Adventure

David Kinder

COLLABORATORS			
	TITLE : Adventure		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	David Kinder	February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Adventure

1.1 Adventure

```
                Adventure was the very first text adventure. It was started by  ↩
                   Will
Crowther, who wrote a Fortran simulation of the Bedquilt cave system. Don
Woods added magic and puzzles, and in 1977 Adventure was released to the
world. Presented here is a version very close to the original (the source
files date from July 1977). There is a comment about the
                Amiga
                 interface, a
description of the
                origins
                 of this version and a
                walk through
                 if you get
stuck. A
                C version
                 was written by Jim Gillogly at The Rand Corporation and
incorporated into the BSD Unix distribution.

The original has been modified and extended by many people. Also included
are several extended versions:
                Adventure 2.5
                ,
                Adventure 3
                 (aka Adventure
550),
                Adventure 4
                 and
                Adventure 6
                 (aka Adventure 551) all came with a
description of their origin, which can be accessed by these links. Buried
deep within Adventure 6 is a
                history
                 of this branch of Adventure. As well
as this, you can look at the full Adventure family
                tree
                 (as compiled by
Russell Dalenberg).
```

Also available are some
notes
on the Amiga implementation of these games.

1.2 Amiga Notes

These ports have an Amiga specific front-end, with proper command line editing, a command line history (use the cursor up/down keys to step through previously entered commands) and file requesters for the save and restore operations. They require at least Kickstart 2.04. The porting was performed by

David Kinder
kinder@teaching.physics.ox.ac.uk

Release Notes

* Added Unix Adventure.

* Added Adventure and Adventure 551.

* Added Adventure 2.5.
* Updated Adventure 550 and Adventure 4 to v1.1 after correcting a problem with the file requesters.

* First release of Adventure 550 and Adventure 4.

1.3 Adventure 2.5

(The author apologises for the style of the code; it is a result of running the original Fortran IV source through a home-brew Fortran-to-C converter.)

1.4 Adventure 3

From kcwellsch@watdragon.UUCP (Ken C. Wellsch) Mon Jul 7 15:10:06 1986
Newsgroups: net.sources.games
Subject: Adventure -- part 0 of 7
Organization: U of Waterloo, Ontario

I have been meaning to submit this game for some time but never seemed to find time to do it. This version of Adventure is taken from a Zerox Sigma-9 (rest her soul!), originally written by Dave Platt of Honeywell under CP-V (in Fort-77). I rewrote it into Ratfor many years ago and a couple years ago rewrote it again in C. This is the 550 point version of Adventure (for those who only know the 350 point original).

I had a lot of fun playing it on the SIGMA-9, and rewriting it, I hope
you enjoy it too! Watch out, having the database can certainly ruin
the game if you look at it!

The system components such as restricting the programs use etc. have
been left incomplete. I had done them back in the old Ratfor version
under V7 but each system can be quite different about determining load
etc. so I didn't bother (a lame excuse I know).

```
+---------------------------------------------------------------------
|
|                        CP-V Adventure
|
|    Originally written by David Platt (from the 350 point Adventure),
|    UNIX C Version written by Ken Wellsch, with several modifications.
|
|    Original release notice from 12/1/79:
|
|        FOR THE OFFICIAL STUFF: PERMISSION IS HEREBY GRANTED TO ALL
|        USERS TO POSSESS, USE, COPY, DISTRIBUTE, AND MODIFY (BUT NOT
|        TO SELL) THE PROGRAMS AND FILES IN THIS PACKAGE.
|
|        IF YOU HAVE ANY PROBLEMS GETTING THIS TURKEY INSTALLED, FEEL
|        FREE TO WRITE OR CALL - AND HAVE FUN, FOLKS
|
|            DAVID PLATT
|            HONEYWELL LOS ANGELES DEVELOPMENT CENTER
|            5250 WEST CENTURY BOULEVARD
|            LOS ANGELES, CALIFORNIA  90045
|            (213) 649-6870 x253
|            HVN 369-1253
|
|    Current release notice as of 8/11/85:
|
|       Permission is hereby granted to all users to possess, use, copy,
|       distribute, and modify the programs and files in this package
|       provided it is not for direct commercial benefit and secondly,
|       that this notice and all modification information be maintained
|       along with the package.
|
|            Ken Wellsch
|            University of Waterloo
|            CS Graduate Department
|            Waterloo, Ontario  N2l 3G1
|            (519) 888-4518
|
|
|History:
|
|      1979 winter:
|            -- Running on XEROX SIGMA-9 under CP-V.
|               Written in Fortran-77 by David Platt.
|
|      1982 winter:
|            -- Completely rewritten for PDP 11/44 under UNIX Version 7.
|               Written in Rational FORTRAN (ratfor) by Ken Wellsch.
|      1984 fall:
```

```
|            -- Once again rewritten, this time for a VAX 11/780,
|               under UNIX BSD 4.2. Written in C by Ken Wellsch.
|
|
|       Copyright (c) 1979 David Platt    Database & Methods
|       Copyright (c) 1984 Ken Wellsch    C Code & Modifications
|
+-------------------------------------------------------------------------

--

-------------------------------------------------------------------------------
-- Ken C. Wellsch, CS Dept., U. Waterloo, Waterloo, Ontario, Canada N2L 3G1 --
CSNET:kcwellsch%watdragon@waterloo.csnet                   HOME:1-519-746-4984
ARPA :kcwellsch%watdragon%waterloo.csnet@csnet-relay.arpa  OFFI:1-519-888-4518
UUCP :...!{allegra|clyde|linus|utzoo|inhp4|decvax}!watmath!watdragon!kcwellsch
-------------------------------------------------------------------------------
```

