Damien Guard), but the font displayed will depend on the font in the x.paw file.



Step 7 - Inserting A New (Virtual) Tape

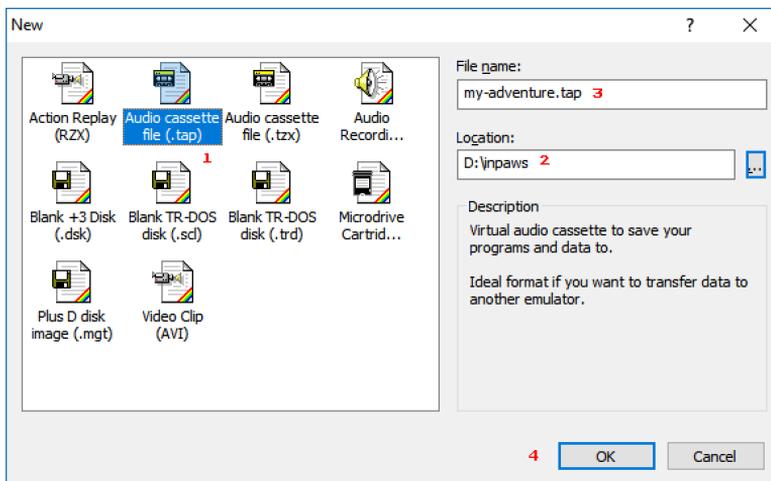
To save the standalone adventure game, we first need to "insert" a new virtual tape.

In Spectaculator, first select "File / New"



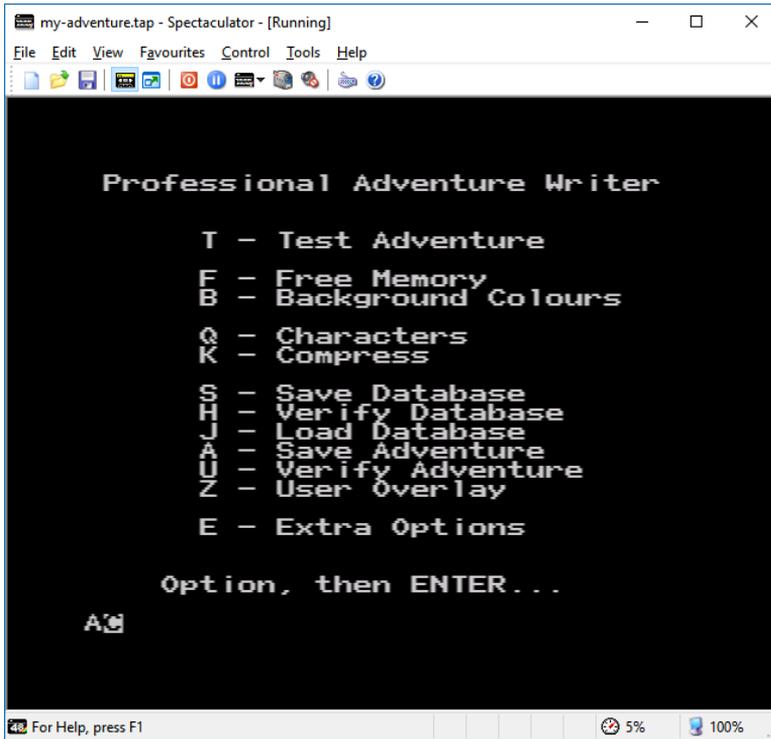
In the "New" window, (1) select Audio cassette file (.tap), the (2) select the inpaws folder then (3) type my-adventure.tap then (4) click OK.

This will place a virtual tape file in the tape player ready to record the standalone game data on the virtual tape.

