

## Wizardry Proving Grounds v3.1 Project - Apple II Version

List of code bugs and issues and enhancements for the Wizardry Proving Grounds v3.1 project.

The baseline for this project is the re-engineered Pascal source code (and assembler code) from the Wizardry\_I program by Thomas William Ewers. This project would not exist without it.

Compilation of bugs and issues as of 24 Mar 2024 by Eric Labelle (snafaru@zimlab.com).

The source info for this list includes E. Labelle, Thomas W. Ewers, DDG Ahab, Denis Molony, and others.

NOTICE: This document is essentially complete now as we have actually ran out of things to fix as per our latest release.

Wizardry Proving Grounds v3.1 Web Site and GitHub:

Web Site: <https://www.zimlab.com/wizardry/proving-grounds-v3>

GitHub: <https://github.com/snafaru>

More details of bugs and issues fixed and enhancements are here:

<https://www.zimlab.com/wizardry/proving-grounds-v3>

Lines with yellow highlights indicate an issue that might still need be looked at.

#001 - The Bishop bug

=====

Status: Fixed in WC001. Recompiled. Verified. Closed.

The bug:

If your character is a Bishop, it can identify the true name of an item. These are in its inventory slots 1 through 8. Pressing other keys besides 1 through 8 can have the following benefits:

- 9 : The Bishop gains 100 million experience points.
- S : The character below the Bishop gains 100 million experience points.
- J : The character below the Bishop gains 100 million gold pieces.

The fix:

This bug is due to a programming error. The code is similar to "IF (CH>="1") OR (CH<="8") THEN". The "OR" should have been an "AND".

Why this fix: To prevent exploit and to prevent game corruption if other keys are pressed.

#002 - Latumapic does not work

Status: Fixed in WC003. Recompiled. Verified. Closed.

The bug: It appears that it was supposed to loop through all four groups and identify all of them. But since the array index is LLBASE04 instead of the GROUPI incrementor, it instead picks a random group and identifies it four times.

Original Code:

```
BEGIN
        FOR GROUPI := 1 TO 4 DO
            BATTLERC[ LLBASE04].A.IDENTIFI := TRUE;  (* BUG?
WITH BASE04*)
END;
```



#003 - The silencing effect of Montino is meant to wear off, but it doesn't

Status: Fixed in WC009. Recompiled. Verified. Closed.

Answer: Applied code adjustments as per DDG Ahab's suggestion below.

The bug:

```
                FOR X := 0 TO ALIVECNT - 1 DO
                    IF BATTLERC[ GROUPI].A.TEMP04[
ALIVECNT].INAUDCNT > 0 THEN
                        BATTLERC[ GROUPI].A.TEMP04[
ALIVECNT].INAUDCNT :=
                            BATTLERC[ GROUPI].A.TEMP04[
ALIVECNT].INAUDCNT - 1
The array index should be X rather than ALIVECNT.
```

Furthermore, the outer function forgets to call this procedure on the fourth monster group.

```
BEGIN (* HEALHEAR *)
    DECINAUD( 0, PARTYCNT);
    DECINAUD( 1, BATTLERC[ 1].A.ALIVECNT);
    DECINAUD( 2, BATTLERC[ 2].A.ALIVECNT);
    DECINAUD( 3, BATTLERC[ 3].A.ALIVECNT)
END; (* HEALHEAR *)
```





1) 2 UNDEAD KOBOLDS (2)

CT6502'S OPTIONS

F)IGHT    S)PELL    P)ARTY  
R)UN      U)SE        D)ISTPEL

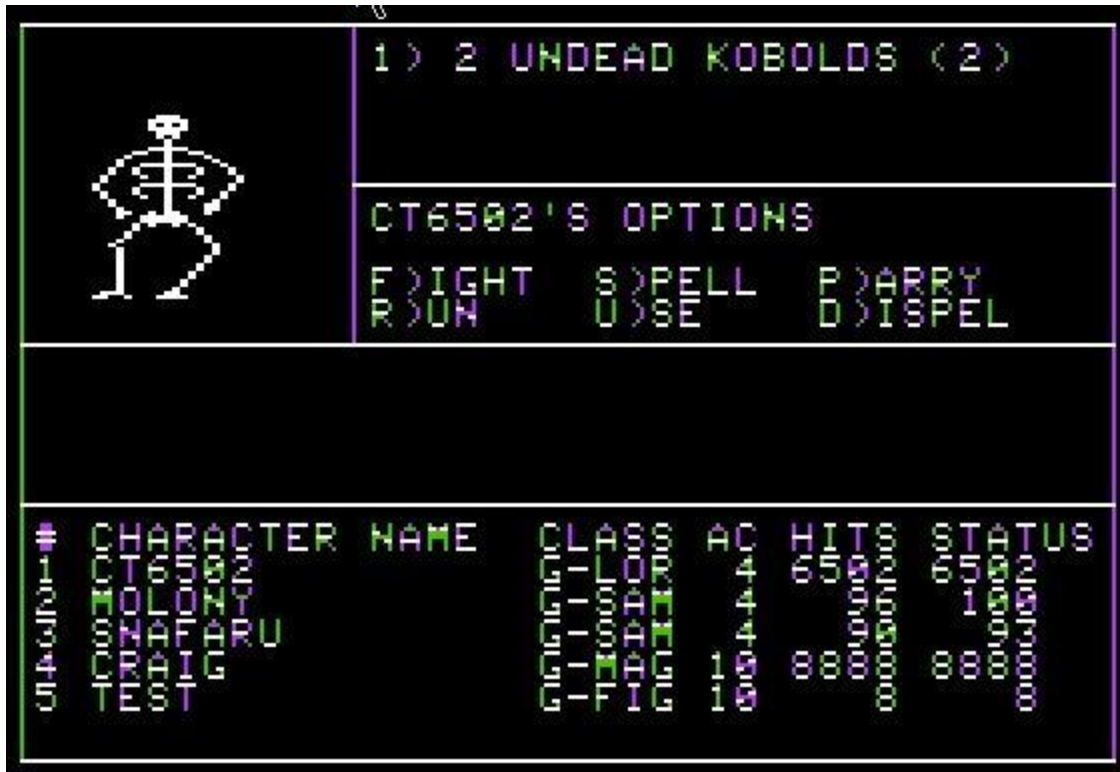
#	NAME	HP	MP	EXP	LEVEL	STATUS
1	CONDOR	100	0	0	1	OK
2	CONDOR	100	0	0	1	OK
3	CONDOR	100	0	0	1	OK
4	CONDOR	100	0	0	1	OK
5	CONDOR	100	0	0	1	OK
6	CONDOR	100	0	0	1	OK
7	CONDOR	100	0	0	1	OK
8	CONDOR	100	0	0	1	OK
9	CONDOR	100	0	0	1	OK
10	CONDOR	100	0	0	1	OK
11	CONDOR	100	0	0	1	OK
12	CONDOR	100	0	0	1	OK
13	CONDOR	100	0	0	1	OK
14	CONDOR	100	0	0	1	OK
15	CONDOR	100	0	0	1	OK
16	CONDOR	100	0	0	1	OK
17	CONDOR	100	0	0	1	OK
18	CONDOR	100	0	0	1	OK
19	CONDOR	100	0	0	1	OK
20	CONDOR	100	0	0	1	OK



1) 2 UNDEAD KOBOLDS (2)

AND THE PARTY!  
LOLLAPALOOZA!

#	NAME	HP	MP	EXP	LEVEL	STATUS
1	CONDOR	100	0	0	1	OK
2	CONDOR	100	0	0	1	OK
3	CONDOR	100	0	0	1	OK
4	CONDOR	100	0	0	1	OK
5	CONDOR	100	0	0	1	OK
6	CONDOR	100	0	0	1	OK
7	CONDOR	100	0	0	1	OK
8	CONDOR	100	0	0	1	OK
9	CONDOR	100	0	0	1	OK
10	CONDOR	100	0	0	1	OK
11	CONDOR	100	0	0	1	OK
12	CONDOR	100	0	0	1	OK
13	CONDOR	100	0	0	1	OK
14	CONDOR	100	0	0	1	OK
15	CONDOR	100	0	0	1	OK
16	CONDOR	100	0	0	1	OK
17	CONDOR	100	0	0	1	OK
18	CONDOR	100	0	0	1	OK
19	CONDOR	100	0	0	1	OK
20	CONDOR	100	0	0	1	OK



The bug: There are supposed to be seven possible effects, with less powerful ones more likely to be selected. Instead, each spell has three possible effects with equal distribution, and two of the effects are impossible.

Haman: HAMCURE, HAMMAGIC, HAMHEAL

Mahaman: HAMCURE, HAMSILEN, HAMTELEP

Unused: HAMPROT, HAMALIVE

Reference: Jun 27, 2014, 5:40:39 PM in

<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

Bug4:

-----

What are the possible results for HAMAN and MAHAMAN?

Here is the code with a very bad bug. See if you can spot it.

```
CASE RANDOM MOD 3 * MAHAMFLG OF (* MAHAMFLG IS 6 OR 7 *)
0, 1, 2, 3, 4, 5: HAMCURE;
7, 8, 9, 10, 11: HAMSILEN;
```

```
12, 13, 22, 23: HAMMAGIC;  
14, 20, 21: HAMTELEP;  
6, 15, 19: HAMHEAL;  
17: HAMPROT;  
16, 18: HAMALIVE;
```

Hint: If you search the internet, you will not find all 7 of these actions for HAMAN and MAHAMAN.

.  
. .  
.

And to be more specific, the case expression can only evaluate to the values 0, 6, 7, 12, and 14. It is likely that they really wanted something like:

```
CASE RANDOM MOD (3 * MAHAMFLG) OF (* MAHAMFLG IS 6 OR 8 *)
```

Note the two changes: the parenthesis and the MAHAMFLG values. It appears they wanted to have the results weighted in favor of the action HAMCURE for example. Completely butchered code for the lack of the parenthesis.



#005 - Regeneration, Degeneration: HEALPTS is probably meant to stack

Status: Fixed in WC010. Recompiled. Verified. Closed.

Answer: Adjust code to have regeneration and degeneration items stack. Poison and stacking poison from chest traps are also taken in account. The Deadly Ring is now actually deadly.

The bug: HEALPTS (hit points regeneration and degeneration) is probably meant to stack but instead, it just takes whatever item has the biggest heal value. Consequently, the DEADLY RING's HP draining property has no effect; the base value of 0 takes priority over the ring's negative value, being larger.

```
CRAIG ELF G-MAGE
      STRENGTH 18      GOLD      2757
      I.Q.      100     EXP      20842466
      PIETY     100
      VITALITY  100     LEVEL    77      AGE    20
      AGILITY   100     HITS    888/888    AC    10
      LUCK      5      STATUS   OK

      MAGE  9/9/9/9/9/9/9
      PRIEST 0/0/0/0/0/0/0

*=EQUIP, -=CURSED, ?=UNKNOWN, #=UNUSABLE
1) DEADLY RING

YOU MAY E)QUIP, D)ROP AN ITEM, T)RADE,
      R)EAD SPELL BOOKS, CAST S)PELLS,
      U)SE AN ITEM, I)DENTIFY AN ITEM,
      OR L)EAVE.
```





#007 - TWIZLONG/ADDLONGS large number calculations bug

Status: Fixed in WC025. Recomplied. Verified. Closed.