1.5 Adventure 4

Hello and welcome to Adventure4!

No, this is not Yet-Another-Adventure variant. It may stick to a single verb/noun commands (shock , horror! - no adjectives, no prepositions... not even a special kludge for putting objects into other objects!!), but it has something many other pretenders do not have - character.

To start with, Adventure4 re-merges the two divergent extensions of the the original game (Peter Luckett's and Jack Pike's Adventure II and Dave Platt's Adventure 550), with some minor improvements all of its own.

More importantly, however, the game had benefited from being a part of an official games package on a very large installation, which allowed me to scan through literally thousands of game logs generated by a large number of players. Adventure4 incorporates a number of modifications and improvements, based solely on this unique experience of seeing the game as it is played and *not* as the author might have wished or expected it to be played. Judging from players' feedback, it does make a difference.

If you have a hard disk, I suggest you copy over to it both ADV4.EXE and ADVENTUR.DAT. The text database size is far too large for most PCs to hold in memory, so the game constantly accesses the data file. ADV4.EXE does a lot of clever internal buffering to make up for this, and the game *is* playable from a floppy, but there is no point suffering needlessly, is there?

Anyway, have a go and see what you think! Afterwards, if you feel strongly enough about it, don't hesitate to let me know what your thoughts were, be they rude or polite - all opinions welcome.

3rd November, 1991

Mike Arnautov

Glaxo Group Research,
Greenford Road,
Greenford,
Middlesex UB6 0HE,
England.

mla1290@ggr.co.uk

1.6 Adventure 6

'Generic Adventure 551' is a cleaned up and enhanced version of the old classic text game 'Adventure'. It is based on a version close to the seriously non-portable (and buggy) version for Primes posted last spring. I have done a fair job of cleaning it up, portabilizing it, and debugging, but it is still Fortran spaghetti. For you with no Fortran compilers, this is the excuse you need to get "f2c" working!!! It has worked on the IBM-PC, VMS, the MIPS 120, the IBM RISCstation 600, the VAX/Ultrix, and a Prime. It will however expose toy Fortran compilers as just that - toys. I would like to thank Larry Estep for finding some hidden bugs.

For those that have not experienced this, the very first text adventure game, it is a romp through the darkest reaches of Colossal Cave in search of riches. You will find evocative descriptions of the darker, more mysterious places of the earth, and severely test your abilities to map the contortions of the cave. You will find some hard - and some deceptively easy - puzzles to solve.

Doug McDonald

This is the hopefully portable ADVENTURE 551 program. Its BETA!!!!!
I've tested it as well as I can, but you never know!!!! It comes as three Fortran files, aamain.f, the main game program; asetup.f, a program that reads the ASCII data file ADVDAT and makes a runtime data file ADVTXT; asubs.f, that needs to be liked with each of the above; and ADVDAT, the ASCII data file itself (note all caps for Unix users).

