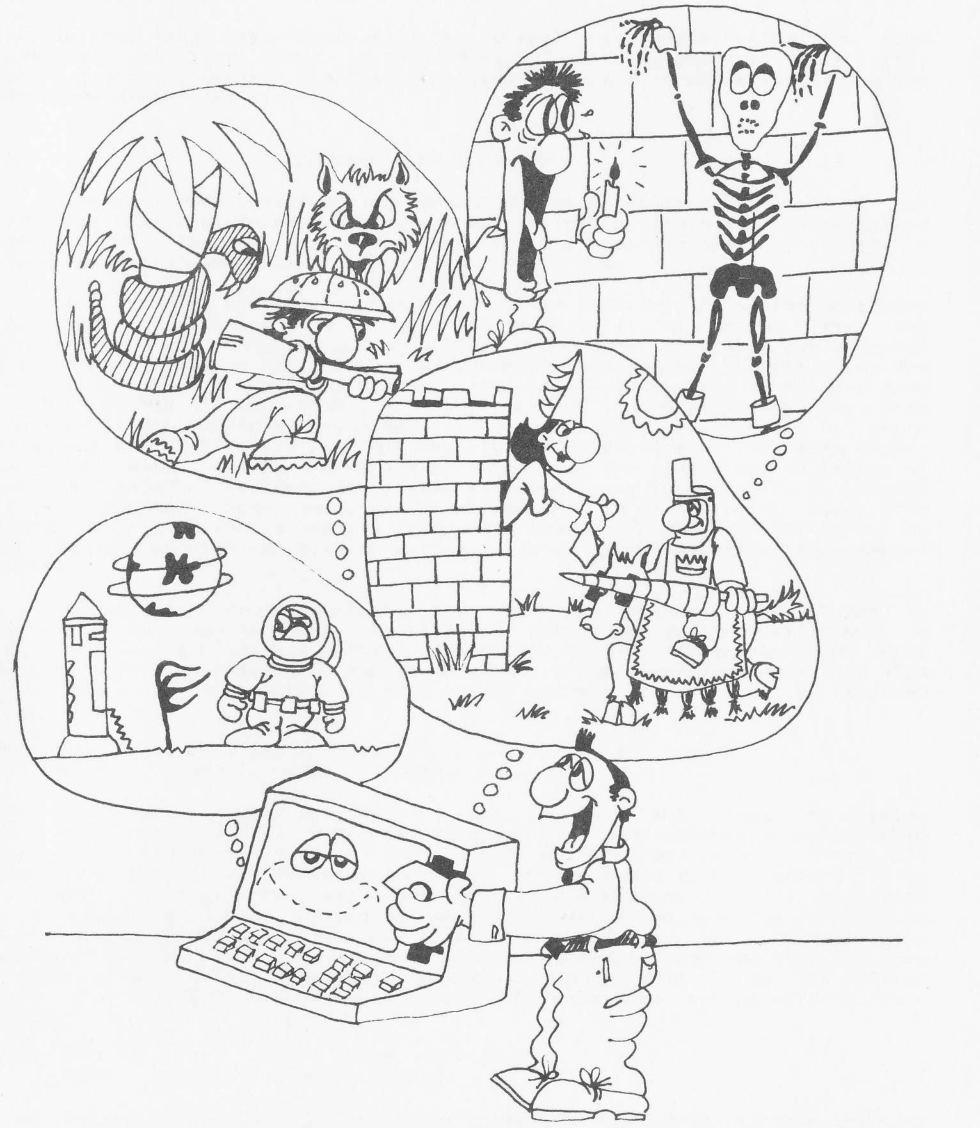
R.B.

# AUGMENT

The official newsletter of "The Auggies"

#1



AUGMENT is published quarterly by Charley Butler and Bruce Hansen
at 704 North Pennsylvania Avenue, Lansing, Michigan, 48906
AUGMENT and "The Auggies" are copyrighted by The Alternate Source

THE ALTERNATE SOURCE

704 N. Pennsylvania
Lansing, Michigan 48906
Ph. (517) 482-TAS0
Source: TCH565
Micronet: 70150,255

Bulk Mail
U.S. Postage Paid
Permit 815
Lansing, MI

Return Postage Guaranteed
Address Correction Requested

# AUGMENT

## Official Newsletter of "The Auggies"

Welcome to the first issue of AUGMENT, the Adventure Users' Group newsletter. AUGMENT will be a quarterly newsletter unless you demand a more frequent contact.