Answer: The Addlongs procedure that adds two numbers has a bug where part of the first value is passed on to the second value because they have the same name thus having a addition/multiplication side effect starting at 10000. This causes the experience points of some monsters to be artificially high.

The fix: Updated the declaration of the procedure to have the second value as a constant.

Why this fix: To fix the Addlongs procedure bug. Also, this makes current monsters or future ones in another scenario to have a more predictable experience allocation. Reported by: DDG Ahab.

The issue: Too much experience awarded by certain monsters.

References: Thomas W. Ewers

Jul 15, 2014, 4:43:46 AM

Jul 15, 2014, 11:41:13 AM

Jul 15, 2014, 9:28:29 PM

Jul 16, 2014, 3:23:47 AM

Jul 16, 2014, 11:12:09 AM

on

<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

Ex: Monsters with at least 80 spell resistance award much more XP than intended due to a long integer multiplication bug.

#008 - The Lord's Garb and other non-weapon item do not grant certain properties

Status: Closed. Fixed in Wizardry Restored v2.2 26-JAN-23.

Answer: In the spirit of standard fantasy role-playing, armor type equipment is not supposed to give offensive values. These attributes should be removed from the Lord's Garb in the Wizardry database. This seems to have been wishful zeal. No code corrections necessary.

The issue: The Lord's Garb's is flagged to grant critical hits and 2x damage towards weres, undead, and demons, but it won't, because the game only checks weapons for these flags.

#009 - First round of combat balance issues

Status: Fixed in WC023. Recompiled. Verified. Closed.

Answer: Modified the code to disallow casting spells and using breath during surprise rounds for both monsters and characters.

Some early thoughts: Normally, in many traditional fantasy dice role-playing games, a magic user would not have time to prepare his spell on a surprise round (no way to tell exactly when the enemy will show up or be in range, the magic user cannot keep his spell in a "prepared" state indefinitely), they would cast only on the second round. Breathers do not have to prepare their "spells". On one hand if the party that surprises the monsters can use their spells, that is quite a survival boon for them. In the code, your party has more chances of surprising the monsters than they have surprising you. This means on a run for Werdna getting one or two lucky breaks (surprise) where you can cast spells actually makes it easier to reach Werdna. If Dragon Zombies get 1 free surprise round of breathing and get the initiative and breathe at the beginning of the second round, it is unlikely your party will survive... that is why high agility is very important on your spell casters to they go before the breathers. Often, getting surprised by level 10 Mages for example means a total wipeout of your party. Would disabling spell casting and breath on a surprise round make the game too easy? Probably not. We are more like fixing things that was killing the fun in the game... if the Dragon Zombies wipe out your entire party twice or more in a row, a player might just give up and stop playing. While we are at it, why not disable

monster's draining abilities on their surprise round? It is not clear here what could/should be disabled on a surprise round without affecting the game balance.

The issue: Getting wiped out on round 1 because you got surprised by spellcasters or breathers. Wiz3 alleviated this by disallowing spells during surprise rounds, though breathers will still ruin you.

#### #010 - Manifo issue

Status: Fixed in WC014. Recompiled. Verified. Closed.

Answer: Adjusted the code so that the monsters have the status PLYZE instead of ASLEEP if affected by Manifo. Reduce the monsters' resistance to Manifo so that it now sits in between the original setting and Legacy of Llylgamyn (over-powered). Ex: Now a level 3 monster has a 65% chance to resist Manifo, compared to a 60% chance to resist Katino. The second fix was done out of necessity and for negligible coding space (less than 512 bytes left on disk for coding) and can easily be reversed or changed if necessary.

Brain juggling:

Although level 1 monsters having a 60% chance to resist Manifo is a bit harsh the objective of this project is to fix bugs. It would be easy to make Manifo to be overpowered, this would be a game balance item, maybe this is for Wizardry PG 4.0.

The Priest spell Manifo (hold monster) is the counterpart of the Mage spell Katino (sleep monster). Although Manifo has less chance of success than Katino, Monsters have much, much less chance to recuperate from Manifo if their status is PLYZE. These are very low-level spells, just like in dice fantasy role playing games do not expect too much from low level spells. Changing the % effect chance to be the same as Legacy of Llylgamyn would make the Manifo spell too powerful. Consider this: if your party has 2 Priests instead of a Priest and a Mage, then casting 2 Manifos should do the trick for a group of low-level monsters.

Here is extra information of what the code is before changes:

A level 5 monster has 0 chances of being affected by either Manifo or Katino.

Katino success chance (here the lower the better):  
20 \* BATTLERC[ CASTGR].B.HPREC.LEVEL, 'SLEPT', 3)

Manifo success chance:  
50 + 10 \* BATTLERC[ CASTGR].B.HPREC.LEVEL, 'HELD', 0)

Both have the same effect:  
CASE DAMTYPE OF  
0, 3: BATTLERC[ GROUPI].A.TEMP04[ CHARI].STATUS := ASLEEP;

At monster level 5 both have no chance of success.

Possible status effects:  
TSTATUS = (OK, AFRAID, ASLEEP, PLYZE, STONED, DEAD, ASHES, LOST);

Monster recovery from ASLEEP:  
TRYHEAL( 20 \* BATTLERC[ T2].B.HPREC.LEVEL);

Monster recovery from PLYZE:  
TRYHEAL( 7 \* BATTLERC[ T2].B.HPREC.LEVEL);

The issue: Manifo is nearly useless. It inflicts SLEEP (not PARALYZE, though the status exists), but even the weakest monster has a 60% chance to resist. Perhaps this is marginally useful against creeping coins, who can't be Katino'd or dispelled, but I can't think of any other situation where it would be slightly useful.

#### #011 - Loktofeit issues

Status: Fixed in WC028. Recompiled. Verified. Closed.

Answer: Updated the code so that the cast chance is the same as Wizardry 3 - Legacy of Llylgamyn which is (65 + Character Level)%.

Early thoughts: Nobody will disagree on this issue. Suggestion: Make Loktofeit teleport to a random location when used in combat. Maybe this would be something for Wizardry PG Expansion 4.0. You could even add the status effect "Afraid" on the party when this spell is used and have it displayed on screen.

The issue: The penalty for using Loktofeit is such that it is never used.

```
IF (RANDOM MOD 100) > 2 * CHARACTER[ BATI].CHARLEV
THEN
    BEGIN
        MVCURSOR( 1, 13);
        PRINTSTR( 'LOKTOFEIT FAILS!');
        EXIT( SLOKTOFE)
    END;
```

That's a 69% failure rate for a level 15 caster, and for the dear price you pay on a success, which is losing all your equipment and practically all your gold, this is just unacceptable. Wiz3 boosts the minimum success rate to 65 + Character Level%.

### #012 - Draining issues

Status: Fixed in WC027. Recompiled. Verified. Closed.

Answer: Monsters can drain your characters several times during combat. This is killing the fun factor in the game.

The fix: Added code for the monsters to not being able to drain your characters if they have already been drained. This includes a character drained by Haman or Mahaman. Although, casting successively Haman or Mahaman will drain your characters each time.

Why this fix: The draining effects are now working the same way they are in Wizardry 3 - Legacy of Llylgamyn.

Pre-implementation thoughts: Is this a coding issue or an inequitable balance issue against the player? It certainly does slow the player's progress down for sure (then again, there's that reset button). Consider this: If Palpatine zaps Luke Skywalker several times, Luke will get drained every time.

**Suggestion: leave it alone. Maybe this would be balance item for a Wizardry PG v4.0?).**

The issue: In Wizardry 3, you can only get drained once per combat, and casting HAMAN/MAHAMAN counts. In Wizardry 1, there is no limit.



#013 - Ninjas not living up its unarmed combat performance promise

Status: Fixed in WC030. Recompiled. Verified. Closed.

Answers: I think it is safe to say everyone has the same opinion on this one. Here is the Ninja definition in the Wizardry manual:

NINJA-are inhuman fighting machines. They can use any weapons or armor but work best without any! When fighting without protection with their bare hands, they can cause havoc and destruction, and may even kill the strongest opponent with a single blow! Their great training gives them a lower and lower ARMOR CLASS as they reach higher and higher levels of ability.

The fix: Added these code enhancements:

- The requirements to change class to ninja are now 15 in all stats instead of 17. Must still be of evil alignment.
- Base bare hands damage increased from 2d4 to 2d8.
- Unarmed Armor Class increased from 1 point every 3 levels, to 1 point every level.
- Unarmed Combat Initiative Bonus of 1 point (10%) for every 3 levels.

Preliminary suggestion: But how much more unarmed damage and natural armor could be given to the ninja without creating imbalance? Maybe this would be for a Wizardry balance project (Wizardry PG v4.0?). Let's remember that the goal of this project is to fix bugs, not change the game.

The issue: The manual says that Ninjas are better off without weapons and armor, but this is false. Ninjas do 2d4 per hit unarmed, which is better than the 2d2 of other classes, but finding better weapons than this is not hard! Critical hits work fine with weapons. Armor is also almost certainly better than going naked; you would need to be level 21 to match the effect of just wearing Evil Plate +3.

## #014 - Saving throws inconsistencies - LUCKSKILL

Status: Fixed in WC016. Recompiled. Verified. Closed.

Answer: Saving throws are based on your character's Luck attribute and a modifier. After a long analysis, everything gives except for saving vs the chests' Poison Gas Trap. It pointed to "3" (Breath) attack, changed it to "2". This makes use of category "2" since it did not make sense for the Poison Gas Trap and Breath being in the same category.

The Wizardry saving throw table is so similar to the 1979 AD&D table found here: <http://grogardia.blogspot.com/2020/12/saving-throws.html> that it is almost certain that this is where it comes from.

### SAVING THROW MATRICES

#### I. SAVING THROW MATRIX FOR CHARACTERS AND HUMAN TYPES

Character Class and Experience Level	Paralyzation, Poison or Death Magic	Attack to be Saved Against			
		Petrification or Polymorph*	Rod, Staff or Wand	Breath Weapon**	Spell**
Clerics*	1-3	10	13	14	15
	4-6	9	12	13	14
	7-9	7	10	11	12
	10-12	6	9	10	11
	13-15	5	8	9	10
	16-18	4	7	8	9
	19+	2	5	6	7
Fighters*	0	16	17	18	19
	1-2	14	15	16	17
	3-4	13	14	15	16
	5-6	11	12	13	14
	7-8	10	11	12	13
	9-10	8	9	10	11
	11-12	7	8	9	10
	13-14	5	6	7	8
	15-16	4	5	6	7
	17+	3	4	5	6
Magic-Users*	1-5	14	13	11	12
	6-10	13	11	9	10
	11-15	11	9	7	8
	16-20	10	7	5	6
	21+	8	5	3	4
Thieves*	1-4	13	12	14	15
	5-8	12	11	12	13
	9-12	11	10	10	11
	13-16	10	9	8	9
	17-20	9	8	6	7
	21+	8	7	4	5

There are the 5 saving throw checks used in the game:

- 0 - Poison & Paralysis & Critical Hit (Special monster skills).
- 1 - Stone.
- 2 - Was not used, moved the Poison Gas Chest Trap here.
- 3 - Breath. Poison Gas Chest Trap, made no sense, moved to 2.
- 4 - Silence. Anti-Mage/Anti-Priest Chest Traps.