It is based on the old old old Fortran one posted recently in comp.sources.games - or rather one very close to that. I added a small piece of game, and the version I started from was lacking a tiny (and uncompleted) part. But mostly I fixed bugs. Oh! My! What a mess it was. You think it is bad now? Well, it is still F66 spaghetti, but previously it was full of Holleriths, system dependencies, and just plain bugs. Code bugs, game bugs, out of range subscripts ad nauseum. I played it to completion, and got all 551 points. That means it is possible to win. But it does NOT guarantee no bugs - as I found a bad one just playing it this last time. So have other testers. The only blatent non-portability is that it assumes ASCII.

For Unix people with Fortran, first make sure that the file is ADVDAT, all upper case, make sure the name of your Fortran compiler is right in the makefile and then type 'make'. To play, type "adventure".

If your Unix system lacks the MIL std Fortran functions ior, ieor, and iand,

you'll have to make up some with whatever bit functions you do have.
It's up to you. These functions just take two integer variables and
return the appropriate bitwise functions OR, XOR, and AND. If you get them
wrong the asetup program will die at section 9 of the data file.

For folks on MS-DOS, rename the .f files to .for. Then compile and link
asetup with asubs to make asetup.exe. Make SURE to have your Fortran
compiler use 4-byte integers!!! (Use /4I4 for Microsoft Fortran). Run
asetup.exe. Then compile aamain.for and link with asubs.obj to make
aamain.exe. Rename aamain.exe to "adventure" and play the game.

For VMS folks, do as MS-DOS except that you have to add ".dat" to the advdat
filename, and compile with /check=noover. This last is necessary as the
random number generator actually depends on overflows.

For folks with Unix systems and no (or broken!!! - Sun 3's) Fortran
compilers, it works with the "f2c" program available from research.att.com.
This took me a bit of work on our Mips machine. It is easy to generate the
f2c program and its two libraries libF77 and libI77 but it didn't link
right. The solution was to use the librarian (ar) to make a single library
libF2C with all the .o files from BOTH libraries, put that in the directory
you are working with and tell cc to use that as an ordinary object file i.e.

```
f2c *.f
<<edit iors.c as instructed in the iors.f file!!!!!>>
cc -c *.c
cc asetup.o asubs.o iors.o libF2C -lm -lc
a.out
cc aamain.o asubs.o iors.o libF2C -lm -lc
mv a.out adventure
```

(Note the lack of a -l before the libF2C - read the READMEs that
come with f2c.)

then just play! I suppose I could distribute the .c files f2c produces,
but the main problem is that I would also have to distribute the libs,
which are the big problem anyway.

It hopefully will just work. There are two places to look at the
source code: First, look in asubs.f for the string ?????, which occurs
twice. Thereabouts is code which works on VMS, the IBM-PC, and f2c to allow
you to get a prompt on the same line as your response. It fails miserably in
F77 on my MIPS Unix machine. If you can make this or some other prompting
mechanism work on your machine, PLEASE report it to me. Second, in aamain.f
look for the the string ?????. There you can uncomment a line to activate
Wizard mode, which allows a modest amount of cheating. Uncautious use
of it, however, can prevent actually winning.

This version compiles and runs using f77 on a MIPS unix box, on the VAX
using VMS VAX Fortran or using Ultrix and its Fortran compiler (though you
must supply the bitwise functions), on the IBM PC using either the MicroWay
386 32-bit compiler or Microsoft Fortran 4.1 in large model, and on the IBM
Riscstation 6000.

It compiles but does NOT run on a Sun 3 (with a F77 compiler with a 1988
date) due to the compiler simply being unable (with no help from
switches) to compile correctly the main program. Tests indicate that
this compiler barfs on files with much more than 400 lines of significant

code. If anybody can get it to run on a Sun 3, please let me know how.
I find it hard to believe that they would let out a compiler this limited.

Doug McDonald (mcdonald@aries.scs.uiuc.edu)

Adventurer Grandmaster!!!
(its not easy, even if you do read the source code)

1.7 Adventure 6 History

ADVENTURE was originally developed by William Crowther, and later substantially rewritten and expanded by Don Woods at Stanford Univ. According to legend, Crowther's original version was modelled on an a real cavern, called Colossal Cave, which is a part of Kentucky's Mammoth Caverns. That version of the game included the main maze and a portion of the third-level (Complex Junction - Bedquilt - Swiss Cheese rooms, etc.), but not much more. Don Woods and some others at Stanford later rewrote portions of the original program, and greatly expanded the cave. That version of the game is recognizable by the maximum score of 350 points. Some major additions were done by David Long while at the University of Chicago, Graduate School of Business. Long's additions include the seaside entrance and all of the cave on the "far side" of Lost River (Rainbow Rm - Crystal Palace - Blue Grotto, etc.). The castle problem was added in late 1984 by an anonymous writer. Thanks are owed to Roger Matus and David Feldman, both of U. of C., for several suggestions, including the Rainbow Room, the telephone booth and the fearsome Wumpus. Most thanks (and apologies) go to Thomas Malory, Charles Dodgson, the Grimm Brothers, Dante, Homer, Frank Baum and especially Anon., the real authors of ADVENTURE.

