**Version 0.1 – 2/17/2015**

This is an initial stab at a manual for the BAP Map Tool, colloquially known as 'Obi.exe'. It is woefully inadequate. We’ll be updating this multiple times, and we need your help. Read this to get the most basic, ugly understanding and go play. Look for what stuff does and if you can clearly describe what’s happening, do it! We’ll post a thread on the LucasForums that we’d like your input on. Then we’ll assemble what we’ve learned and update this manual. Repeat ad nauseum.

Right now you’re looking at V0.1. I apologize in advance since it’ll become clear how much of this tool I didn’t use. I was primarily dealing with the AI scripting language and actor placements within the Map tool, plus some Model assembly.

Here's the obligatory 'with great power comes great responsibility' speech. The absolute best way to get this shut down is if we do something stupid with this. We're not going to make money, we're not going to use Lucasarts property inappropriately, this is strictly for private use. We shouldn't be distributing this since many of the sound effects are straight out of Lucas archives. Technically it's in our EULA to not monkey with the TPM files.

* Zanzibar

**Basic Instructions**

**Running the Tool**

Run in Win95 compatibility mode. Please save a copy of your original TPM files elsewhere so you can revert easily if you screw something up.

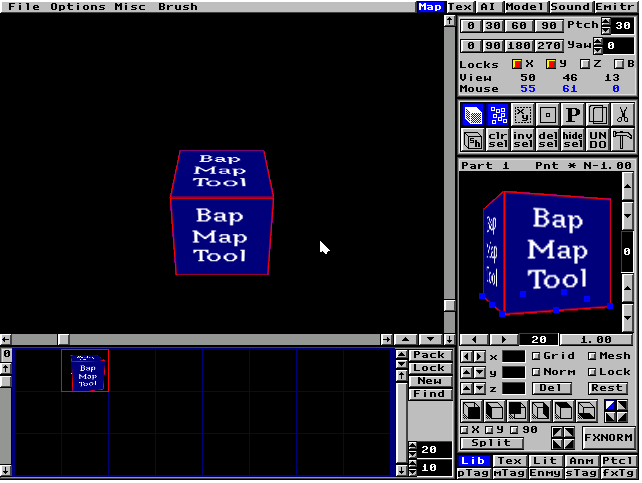
I \*think\* that if you put the contents of the Obi folder within the root TPM directory then you'll be using the installed sounds and you'll avoid some error messages at runtime. If you’d rather not risk it, the folder contains everything you need to edit and enter a level.

Try obi.exe, obiold.exe, bafobi.exe, netobi.exe, robsobi.exe . They are slightly different versions of the same executable and some run better than others in different OS. I could never get obi.exe to work in Win7 Home but got the rest to work. Win7 Pro would only run netobi.exe .

**Questions needing answering:**

* Does the tool need to be placed in the TPM directory to avoid the missing streamed sound errors?
* What specifically out of the main obi directory is needed to run the game if TPM is already installed?
* Can we figure out system settings so that the ‘kosher’ version - Obi.exe – can run? That’s the version that WMain.exe is built out of, which runs the actual TPM game.

**Tool layout/Game mode**



BUG: Cursor can get into a mode that can ‘erase’ the screen when over the buttons etc. Click on the TEX or AI tabs at the top, then back to MAP and the problem should go away.

Arrow keys move around the map, camera relative. Pressing CTRL with left/right arrow keys will rotate camera, CTRL with up/down will change camera elevation.

Spacebar activates in-game engine, then press A to drop into the game. Pressing Esc when in the game will return you to the tool (BUG:often the tool rendering becomes corrupted, particularly in Win7 Pro. If you can't see the polys, you should quit and restart.)

**MAP:**

Map creation. The map is made up of blocks, each block can have multiple polygons. Adding polys to a block will add to ALL placements of that block. Add a polygon to the selected block by clicking one of the 'cube' faces; the tool will place a full-cube-size poly on the side of the cube you clicked.

Highlight one of the Vertex buttons on your selected poly to adjust its position, then click the arrow buttons next to x/y/z to adjust that poly's orientation.

If you need to make a 'one-off', highlight your map part in the blocks below, and while it's highlighted in blue, click the 'new' button and it will duplicate your selection as a new block. You can then place the new block and modify it, the changes you make to this block will not be reflected in the previous blocks.

You can change the assigned texture on a poly by clicking the Tex lower right tab and choose the texture you want assigned to it. The textures are organized by pages which you can switch between on the lower left.

You can double-up on blocks, so that you can stamp down 2 or more blocks on the same location, but that becomes difficult to manage later when you're trying to select one block and not the other.



**Lower right tabs When in Map mode:**

**Lib -** changes the bottom panel to show the library of created blocks within your level.

**Tex -** Textures that you assign to polys. Created in the TEX tab at the top.

**Lit -** lights within the level.

**Enmy -** Actor placement editor

**Ptcl –** placed Emitters (particle systems).

**Anm -** the built-in map animations that handle doors etc. I know next to nothing about how to create them.

**Ptag/Mtag/Stag/FXtag** are various methods for tagging the geometry with properties for sound effects, collision etc. I didn't play much with these.

**FXNORM** attempts to conform the normal to face ‘outwards’. Doesn’t always work as intended, we generally stayed away from it.