Here are the compiled resistances, after changes, each point is 5%:

- [0] Vs. Poison & Paralysis & Critical Hit: Fighter 15%, Samurai 10%, Lord 10%, Ninja 15%, and if race is Human 5%.
- [1] Vs. Stoning: Priest 15%, Bishop 10%, Lord 10%, Ninja 10%, and if race is Gnome 10%.
- [2] Vs. Poison Gas Chest Trap: Bishop 10%, Ninja 20%, and if race is Elf 10%.
- [3] Vs. Breath Attacks: Thief 15%, Ninja 15%, and if race is Dwarf 20% to reduce the Breath damage by half.
- [4] Vs. Anti-Mage and Anti-Priest chest trap and Silence: Mage 15%, Bishop 10%, Samurai 10%, Ninja 10%, and if the race is Hobbit 15%.
- Vs. all of the above: add 5% for every 5 Levels of your character.
- Vs. all of the above: add 5% if your Luck is 6, 10% if your Luck is 12, and 15% if your Luck is 18.

LUCKSKIL analysis:

wiz  
---

```
PROGRAM WIZARDRY;  
    TCHAR = RECORD  
        LUCKSKIL : PACKED ARRAY[ 0..4] OF 0..31;
```

roller  
-----

```
PROCEDURE MAKECHAR; (* P010B09 *)
```

```
PROCEDURE INITCHAR; (* P010B0A *)
```

```
VAR  
    LSI      : INTEGER;  
    UNUSED  : INTEGER;
```

```
BEGIN  
    FILLCHAR( CHARREC, SIZEOF( CHARREC), 0);
```

```

CHARREC.NAME := CHARNAME;
CHARREC.AGE := (18 * 52) + (RANDOM MOD 300);
CHARREC.GOLD.LOW := 90 + (RANDOM MOD 100);
CHARREC.STATUS := OK;
FOR LSI := 0 TO 4 DO
  BEGIN
    CHARREC.LUCKSKIL[ LSI] := 16
  END;
CHARREC.MAXLEVAC := 1;
CHARREC.CHARLEV := 1;
CHARREC.ARMORCL := 10
END;

```

utilitie2

-----

```

PROCEDURE UPLCKSKL( LSSUB:    INTEGER;  (* P010123 *)
                   LSMODAMT: INTEGER);

  BEGIN
    LSMODAMT := CHARACTER[ CHARI].LUCKSKIL[ LSSUB] -
LSMODAMT;
    IF LSMODAMT < 1 THEN
      LSMODAMT := 1;
    CHARACTER[ CHARI].LUCKSKIL[ LSSUB] := LSMODAMT
  END;

PROCEDURE INITSTUF;  (* P010124 *)

  VAR
    X : INTEGER;
    Y : INTEGER;

  BEGIN
    WITH CHARACTER[ CHARI] DO
      BEGIN
        FOR X :=
          BEGIN
            WEPVSTY2[ 0][ X] := FALSE;
            WEPVSTY2[ 1][ X] := FALSE;
            WEPVSTYP[ X] := FALSE
          END;
        FOR Y := 0 TO 6 DO
          BEGIN
            WEPVSTY3[ 0][ Y] := FALSE;
            WEPVSTY3[ 1][ Y] := FALSE
          END;
        END;
      END;
    END;
  END;

```

```
        END
      END
    END;
```

```
6);
BEGIN (* EQUIPCHR *)
  WITH CHARACTER[ CHARI] DO
    BEGIN
      TEMPX := (20 - CHARLEV DIV 5) - (ATTRIB[ LUCK] DIV
      IF TEMPX < 1 THEN
        TEMPX := 1;
      FOR LUCKI := 0 TO 4 DO
        LUCKSKIL[ LUCKI] := TEMPX;

      CASE CLASS OF

        FIGHTER :   UPLCKSKL( 0, 3);
                   MAGE :   UPLCKSKL( 4, 3);
                   PRIEST :  UPLCKSKL( 1, 3);
                   THIEF :   UPLCKSKL( 3, 3);

        BISHOP : BEGIN
                   UPLCKSKL( 2, 2);
                   UPLCKSKL( 4, 2);
                   UPLCKSKL( 1, 2);
                   END;

        SAMURAI : BEGIN
                   UPLCKSKL( 0, 2);
                   UPLCKSKL( 4, 2);
                   END;

        LORD : BEGIN
                   UPLCKSKL( 0, 2);
                   UPLCKSKL( 1, 2);
                   END;

        NINJA : BEGIN
                   UPLCKSKL( 0, 3);
                   UPLCKSKL( 1, 2);
                   UPLCKSKL( 2, 4);
                   UPLCKSKL( 3, 3);
                   UPLCKSKL( 4, 2);
                   END;

      END;
    END;
  END;
```

```

CASE RACE OF
  HUMAN:  UPLCKSKL( 0, 1);
  ELF:    UPLCKSKL( 2, 2);
  DWARF:  UPLCKSKL( 3, 4);
  GNOME:  UPLCKSKL( 1, 2);
  HOBBIT: UPLCKSKL( 4, 3);
END;

```

rewards

-----

```

BEGIN (* ANTIPM *)
  FOR CHARPM := 0 TO PARTYCNT - 1 DO
    BEGIN
      PLYZSTON := (RANDOM MOD 20) < CHARACTER[
CHARPM].LUCKSKIL[ 4];

      CASE CHARACTER[ CHARPM].CLASS OF

        MAGE:  IF BMAGEDAM THEN
                IF PLYZSTON THEN
                  ISPLYZE
                ELSE
                  ISSTONED;

        SAMURAI:  IF BMAGEDAM THEN
                    IF NOT PLYZSTON THEN
                      ISPLYZE;

        PRIEST:  IF NOT BMAGEDAM THEN
                    IF PLYZSTON THEN
                      ISPLYZE
                    ELSE
                      ISSTONED;

        BISHOP:  IF NOT BMAGEDAM THEN
                    IF NOT PLYZSTON THEN (* IF NOT
SET... *)
                                ISPLYZE;

      END
    END
  END; (* ANTIPM *)

.
.
.

BEGIN (* DOTRAPDM *)

```

```

CLRRECT( 13, 8, 26, 2);
MVCURSOR( 13, 8);
IF TRAPTYPE <> 0 THEN
  BEGIN
    PRINTSTR( 'OOPPS! A ');
    PRTRAPTY( TRAPTYPE, TRAP3TYP)
  END
ELSE
  PRINTSTR( 'THE CHEST WAS NOT TRAPPED');
  PAUSE2;

CASE TRAPTYPE OF

  1:  (* POISON *)

        CHARACTER[ CHRXCST].LOSTXYL.POISNAMT[ 1] :=
        CHARACTER[ CHRXCST].LOSTXYL.POISNAMT[ 1] +
1;

  2:  (* GAS *)

        FOR CHARX := 0 TO PARTYCNT - 1 DO
          IF (RANDOM MOD 20) < CHARACTER[
CHARX].LUCKSKIL[ 3] THEN
            CHARACTER[ CHARX].LOSTXYL.POISNAMT[ 1] :=
1;

```

combat4

-----

```

PROCEDURE DOSILENC;  (* P01080A *)

VAR
  CHARX : INTEGER;

BEGIN
  FOR CHARX := 0 TO BATTLERC[ CASTGR].A.ALIVECNT - 1 DO
    IF CASTGR = 0 THEN
      ISISNOT( CASTGR,
        CHARX,
        100 - 5 * CHARACTER[ CHARX].LUCKSKIL[ 4],
        'SILENCED',
        1)
    ELSE
      ISISNOT( CASTGR,
        CHARX,

```

```

        10 * BATTLERC[ CASTGR].B.HPREC.LEVEL,
        'SILENCED',
        1)
END;

combat5
-----
PROCEDURE RESULT( ATTK0123: INTEGER; (* P01090B *)
                 STONFLAG: INTEGER;
                 POISSTON: INTEGER;
                 DAMSTR:  STRING);

VAR
    CHANCBAD : INTEGER;

BEGIN
    IF (RANDOM MOD 20) >
        CHARACTER[ MYVICTIM].LUCKSKIL[
STONFLAG] THEN
        EXIT( RESULT);
    IF ATTK0123 = 3 THEN
        BEGIN
            CHANCBAD := BATTLERC[ BATG].B.HPREC.LEVEL *
2;

            IF CHANCBAD > 50 THEN
                CHANCBAD := 50;
            IF (RANDOM MOD 100) > CHANCBAD THEN
                EXIT( RESULT)
            END;
        IF POISSTON > 0 THEN
            IF CHARACTER[ MYVICTIM].WEPVSTY3[ 1][ POISSTON]
THEN
                EXIT( RESULT);
            IF CHARACTER[ MYVICTIM].STATUS >= DEAD THEN
                EXIT( RESULT);
            CLRRECT( 1, 14, 38, 1);
            MVCURSOR( 1, 14);
            PRNAME( 0, MYVICTIM);
            PRINTSTR( 'IS ');
            PRINTSTR( DAMSTR );
            CASE ATTK0123 OF

                0:  IF BATTLERC[ 0].A.TEMP04[ MYVICTIM].STATUS
< STONED THEN
                    BATTLERC[ 0].A.TEMP04[ MYVICTIM].STATUS
:= STONED;

```



```

:= 1;
1:  CHARACTR[ MYVICTIM].LOSTXYL.POISNAMT[ 1]
< PLYZE THEN
2:  IF BATTLERC[ 0].A.TEMP04[ MYVICTIM].STATUS
    BATTLERC[ 0].A.TEMP04[ MYVICTIM].STATUS
:= PLYZE;
3:  BEGIN
    BATTLERC[ 0].A.TEMP04[ MYVICTIM].STATUS
:= DEAD;
    BATTLERC[ 0].A.TEMP04[ MYVICTIM].HPLEFT
:= 0
    END
END;
PAUSE1
END; (* RESULT *)

```

```

BEGIN (* CASEDAMG *)
  WITH BATTLERC[ BATG].B DO
    BEGIN
      IF SPPC[ 1] THEN
        RESULT( 1, 0, 3, 'POISONED');
      IF SPPC[ 2] THEN
        RESULT( 2, 0, 0, 'PARALYZED');
      IF SPPC[ 0] THEN
        RESULT( 0, 1, 5, 'STONED');

      IF DRAINAMT > 0 THEN
        DRAINLEV;

      IF SPPC[ 3] THEN
        RESULT( 3, 0, 0, 'CRITICALLY HIT')
    END
  END; (* CASEDAMG *)

```

combat5  
-----

```

PROCEDURE DOBREATH; (* P010906 *)

VAR
  UNUSED : INTEGER;
  HITDAM : INTEGER;

```



```
IF SPPC[ 2] THEN
    RESULT( 2, 0, 0, 'PARALYZED');
IF SPPC[ 0] THEN
    RESULT( 0, 1, 5, 'STONED');
```

The issue:

There are saving throw types, simply called LUCKSKIL in the code. These confer a variety of protective effects but are completely invisible to the player. LUCKSKIL[2], which elves get a -2 bonus on, seems to do absolutely nothing! I call this "save vs. wands" as the rest of them seem to correspond well to death, breath, petrify, and spell. Save vs. Death resists poison, paralysis, and critical hits in combat. But it has no benefit against corresponding trap types. Save vs. Petrify resists stoning hits in combat, but also has no benefit against corresponding trap types.

#015 - Bonus points on character creation are handled strangely

Status: Closed.