In this issue many topics will be covered, including: alternate room registers, counters, errors in the TAS manual, utility program availability, future products, general tips and licensing procedure to allow you to use The Adventure System driver.

### ALTERNATE ROOM REGISTERS

Alternate room registers are perhaps the most confusing facet of the adventure data base format. There are currently six alternate room registers numbered 0-5. The main purpose of the alternate room registers is to save a player's current location for recall at a later time.

For example, suppose you had an adventure with days being accounted for (night and day). After so many moves the player will fall asleep. When the player falls asleep he has a dream. This dream could be a hint for upcoming attractions or a clue for solving the adventure, whatever. In this case, the dream will move the player to a "hint" room, display the hint room, then move him back to his original room. A "GOTOY" command in the action entries with the appropriate parameter could move the player to the "hint" room, but there is no way a "GOTOY" could send him back to the original room (the room number parameter would change depending on which room the player was in before he had his "dream"). However, the player's current room number could be stored in an alternate room register before moving him to the "hint" room. The commands for using the alternate room registers are "EXRMO" and "EXC,CR". In this example we will use alternate room register 0 thus utilizing the command "EXRMO".

The scenario to demonstrate this goes as follows: the player is currently in room number 5. He will fall asleep and be temporarily sent to room 17 (the "hint" room). After a delay (via the "DELAY" command) he will return to his original room number or room 5. For simplicity sake, bit flag 1 will be set if the player has just fallen asleep. The actions to perform this could be:

```
AUTO 100     BIT 1     PAR 1     PAR 17       0          0
    CLRZ      EXRMO     GOTOY      CONT
```

This action will test if bit flag 1 is set. If it is set, then the player has just fallen asleep. NOTE: in this example, some actions preceding this one would have set the flag if the player had fallen asleep. Since all conditions in this action entry were true, the commands will be executed. In this case, bit flag 1 is cleared and the player's current room is exchanged with the room number currently stored in alternate room register 0. It should be noted at this point that after the exchange, the player's room number will be a bogus value unless the alternate room register had been previously set. At this point the player is sent to room 17 or the "hint" room. The continue flag is also set. The subsequent action entries could be:

```
AUTO    0         0         0         0          0          0
    DSPRM     DELAY     EXRMO     DSPRM
```

This action will display the player's current room, which at this point is room number 17 (the "hint" room). The "DELAY" command will pause the display for about one second. After the pause, the player's room number before he fell asleep is recalled by doing another "EXRMO". This command will exchange his current room number (17) with the room number held in alternate room register 0 (room 5). The "DSPRM" command will display the player's current room which at this point is room number 5.

## SETTING ALTERNATE ROOM REGISTERS

There is no application I can think of for setting the room registers but an explanation will be given anyway. My not seeing any application doesn't mean there aren't any - just none I have thought of.

Suppose we are currently in room number 5 and we want to set alternate room register 2 to room 10 and alternate room register 5 to room 23. The action entry to do this could be put in some initialization routine, such as displaying the opening message, etc. The example here assumes this case.

```
0: AUTO 100    -BIT 1     PAR 2      PAR 10       0          0
        EXC,CR    GOTOY      CONT       MSG8
1: AUTO  0       PAR 2      PAR 5      PAR 23       PAR 5      0
        EXC,CR    EXC,CR    GOTOY       EXC,CR
2: AUTO  0       PAR 1      0          0            0          0
        SETZ        -          -          -
```

Action 0 is the initialization action entry. When adventures are started, all bit flags are cleared. Therefore, at the start of an adventure, a "-BIT" condition will always be true. If in the same action(s), the bit flag is set, the initialization actions will not by executed again. The first "EXC,CR" with an associated parameter of 2 exchanges the player's current room with alternate room register (arr) 2. In this case, arr 2 will hold a 5 after the switch since the player's current room was 5. The GOTO 10 sets the player's room number to 10. The CONT is used to tell the adventure driver program that this action entry will CONTinue on with AUTO 0 actions. Message 8 was displayed as some sort of opening message.

Action 1 exchanges the player's current room and arr 2 again. Before the switch, arr 2 holds 5 and the current room is 10. The switch sets arr 2 to 10 and the current room to 5. Thus, arr 2 is set as we originally determined. Next, arr 5 and the current room are exchanged. This sets arr 5 to room number 5. The GOTO 23 sets the current room to 23. The last EXC,CR switches arr 5 and the current room back again. This second exchange sets arr 5 to 23 and the current room to 5. Our task is now completed except for one detail, the initialization bit flag must be set so this set of actions are not executed again. Action 2 accomplishes this by setting bit flag 1.

