DUMPTUME.DOC

PROGRAMMER : Gregg A. Tavares
VERSION : 00.000
CREATED : 03/20/92

DUMPTUME:

Takes one or more tume files and dumps all the information contained there in to the display (or stdout in C terms).

This program is intended as an example of reading a tUME map using our tUME map reading routines.

There is not a lot of documentation. It is expected that since the program is printing all tUME map information it is pretty self explanatory. (ie. if the code says:

```
printf ("Room Width = %d\n", room->Width)
```

then it should be fairly obvious that 'room->Width' contains the width of a room.

)

A few notes:

* To load one tUME map. Call one of the two following functions:

CreatetUMEMapTMGX or CreatetUMEMapTMGC.

If you look in the code for dumptume.c you will see it calls AppendtUMEMap. It only calls this because dumptume allows you to load more than one map and when more then one map is loaded the tilesets from both maps must be merged. I have personally never had to load more than one map at a time so I would usually call CreatetUMEMap????. If you look in ECHIDNA\BTUME.H you will see that CreatetUMEMap???? is just a macro that calls AppendtUMEMap.

* The difference between calling CreatetUMEMapTMGX or CreatetUMEMapTMGC is as follows.

If your map has tile graphics saved in it (because you chose Save+TMGC or Save+TMGX while inside tUME) then when you load the map the graphics will be loaded. Now REGARDLESS of whether you saved +TMGC or +TMGX if you call: ----------

CreatetUMEMapTMGX

your tile graphics will be loaded as bitplanes similar to the Amiga and also similar to NES, SNES and GAMEBOY.

If you call:

CreatetUMEMapTMGC

your tile graphics will be loaded as one byte per pixel just like MCGA mode. This is also similar to SEGA Genesis and TANDY.

It is upto you do decide which way you need your graphics loaded. If you are writing a tUMEPACK for the IBM or SEGA Genesis you will probably want to call CreatetUMEMapTMGC because with your graphic data loaded as one byte per pixel it will be easier to manipulate into the kind of data you are likely to need.

If you are writing a tUMEPACK for the Amiga or NES, SNES or GAMEBOY you will probably want to call CreatetUMEMapTMGX because your graphic data will be loaded as bitplanes which is exactly what these systems need.

* The tUMEMap loading routines will use EMS and/or XMS IF it is available. This is because tUMEMaps can get extremely large. If you look at the

code to dumptume.c you will notice calls to a function ActivateXTRA(). This function takes a special pointer to what we call XTRA memory and returns a far pointer to real (conventional) memory. ActivateXTRA takes the XTRAPntr and makes the data it represents available in conventional memory. That data will only be available until the next call to ActivateXTRA. Also if you modify this memory it is not guarenteed to be saved between calls to ActivateXTRA. If you want it to be save you must call UpdateXTRA and pass it the XTRAPntr you passed to ActivateXTRA.
ActivateXTRA and UpdateXTRA are prototyped in the file ECHIDNA\XTRAMEM.H.

* Rooms and Tilesets start at 1 meaning the first room is at map->Rooms[1] NOT map->rooms[0].

* Please check the routines PrintTileInfo() in dumptume.c. It will show you how to randomly access any tile in any room.

* The struct of a MapType is defined in ECHIDNA\BTUME.H

GreggT.

PS. I know that processing tUME Maps can be a pain especially if you need to process multiple rooms and multiple tilesets and you haven't done it before. I'm sorry but this amount of complexity comes with such a powerful and flexible map editor. Now before you start complaining think about it for a while. What are your alternatives? If you chose to use another map editor you would most likely be limited to a fixed number of tiles and a possibly even fixed sized maps. You would not be able to edit more than one room/level at a time nor would you be able to create layers or load multiple groups of tiles. Your tiles would be a fixed size. You would not be able to create composite tiles or if you could they would also be a fixed size.

Ok you say but it's still to hard to deal will multiple tilesets and multiple rooms. Well you can always tell your artists and designers to only ever create one room at a time in tUME and only ever use one tileset. Then when you load the map using these routines you will only have to deal with one tileset and one room at a time which makes your job very easy.

Of course if you do this you will be throwing away all the flexibility of tUME but you will get very simple data just like every other unflexible map editor gives you.