

CellarDoor 1.1.2 Documentation



Go directly to the [SourceForge.net Project Page](#)

Thanks for trying CellarDoor, an Interactive Fiction interpreter for PalmOS devices. CellarDoor supports Z-Machine (Z-Code) and Glulx game files, including those wrapped in the Blorb file format (.zblorb and .gblorb files).

CellarDoor's Z-Code support is based on the Frotz v2.43 interpreter. Frotz is an Interactive Fiction interpreter which permits the play of Z-Machine (Infocom, Inform, Z-Code) games such as Zork and Hitchhiker's Guide to the Galaxy. CellarDoor supports V1-V8 games (including V6 with MG1/Blorb graphics).

CellarDoor offers Glulx support via the Git interpreter (1.2.2) and uses a (rather) modified version of the CheapGlk library (0.9.0). It supports JPEG and PNG graphics, if they are found in Blorb files.

It requires an ARM-based PalmOS device with hi-res screen, OS4 or later, an expansion card and the large dynamic memory area available from OS4 onwards. If your handheld does not meet these requirements, you may want to try Frobnitz, PilotFrotz or Kronos.

CellarDoor is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. Elements of CellarDoor (CheapGlk, Git, Frotz, PEAL) may have their own, less restrictive, licensing terms. Please read the documentation for these packages for more information.

Table of Contents

- [Principle Features](#)
- [Instructions](#)
 - [INSTALLATION](#)
 - [STORIES, FOLDERS & CATEGORIES](#)
 - [GRAPHICS AND AUXILIARY FILES](#)
 - [LANDSCAPE SUPPORT](#)
 - [GLULX](#)
 - [FINDING GAME FILES](#)
 - [USING](#)
 - [FIRST THINGS FIRST](#)
 - [THE STORY LIST](#)
 - [The Details form](#)
 - [Main Preferences](#)
 - [THE INTERPRETER WINDOW](#)
 - [Interpreter Preferences](#)

- [The Verb List and the Word List](#)
 - [Saving and Restoring](#)
 - [Notes](#)
 - [Miscellaneous Menu Items](#)
- [TROUBLESHOOTING](#)
- [Frequently Asked Questions](#)
- [Known Issues](#)
- [Bug Reporting](#)
- [Acknowledgements](#)
- [Abridged Changelog / Revision History](#)
 - [CellarDoor 1.1.2 \(2 February 2009\)](#)
 - [CellarDoor 1.1.1 \(23 May 2008\)](#)
 - [CellarDoor 1.1.0 \(20 May 2008\)](#)
 - [CellarDoor 1.0.3 \(27 April 2008\)](#)
 - [CellarDoor 1.0.2 \(14 April 2008\)](#)
 - [CellarDoor 1.0.1 \(12 April 2008\)](#)
 - [CellarDoor 1.0 \(25 April 2007\)](#)
- [Appendices](#)
 - [A: Principle Changes from CliFrotz](#)
 - [Story Support](#)
 - [Miscellaneous](#)
 - [Technical](#)
 - [B: CliFrotz 1.6 Change History \(for the archives\)](#)
 - [C: Links](#)

Principle Features

CellarDoor can run games in the following formats:

- Z-Machine (Z-Code, Infocom, Inform) V1-V8, including V6 with graphics, and newer games produced with Inform 7
- Glulx (Inform)
- Both of the above in Blorb files, supporting scalable JPEG and PNG graphics

Usage features:

- Autosave (switching out of CellarDoor to another application pauses your game, permitting you to pick it up from where you stopped, without requiring a manual Save and Restore)
- Category support helps you manage your story files
- Pop-up verb/word lists, for quick access to most-used commands and actions. Two independent lists are available (useful for multilingual users).
- Recording and playback of user input
- Transcript support
- User-definable undo level (for games which support it)
- Frotz hotkey support (Z-Machine only)
- In-game notes
- Really, very, extremely stable

Appearance:

- Multiple fonts
- Landscape and widescreen support for capable devices
- 8- or 16-bit color graphics, with 2 interpolation quality levels (4-bit greyscale and 8-bit color)

- available on OS4 Sony Clie devices)
- Custom fore- and background colors make CellarDoor easier on the eyes

CellarDoor was based on the source code of CliFrotz 1.6. For CliFrotz users, the PRINCIPLE CHANGES appendix, below, outlines the most significant differences between CliFrotz 1.6 and CellarDoor.

Instructions

Note that some sections of this document currently presume a passing familiarity with CliFrotz, and describe several features relative to their function or (non)existence in that software. As CellarDoor continues to evolve, so will the documentation. Until then, new users who have no familiarity with CliFrotz, please bear with the presumption -- it shouldn't hinder getting CellarDoor up and running quickly.

Throughout the documentation, the words "story" and "game" are used interchangeably, depending on the author's attitude toward interactive fiction at the moment of writing.

INSTALLATION

Hotsync the CellarDoorInstaller_112.prc file with your PalmOS handheld. This will install CellarDoor and any required libraries on your Palm and launch the application. Make sure the expansion card is inserted before you install.

The first launch of CellarDoor will cause a Preference file to be created (it's called CellarDoorDB.pdb). Choose Preferences from the menu on the opening screen to choose your memory card (in case your Palm supports multiple) and to set various display options.

See below, in the sections "STORIES, FOLDERS & CATEGORIES" and "GRAPHICS AND AUXILIARY FILES" about installing your story files and any associated data.

NOTE for Tungsten T3 users only: Install the AppSlipRotate.prc and StatusBarLib.prc files (they **MUST** be installed together) from the PalmOne web site if they haven't already been installed. This will enable widescreen support. Or you can get them [here](#).

STORIES, FOLDERS & CATEGORIES

Before you can actually play any games with CellarDoor, you'll need to a) have some and b) install them. Assuming that you already have some games, you need to get them, and all supporting files (.mg1 graphics, for a few Infocom titles) on your memory card. If you don't, see the section [FINDING GAME FILES](#), below.

*All supporting files (stories, saves, notes) **must** be located on your memory card, preferably in the /PALM/Programs/CellarDoor folder (generally in subdirectories, as outlined below). The easiest way to manage this is to use a memory card reader with your computer or a program like Sync Buddy (OSX: <http://perso.orange.fr/fpillet/syncbuddy.html>). Palm's Hotsync utility won't properly sync the types of files used by CellarDoor.*

The first time CellarDoor runs (and on subsequent launches, if the folders are missing), it creates several folders:

- /PALM/Programs/CellarDoor

- `/PALM/Programs/CellarDoor/Saves`
- `/PALM/Programs/CellarDoor/Misc`
- `/PALM/Programs/CellarDoor/Notes`
- `/PALM/Programs/CellarDoor/Stories`

Your **story files** can live either in `/PALM/Programs/CellarDoor` or in `/PALM/Programs/CellarDoor/Stories` or in a subfolder which you create yourself inside of the *Stories* folder (e.g. `/PALM/Programs/CellarDoor/Stories/Glk` or `/PALM/Programs/CellarDoor/Stories/Infocom`). Each subfolder of the *Stories* folder will appear as a category in the categories popup in the opening screen. Three additional categories, "All", "Unfiled" and "Saves" will show up, as well. *You shouldn't use these special names for your folders.*

Story files should be in their original format (for Z-Code files, this is usually a `.z#` file (where # is a number between 1 and 8) or a `.zblorb/.zlb` file), for Glulx, this is usually either a `.ulx` or a `.gblorb/.glb`), *and not in a Palm database format*. Files are not parsed based on extension, but are 'sniffed' to determine their validity and format, so any extension is acceptable, as long as the file is the correct type.