## COUNTERS

Counters are very useful in adventures. Applications include counting the number of times a situation has occurred. Scott Adams uses a counter in adventure 6 (Strange Odyssey) to keep track of the number of moves you've had the space suit on. The space suit has a limited air supply so the number of moves you've drawn air from its tanks is counted.

Our example will involve a six shooter. Originally this gun has six shots. However, each subsequent shot subtracts one from a counter holding the number of shots left. Our scenario is that you shoot a Buffalo with the gun. If you have shots left in your gun when you shoot the Buffalo ("SHOOT BUFF"), you will kill it and get some "dead meat". Other actions include "EXAM GUN" which will tell you the number of shots left. A "SHOOT ANY" action is used to subtract from the shots left when you shoot something other than the Buffalo. Since at the start of the adventure all counters are not set to any predetermined value, an initialization action should set our shot count to 6. In this case we will use counter 3 for our shot count. The initialization action could be something like this:

```
AUTO 100    -BIT 1    PAR 1     PAR 3      PAR 6      0
    SETZ       EXM,CT    CT<-N     CONT
AUTO  0       PAR 3      0          0          0          0
```

```
    EXM,CT    MSG8       -          -
```

The first action tests if this is an initialization action by testing bit flag 1. If bit flag 1 is cleared, this action is executed. Bit flag 1 is subsequently set so this action will not be executed again. The "EXM,CT" command and the associated parameter 3 exchange the CT counter value with alternate counter 3. Next, the CT value is set to 6 with the "CT<-N" command (and parameter 6). The CONTinue flag is set since more commands are required than exist in one action entry. The next action switches CT and alternate counter 3 back again. Then our friendly opening message 8 is displayed.

The "EXM,CT" commands were necessary since adventure allows only testing, setting and arithmetic to be done to the CT counter. The other alternate counter (0-7) can not be set or tested directly, they must first be exchanged with the CT counter.

Before giving the rest of the actions in this example, the objects and messages used are defined:

| Object # | Object description |
|---|---|
| 3 | Buffalo |
| 4 | Dead meat/MEA/ |
| 5 | Six shooter/GUN/ |

| Message # | Message text |
|---|---|
| 8 | Welcome to example adventure! |
| 9 | GOT HIM! |
| 10 | BANG!! |
| 11 | It's a fine looking thunder stick! It has |
| 12 | shots left. |
| 13 | Click |
| 14 | No buffalo here |

These objects and messages will be used in the rest of the action entries. Objects 4 and 5 may be carried since they are named objects.

```
10: SHOO BUFF    IN/W 3    HAS 5     PAR 3      0          0
         EXM,CT     CONT        -          -
11: AUTO  0       CT=0       0          0          0          0
         MSG13       -          -          -
12: AUTO  0       CT>0       PAR 4      PAR 3      0          0
         MSG10      MSG9       EXX,X      CT-1
13: AUTO  0       PAR 3      0          0          0          0
         EXM,CT      -          -          -
14: SHOO BUFF    -IN/W 3    0          0          0          0
         MSG14       -          -          -
```

Action 10 is evaluated if the player types "SHOO BUFF". The conditions are "IN/W 3" and "HAS 5". These conditions check if the player is in the same room as the buffalo and if he is carrying the gun. If either of these conditions are false, then this "SHOO BUFF" action is skipped and the next "SHOO BUFF" action is tested. For now let's assume the conditions are true. The commands of action 10 will exchange counter 3 and the CT value. The CONTinue flag is also set.

Action 11 is considered next. If the CT value is equal to zero (no more shots left), then message 13 is displayed (Click). Regardless if action 11 was true or false, action 12 is also considered. This is the nature of CONTinued actions. Action 12 checks if the CT value is greater than zero (some shots are left in the gun). If there are some shots left, messages 10 and 9 are displayed (Bang! Got him!), the location of the Buffalo and dead meat are switched (the dead meat would have originally been in the storeroom) and one is subtracted from the CT value. The order of actions 11 and 12 was not arbitrarily selected. If the order were switched and the shot counter would have had one bullet left, subtracting one from it would set it to zero. The action 11 would have seen a zero CT value and displayed the message "Click". In is intended that either one of these CONTinued actions be true, not both. Switching their order raises the possibility that both could be true.

Action 13 is used to switch the CT value and counter 3 back again. Whenever a counter is switched, care should be taken to switch it back again. This is the reason that the second exchange was done on a separate action in this example.