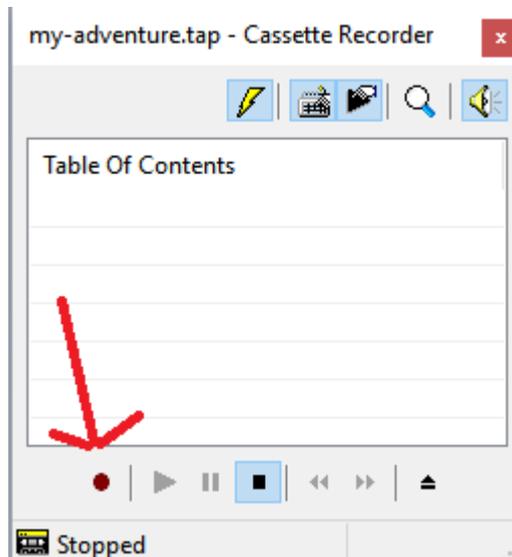


Step 8 - Saving A Standalone Game

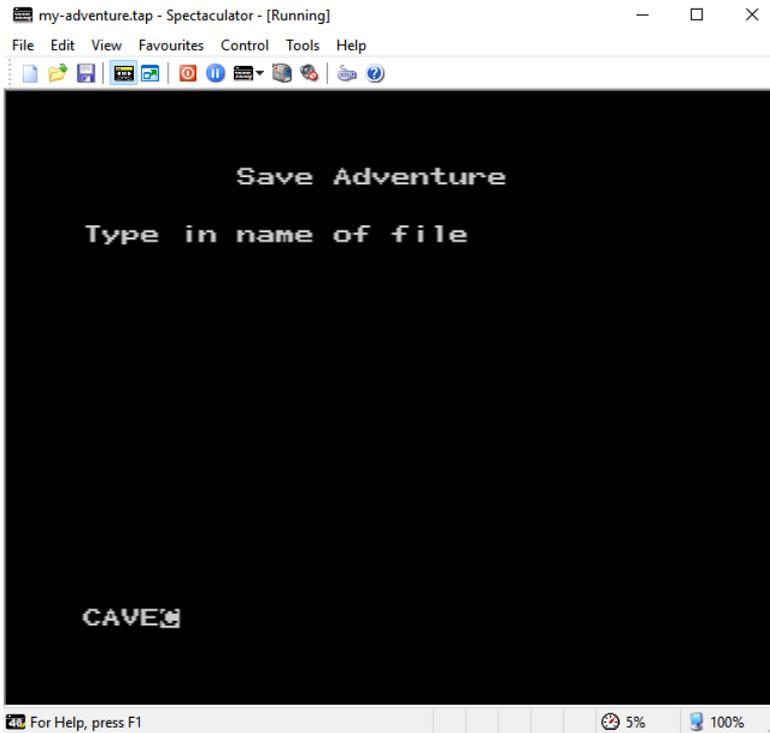
Now, at the PAW menu screen, type A, then press ENTER.



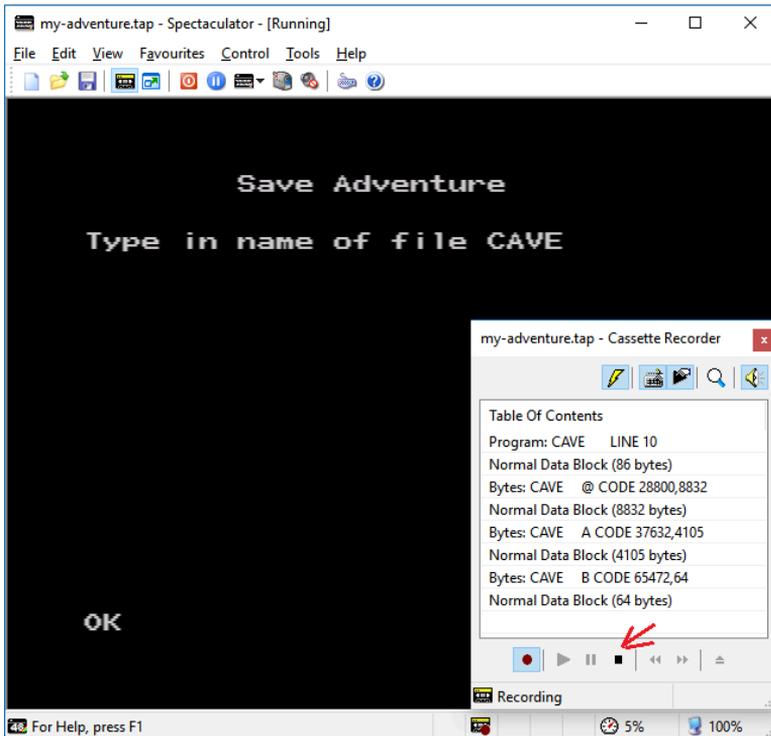
Now, click the RECORD button on the virtual tape recorder.



In the SAVE ADVENTURE screen, type the name of your game (such as "CAVE" in this example), then press ENTER.



After saving the game, the OK message will be displayed in the Spectaculator window, and after you see this, press the STOP button on the virtual cassette recorder window (as shown):



Your ZX Spectrum game is now saved inside the tapfile named my-adventure.tap. The tapfile includes the PAW database, and the PAW engine. Again, remember that you should own a copy of PAW in order to distribute games utilizing the engine.

And that's that!