To create category subfolders and/or move your story files around, you'll naturally require a file manager or memory card reader. FileZ (<http://www.freewarepalm.com/utilities/filez.shtml>) is free and can help you with this task. [Resco Explorer](#) is not free, but is somewhat more pleasant to use. For convenience, you can move files between existing categories (folders) from within the Details screen inside of CellarDoor (but you can't create new ones).

If you don't want to use categories, instead using the "classic", CliFrotz mode, simply don't put any folders inside of `/PALM/Programs/CellarDoor/Stories`, in which case you'll have 2 tabs - Stories and Saves, just like in CliFrotz.

The old (1.0.0) CellarDoor base file location of `/CellarDoor` is still supported, if present. Fresh installations of CellarDoor 1.0.1 and above use the new location.

****NOTE**:** If you are updating from CliFrotz, you can rename your `/Frotz` folder, at the root of your memory card, to `/CellarDoor`, or move it to `/PALM/Programs/CellarDoor`.

Otherwise, you'll want to copy your games from `/Frotz` to `/PALM/Programs/CellarDoor` after you've run the program once.

GRAPHICS AND AUXILIARY FILES

- **Classic Z-Code:** If you are using a classic Z-Code story (e.g. `Zork0.z6`), for which you have a graphics file (`Zork0.mgl`), simply place the graphics file in the same folder as the story file. The file names should match the equivalent story files, i.e. the graphics file for `Shogun.z6` should be `Shogun.mgl`. These files are available from: <http://ifarchive.org/if-archive/infocom/media/mcga/>. You might also want to look into getting or making `.zblorb` versions of these games, to take advantage of the new scalable graphics features in CellarDoor, as detailed here: <http://babel.ifarchive.org/infocom.html>.
- **Blorbs:** Newer story files produced with Inform (Glulx or Z-Code) can include graphics, wrapped in the Blorb file format. Graphics for these games are "included" with the game, so to speak, so there's nothing for the user to do in this case. JPEG and PNG support is provided via the file `CellarDoorLib.prc` (included in the CellarDoor installer).

Note that PNG alpha channels are not currently supported.

Blorb graphic support can be turned on and off in the 2nd Preferences window.

- Other files: Some games contain references to additional files. "Dracula2: The Arrival" for instance, uses a little file called HORRDAT for some reason I can't ascertain and completely freaks out if it can't find it. If the auxiliary file is located in the same folder as the story file, you'll be fine.

LANDSCAPE SUPPORT

At the present time the Tungsten T3, T5, T1X, UX50/40 and Tapwave Zodiac support full landscape mode.

You will not be able to change the orientation while playing a game, because the interpreter engines get confused if the screen size changes in the middle of play. When accessing the Preferences from the Story List, there is an option to allow portrait mode (this is option is disabled when playing a game). Changing this option requires a restart of CellarDoor. On the Tungsten T3, T5 and T1X, the screen is forced to landscape widemode even when the slider is shut, or the DIA open.

GLULX

CellarDoor supports Glulx stories in .ulx or Blorb file formats, thanks to Git (<http://diden.net/if/git/>) and CheapGlk (<http://www.eblong.com/zarf/glk/index.html>). I've had to make significant modifications to both in order to support Palm OS and the CellarDoor application model, so don't assume that anything will work as expected. ;)

In particular, CellarDoor supports multiple windows, including graphics, while CheapGlk only supports a blind, single-window terminal. Because of the limitations of the Palm screen (that is, because it's so small), I've deviated a bit from the Glk specification to try to make as many stories look as good as possible. This is principally achieved by limiting graphics and their windows to 1/2 of the requested dimensions. This may have some unexpected consequences, but testing indicates that this is a lot better than the default values on little screens. Most games look and play great.

Sound is not supported. Graphics in *graphic windows* are supported. Graphics in *text buffer windows* are partially supported (the graphic will be drawn, but the positioning will likely not be correct).

FINDING GAME FILES

Infocom's games are still copyrighted, so you'll need to buy them or locate the ones which have been released into the public domain. But there's more to Interactive Fiction than Infocom, and there are hundreds of great, free Inform/Infocom (Z-Code) games available from:

<http://mirror.ifarchive.org/if-archive/games/zcode/>

You can find Glulx games at:

<http://mirror.ifarchive.org/if-archive/games/glulx/>

There's a large and active community of people making new and involving Interactive Fiction. See the list of "LINKS" at the end of this document for some other starting points for finding new story files.

USING

FIRST THINGS FIRST

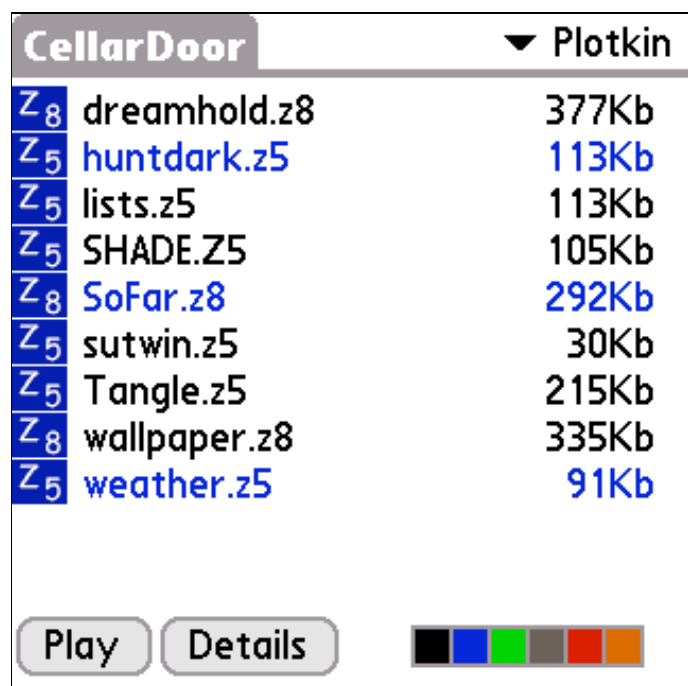
When running CellarDoor for the first time, check the "Check Heap RAM" menu to see if you have enough RAM available. If you have less than 800KBytes free then you will likely have problems running many larger Z-Code games, and you can probably forget about Glulx entirely. This free RAM is not the same as that shown in the Application launcher.

If the available RAM is low, pictures may not be drawn. For Z-Code games with associated .mg1 picture files, if you have lots of free heap RAM available, the picture file will automatically be cached in memory resulting in a speedup; on OS5 ARM devices this speedup is substantial. Blorb graphics are not currently cached.

For more information about memory-related issues, see the FAQ, below.

THE STORY LIST

The **Story List** is the central place in CellarDoor for finding and working with your story files. It looks like this:



If you created subfolders in your *Stories* folder (see above in [STORIES, FOLDERS & CATEGORIES](#) for more information), they are available from the drop-down menu in the top-right corner of the screen as categories. Otherwise, you'll have 2 categories: Stories and Saves.

On the bottom of the screen, you can see the 'Play' and 'Details' buttons. 'Play' takes you to the [interpreter window](#), while 'Details' opens the [details form](#), where you can rename, delete, move and tag the selected story file. The color widget on the bottom right allows you to filter your games by tag. Here's the quick rundown:

- Black: any game which is neither tagged nor has a saved game file.
- Blue: any game which has no tag, but does have a saved game file.
- Green: any game which is tagged 'In Progress'.
- Grey: any game which is tagged 'Finished'.
- Red: any game which is tagged 'Flagged'.
- Orange: any game which is tagged 'Other'.

Clicking in the menu bar area of the screen, or touching the menu button on your device (if it has one), will open the menus, where you can get information about your Heap RAM, open the [Main Preferences](#) screen, or check which version of CellarDoor you're running.