If the player typed "SHOO BUFF" and the conditions of action 10 were not met, then action 14 would be considered. The condition of action 14 is a "-IN/W 3". Therefore, if the player is not in the same room as the Buffalo, this action is executed. If this is the case, then message 14 is displayed (No buffalo here).

The "EXAM GUN" action could be used to find out how many shots were left in the gun. It could read:

```
20: EXAM GUN    IN/W 5   PAR 3      0         0         0
        EXM,CT    CONT     MSG11       DSPCT
21: AUTO  0       PAR 3     0          0         0         0
        EXM,CT    MSG12     -          -
```

Action 20 is executed if the player is in the same room as the six shooter. If this is true, then counter 3 is exchanged with the CT value. Message 11 is displayed next (It's a fine looking thunder stick. It's got). The number of shots left is then displayed by the "DSPCT" command (CT holds the number of shots left). The CONTinue flag is also set. Action 21 is used to exchange CT and counter 3 back again. Message 12 is also displayed (shots left).

Some action is also needed so something is displayed when the gun is shot. There are two cases involved. First, if the gun is empty, the message "Click" is displayed. Secondly, if the gun has some shots remaining, the message "BANG!" is displayed and one is subtracted from the shot counter. These actions could be:

```
96: SHOO ANY   HAS 5    PAR 3      0         0         0
        EXM,CT   CONT      -          -
97: AUTO  0     CT=0      0          0         0         0
        MSG13    -         -          -
98: AUTO  0     CT>0      0          0         0         0
        MSG10    CT-1      -          -
99: AUTO  0     PAR 3     0          0         0         0
        EXM,CT   -         -          -
```

Action 96 checks if the player is holding onto the gun. If he is, then CT and counter 3 values are exchanged and the CONTinue flag is set. Action 97 is considered next. If there are no shots left, then message 13 is displayed (Click). Action 98 checks if any shots remain. The number of shots left is greater than zero, then message 10 is displayed (BANG!) and one is subtracted from the shot count. Action 99 is used to exchange CT and counter 3 back again.

Notice that the "SHOO BUFF" action preceded the "SHOO ANY" action. If the "SHOO ANY" action preceded the "SHOO BUFF" action, the later would never be executed. If the player is holding onto the gun when he types "SHOO BUFF", the "SHOO ANY" action would be found true and the "SHOO BUFF" action would never be reached. It is good practice to put the more "limited" actions before the less "limited" ones. See the general tips section for more information.

## MANUAL ERRORS

Whenever a document the size of THE ADVENTURE SYSTEM manual is produced, there are bound to be a few errors. Fortunately, these were kept to a minimum in this case.

However, there is a big problem with the manual. Somebody forgot to number the pages! The pages should be numbered as follows: the Preface should be "i". The Introduction should be "iii". The chapters should be

numbered in the form "c-n" where "c" is the chapter number and "n" is the page number of that chapter. For example, the fifth page of chapter 4 would be numbered "4-5". From this point on I refer to pages in this format.

## ADVENTURE SYSTEM UTILITIES

There are currently three utilities for TAS, "ADVTAPE", "ADVCOPY" and "ADVDUMP". There may be more in the future if the need arises.

"ADVTAPE" will read in a disk data base and dump out a tape version of the adventure. This tape contains the copyrighted program "ADV" and tapes cannot be sold. However, you may give a copy to a friend so he can play your adventure.

"ADVCOPY" will read the adventure data base from a Scott Adams' protected diskette and dump it out to a non-protected disk. Scott has recently started selling his adventures in a non-protected format, thus eliminating the need for this program. For this reason, the assembly language listing of this program is presented next issue in this newsletter! There are some limitations with this program. Being of the lazy nature prompted me not to fix some major problems and present the program on an "as is" basis. One serious limitation is that the program will read only a one adventure protected diskette. The three adventure diskettes will not work properly. Enterprising users can figure out how to read the three pack diskettes. If anyone does come up with a solution and would like it printed in this newsletter, send it to AUGMENT. Remember, this newsletter pays bucks for your submissions (I don't want to write this whole thing myself)!

"ADVDUMP" is the newest of the utilities. This program will read in a tape adventure and write the disk data base to disk. It appears from the phone calls I have received that there is a demand for this program. Suppose you originally bought a tape machine and thus tape adventures. Upgrading to disk left you with adventures on tape that can not be loaded into the "ADVEDIT/BAS" program or played with the "ADV/CMD" program. "ADVDUMP" will read in the adventure tape and dump out a disk file that can be used by "ADVEDIT/BAS" and "ADV/CMD".

