



A Halloween short story/adventure game for the 128K Sinclair ZX Spectrum
and modern web browsers

Written by [@Blerkotron](#)
(C)2018, Blerkotronic Software

Created with Professional Adventure Writer
©1986, Gilsoft International
<http://8-bit.info/the-gilsoft-adventure-systems/>

HTML version created with Adventuron
©2018, Chris Ainsley
https://medium.com/@model_train/introducing-adventuron-4f95909d6aaf

Find the latest version at
<https://blerkotron.itch.io/unhallowed>

Introduction

It's Halloween and you're on your way home from work, tired and hungry and wanting nothing more than to put your feet up in front of the TV. But something dark and evil lurks within your home, waiting for you to return...

Unhallowed is a classic text-adventure game with a few twists to make it more palatable for modern players.

Loading the game (ZX Spectrum version)

The game is compatible with 128k Spectrums *only* – i.e. a 128, +2, +2A or +3. It won't work on a 48k model, sorry!

128K/Plus 2/Plus 3

Use the LOADER function from the main menu, then play the tape.

The game will also work on your favourite Spectrum emulator, and has been tested as working (under emulation, at least) on the Pentagon 128 and Scorpion 256.

Loading the game (HTML version)

The HTML edition of the game requires a modern Internet browser with Javascript enabled.

To load and play, double-click on the Unhallowed.html file, or drag it into a browser window.

Important

This game doesn't work like most other Spectrum text adventures. Instead it takes its lead from the classic LucasArts graphical adventures in order to try to create an adventure that can be enjoyed by people who are usually turned off by complicated parsers and instant deaths.

Even if you think you know how to play adventures, please take a quick flick through this document before you start playing the game.

Gameplay

There is *no way to die* in this adventure. You can experiment as much as you like, you'll never see a GAME OVER until you finish the game.

Similarly, there is no way to miss an important object or end up in an unfinishable state – the game is specifically designed to be foolproof.

This doesn't mean that it's easy, however – although there's a lot more text than the average adventure, the puzzles are still puzzles and you have to solve them to progress.

Parser

This game uses a compact vocabulary to help prevent frustration. Rather than having to know or guess the words you need from a large dictionary, the verb set is deliberately kept small so that you always know what you need to type in order to achieve your objective.

You can use *HELP* to get a synopsis of the commands that the game understands, but here's a more detailed overview.

Actions

Every action in the game can be achieved using the following verbs:

GET
EXAMINE
USE
GIVE
OPEN
CLOSE
PUSH
PULL

You'll note that these match the classic LucasArts icons. That's on purpose – if it ain't broke, don't fix it, and if they're good enough for Day of the Tentacle then they're good enough for me.

On top of that, you can *WAIT* to let some time pass – that will occasionally come in useful but is not essential to completing the game. You'll see.

Movement

The game doesn't use classic compass directions to move around. Instead, available destinations are highlighted in green in the room descriptions. To get there, type *GO* or *WALK TO* and the name of the room.

e.g. *GO KITCHEN*, *GO GARDEN*, etc.

You can only *GO* to places which are highlighted in green in the current room description, however; you can't *GO* directly to the bathroom from the kitchen, for instance.

If you want to get around as quickly as possible, each room can be referenced by just the first two letters of its name, and *GO* can be shortened to just 'G'.

e.g. *G KI* (*GO KITCHEN*), *G GA* (*GO GARDEN*)

Utilising objects

USE is the only verb you need to manipulate or combine items – you don't need to search for the specific verb to make use of your item, the parser insinuates its use based on the context.

For example, to use your phone you could just:

USE PHONE

To combine objects you *USE object1 WITH object2*. '*WITH*' is the *only* preposition that you'll need for this – you don't need to worry about *ON*, *IN*, *UNDER*, etc.

For example, to use a key with a door you'd say:

USE KEY WITH DOOR

Or to put a sandwich in the fridge you'd say:

USE SANDWICH WITH FRIDGE

Standard Commands

The standard PAW commands are available as normal:

SAVE

LOAD

RAMSAVE

RAMLOAD

QUIT

Remember – you don't need to save unless you want to stop and continue later, as there's no way to die or mess up the game so that you can't solve it.

Text Colouring

Some of the text in descriptions and messages is colour-coded to highlight specific things:

GREEN – exits

YELLOW – items of interest that you can't pick up

CYAN – items of interest that you can pick up

Solving the game

In general you should EXAMINE as many things as possible, even if they're not things that you can pick up (anything listed in yellow or cyan may be of interest).

Pick up anything you think might be useful. There are *no inventory restrictions* – if you see it, you can take it. Equally you *can't drop objects* (although they may occasionally be taken from you), so don't worry about leaving something important behind.

USE objects with other objects, read the text carefully, and don't presume that because something doesn't work right now that it will stay that way for the rest of the game.

About the game

Two years ago I decided it would be fun to write a Halloween mini-adventure with PAW – something short but text-heavy, with a small play area. I also thought it would be fun to experiment with ditching a lot of the old text adventure tropes – compass directions, limited inventories, instant deaths and, most importantly, the large ‘guess which word I want you to use’ dictionary.