Answer: Going back to the roots of dice fantasy role-playing games this is normal when rolling characters, a bit harsh, but normal. As an example, in AD&D, what are the chances of rolling 18 Strength on 3d6 then 100 exceptional Strength on 1d100 for a score of 18/00. Strange but not a code defect.

The issue: There is a roughly 91% chance that the bonus points are 1d4+6, a roughly 8.5% chance of it being 1d4+16, and a roughly 0.5% chance of it being 1d3+26. I am not sure what the intended point distribution is, but this end result doesn't sound like it was arrived at on purpose.

#016 - Luck vs. teleport into rock

Status: Closed.

Answer: BREAKPOS means "Break Possession". Luck in this case has to do with the chance of an item not breaking when going to the cemetery when your entire party dies in the maze, and besides being dead all your party member's gold is divided by 2. Your entire party also dies and goes to the cemetery when teleporting into rock. Except, the game code specifically bypassed the check against breaking your items and losing half your gold when you teleport into rock, then you recover your "OUT" characters from the boot screen U)tilities to play your characters again. Essentially, the authors of Wizardry are giving you a break. It would be too cruel if the authors made you start all over if you teleported into rock by mistake.

The issue: Luck has some purpose for the event of teleporting into rock (see the function BREAKPOS but I cannot figure out what, or why it would matter.

#017 - AFRAID issue

Status: Closed. See #011 above for possible implementation.

Answer: It is just not used in the game. Could be used if building another scenario and giving a spell the capability to inflict this effect. Maybe in Wizardry PG Expansion 4.0, see #011 above.

The issue: An AFRAID status effect exists, but there is no way to inflict or be affected by this.

#018 - PARALYZE issue

Status: Closed. See "#010 - Manifo issue". It is implemented there.

Code exists for inflicting PARALYZE on monsters, but there is no way to actually do this.

#019 - To Hit issue

Status: Fixed in WC015. Recompiled. Verified. Closed.

Answer: True, monsters at the back were easier to hit. Victim is monster groups 1 to 4. Changed the code to:

+ ((3 \* VICTIM) - 6); Extra set of parentheses here to make sure nothing stupid happens and for clarity.

This way, the chance to hit the first monster group does not change, i.e., -3 (-15%), a minus number here is good, but decreases for each subsequent monster groups by 3 (15%).

The issue:

The to-hit chance against an enemy is calculated like this:

```
HPCALCPC := 21
            - BATTLERC[ VICTIM].B.AC
            - CHARACTER[ BATI].HPCALCMD
            + BATTLERC[ VICTIM].A.TEMP04[
SINGLEX].ARMORCL
            - 3 * VICTIM;
```

These line items correspond to:

- Monster's natural armor class

- Attacker's tohit bonuses (level, strength, weapon properties)
- Monster's AC penalties (dilto, morlis, mamorlis)

But I am not sure what VICTIM means. I think it corresponds to the monster's group number, but if so, it would mean that monsters in the front row are harder to hit than monsters in the back row, which makes no sense.

#### #020 - SWINGCOUNT issue

Status: Closed. Fixed in Wizardry Restored v2.2 26-JAN-23.

Answer: The swing count of these weapons should be all set from 1 to 0 in the Wizardry database. This is not a WIZARDRY.CODE coding issue.

The issue: There is absolutely no difference between 0 and 1; all classes have a base SWINGCOUNT value of 1, which does not stack with the weapon, but the larger SWINGCOUNT takes.

#### #021 - POISON issue

Status: Closed. May require further balance discussions.

Answer: All true. But if poison stacks then this can seriously diminish the chances of a character finishing a combat round before dying of poisoning or walking back to the castle. This could seriously affect the balance of the game, the lower the character level the worse the imbalance. Chest poison traps does stack though until reset by another poisoning event.

The issue: The engine contains logic to handle poison values above 1, but it is impossible for your poison value to ever exceed 1. All poison events set your poison level to 1 rather than increment it, except in the case of repeatedly opening chests that have the Poison Needle trap where it does stack properly, but it still goes back to 1 afterwards if poisoned by any other means.

#### #022 - Disbanding your party cures poison

Status: Closed. Further discussions can be done about this.

Answer: In a role-playing sense, the purpose of disbanding a party is for them to lick their wounds, take care of ailments and pray for a rescue. You could argue that by the time a rescue party arrives, the poisoning has been curbed. Besides, if you disband you party, its characters age by 25 weeks.

### #023 - Change class vs spell learning

Status: Closed. Further discussions can be had about this but see the answer below.

Answer: This is the eternal debate that will never come to an end of dual-class and changing class in dice and computer fantasy role-playing games. Some game masters would allow it (succumbing from the pressure by the players?) because arguably you are still at that level 10 mage proficiency even if you changed class to a fighter.

The issue: If you change class to a non-spellcaster, you can continue learning new spells in any circle where you already know at least one spell.

### #024 - Resting vs Age

Status: Closed. More discussions can be had about this.

Answer: True. But if resting at the Adventurer's Inn ages your characters, would your characters then age too quickly and thus make it even harder to gain stats and therefore even harder to make it to a Lord or Ninja class which are already hard enough? In Wizardry, the older your character, the less likely they will gain stats when leveling up. In dice role-playing games resting is short (regenerating your fatigue), if you have no magical means to cure wounds then it takes time (days, weeks, to heal), Wizardry reflects that in some way.

The issue: Resting does not age the characters.

### #025 - Resurrection vs Age

Status: Closed.

Answer: Exactly. Resurrection success chance is based on Vitality. Other role-playing systems may or may not include age as a factor. Wizardry just does not. This would mean changing the game, not fixing a bug.

The issue: Age does not directly affect the success rate of resurrection spells. Vitality does, though, both with Di/Kadorto and temple services.



## #026 - Highwaymen Issue

Status: Closed.

Answers: Good observation. But it is not a bug.

The issue: Highwaymen, who have monster id=6, only spawn in level 2, and only as companions to LVL 1 Mages. Apart from special encounters, this is the only monster in the game who will never spawn as a party leader.

## #027 - Anti-magic area issue

Status: Closed.

Answer: Further analysis shows that it may be possible to keep the party in a state where they cannot cast spells until they are back at the castle. This though can cause havoc in other made scenarios such as the DragonQuest scenario which uses anti-magic area as part of the "story". Also, if it was made to last the entire expedition, it would just be a game killer, and looking at it this way there is just no point. Keeping the current behavior though gives the scenario designer the possibility of creating "warded" areas where spells have no effects.

Early Answer: The game mechanics/database does not have that kind of multiple attributes, I mean stacking, for squares. This would require a hefty re-write of code and would actually change the game. Also, this would probably mean changing the structure of the database tables. **Going back dice role-playing game terms, one could have the very weak argument that the "fizzle wards" are replaced by "darkness wards" when getting in a dark area of the maze for example, but this is obviously not a very convincing explanation to justify this game behavior.**

The issue: Anti-magic fizzle effect wears off when you step on a square of any type other than NORMAL or ANTIMAGIC.

## #028 - Breath attack issue

Status: Closed. Could be a future consideration.

Answer: The poison breath could probably be coded to poison the party but that could potentially make some monsters too powerful and create imbalance. Maybe it could be implemented in an "4.0 expansion". Also, the code would probably have to have poison stack for this to work. Imagine if you get hit by a Poison Breath, then if you survive the encounter, you would lose 10, 20 hit points due to poisoning for every few steps you take until cured.

The issue: On breath attacks, an example is the gas dragon. Its breath property has value "3" which corresponds to poison. You would intuitively expect its breath attack to poison your party, but this can't happen. All it means is that if someone has poison protection (e.g. Mace Pro Poison) then the attack will do half damage.

### #029 - Alarm trap issue

Status: Closed. Tested.

Answer: Pretty sure this is the way it is supposed to work. The chest from the initial group of monsters because you attacked them in their lair. The alerted monsters came about in a state of "alertness", just grabbing their weapons and without carrying treasure since they showed up because of all the commotion your fight with the first monsters caused. So, the chest from the initial group is still there and opens after you popped the alarm.

The issue: If you set off an alarm trap, you will automatically get a chest reward from the next encounter, but the reward type is determined by the monsters you alerted, not the monsters you fought to get the chest.

### #030 - Monster resistances issues

Status: Closed.

Answer: Concerning monster resistances to poison and magic, maybe a game "4.0 expansion" with some spells having those effects can be done. Right now, it is just not used.

The issue: Let's take for example the werebear, who has resistance to cold and poison. The cold resistance will halve the damage of Dalto and Madalto. But the poison resistance does absolutely nothing, as no spells in Wizardry inflict poison damage. Only fire, physical, and cold resistances will do anything for the monsters (even magic resistance does nothing).

### #031 - Maelific issue

Status: Closed. Fixed in Wizardry Restored v2.2 26-JAN-23.

Answer: This would be an adjustment in the database record of the monster, this is not a WIZARDRY.CODE coding issue.

The issue: The Maelific has a group size of 1d0+1. I'm pretty sure it was supposed to be entered as 1d1, but the end result is the same.

### #032a - Extra treasure tables

Status: Closed.

Answer: They are just not used. Extras for another scenario.

The issue: Treasure tables 22 & 23 are present, but are not used.

### #032b - Shuriken issue

Status: Closed.

Answer: True. The code is

```
CRITHITM := CRITHITM OR OBJECT.CRITHITM;
```

If you would give the Ninja twice a shot at critically hitting the monster every round, then the Ninja would be way too powerful, thus creating imbalance. So, we have here a redundancy of the critical hit effect and although changing that weapon's properties in the Wizardry database could be done, it makes no difference. In another scenario, a weapon could be given the critical hit property on an item that can be used by a non-ninja character.

The issue: The Shuriken confers the critical hit ability! Too bad only ninjas can use it, making this pointless.

### #033 - Dagger of Speed issue

Status: Closed.

Answer: Working as expected. Reflective of actual real human combat where a wild attack more than often means you have opened up your defenses.

The issue: The Dagger of Speed has an armor malus. I am unsure if this is a mistake or not.

#034 - The monsters' friends/no friends attribute does not work at all

Status: Fixed in WC005. Recompiled. Verified. Closed. Also, WERDNA's resistance to befriending has been turned on in Wizardry Restored v2.2 26-JAN-23.

Answer: Fixed by adding a line of code to the FRIENDLY procedure.

The issue: 'A FRIENDLY GROUP OF'. In a monster record, there is an attribute to resist befriending, meaning you cannot have a friendly encounter with them if set to on (or true). The issue was that even if that attribute was on, you could still have a friendly encounter with them. This means you might not get Werdna's amulet if he befriends you when you meet him for example and decide to not fight him. Of course, Werdna's resistance to befriending would have to be set to on so that he never be friends with your characters.

#035 - Inventory full issue

Status: Fixed in WC033. Recompiled. Verified. Closed.

The issue: Before the fix, a random character was selected to receive a treasure item. If that character's inventory was full, then the item was lost. This meant you might not have gotten Werdna's amulet upon defeating him for example.

The fix: Look at the entire party's inventory for space to receive an item. Display a message when the entire party's inventory is full.

Why this fix: To avoid having to continuously micro manage the party's inventory to ensure that there is enough space for treasure drops.

Preliminary Answer: The short answer: Make sure you have inventory room on all your characters when meeting Werdna or opening lucrative deep levels chests. The long answer: There are less than 512 bytes of free space on disk for more code, this may require a few precious coding lines. Maybe something for Wizardry PG Expansion 4.0.