Many have asked about the price of the utilities. The utilities are $14.95 each or two for $24.95. They are only available on disk (since only disk people are using TAS). The "ADVCOPY" program will not be sold. Orders should be sent to AUGMENT. The master disk containing both utilities will also include the source and object code for ADVCOPY.

## FUTURE PRODUCTS

At this time I am working on a new version of the adventure editor program "ADVEDIT/BAS". This new version will be written totally in machine language. This will increase speed and allow more actions, rooms and objects to be input.

Once this is accomplished, the editor and driver program will be combined into one program! For those who have written adventures of any length you know how much of a pain it is move back and forth between the editor and driver. This new editor/driver will allow movement between the two at the press of the CLEAR key! Suppose you were working on an adventure and were testing it by playing it. Alas, a problem arises. Now you must exit the driver program and go to the adventure editor. Now your data base must be loaded, fixed and written out again. Finally, you run the driver program an continue playing your adventure. This can be very time consuming. However, with the new editor/driver, all that is required to move between the editor and driver is a press of the CLEAR key! Needless to say, we are very excited about this new product. Availability should be somewhere around summer (1982 hopefully). AUGGIES will be able to upgrade before anyone. Information next issue.

A tape version of THE ADVENTURE SYSTEM is also in the works. This will come out some time after the above mentioned editor/driver. It will be easy to support on the Model III because Model III people don't have to have an interface before they add more memory. We probably will make a version available for Model I people, if enough people request. The Model III 1500 baud cassette is much more tolerable than the Model I 500 baud, also.

## USERS INPUT

We are actively seeking user written adventures either for publication in this newsletter or in our adventure library. For AUGGIES to take advantage of their special discount on adventures, users are going to have to submit them. Royalty percentages on adventures accepted for marketing are substantial, so send them in! We need small adventures, too!

The adventure driver program currently has 13 unused commands (commands such as GETX, GOTOY, SAYW, etc.). We plan on expanding and enhancing the adventure data base format over its present form. If there are any commands you would like to see added please let us know! The command you ask for just might be added! An example of an added command is "CT+1". There is currently a command to decrement the CT value "CT-1", but no command to increment it. It is possible to increment CT with the "CT+N" command, but this uses a parameter also.

This newletter will probably be used to get opinion on potential commands. More information to come in future issues.

## LICENSING PROCEDURE

As you have read before, the adventure driver program "ADV/CMD" is copyrighted by yours truly. You have four choices if you want to market your own adventures: 1. Send them to AUGMENT for possible inclusion in this newsletter or adventure library, 2. Try to sell them to another company, 3. Sell them on your own without the adventure driver program, "ADV/CMD", or 4. Sell them on your own with the adventure driver program "ADV/CMD".

The cost of licensing the adventure driver program is $200. This will give you rights to distribute any adventures you write with the "ADV/CMD" program and the "ADVTAPE" program. You pay $200 if you sell one adventure or a dozen. This does not given to the power to transfer rights to market to a third party. In other words, you can't sell or give the driver to someone else (such as a dealer) with the intent to market their adventures with the driver. Software dealers must purchase a set of rights for each person's adventures the dealer markets. For example, if company ABC intends on marketing adventures by John Doe and Joe Doaks, the dealer must purchase two packages, one for John Doe and one for Joe Doaks.

What you get for $200 is a copy of "ADV/CMD" and "ADVTAPE/CMD". Any bugs found in either of these programs will be fixed free of charge. The source code for "ADV/CMD" or "ADVTAPE/CMD" is not included. The "ADV/CMD" program released will be the Scott Adams compatible version. Any version of "ADV/CMD" with new commands added (such as CT+1) will not be distributed. These enhanced versions are reserved for adventures marketed through The Alternate Source.

## GENERAL TIPS

This section will be a regular in upcoming issues of AUGMENT. It will include coding methods to perform certain tasks. If you have any suggestions, please send them to the editor.

### TIP ONE

The first tip covers the proper order for "limited" action entries. Action entries are evaluated in numeric order. Normally the order does not matter (except for auto actions), but in the case of "limited" action entries, order does matter. This is better explained with an example.