1.8 The Original Adventure

This is a resurrection of the old Adventure, written for the DEC-10 and ported to the PDP-11/70, ported this time to the MS-DOS environment. No new features have been added. The only changes made were those required to get the program to compile using the Microsoft FORTRAN V5.0 compiler. This entailed a number of minor changes, such as replacing the "accept" statements with "read *" statements, and similar trivial items. Also, two string handling subroutines ("getin" and "a5toa1") were totally rewritten using the FORTRAN 77 string functions and operators. Four string handling utility subroutines were added: "upcase," "shiftc," "lchar," and "fchar." The old sources for "getin" and "a5toa1"are included as "*.old" files. They are not needed; I simply included them in case anyone wanted to see them. I also slightly modified the input subroutine "getin" to accept lower-case letters. All output, however, is retained in upper-case letters, as it originally was. The message of the day file, motd.for, was a no-op when I received the code. I added some code to support this feature.

No make file is included. To create the program, I compiled all the subprograms (but not advent.for), and put the resulting objects in a library, which I called advlib.lib. The compiler command line is:

```
  fl /c /AL file.for
```

Once the library was created, I compiled the main program and created
the executable with the line:

```
  fl /AL advent.for /link advlib.lib
```

(The /AL switch forces use of the large memory model.) Optimizations
are at their default state. You will get a number of warning messages
about variables defined but not used. These are benign. Also, when
compiling the large file search.for, you will likely get a complaint
that it is too big for the optimizer. No problem; it works fine.

The file text.txt contains all the data for the program. When the program
starts, it reads this sequential file, and uses the data to create a
random-access file, which is deleted when the program exits. This file
requires approximately 70K bytes on your disk. Initialization takes about
four seconds on my 25MHz, 386-based system, using a Microscience 160 Meg
hard drive with a random access of 28 milliseconds. While it is possible
to run this program from a floppy, I don't recommend it. It can be
painfully slow to initialize, and there are delays when the random access
file is read.

The save and restart, under the old DEC system, were simply core saves,
with a couple of variables set to show that a restart is in progress.
I haven't yet implemented a save-to-disk feature. This means, unfortunately,
that you must start over each time you play. Ultimately I will get this
fixed.

The wizard stuff, which permits you to specify hours when the cave
is closed, is now implemented.

Address questions and comments to Don Ekman at

   3586 Berry Way
   Santa Clara, CA 95051
   USA

or, Internet:

   ekman@wdl30.wdl.loral.com

1.9 Walk Through for Adventure

ADVENTURE WALKTHROUGH

This is not claimed to be an optimized walkthrough, but it comes reasonably
close and, if you don't get killed by a little dwarf and if the pirate shows
up before you finish, you can get all the points.

When you get well into the cave you will encounter some nasty dwarves. The
first one simply tosses an axe at you and runs away. The axe misses. Get
the axe; you'll need it when you see the dwarves again. On your second
encounter with the dwarves, you'll see one or more, who will throw knives at
you. The first salvo always misses. Toss the axe at a dwarf, fetch it

again, and keep tossing until you have killed all the dwarves present. You'll need to do this each time a dwarf appears. If you don't, you'll get killed. (Be sure to retrieve the axe after you have killed the last dwarf that is with you.)

Don't worry if a pirate appears and steals your treasures from you. You'll find them again later on.

Walkthrough: Answer "no" when you are asked if you want instructions.

You are standing in front of a building. Enter the commands as listed below (first column only; the second column shows what the command does or where it takes you), except deal with dwarves whenever they appear.