The Details form

The Details form looks like this:

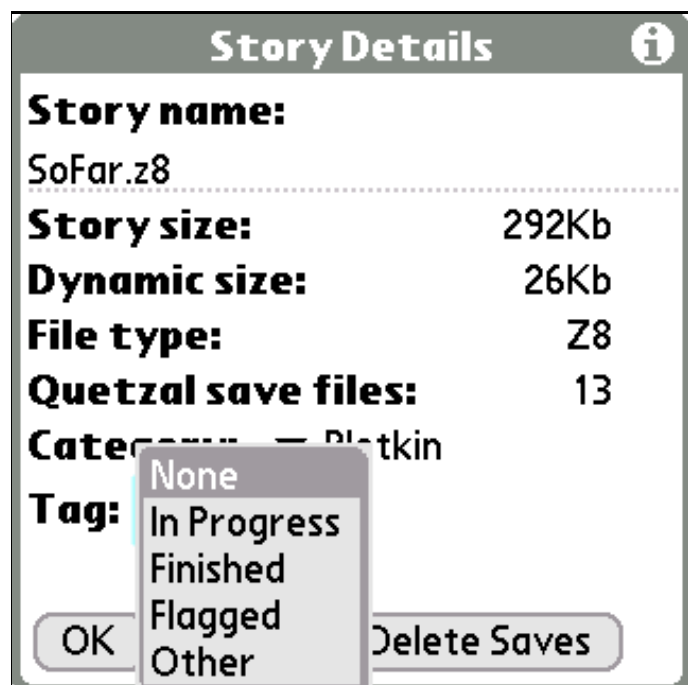


The screenshot shows a window titled "Story Details" with an information icon in the top right. The form contains the following fields and values:

Story name:	SoFar.z8
Story size:	292Kb
Dynamic size:	26Kb
File type:	Z8
Quetzal save files:	13
Category:	▼ Plotkin
Tag:	▼ None

At the bottom, there are three buttons: "OK" (highlighted with a blue border), "Delete", and "Delete Saves".

Most of the items here are self-explanatory, although you can click in the Information widget in the top-right corner for a description of the items displayed here. If you change the story's category, it will be moved to the selected category folder (this might take a few seconds). You can choose a new tag for your story by clicking on the Tag drop-down menu:



This screenshot shows the same "Story Details" window, but with the "Tag" dropdown menu open. The menu lists the following options:

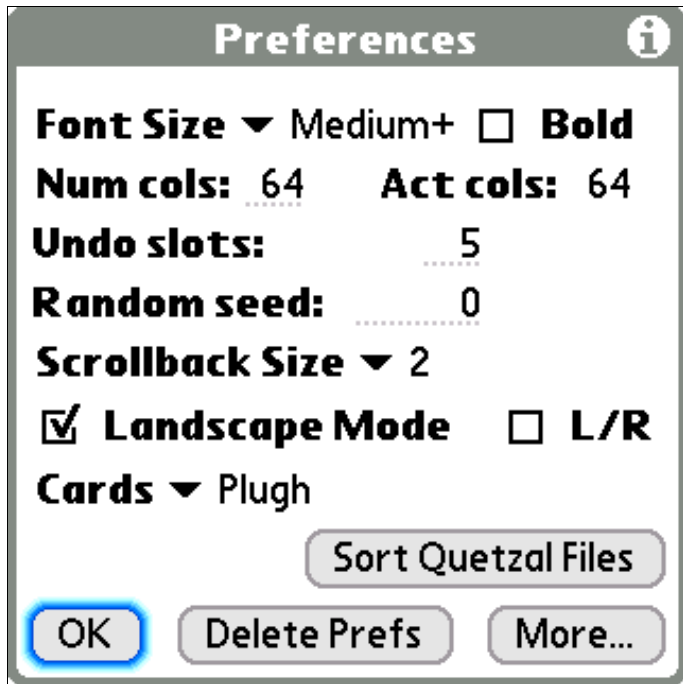
- None
- In Progress
- Finished
- Flagged
- Other

The "OK" button is now partially obscured by the dropdown menu.

Main Preferences

There are actually **two** Preference screens. The main screen (which is 2 forms) can be accessed from the Story List, and permits changing all configurable options in CellarDoor. The second screen (a single form) can be accessed during game play, but has only a very limited selection of options. This section deals specifically with the first set of Preferences screens, but is applicable to the second (since it's just a

subset of the full preferences). Anyway, the first form looks like this:



As with the Details form, the Information icon yields an in-application description of the items here. But here's a quick summary:

- Font Size: you can choose between 5 different font sizes, from Small (really small) to two different kinds of Medium (I like Medium+, myself), to a couple of different 'largish' fonts.
- Bold: check this to display normal text as bold and vice versa.
- Num cols/Act cols: set 'Num cols' to the screen width (calculated as number of '0' characters which can fit from left to right -- this is the read-only value after 'Act Cols'). In general, this is properly set, but you may need to adjust it if you change fonts, or if you're trying to fake out the interpreter.

****IMPORTANT**:** In general, the "NUM COLS" value in the Preferences should always be set to the same value as "ACT COLS". When you start CellarDoor for the first time, or when you rebuild your preferences, "Num cols" will automatically get set to a correct default value, based on your screen resolution.

There are some games which don't run, or don't run properly, if they don't have a certain number of columns available to them. If you are trying to get one of these games working, you may need to experiment with the "Num Cols" setting. If you are experiencing problems when using a non-widescreen device, try setting the "Num cols" to a value of 63 or more.

- Undo slots: the number of undo moves to allocate. Not all games support multiple undo.
- Random seed: if you know what this is, enjoy yourself. If you don't, just leave it at 0.
- Scrollback size: the number of vertical 'screens' to save for the scrollback. Each screen saved eats a nice chunk of memory.
- Landscape mode: if your device supports landscape orientation, this option will let you enable/disable it. On by default for landscape-capable devices.
- L/R: in landscape mode, the left-/right-handedness of the landscape orientation.

Changing either Landscape preference necessitates a restart of CellarDoor

- Cards: if your device supports multiple memory cards, you can choose the one to use for CellarDoor here.
- Sort Quetzal Files: push this to organize your save game files into folders based on their source game. Should only be necessary for users upgrading from CliFrotz, or who have copied pre-existing Quetzal save files to their memory card. See [Saving and Restoring](#), below, for a little

more discussion of this.

- Delete Prefs: deletes the preferences, including any custom word or verb lists you may have created.

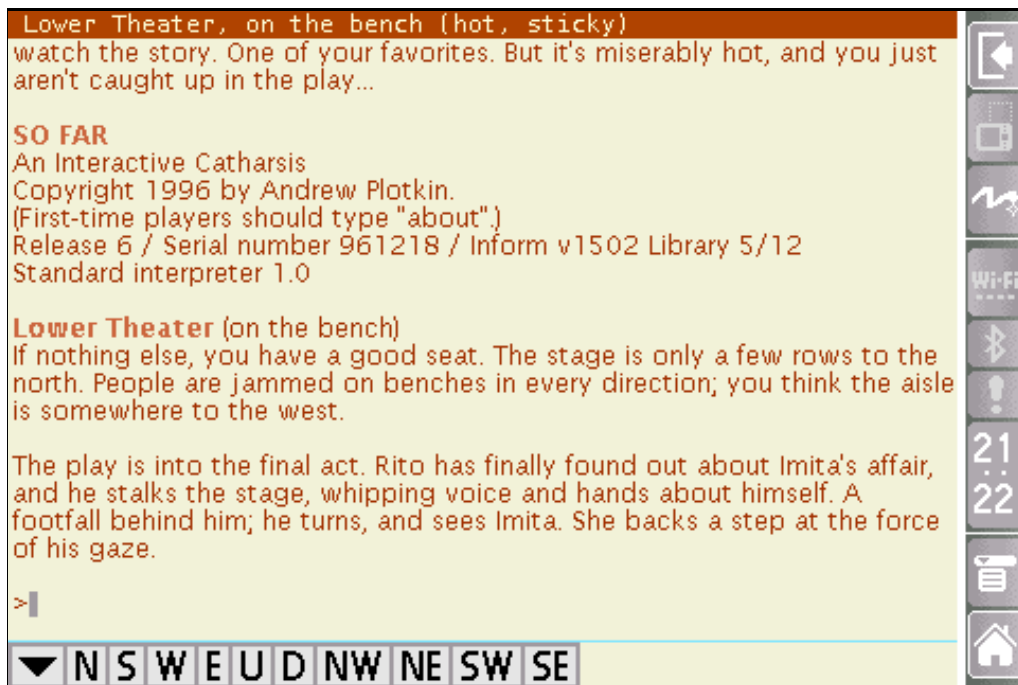
Pushing that 'More...' button gets you to the second screen:



- Graphics: master switch for Blorb graphics. Purists, turn this off.
- Resize algo.: switch between a 'Fast and Ugly' and a 'Slow and Pretty' algorithm for graphics scaling.
- Bit depth: if your device supports multiple bit depths, two options are available here -- 'Low' and 'High'. On OS5 devices, 'Low' means 8-bit and 'High' means 16-bit color. On OS4 devices, 'Low' means 4-bit greyscale and 'High' means 8-bit color. 'Low' is the default. Graphics in 16-bit color look beautiful, but use a lot more RAM. Since CellarDoor's entire display is a graphic window, using 16-bit color may also slow down overall performance and make certain games unplayable on older machines (e.g. Tungsten T). The 'High' setting is recommended only for newer machines with plenty of heap ram to spare.
- Hard button remap: enable this to use the two buttons to the left and right of the 5-way control (datebook and address) to move the cursor left and right.
- Auto <MORE>: prevent the story from pausing at screen breaks.
- Word List: switch between the two word/verb lists (described below in [Word and Verb Lists](#)).
- FG Clr/BG Clr: set the default fore- and background colors for the interpreter. Note the any given game might override your default settings.

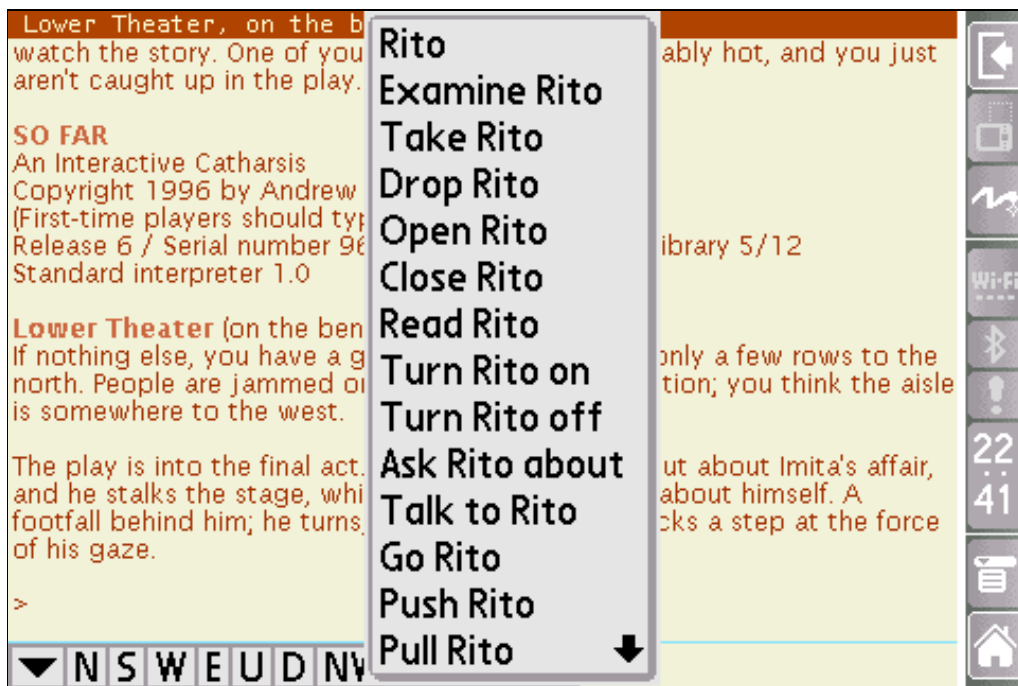
THE INTERPRETER WINDOW

This is where you play.



Most of what happens here is up to the game, but there are a few things to be aware of.

When playing a game, tapping on a word will bring up a menu permitting quick word entry, with contextual substitution (the Verb List). Tapping anywhere on the screen where there isn't a word brings up the command list menu (the Word List). This is what the Verb List looks like when clicking on the word "Rito":



You can scroll backward by dragging the screen. When you scroll back, you can tap on words and whitespace, just as when the screen is in its "normal" position. Selecting a word, writing a letter, etc. will cause the screen to jump back to the text entry line.

Some games support a "Mouse" such as Journey and Zork Zero. CellarDoor emulates mouse "clicks" by tapping on the screen, e.g. you can tap on a word in Journey.

The most important screens which you can reach from the menu in the interpreter are the [Interpreter Preferences](#), the [Verb List and Word List editors](#), the [Save and Restore forms](#) and the [Notes](#).

Interpreter Preferences



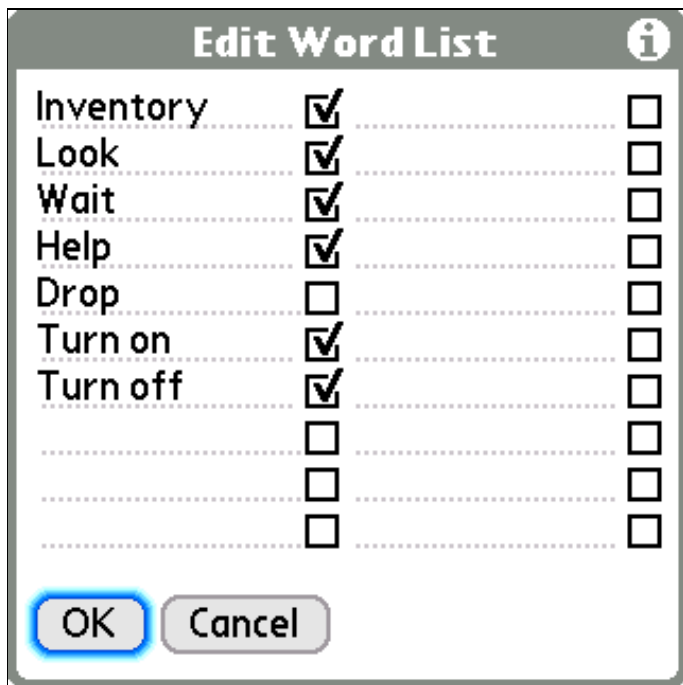
Note this this is a slimmed-down version of the second preferences screen, discussed [above](#). There's nothing new here; these are, however, the only items which can be changed during game play.

The Verb List and the Word List

The Verb List:



and the Word List:



Edit Word List ⓘ

Inventory	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Look	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wait	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Help	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Drop	<input type="checkbox"/>	<input type="checkbox"/>
Turn on	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Turn off	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>

OK Cancel

can be used to quickly insert commonly used text into the interpreter. They both function identically, except that the **Word List** doesn't support contextual substitution. The **Verb List** does: the '\$' character will be replaced by whatever word was tapped in the interpreter. The check mark after the entries determines whether a carriage return/enter will be added after the selected text is inserted in the interpreter.

You can add up to 20 verbs (with contextual substitution) and another 20 words (no substitution) to the pop-up menus which appear when you click on text (for the verbs) or non-text (for the words) in the Interpreter window. There are, in fact, 2 sets of verb and word lists, in principle for users who like to play in multiple languages, but you can use them for whatever you like. Switching between the 2 list sets can be accomplished from the 2nd Main Preferences screen (or from the Interpreter Preferences).