The issue: If the character receiving an item from a treasure chest has a full inventory, they do not get it. This means you might not get Werdna's amulet upon defeating him for example.

#036 - Treasure tables range bug.

Status: Fixed in WC022. Recompiled. Verified. Closed.

Also, all database updates brought forward in WC024 - Interim Release 2023-02-09 including Treasure Tables 18 and 19 updates.

Answer: Adjust to code and database tables for the monsters' treasure to give the intended treasures. The monsters were not giving the intended range of treasures they were set to across all the treasure tables. The most blatant example: the monsters are not supposed to drop a Deadly Ring which is a one-time event in the game. The fix also takes care of a division by zero problem, i-e,  $x \text{ MOD } 0$ .

Reported by: DDG Ahab. Contribution Eric Labelle, Denis Molony.

Other References: Thomas W. Ewers at Jul 11, 2014, 3:59:55 AM on <https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

The issue: In short, the lower range of the treasure tables is off by two, and the upper range off by one.

#037 - The Halito vs Sopic bug

Status: Closed.

Answer: Analyzing all the monsters' spell casting possibilities, one could say the same for Lahalito and Madalto. Essentially, it seems clear the authors wanted to give the monsters offensive spells, thus Halito at level 2. If the monsters would have Sopic at that level, it could potentially make it extremely hard for low level characters to hit the monsters, thus creating imbalance. Same for Priest spell, at level 7 the monsters cast Mabadi but should it not be Malikto? The authors decided to go with Mabadi. So, no code fixing is needed since nothing is broken. Here more of the code:

CASE SPELLLEV OF

```
1:  IF TWOTHIRD THEN
      SPELLCAS := KATINO
    ELSE
      SPELLCAS := HALITO;

2:  IF TWOTHIRD THEN
      SPELLCAS := DILTO
    ELSE
      SPELLCAS := HALITO;  (* BUG *)

3:  IF TWOTHIRD THEN
      SPELLCAS := MOLITO
    ELSE
      SPELLCAS := MAHALITO;

4:  IF TWOTHIRD THEN
      SPELLCAS := DALTO
    ELSE
      SPELLCAS := LAHALITO;  (* ...HMMM *)

5:  IF TWOTHIRD THEN
      SPELLCAS := LAHALITO  (* ...HMMM *)
    ELSE
      SPELLCAS := MADALTO;

6:  IF TWOTHIRD THEN
      SPELLCAS := MADALTO  (* ...HMMM *)
    ELSE
      SPELLCAS := ZILWAN;
```

```
7: SPELLCAS := TILTOWAI;
END;
```

```
CASE SPELLEV OF
```

```
1: SPELLCAS := BADIOS;
2: SPELLCAS := MONTINO;
3: IF TWOTHIRD THEN
    SPELLCAS := BADIOS
    ELSE
    SPELLCAS := BADIAL;
4: SPELLCAS := BADIAL;
5: IF TWOTHIRD THEN
    SPELLCAS := BADIALMA
    ELSE
    SPELLCAS := BADI;
6: IF TWOTHIRD THEN
    SPELLCAS := LORTO
    ELSE
    SPELLCAS := MABADI;
7: SPELLCAS := MABADI;
END;
```

The bug:

HALITO should be SOPIC

```
4555 12 24:1 67          2: IF TWOTHIRD THEN
4556 12 24:3 70          SPELLCAS := DILTO
4557 12 24:2 70          ELSE
4558 12 24:3 77          SPELLCAS := HALITO;
(* BUG *)
4559 12 24:3 84
```



## #037 - Lostxyl error - aka Roster "Out" Characters

Status: Fixed in WC032. Recompiled. Verified. Closed.

Answer #1: "INMAZE" indeed indicates if the character is in the party or can be R)ecovered using the boot side U)tilities, the later happens if the game crashes or you press reset or teleport into Rock.

Answer #2: This was a bug. In the Training Grounds, the Roster of players does not always display characters that are "Out" in the dungeon. Note that characters that are "Out" cannot be added to the party at Gilgamesh's Tavern. Now there is an indication as to why a character cannot be added to the party.

The fix: Updated the code to display all characters that are "Out".

Code to:

```
          (* WC032 FROM -LOCATION [ 1]- TO -LOCATION [ 3]-
*)
          IF CHARREC.INMAZE OR (CHARREC.LOSTXYL.LOCATION[
3] <> 0) THEN
          WRITE( ' OUT');
```

Following the email that led to this fix:

From: Denis Molony <dmolony@iinet.net.au>

Sent: Tuesday, December 13, 2022 10:29 PM

To: Eric Labelle <eric@zimlab.com>

Subject: inmaze

This is a list of references to the INMAZE attribute. I think it indicates whether a character is in a party, not necessarily in the maze.

It also looks like a bug in DSP20NM where it compares LOSTXYL[1] <> 0. I think that should be the level coordinate ([3]), not the x axis.

Teleport:

```
BEGIN
  WRITELN( 'YOU LANDED IN SOLID ROCK OUTSIDE THE');
  WRITELN( 'DUNGEON - YOU ARE LOST FOREVER!');
  FOR X := 0 TO PARTYCNT - 1 DO
    BEGIN
      CHARACTER[ X].INMAZE := FALSE;
      CHARACTER[ X].STATUS := LOST
    END;
  XGOTO := XCEMETRY;
  EXIT( UTILITIE)
END;
```

Temple of Cant:

```
IF ((WHO.LOSTXYL.LOCATION[ 1] +
    WHO.LOSTXYL.LOCATION[ 2] +
    WHO.LOSTXYL.LOCATION[ 3] ) <> 0)
  OR
  WHO.INMAZE THEN
  DSP2STR( WHO.NAME, ' IS NOT HERE');
```

More Cant:

```
BEGIN
  IF WHO.STATUS = DEAD THEN
    WHO.STATUS := ASHES
  ELSE
    WHO.STATUS := LOST;
  WHO.INMAZE := FALSE;
  MOVELEFT( WHO,
    IOCACHE[ GETRECW( ZCHAR, WHOHELP,
  SIZEOF( TCHAR))],
    SIZEOF( TCHAR));
  WRITELN;
  IF WHO.STATUS = LOST THEN
    DSP2STR( WHO.NAME, ' WILL BE BURIED')
  ELSE
    DSP2STR( WHO.NAME, ' NEEDS KADORTO NOW')
END; (* ASHLOST *)
```

Cemetary:

```
IF CHARACTER[ LLBASE04].STATUS <> LOST THEN
  BEGIN
    WITH CHARACTER[ LLBASE04] DO
```



```
IF PICKCNT = 0 THEN
    WRITELN( '** NO ONE **')
END; (* LOOKLOST *)
```

PICKUP:

```
CHARDISK[ PARTYCNT ] := PICKLIST[ PICKCHAR ];
CHARACTR[ PARTYCNT ].LOSTXYL.LOCATION[ 1 ] := 0;
CHARACTR[ PARTYCNT ].LOSTXYL.LOCATION[ 2 ] := 0;
CHARACTR[ PARTYCNT ].LOSTXYL.LOCATION[ 3 ] := 0;
CHARACTR[ PARTYCNT ].INMAZE := TRUE;
```

ADDPARTY:

```
IF CHARI = SCNTOC.RECPERDK[ ZCHAR ] THEN
    EXITADDP( '** WHO? **')
ELSE
    IF CHARACTR[ PARTYCNT ].INMAZE OR
        (CHARACTR[ PARTYCNT ].LOSTXYL.LOCATION[ 3 ] <> 0)
    THEN
        EXITADDP( '** OUT **')
    ELSE
        IF (PRTYALGN <> NEUTRAL) THEN
            IF (CHARACTR[ PARTYCNT ].ALIGN <> NEUTRAL) THEN
                IF (PRTYALGN <> CHARACTR[ PARTYCNT ].ALIGN)
                THEN
                    EXITADDP( '** BAD ALIGNMENT **');
                GOTOXY( 0, 20);
                WRITE( 'ENTER PASSWORD >');
                GETPASS( CHARNAME);
                GOTOXY( 0, 21);
                IF CHARNAME <> CHARACTR[ PARTYCNT ].PASSWORD THEN
                    EXITADDP( '** THATS NOT IT **');
                CHARDISK[ PARTYCNT ] := CHARI;
                CHARACTR[ PARTYCNT ].INMAZE := TRUE;
```

REMOVE:

```
BEGIN
    CHARI := GETCHARX( FALSE, 'WHO WILL LEAVE');
    IF (CHARI < 0) OR (CHARI = PARTYCNT) THEN
        EXIT( REMOVE);
    CHARACTR[ CHARI ].INMAZE := FALSE;
```

DSP20NM:

```
IF CHARREC.INMAZE OR (CHARREC.LOSTXYL.LOCATION[
1] <> 0) THEN
    WRITE( ' OUT');
```

LOSECHAR:

```
BEGIN
    CHARREC.STATUS := LOST;
    CHARREC.INMAZE := FALSE;
    PUTCHARC( CHARREC, CHARACX);
    GTSCNTOC
END;
```

DISBAND:

```
INMAZE := FALSE;
LOSTXYL.LOCATION[ 1] := MAZEX;
LOSTXYL.LOCATION[ 2] := MAZEY;
LOSTXYL.LOCATION[ 3] := MAZELEV;
AGE := AGE + 25;
```

#038 - Mabadi issue

Status: Fixed in WC013. Recompiled. Verified.

Answer: Added code to skip that message if the monster is dead. Also, removed extra space at the beginning of " IS HIT...".

The issue: Monsters " IS HIT BY MABADI!" message is displayed although no monsters are left alive.

#039 - Password issue

Status: Enhancement WC019. Recompiled. Verified.

Answer: The main reason for this is that removing this from the game saves a good chunk of coding space that can be used for enhancements instead. Having to enter passwords for characters is superfluous, especially nowadays.

The issue: Recuperate coding space. Remove the prompts for a password when adding characters at Gilgamesh's Tavern and at the Training Grounds.

#040 - Using an item still shows its non-decayed name

Status: Fixed in WC017. Recompiled. Verified.

Answer: Add a line of code to refresh the items list after an item has been used.

The issue: When in Camp, if using an item and it decays (Ex: Dios Potion > Broken Item), the display still shows the undecayed item name until refreshed.

#041 - Misspelling of Dispel, Morgue, Group #

Status: Fixed in WC002. Recompiled. Verified.

#042 - Gilgamesh misspelling

Status: Fixed in WC004. Recompiled. Verified.