One very common action entry is "EXAM ANY" which responds with something like "I see nothing special". This action is used in response to a user input of "EXAM" and any legal noun. Some objects may have a specific action describing them when "EXAM object" is entered (as in the "EXAM GUN" example in the COUNTERS section). Suppose when we typed "EXAM DESK", we wanted to display the message "It has a drawer". Let's assume the desk is object 21, Message 16 is "I see nothing special" and Message 17 is "It has a drawer". The actions to display these messages could be:

```
35: EXAM ANY    0         0         0         0         0
         MSG16
36: EXAM DESK   IN/W 21   0         0         0         0
         MSG17
```

Now suppose the player typed in "EXAM DESK". Adventure would start scanning the actions for a match. The first match found would be action 35 ("EXAM ANY"). Since no conditions are present, they are all true and message 16 ("I see nothing special") is displayed. Notice that action 36 ("EXAM DESK") was never reached so the message "It has a drawer" will never be displayed. Action 36 is a more "limited" action than action 35. As a result, the order of these two should be switched for proper operation.

### TIP TWO

The second tip given is a potentially powerful technique currently not used in any of Scott Adams' adventures. Anyone who has played one of Adams' adventures (or the ones supplied on the TAS master diskette) knows that if an obvious exit is indicated (such as NORTH, EAST, etc.), that the player can always move in that direction. The technique presented here overrides this movement.

Our scenario is this: We are moving through some caves. A dragon is setting at a fork in the caves. This fork was reached by moving NORTH, therefore, moving SOUTH takes us back. However, once we reach the fork, possible exits are NORTH, SOUTH, EAST and WEST. The NORTH, EAST and WEST exits go further into the caves. There will be a stipulation on the NORTH, EAST and WEST exits. You can only go in one of those directions if the dragon is gone. For example, you can't go NORTH at the fork if the dragon is in the room. In this example, "SAY BOO" will get rid of the dragon. You can go SOUTH if the dragon is there or not. The data base sections used in this example are listed as follows:

```
Msg#  Message Description
5:    Dragon won't let me
6:    I scared him!  He flew away!

Obj#  Object Description
3:    Dragon

Rm#  N   S   E   W   U   D   Room Description
11:  12  0   0   0   0   0   Cave
12:  5   11  7   9   0   0   Fork in caves
```

The actions to perform the described functions are:

```
4:  AUTO 100   IN/W 3    PAR 1     0         0         0
         SETZ  -         -         -
5:  AUTO 100   IN 11     PAR 1     0         0         0
         CLRZ  -         -         -
6:  AUTO 100   -IN 12    BIT 1     PAR 12    0         0
         MSG5  GOTOY     -         -

50: SAY  BOO   IN/W 3    PAR 1     PAR 3     0         0
         MSG6  CLRZ      X-RMO     -
```

Action 4 sets bit flag 1 if the player is in the same room as the dragon. This flag indicates if the player is in the same room, or was just in the same room as the dragon.

Action 5 clears hit flag 1 if the player has moved go room 11. Room 11 is the always allowed exit from the fork in the caves.

Action 6 tests if the player is not in room 12 (fork in the caves) and if hit flag 1 is set. The only way hit flag 1 is set is if the player is in contact, or was just in contact with the dragon. The "-IN 12" condition tests if the player has left the room described as "Fork in caves". If both of these are true (as they would be if the player moved NORTH, EAST or WEST when the dragon was in the room), then message 5 is displayed ("Dragon won't let me") and the player is sent back to room 12. Since moving to room 11 clears hit flag 1 (in action 5), a move to the SOUTH is allowed.

Action 50 checks if the player is in the same room as the dragon when he "SAY BOO". If so, message 6 is displayed, the dragon is sent to the storeroom and hit flag 1 is cleared.

If, for example, the EAST exit to room 7 was to be allowed if the dragon was at the fork or not, an action exactly like action 5 (except "IN 7" would be substituted for "IN 11") would be placed between actions 5 and 6. Note that the order of actions 4-6 is very important. If action 5 was entered after action 6, then a move to the SOUTH or room 11 would not be allowed.

## IDEAS FOR POTENTIAL ADVENTURES

Can't find that spark of inspiration to get you started? Try these:

* How about a tour of the 50 states? Each state has its own distinct characteristics. Perhaps the player would have to guess the capitol before moving on. You would only have 5-6 actions in each state. Designing your rooms would be a bear!

* Most adventures I've seen are horizontal. Why not design a vertical adventure? Perhaps you could use the blazing inferno theme.