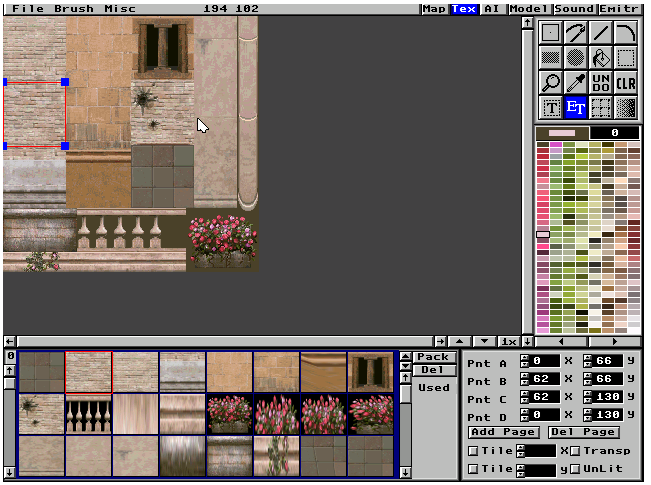
**Split** divides a 4-point poly into 2 3-point tris.

**Del** deletes the selected poly. **Rest** is supposed to restore it – but it doesn’t. I have no idea what it ‘actually’ does.

**Pack** scans the Map library for duplicate pieces and deletes the dupes and replaces them with the originals.

**Questions needing answering:**

* What do the Grid/Mesh/Norm/Lock buttons do?
* What do the 2 4-blocks near the FXNORM button do?



**TEX:**

Textures for the levels use a 256-color palettized system. Generally we would create texture pages that all live within the same color range in true color and then convert them into 256 colors using JKRes. The PCX files stored their own palettes within the image.

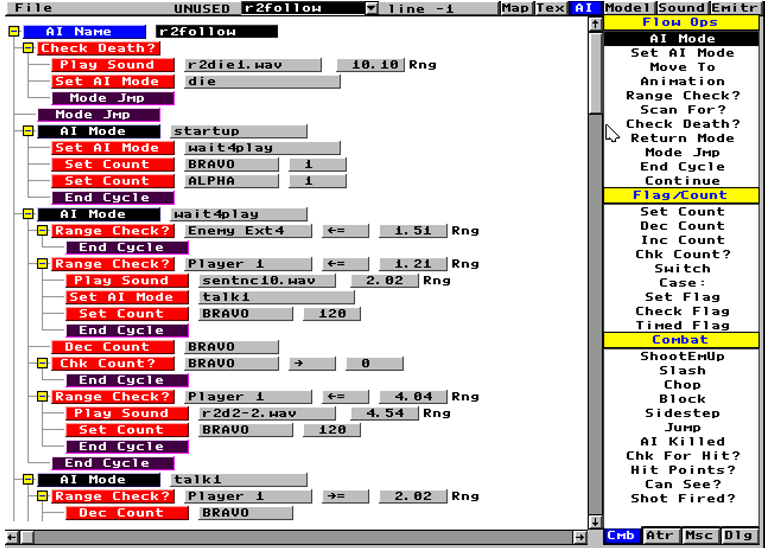
Textures are created in PCX format 64x64, 128x128 or 256x256, you can create and edit them within the tool here or else do it externally and import. To import, click the Add Page button and choose your PCX file. The color at Palette 0 would always be transparent; any pixel that had Palette 0's color assigned wouldn't get drawn.

Create a new texture from the image by pressing the T button, it will create a cursor box that you'll then place on your texture page. After it's created, Pressing the ET button will show you the 4 vertices' location for that texture. You can adjust each vertex by using the Pnt A-D value boxes in the lower right.

Each texture image page can have a max number of 96 individual textures referencing it.

I don’t think the **Tile** fields are used very much, if at all.

**Transp/Unlit** are used for glowy things, like readouts. I think.



**AI:**

This is a scripting language that is very powerful. Every frame, the engine starts at the top of the script and starts processing. It goes through blocks of script until it reaches an end cycle, then it will cease processing that script and move on to the next one.

An ‘AI Mode’ is a block of code that the script stays within until it is told otherwise. The script defaults to the first AI Mode encountered when the actor starts up.

Any lines of code that are ABOVE the first AI Mode will get processed every frame. We call that the ‘preamble’.

See how the opcodes can have an ‘indent’? The opcodes contained within the indent will only process if the associated check returns true. The ‘check death’ in the 2nd line will only process the Play Sound and Set AI Mode opcodes if the creature dies, at which point the check death will return true and the script will jump to the Die AI Mode.

**AI Opcode Defs/Components**

**Counters Alpha-Echo:** Each script can have up to 5 variables to keep tabs on stuff. We often use them as timers, where we’ll set it to 120 or something on startup and then dec it every frame, and when it is 0 executes something.

**Flags 1-4:** Flags are used to send simple messages. If several NPCs are checking for a flag to be red, then they’ll all react if a single actor sets it to red. Green is default, you can use the Timed Flag to set it to a different color for a period of time, at which time it will revert to green.

**Switch/Case:** You can have the script execute different commands based on a Counter value.

**Enemy Ext (1-12):** If you need a script to examine a specified Actor placement for range check etc, choose an Enemy Ext and then in the AI Def Externs window in the Map editor/Enmy, that external reference will appear. Set that value to the the Index of the Placement you want it to examine. What REALLY sucks is if you delete an Placement, it changes the index of all subsequent Placement, but it doesn’t update these External references – you gotta do it by hand. Sucks, I know.

**Map Anim:** Similar to Enemy Ext, you can specify a Map Animation to start/stop in the AI Def Externs window, and assign it by index.



**Other notes:**

**Questions needing answering:**

* Under the AI Characteristics, there are a ton of buttons that I don’t know what they do.
* H/W Performance and Player Skill – I think if those values are not set to 0 the placement wouldn’t appear unless it’s a higher-level Hardware parameter and Player Skill parameter – how are those set?
* Active range is number of cubes before the placement activates, Deact range means ?
* Fine Tune?