Saving and Restoring

Choosing the Save menu item yields the Save form:



Save ⓘ

D2.sav
Dome.sav
gate.sav
gate2.sav
Grassy.sav
Ice.sav
Ice2.sav
Ice3.sav

New Name:

Save Delete Cancel

Here, you can save your game under a new name, or select a previously saved game and save over it. The Restore form looks more or less the same, just without the 'Delete' button.

New saved games are written to subfolders of the */PALM/Programs/CellarDoor/Saves* folder. The subfolders are named after the story file being saved (e.g. SLOUCH.z5's saved games are written to a folder called */PALM/Programs/CellarDoor/Saves/SLOUCH/*).

If you are upgrading to CellarDoor from CliFrotz, you may want to go to the (Main) Preferences screen, where you'll find a button labeled "Sort Quetzal Files". Tapping this button will cause saved games to be sorted into folders. "Orphaned" saved games (for which there are no story files) are moved to a folder called "Unsorted".

Regardless of whether you sort your saved games or not, CellarDoor will find any saved games in the */PALM/Programs/CellarDoor/Saves* directory or in a subfolder thereof.

****NOTE**:** Due to a limitation of the interpreter, you can't automatically restore saved games by choosing them from the Saves category in the Story List. The associated story file will be opened, and you can manually load the saved game with the "Restore" menu command.

Notes

Notes are stored as simple text files named on a per-game basis (SLOUCH.z5's notes are called SLOUCH.txt) and can be found in the */PALM/Programs/CellarDoor/Notes* folder. They are available from the interpreter during game play, and can be up to 64K in size.

Miscellaneous Menu Items

The "Hotkey" menu enables access to the Frotz Hotkey functions such as undo, record, playback and debugging. When using Record and Playback the file is put into the */PALM/Programs/CellarDoor/Misc* directory, the name is automatically generated from the Story filename. For example, a playback file for story Zork1.z3 is named Zork1.rec

Transcripts, i.e. the "script/unscript" command also go into the */PALM/Programs/CellarDoor/Misc* directory. These are plain text files, and have the extension .scr.

TROUBLESHOOTING

If CellarDoor crashes before a game starts, try freeing up some internal memory (do a soft reset, for instance, or run a cache flushing utility). If it continues to crash and you can't get the game to start, try deleting the CellarDoorAS and CellarDoorDB databases using a File Manager utility such as FileZ (hint: they are generally right next to the program file). Some games also understand the "verify" (or "\$verify") command. If your game starts, but crashes after a few turns, try running "verify"/"\$verify" to see if the file is corrupt.

If you get stuck in an endlessly screwing-up Autosave loop (try running Winter Wonderland with "Num Cols" set to 45), where you can't regain menu control over CellarDoor, you can either use a file manager to delete the CellarDoorAS database, or you can start CellarDoor with the 'down' button held (or press it immediately as you launch). If an Autosave session is about to be restored, you'll get a dialog box prompting you whether you'd rather just delete the Autosave file and return to the Story List.

Frequently Asked Questions

- My game looks kind of funny. If I had to describe it, I'd say it's leaning to the right -- text is centered weirdly and gets cut off on the right margin sometimes.

You should double check that "Num cols" matches "Act cols" in the Preferences. This is only a preference because some games won't work at 45 column width (e.g. Winter Wonderland), so you have to trick them (hint: Winter Wonderland *does* work at 47 column width, and looks fine at 64 on a wide screen). In general, you always want "Num Cols" and "Act Cols" to match.

- Can I add my own fonts to CellarDoor?

Golly, no. Not yet, in any case.

- My kind of big built-with-Inform-7 game (probably .z8 or Glulx) is kind of SLLLLLLOOOOOOOOOOOWWWW. What's up with that, dude?

Dude, it's a Palm. 312 MHz v. 2.6 GHz (to compare my Palm to my laptop) is a HUGE difference, if we're just comparing processor speed, and there are many, many more hardware and software considerations. I think it's all running pretty quickly, all things considered. Theoretically, some of the core interpreter code (which is where the app spends most of its time) could be moved out of the 68K application and into ARM resources for an unknown potential speedup. But that's not much fun, and I haven't done the necessary research to see if it's really feasible.

- I want to play MEGAKLO.z8 or MAGDJAGD.blb and CellarDoor seems to load it ok, but then it gives me a memory error. Sometimes it seems to work ok, but then it just crashes!

Most likely, you have reached the frontier of your Palm's memory allocation abilities. There are a couple of things which you can try, although no guarantees. On newer Palms, check out Dmitri Grinberg's UDMH (<http://palmpowerups.com/modules.php?name=Content&pa=showpage&pid=2>). On the Tungsten T or T2, look at the Fargo Heap Resizer (<http://fanoush.wz.cz/palm/fhr.html>).

- Does CellarDoor support TADS, Magnetic Scrolls, or any other kind of game not listed above in the features list?

No. I would have mentioned it otherwise. Everyone likes bullet points.

- Why does the world need another freakin' Open Source PalmOS interactive fiction interpreter?

It probably doesn't. But I wanted to fix some bugs in CliFrotz and got a little carried away. As far as I know, none of the other interpreters are in active development, and it's arguably more important to have 1 active project than 3 inactive ones. CellarDoor is approaching what I want, personally, in a portable interactive fiction interpreter, and none of the other options were working for me. I presume I'm not the only one in that situation.

- My game's frontispiece graphic isn't showing up!

If the frontispiece graphic doesn't fit in the Palm dbCache, it won't display. For instance, on a Palm Tungsten E2, none of Emily Short's newer frontispieces (several megabytes decompressed) appear. This appears to be a limitation of the graphic libraries I am using. You can try something like dbCacheTool or Reset Doctor to flush the dbCache (or a soft reset), or you can remind yourself that you're playing text games, and not worry about it.

See the next question for even more information about this.

- What is the CellarDoorLib.prc and why do I need it?

CellarDoorLib.prc is a kind of 'best-of' image reading library, comprising certain elements lifted from [pnoJpegLib](#) and [palmPNG](#), modified and relinked into a new .prc file. Earlier versions of CellarDoor used these two libraries to manage JPEG and PNG graphics. Starting at version 1.1.0, CellarDoorLib.prc is used exclusively for graphics.

Why the change? Due to a problem/limitation of UDMH, it was necessary to adjust the way images are allocated in CellarDoor. When you allocate images on Palm, you generally do it in the dynamic heap. Since UDMH "extends" the dynamic heap into the storage heap behind the scenes, there's always a chance that you will allocate images in the storage heap when UDMH is enabled. In this case, though, calls to `_free_` the image don't work, because the Bitmap Manager assumes that you're reading an image out of a database "on disk". So memory begins to fill up without being released. It looks and feels like a bad memory leak, causes the dbCache to fill up with unfreed blocks, and after a little while, or after loading a few large images (such as blorb frontispiece graphics), you have to soft reset in order to have normal use of your Palm again.

Because CellarDoor, starting at version 1.1.0, allocates a little more screen memory than previously (and permits the user to allocate even more for scrollback), I wanted to solve this problem. It involved making a fundamental set of changes to the 2 graphics libraries -- they now create Bitmaps 'by hand' using some code I found [here](#), allocated using normal Memory Manager functions, rather than using the OS's Bitmap API. This ensures that they can be freed.

Since it made me nervous to distribute modified versions of perfectly good libraries, I decided to recompile the elements I'm using under a new name and type code. The result is a somewhat leaner library (142K instead of 190K for the two libs together) without any compatibility issues for people who use those libraries for other software. CellarDoorLib.prc can exist perfectly fine next to the originals and will cause no conflicts or problems. Using this library (and code from the library which is also in the main application), CellarDoor is now able to release all of the memory that it uses for graphics (and consider, the playing screen is a graphic!), so this is a massive improvement whether you are using games with pictures or not.