* Our first adventure data base submitted designed with The Adventure System was written by Jack Scantlin, owner of a ski resort in California. Any recreational area should be interested in the additional publicity they could get from a "vicarious implementation" of their area. We are more than willing to discuss arrangements for co-op advertising with various parks. Design the adventure, let them know they would be reaching one of the most affluent groups in the world and have them (or you) contact us.

Got plenty of ideas for adventures? Let us know. $5 for any idea published in AUGMENT! All submissions must be written. In case of duplicate ideas, the earliest postmark will be printed. We are especially interested in potential educational applications. We will also guarantee you a free copy of any adventure created using your idea. Send ideas to AUGMENT.

## THE BEGINNERS CORNER

We are confident that The Adventure System manual is adaquate for people used to learning new languages in one setting. Not so for people who may just now be learning their second (maybe even first!) language. This portion of AUGMENT is for persons just breaking into the study of languages. Our goal is to provide, step by step, instructions for getting you familiar with the System, by providing dialogue that you can imitate and gradually incorporate into your library of "how to do it." Imitation is used by most schools for entry level programming classes.

The following session will allow you to make modifications to our sample adventure contained on your master TAS disk (ADVENT/Z). Our goal will to make the adventure just a little better by adding one more room and a new treasure. Perhaps after making this modification, you will want to add even more to ADVENT/Z -- maybe even convert it into a full length adventure. Tell you what: we will provide a free adventure to the three people who submit the best expansion of ADVENT/Z before the next issue of AUGMENT. One or more of the winners will be published in an upcoming issue of AUGMENT (and receive

our regular author compensation)! You have about two months, so get busy!

In the following dialogue, MENU-P stands for the primary menu; that is, the menu that displays which portion of ADVEDIT you want to go to. MENU-S will will refer to any secondary menu which further defines how deep you want to go into the editor. For example, if you select V (for vocabulary) from the primary menu (the first one displayed when you run ADVEDIT and the one you can usually get back to simply by hitting "-"), you will be asked whether you want to modify Verbs or Nouns (V,N). The "Verbs or Nouns" question is the secondary menu. Notice that in the column below, under "You Say:" will always contain an option from the menu that should be on your screen. If not, suspect errata!

```
When the computer           You say:                 Comments:
     says:

DOS READY                   XREF                     ;load machine  language  portion
of                                                     ADVEDIT
DOS READY                   BASIC                    ;read the Disk BASIC Chapter
NUMBER OF FILES?            <ENTER>                  ;  in your DOS manual
MEMORY SIZE?                60000                    ;Protect XREF
BASIC READY                 RUN"ADVEDIT/BAS"         ;to load BASIC portion of
                            <ENTER>                    ADVEDIT
MENU-P                      R                        ;read a data base from disk
ADVENTURE#                  Z                        ;that's our "baby" adventure
DRIVE#                      <ENTER>                  ;take the first "Z" found
MENU-P                      M                        ;go into modify mode
MENU-S                      H                        ;always modify the header first
OPTION1                     <ENTER>                  ;leave actions at 41
OPTION2                     <ENTER>                  ;ditto for vocabulary
OPTION3                     9                        ;we need another room
OPTION4                     17                       ;I think we'll need another
                                                       message
OPTION5                     15                       ;we'll need another object
OPTION6                     <ENTER>                  ;carry limit is okay
OPTION7                     <ENTER>                  ;start room okay
OPTION8                     2                        ;we'll add Radio Shack's answer
                                                       to IBM
OPTION9                     <ENTER>                  ;word length okay
OPTION10                    <ENTER>                  ;light duration okay
OPTION11                    <ENTER>                  ;treasure room okay
MENU-S                      R                        ;You're still in modify mode
WHICH ONES                  9,9                      ;only one
ROOM 9                      0,1,0,0,0,0              ;zeros, not ohs
                            <ENTER>
DESCRIPTION                 *I see a Radio Shack Computer Center <ENTER>
                                                     ;refer to Note One
MENU-S                      R                        ;modify another room
WHICH ONES                  1,1                      ;only one
ROOM 1                      9,0,2,0,0,0              ;access room 9 from room 1
                            <ENTER>
DESCRIPTION                 <ENTER>                  ;leave it alone
MENU-S                      O                        ;Oh, for objects -- still in
                                                       modify mode
WHICH ONES                  15,15                    ;modify object 15 in modify mode
OBJECT16                    6                        ;starts in room six
DESCRIPTION                 *White Model III with RAM BASIC */TRS-/;
                                                     ;What?  No hard disk?
MENU-S                      V
WHICH ONES                  21,22                    ;only number 21
MENU-S                      N
NOUN 21                     STORE
NOUN 22                     TRS
MENU-S                      A
WHICH ONES                  41,41
ACTION 41                   GO,STORE
COND,VALUE                  IN,9
COND,VALUE                  <ENTER>
```

```
COND,VALUE     <ENTER>
COND,VALUE     <ENTER>
COND,VALUE     <ENTER>
CMD1           17
CMD2           <ENTER>
CMD3           <ENTER>
CMD4           <ENTER>
TITLE          <ENTER>
CORRECT?       Y
MENU-S         M
WHICH ONES     16,16
MESSAGE        IT'S CLOSED.
MENU-S         -                ;minus sign = exit
MENU-P         W                ;write to disk, v1.02
ADVENT#        0                ;don't overwrite "Z" yet
DRIVE          0                ;Or whatever drive has room
MENU-P         E                ;Back to BASIC

READY          CMD"S"           ;Back to DOS
DOS READY      ADV<ENTER>       ;Execute Adventure 0!
```

