AmiMUD

Gabriele Greco

COLLABORATORS			
	TITLE : AmiMUD		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Gabriele Greco	January 18, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AmiMUD

1.1 manual"

AmiMUD 1.0 by Gabriele Greco

1.2 dist

Distribution

AmiMUD is shareware, the shareware version may be distributed everywere through internet, bbs, cd-rom, cover disk provided that the archive is distributed in his original form and that the price of the cd-rom/cover disk isn't too high.

The shareware version isn't crippled in any way. It opens only a "reminder" requester when you launch the program.

Probably future versions may be crippled, I think that the planned arexx port available in the next major upgrade will work only in the registered version of AmiMUD.

How to register

1.3 req,"requirements"

Requirements

AmiMUD needs to work:

- an Amiga with at least OS 2.0 and 1MB of RAM (some features work only with OS 3.0+).

- reqtools.library (included in the archive)

- a TCP stack installed and active (tested with AS225, AmiTCP and MLink)

- if the TCP stack is amitcp it requires socket.library (included in the archive)

Not required but useful:

- the file s:amimud.prefs to keep all the configuration settings of the program

- KingCON to have a review buffer on the output window. (you can find KCON on aminet util/shell/kingcon13.lha)

1.4 intro,"introduction"

Introduction

AmiMUD is a MUD client. It runs using socket and can be used to play nearly every text-only mud on the internet or on other TCP/IP based networks.

I wrote this because I saw that there wasn't any decent MUD client for amiga, since I enjoy myself playing muds and I don't like using term or the amitcp telnet to do that I decide that writing a MUD client was a good Idea :)

It was thought for DIKU-type muds because that's the kind of mud I play, but can be used with nearly any type of mud without any problem, obviously some of the features (for instance tickcounter) are not useful on some types of mud, but I don't think this is a big problem, if you have suggestions for features you'll like to see in AmiMUD mail me them.

My initial thought was to make it freeware, but I spent many time (and money in phone bills :)) on it so I decide to make it shareware, so if you like it and you use it regularly please register, you will make future improvements being possible.

1.5 Features..."

These are some of the features of AmiMUD:

- Input and Output on two windows.

- Command history buffer on output window.

- Possibility to log to file what you do.

- Can be opened on Workbench, Public screens or on a custom screen...

- ANSI color support (on custom screen).

- Macros (with or without arguments) and with hotkeys.

- Numeric Keypad walking.

- Autologin.

- Triggers.

- Variables.

- Many internal commands to use in triggers and macros.

- Dynamical tick length counter (for DIKU muds).
- Nearly all function may be used by menu.
- AmiMUD is not a unix porting!
- Multiple config capability.
- Possibility to edit / use custom palettes.
- Very little (50k) and doesn't need fast cpus.
- Commodity interface.
- Send a file or clipboard to the mud.

...and many other features!

1.6 inst,"installation"

Automatic installation

Simply click on the Install_AmiMUD icon in the AmiMUD drawer.

Manual installation

To install amimud simply copy the file AmiMUD (and optionally the icon AmiMUD.info) on your HD (or on a disk).

To use all the features of the program you'll need also to copy the file AmiMUD.prefs to your s: assignment.

Then you'll need to edit the prefs file as you like (see Configuration section for more info) then you can launch the program from WB or CLI.

If you don't have already installed reqtools.library and socket.library you'll need to install them:

To install reqtools simply copy amimud/libs/reqtools.library to your libs: assignment. Reqtools has also a prefs program, if you want it get the last version of the library from aminet (util/libs/reqtools*).

The installation of the socket library is needed only if you are using amitcp. Actually you DON'T need to install socket library if you use AS/I-Net225, Mlink or AmiTCP 4.3+. To install the library follow these steps (please note that if you have already Grapevine or Thor running you have already the socket.library installed):

makedir amitcp:libs (if the directory doesn't already exists)

copy amimud/libs/socket.library amitcp:libs/

echo >envarc:SOCKETCONFIG "UID=200 GID=200 USER=ggreco DOMAIN=tn.village.it UMASK=022"

echo >env:SOCKETCONFIG "UID=200 GID=200 USER=ggreco DOMAIN=tn.village.it UMASK=022"

(you have to substitute ggreco and tn.village.it with your login and your domain)

assign inet: amitcp: (better if you put this line also in your user-startup)

Please note that if you launch the program from CLI you can specify some parameters not (yet) available if launched from WB. See the Program usage section for more infos...

1.7 config,"configuration"

Configuration

When you launch AmiMUD the program will look for the file s:amimud.prefs to found the settings to use.

AmiMUD has many configuration parameters, so I suggest you to copy to s: the example configuration file and then edit it to your needs.

Configuration keywords

1.8 keywords"

Configuration keywords

The configuration keywords ARE case sensitive, must be written lowercase and without spaces, tabs or other characters before or after the keyword.

Example of correct use:

userprompt=name?

Examples of WRONG use:

userprompt=name?

Userprompt=name?

userprompt =name?

Remember also that you can place comments and blank lines everywhere, but you can't place a comment on the same line of a command.

The lines beginning with the character ";" are considered comments.

Available keywords:

General keywords

host

port

output

variable

command

history

crmode

verbose

Autologin keywords

user

userprompt

password

passwdprompt

afterpasswd

Ticks related keywords

ticks

mark

tickseconds

minimum_tl

prompt

beforetick

tickmacro

aftertick

Screen menagement keywords

customscreen

displayid

screenwidth

screenheight

screendepth

palette

ansi

Advanced features

macro

trigger

gag

gagline

highlight

1.9 keywords"

Keyword: variable

Usage

This character identify variables in trigger and macros. You must pay attention using a character you doesn't need to specify in the text of macro/triggers, otherwise you can have strange problems.

Example:

variable=%

(set the variable char to "%", like tinyfugue)

Default value: $

See also: variable character, keyword command, use of variables

1.10 keywords"

Keyword: command

Usage

Internal commands and macros to be recognised have to be preceded by a char, the command char. The default value of this character is "#" (so you can for instance call the command cycle with "#cycle"), but this can be changed if you prefer to use another character. The command character is recognised {b}only{/b} if specified at the beginning of a line, so you can give commands with the command character inside without problems (eg. gossip hello!#!#!#?). If you have to use the command character as the first character of the command you can simply substitute it with a double command (eg ##\n will send #\n to the mud).

Example:

command=/

(set the command/macro character to "/", like tinyfugue)

Default value: #

See also: use of command, keyword variable

1.11 keywords"

Keyword: host

Usage

This define the host to automatically call when amimud is launched. This option can be overridden if you specify the host to contact through command line or if you specify the NOCONNECT switch.

Probably you'll need also to specify a valid port because the large majority of the muds doesn't run on the default telnet port (23).

Example:

host=realms.community.net

See also: port

1.12 keywords"

Keyword: port

Usage

The port (tcp service) to connect to.

Usually MUDS don't run on the default telnet port (23), but on higher port numbers. So if you want AmiMUD be able to connect your favourite MUD you'll need to set this keyword to the right value.

This keyword is ignored if you specified a valid port number through command line. If no port is specified the default value of the port AmiMUD will try to connect to is 23.

Example:

port=7777

Default value: 23

See also: {"host" link host}

1.13 keywords"

Keyword: output

Usage

This option let you specify dimensions and title of the window in which will be displayed the texts coming from the mud. Actually this window is a standard amiga console, you can improve it for instance using KingCON that add a review buffer to the window and some useful menus for things like blocking the output (useful to examine an important text without problems dued to other texts coming from the MUD).

Example:

output=con:5/15/635/420/Output window/CLOSE

(For a 1:1 WB)

output=kcon:5/15/635/200/Output window/CLOSE

(If you use kcon without replacing original handler)

Default:

output=con:5/15/635/200/Output window/CLOSE

You can find KingCON on aminet (util/shell/kingcon13.lha).

See also:

1.14 keywords"

Keyword: history

Usage

This command let you choose how many commands will be remembered.

AmiMUD offers you an history buffer (you can access it with curs up and curs down keys), the default dimension of this buffer is 20 commands, after the 20th command AmiMUD will begin to delete the older ones. You can change this limit using this keyword.

Example:

history=50

(remembers 50 commands)

Default value: 20

See also:

1.15 keywords"

Keyword: crmode=(crlf|lf|cr)

Usage

This command let you choose the type of lines the mud send to you. Actually every mud uses the "crlf" mode (default value), so you probably will never need to use this keyword. Some strange telnet services may require to use "cr" or "lf" mode, I've included this feature for future telnet compatibility...

Example:

crmode=cr

(expects that carriage returns are sent as CR character)

Default value: crlf

See also:

1.16 keywords"

Keyword: verbose=(yes|no)

Usage

This keyword is very important. Many AmiMUD commands can create output messages on the output window, if verbose is disabled none of the not strictly needed output will be displayed. This can be useful for an expert user, but isn't suited for a beginner or if you are testing new trigger/macros...

Example:

verbose=no

(Disable verbose outputs)

Default value: yes

See also:

1.17 keywords"

Keyword: user

Usage

This is the first of the four keyword used to define autologin.

In user you have to specify the name of character you want to login automatically when you start AmiMUD.

IMPORTANT: To enable autologin you MUST specify user, password, userprompt and passwdprompt. If one of these is not specified in the configuration autologin will not work.

Example:

See userprompt

See also: userprompt password passwdprompt afterpasswd

1.18 keywords"

Keyword: userprompt

Usage

This keyword tell AmiMUD when it have to send the character name (user) to the mud. It must be a part of the string received from the mud after the banner.

Example:

userprompt=name?

(for shadowdale and similar muds)

See also: user

1.19 keywords"

Keyword: password

Usage

In password you have to specify the password of the character you specify in user keyword.

Example:

See userprompt

See also: user

1.20 keywords"

Keyword: passwdprompt

Usage

Here you have to specify the prompt received by the mud before typing the password.

Example:

passwdprompt=assword:

(for nearly every mud)

See userprompt

See also: user

1.21 keywords"

Keyword: afterpasswd

Usage

Usually after the password you'll need to send to the mud other text to complete the login operation. For instance on diku muds you'll need to confirm the MOTD (message of the day) with RETURN and to select option 1 from the menu. With afterpasswd you can do this automatically.

This feature is only used if autologin is enabled.

Example:

afterpasswd=1\n1\n

(afterpasswd for a dikumud)

See also: user

1.22 keywords"

Keyword: ticks=(yes|no)

Usage

This keyword is useful if you use AmiMUD to play LPMud or other mud types that don't use ticks.

Example:

ticks=no

(disable tickcounter)

Default value: ticks=yes

See also: Use of tickcounter

1.23 keywords"

Keyword: mark

Usage

If you want to use the tick counter properly and to see your prompt in the title of the input window you need to modify your prompt (or the mark) making AmiMUD able to detect it in the incoming text from the mud. If you can modify the prompt (in many mud you can do it with the command "prompt") you can simply use the default mark and modify your prompt as follow:

prompt ##[your old prompt]>

The prompt must begin with ## (or another mark you can configure with the mark keyword) and end with ">" (this cannot be changed). I made this choice because I see that the way TinyFugue recognises the prompt (a line without the final CR) has many troubles in lag condition...

This way is a bit more complex but works always :)

Example:

mark=**

(identify the prompt as the line that begin with "**" and ends with a ">")

Default value: mark=##

See also: Use of tickcounter

1.24 keywords"

Keyword: tickseconds

Usage

Use this if you know the length of the tick of the mud you are playing. For DIKU type muds it's usually 75seconds, for other mud types 90 and 120, this may not be true if the mud has many players online and then the ticks will be longer.

Example:

tickseconds=90

(set tick length to 90 seconds)

Default value: 75

See also: Use of tickcounter

1.25 keywords"

Keyword: minimum_ticklength

Usage

Set a minimum value to accept as tick length in the automatical tick calculation process, very useful if you have not defined triggers for heals, refresh, etc...

Example:

minimum_ticklength=20

(set the minimum accepted tick length to 20 seconds)

Default value: 60

See also: use of the tickcounter

1.26 keywords"

Keyword: beforetick

Usage

A simple macro to be executed before the tick, useful to recover mana and hp more quickly. Use it in conjunction with aftertick.

Example:

beforetick=rest\nsleep\n

(make your character rest and sleep 3-5 seconds before the tick)

Default value: not enabled

See also: use of the tickcounter

1.27 keywords"

Keyword: tickmacro

Usage

A macro to be executed when the tick occurs.

Example:

Default value: not enabled

See also: use of the tickcounter

1.28 keywords"

Keyword: aftertick

Usage

Simple macro that will be executed after the tick, very useful in conjunction with beforetick.

Example:

aftertick=wake\nstand\n

(make your character wake and stand up right after the tick)

Default value: not enabled

See also: beforetick use of the tickcounter

1.29 keywords"

Keyword: prompt

Usage

Very useful keyword. If you want to use the tick counter with the automatical tick length calculation you'll NEED to specify a correct value for this one. If the ticks of your mud are fixed length then you have to delete this keyword from your config (this will disable automatical tick calculation) and specify the right tick length.

Example:

if the prompt of the mud is:

##HP:3049 MA:2020 MV:111 (------) * -*>

You have to set:

prompt=HP:%ld MA:%ld MV:%ld

(recognise a prompt with 3 variables)

Default value: not enabled

See also: use of the tickcounter

1.30 keywords"

Keyword: customscreen

Usage

AmiMUD will open as default on Workbench screen, if you want to make it opens on a custom screen you need to specify this keyword. Note that ANSI mode is available only if amimud is opened on a custom screen (or on a public screen with ansi colors :)).

Example:

customscreen

(will popup a requester if the other screen parameters are not defined)

Default value: not enabled

See also: AmiMUD on custom screen

1.31 keywords"

Keyword: displayid

Usage

This keyword specify the displayid to use for the screen, if you want to specify it you'll need to specify it as a DECIMAL number. It is considered only if also screenwidth, screenheight and screendepth are correctly specified in the configuration.

Example:

Default value: not enabled

See also: AmiMUD on custom screen

1.32 keywords"

Keyword: screenwidth

Usage

The width of the custom screen to open. It is considered only if also displayid, screenheight and screendepth are correctly specified in the configuration.

Example:

screenwidth=640

Default value: not enabled

See also: AmiMUD on custom screen

1.33 keywords"

Keyword: screenheight

Usage

The height of the custom screen to open. It is considered only if also displayid, screenwidth and screendepth are correctly specified in the configuration.

Example:

screenheight=512

Default value: not enabled

See also: AmiMUD on custom screen

1.34 keywords"

Keyword: screendepth

Usage

The depth of the custom screen to open. It is considered only if also displayid, screenheight and screenwidth are correctly specified in the configuration.

Example:

screendepth=3

(the minimum depth for for ANSI mode)

Default value: not enabled

See also: AmiMUD on custom screen

1.35 keywords"

Keyword: ansi

Usage

If you specify this keyword (and you open AmiMUD on a custom screen) you'll be able to see muds in ANSI colors (if the mud you are playing supports them).

Example:

ansi

(this will enable ansi colors)

Default value: Not enabled

See also: palette

1.36 keywords"

Keyword: palette

Usage

If you want to use a particular palette with AmiMUD you can. You can edit it with the menu option Prefs->Edit palette.. and then save it to a suitable file (for example s:amimud.palette) and then add the keyword palette=s:amimud.palette to your configuration. Please note that the palette keyword works only if AmiMUD is opened on a custom screen.

Important: This feature works only with OS 3.0+

Example:

palette=work:pictures/mypic.iff

(load in the amimud screen the palette of mypic.iff)

Default value: Not enabled

See also: Open AmiMUD on a custom screen

1.37 keywords"

Keyword: macro

Usage

Macro are very useful in muds, expecially if you need to do things very quickly... A macro is a way to send many commands to the mud in a quick way. Macros can be defined in the configuration file, with the command #macro or through the menus. You can also see a list of all the defined macros with the command #list or through menus.

Macro may require arguments or not and the macro text may contain variables. Actually if you want your macro have two parameters you have to add after the macro name $0 $1. Actually macro parameters MUST be in crescent order 0..4.

macro=givefb $0 $1

rem bag\nget $0 bag\nwear bag\ngive $0 to $1\n

This is a correct definition for macro with arguments, the following one instead will NOT work:

macro=givefb $1 $2

rem bag\nget $1 bag\nwear bag\ngive $1 to $2\n

A macro with parameters cannot be launched through hotkeys.

Macro can also use variables. The only difference between a macro with parameters and a macro that contains variables is that the second can be launched with an hotkey. You can set the variables with triggers or with the #set command. See below for some examples.

Hotkeys:

The macro hotkeys use the standard Amiga hotkey format (chapter 10-25 of OS 3.1 Workbench manual) here is a short summary:

alt, lalt, ralt - alt keys

shift, lshift, rshift - shift keys

lamiga, ramiga - amiga keys

ctrl, control - control key

leftbutton, middlebutton, rightbutton - mouse buttons

F1..F10 - function keys

Examples of hotkeys:

hotkey=shift F1

hotkey=lamiga ctrl 5

hotkey=leftbutton F1

Important: The macro text must always end with a carriage return (\n) otherwise the macro execution will not be confirmed...

Warning the hotkeys defined in AmiMUD are GLOBALS, this means that if you assign F1 to a macro and you press F1 the macro will be executed also if the amimud window/screen is not the active one! This is dued to the way commodities handles hotkeys, if you need an hotkey you can simply disable AmiMUD commodity interface with Exchange or similar programs.

Example:

(These examples are thought for diku-like muds)

macro=hungry

hotkey=lamiga h

rem bag\nget bread bag\neat bread\nwear bag\n

(Macro without parameters, useful if your character is hungry (may be activated also with the hotkey amiga+h...)

macro=getbag $0

rem bag\nget $0 bag\nwear bag\n

(Get an object from the bag)

macro=hun&thir

#hungry\n#thirsty\n

(You can also call macros in a macro)

macro=disint

hotkey=alt 3

cast 'disintergrate' $7\n

macro=meteor

hotkey=alt 2

cast 'meteor swarm' $7\n

macro=dispel

hotkey=alt 1

cast 'dispel magic' $7\n

These macros are targeted, using the command #set you can specify what does the variable $7 contents (example: #set 7 dracolich) and then you can call them with hotkey or typing the macro name without the need to specify the target each time.

Examples of macro defined through the command macro:

#macro "hungry" "rem bag\nget bread bag\neat bread\nwear bag\n"

(a macro defined with this command can NOT have hotkeys).

See also: Add a macro, Advanced use

1.38 keywords"

Keyword: trigger

Usage

Triggers are the most powerful feature of a mud client over a normal telnet connection. Triggers may make automatic many actions that have to been performed quickly or that have to be done very often.

Triggers may contains variables, this can be useful in two different ways.

The first (and obvious) way is to use part of the text that activate the trigger in the trigger body, for instance to split some coins through the group members (see examples below). If you put a variable in the trigger definition (obviously with some other text, otherwise ANY line willl match that trigger) if the trigger is found the variable will contain the text between the words that make the trigger match.

trigger=A $1 C

#echo trigger found, $1 is between A and C\n

if the mud sends to the player "A B C" the trigger will be activated and the phrase:

trigger found, B is between A and C

Will be printed in the output window.

The second way to use triggers is an alternative to #set. You can set variables with a trigger without using them. For instance if you need (on a diku-type mud) to know who is the group leader and keep the value in a variable for future use you can make a trigger to get the name from the text the mud sends in some cases:

trigger=now member of $7's group

#echo Put groupleader name ($7) in variable 7\n

Remember that macros and trigger must always end with a carriage return (\n) otherwise the text will be send to the mud, but not confirmed!

Example:

trigger=ou are hungry

#hungry\n

(See the section macro to see how #hungry is defined.)

trigger=as you hear $0m$1's

get all.coins $1\n

trigger=as you hear $1's

get all.coins $1\n

(These two macros make the same thing, the first works if you are playing in ansi mode, the second in plain mode, please note the use of the $0 variable to strip the ansi code)

trigger=here were $1 coins

split $1\n

(This macro split the coins you get from a corpse to the members of your group)

See also: Add a macro, Advanced use

1.39 keywords"

Keyword: gag

Usage

This keyword let you define some text you want to be displayed. It can be useful to make output of some commands cleaner.

Example:

gag=is in excellent condition.

(Will be useful to make the output of command "equip" on diku-mud shorter)

See also: gagline, highlight

1.40 keywords"

Keyword: gagline

Usage

With this command you can tell amimud to not display a line of text. This is useful for instance in fightning with many fighters. You can disable gagline you have defined or add a new gagline in any moment through menus.

Example:

gagline=misses you

(You will see only the hit of enemies if they hit you)

See also: gag, highlight

1.41 keywords"

Keyword: highlight

Usage

Highlight may be used to make some text be highlighted (:-)). This can be useful in many cases. Highlights can be inserted in the configuration file or through the menu option, you can enable/disable them through menus or with the command #high.

These are the actually defined highlights:

BOLD 0

UNDERLINE 1

ITALICS 2

REVERSE 3

BOLD_UNDERLINE 4

CLEAR_SCREEN 5

COLOUR_0 10

COLOUR_1 11

COLOUR_2 12

COLOUR_3 13

COLOUR_4 14

COLOUR_5 15

COLOUR_6 16

COLOUR_7 17

Example:

highlight=0

extemely well

highlight=1

massacres

highlight=4

devastates

(these three gags will highlight in different ways different hits)

Default value:

See also:

1.42 on

Open AmiMUD on a custom screen

There are two ways to make AmiMUD open a custom screen.

1 - Using the CS or CUSTOMSCREEN command line switch if you launch AmiMUD from a shell.

2 - Specifying customscreen keyword in the configuration file.

These two option will pop up a screenmode requester, if you use AmiMUD always on the same screen you may like to select the mode once for all. You can do this in two ways:

1 - (Easy way) Select the mode as usual in the screenmode requester then save the preferences of AmiMUD with the "Save Prefs" menu option.

2 - (Tricky way) Edit displayid, screenwidth, screenheight, screendepth. These FOUR parameters (and customscreen) have to be specified if you want to skip the screenmode requester. Optionally you can also specify ansi (that init the screen with the default ANSI palette) or palette that let you load an IFF palette (also from a picture) in the screen.

1.43 the

Using the program

Start AmiMUD

Connect to a MUD

Log to file

Use the numeric keyboard to move

Add a macro or a trigger

Use of gags and highlights

Send text to the MUD

Enable/Disable triggers/macros...

Use of the tick counter

Advanced use of triggers/macros...

Run multiple AmiMUDs

Quit the program

1.44 connections..."

Multiple connections?

Often mud players like to play more than one mud at once, so unix clients often supports multiple connections, some guys that try the beta of AmiMUD asks me if I will implement multiple connections on future versions, I don't think I'll do it because actually is already possible to run multiple copies of AmiMUD, also on the same screen. Follows some suggestions to do "multiple connections" with AmiMUD (remember that the program executable is very little, so running multiple copies of the program is not a waste of memory).

1) Run many amimud (on different MUDs) on the WB screen:

Simply make two configurations with the right host,port,name,password... obviously without the key customscreen and run amimud from shell with the following parameters:

run amimud CFG mud1.prefs

run amimud CFG mud2.prefs

You will have two connections opened on wb :)

2) Run many amimud (on different MUDS) on the same customscreen:

Put customscreen in the first configuration (and ansi if you want) and remove customscreen from the other configurations then run AmiMUD:

run AmiMUD cfg mud1.prefs

run AmiMUD cfg mud2.prefs SCREEN AmiMUD.1

run AmiMUD cfg mud3.prefs SCREEN AmiMUD.1

[...]

1.45 connect,"connecting..."

Connect to a MUD

To connect a mud you usually need to specify TWO parameters.

Address:

May be an IP (192.106.166.6) or a mnemonic name (mclmud.mclink.it). The first one works also if you don't configure properly the DNS (name server) on your tcp stack. You have to write the address calling the menu item "Set Address..." in the menu "File" if you have not already defined them in the configuration file.

Port:

Usually muds don't run on the default telnet port (23). So if you want to connect a mud generally you have to specify also the port number the mud runs on. To do this use the "Set Port..." menu item in the menu "File". The port may also be specified in the configuration file.

Once you have specified the address and the port and obviously you have already the TCP stack running you can connect to the mud. Use the "Connect" menu item or the shortcut RightAMIGA+C.

The "tick" near the "Connect" menu item means you are connected.

1.46 of

Keymap Movement

One intresting feature of AmiMUD is the "keymap movement", AmiMUD remaps the classic cardinal directions on the numeric keypad, so you can move using it, without need of hitting return for each keystroke...

This is how the directions are remapped:

7 8 9 -

nw n ne d

4 5 6 +

w exits e u

1 2 3

sw s se

Actually this is a fixed setup, in the future I may insert some editing on this. Please note that this may not work on every mud (expecially diagonal directions).

I suggest everyone to use keypad to move it's very quickly and handy if you use it often.

1.47 character"

Variable character

The variable char is a character that identify a variable inside a trigger or a macro, so you have to use a character you don't need in the text, otherwise you will get an error if you will use it.

Dafault value: $

Example:

(in the configuration file)

variable=%

macro=hello %0

gossip Hello %0!\n

(This will change the variable char from '$' to '%', like tinyfugue)

1.48 character"

Command character

Actually all AmiMUD internal commands and all macros are detected through a special character at the beginning of the line. The default command character is '#' but may be changed through the config keyword command.

1.49 internal

Command: beep

Usage

If you call beep the screen will blink and/or a sample will be sound (it depends from your system preferences). This command may be very useful to make a trigger that call you if you are away from keyboard and someone wants to talk to you.

Example:

(in the configuration file if your name is Exodus)

trigger=xodus

#beep\n

(will perform a display beep if someone on the mud gossips says or tell "xodus" the first letter of the name is not specified to avoid lower/upper case problems).

1.50 internal

Command: echo

Usage

-----

echo is a small and very useful command for complex triggers. It simply prints to the output window the contents of the line after his invocation. This sometimes may not be very useful, but often can be very handy (expecially if verbose=no in your configuration file!).

Example:

#echo hello

(will print hello in the output window)

#echo $1

(will print the contents of variable "1" to the output window, a bit more useful than the previous example)

1.51 internal

Command: cycle

Usage

The cycle command let you repeat <times> times a macro or some commands. It can be used also into a trigger.

Syntax: #cycle <times> <macro or text>

Examples:

#cycle 5 buy bread\n

Buys 5 peaces of bread

#cycle 20 kick $9\n

Kicks 20 times the guy in variable $9

#cycle 2 #eat\n

Executes 2 times the macro #eat.

1.52 variables"

Variables

Actually AmiMUD supports variables in a limited way. You can use variables from 0 to 9. These are used by triggers/macros and all the other commands. A variable is identified by the variable char (default '$') followed by a number from 0 to 9. The maximum length of the text in a variable is 30 characters. A variable is keep in memory until it's replaced by another value, this can be done through macros, triggers or the #set command.

See also: Advanced use of variables, Variable char, variable keyword

1.53 internal

Commands

All these command may be used directly through the input window or in triggers and macros. All the commands must be used with the following syntax:

#commandname <arguments>

action

beep

clip

cycle

echo

file

gag

gagline

help

high

list

log

macro

send

set

skiptick

stoplog

sync

trigger

wait

1.54 internal

Command: wait

Usage

Syntax: #wait <seconds*50>

The wait command can be used in triggers to delay a command, it accept as parameter a number identifying the 1/50 of seconds to wait, it practically passes this number to the amigados function Delay(). While waiting both output and input will be freezed, so don't wait for very long periods :)

1.55 internal

Command: action

Usage

A trigger can be added through the config keyword trigger, through menus or through this command. If you use #action to define a trigger you CAN'T define hotkeys for triggers...

Example:

#action "ou feel less protected." "cast 'armor' exodus\n"

If the magic shield of your character fades cast on yourself another one.

1.56 internal

Command: help

Usage

This command list all available internal commands and provides a brief description of their usage. Call it without arguments.

1.57 internal

Command: set

Usage

Syntax: #set <var num> <value>

This command assign the text in <value> (until the end of the line) to the variable <var num>. It can be very useful for targeted macros.

See also: AmiMUD variables, Macros

Example:

#set 7 beholder

1.58 internal

Command: list

Usage

This command lists all defined macros. You can also see them through the menu option "Macro List...". It doesn't require arguments.

1.59 to

Log to file

If you are exploring an unknown area or you are trying an heroic mission you probably want to store what you are doing on disk for future use.

AmiMUD let you log to file through two simple commands or a menu item.

#log <filename>

Will start to log all the output and everything you send to the mud to the file <filename>. You can stop the operation with the command: #stoplog.

Otherwise you can use the menu item "Log to file..." in the menu "File". When you select it for the first time you will be prompted with a file requester to select the file to write to. To end the log operation you have to select the "Log to file..." item another time.

If there is a "tick" near to the "Log to file..." text it means you are logging. You can also activate/deactivate logging with the shortcut rightAMIGA + L.

Note: If you are using ANSI your log will contain ANSI codes, I suggest using stripansi.lzh (on aminet) to remove ansi codes from log files.

1.60 a

Macros & Triggers

Macro and triggers can be added in three ways. The first (and simpler) is in the configuration file, with the keyword macro and trigger, otherwise you can define them online with the #macro and #action commands or also through the menu options "New Macro..." and "New Trigger...", remember anyway that with the macro command you can't define macro hotkeys while with the config keyword or through menu you can. Usually is better to define macro and trigger in the configuration, so that you don't need to save the configuration to keep them for future use.

1.61 use

Advanced use of macros and triggers

Macros and triggers are very powerful if used properly. Actually the only limitation they have is the not yet supported command #if (probably it will be in the next version) and the maximum size of 159 characters. This second problem can be easly avoided making a very long macro as the concatenation of different macros:

```
macro=verylong
#part1\n#part2\n#part3\n
macro=part1
[160 chars available...]
macro=part2
[160 chars available...]
macro=part3
[160 chars available...]
```

Note that if you put more than 160 characters on a macro definition it will be cutted, but amimud will not complain about it...

One use of macros very intresting expecially for combat oriented muds is the targeting. This let you perform the commands very quickly, without the risk of typing errors caused by a too fast typing on the keyboard. Follow an example of targeted macros for diku-type muds:

```
;This macro set the target and the tank and tell it to the group
macro=target $0 $1
#set 8 $0\n#set 9 $1\ngtell Ok, the tank is $0 and the target is $1\n
; Attack macros (target is $9) note that all macros are available with
; the single keypress of a function key. This can be done also defining these macros
; as the first ten in the config file WITHOUT specifying an hotkey.
macro=disint
hotkey=F1
cast 'disint' $9\n
macro=meteor
hotkey=F2
cast 'meteor' $9\n
macro=dispel
```

hotkey=F3

cast ’dispel’ $9\n

macro=slow

hotkey=F4

cast ’slow’ $9\n

; Defence macros (target is $8)

macro=heal

hotkey=F6

cast ’heal’ $8\n

macro=sanc

hotkey=F7

cast ’sanc’ $8\n

macro=armor

hotkey=F8

cast ’armor’ $8\n

1.62 a

Add a gag or an highlight

A gag (word or line) and an highlight may be added through the config keyword gag, gagline and highlight or through the menu options Prefs->New Gag->(Line...|Word...) and Prefs->New HighLight... A gag simply cut the specified part of text from the incoming text from the mud, a gagline cut the entire line if it found the specified text on it (so a "gagline=e" will cut every line containing the letter e, practically every line!). An highlight perform some modifications in the style or in the colors of the specified text, look at highlight keyword documentation to see what styles/colors are actually available.

1.63 internal

Command: send

Usage

This command is very useful to test triggers/highlights/gags without need to be connected (you only need to start amitcp/inet-225, I’m not sure it can be done with other TCP stacks). If you type the command "#send hello world!" the program will think it has received "hello world!" from the mud, then if you have some trigger/highlight/gag on these words they will be performed.

Note: the #send command to be effective must be performed in verbose mode.

Example:

#action "hello world!" "shout I hate the people shouting hello world!"

#send "Ehi, hello world!"

this will make the previously defined trigger being activated, obviously if you are connected the shout command will be sent to the mud, otherwise only to the output window (if you are in verbose mode).

See Also: Triggers

1.64 text

Send text to the mud

Sometimes may be useful to send some text previously written to the mud. This can be done in two ways in AmiMUD.

1) Clipboard - You can edit the text in a text editor or word processor or also a shell then copy it to the clipboard usualy through the combination right amiga+c, then you can use the #clip command (without parameters) or the "Send Clipboard" menu option (in Actions).

2) File - You can edit the text with your favourite text editor/wp and then save it to a file in plain ASCII text mode, then with the command #file <filename> or the "Send File..." menu option (in Actions) you can send this file to the mud.

Note that in both cases the text will be send also to the output window ONLY if the verbose mode is active.

1.65 the

Quit the program

Clicking on the input window close gadget or on the quit menu option will make the program quit (after a requester asking for confirm the operation) and close the socket. Remember that this don't perform the mud quit procedure (you may need to go to an inn or at least to perform a save command) then your character will be classified as link dead. So to be sure all it's ok you have to wait the message "connection closed by foreign host" before quitting the client.

1.66 Trigger/Highlight/Gags..."

Disabling Trigger/Highlight/Gags...

In some cases may be useful disabling trigger/highlights or gags (expecially trigger during fightings...) so AmiMUD has some commands to do it and also some menu options...

All these commands are enabled by default.

Commands:

#trigger on

Enable triggers

#trigger off

Disable triggers

#high on

Enable text highlights

#high off

Disable text highlights

#gag on

Enable gags

#gag off

Disable gags

#gagline on

Enable gaglines

#gagline off

Disable gaglines

There are also in the preferences menu options to do the same thing.

1.67 of

Tick Counter

The tick counter is one of the more impressive (and complex) features of the client. It's very simple to use if your mud as a fixed length tick. In this case you have only to configure the tickseconds keyword to the right value (default for diku is 75 seconds). Probably you'll need also to perform a #sync call to syncronize the tick counter with the real ticks. You can also make a simple trigger that catch the messages that mud sometimes send at the tick (The day has begun, The moon is going down...) to perform the #sync command automatically.

If your mud as a very high load (many users) or is of another type than diku the ticks may have a NON fixed length. With non fixed length the use of the tick counter is a little more difficult:

You'll need to configure your prompt with a mark at the beginning (default ##, but may be changed with the keyword mark) and the final character '>' (this is fixed). In a diku type mud you can do this with the following command:

prompt ##H:%h M:%m V:%v XP:%x %c -%C>

This example prompt uses the standard mark so you don't need to use the keyword in the configuration.

You'll need also to tell the program where there are movement, hitpoint and mana in the prompt string. This is done with the prompt keyword.

This because AmiMUD recognise the tick making a comparison of the values of the first TWO or THREE numeric values of the prompt, this detection is automatical, it does depends on how many %ld the program finds in the prompt definition)

It is important that the two or three number you want to use are the first two or three you specify in the prompt. I don't think that it's a big limitation :)

Example:

##H:559 M:200 V:100 bla bla bla >

This is how your prompt should look with the previous configuration, the important is that the first two or three parameters are the ones that RAISE at ticks. With this example prompt you should configure the prompt keyword as follow:

prompt=H:%ld M:%ld

If you want to use only mana and hp for detecting ticks or:

prompt=H:%ld M:%ld V:%ld

If you want to use also movement.

Obviously this will make the tick counter fail if a character is healed or refreshed, in the example configuration there are some triggers that using the #skiptick command let you avoid this. Basically the #skiptick command tell the mud that the next increase of HP or Mana or Movement as to be ignored.

1.68 Launch the program

Start AmiMUD

The program can be launched from CLI or from workbench if you launch it from WB you loose the possibilty to specify some option available only through command line.

Command line options

This is the AmiMUD argument pattern:

HOST,PORT,CFG=CONFIG/K,SC=SCREEN/K,NOC=NOCONNECT/S,NC=NOCONFIG/S,

CS=CUSTOMSCREEN/S,REQ=REQUESTER/S

HOST, PORT - These are the host and port to connect to. If you specify these you override the prefs settings (if present).

Example: AmiMUD realms.community.net 7777

CONFIG filename - This potion make AmiMUD load "filename" as configuration instead of the default "s:amimud.prefs", useful if you use multiple characters with different configs.

SCREEN - Name of the public screen where to open AmigaMUD

NOCONNECT - If autologin is actived in the prefs file this option will disable it.

NOCONFIG - Doesn't load the prefs file. As if s:amimud.prefs doesn't exists.

CUSTOMSCREEN - Force AmiMUD on a custom screen, if this option is specified AmiMUD will pop up a screen mode requester to choose the screenmode you want to use.

REQUESTER - Open a file requester to choose the name of the prefs file to load, useful if you own many different configurations :)

1.69 hist,"history"

History

1.0 - First public release

1.70 author,"author"

Author

You can contact me at the following addresses:

Internet: ggreco@tn.village.it

Fidonet: 2:332/235.9

Address:

Gabriele Greco

Via Banchi 12

16030 Uscio (GE)

Italy

You can also found me on the following italian mud:

Lumen et Umbra: mclmud.mclink.it 6000

Colosso/Exodus/Mongo

1.71 AmiMUD..."

Registering AmiMUD

AmiMUD is shareware, if you use the program after 1 month of testing you HAVE to register. If you register the program you will receive through e-mail or snail mail a keyfile that will remove the opening requester, you will also receive an e-mail notifying if new versions are available.

Registration fee:

US dollars 15

DM 20

Italian lire 20000

UK pounds 10

French Francs 60

You can send me the money enclosed in an envelope or you can use an international money order. For italians the best choice is a "vaglia postale".

If you send the money in the envelope please send me that in one of the currencies I listed...

Send the cash/postal money order to my address, I will respond as soon as possible.

In the envelope enclose at least you e-mail address or your postal address (e-mail preferred).

I accept also as "registation fee" the registered version of a shareware program YOU write.

Thank you for supporting shareware concept!

1.72 to

Future features

- Arexx port (will make possible writing the most powerful triggers of the world :-))

- Full telnet support (actually doesn't work with host that uses character mode).

- Possibility to use AmiMUD as a terminal program with serial.device and similar devices.

- Localization (if you want to write a catalog send me an e-mail).

- Alphanumeric variables support (e.g. $tank)

- Pen control preferences for custom screen mode with strange palettes (like Term).

If you have an idea you'll like to see implemented in AmiMUD mail it to me.

1.73 to..."

Thanks

I would like to thank the following persons for suggestions/bug fix/betatesting:

- Riccardo Zironi

- Stuart Logan

- Agostino Fanti
- Blitter
- Max Gargani
- Gianluca Porta

And also:

Emmanuele Benedetti

Giuseppe Caggese

Because they create Lumen et Umbra the mud that make me decide to write an Amiga mud client...