So, you have to use it if you intend to read images from blorb files. If you're not using pnoJpegLib or palmPNG for anything else, you can safely delete them.

- I have a high-resolution OS4 Sony Clie. What can I expect?

Good question. Given that the original app I was working with was called 'CliFrotz', I've been somewhat reluctant to leave Clie users out in the cold, despite the limitations of the OS4 Clie devices. OS4 Clie support was broken in versions 1.1.0 and 1.1.1, but fixed back up for 1.1.2. There are, however, some caveats.

First, CellarDoor needs more memory for display than before. As a result, OS4 Clie devices are, by default, put into 4-bit greyscale mode, rather than 8-bit color mode. You can override this by choosing the 'High' bit depth setting in the Preferences form, but I wouldn't recommend it, unless you are mostly playing .z3 and .z5 games.

Second, I've limited the ZCode stack size to 1K on OS4 Clie devices, rather than the ~32K available to OS5 devices. This means, practically, that you can't play games made with Inform 7 on an OS4 Clie. I doubt that Glulx games run at all, but I didn't make any changes to the Glulx code to accomodate OS4 machines.

If none of that sounds so bad, then please enjoy!

- Can I contribute to the continued development of CellarDoor?

Depends what you mean.

If you mean, give me money so that I will continue to work on it, I'd prefer that you donate it to a left-leaning, non-religious charity or NGO of your choice. I'll continue to work on CellarDoor for free, at least for the foreseeable future. It's just that much fun. If, given all that information, you really can't resist the urge to press a monetary token of appreciation directly into my virtual hot grubby little hand, I have a PayPal account registered under the email address below.

If you mean, work on the source code and add new features, that's even better! Drop me a note, tell me about what you want to work on, and we'll figure something out.

Known Issues

- Glulx graphics-in-text-windows are not handled according to spec. The graphics are drawn, but without any text flow. Sorry, I was too lazy.
- CellarDoor uses a non-standard text entry method, which doesn't work with the PalmOS's pop-up keyboard. Software which overrides the graffiti area, like FitalyStamp or SilkyBoard, should work fine, but any programs which use the same cues as Palm's keyboard to know when text is being entered won't. Units with a DIA are not affected, but older Tungstens (T, T2, E2) and Zires are. This just means: graffiti, graffiti, graffiti. Sorry. I'm not opposed to reexamining the text entry code at some point, but that's for a later revision of CellarDoor.
- My algorithm for making sure that the screen splits properly doesn't work right for everything. In particular, situations where a graphic window is drawn first, then split to a text window, causes smaller-than-desired text windows. (Lock & Key exhibits this, as does Stiffy Makane). In the case of Stiffy, just turn off the graphics if it bugs you. Lock & Key is currently more or less unplayable.

The reason for this, if you're interested, has to do with the forced size-reduction I'm applying to graphics windows. Glulx game designers have it really, really good, and can generally assume that they have at least 640x480, if not 800x600, if not 1024x768 pixels of beautiful, full-color screen real estate to play with. As such, many simply don't take into account what might happen if someone tries to squeeze their game into a PDA. I've taken some liberties with the Glk spec, and given preference to text windows: I always give graphics half as much as they ask for. I haven't yet found a game which looks better (on Palm) when I follow the spec. The problem arises when a window I care about is split off from a window I only cared 50% about. I could retroactively resize the graphics window at this point, wreaking havoc with the existing (mostly well-working) code and providing myself weeks of migrainey fun, or I could say "oh well...". Theoretically, Glulx games can ask the interpreter how much window space they really got and adjust accordingly. There's probably an algorithm out there to make everyone happy. But I didn't find it.

- Unicode "support" is limited to Unicode reflecting the Latin-1 character set. Nevertheless, CellarDoor properly displays the subset of Unicode that it understands.

Bug Reporting

CellarDoor is as bug-free and play-tested as possible. While it won't eat your lunch, leave the seat up or make international calls with your cellphone, there are probably still some things wrong with it. If you encounter any:

Please tell me what game you were playing (and where I can find it, if it's the first time you've mentioned it), what you were doing, if you can reproduce the problem or if it was a one-timer. For display bugs, please try the same game in a "standard" PC interpreter and compare the behavior in WinGlulx or Zoom or Spatterlight to what I've got. Don't forget to shrink the game window to 320x320 or 480x320 and

compare!

Acknowledgements

First of all, CellarDoor would not have been in any way possible without the amazing work of **Simon Quinn** (aka **Fangorn**) on CliFrotz. Without CliFrotz, CellarDoor would never have been started. After the release of CellarDoor 1.0, I heard from Simon, who gave me his blessing for the continued work on the app. So, thank you Simon for some great work. Simon based CliFrotz on **Rick Reynolds'** PalmFrotz project, so I raise a glass in tribute to Rick's hard work, as well.

Next, both **Andrew Plotkin** ([Gik](#), [Glulx](#)) and **Iain Merrick** ([Git](#)) provided extremely valuable insight and encouragement while I was struggling with the Autosave implementation (sorry for being such a pest, Iain!). The first implementation of Glulx in CellarDoor was actually done in Andrew's Glulxe and, after I had spent 2-3 extremely unpleasant weeks getting Autosave working, I determined that it (the interpreter) was simply too slow for a PDA. So I ripped out all the hard-fought-and-won code, replaced it with Git, and then spent another 2 months of intermittent unpleasantness getting it all working again. And now it works. And I'm thrilled about that, so thanks guys!

Selmi ([palmPNG](#)) was incredibly helpful while I was ironing out the PNG graphics support, and continues to be very supportive. And herzlichen vielen Dank to Stefan Stolz (to whom I've never had the presumable pleasure of speaking) for his [pnoJpegLib](#), a modified version of which I'm using for JPEG graphics + general image scaling.

Finally, a big thanks to my small team of enthusiastic beta testers, without whom this app would still be a pile of undiscovered bugs and lost opportunities. In order of appearance: **Luis Fernández**, **Mike Russo**, **Michael Ferrador**, **Rick Reynolds**, **Incanus**, **Domingo Stephan** and **Doug Roberts**. For version 1.1.2, I additionally thank **Sean Huxter**, **JoAnn Radway**, **Mischa Magyar** and **Jeff Sonas**

Berlin, 2 February 2009

Jeremy Bernstein jeremy@bootsquad.com

Abridged Changelog / Revision History

CellarDoor 1.1.2 (2 February 2009)

- Updated Glulx engine to Git 1.2.2, supporting @malloc and @mfree.
- Fixed several Glulx text positioning and drawing bugs.
- Fixed a minor bug in the Save form, whereby it was impossible to overwrite existing files.
- Resolved Tapwave Zodiac compatibility issues and fixed DIA open/close behavior.
- Resolved Sony Clie (OS4/5) compatibility issues and reduced memory requirements on OS4 devices.

CellarDoor 1.1.1 (23 May 2008)

- Fixed completely broken Notes from 1.1.0. Sorry about that.
- A couple of late-breaking bugs were slipstreamed into 1.1.0 in the first hour after it went live, dealing mostly with odd behavior during autosave/autoload of a few specific games. These fixes are incorporated in 1.1.1.
- Listed bug fix for 1.1.0 about landscape/portrait flipping is retracted. It's not fixed, and I don't know how to fix it. Yet.
- Updated docs.