1.	in	inside building
2.	get lamp	
3.	on	turn on lamp
4.	plugh	y2
5.	plover	plover room
6.	ne	dark room
7.	get pyramid	
8.	s	plover room
9.	plover	y2
10.	s	low n/s passage
11.	get silver	
12.	n	y2
13.	plugh	inside building
14.	drop pyramid	
15.	drop silver	
16.	get water	in bottle
17.	plugh	y2
18.	s	low n/s passage
19.	d	dirty passage
20.	bedquilt	bedquilt
21.	slab	slab room
22.	s	west end of twopit room
23.	d	west pit
24.	water plant	
25.	u	west end of twopit room
26.	w	slab room
27.	u	secret n/s canyon
28.	reservoir	reservoir
29.	get water	
30.	s	mirror canyon
31.	s	secret n/s canyon
32.	d	slab room
33.	s	west end of twopit room
34.	d	west pit
35.	water plant	
36.	u	west end of twopit room
37.	e	east end of twopit room
38.	d	east pit
39.	get oil	
40.	u	east end of twopit room
41.	w	west end of twopit room
42.	d	west pit
43.	climb	narrow corridor

```
 44.  w                   giant room
 45.  n                   immense n/s passage (iron door)
 46.  oil door
 47.  drop bottle         you no longer need it
 48.  n                   waterfall cavern
 49.  get trident
 50.  w                   steep incline
 51.  d                   large low room
 52.  se                  oriental room
 53.  n                   misty cavern
 54.  w                   alcove
 55.  drop trident        to get through narrow passage
 56.  drop lamp
 57.  drop axe (if carrying it)
 58.  e                   plover room
 59.  get emerald
 60.  w                   alcove
 61.  get trident
 62.  get lamp
 63.  get axe (if here)
 64.  nw                  misty cavern
 65.  s                   oriental room
 66.  get vase
 67.  se                  swiss cheese
 68.  e                   soft room
 69.  get pillow          needed for vase
 70.  w                   swiss cheese
 71.  ne                  bedquilt
 72.  e                   complex junction
 73.  n                   shell room
 74.  open clam           with trident
 75.  d                   well, it will roll down, won't it?
 76.  d
 77.  get pearl
 78.  u
 79.  u                   shell room
 80.  s                   complex junction
 81.  u                   dusty rock room
 82.  e                   dirty passage
 83.  u                   low n/s
 84.  n                   y2
 85.  plugh               inside building
 86.  drop pillow         pillow first, then vase
 87.  drop vase
 88.  drop trident
 89.  drop pearl
 90.  drop emerald
 91.  xyzzy               debris room
 92.  get rod
 93.  e                   cobble crawl
 94.  get cage
 95.  pit                 top of small pit
 96.  e                   bird chamber
 97.  drop rod            rod scares bird
 98.  get bird
 99.  get rod
100.  w                   top of small pit
```

```
101.  d                   hall of mists
102.  n                   hall of mt king
103.  free bird           bird drives away snake
104.  drop cage           no longer needed
105.  sw                  secret e/w canyon
106.  w                   secret canyon
107.  kill dragon         no weapon needed
108.  yes                 yes, with bare hands
109.  get rug
110.  e                   secret e/w canyon
111.  e                   hall of mt king
112.  w                   west side chamber
113.  get coins
114.  e                   hall of mt king
115.  s                   south side chamber
116.  get jewelry
117.  n                   hall of mt king
118.  e                   hall of mists
119.  w                   east bank of fissure
120.  wave rod            create bridge
121.  drop rod            no longer needed
122.  w                   west bank of fissure
123.  get diamonds
124.  e                   east bank of fissure
125.  e                   hall of mists
126.  s                   nugget room
127.  get gold
128.  n                   hall of mists
129.  y2                  rock jumble (not y2)
130.  d                   y2
131.  plugh               inside building
132.  drop rug
133.  drop coins
134.  drop jewelry
135.  drop diamonds
136.  drop gold
137.  get food            for bear
138.  get keys            for locked chain
139.  plugh               y2
140.  s                   low n/s
141.  d                   dirty passage
142.  bedquilt            bedquilt
143.  w                   swiss cheese
144.  w                   east end of twopit room
145.  w                   west end of twopit room
146.  d                   west pit
147.  climb               narrow corridor
148.  w                   giant room
149.  get eggs            for troll
150.  n                   immense n/s
151.  n                   waterfall cavern
152.  w                   steep incline
153.  d                   large low room
154.  sw                  sloping corridor
155.  u                   s/w side of chasm
156.  toss eggs           pay troll
157.  ne                  n/e side of chasm
```