Tutorial Complete !

6. Debugging

This converter is extremely beta, so bugs likely remain. You can test the ZX Spectrum version of your game either from the produced tapfile or if instead of step 6, you can select menu option T (then press ENTER) from the PAW menu to test the game inside the PAW application.

You can test diagnostics this way too, but the subject is too large to cover here.

7. Compatible Features

7.1. Compatible Variable Types

- boolean
- integer

7.2. Compatible Boolean Conditions

- is_at
- is_present
- is_carried
- is_worn
- is_beside
- is_exists
- ! (Not operation)
- && (And operation)
- || (Or operation)
- P1 == P2
- P1 == P2
- P1 < P2
- P1 # P2
- P1 > P2
- P1 >= P2

7.3. Compatible Commands

- beep (currently buggy)
- clear_screen
- create
- destroy
- done

- drop
- end_game
- get
- goto
- inventory
- match
- ok
- pause
- press_any_key
- print
- redescribe
- return
- set_false
- set_integer
- set_true
- swap
- turns

7.4. Additional Support - Barriers

The following barrier types are supported:

- block
- block_path

7.5. Markup

Text can be coloured in using the following format :

```
This is my <red<2>> text, this is my <cyan<5>> text, and this text is  
<white<15>>.
```

Paragraph separators can be inserted using \n (backslash then n).

This is paragraph one.\nAnd this is paragraph two.

Adventuron Standard Palette (Same as ZX Spectrum)

- 0 = black (#000)
- 1 = blue (#00c)
- 2 = red (#c00)
- 3 = magenta (#c0c)
- 4 = green (#0c0)
- 5 = cyan (#0cc)
- 6 = yellow (#cc0)
- 7 = grey (#ccc)
- 8 = black (#000)
- 9 = bright blue (#00f)
- 10 = bright red (#f00)
- 11 = bright magenta (#f0f)
- 12 = bright green (#0f0)
- 13 = bright cyan (#0ff)
- 14 = bright yellow (#ff0)
- 15 = white (fff)



Adventuron has remappable palettes, but the PAW exported will not remap colours.

NOTE : Doors (in the barriers section) are not yet supported. They can still be directly coded however using flags.

IMPORTANT : The PAW engine itself is greatly more capable than this subset of commands would suggest and additional mapped commands will likely be added in the future.

8. "It's too much hassle "

If 8-bit is your primary target, then it may be better to write in InPaws directly, then use Antur to convert to DAAD source, then use [Daad 2 Adventuron](#)⁸ to port to Adventuron.

This process will improve with time if there is interest in this conversion tool (no point investing the effort if there is no interest).

9. Other Converters?

9.1. *What about Adventuron 2 DAAD?*

Adventuron 2 DAAD will be coming soon.

In the meantime, the ANTUR project should be able to port an InPaws game to DAAD format. Check out the ANTUR project [here](#)⁹.

One advantage will be that (with the use of the maluva plugin) it may be possible to pass through Adventuron images as long as they are ZX attribute / resolution compliant. One disadvantage is that text colouring may not be supported initially.

9.2. *What about DAAD 2 Adventuron?*

Available [here](#)¹⁰ ?

9.3. *What about PAW 2 Adventuron or Quill 2 Adventuron or GAC 2 Adventuron ?*

If you are the original author of a PAW/QUILL/GAC game, then contact info@adventuron.io¹¹ and a conversion to Adventuron sourcecode can be arranged.

This is not being publicaly released in order to restrict lazy or unauthorised ports, especially because the converter requires a decompilation step.

⁸ <https://eeyo.io/da>

⁹ <https://github.com/daad-adventure-writer/antur>

¹⁰ <https://eeyo.io/da>

¹¹ <mailto:info@adventuron.io>

Some games are currently not supported if they use non mapped CONDUCTS.

10. Sample Adventure Game

If you want some simple code to test with, then here is a small single puzzle adventure game that can be copied and pasted into "Adventuron 2 PAW" (and Adventuron Classroom).