In the Castle Market, change "Gilgamesh' Tavern" to "Gilgamesh's Tavern".

```
+-----+
! CASTLE                                MARKET !
+-----+ CURRENT PARTY: +-----+
# CHARACTER NAME  CLASS AC HITS STATUS

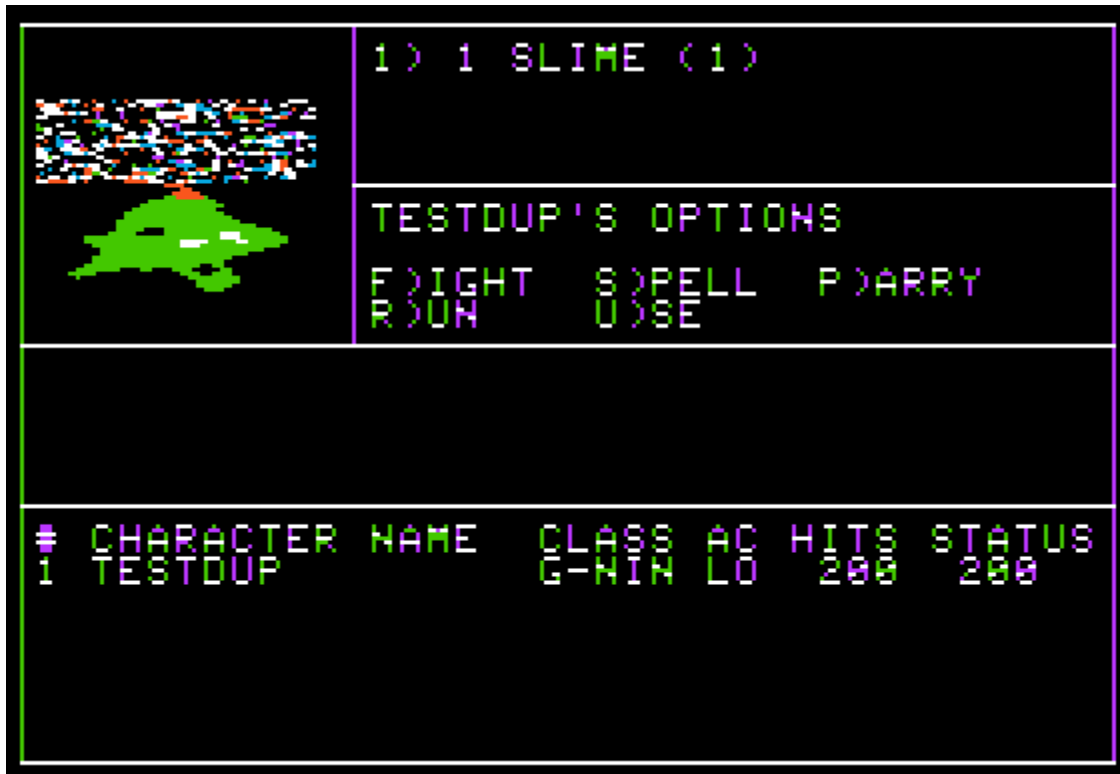
+-----+

                YOU MAY GO TO:
THE A)DVENTURER'S INN, G)ILGAMESH'S
TAVERN, B)OLTAC'S TRADING POST, THE
TEMPLE OF C)ANT, OR THE E)DGE OF TOWN.
```

#043 - Bug1 reported by Thomas W. Ewers

Status: Fixed in WC040. Recompiled. Verified.

The issue: On the boot side of Wizardry, there is a Utility to "Make Scenario Disk". An off-by-one error in the code had the top half part of the first monster picture overwritten with junk when making the scenario disk.



The fix: Corrected the off-by-one error.

Reported by: Thomas W. Ewers

Ref: Jun 27, 2014, 5:40:39 PM in

<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

Bug1:

-----

Last weekend I finished retrieving the hi-res "pictures" from SCENARIO.DATA for the monsters and enemies you see in the game. These are displayed in the upper left corner of the screen while playing. I then went to the first level of the game and encountered one of the earliest "bad guys": the BUBBLY SLIME. Unfortunately, the hi-res part of the screen was somewhat



mangled where the picture of the SLIME was partially displayed.

I thought, "How could my disk image have gotten corrupted?" Did I insert this diskette at the wrong time when doing something else? Did I mix and match my Scenario version with the other Boot version in the game? Was my diskette not formatted correctly to start with? Was there a bug in the software trying to display the image? Eventually I looked at the SCENARIO.DATA file on the SCENARIO side of the disk I started with. On that diskette the image was intact and correct, but on the "DUPLICATE/SCENARIO" disk I had made, the first block (1 Pascal block = 2 DOS sectors) were clearly just garbage.

I then set out to find the bug, I started from scratch and initialized a diskette and then used the Wizardry "Utilities" to make my scenario diskette.

Sidetrack: Versions 2 and 3 of Wizardry clearly state you must create your own DUPLICATE/SCENARIO diskette to play the game. The instructions for Wizardry I are not very clear and make it sound like you should just use the "MASTER/SCENARIO" to play the game. Since playing the game has some destructive effects on the data on your scenario diskette, if you go that route and solve the game you will never be able to start over and play the entire game. Good for sales, bad for players (and their brothers and sisters).

Ok, so all I did was create the DUPLICATE/SCENARIO and used my DUMPDATA program to extract the image from SCENARIO.DATA and display the image on the Apple screen. And it was already corrupt! I did not need to even start playing the game!

I realized the "make a DUPLICATE/SCENARIO" code must be broken.

In WIZBOOT (that's the name for WizUtil.code), PROCEDURE MAKESCEN() has a bug:

```
CHARREC0.STATUS := LOST;
FOR TEMPI := 0 TO SCNTOC.RECPERDK[ ZCHAR] DO
WRCHARAC( CHARREC0, TEMPI);
```

This code occurs right after 35 tracks of data have been copied from the MASTER/SCENARIO to your DUPLICATE/SCENARIO. It is clearing out any CHARACTRs that might have been on the

MASTER/SCENARIO disk.

RECPERDK is "RECORDs of this type PER DisKette". Note TEMPI starts at 0. There is room on the diskette for 20 CHARACTR (ZCHAR) characters, and RECPERDK[ ZCHAR] is 20. Yup! Another off-by-one error. It is writing 1 block too many and clobbering the first hi-res picture image.

Another interesting twist to this bug. If you use only the MASTER/SCENARIO and encounter SLIME (and there are a few more creatures that use this same picture) you see the actual image the way it should be. It is only when you make your DUPLICATE/SCENARIO that you see this error.

low:

**#044 - Bug2 reported by Thomas W. Ewers**

Status: Pending.

Note by Snafaru: Below it seems every mention of 005FB should be 005FA?

By: Thomas W. Ewers

Ref: Jun 27, 2014, 5:40:39 PM in

<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

Bug2:

-----

I have this comment in MVCURSOR() code:

```
; MVCURSOR( 80, Y) ADJUST RANDOM # (UNTIL KEY IS AVAILABLE)
```

In GETKEY() in the main WIZARDRY() program is this:

```
MVCURSOR( 80, 0); (* ADJUST RANDOM #, AND RETURN WHEN A CHAR IS  
AVAIL *)
```

This is a way for the program to help "randomize" the seed for the next time the RANDOM number generator is called.

(As mentioned in an earlier post, there's a bit of strange code associated with MVCURSOR. Sometimes it doesn't do "cursor" stuff.)

Here's the code in MVCURSOR( 80, 0):

```
L5931  
LDX #00  
L5933  
INC 0047A,X ; INCREMENT RANDOM #  
BNE L593D ;  
INX ;  
CPX #04 ;  
BNE L5933 ;  
L593D  
LDA 0C000 ; KEYBOARD DATA
```

Here is a partial listing for the table in the APPLE II Reference manual, p. 134:

Table 26: I/O Scratchpad RAM Addresses

Base Address	1	2	3	4	5	6	7
\$0478	\$0479	\$047A	\$047B	\$047C	\$047D	\$047E	\$047F
\$04F8	\$04F9	\$04FA	...				
\$0578	\$0579	\$057A	...				
\$05F8	\$05F9	\$05FA	...				
\$0678	\$0679	\$067A	...				
\$06F8	\$06F9	\$06FA	...				
\$0778	\$0779	\$077A	...				
\$07F8	\$07F9	\$07FA	...				

Here's some code from the RNG:

```
LDY #07 ; SHIFT SOME BITS 7 TIMES
L5CCC
ASL 0047A
PHP
ROL 004FA
ROL 0057A
ROL 005FB
BMI L5CE0
```

It's not the most kosher thing in the world to just use these slot memory locations for your own purposes!

But there is something even worse going on here. This code in MVCURSOR wants to increment the first 4 memory locations listed under slot #2 as found in the table on p 134. In reality it is incrementing the first memory location for slot#2, slot #3, slot #4, slot #5.

Yikes!

Not to mention that only 1 of the 4 bytes used in the Random Number Generator are being altered by this code. Therefore, the RNG they thought they were implementing is not what they wrote! I'm not sure how this affects the numbers generated but it is clearly not a good thing. Maybe altering one byte while using keyboard input makes this RNG "good enough", but further testing might prove otherwise. One other thing, did you notice that the RNG is accessing the first 3 bytes of slot#2 and 1 byte of slot#3?

#045 - Bug3 reported by Thomas W. Ewers

Status: Closed.

Answer: 2 Jan 2023 by Eric L. Testing shows everything seems to work normally. Although the Evil Helm +2 is not a cursed item, it will become cursed on a Good character that equips it. Thereafter it cannot be sold at Boltac's by that character but, if the character has not equipped it, it can be sold as it has not turned into a cursed state. Note: tested also the Neut P-Mail + 2 with the same results.

By: Thomas W. Ewers

Ref: Jun 27, 2014, 5:40:39 PM in  
<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

Bug3:

-----

While playing the game the other day (oops, I meant to write "testing the game"), I had a BISHOP identify an object. No, this is not the famous "identify item #9" bug. I noticed the object had a "-1" in the name. As I found by examining SCENARIO.DATA, the CURSED field for OBJECTS is set for every item that has a negative number in the name. In this instance I did not need to EQUIP the item. My BISHOP took the item to BOLTACs trading post and was able to sell it. But there is code that usually produces this message when you try to sell a cursed item:

```
*** WE DON'T BUY CURSED ITEMS ***
```

The code in IDITEM() has a bug where the "CURSED" field is not always set after an item has been identified by a BISHOP.

Although they fixed another bug in this code for WizardyIII (LOL), I think this bug is there also.

#### #046 - Object special number 23

Status: Fixed in WC007. Recompiled. Verified. Closed.

Answer: Altered code as suggested by Thomas, test.

By: Thomas W. Ewers comment in UTILITIE2.TEXT

Code:

```
23: BEGIN
    (* LOOKS LIKE BUG!  PARTYCNT - 1  !!! *)
    FOR SPCTEMP := 0 TO PARTYCNT DO
        CHARACTER[ SPCTEMP].HPLEFT :=
            CHARACTER[ SPCTEMP].HPMAX
END;
```

#### #047 - Hidden CTRL-G Bell

Status. Fixed in WC008. Recompiled. Verified.

Answer: To avoid future problems when editing or viewing COMBAT5.TEXT. To make it clear what is going on in the code.

The issue:

SOMETHING TO WATCH FOR TO NOT LOSE THE "BEL" WHEN EDITING CODE:

The COMBAT5.TEXT contains three hidden CTRL-G ascii characters to ring the BELL 3 times for when your character scores a critical hit. Key word here: hidden.

Reference:

Posts: May 22, 2014, 12:42:15 PM and  
May 23, 2014, 11:54:02 AM on:  
<https://groups.google.com/g/comp.sys.apple2/c/aI5ob1mLUwY/m/CIqEeVUvDAAJ>

The R)place technique by Tommy works.

The WRITE( CHR(7), CHR(7), CHR(7)); technique would probably work too since it works in a short program. See the proof of concept for three ways to ring the bell in Wizardry here:

<https://github.com/snafaru/Hello>

#048 - 80-Column on Apple IIe

Status. Fixed in ST004. Recompiled. Verified.

Answer: Implement Chris Torrence's solution.

The issue: When playing on an Apple IIe, Wizardry text mode is displayed in 80 Columns which gives a weird spacing between all characters and lines on the screen.