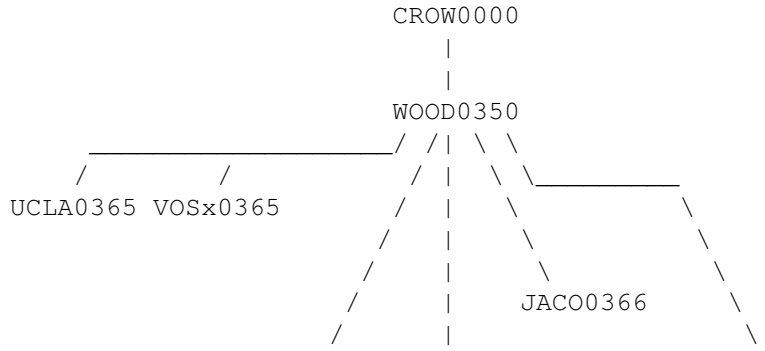
```
158.  ne                  corridor
159.  barren              outside barren room
160.  in                  bear in room
161.  feed bear           calm him a little
162.  unlock chain        with key, of course
163.  get chain
164.  get bear            to scare troll
165.  w                   outside barren room
166.  fork                fork in path
167.  ne                  warm walls
168.  e                   boulders
169.  get spices
170.  fork                fork in path
171.  w                   corridor
172.  w                   n/e side of chasm
173.  sw                  troll appears again
174.  free bear           scare off troll
175.  sw                  s/w side of chasm
176.  sw                  sloping corridor
177.  d                   large low room
178.  se                  oriental room
179.  se                  swiss cheese
180.  w                   east end of twopit room
181.  w                   west end of twopit room
182.  d                   west pit
183.  climb               narrow corridor
184.  w                   giant room
185.  fee
186.  fie
187.  foe
188.  foo                 that's what it says
189.  get eggs            magic
190.  s                   narrow corridor
191.  d                   west pit
192.  u                   west end of twopit room
193.  w                   slab room
194.  u                   secret n/s
195.  s                   secret canyon
196.  e                   secret e/w canyon
197.  e                   hall of mt king
198.  n                   low n/s
199.  n                   y2
200.  plugh               inside building
201.  drop spices
202.  drop chain
203.  drop eggs
204.  plugh               y2
205.  s                   low n/s
206.  s                   hall of mt king
207.  e                   hall of mists
208.  w                   east bank of fissure
209.  w                   west bank of fissure
210.  w                   west end hall of mists
211.  s                   all alike maze
212.  e                   maze
213.  s                   maze
214.  s                   maze
```