CellarDoor 1.1.0 (20 May 2008)

- Story List rewrite: after a slightly longer initialization, switching between categories is now more or less instantaneous. The new code improves CellarDoor's performance when dealing with large game libraries, and allows the implementation of more extended list behaviors such as tagging (see below).
- Interpreter screen model rewrite (now with Super-Scrollback!): A new, improved screen model permitted the implementation of scrollback on a per-window basis, which doesn't snap back automatically. Meaning that you can scroll back N screens (as set in the Preferences from the Story List) and click on stuff.
- Tags: games may now be tagged (with a few default values), and filtered by tag in the Story List.
- Glulx Unicode compliance -- CellarDoor is now Glulx 3.1-compliant (with the exception of @malloc and @mfree).
- Glulx: mouse requests now qualify a window to be the focus window, if no other windows have pending requests.
- Fix for landscape/portrait flipping when clicking on gadgets in forms, or using a pop-up DA manager. Ensured that the orientation returns to the initial state when CellarDoor exits. NOTE: this was not really fixed (see above).
- Partial rewrite of pnoJpegLib and palmPngLib, implementing 'hand-built' Bitmaps to get around problems with unfreeable graphics memory when using CellarDoor with UDMH. The rewritten libraries have been slimmed, combined and relinked into CellarDoorLib.prc, and should cause no conflicts with pnoJpegLib or palmPngLib (neither of which are used by CellarDoor anymore).
- Z-Machine: fixed an obscure crasher associated with quitting a non-MG1-graphics game after having already quit an MG1-graphics game (so, Shogun, Zork0, Arthur and friends). Probably never found by anyone.
- Medium+ Font no longer displays size irregularities with capital letters featuring accents, umlauts, etc.
- Sundry fixes.

CellarDoor 1.0.3 (27 April 2008)

- Added default color to the Preferences. Note that choosing custom colors might cause problems with games where some, but not all, colors are specified.
- Adding color required that we change the screen-compression scheme used for Autosave. Now using SysZLib for Autosave/load, included in the installer.
- Increased word/verb list size to 20, and grew the forms to match. 20 feels like a good, final, never going to change again kind of number to me.
- Numerous fixes to the drawing engine, eliminating some unnecessary refresh overhead and getting some UI elements to clean up better after themselves
- Added 3 pixels to screen height and adjusted bottom bar color.
- Fixed crasher in 'Delete' in the Save Game form.
- Glulx: Now has 32 visible windows permitted (64 total). Fixes Jon Ingold's Dead Cities, and probably some other games I haven't tried.
- Glulx: glk_window_erase_rect() and glk_window_fill_rect() were using the wrong area of the screen. Fixed.
- Glulx: Improved sizing of graphic windows upon Autoload.
- Glulx: Random seeding now works properly.
- Glulx: Fix for broken Autoload of complex window setups in Glulx games (CoS, for instance).
- Glulx: Fixed mouse event bug, whereby only the first mouse event was captured under some circumstances, while all others were ignored.
- Glulx: Improved Autoload's handling of style hints.
- Fixed completely broken mouse events (broken in 1.0.2).
- Autoload failure is now more graceful.

- Word/Verb-popup no longer strips apostrophes.
- Rewrite of word/verb list code; somewhat more efficient now, doesn't require constant database access.
- Lots of drawing and performance fixes in the Notes window. No longer feels like an afterthought.
- Added standard Edit menus to Words, Verbs and Notes. Using the Command Toolbar has never been so satisfying.
- Renamed the ambiguous "Free Heap RAM" to "Check Heap RAM", which now reports in KB, rather than in bytes.
- Numerous minor fixes to Autosave/load.
- 'down' can now be used to get out of MORE.
- 'down', if there is no 'next command' in the history, erases the current line.
- Cosmetic changes (border around the more button and new button graphics, cursor color, etc.).

CellarDoor 1.0.2 (14 April 2008)

- Fix for nasty crashing bug with the command history: trying to access or insert a command into the history on a 2nd game launch in the same session (e.g. play a game, close it, play a new game) had an excellent possibility of corrupting memory, causing instability and resets.
- Infocom V6 stories, if the graphics can't be loaded, or (for blorbs), if graphics have been disabled, no longer exhibit screen corruption, and are now playable. "Zork Zero" (non-blob) is an exception, and hangs CellarDoor (and Frotz). The "Zork Zero" blob works fine with or without graphics, and in general, you should now be running all of the Infocom V6 games as blorbs, even if you have graphics disabled, so that they scale properly. There's now code for coaxing width and height info out of blob data without using the graphics plugins, so that the screen will format correctly, even if nothing is drawn.
- Fix for input behavior in some games when tapping on the text entry line, whereby the cursor would jump back to the end of the line when you are trying to place it somewhere else.
- Fixed eternally old cosmetic problem from back in Frobnitz days, where interface text and/or the 'MORE' prompt would get underlined, if, at the moment when it's made visible, the output text of the game is underlined.
- Improved compliance: ZCode color now handled like DOS/WinFrotz -- bold is brighter than normal style. This allowed me to remove several color hacks, and Varicella is still playable. CellarDoor now passes item 5 in etude.z5.
- Improved compliance: ZCode 'pre-loading of input line' now works better (there were some positioning issues, and the interpreter was redrawing text already drawn by the game). Item 12 in etude.z5.
- Improved compliance: After a 'clean' end to a game (quit, win, etc.), CellarDoor prompts for a single key press/input event before returning to the Story List. Item 14 in etude.z5.
- That means we're 100% for etude.z5, by the way.
- Improved compliance: Glulx games which do not specify a style or color for status windows (e.g. the header) don't automagically get reverse text. Reverse text looks better in many cases, but causes unwanted behavior when entering text in those windows.
- Selection and scroll position are now remembered when you return to the Story List from the interpreter.
- Graphics on/off and bit depth are now hidden from the user during game play. They may be set from the Story List.
- Improved program behavior if the memory card is removed from play. Your game will get safely autosaved, and CellarDoor should gracefully exit.

CellarDoor 1.0.1 (12 April 2008)

- Support for newer Glulx games (via Git 1.2, see below)
- Command history filters out consecutive duplicates.
- Fixed completely broken resolution scaling for blobbed Infocom graphical games.

- Fix for broken MORE prompts on the (>1)th start of Glk games
- Redesigned Preferences screens. System settings which cannot be changed during game play are now only available from the Story List. Landscape and L/R require a restart of the app to avoid graphical weirdness.
- The "Num cols" setting in the Preferences now defaults to the "Act cols" value, when new Preferences are created.
- "Sort Quetzal Files" no longer leaves a file reference open.
- Plugged small, but potentially nasty, memory leak in pop-up word lists.
- Interpreter sometimes wasn't redrawing completely after using (for example) the word list form. Fixed.
- Graphic libs won't be initialized if graphics have been disabled.
- Updated Git engine to 1.2 (from 1.1.3), adding support for @mzero and @mcopy opcodes
- Updated palmPNG to latest version (slightly modified to permit operation on screens set to 8-bit), changing from static to dynamic lib linkage. The CellarDoor app is weighing in at only about 296K now. Of course, you have to install the 100K shared library, too (included).
- /PALM/Programs/CellarDoor is now the preferred documents location. The old location is used, though, if it's there.
- Fix for paths specified with a '/' on the end causing issues in the simulator. I think that this was a non-issue on the actual hardware, but it's fixed now.
- Fixed very minor Git unicode bug in the Glulx interpreter.
- Minor fixes (memory leaks, etc.)
- Implemented 'down' check at startup for prompted Autosave removal.
- Added a timestamp to the About screen.

