NAME

gma mapper – GMA battle grid map

SYNOPSIS

`gma mapper -- --help`

`gma mapper` [`-display` *name*] [`-geometry` *value*] [*other wish options*] `--` [`-A`] [`-a`] [`-B`] [`-b` *pct*] [`-C` *file*] [`-D`] [`-d`] [`-f` *fmt*] [`-G` *n*[+*x*[`:`*y*]]] [`-g` *n*[+*x*[`:`*y*]]] [`-h` *hostname*] [`-k`] [`-L`] [`-l`] [`-M` *module*] [`-n`] [`-P` *pass*] [`-p` *port*] [`-S` *profile*] [`-t` *transcript*] [`-u` *name*] [`-X` *proxyhost*] [`-x` *proxyurl*] [`--button-size small`|`medium`|`large`] [`--chat-history` *n*] [`--curl-path` *path*] [`--curl-url-base` *url*] [`--dark`] [`--debug-protocol`] [`--mkdir-path` *path*] [`--nc-path` *path*] [`--no-animate`] [`--no-blur-all`] [`--no-char`] [`--no-dice`] [`--preferences` *path*] [`--scp-dest` *dir*] [`--scp-path` *path*] [`--scp-server` *hostname*] [`--ssh-path` *path*] [`--update-url` *url*]

`gma mapper` [`-display` *name*] [`-geometry` *value*] [*other wish options*] `--` [`--`[`no-`]`animate`] [`--`[`no-`]`blur-all`] [`--blur-hp` *pct*] [`--button-size small`|`medium`|`large`] [`--config` *file*] [`--chat-history` *n*] [`--curl-path` *path*] [`--curl-url-base` *url*] [`--dark`] [`--debug`] [`--debug-protocol`] [`--guide` *n*[+*x*[`:`*y*]]] [`--host` *hostname*] [`--image-format` *fmt*] [`--keep-tools`] [`--list-profiles`] [`--major` *n*[+*x*[`:`*y*]]] [`--mkdir-path` *path*] [`--module` *module*] [`--nc-path` *path*] [`--no-char`] [`--no-chat`] [`--no-dice`] [`--password` *pass*] [`--port` *port*] [`--preferences` *path*] [`--preload`] [`--proxy-host` *proxyhost*] [`--proxy-url` *proxyurl*] [`--scp-dest` *path*] [`--scp-path` *path*] [`--scp-server` *hostname*] [`--select` *profile*] [`--ssh-path` *path*] [`--transcript` *filename*] [`--update-url` *url*] [`--username` *name*]

Options cannot be combined into a single CLI argument. They are evaluated in the order they are received.

As shown above, if you have the GMA Core package installed, you may invoke the `mapper` program via the main `gma` program as

```
gma mapper ...
```

If you installed `mapper` as a stand-alone package by itself, you need to invoke it by running the `mapper.tcl` program found in the `bin` directory of the `GMA-Mapper` package as unpacked or cloned from git. Depending on your system's particular configuration, you may be able to run it directly, as

```
bin/mapper.tcl
```

or you may need to explicitly run the `wish`(1) interpreter in a form such as

```
wish bin/mapper.tcl
wish8.6 bin/mapper.tcl
```

DESCRIPTION

The `mapper` program displays a battle grid map for the players to see their tactical positions with respect to their opponents and features of their environment (e.g., dungeon rooms).

If one or more map file names are listed on the command line, those files are loaded into the grid display. Otherwise, a blank map grid is shown. Map files may subsequently be loaded via interactive commands or upon request from a control service (see the `-h` option.

See notes at the bottom of this manpage for installation requirements.

CONFIGURATION

Starting with version 4.4, the mapper now uses an integrated preferences editor for its configuration. This eliminates the necessity of controlling the mapper via command-line options and/or configuration files, although those may still also be used.

Selecting `Edit > Preferences...` from the menu calls up the preference editor window. There you can enter global preference settings as well as server-specific ones. In the latter case, multiple server profiles may be created. Selecting one of these profiles in the list will make it the current profile. Every time the mapper is started, that server profile will be loaded and a connection opened to the server named in it.

If you wish to simply select a different server profile as the current one and connect to it, access the `Play > Connect to` menu and choose the desired server profile.

Some preferences may not take effect until the mapper is restarted. When leaving the editor window, you will be given the opportunity to restart the mapper if you desire.

Once the editor window is used, a preferences data file will be created as `~/.gma/mapper/preferences.json`. If that file exists, it will always be loaded when the mapper starts. In this case, the mapper will not load the `~/.gma/mapper/mapper.conf` file anymore unless explicitly requested via command-line option.

If any command-line options are given, they will override the settings loaded at program start-up from the `preferences.json` file.

HELPER PROGRAMS

The `mapper` client makes use of a few other programs to facilitate the efficient handling of data transfers. These are only used if the client is working in networked mode rather than as a stand-alone map drawing program.

`curl` Assuming the GM (or whomever is administrating the map assets) has arranged for a web server to host content for the map, the `mapper` client will invoke the `curl`(1) program to actually retrieve the files from that server. If needed, you will need to provide options and/or configuration file parameters as documented below to let `mapper` know where to find the `curl` binary and what options it needs to find the web server, navigate through proxy servers, etc.

`scp` For users with privileges to *store* data on the web server (generally, this would be the GM or other designated administrative users, not the general player group), `mapper` will invoke `scp`(1) to copy new data files up to the server. For example, in store-and-forward mode, when opening a local data file or pushing map contents out to other clients, `mapper` checks to see if the server has the data file(s) available for everyone to download. If not, it will use `scp` to upload them to the server before sending out a command to peer clients to retrieve them.

This requires that the user is authorized to perform this action on the remote server. Usually this is done by having an SSH certificate loaded via `ssh-agent`(1) and the server set up to recognize that for the needed operations without requiring a password to be manually entered.

`ssh` Along with the use of `scp` to transfer new data to the web server, `mapper` will use `ssh`(1) to create directories and set file permissions as needed on the server for the content being uploaded.

OPTIONS

The following options control the behavior of `mapper`.

`-a`, `--animate` This option causes the mapper to make a lame attempt to "animate" the appearance of objects being drawn onto the screen by updating the display after each one. (The author was feeling oddly nostalgic about working with the Tektronix storage-screen computer displays of his childhood at the time. If you don't know what that means, go watch an episode of the 1970s-era Battlestar Galactica and look at most of their computer screens.)

`-A`, `--no-animate` Suppress the animation effects enabled by the `-a` (`--animate`) option. This is the default, so it only needs to be given to cancel an `-a` option which appeared previously or which was read from a configuration file.

`-B`, `--blur-all` Normally, the imprecision introduced to health bar displays by the `-b` (`--blur-hp`) option applies only to "monsters" (opponents). With this option, it applies to all participants.

`--no-blur-all` Cancels the effect of the `-B` (`--blur-all`) option.

`-b`, `--blur-hp` *pct* This option "blurs" the health bar displays by rounding off the displayed hit point total to only be accurate in *pct*-percent intervals. For example, a setting of `10` means the health bar will blur the value by 10%; in other words, rather than every hit point showing proportionally on the health bar, the health bar will only show

10 possible intermediate values, corresponding to the hit points being 1–9%, 10–19%, 20–29%, ..., 90–99% of the total, as well as 0% and 100%. Thus, higher *pct* values indicate less accurate displays.

Setting *pct* to 0 (or less) indicates that no blurring is desired; in this case the display is precisely accurate. This is the default, but note that the hit points reported may be blurred on the server (GM)'s side independently.

Once a creature reaches 0 hit points, no further blurring is done, so the bleed-out sequence is accurate (but this is fairly quick for almost all creatures and is of less consequence than the hit point totals while they are still alive and fighting, so this was considered a better course of action).

`--button-size` *size*

Change the size of the toolbar buttons. The *size* value may be any string starting with `s`, `m`, or `l`, representing small, medium, or large-size icons. Small buttons are the default.

`-C`, `--config` *file* Read a set of command-line options from the named *file* as if they appeared at this point in the list of command-line options. Only the long-form option names are allowed and are given without the leading hyphens. The file must contain a single option per line. Options which take a parameter are separated from their parameter with an equals sign (although this is currently not supported in the command line itself). For example, a configuration file might contain the following:

```
# My configuration settings
scp-dest=/usr/local/game-support
scp-server=www.example.org
curl-url-base=https://www.example.org/game
no-animate
keep-tools
```

Note that any line whose very first character is an octothorpe ("`#`") is ignored as a comment.

If the file `~/.gma/mapper/mapper.conf` exists, it is read first before command-line options or (other) configuration files are loaded.

Note that more than one `--config` (and/or `-C`) option may be given, in which case the files are read in the order they appear in the command line. This may be used to split up options into different files, such as general settings common to all sessions, and specific settings based on networks or different games.

`--chat-history` *n*

Limits the retained chat history to *n* messages. When `mapper` starts, it reloads the chat history it had cached from the previous session on the current *host* and *port* but if that results in more than *n* messages being in the history, the list of messages is truncated to the most recent *n* (both in-memory and in the cache file). Any additional messages received will be kept, so the actual history will be a little larger than *n* until the next time `mapper` is started. If *n* is less than or equal to 0, then no limit is placed on the history size. The default limit is 512.

`--curl-path` *path* Specify the path to the `curl`(1) program on your system, if the built-in default doesn't work for you. This is used when fetching image and map files from the server.

`--curl-url-base` *url*

Specify the base URL on the data server. The files downloaded by mapper clients will be in a directory hierarchy appended to this string.

`-d`, `--dark` This option changes the default color palette to use a darker background which may be easier to look at for longer periods of time. On macOS systems running

up-to-date versions of Tcl/Tk (not the default legacy version supplied by Apple), dark mode is automatically selected if the macOS session is also configured for dark mode.

`-D`, `--debug` Increase debugging output level. Multiple `-D` options further increase verbosity of debugging messages. These are displayed in a separate window.

`--debug-protocol`
Turns on debugging messages showing all interactions between client and server, and the client-side procesesing that is done to support that interaction.

`-g`, `--guide` *n* Make every *n*th gridline red (thick lines). This is for minor guide lines. The value of *n* may be specified in all the same ways as for the `-G`/`--major` option (see below).

`--help` Print a summary of the command invocation options and exit.

`-h`, `--host` *hostname*
Connect to a map control service running on the designated host. This will send updates to item positions, display of rooms, etc. If this option is not specified, no control connection is made, and the mapper runs in stand-alone mode.

`-f`, `--image-format gif|png`
Sets the preferred image file format to request when retrieving images from the server to display on the map.

`-k`, `--keep-tools` Normally, map clients have their toolbars turned off to maximize the available screen space for the battle map. The GM can turn on and off their toolbars from his console as needed. If this option is given, this causes the client to unconditionally display its toolbar anyway. This is used for the main map run by the GM or whomever else is managing the group map and needs the toolbar active, or if people just want to keep the toolbar all the time.

`-L`, `--list-profiles`
List all the available profiles in the current preferences settings (either the default set for the mapper client or the one specified with the `--preferences` option). This list is printed to the standard output and then the mapper exits. These are the profile names which may be given to the `--select` option.

`-G`, `--major` *n* Make every *n*th gridline green (very thick lines). This is for major guide lines on the map.

`-G`, `--major` *n*+*o* As above, but offset the major guide lines to the right and down by *o* lines. The + character is required, but the value of *o* may be negative, so the option "`-G` *n*`+-3`" would move the lines to the left and up by 3 lines.

`-G`, `--major` *n*+*x*`:`*y* If expressed this way, rather than use the same offset in both directions, move the guide lines *x* lines to the right and *y* lines down.

`--mkdir-path` *path*
Specify the *server-side* path to the `mkdir`(1) program which will be used when uploading files *to* the data server (authorized users only).

`-M`, `--module` *module*
Use the module ID code *module* for this session. This is used to differentiate server-side resources between campaigns which have conflicting names. This is only needed for the mapper clients used as the forwarder in store-and-forward mode (typically the GM's own client).

`--nc-path` *path* Specify the path to the `nc`(1) program which will be used when sending files *to* the data server (authorized users only) through a SOCKS proxy server.

`--no-char` Don't prompt the user to say what character they're playing if they log in as a user name that doesn't match any known character on the map. This leaves no known mapping between the player and the characters, which may limit effective use of the mapper tools.

`-n`, `--no-chat` Suppress the display of incoming chat messages including die rolls.

`--no-dice` Disable the ability to send chat messages and roll dice. The mapper's die roller window will only receive those messages.

`-P`, `--password` *pass*
For servers which require authentication, this specifies the password to gain entry to that server. This is a fairly simple authentication mechanism intended to block nuisance connections, spam, and accidental connections of legitimate clients to the wrong game server. If *pass* is given as a single question mark ("`?`"), then the user will be prompted to enter their password manually when connecting to the server. This avoids placing the plaintext password on the command line or in a configuration file.

`-p`, `--port` *port* If the `-h` (`--host`) option is given, connect to the specified TCP *port* number on that host. The default is port 2323.

`--preferences` *path*
Use the specified *path* instead of `~/.gma/mapper/preferences.json` for the set of user preferences to use for this invocation. This allows a completely separate set of preferences to be used for different users or purposes. This command may not appear in a configuration file (and is ignored if it is). It must be given only on the command line.

`-l`, `--preload` Pre-load all the cached images into memory at start-up, instead of loading them as needed during the map's operation. Note that this only loads cached images which are new enough that the mapper wouldn't check the server for newer versions anyway, thus allowing a mapper client to be restarted mid-game with a minimum of impact to game performance.

`-X`, `--proxy-host` *host*[`:`*port*]
For sending files *to* the server (for authorized users only), use the specified SOCKS5 proxy server. (E.g., `-X proxy.example.org:1080`.)

`-x`, `--proxy-url` *proxyurl*
Use the given URL to connect through a proxy server to fetch image data. This does not affect the connection to the map server used by GMA (yet). (E.g., `-x http://proxy.example.org:1080`.)

`--scp-dest` *path* Specify the *server-side* directory into which files will be uploaded (authorized users only). This will be the top-level data directory for the mapper; subdirectory names will be appended to this string.

`--scp-path` *path* Specify the path to the `scp`(1) program which will be used to send files *to* the data server. (Authorized users only.)

`--scp-server` *hostname*
The host name of the storage server. Only used when sending files *to* the server (authorized users only).

`-S`, `--select` *profile*
Selects the named server *profile* from the `preferences.json` file without making it the selected profile for other invocations of the mapper.

`--ssh-path` *path* Specify the path to the `ssh`(1) program used to send files *to* the storage server (authorized users only).

`-t`, `--transcript` *path*
: Records all chat window activity (including results of die rolls) to the specified file *path*. If this file exists, it will be appended to with the new information.

  The following special tokens may appear in the *path* string, which will be replaced with values based on the time of day the file is opened:

  `%a` Mon, Tue, etc.

  `%A` Monday, Tuesday, etc.

  `%b` Jan, Feb, etc.

  `%B` January, February, etc.

  `%d` Day of month (1–31).

  `%j` Julian day of the year.

  `%m` Month (01–12).

  `%y` Year (00–99).

  `%Y` Year (all digits).

  `%H` Hour (00–23).

  `%I` Hour (01–12).

  `%M` Minutes (00–59).

  `%S` Seconds (00–59).

  `%p` AM or PM.

  `%D` Date (%m/%y/%d).

  `%r` Time (%I:%M:%S %p).

  `%R` Time (%H:%M).

  `%T` Time (%H:%M:%S).

  `%Z` Time zone name.

`--update-url` *url* Specifies the URL where updated versions of the `mapper` program may be obtained. This enables automatic upgrades. The `mapper` will, with the user's approval, download updated versions of itself from this URL and install them.

`-u`, `--username` *name*
: Sets the name used to identify you amongst the other players on your server. If this option is not provided, your current system username will be used instead.

INVOCATION

As of this writing, the mapper still has not been ported to the new GMA code in Python, and is still implemented as a Tcl/Tk script. This means you need to have a Tcl/Tk interpreter installed on your system. (See `http://tcl.tk` if you need more information about that.) Since it's a GUI application, it is run using the `wish` command (the Tcl Windowing Shell).

We have noted that at least on the Mac platform, the `wish` program will refuse to let you expand the map window larger than the largest dimensions of the screen(s) when it was launched, so you want to plug in any projector or external displays before starting the map.

INTERACTIVE USAGE

The mapper shows the dungeon area around the players and includes features which are helpful for managing game mechanics, particularly those relating to combat. It is intended to be fairly self-explanatory (and I don't have time to thoroughly document everything at the moment), so the following brief notes will hopefully suffice to help a new user navigate its quirks.

The system menu bar is not used for this application, and is left to whatever the `wish` program sets it to

for generic scripts. Instead, all of the interaction with the mapper is done through its toolbar and context menu.

Tool Bar

Across the top of the map is a graphical toolbar. Click on each button to activate its features. Note that some of these turn on/off different modes of operation for the map. When this happens, the mouse cursor will change to show the mode the map is currently in.

Each button is described briefly below. The first block of buttons control the mapper's mode of operation. They function as radio buttons (only one is active at a time, and selecting one de-selects the previously selected one).

Line Tool (cross-hair cursor)

Selects the line drawing tool. When this tool is active, click the left button to start drawing a line, then click it again at the other end of the line. You may continue clicking to get multiple connected line segments (which all count as a single object on the map). When finished, press the Escape key or click the middle button. Cancel by pressing Escape or the middle button without having defined any points on the line at all. Note that the current FILL color (not OUTLINE color) is used to draw the line on the map.

Rectangle Tool (square cursor)

Selects the rectangle drawing tool. When this tool is active, click the left button where you want one corner of the rectangle to go, then click again where the diagonally opposite corner should go. The rectangle will be outlined in the OUTLINE color and filled in with the FILL color. Cancel by pressing Escape or the middle button.

Polygon Tool (polygon cursor)

This works like the Line Tool except that the region inside the shape defined by the line segments is filled in with the FILL color, while the outline is colored in the OUTLINE color.

Note that when this tool is selected, the two option buttons become active, offering some different options for how the lines of the polygon are to be joined:

Corner Each time you click on this button, it cycles through the different corner-join options: beveled, mitered, and round.

Spline Each time you click on this button, it cycles through the spline levels from 0 (no splines, just straight lines), to 9 (use 9 lines between points to make a smooth curve).

Ellipse Tool (circle cursor)

This works like the Rectangle Tool except that it draws an elliptical shape inscribed within (tangent to) the rectangular area defined by the two mouse clicks.

Arc Tool (diamond crosshair cursor)

This tool is for making various semicircular shapes. Its operation is a little more complex than the others. When this tool is active, the option is also active, allowing you to choose the type of arc to create:

Arc type Each time you click on this option button, it cycles through the choices of arc types: pie slice, chord, and arc.

First, draw the elliptical shape for the arc (as if it were a complete ellipse) as described for the Arc Tool. Then, move the mouse horizontally to rotate the arc and vertically to adjust the length of the arc. When satisfied, click the left button to complete the arc. Cancel by pressing Escape.

Text Tool (i-beam cursor)

This is used for placing text on the map. Its operation works much like the stamp tool (q.v.), in that left-clicking on the canvas will place a new copy of the current string at that location. If there is no current string, you will be prompted to enter one. Right-clicking will prompt you for a new string rather than using the current one. The current string is displayed below the tool bar.

With this tool active, a font selection button is available. Clicking this toggles the font selection dialog. Changing the font in that dialog will alter the font of the most recently placed text object (as long as the text tool remains active) and sets the font for future text objects.

There is also an anchor selection button while this tool is active. This shows as a centered cross (+) to indicate that the text will be centered around the point where the mouse is clicked. Clicking on the anchor selection button will cycle through all of the anchor directions available: north, south, northeast, etc. These mean that the text will be aligned so that the point where the mouse is clicked will be that direction from the text. Thus, for example, selecting an anchor of "west" (incidated by a left-pointing arrow) will center the text vertically but align it horizontally so that the point is to the left of the text.

Move Tool (iron cross cursor)

This is the default mode, and the one you should keep the mapper in when not changing the map features. With this mode, you can drag creatures around the map as described below.

If the mouse is not over a creature token when starting to drag the mouse, the map grid itself is dragged, providing an easy way to scroll the map.

If you hold down the shift key while clicking the left button on the canvas in this mode, it will briefly show a marker to draw attention to the grid square the mouse is in.

Delete (aka Cut) Tool (skull cursor)

With this tool active, any object you click on with the left button will be deleted from the map immediately (no saving throw). if you click where there are multiple overlapping objects, you will be prompted to select which to delete. Press Escape if you don't want to delete any of them.

Object Move Tool (multi-arrow cursor)

This tool allows the map objects (as opposed to creature tokens) to be dragged to new locations. Note that you are dragging the object's *reference point* with the cursor. Once an object has been moved any distance with the mouse, the arrow keys (or the standard `vi`(1) movement keys) may be used to "nudge" the object by one pixel at a time up, down, left, or right; additionally the keys `u`, `d`, `f`, and `b` may be used to move the object up, down, to the front, and to the back in the stacking order (*z* coordinate), respectively. Holding the shift key while pressing the up, down, left, or right controls moves the object by 10 pixels at a time instead of 1.

Stamp Tool (star cursor)

This allows graphical tiles to be "stamped" onto the map. If there is a current tile already chosen, a new copy of it is placed on the map with the upper-left corner at the point the mouse was clicked. If no such tile was chosen, you will be prompted for its name. Right-clicking will force the selection of a new tile image rather than re-stamping the current one. See `gma-rendersizes`(6) for more information about the format of these tile files. They should be rendered and (if using a map server) uploaded ahead of time so they are visible in the map.

The next block of buttons control the appearance of any new objects added to the map.

Fill Mode

Clicking on this button toggles whether the shape will be filled or not. (Somewhat counter-intuitively, lines are filled with the FILL color, not the OUTLINE color, so turning off fill will just give you invisible lines.)

Fill Color

Clicking on this button selects the FILL color to be used to fill in new object areas. This is disabled if fill mode is turned off.

Outline Color

Clicking on this button selects the OUTLINE color to be used to draw around new object areas.

Grid Snap

Clicking on this button cycles through the grid snap options:

Off — Points may be added anywhere on the canvas (free form drawing).

1 — Points may only be added at the intersections of grid lines.

1/2 — Points may be added at grid intersections, and 1/2 way between them horizontally or vertically.

1/3 — Points may be added at grid intersections, and every 1/3 of the way between them horizontally or vertically.

1/4 — Points may be added at grid intersections, and every 1/4 of the way between them horizontally or vertically.

Line Width
: Clicking on this button cycles through the line widths from thinnest to thickest.

The next block of buttons clear the map:

Clear Features
: Clicking this button wipes the map clean except for creatures.

Clear Creatures
: Clicking this button removes all creatures from the map.

The next block gives access to tactical displays.

Toggle Combat Mode
: Normally, the GM console will automatically turn on combat mode, but if you want to manually enable or disable it, click this button. When active, the threat zones around each creature are highlighted using colored cross-hatch patterns.

  If health tracking is in effect (i.e., for creature objects which have a non-empty `HEALTH` attribute), a health bar is displayed across the bottom of each creature's token. The appearance of this bar depends on the current health of the creature. For the description that follows, the significant health statistics are:

  t — The total number of hit points the creature has when at maximum health.

  x — The extra points (below zero) which define the amount of lethal damage a dying creature can sustain before being dead. In Pathfinder and compatible d20 games (and perhaps others), this is the Constitution score for the creature.

  l — The number of hit points worth of *lethal* damage sustained by the creature.

  n — The number of hit points worth of *non-lethal* (i.e., subdual) damage sustained by the creature.

  The health bar indicates graphically the creature's health condition and relative amount of damage they have taken, as follows:

  Full health — A creature in full health will have a solid green bar across the entire width of their token's space on the map.

  Injured — The full width of the token space represents the creature's total (maximum) hit points (t). A red bar will start encroaching over the green in proportion to the number of lethal hit points (l) they have taken. A yellow bar will likewise represent the number of non-lethal hit points (n) taken. Thus, the health bar will be shifting more from green to red/yellow as the creature gets more and more injured, until as it nears the point of meeting its maker, the entire bar will be red.

  Flat-footed — A flat-footed creature (which does not also have any of the conditions listed below) will have a blue frame around the health bar.

  Staggered — When staggered due to non-lethal damage (i.e., $n>0$ and $l+n=t$), the health bar has a yellow frame around it. The creature will move to unconscious if it suffers more damage.

Unconscious — When unconscious due to non-lethal damage (i.e., $n>0$ and $l+n>t$), the health bar has a violet frame around it.

Disabled — When disabled, a red frame will appear around the health bar. The mapper will automatically assume disabled condition if a creature has exactly 0 hit points left (i.e., $l=t$.)

Dying — When at negative hit points but still above the death level ($-x<t-l<0$), a red frame will appear but the red bar will retreat to the left as more lethal damage is taken, until it's fully black at the point of death.

Stable — If dying but stabilized, the health bar will have a brown frame around it.

Dead — When completely mortally wounded ($t-l<=-x$), the health bar is solid black.

Show HP Values

This toggles the display of health statistics for players (not monsters) over the health bars. If only lethal damage has been inflicted, it displays "*hp*/*max*" where *hp* is the current number of hit points remaining, out of a maximum of *max* hit points. If non-lethal damage has been suffered, then the display is "*hp*(*nl*)" where *nl* is the amount of non-lethal damage. If a creature is fully dead, it simply says "`DEAD`".

Spell Area of Effect

This adds a spell area of effect to the battle grid. Once created, this becomes a permanent map feature which may be removed using the Cut Tool (q.v.). When this tool is activated, two option buttons are enabled which allow you to control the shape of the spell area:

Shape — This button cycles through the supported spell shapes: radius, cone, and ray.

Spread — This button toggles whether the spell effect "spreads" around corners. This is not yet implemented.

Select the point of origin for the spell by clicking the left button over a grid intersection (the tool will snap to intersections automatically). Then move the mouse to the target point of the spell and click again to complete the area. As you move the mouse, the spell's area and affected grids will be shown. The area of effect is filled in with cross-hatch patterns in the FILL color. Cancel by pressing Escape. This is an active tool like the other drawing tools. When finished, select another mode such as the Move Tool.

Ruler Tool

Selecting this tool allows you to measure the distance along a path. Click the left button on a point, then move the mouse to another point. If desired, multiple points may be clicked to build a path. Middle-click or press Escape to end the measurement.

Grid Display Toggle

Clicking this button turns on and off the display of the gridlines on the map. This is a local display setting only, and is not broadcast to other clients.

Die Roller

If connected to a server, this button brings up the chat window. In this window, you may send and receive messages and die rolls to other connected users. This window is split into three adjustable panes, described individually below. The division between each pane may be moved by dragging the mouse over the separation point or pane handle.

`Chat Messages Pane`

In the main portion of this pane displays incoming chat messages. Each is prefixed with the name of the sender. If the message was addressed only to you, the tag "*(private)*" is added after the sender's name. If it was sent to a specific subset of users, their names will be listed as "*(private to alice, bob, charlie)*".

There are two entry lines below the chat window. The top one is for sending chat messages. Anything typed in the entry box will be transmitted when the Return key is hit. To the left of this entry box is a menu button which controls who the message is sent to. If

"(all)" is selected, the message is sent to all listening clients (which need not be listed in the menu; the message will be sent to everyone at the server level). If a recipient's name is selected, it will only be sent to them. If another recipient's name is selected, they are *added to* the list of recipients. These selections are actually toggles—selecting a recipient's name again will remove them from the list. This allows for messages to be sent to any arbitrary subset of users. Selecting "(all)" will clear all selections again. The "refresh" button to the right of the entry box will update the recipient selection menu with the current set of logged-in users.

To the right of the entry line there is also a checkbox with the letter "M". If selected, this allows GMA markup formatting codes to be typed in your messages. This allows, for example, boldface text to be specified by surrounding text in ****double asterisks**** and italics by surrounding text in *//double slashes//*. See `gma-markup`(7) for a more complete explanation.

The bottom entry line is for making die rolls. Into the entry box you may type any die roll string such as would be accepted to the `DieRoller.do_roll()` method as documented in `gma-dice-syntax`(7). When the Return key is pressed, this die roll is sent to the server, which will roll the dice and transmit the results just like a chat message (which includes the currently-selected chat recipient list). The "(i)" button to the right of the entry box will bring up a help window explaining what may be entered for die rolls.

`Recent Rolls Pane`

The most recent 10 rolls entered into the above-mentioned entry box are kept in a list in this pane, with the most recent on top. Clicking on the die button next to any of these will re-roll it again. If additional modifiers are in play, they can be typed into the entry box next to the die button. Whatever is entered is simply appended to the original die expression after a plus sign. Thus, entering "5" will add "+5" to the roll, and entering "1d6 fire+3" will add "+1d6 fire+3" to the roll.

`Preset Rolls Pane`

A set of commonly-needed die rolls may be pre-set into the tool and then invoked using the third pane. Clicking the "(+)" button will add a new preset by prompting for its name, description, and die roll. The name uniquely identifies the preset within the list. The description will appear as a tooltip for your reference when looking at your presets. The new preset is saved on the server and will be loaded into your client every time it's started. Presets are invoked in the same manner as described above for recent rolls. Clicking the "(−)" button removes the preset from the list.

If a preset name includes a vertical bar (e.g., "`12|WillSave`"), then only the part after the bar will be displayed on-screen, but the entire name is used to sort the presets in the window. This allows arbitrary sort ordering without cluttering the display.

The file load and save buttons at the bottom of the pane are used to load and save the preset list to local disk files which have the format documented in `gma-dice`(5).

If your (or the system-wide) set of presets includes variables, modifiers, or lookup tables, they will appear in this pane as well.

Display Initiative Clock

This button calls up the combat initiative window which displays a set of timers which keep track of interesting events such as how long spells or other timed effects have left until they expire, the running campaign time clock, initiaitive turns, and other tactical information of use during combat.

Request a New Timer

Clicking on this button allows you to make a request to the GM to add a new timer to the initiative clock display for something you want to track in the game.

The final block of tool buttons control global operations of the mapper:

Zoom In
: Double the visual size of grid blocks.

Zoom Out
: Halve the visual size of grid blocks.

Un-Zoom
: Restore the zoom level to normal.

Load
: Add all the map objects from a disk file onto the map tool, replacing all the map features previously on the map (but not the creatures).

Merge
: Like Load, but add to the existing objects rather than replacing them.

Unload
: All the objects saved to a selected disk file are *erased* from the map.

Push
: Push the entire contents of this map client to all other clients, replacing their current contents. (Only available in store-and-forward mode, generally only for GM use. Since the server now tracks game state and clients and re-sync with it directly, there is no longer a need for clients to push their contents to each other, and that was a problematic operation anyway.)

Store and Forward
: Toggles store-and-forward mode. When enabled, this changes the behavior of the following other buttons, providing a client update path that is much more efficient and less error-prone than streaming the object updates through the server. Stand-alone (non-networked) map clients should use the normal mode of operation instead.

  Load
  : Prompts for the selection of a map file from disk as usual. However, rather than loading that file directly, it checks to see that the file is available from the server by checking the local cache and (if necessary) downloading from the server. If the file is not found by those operations, it will be uploaded to the server (assuming the user has the proper SSH access active at the time). Other clients are then instructed to load the map file from the server.

  Merge
  : As with the Load button, but merges the map file with the existing map contents rather than replacing them.

  Unload
  : Ensures that a server copy of the map file exists as the Load button does, but then instructs the remote clients to delete the contents of that file rather than sending individual object deletion commands over the network to them.

  Push
  : Saves the current map contents to a temporary file, uploads it to the server, and then instructs the other clients to load that file.

Sync
: *(Note that this button's function has changed as of version 3.25.)* This clears the contents of the mapper client and requests a fresh set of data from the server, thus synchronizing this client to be in line with the server's idea of the current game state. Depending on how the server is configured, it may automatically perform this operation for you when you connect to it.

Save
: Save everything on the map to a disk file. You will be prompted to decide whether this includes creatures as well.

Exit
: Exit the mapper program.

Context Menu

Clicking the right button over an object calls up a context-sensitive menu with the following options. Not all options will be enabled in all cases. Most of these involve performing actions on creatures.

`Remove` *name*
: Remove the creature from the map. If there are multiple creatures in the same grid, a submenu will allow you to select which one to remove, or allow you to remove them all.

`Add Player...`
: Add one or more new player tokens into the grid clicked. This pops up a dialog box to enter the relevant information about the new player:

name The name by which the creature is to be known on the map. This *must* match the name the GM console is using to track initiative, or it'll never be highlighted when its turn comes up (otherwise the name doesn't matter). If the name coincides with graphical tile images already loaded, that image will be used instead of a plain circle with the creature's name inside. If a range in the form `#`*n*−*m* is appended to the name (usually with a space between the name and this notation), then *m*−*n*+1 copies of the creature are added in a series of grid spaces starting with the one right-clicked and continuing to the right. For example, entering the name "Orc #1−3" will create three creatures, named "Orc #1", "Orc #2", and "Orc #3". Names must be unique. If another token was already on the map with the same name, it is replaced with this one.

If a different image file is needed than the default (named the same as the person's name), specify it as *imagename=creaturename* (optionally followed by the `#` notation described above).

size The size, in units of grid squares (diameter), of the creature's token. You can also use standard size designations `f` (fine), `d` (diminutive), `t` (tiny), `s` (small), `m` (medium), `l` (large), `h` (huge), `g` (gargantuan), `c` (colossal). Where it makes a difference, indicate "tall" creatures by using a capital letter and "wide" creatures with a lower-case one. Since the recommended practice is to use the size codes, which means you would use the same code for both *size* and *area* fields, any time you type into the *size* field, that will update *area* at the same time. If a different *area* is needed, that can be edited afterward separately.

area The threat area in the same units as the *size* field. This may also be one of the standard size designator codes as with *size* (and this is generally preferred). In that case, for size categories larger than medium, use upper-case (tall) letters for size categories of tall creatures, and lower-case for long creatures.

color The color of the threat zone to draw around the creature in combat mode.

reach? Check this box if the creature has a reach weapon in hand.

Clicking `Ok` places the creature(s) on the grid and dismisses the dialog box, while clicking `Apply` places the creature(s) but leaves the dialog up in case you want to add more creatures to that grid square.

`Add Monster...` Just like `Add Player` but adds a monster token.

`Toggle Death for` *name*
Flips the creature token between living and dead states. The mapper will automatically draw an "X" across the creature token in addition to switching to the "dead" image (if images are used).

`Cycle Reach for` *name*
Cycles through the extra threat zone for reach weapons. This goes from normal threat area to reach area and then to extended reach (both adjacent and reach zones together), then back to normal again.

`Toggle Spell Area for` *name*
Defines a spell effect which is described as a radius "centered on you" (or some creature). After choosing this item, click the left button to define where the radius extends from the creature's perimeter. If there was already a spell in effect, this cancels it. This differs from the spell area tool from the toolbar in that it moves with the creature and radiates from the creature's entire space rather than coming

from a fixed point on the map. The area is filled in with the current FILL color.

`Polymorph` *name* If alternative images are available for a creature, this selects which is to be displayed. If the creature has a `SKINSIZE` attribute which indicates the size of each of its polymorphed forms, then this menu will allow you to choose between the number of forms defined for that creature, and will automatically adjust the creature size at the same time. Otherwise, the mapper program doesn't know what alternate forms are available so it will offer you a choice of three different forms, and will make a best-effort attempt to locate and display the corresponding images. In this case, you will need to manually adjust the size if needed.

`Change Size of` *name*
Alter the size of a creature token.

`Toggle Condition for` *name*
Selects a condition from the list of conditions built in to `mapper` or defined by the map service for custom game-specific conditions. If the selected condition is already set for the target creature(s), then it is removed. Otherwise it is added to the target(s).

`Tag` *name* Add a tag to a creature token to indicate their conditions. The recent tags which were set are remembered and available in a sub-menu for convenience.

`Set Elevation of` *name*
Specify how high above (or below) the obvious reference level a particular creature is. This puts a tag in the upper right corner of their token in which is shown their elevation (as a simple number). The sub-menu triggered by this item allows easy selection of relative distances, so you can quickly note that a creature moved up by 10 feet, for example. Any arbitrary elevation may be directly input by selecting "`(set)`" and typing the desired elevation into the dialog box that appears. If an absolute number is input, that will be the new elevation. If the number begins with a `+` or `−` sign, its value will be added or subtracted to the current elevation instead.

`Set Movement Mode`
Various modes of locomotion are denoted in the elevation tag (q.v.) by using a different color for each. Use this menu item to select the mode currently employed by the creature:

`land` (white text on a black background)

`fly` (black text on a deep sky blue background)

`climb` (white text on a forest green background)

`swim` (white text on a teal background)

`burrow`
(white text on a sienna background)

`Deselect All` Cancels the multiple-creature selection.

`Show Visible Objects`
Moves the scrollbars to bring map features into view.

`Sync Others Views`
Moves all the other map clients scrollbars to see what this client is showing.

`Refresh Display` Redraws the contents of the local mapper client. This does not reload any data (see the `Sync` button in the toolbar for that), but just locally re-draws everything again. This is useful, for example, if the client didn't know about image data for tiles or creature tokens when it first rendered the display. Often, it will work in the background to discover the missing image data, so refreshing the display will then

render everything properly.

name Add a player token for the named player to the map, or move it to this location if it was already on the map.

Creatures

With the Move Tool (q.v.) selected, click and drag creatures to move them around the map.

If you hold the control key down while left-clicking on creature tokens, it toggles whether that creature is included in the group selection. When a group is selected, dragging any member of the group moves the entire group at once. Context selections will also apply to the entire group (e.g., to toggle death for all the selected tokens).

In combat mode, the area threatened by each creature is shown as a dashed outline, and is cross-hatched when that player's turn is up for action. Arrows are drawn between creatures in range to be melee targets.

INSTALLATION

To run the mapper, you'll need an up-to-date Tcl/Tk interpreter (8.6 or later), and the tcllib and tklib standard libraries.

Additionally you will need a copy of `curl`(1) installed on your system.

If you will be uploading content to the web server (and are authorized to do so), you will also need to have `ssh`(1) and `scp`(1) on your system.

You will need to ensure that the paths to these commands, server name(s), data paths, etc, are configured correctly for your needs as well.

SEE ALSO

`curl`(1), `scp`(1), `ssh`(1), `gma-dice`(3), `gma-dice`(5), `gma-mapper`(5), `gma`(6), `gma-dice-syntax`(7), `gma-mapper-protocol`(7).

AUTHORS

Steve Willoughby / steve@madscience.zone; John Mechalas (elevation and movement modes).

HISTORY

This document describes version 4.x of `mapper`. This introduces a breaking change from versions 3.x, mostly in terms of the communications protocol used between the mapper server and clients and the way map data are represented internally and in the disk files used by the mapper.

A version of `mapper` was also in version 3 of GMA, but was different in operation.

As of version 3.25, the operation of the "Push to other clients" button was changed so that it only works in store-and-forward mode (and is thus reserved essentially for privileged users only). This was done because the old function of that button is no longer needed and tended to cause more trouble than it was worth anyway.

Also changed in 3.25 is the function of the "Sync" button. It used to simply attempt to reconnect a client to the server (which should automatically happen anyway). Now, since the server tracks game state, simply exiting and restarting the map client accomplishes the same effect (possibly better). Now this button requests a "sync" operation with the server.

COMPATIBILITY

This program requires a reasonably modern version of Tcl/Tk, tcllib and tklib to function properly. We strongly recommend running it with the latest versions of all of those.

It is known to run on the macOS Mojave platform (tested on 10.14.6), macOS Catalina (tested on 10.15.3, but note that Apple's support for python3, tcl, and tk is such that you may want to install your own versions of those tools); and should run fine on any modern *NIX-like platform (tested on FreeBSD 12.0, Ubuntu 18.04 LTS, and Ubuntu 16.04 LTS).

It was also tested (briefly) on Windows 10. See the detailed installation and usage tutorials in the *GMA Game Master's Guide*.

FILES

`~/.gma/mapper/preferences.json`
: Preferences data storage file, normally edited via the in-application preferences editor window.

`~/.gma/mapper/mapper.conf`
: (DEPRECATED) Default configuration file read if no explicit `-C` or `--config` option is given and no `preferences.json` file was read.

`~/.gma/mapper/style.conf`
: Default custom style configuration file read if no explicit `-s` or `--style` option is given.

`~/.gma/mapper/debug.log`
: Location where debugging messages are written in addition to being displayed in the debugging window.

`~/.gma/mapper/logs`
: Runtime logfiles are stored here for each execution of the mapper.

`~/.gma/mapper/cache`
: Cached copies of images and other content are stored here to improve speed of the mapper.

BUGS

There are numerous hacks in the program which really should not be there. In fact, at this point the thing just needs to be rewritten using the newer GMA code base.

Calculation of threatened spaces needs to take elevation into account. (Although the mapper now includes a 3D-aware distance calculation when requested.)

The `-h` option really should have been for `--help` to conform to usual command-line conventions, and the `--host` option should instead have been `-H`. This may change in the future.

In previous versions, `--keep-tools` (`-k`) was called `--master` (`-m`), but this never really made sense, as it didn't really mean the mapper was in any sort of controlling or leadership role. It only meant it would refuse to turn off its own toolbar if asked to do so. The new name is more descriptive of the actual function.

Some of the menu behavior required by the mapper client has been shown not to work with Tcl/Tk version 8.6, so the mapper avoids using those features when the Tcl/Tk version is less than 8.7.

DEPRECATED FEATURES

The preferred way to configure `mapper` is to use the built-in preferences editor. This obsoletes the use of the older simple configuration file which needed to be edited by hand, and makes it unnecessary to use CLI options to specify the runtime parameters for the mapper's operation. The use of the legacy configuration file is still supported, however, as is the use of CLI options where desired for *ad-hoc* changes from the saved preferences.

However, the switch to the preferences system makes the following CLI options now obsolete and they no longer perform any function: `--generate-config`, `--generate-style-config`, `-s`, and `--style`.

The `-c` (`--character`) option is no longer supported.

COPYRIGHT