#049 - When entire party gets slaughtered, there's a roll vs your Luck skill to see if your items break, but it is not working

Status: Closed. See #016 - Luck vs. teleport into rock.

Answer: Tested, the BREAKPOS (Break Possessions) works when your party get slaughtered in the maze. But the programmers decided to skip it if you teleported into rock, giving you a break.

Missing **CHARACTR[ LLBASE04]. ????**

```
PROCEDURE BADSTUFF; (* P010218 *)

PROCEDURE BREAKPOS; (* P010219 *)
VAR
    X      : INTEGER;
    POSSX  : INTEGER;

BEGIN
    WITH CHARACTR[ LLBASE04] DO
        BEGIN
            FOR POSSX := 1 TO CHARACTR[
LLBASE04].POSS.POSSCNT DO
                IF NOT CHARACTR[ LLBASE04].POSS.POSSESS[
POSSX].CURSED THEN
                    IF (RANDOM MOD 21 > ATTRIB[ LUCK]) THEN
                        CHARACTR[ LLBASE04].POSS.POSSESS[
POSSX].EQINDEX := 0;
                        X := 0;
                        FOR POSSX := 1 TO CHARACTR[
LLBASE04].POSS.POSSCNT DO
                            IF CHARACTR[ LLBASE04].POSS.POSSESS[
POSSX].EQINDEX <> 0 THEN
                                BEGIN
                                    X := X + 1;
                                    CHARACTR[ LLBASE04].POSS.POSSESS[ X] :=
CHARACTR[ LLBASE04].POSS.POSSESS[ POSSX]
```





#050 - Mamorlis issue

Status: Fixed in WC012. Recompiled. Verified.

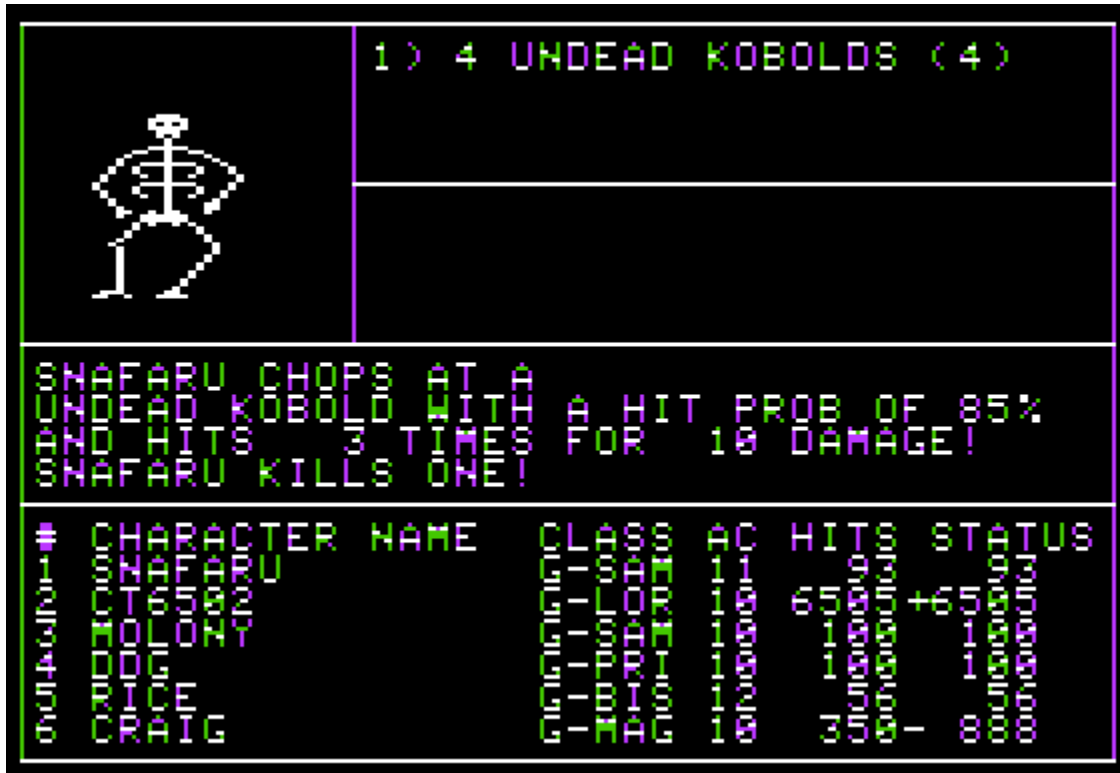
Answer: Fixed code to include the 1<sup>st</sup> monster in each group.

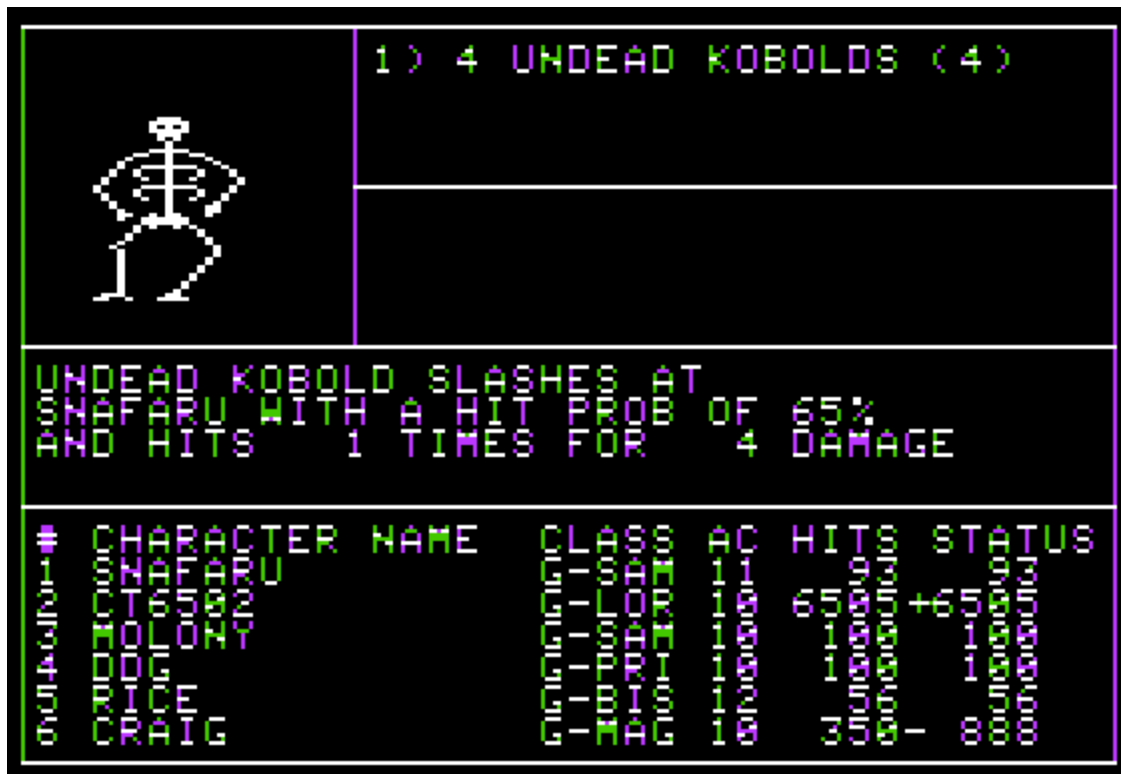
The issue: Mamorlis is supposed to fear all the monsters (reduce their armor class by 3) but the 1<sup>st</sup> monster of each group is not affected.

#051 - Display Hit Probability

Status: Enhancement WC018. Recompiled. Verified.

The issue: Enhance the battle details display by showing the hit probability of characters and monsters on-the-fly.





#052 - Display the Spells the Monsters are Casting

Status: Enhancement WC020. Recompiled. Verified.

Answer: Added code to show what spells are cast by both the monsters and characters in the combat window.

The issue: Enhancement suggested by DDG Ahab. It would be nice to see what spells the monsters are casting.



#053 - Monsters cast Zilwan

Status: Status: Fixed in WC012. Recompiled. Verified.

Answer: Changed the monsters' casting of ZILWAN for MADALTO instead. Madalto was the only spell that made sense. Other high level spells like Makanito, Mamorlis, Lakanito, Masopic do nothing at all or nothing useful for the monsters.

The issue: Some monsters cast the spell ZILWAN, which has absolutely no effects on the player characters.

#054 - Misspelling Bad Amount

Status: Fixed in WC026. Recompiled. Verified.

The issue: When trading gold between two characters, and the amount entered is invalid you get a message. The contraction of a word in that message is slightly annoying and unnecessary.

The fix: Changed the message from "Bad Amt" to "Bad Amount".

Why this fix: Clarity. To remove the distraction caused by the contraction of a word.

#055 - Several words are contracted and missing the apostrophe

Status: Fixed in WC029. Recompiled. Verified.

Answer: For consistency.

#056 - Mapiro Mahama Diromat

Status: Fixed in WC031. Recompiled. Verified.

Answer: Added a delay after the incantation before going back to the castle. This gives the player a chance to read the message.

The issue: On level 1, after coming back from the elevators, or from farming the Murphy's Ghosts, there is a shortcut back to the castle. The shortcut happens when you enter a certain room, and the mage therein sends you back to the castle after an incantation. The problem is that part of the message appears and disappears from the screen so fast that you cannot read it.

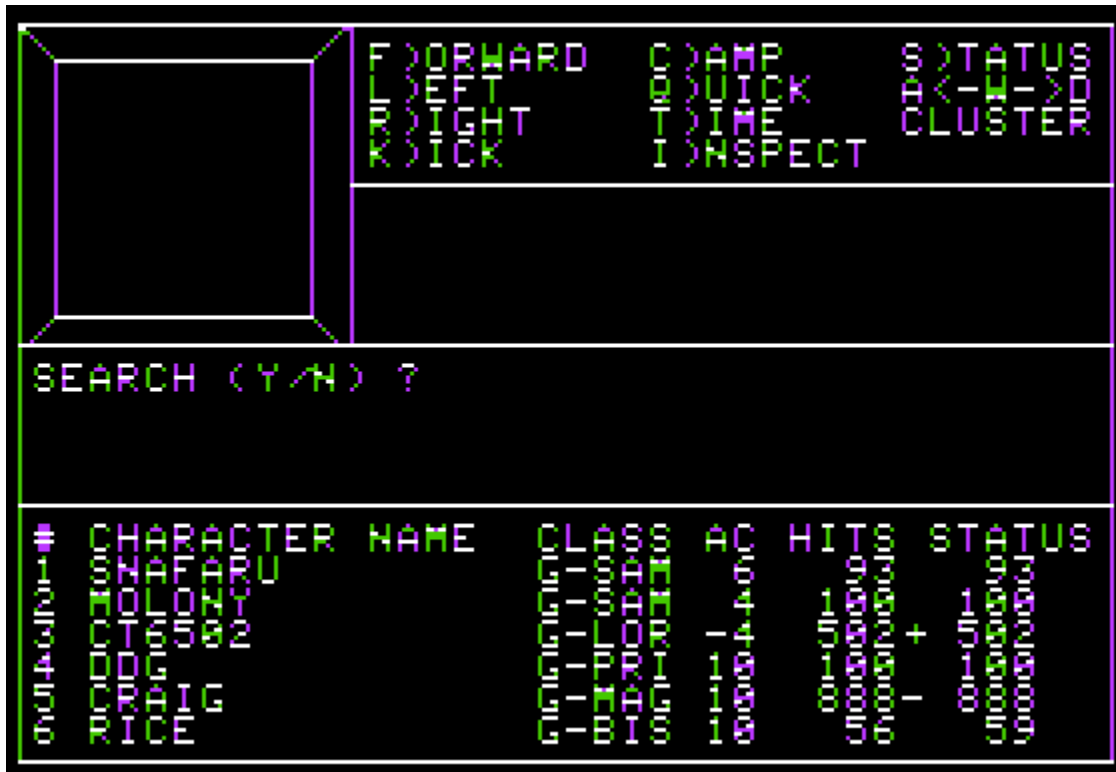
#057 - Search for Item

Status: Fixed in WC031. Recompiled. Verified.