Taking my cues from the old LucasArts graphic adventures I aimed to produce a game which used just a handful of verbs to solve every problem in the game. It would highlight items of interest so that you always knew what you could interact with. It wouldn’t kill you – *at all*. But it would still be an adventure, requiring the same puzzle-solving skills you always needed back in the olden days.

As usual, I overstretched. The ‘mini’ project grew and expanded beyond what I’d originally intended, meaning I ran out of time for the Halloween release and ultimately shelved the entire thing. Instead I went on to learn Z80 and write a Game Jam-winning strategy game in Spectrum BASIC (seriously) and the adventure was lost to the dusty corners of my hard drive.

Fast forward 18 months and 8-bit text adventures suddenly seemed to be having something of a mini renaissance. The [2018 Crash Annual](#) had a brilliant text adventure section; Stefan Vogt released [Hibernated 1](#) to great acclaim and worked with Gilsoft’s Tim Gilberts to recover the ‘lost’ adventure system, [DAAD](#); Mark Hardisty released [The Big Sleaze 2.5](#) and published four issues of [The Classic Adventurer](#); Chris Ainsley created the amazing [Adventuron](#), a PAW-like adventure creation system for modern web browsers; and even John Wilson and [Zenobi Software](#) started publishing games again. Adventures were *back*!

My thoughts returned to my abandoned project and I started tinkering with it again, but not seriously. Then in September I suddenly decided to try and finish it off... before Halloween. On top of that, I was approached by Chris Ainsley who thought the game might be a cool fit for porting to Adventuron. I thought so too, so that also went on the to-do list! Two months seemed both a tight deadline and probably achievable; as it turns out I was right on both counts – I got there, but *man* was it hard work!

What you see here is practically a complete rewrite of what I had at the start of September apart from the opening two or three locations. It’s both a text adventure and a short story, more akin to modern interactive fiction than the classic 8-bit adventures of the ‘80s, but (hopefully) not losing that nostalgic ‘feel’. It’s *very* text-heavy, so I hope you’re prepared for a lot of reading, but I really wanted to create something that was more like a story, with far greater detail than you would usually see in an 8-bit adventure. I hope it’s also well-written; it’s hard for me to tell given that I’ve basically lived it for the last two months.

Similarly I hope the changes I’ve made to the classic text adventure gameplay have proven successful and that I can attract some players who usually wouldn’t go for a text-based game. And also that I haven’t driven away too many classic adventure fans!

But most of all I hope you enjoy the game. Do let me know via Twitter if you liked it, or even if you didn’t (but please be gentle).

Hugs!

[@Blerkotron](#), 31/10/18

Acknowledgements

Along the way I've had help and advice from a number of excellent folk who I'd like to thank personally for their time and patience:

- [Tim Gilberts](#), Graeme Yeandle and Phil Wade of Gilsoft for creating PAW. Without this amazing adventure creation system this game would not exist, and it's a credit to their artistry and vision that it still holds up so well today.
- Paul van der Laan for creating the brilliant '[Clairsys](#)' font which the game uses. It's beautiful and incredibly readable.
- [Chris Ainsley](#) for encouraging me to finish and port the game to Adventuron. This was actually a big help in getting to the finish line on time, and also gives me the opportunity to get the game in front of people who would never touch a Spectrum or emulator. Do give Adventuron a go if you're not the kind of lunatic who tries to create things in an 8-bit environment; it's a superb system and should be very familiar to anyone who played with The Quill or PAW during the '80s.
- [Pixel Rated](#), [SuperRetroid](#) and [Quantum Sheep](#) for offering to playtest the game prior to release. I was really nervous about showing it to anyone but these gents were super-nice and very constructive.
- And thanks also to everyone on Twitter who offered ideas and encouragement during the development process, and have helped spread the word about the game during its development. So many of you said nice, positive things, it's always a huge encouragement to me when I'm working on stuff. You're the best!

Tools and sources

This game came to life thanks to the efforts of some wonderful people who provided the tools and utilities that helped me along.

Development tools:

[Professional Adventure Writer](#) A17C by Gilsoft International

[Adventuron](#) by Chris Ainsley

[SpecEmu 3.1](#) ZX Spectrum emulator by Mark Woodmass

[UnPAWS](#) by Carlos Sanchez, Jose Luis Cebrian, Alexander Katz and German Gonzalez

[Paint.Net](#) by Rick Brewster

[ZX-Paintbrush](#) by Claus Jahn

[BMP2SCR](#) by Leszek Daniel Chmielewski

[ZX7](#) by Einar Saukas

[Bitbucket](#) by Atlassian

Fonts:

[Clairsys ZX Spectrum font](#) by Paul van der Laan (in-game)

[Cold Night for Alligators](#) by junkohanhero (loading screen)

Revision history

1.0 (31/10/18)

- Initial release. Enjoy!