CellarDoor 1.0 (25 April 2007)

- Added 3 additional typefaces and changed font menu to read "Small, Medium, Medium+, Large, Larger". Now even people with eyes as bad as mine can play at all hours of the day. Medium and Medium+ have the same fixed-font width, but the proportional font in Medium+ is a little larger.
- Devices with 16-bit capable screens can now use 16-bit color density. The default is 8-bit, which was the previous setting. Using 16-bit looks much nicer with graphics, but uses somewhat more memory. If your device is already low on memory, or if you're not using graphics, stick with 8-bit color. There is a preference to control this.
- Enabled Glulx frontispiece
- Disabled menus in MORE prompt
- Hard buttons (phonebook and todo) emulate left/right, rather than up/down, which should be more useful.
- Bug fixes w/r/t the insertion point, command history and graphics resizing.
- Implemented bare-bones graphics-in-text-buffer-window, without text flow. Graphics are drawn left-aligned, essentially as a paragraph. Text restarts beneath.
- .rec file recording now works if the active window is not #0
- Fixed unwanted MORE prompt when playing back .rec files for Glulx games
- Minor fixes to opening screen w/r/t scrollbar & update frequency
- New icons
- Increased Palm OS stack size to 6K from 4K
- History increased to 40 entries
- Got hard key remapping working (again?)
- Fixed problems with Very Long Filenames© crashing the opening screen
- Implemented Autosave for Glulx (Yay!)
- Implemented multiple word lists, plus a control for them in the Preferences form
- Implemented save game folders plus "Sort" button for moving unfiled saved games into folders
- palmPNG code hacked to improve legacy Infocom .zblorb support (scaling, adaptive palettes, transparency)
- Fixed memory leaks in palmPNG code

- Enabled command recording/playback for Glulx
- Added Transcript on/off menu item (just sends Script and Unscript commands - nothing special).
- Autosave now deletes partial input line before doing its thing.
- .zblorb graphics are now supported. ZCode graphics are NOT auto-scaled like Glk graphics, but rely on the 'Reso' chunk of the Blorb for instructions. I didn't notice any major gameplay problems with this, but I haven't had time to play around all that much. Seems to be "doing the right thing". This mostly applies to old Infocom games which have been retroactively Blorbed.
- Support for the .zblorb frontispiece graphic also supported. I'm not handling this for Glk, since most Glk games seem to deal with this themselves.
- .zblorb support!
- Finally got it working again on OS/hardware that doesn't know about the DIA.
- Stability/memory hygiene improvements. No leaks currently reported in the simulator.
- Z-Code stack size increased (necessary for Inform 7-generated games)
- Memory reading and writing is now properly aligned: this may slow things down a bit, but is the way 68k code needs to be...
- Implemented Glulx graphics windows
- Added Graphics checkbox to Preferences - this is the master switch for .gblorb and .zblorb games. The preference only affects *new* windows. Old windows will continue to exist (although images won't get drawn) until they are closed.
- Implemented limited support for Glk style hints (colors & reverse)
- Increased possible number of windows to 16 (from 8) and added checks to set_window to prevent window indices < 0 from wreaking havoc with pointer arithmetic
- Changes to Quetzal save routines to support .zblorb games
- Additional improvements to the window appears/disappears/appears situation (e.g. Beyond), so that the cursor (hopefully) never gets trapped in the wrong window.
- Support for additional text windows improved.
- Additional text windows can now scroll independently of the main window. Nevertheless, mouse-driven scrolls (to read the back buffer) still affect the entire screen.
- Support for pop-up word lists has been extended to secondary/tertiary... text windows.
- Text is now displayed with a 2 pixel margin on the left hand side. This was a lot trickier than it sounds!
- "Delete Saves" button in Details panel to kill all saves for a particular game
- Category changing is supported from the Details panel (but not new Category creation)
- MORE button back in landscape mode
- Fix for insertion point issues
- Categories for games (managed as folders - maybe could build in built-in editor, but seems unnecessary)
- Note form + note folder + note DB per game
- l/r landscape option in Preferences to force handedness
- Fix for post-form Graffiti constraint size on 480x320 devices: now can Graffiti on the right hand side, too!
- Reading a corrupt Autosave will no longer crash CellarDoor
- Autosave is backed up after read
- Save dialog now filters on game, rather than showing everything
- Fixed endless node list crashes
- Graphics support improved
- Numerous signed/unsigned overflow bugs (presumably differences betw. gcc & CW compiler behaviors)
- Added PEAL layer for ARM
- Fixes for gcc compile

Appendices

A: Principle Changes from CliFrotz

Story Support

- Support for Glulx (.ulx), .zblorb and .gblorb files, including Autosave.
- Previous support for Z-Code has been updated to accomodate games produced with Inform 7 (with greater memory demands)
- Improved Z-Machine compliance
- Support for scalable Blorb graphics (JPEG and PNG)
- Categories

Miscellaneous

- Notes
- 2nd Word List (for multilingual users)
- Quetzal save files now written to per-game folders to help avoid naming collisions.
- More predictable Landscape mode support
- 16-bit color support on machines which support it
- Additional, larger font sizes

Technical

- Now compiling with gcc (prc-tools) instead of CodeWarrior
 - Using PEAL for ARMlet support
 - Using multilink for lazy code sectioning
 - Many, many, many bug fixes/memory leak plugs.
-

B: CliFrotz 1.6 Change History (for the archives)

v1.6 24/03/05

- * Added support for the TH55 landscape utility
- * Speedup for Zodiac screen display
- * Fixes to word popup on command line
- * Fixes to more prompt

v1.5 25/01/05

- * Fixed Tungsten T5 and W recognition
- * Zodiac joystick support and grafitti window event fix
- * Zodiac disable landscape option added
- * Clie double event on jog dial fixed
- * Reinstated Function Key menus
- * Popup Words menu is now free floating instead of using the button bar.
- * Added Carriage Return button to button bar
- * Saved games list now displays dates instead of sizes
- * Removed Hotsync registrations due to instability issues - PalmOS bug?
- * Fixes to popup word tapping

v1.4 14/03/04

- * Added "mouse" support for Journey, ZorkZero etc, to use tap the screen

- * Added ARMlet for pictures, gives faster picture drawing on OS5 and later
- * Picture file now cached in memory if lots of free heap RAM for speedup.
- * Fixed crashes and improved expansion card support
- * Added Hotsync registrations for story and graphics files
- * Fixed TH55 screen size
- * T3 screen slide redraw fix

v1.3 04/03/04

- * Fixed menu problems with OS4 devices
- * Fixed ROM version checking for devices such as 610C
- * Added memory card selection to preferences
- * Some screen drawing speed ups

v1.2 25/02/04

- * Added V6 Graphics (MG1) support!!!
- * Added option for UX50 to disable wide landscape support
- * Fixed messy screen clearing on some games
- * Fixed line input erasing and cursor positioning
- * Fixed Shogun ZC_GAP displaying, gets rid of square character
- * Save files automatically have ".sav" appended

v1.1 09/02/04

- * Change to line input event processing to fix issues with menu shortcuts, etc.
- * Added transcription support, i.e. "script/unscript" command.
- * Added Frotz Hotkey support with Debugging, Recording and Playback.
- * Fixed colour issues with popupforms, DIA and terpetude.
- * Fixes to Clie NX display and DIA.
- * Fixes to Insertion Point, and changed its colour to stop it disappearing.
- * Added "Free Heap RAM" menu option.

v1.0 24/01/04

- * First release

C: Links

rec.arts.int-fiction: <http://groups.google.com/group/rec.arts.int-fiction/topics>

rect.games.int-fiction: <http://groups.google.com/group/rec.games.int-fiction/topics>

The Interactive Fiction Archive: <http://www.ifarchive.org/>

Baf's Guide to the Interactive Fiction Archive: <http://www.wurb.com/if/index>

IFWiki: http://www.ifwiki.org/index.php/Main_Page

Interactive Fiction Reviews: <http://www.ifreviews.org/>

Society for the Promotion of Adventure Games (SPAG): <http://www.sparkynet.com/spag/>

IFDB, the Interactive Fiction Database: <http://ifdb.tads.org/>

Interactive Fiction Ratings: <http://www.carouselchain.com/if/>