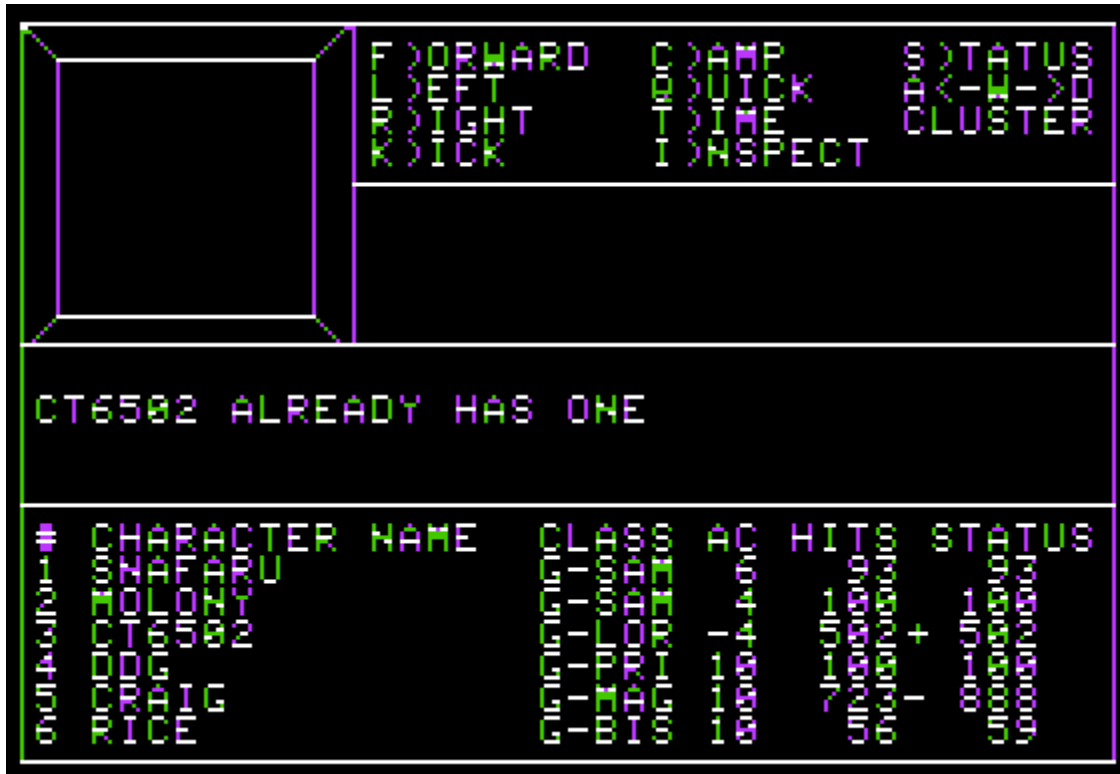
The issue: In the maze, the party enters areas where Wizardry tells a story and then asks you if you want to "Search (Y/N)?". If the party's inventory is full then nothing happens and that leaves the player wondering what is going on.

The fix: Added messages to inform the player when the characters' inventories are full as well as when the characters already have the item they are supposed to get.

Why this fix: To inform the player as to what is going on when responding yes to "Search (Y/N)?".







#058 - "Resurrects" misspelled

Status: Fixed in WC035. Recompiled. Verified.

Answer: Fixed.

The issue: When casting Haman or Mahaman, the word "Resurrects" is misspelled "Ressurrects".

#059 - Run the game in ProDOS

Status: Fixed in WC036. Recompiled. Verified.

Answer: Added code to Q)uit the game. Peter Ferrie built a ProDOS "shell" to run Wizardry in ProDOS.

The challenge: Run Wizardry in ProDOS.

### #060 - Disarm Trap Bug

Status: Fixed in WC037. Recompiled. Verified.

Contributors: Eric Labelle. Bug reported by Reiska42.

The issue: This happens when finding a chest that is booby trapped with a trap of type 3 (Crossbow Bolt, Exploding Box, Splinters, Blades, and Stunner). Upon disarming the trap if you enter any valid trap name besides the current trap then nothing happens and you go back to the chest options. You can continue entering all the trap names and nothing will happen. The only way out is to open the chest, leave it, or enter the right trap name.

The fix: Added the same code as in Wizardry III Legacy of Llylgamyn to fix this bug.

Why this fix: To make disarming chest traps work as intended.

### #061 - Camp Screen Update

Status: Fixed in WC038. Recompiled. Verified.

Contributors: Eric Labelle. Enhancement suggested by Reiska42.

The suggestion: Update the gold, hit points and status of a character when performing actions such as trading gold, or casting spells to cure hit points or ailments.

The action: Added the necessary code to allow these enhancements.

Why this: It is nice now to see the screen updated on the fly when performing these actions.



## #062 - Fix Maze Text Entry Overflow Crash

Status: Fixed in WC039. Recompiled. Verified.

Contributors: Eric Labelle.

Bug reported by: Reiska42.

The issue: In the maze, when entering text for a spell name, chest trap name, time delay, or answer to a riddle, if you continue typing characters the text will overflow to next lines and eventually the game will crash.

The fix: Restrict the length of the entry to the minimum necessary to prevent the game from crashing.

Why this fix: To prevent the game from crashing. But did not make it so short as to prevent answering a fairly long riddle of up to 22 characters. One can still corrupt the screen by overtyping but at least the game will not crash anymore.

## #063 - Pool and Divide Gold and More

Status: Fixed in WC040. Recompiled. Verified.

Contributors: Eric Labelle.

With suggestions from: Marshall Bockrath, Todd Borax, Chris Torrence

Enhancements: Added the possibility for a character to "Pool the Gold" of the entire party in his pocket. This can be done in Gilgamesh's, Boltac's, Adventurer's Inn and Camp. Furthermore, you can also "Divide the Gold" between all characters whilst inspecting your character at Gilgamesh's or Camp.

Enhancements: Updated the code to allow characters to be transferred between scenario disks even if they are carrying quest items. Note This was reverted in WC041 on 5 Feb 2024.

Fix: Also see WC043 above. On the boot side of Wizardry, there is a Utility to "Make Scenario Disk". An off-by-one error

in the code had the top half part of the first monster picture overwritten with junk when making the scenario disk.

Consistency: The Evil Shield +3 armor class has been improved from 5 to 6 to be consistent with the armor protection of all the other shields.

At Gilgamesh's Tavern:

```
NIN HUMAN E-NINJA
STRENGTH 13      GOLD      2886
      I.Q. 14      EXP      147427
      PIETY 15
VITALITY 15      LEVEL 10      AGE 25
AGILITY 16      HITS 71/ 71      AC 0
      LUCK 13      STATUS OK
      MAGE 0/0/0/0/0/0/0
      PRIEST 0/0/0/0/0/0/0
*=EQUIP, -=CURSED, ?=UNKNOWN, #=UNUSABLE

YOU MAY E)QUIP, D)ROP AN ITEM, T)RADE,
      POOL G)OLD, DIV)IDE GOLD,
      R)EAD SPELL BOOKS, OR L)EAVE.
```

At camp:

```
NIN HUMAN E-NINJA

STRENGTH 13      GOLD      2886
I.Q.      14      EXP      147427
PIETY     15
VITALITY  15      LEVEL    10      AGE     25
AGILITY   16      HITS     71 / 71   AC      0
LUCK      13      STATUS   OK

MAGE      0/0/0/0/0/0/0
PRIEST    0/0/0/0/0/0/0

*=EQUIP, -=CURSED, ?=UNKNOWN, #=UNUSABLE

YOU MAY E)QUIP, D)ROP AN ITEM, T)RADE,
      G)OLD, DIV)IDE GOLD,
      U)SE AN ITEM, I)DENTIFY AN ITEM,
      CAST S)PELLS,
      R)EAD SPELL BOOKS, OR L)EAVE.
```

At the Adventurer's Inn:

```
+-----+
! CASTLE                                INN !
+----- CURRENT PARTY: -----+
# CHARACTER NAME CLASS AC HITS STATUS
1 NIN 1 14 000000
2 NIN 2 12 000000
3 PRI 1 10 000000
4 MAG 1 10 000000
5 PRI 1 10 000000
+-----+

WELCOME NIN. WE HAVE:

[CA] THE STABLES (FREE!)
[CB] COTS. 10 GP/WEEK.
[CC] ECONOMY ROOMS. 50 GP/WEEK.
[CD] MERCHANT SUITES. 200 GP/WEEK.
[CE] ROYAL SUITES. 500 GP/WEEK.

[CG] POOL PARTY GOLD.

OR [RETURN] TO LEAVE
```

At Boltac's Trading Post:

```
+-----+
! CASTLE                                     SHOP !
+-----+
CURRENT PARTY:
# CHARACTER NAME CLASS AC HITS STATUS
1 FIG1 -FIG -4 500 500
2 FIG2 -FIG -4 500 500
3 NIN -NIN 100 500 500
4 PRI -PRI 100 500 500
5 MAG -MAG 100 500 500
6 PRI2 -PRI 10 800 500
+-----+

WELCOME NIN
YOU HAVE 2886 GOLD

YOU MAY POOL GOLD, BUY AN ITEM,
SELL AN ITEM, HAVE AN ITEM
UNCURSED, OR HAVE AN ITEM
IDENTIFIED, OR LEAVE
```

#064 - Allow transfer characters to any scenario copy

Status: Fixed in WC042. Recompiled. Verified.

Contributors: Eric Labelle.

Enhancements: Added the possibility transfer characters to any scenario copy even if it has the same serial number as the source.

Spelling: Full spelling and corrected double "R" from "\*\*\* CHAR HAS NON-XFERRABLE ITEMS \*\*" to "\*\*\* CHAR HAS NON-TRANSFERABLE ITEMS \*\*".

#065 - Kandi spell enhancements

Status: Fixed in WC043. Recompiled. Verified.

Contributors: Eric Labelle.



### #067 - Auto Update Hit Points of Poisoned Characters

Status: Fixed in WC045. Recompiled. Verified.

Contributors: Eric Labelle.

Enhancements: Added code to automatically update the remaining hit points of poisoned characters in the status window while walking in the maze.

### #068 - Monsters cannot yell for help when surprised

Status: Fixed in WC046. Recompiled. Verified.

Contributors: Eric Labelle.

Enhancements: Prevent monsters from yelling for help when surprised. This was a guaranteed party slaughter (game breaker) in many scenarios. For Proving Grounds for example, on level 10, if four Dragon Zombies surprised you, you could end up facing eight Dragon Zombies the next round, no party can survive this. And if there are multiple groups of monsters surprising you it was even worse.