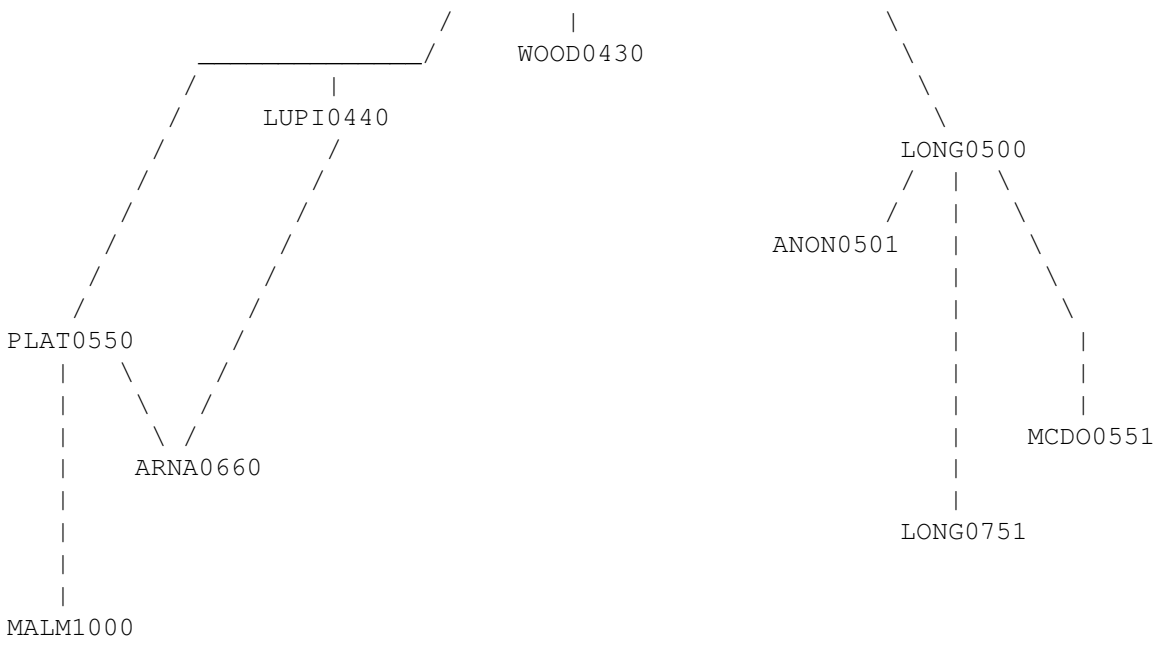
```
215.  s                   maze
216.  n                   maze
217.  e                   maze
218.  e                   maze
219.  nw                  dead end
220.  get chest (and anything else the pirate may have stolen)
221.  se                  maze
222.  n                   maze
223.  d                   bird chamber
224.  e                   e/w canyon
225.  e                   debris room
226.  xyzzy               inside building
227.  drop chest
228.  plugh               y2 (why two?)
229.  s                   low n/s
230.  d                   dirty passage
231.  w                   dusty rock room
232.  d                   complex junction
233.  e                   ante room
234.  get magazine        it's addressed to witt
235.  e                   witt's end
236.  drop magazine       the elusive one point
237.  Just do anything, but don't go west from here; you'll never get out
      that way.  Any other direction will eventually (with low probability)
      get you out.  Move around in the cave until something interesting
      happens.
238.  sw                  s/w end
239.  get rod             this one is dynamite
240.  ne                  n/e end
241.  drop rod            plant explosive
242.  sw                  stand clear
243.  blast               make a hole in wall to main office
```

If all went well, you have finished with a maximum score.

If the pirate never appears, you probably won't make it. When you reach the dead end where the chest is supposed to be, you could backtrack (if you know how) through the maze and then forward again until he appears. If this takes too many turns your lamp batteries will wear out before the end game, and then it's tough beans.

1.10 Adventure Family Tree






```
CROW0000 -- Initial "Adventures" game by Will Crowther
WOOD0350 -- Original 350-point version in FORTRAN by Don Woods
EKMA0350 -- Microsoft FORTRAN port of WOOD0350 by Don Ekman
BAGG0350 -- TADS port of EKMA0350 by David M Baggett
GILL0350 -- Unix/C port of WOOD0350 by Jim Gillogly
MALM0350 -- AGT port of WOOD0350 by David Malmberg
VMCM0350 -- PL/I port of WOOD0350 for VM/CMS
UCLA0365 -- UCLA extension to WOOD0350
VOSx0365 -- FORTRAN extension to WOOD0350
JACO0366 -- Prime FORTRAN extension of WOOD0350 by John Jacobsma
WOOD0430 -- C extension of WOOD0350 by Don Woods
LUPI0440 -- Extension of WOOD0350 by Peter Luckett & Jack Pike
LONG0500 -- DEC FORTRAN extension of WOOD0350 by David Long
IITa0500 -- Initial I.I.T. port of LONG0500 to Prime FORTRAN
IITb0500 -- Final I.I.T. port of LONG0500 to Prime FORTRAN
ANON0501 -- Prime FORTRAN extension of IITa0500
PLAT0550 -- FORTRAN extension of WOOD0350 by David Platt
WELL0550 -- Unix/C port of PLAT0550 by Ken Wellsch
MCDO0551 -- FORTRAN extension of IITa0500 by Doug McDonald
ARNA0660 -- Merge LUPI0440 with PLAT0550 by Mike Arnautov
LONG0751 -- DEC FORTRAN extension of LONG0500 by David Long
MALM1000 -- AGT extension of PLAT0550 by David Malmberg
```

1.11 Amiga version of Adventure

This very early version of Adventure still has the all-capitals text of the PDP-10 version. While this is authentic, it can be somewhat annoying after a while, so the program has the ability to convert the text to a more normal output. To do this either run the game from the Shell with

```
  Adventure -c
```

or, for Workbench, add the tooltype "CASE" to the icon.

1.12 Unix Adventure

When this version of Adventure is run from the Workbench, a file requester to load in a saved game opens. To start a new game, just cancel the requester.

The following license is from the source code: