tUME

the Universal Map Editor

Configuration Guide

May 3, 2010

Gregg A Tavares

♻ Printed on recycled paper (contains 50% waste paper including 10% post consumer)

Configuring tUME

tUME.INI

The tUME.INI file holds information for configuring tUME to your particular needs. Usually, we will supply a tUME.INI file that is setup for your particular needs, but, if for some reason you'd like to change the tUME.INI file for some other needs, or if you just find that you'd like to change the menus or the keys to something you are more comfortable with, then here is how to do it.

Setting Initial Global States

You may set the initial state of global variables, such as menu bars visible/invisible, tile-brush visible/invisible, stratify paste on/off, etc., by specifying a list of events that toggle the state of the particular global variable when tUME starts up.

The list of initial events are specified in the `[Initial Events]` section of tUME.INI. All global events default to the off state, so to turn them on, add them to this section.

E.g., to make tUME default to stratified pasting, add the following line to the `[Initial Events]` section:

```
ToggleStratifyPaste
```

Redefining Room Types

Room types define how tUMEPack interprets a particular room, whether to treat a particular room as a level graphics room, as a conversion room, etc. They are listed when you select **Room|Set Info...** They may be re-defined to suit your particular application. If you change the room types, you'll want to change your tUMEPack program to match.

In the tUME.INI file, there is a section called `[Type Groups]` with three lines in it:

```
TileType=MCKids 2 Tileset Types
RoomTypes=MCKids 2 Room Types
LayerTypes=MCKids 2 Layer Types
```

`RoomTypes` are described here, `TileTypes` are described in the section **Redefining Tileset Types**, below, and `LayerTypes` are described in the section **Redefining Layer Types**, below. The string following the `RoomType=` defines the name of the section to look in for the list of room types. You may have several room type lists in the tUME.INI file, and select one just by changing the label following the `RoomType=`. Continuing with this example, there is a section called `[MCKids 2 Room Types]` that looks like this:

```
[MCKids 2 Room Types]
Level Room=0
Table Room=1
Picture Room=2
Parallax Room=3
Mode 7 Room=4
Flat Conv Room=5
Layered Conv Rm=6
```

This section lists all the room types that are defined. The text that appears to the left of the '=' is the name of the room type. The number following the '=' is the room type. This is the number that tUMEPack looks at to determine how to deal with a particular room.

Redefining Layer Types

Layer Types are currently only used by the **Smart Flip** feature. Layer Types define which tileset user types are associated with each layer. Thus, when you press `[Spacebar]` while **Smart Flip** is enabled, tUME will find the last source room with the same tileset user type for the floor layer.

In the tUME.INI file, there is a section called `[Type Groups]` with three lines in it:

```
TileType=MCKids 2 Tileset Types
RoomTypes=MCKids 2 Room Types
LayerTypes=MCKids 2 Layer Types
```

`LayerTypes` are described here, `TileTypes` are described in the section **Redefining Tileset Types**, below, and `RoomTypes` are described in the section **Redefining Room Types**, above. The string following the `LayerType=` defines the name of the section to look in for the list of layer types. You may have several layer type lists in the tUME.INI file, and select one just by changing the label following the `TileType=`. Continuing with this example, there is a section called `[MCKids 2 Layer Types]` that looks like this:

```
[MCKids 2 Layer Types]
Layer 1=0,4,5,6,7,8
Layer 2=1
Layer 3=2
Layer 4=3
Layer 5=0,4,5,6,7,8
Layer 6=1
Layer 7=2
Layer 4=3
```

This section lists all the layers that are defined. The text to the left of the '=' is the layer number being defined. The comma-separated list following the '=' lists all tileset user types that may be pasted into the layer on the left.

Thus when **Smart Flip** is enabled, and the floor is layer 2 of an edit room, pressing `[Spacebar]` will display the source room that contains the most recently selected tiles of user type 1 (defined as **Contour Tiles** in the next section). Or if the floor is layer 1 of an edit room, will display the most recently selected tiles in a source room containing tiles of user type 0, 4, 5, 6, 7, or 8.

Redefining Tileset Types

Tileset types define how tUMEPack interprets a particular tileset, whether to treat a particular tile as graphics, as contour information, or as object type, etc. They are listed when you select **Tiles|Set Info...** They may be re-defined to suit your particular application. If you change the tileset types, you'll want to change your tUMEPack program to match.

In the tUME.INI file, there is a section called `[Type Groups]` with three lines in it:

```
TileType=MCKids 2 Tileset Types
RoomTypes=MCKids 2 Room Types
LayerTypes=MCKids 2 Layer Types
```

`TileTypes` are described here, `RoomTypes` are described in the section **Redefining Room Types**, above, and `LayerTypes` are described in the section **Redefining Layer Types**, above. The string following the `TileType=` defines the name of the section to look in for the list of tileset types. You may have several tileset type lists in the tUME.INI file, and select one just by changing the label following the `TileType=`. Continuing with this example, there is a section called `[MCKids 2 Tileset Types]` that looks like this:

```
[MCKids 2 Tileset Types]
Image Tiles=0,1,11111111
Contour Tiles=1,2,xxx11111
Special Tiles=2,3,xxx11111
Object Tiles=3,4,xxx11111
4-Color Tiles=4,1,11111111
256-Color Tiles=5,1,11111111
Mode 7 Tiles=6,1,xxxxxxxx
Global 16 Tiles=7,1,11111111
```

```
Logic Tiles=8,1,xxxxxxxx
```

This section lists all the tileset types that are defined. The text that appears to the left of the '=' is the name of the tileset type. Three fields follow the '=' separated by commas; the first field is required, the other two are optional. The first field following the '=' is the Tileset Type. This is the number that tUMEPack looks at to determine how to deal with a particular tileset.

The second comma-separated field following the '=' specifies the stratify paste layer. Thus, in the above example, if stratify paste is turned on, then `Image Tiles`, `4-Color Tiles`, `256-color Tiles`, and `Mode 7 Tiles` will be pasted into the first layer, `Contour Tiles` will be pasted into the second layer, `Special Tiles` into the third layer, and `Object Tiles` into the fourth layer.

The third comma-separated field following the '=' specifies which tile flag bits are enabled for this particular tileset. In the above example, with the exception of `Contour Tiles`, all tile flag bits are enabled. For the `Contour Tiles`, the `x`'s in bit 5, 6 & 7 positions mean that bits 5, 6 & 7 do not get set for that tileset. The current definition for bit 7 is priority bit, bit 6 is X Flip tile, and the definition for bit 5 is Y Flip tile. Please see **Redefining Tile Flip Bits**, below.

Redefining Tile Flip Bits

Every tile in the map has a tile flag byte associated with it. These eight bits may be redefined for each individual tile in the map. Currently, they are used to represent tile X-flipping, Y-flipping, and colorset information. By default, bit 6 is used to represent an X-flipped tile, and bit 5 is used to represent a Y-flipped tile. The relevant configuration lines look like this:

```
[X Flip Bit]
Enable=x1xxxxxx

[Y Flip Bit]
Enable=xx1xxxxx
```

If you change the enable so there are no 1 bits, like this:

```
[X Flip Bit]
Enable=xxxxxxxx
```

then X-flip is globally disabled.

Redefining Colorsets

Different target hardware platforms have different colorset definitions. You can change the way tUME deals with colorsets. Associated with every tile in the map is a tile flag byte. These eight bits may be redefined for each individual tile in the map. Currently, they are used to represent tile X-flipping, Y-flipping, and colorset information. Below are the relevant lines from the tUME.INI file that deal with colorsets.

Example tUME.INI colorset sections

```
[Color Mask Groups]
16-Bit Nintendo Color
16-Bit Nintendo Priority

[Tile Mask Events]
ClearColor=0xxxx000,0xxxx000
 SetColor0=1xxxx000,0xxxx000
 SetColor1=1xxxx001,0xxxx000
 SetColor2=1xxxx010,0xxxx000
 SetColor3=1xxxx011,0xxxx000
```

```
 SetColor4=1xxxx100,0xxxx000
 SetColor5=1xxxx101,0xxxx000
 SetColor6=1xxxx110,0xxxx000
 SetColor7=1xxxx111,0xxxx000
    SetPri=xxx1xxxx,xxx0xxxx
  ClearPri=xxx0xxxx,xxx0xxxx

[8-Bit Nintendo Color]
Enable=1xxxxxxx
xxxxxxx0:xxxxx0xx
xxxxxxx1:xxxxx1xx
xxxxxx0x:xxxx0xxx
xxxxxx1x:xxxx1xxx

[8-Bit Sega Color]
Enable=1xxxxxxx
xxxxxxx0:xxx0xxxx
xxxxxxx1:xxx1xxxx
xxxxxx0x:xx0xxxxx
xxxxxx1x:xx1xxxxx
xxxxx0xx:x0xxxxxx
xxxxx1xx:x1xxxxxx
xxxx0xxx:0xxxxxxx
xxxx1xxx:1xxxxxxx

[16-Bit Nintendo Color]
Enable=1xxxxxxx
xxxxxxx0:xxx0xxxx
xxxxxxx1:xxx1xxxx
xxxxxx0x:xx0xxxxx
xxxxxx1x:xx1xxxxx
xxxxx0xx:x0xxxxxx
xxxxx1xx:x1xxxxxx

[16-Bit Nintendo Priority]
Enable=xxx1xxxx
xxx1xxxx:1xxxxxxx
```

The first section `[Color Mask Groups]` specifies which Color Mask Groups are active. The next two lines are the name of the sections to use, and are names chosen by the user to describe each color mask group. The above example says that both the '`16 bit Nintendo Color`' group and the '`16-bit Nintendo Priority`' group are active.

Now, look down at the `[16-bit Nintendo Color]` section. The first line

```
ENABLE=1xxxxxxx
```

means that if a particular tile's flags have bit 7 set then affect them with this Color Mask Group.

To understand how colorsets work you need to know somethings about tUME. Every tile in a tUME map has a one byte flag. These flags are used to track colorsets, priority and X and Y flipping. Also, tUME draws it's graphics in a 256 color mode. This mode uses one byte per pixel.

In the above example, if a particular tile's flags has bit 7 set then when each pixel of that tile is drawn on the screen in tUME each pixel's value will be affected by the '`16-Bit Nintendo Color`' group. The way the pixels will be affected is defined by the following lines:

```
xxxxxxx0:xxx0xxxx
```

```
xxxxxxx1:xxx1xxxx
xxxxxx0x:xx0xxxxx
xxxxxx1x:xx1xxxxx
xxxxx0xx:x0xxxxxx
xxxxx1xx:x1xxxxxx
```

The first line says, if bit 0 of the tile's flags is 0 then bit 4 of the tile's pixels should be set to 0. The second line says, if bit 0 of the tile's flags is set to 1 then bit 4 of the tile's pixels should be set to 1. Putting all these lines together says set bits 4 thru 6 of every pixel in this tile to the same bits as bits 0 thru 2 of this tile's flags. This means a particular tile will be drawn in either the first 16 colors, the second 16 colors, the third 16 colors on up to the eighth set of 16 colors.

Now, in our example we have two Color Mask Groups active. The other group '`16-Bit Nintendo Priority`' says:

```
Enable=xxx1xxxx
xxx1xxxx:1xxxxxxx
```

The first line says: if bit 4 of this tile's flags is 1, then (the second line says) if bit 4 of this tile's flags is 1, then set bit 7 of this tile's pixels to 1. (No, I did not stutter in the last sentence. In this Color Mask Group, the same bit (number 4) is used for two things.)

This is all fine and dandy, but how do we affect the actual tile flags? Well, you create `Tile Mask Events` in the `[Tile Mask Events]` section. Look at the one below. The First line

```
ClearColor=0xxxx000,0xxxx000
```

defines a new Event. This is an event just like all the Events listed at the end of this document. You may assign it to a key or to a menu item. When you choose this event you put tUME into Color mode. In Color mode, when you draw in a room the tiles you draw over will have their tile flag bits affected, and nothing else. They will be affected as specified by the Event in the TUME.INI. The two bitmasks defines the action for the left mouse button, followed by the action for the right mouse button. For the ClearColor event, the left and right mouse button have the same effect; each tile's flag will have its bit 7, bit 2, bit 1 and bit 0 set to 0 and the other bits will be left unchanged.

```
SetColor0=1xxxx000,0xxxx000
```

For the above SetColor0 event, pressing the left mouse button will set each tile's flag bit 7 to 1 and reset bits 2, 1 and 0 to 0. Pressing the right mouse button will reset each tile's flag bit 7, 2, 1, and 0 to 0.

These new tile flag bits will now be used when drawing the tiles. Of course as always it is up to tUMEPACK to interpret these bits and turn them into something useful for your project.

Configuring Search and Replace

As supplied, when tUME is counting brushes, searching, or replacing, it tries to match tiles exactly; i.e., it searches for tiles with the same flipping, same priority, and same colorset. The `FlagBits=` line in the group `[Search Options]` defines which tile flag bits to match. Thus if the tile flag bit 4 is the priority bit, then the following lines will ignore tile priorities when searching:

```
[Search Options]
FlagBits=111x1111
```

Configuring Character Counting

As supplied, tUME will count the number of 8x8 image characters (TilesetType = 0) loaded, whether or not they appear in an edit room. It counts SNES mode 2 characters, so it only looks at the lowest four bits in determining whether two tiles have the same image. It will check for X-flip, Y-flip, and XY-flip in determing whether two

characters are the same. Tiles that also appear in Table Rooms (RoomType = 1) get count separately, without merging duplicate tiles nor checking for flipped tiles.

The relevant lines in the tUME.INI files look like this:

Example tUME.INI sections to count all characters:

```
[Count Characters]
CharacterWidth=8
CharacterHeight=8
CompareMask=xxxx1111
Groups=Character Groups

[Character Groups]
CharGroup1=MergeDuplicates,MergeXFlips,MergeYFlips,MergeXYFlips
CharGroup2=NoMerge

[CharGroup1]
Tileset=0,Room=none
Tileset=0,Room=0..65535

[CharGroup2]
Tileset=0,Room=1
```

The `[Count Characters]` section describes the basic parameters of the characters to count. `CharacterWidth` and `CharacterHeight` says to count 8x8 pixel characters. If both are zero, it counts all tiles, irregardless of size.

`CompareMask` says to only examine the lowest four bits in each pixel when determining which characters are duplicates. `Groups` says the section name of the different characters to count is `[Character Groups]`.

The `[Character Groups]` section says there are two logical groups of characters to count, `CharGroup1` and `CharGroup2`. Both names ares user-defined, and refer to sections by the same name.

Following each character group name is one of five keywords in a list, separated by commas. The keywords and their meanings are:

`NoMerge`	treat every character as unique
`MergeDuplicates`	count characters with same image as one tile
`MergeXFlips`	count characters that are flipped about the X-axis as one tile
`MergeYFlips`	count characters that are flipped about the Y-axis as one tile
`MergeXYFlips`	count characters that are flipped about the X and Y-axis as one tile

The `[CharGroup1]` section determines how to count characters for the first logical character group. The first line says to count all characters of tileset type 0 that have not been placed in any edit rooms (none). The second line says to count all characters of tileset type 0 that have been placed in any edit rooms (rooms with a room type in the range of 0..65535). Note that when we count the characters in this logical group, we `MergeDuplicates`, `MergeXFlips`, `MergeYFlips`, and `MergeXYFlips`.

The `[CharGroup2]` section determines how to count characters for the second logical character group. The line says to count all characters of tileset type 0 that appear in an edit room of room type 1. Note that when we count the chracters in this logical group, we do `NoMerge`.

Other sample lines and their meanings:

Tileset=0..2,Room=none	count characters of tileset type 0, 1 or 2 that do not appear in an edit room
Tileset=0..2,Room=0	count characters of tileset type 0, 1 or 2 that appear in an edit room of type 0

Tileset=0..255,Rooms=0..65535 count all characters (0..255) that do appear in some edit room (0..65535)

Each character will only be counted once in a given logical character group, irregardless of how many lines it satisfies within that group. Thus, any given line may appear multiple times in a given logical character group, and tUME will not count the characters more than once (it's the union of characters in each line and not the sum of all characters in each line). However, if any given line appears in *n* different logical character groups, each time a tile satisfies the condition on that line, it will be counted *n* times.

Counting Only Characters That Appear in Edit Rooms

Here are the lines you want to change in your tUME.INI file to make tUME count only the characters that are actually used in edit rooms:

Example tUME.INI sections to count only the characters that appear in an edit room:

```
[Count Characters]
CharacterWidth=8
CharacterHeight=8
CompareMask=xxxx1111
Groups=Character Groups

[Character Groups]
CharGroup1=MergeDuplicates,MergeXFlips,MergeYFlips,MergeXYFlips

[CharGroup1]
Tileset=0,Room=0..65535
```

Notes that these commands count ALL tiles of tileset type 0 in ALL edit rooms. If you only want to count a specific edit room, you could give that room a unique room type, and change the expression `Room=0..65535` in the `[CharGroup1]` section to `Room=<your unique room type>`.

Configuring the File Requester

You may configure the file requester to either display the directories and files alphabetically intermixed or display all the directories first, then all the files.

To configure the file requester to list all the directories first, then all the files, set `DirsAtTop=` in the `[File Requester]` section of the tUME.INI file to `1`; to configure the file requester to list the directories and files intermixed alphabetically, set `DirsAtTop=` in the `[File Requester]` section of the tUME.INI file to `0`:

```
[File Requester]
DirsAtTop=0
```

Note that if you set `DirsAtTop=0`, the `[Files]` and `[Dirs]` buttons do not appear in the file requester.

Configuring Mouse Sensitivity

You may set the mouse sensitivity by changing the numbers in the `[Mouse]` section of the tUME.INI file:

```
[Mouse]
MouseXRes=640
MouseYRes=400
```

The `MouseXRes=`sets the X-axis sensitivity; the default value is 640. The `MouseYRes=` sets the Y-axis sensitivity; the default value is 400. In both instances, setting the numbers to a larger number will make the mouse less sensitive, and setting the numbers to a smaller number will make the mouse more sensitive.

Configuring Printing

You may specify which port to print to and page numbering `[Print Maps]` section of the tUME.INI file:

```
[Print Maps]
PrintTo=lpt1
NumberPages=1
```

To print to a different port, say LPT2, change the command to `PrintTo=lpt2`. To print to a file, specify a DOS filename instead of `lpt1`. To disable page numbering, change the command to read `NumberPages=0`.

Configuring Tileset Search Path

By default, tUME loads tilesets using the directory and filename specified in the tUME map file. This can be a problem if you move tilesets to another sub-directory (perhaps on a different machine). To adress this problem, tUME will use the following search strategy in loading tilesets:

1. [tUME.INI]	Load tileset using the directory saved in the tUME map file, if `SearchAsSpecified=1`. If not found, then
2. [USER-SPECIFIED]	Search sub-directories specified by user, if any. If not found, then
3. [tUME.INI]	Search sub-directories specified by `SearchDir`'s, if any. If not found, then
4. [tUME.INI]	Search current sub-directory, if `SearchCurrentDir=1`. If not found, then
5. [ASK USER]	**Can't load 'tileset'! Would you like to try a different file?**

Option 2 sub-directories are added by the user when they click **Yes** in response to the question "**Try to load other 'problem' tilesets from here?**"

Option 3 is useful when you copy maps from another machine and you want to place all tilesets in a separate sub-directory from the tUME map. E.g., if you want tUME to search the sub-directory `Z:\PROJECT\ART` for tileset files, add the following lines to the tUME.INI file:

```
[Load Options]
SearchDir=Z:\PROJECT\ART
```

If you want to search the sub-directory `C:\GOOD\ART` as well, simply add another line:

```
SearchDir=C:\GOOD\ART
```

Option 4 is useful when you copy maps from another machine and you want to place the tUME map and its tileset files in the same sub-directory. To enable this option, add the following lines to the tUME.INI file:

```
[Load Options]
SearchCurrentDir=1
```

Option 5 is useful if you want to rename a tileset. Change the tileset name before you start tUME. When tUME tries to load the renamed tileset, it won't find it, and then it will give you a chance to specify the new filename.

Thus to make tUME search the current sub-directory of the map file, set `SearchAsSpecified=0` and `SearchCurrentDir=1`. Do not specify any `SearchDir`'s.

Redefining Keys

To define a key, edit the tUME.INI file with your favorite text editor. Move down through the file until you see a lot of lines that start with the word 'KeyEvent.' This will be in the [KEYS] section. Notice the syntax of the lines:

```
KeyEvent    'Key' Event
```

You must put the word 'KeyEvent' at the beginning of the line. This tells tUME what the line is for. Next you specify the key you are assigning. All keys must by surrounded by single quotes and all key combinations and special keys must be surrounded by angle brackets '<', '>'. Some example keys might be:

'q'
'p'
'2'

Some key combinations are:

'<Alt-p>'
'<Home>'
'<Ctrl-PgUp>'
'<Keypad-6>'
'<Alt-Ctrl-t>'

Then you specify which event to assign to this key. All the events are listed below. For example if you wanted to be able to save your map by pressing Alt and the 's' key you would put

```
KeyEvent    '<Alt-s>'   SaveMap
```

Key List

Most alphabetic, numeric and symbol keys may be specified by putting the symbol, letter or number in single quotes. If the specification requires more than one character you must also use angle brackets '<>' in your key specification. NOTE: Capital letter will require you to press <SHIFT>. In other words there is a difference between assigning something to 'a' and 'A'.

Below is a list of special keys.

'<F1>'	'<Down>'	'<Ins>'
'<F2>'	'<Left>'	'<Insert>'
'<F3>'	'<Right>'	'<Home>'
'<F4>'	'<BkSp>'	'<End>'
'<F5>'	'<Esc>'	'<PgUp>'
'<F6>'	'<Tab>'	'<PgDn>'
'<F7>'	'<Space>'	'<Grey-+>'
'<F8>'	'<Return>'	'<Grey-->'
'<F9>'	'<Enter>'	'<Grey-*>'
'<F10>'	'<Del>'	'<Grey-/>'
'<Up>'	'<Delete>'	

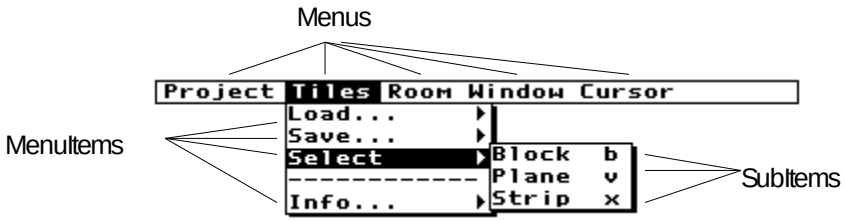
Below is the list of Qualifiers that may be added to a key specification. More than one may be added separated by a dash '-' for example '<Alt-Shift-p>'.

Alt
Shift
Ctrl
Keypad

Redefining Menus

All menus may be redefined. To do this edit the TUME.INI file with your favorite text editor and look at the `[MENU]` section. There are four menu keywords:

POPUP
POPDOWN
MENUITEM
STICKYMENUS



Look at the `[MENU]` section and compare it against the menus you see in tUME. A **POPUP** line displays a choice that brings up more menu choices. A **POPUP** has one optional parameter which is the text you want to appear on the **POPUP**. You may also use any of the switches (see below). For every **POPUP** statement you MUST have a corresponding **POPDOWN** to mark the end of the current **POPUP**. **POPDOWN** takes no parameters.

The **MENUITEM** keyword specifies a selection or choice on the menus. **MENUITEM** has two optional parameters. The first is the text that appears for this **MENUITEM** and the second is the **event** that happens when this **MENUITEM** is chosen (see **List of "Events"**, below). If you don't specify an event, then this menu item will appear on the menus but it won't have any effect. You may use any of the **switches** with **MENUITEM** (see below).

The keyword **STICKYMENUS** affects the entire menu. **STICKYMENUS** makes the menus work like Microsoft Windows in this sense: if you bring the menus down with the mouse and then let off the mouse the menus will stay active. This means you can click once to bring the menus down and click again to choose the menu item you want. If you don't use the **STICKYMENUS** option then menus always disappear when you let off the mouse, which is more Macintosh / Amiga like.

When you specify text for a **POPUP** or **MENUITEM** you can put an '`&`' symbol before any letter and then the following letter will be '<u>underlined</u>' and will also become a key you can use to choose the **POPUP** or **MENUITEM** when using the menus from the keyboard. You may also put the characters '`\t`' once in the text. All text to the left of the '`\t`' will appear left justified in the menus and all text to the right of '`\t`' will appear right justified.

For all **MENUITEM**s with an associated event, if the event has any keys assigned to it (see **Redefining Keys**, above) the first assigned key will appear right justified in the menus.

Switches

Finally there are two switches, **SEPARATOR** and **MENUBREAK**. **SEPARATOR** adds a horizontal line just before the current **MENUITEM** or **POPUP**. **MENUBREAK** will start another column of menu items for the current popup.

[COLOR MENU]

In addition to the main `[MENU]` section, tUME also expects to find a `[COLOR MENU]` section in tUME.INI. This defines the menu commands available while the palette requester is active. The default `[COLOR MENU]` is:

```
[COLOR MENU]
POPUP "&Colors"
      MENUITEM    "&Load..."        LoadPalette
      MENUITEM    "&Save..."        SavePalette SEPARATOR
      POPDOWN
POPUP "&Range"
      MENUITEM    "&Load..."        LoadPaletteRange
      MENUITEM    "&Save..."        SavePaletteRange  SEPARATOR
      POPDOWN
```

Redefining Scrolling

Scrolling events and the keys they are attached to are user defined in the tUME.INI file. The two steps involved are first, create a new event that specifies how far and in which direction to scroll the room, and second, attach a key to this new event.

All cursor movement events are creating by listing them in the `[Cursor Movement Events]` in the tUME.INI file. As supplied, there are eight events in this section:

```
RightScroll=-1,0
RightScrollMultiple=-5,0

LeftScroll=1,0
LeftScrollMultiple=5,0

DownScroll=0,-1
DownScrollMultiple=0,-5

UpScroll=0,1
UpScrollMultiple=0,5
```

The two numbers that follow specify how far <x>,<y> to move the screen. So if we wanted to add a new event that scrolled the screen up and right one tile time, we would add the following line:

```
UpRightScroll=-1,1
```

The name (`UpRightScroll`) we select is totally arbitrary; you may call the new event `Foobar` if you wish.

After you define a new cursor scrolling event, you need to attach a key to it, otherwise you won't be able to access it. We skip down to the `[KEYS]` section in the tUME.INI file, and add this line:

```
KeyEvent     '<Keypad-9>'       UpRightScroll
```

Now whenever we press `9` on the numeric keypad, we will scroll right and up one tile. See the section Redefining Keys above for a more detailed description of attaching keys to events.

List of "Events"

Below is a list of 'Events'. These events can be assigned to almost any key combination and to any menu item.

Some of the events listed below, such as **LeftScroll**, **LeftScrollMultiple**, **RightScroll**, etc. are **user-defined** events rather than built-in events; this means the name of the event has be created by the user and entered in the **tUME.INI** file. In the case of the aforementioned scrolling events, they are defined in the **[Cursor Movement Events]** section in tUME.INI file.

The keys listed following each event is what is currently attached to the event as defined in tUME.INI. These keys are completely user redefinable.

About

Displays the About tUME box.

AppendLayer

Load a layer and append it as the topmost layer of the current room.

AppendMap

Load a map without clearing out the previous map. If the filespecs of the tilesets in the new map match the tilesets filespecs of the currently loaded map then the new map will use the currently loaded tiles. If the filespecs do not match but the file names do then you will be asked if tUME should load a new tileset or use the one it already has loaded. You usually don't want it to load the new tileset.

CenterRoomOnCursor KEY = 'n'

Moves the tile under the cursor to the center of the display or window. Same as DPaint.

ClearMap

Clears all rooms and tilesets from memory.

ClearRoom

Clears all tiles from a room.

ClearRoom3

Clears the room and it's tile size so that a new tile size may be stamped into it.

CopyColors1

Copies the color palette and all color cycles from the room the current tile-brush was grabbed from to the current room.

CopyColors2

Copies just the color palette from the room the current tile-brush was grabbed from to the current room.

CopyColors3

Copies just the color cycles from the room the current tile-brush was grabbed from to the current room.

CopyrightStatus

Displays the copyright message in the status bar. This is what you usually see in the status bar when you first run tUME.

CountChars KEY = '<Alt-C>'

Count the characters used in the map.

CreateRoom

Creates a new room by first asking for the width and height of the new room.

CursorStatus KEY = '<F9>'

Sets the Status bar to Coordinates mode. See 'Coordinates' in the Status Bar section.

DecBackColor KEY = '<Ctrl-[>'

Changes to background color tUME uses to draw areas where there are no tiles.

DeleteLayer

Deletes the current floor layer.

DeleteRoom

Deletes a room from memory.

DeleteTileset

Deletes a source tileset from memory.

Download16 KEY = '<Alt-D>'

Download current layer of current room as 16-color characters to target development system.

Download256 KEY = '<Ctrl-D>'

Download current layer of current room as 256-color characters to target development system.

DownScroll KEY = '<Down>'

User-defined event; see tUME.INI. Scrolls the current room down one tile.

DownScrollMultiple KEY = '<Ctrl-Down>'

User-defined event; see tUME.INI. Scrolls down multiple tiles. How many is defined in tUME.INI.

EditOnlyFloor KEY = '<Alt-F>'

Makes only the floor layer visible, and make only the floor layer editable.

EditRoom KEY = '<Keypad-.>'

Put brush stamping in Edit Room Mode (the default mode). This as opposed to the colorset modes. See **Colorsets** in the *tUME User's Guide* and **Redefining Colorsets** above.

ExportBrush

Writes the current brush as an IFF picture.

ExportRoom

Writes the current room as an IFF picture.

ExportScreen

Writes the current screen as an IFF picture.

FlipPanes KEY = '<space>', 'j'

Flips between the two panes of a window.

FloorDown KEY = '<Alt-Down>', '4'

Moves the floor down one layer.

FloorUp KEY = '<Alt-Up>', '3'

Moves the floor up one layer.

GetGridFromBrush KEY = '<Alt-G>'

Set the size of the grid based on the size of the tile-brush.

GetGuideFromBrush KEY = '<Alt-O>'

Set the size of the guide based on the size of the tile-brush.

GoEnd KEY = '<Ctrl-End>'

Show the lower-right corner of the current room.

GoHome KEY = '<Ctrl-Home>'

Show the upper-left corner of the current room.

GroupSaveTiles

Saves the current map just like SaveMap but also saves the tile graphics into the map in the TMGX format.

HideCursor KEY = '<F8>'

Turns the mouse pointer off.

HilightTile KEY = '<Alt-h>'

Show the source room that contains the tile contained in the tile-brush. See **Highlight Tile in Source Room** in the *tUME User's Guide*.

IncBackColor KEY = '<Ctrl-]>'

Changes the background color tUME uses to draw areas where there are no tiles.

InsertLayer

Inserts a new layer to the current room, and pushes the previous floor layer and layers above it up by one.

KeepDownloadPalette

Toggles whether to keep updating the palette on the target development machine or not.

LastMenuEvent KEY = 'a'

Repeats the last menu command. Same as DPaint.

LeftScroll KEY = '<Left>'

User-defined event; see tUME.INI. Scrolls the current room left one tile.

LeftScrollMultiple KEY = '<Ctrl-Left>'

User-defined event; see tUME.INI. Scrolls multiple tiles to the left. How many is defined in tUME.INI.

LoadLayer

Load a layer to the floor of current room, and pushes previous floor layer and layers above it up by one.

LoadMap KEY = '<Alt-L>'

Loads a map after first clearing any old map from memory.

LoadPalette

Loads an IFF palette into the currently selected palette. You may not attach a key to this event.

LoadPaletteRange

Loads an IFF palette range into the selected palette range. You may not attach a key to this event.

LoadRoom

Loads a room, same as AppendMap.

LoadTilesAllTiled KEY = '<Ctrl-L>'

Loads a tileset from a DPaint picture by cutting out every tile on the grid. See **All Tiled** in the **Tilesets** section in the *tUME User's Guide*.

LoadTilesBoxed

Loads a tileset from a DPaint picture using the boxed method. See **Boxed** in the **Tilesets** section in the *tUME User's Guide*.

LoadTilesCookieCutter

Loads a tileset from a DPaint picture using the Cookie Cutter method. See **As Brushes** in the **Tilesets** section in the *tUME User's Guide*.

LoadTilesFullTiled

Loads a tileset from a DPaint picture using the Full Tiled method. See **Full-Tiled** in the **Tilesets** section in the *tUME User's Guide*.

LoadTilesFullTiledNoBlank

Loads a tileset from a DPaint picture using the Tile-Blanks method. See **Tile-Blanks** in the Tilesets section in the *tUME User's Guide*.

MakeCompositeTiles

Create a composite tileset from an edit room. See Composite Tiles section in the *tUME User's Guide*.

NextRoom KEY = '2'

Switches to the next room.

OpenLayer

Adds a new layer to the current room. The new layer becomes the topmost layer.

PreviousRoom KEY = '1'

Switches to the previous room.

PrintMap

Prints current room to HP LaserJet compatible printer.

QuitAndExit KEY = '<Shift-Q>', '<Alt-x>'

Exits tUME.

Replace KEY = 'r'

Find the next occurrence of the Search Buffer, and replace it with the current tile-brush.

RightScroll KEY = '<right>'

User-defined event; see tUME.INI. Scrolls the current room to the right one tile.

RightScrollMultiple KEY = '<Ctrl-right>'

User-defined event; see tUME.INI. Scrolls the current room to the right multiple tiles. How many is defined in tUME.INI.

RoomStatus KEY = '<F6>'

Sets the status bar to Room Info mode. See **Room Info** in the *tUME User's Guide*.

SaveLayer

Saves the floor layer of the current room into a map file and only those tilesets that are used in the layer.

SaveMap KEY = '<Alt-s>'

Saves all the current rooms and tilesets into a map file.

SavePalette

Saves the currently selected palette to an IFF file. You may not attach a key to this event.

SavePaletteRange

Saves the currently selected palette range to an IFF file. You may not attach a key to this event.

SaveRoom

Saves the current room into a map file and only those tilesets that are used in this room.

SaveRoomAll

Saves the current room into a map file and also saves all the tilesets that are currently loaded.

SaveTilesAsBrushes

Saves all the tiles as DPaint brushes in a directory you specify.

ScrollLock

Prevents tUME from scrolling around a room.

ScrollLockKEY

Prevents tUME from scrolling around a room.

SearchNext KEY = 's'

Find the next occurrence of the Search Buffer in the current room.

SelectBlock KEY = 'b'

Removes the current tile-brush from your pointer, and set tUME so next left mouse button press select a new tile-brush.

SelectSquare KEY = 'v'

Removes the current tile-brush from your pointer, and set tUME up so that you may select another. When you select another it will only be one layer deep regardless of how many layers deep the current room is.