Now try your changes! Two things that would be really easy to change:

1. You can't "GO CENTER" or "GO RADIO" or "SHACK."
2. You can't pick up the new TRS-80

Perhaps you can try these modifications yourself!

We would be interested in someone to develop a beginner's column for a regular series. You could cover a variety of areas, including modifying existing adventures (you will always present user dialogue) and creating new ones. Perhaps you could develop an adventure while writing the column.

## PATCHWORK!

Alas! We thought The Adventure System had been thoroughly tested before we shipped any copies. Only two small problems have appeared. Note! Most versions of TAS that we have shipped have these corrections already made! Neither problem will cause severe errors, but we want you to be aware of them, just in case. For making the patches, we recommend TASMON, the utility monitor available from The Alternate Source.

All changes are to be made to ADV/CMD. Load ADV/CMD into memory using TASMON, and examine the bytes starting at the address of "Location" below. If the "Old Byte" value is there, replace them with the value under the column "New Byte." Then save the corrected file back to disk. Model III people can also use the PATCH command to change the bytes. See your TRSDOS 1.3 manual for instructions on using the PATCH command.

| Location | Old Byte(s) | New Byte(s) |
|---|---|---|
| 5E51H | C4,64 | 6F,65 |
| 5E66H | C4,64 | 6F,65 |
| 5EA2H | C4,64 | 6F,65 |
| 55C5H | C6,4B | D6,57 |

These changes fix two problems: If an adventure is played a second time, that is, if you get the message "This Adventure is over. Play Again (Y/N)?" and you respond "Y" the light source is not initialized for the second game. The other bytes improve the Random function for AUTO action entries.

## BYE, FOR NOW

Well, that's it for this issue. Please send any comments or requests for future coverage to either the editor (Bruce Hansen) or AUGMENT. A section I would like to add to this newsletter is "QUESTIONS AND ANSWERS", so send any questions you have. All perspective authors should send their

submissions to either the editor, Bruce Hansen, or AUGMENT. Sunday from noon to six is our "idea day." On that day personnel will be standing by to discuss potential adventure ideas, articles you could write for AUGMENT, listen to your comments on improving The Adventure System, and general comments and criticisms (how was our service in delivering The Adventure System to you? Were you treated promptly and courteously by personnel handling your problems with TAS?), a general "let us know" day. Only qualified personnel will be on hand that day. If no one answers, or you get a busy signal, it means all available personnel are tied up. We do not require that anyone work on Sunday; it is entirely voluntary, although all personnel are compensated for time invested. Please try again later. If you have consistent trouble getting through, send us a letter and include you phone number. If you can sell your idea to us through the letter, perhaps we will call you to discuss possible manifestations.

**********

## NEXT ISSUE...IN AUGMENT!

* A free user submitted adventure
* Complete source listing to ADVCOPY
* Information about the NEW adventure driver package
* More information to help you understand the System better

# Join the Auggies!

☐ I'd like ADVTAPE for $14.95, please.
☐ I'd like ADVDUMP for $14.95, please.
☐ I'd like BOTH utilities for $24.95, please.
☐ I'll join the AUGGIES for $12.00!

*Name:* ______________________
*Address:* ______________________
*City:* ______________________
*State, Zip:* ______________________

*Mail to:*

The Alternate Source
704 N. Pennsylvania Ave.
Lansing, Michigan 48906
Ph. 517-482-TASO

Please add $3.00 shipping.
Please add $1.75 for COD.
No extra charges for M/C or Visa.

## NOTES