10.1. THE CAVE OF MAGIC

The Cave Of Magic, © Chris Ainsley.

```

start_at          = lakeside

game_information {
  game_name       = The Cave of Magic
  game_shortcode  = Magic Cave
  written_by      = Chris Ainsley
  year            = 2019
  short_synopsis  = Find the treasure
  game_version    = 1.0.0
}

locations {
  forest          : location "You are on the forest path.\nTall <TREES<4>>
tower over you on both sides." ;
  outside_cave   : location "You are standing outside <THE CAVE OF
MAGIC<5>>" ;
  inside_cave    : location "You are inside <THE CAVE OF MAGIC<5>>" ;
  lakeside       : location "You are by the side of a <BEAUTIFUL
LAKE<2>>." ;
}

objects {
  troll          : scenery "an enormous troll" start_at = "outside_cave" ;
  sleeping_troll : scenery "an enormous troll (sleeping)" ;
  apple          : object  "an apple" ;
  treasure       : object  "a pile of treasure" start_at = "inside_cave" ;
}

connections {
  from, direction, to = [
    lakeside,      north, forest,
    forest,        north, outside_cave,
    outside_cave, north, inside_cave,
  ]
}

```

```
barriers {
  block_cave : block {
    location          = inside_cave
    message           = THE TROLL IS GUARDING THE CAVE.
    block_when_exists = troll
    show_blocked_exit = true
  }
}
on_startup {
  : print "\^c^THE MAGIC CAVE AWAITS YOU" ;
  : beep millis = "100" pitch = "0" ;
  : beep millis = "100" pitch = "2" ;
  : beep millis = "100" pitch = "4" ;
  : beep millis = "100" pitch = "6" ;
  : press_any_key ;
  : beep millis = "100" pitch = "6" ;
  : beep millis = "100" pitch = "4" ;
  : beep millis = "100" pitch = "2" ;
  : beep millis = "100" pitch = "0" ;
}
on_command {
  : match "pick apple;get apple" {
    : if (is_at "forest" && has_not_created "apple") {
      : create "apple" ;
      : redescribe;
    }
  }
  : match "examine trees" {
    : if (is_at "forest") {
      : print "Apple trees." ;
    }
  }
  : match "examine troll;talk troll" {
    : print "<\<"I'm so hungry\<3>>, says the enormous TROLL in the
deepest possible voice." ;
  }
  : match "give apple" {
    : if (is_present "troll" && is_carried "apple") {
      : print "The troll grabs the apple from you hungrily.
Unfortunately (for the troll), the apple is an <ENCHANTED APPLE<12>>,
and sends the troll directly to sleep." ;
      : destroy "apple" ;
      : swap o1 = "troll" o2 = "sleeping_troll" ;
      : press_any_key ;
      : redescribe;
    }
  }
}
```

```

}
: match "eat apple" {
  : if (is_present "apple") {
    : print "Unfortunately, the apple was an <ENCHANTED APPLE<12>>,
and you will now go to sleep - forever." ;
    : print "\r^\<GAME OVER<2>>" ;
    : end_game ;
  }
}
}
}
on_tick {
  : if (is_at "inside_cave" ) {
    : beep millis = "200" pitch = "0" ;
    : beep millis = "400" pitch = "10" ;
    : print "\r^\<CONGRATULATIONS !" ;
    : print "\r^\<YOU WON THE GAME !" ;
    : print "\r^\<YOUR RANKING IS : JUNIOR ADVENTURER !" ;
    : press_any_key ;
    : clear_screen;
    : print "This tiny adventure was written using Adventuron." ;
    : print "Adventuron is a free text adventure creation language and
development system." ;
    : print "Visit the website, and make a world of your own." ;
    : print "\r^\<www.adventuron.io<12>>" ;
    : end_game ;
  }
  : if (is_present "troll") {
    : beep millis = "100" pitch = "-2" ;
    : beep millis = "100" pitch = "-4" ;
    : beep millis = "300" pitch = "-8" ;
    : print "The troll says, <\"THE CAVE IS MINE, GO AWAY\"<2>>." ;
  }
}
}

```

11. Bugs

Report bugs to info@adventuron.io¹²

12. Acknowledgements

- Adventuron 2 PAW uses various fonts (including Plotter Bold) by Damien Guard. These fonts has been marked as free for all uses including commercial,

¹² <mailto:info@adventuron.io>

and free to redistribute, as long as the font is not redistributed in isolation. Please check out the (frankly wonderful) fonts [here](#) ¹³.

- Adventuron 2 PAW does NOT embed any the PAW utility, or the PAW runtime, it is down to the implementer to ensure they have a valid license to use PAW when distributing a game using the game engine.
- Adventuron 2 PAW is copyright of Adventuron Software Limited.
- This document (PDF / HTML version) is created using AsciiDocFX and AsciiDoctorJ.
- Adventuron 2 PAW uses the following libraries : Ace Editor (MIT License) / Ace Editor GWT (MIT License) / GWT (Apache V2 License).

13. Final Words

It probably won't even work.

¹³ <https://damieng.com/typography/zx-origins/>