SetGridOrigin KEY = '<Shift-G>'

Sets the upper left corner of the grid.

SetGridSize

Brings up dialog box that asks user to set the grid size.

SetGuideOrigin KEY = '<Shift-O>'

Sets the upper left corner of the guide.

SetGuideSize

Brings up dialog box that asks user to set the guide size.

SetMaxTileUsage

Sets the largest tile usage count that will be displayed.

SetRoomInfo

Brings up a dialog box that allows you to edit a room's name, User Type and User Number.

SetSearchBuffer KEY = '<Ctrl-S>'

Copies the current tile-brush to the Search Buffer. Used with search, and with search and replace.

SetStampPaint KEY = '<F1>'

Set the tile brush so it does not stamp NULL tiles.

SetStampReplace KEY = '<F3>'

Set the tile brush so it does stamp NULL tiles.

SetTheColors KEY = 'p'

Brings up the Palette Requester allowing you to edit the palette of the current room.

SetTilesetInfo

Brings up a dialog box that allows you to edit a tileset's name, User Type and User Number. The tileset is determined by the tile in the top left corner of your current tile-brush.

ShowBrushCount KEY = 'h'

Count and display how many times the current brush occurs in the floor and above layers of the current room.

SpaceToggle KEY = '\'

Toggles whether or not a single pixel space is drawn between each tile in a room or not. By default Source Panes display tiles with the space and Edit Panes do not.

StripDownBlockCopy KEY = '<Shift-X>'

Strips the current tile brush to just one layer.

TileStatus KEY = '<F5>'

Sets the status bar to Tiles mode. See Tiles in the Status Bar section above.

TMGCSaveTiles

Saves the current map just like SaveMap but also saves the tile graphics into the map in the TMGC format.

ToggleCycleColors KEY = '<Tab>'

Toggle Cycling Colors on and off.

ToggleDownloadOneScreen

Toggle between downloading one screenful and downloading entire room.

ToggleJamPalette KEY = '<Alt-p>'

tUME will force the last 4 colors of the palette to something you can read the status bar with. This event toggles that feature on and off. The default is off.

ToggleLAll KEY = 'i'

Toggles the Invisible and Locked flags of the floor layer on or off.

ToggleLInvisi

Toggles the Invisible flag of the floor layer on or off.

ToggleLLock KEY = 'l'

Toggles the Locked flag of the floor layer on or off.

ToggleLockRoom

Locks or Unlocks the current room. A locked room acts like a source room meaning when you click on a locked room you are picking up tiles and not placing them.

ToggleShowBrush

Toggles whether or not you see the tiles you are carrying in your current tile-brush to just a rectangular outline.

ToggleShowGuide KEY = 'o'

Toggles the guide on and off.

ToggleShowTileUsage KEY = '<Alt-u>'

Toggles the display of tile usage numbers on and off.

ToggleSmartFlip

Modify FlipPanes so it flips to the appropriate source room for the current edit room floor layer. See **Smart Flip** in the *tUME User's Guide*, and **Redefining Layer Types**, above.

ToggleStratifyPaste

Enable paste mode where tiles in brush gets pasted into correct layer as defined by the tileset type.

ToggleSwankyMode

Does nothing. Really!

ToggleTitleBar KEY = '<F10>'

Turns the Status Bar on or off.

ToggleUseGrid KEY = 'g'

Toggles the grid on and off.

ToggleZoom KEY = 'm'

Toggles between last zoom setting and zoom off.

Undo KEY = 'u'

Undoes the last edit you made to a room.

UpScroll KEY = '<Up>'

User-defined event; see tUME.INI. Scrolls the current room up one tile.

UpScrollMultiple KEY = '<Ctrl-Up>'

User-defined event; see tUME.INI. Scrolls the current room up multiple tiles. How many is defined in tUME.INI.

UseEditPalette KEY = '<Alt-e>'

This event displays all subsequent source rooms using the last edit room's palette.

UserStatus KEY = '<F7>'

Sets the status bar to User Info mode. See **User Info** in the **tUME User's Guide**.

VersionStatus

Shows the current version of tUME in the Status bar.

WClose

Closes the current Window.

WCreate

Opens a new Window.

WLockClear KEY = '<Alt-a>'

Set the current Window's current pane to show both edit rooms and source rooms.

WLockToEdit

Sets the current Window's current pane to show edit rooms only.

WLockToSame

Sets the current Window's current pane to show rooms that are the same type as the current room.

WLockToSource

Sets the current Window's current pane to show source rooms only.

XFlipBrush KEY = 'x'

Flips the current tile-brush horizontally.

YFlipBrush KEY = 'y'

Flips the current tile-brush vertically.

ZaveMap

Saves a map and saves every tile in every tileset as a separate DPaint brush (**Xave**).

ZeroBackColor KEY = '<Ctrl-'>'

Sets the background color to zero. tUME uses this color to draw all areas where there are no tiles.

ZoomIn KEY = '<>>'

Display the current room using the next available zoom in setting.

ZoomOut KEY = '<<>'

Display the current room using the next available zoom out setting.