

Hacking For GBA Doom: a hacking guide by Samuel 'Kaiser' Villarreal (svkaiser@gmail.com)

Before we begin, I will assume that you are already familiar with the Doom engine/code, if not then check out <http://doom.wikia.com/wiki/Entryway> to learn everything about the Doom engine. Also I haven't gotten into detail about hacking everything, but if you find the tools useful and want to get more serious into GBA Doom hacking, email me and I'll consider updating and improving the tools/articles. Forgive me as this article isn't that great.

Requirements:

Tools to extract the Doom ROM from your GBA cartridge, a hex editor (included) and the hacking tools. If you extracted the Doom ROM correctly, it should be about 8 megabytes (8,388,608 bytes total)

Introduction:

This guide will give you a quick explanation of how to hack GBA Doom as well as how to use the hacking tools. One of the first tools we will need is a hex editor. I recommend using Hexplorer (which is included with this kit) because of its easy to use interface.

GBA Doom is a direct port of Jaguar Doom but the blood has been changed to green and several 'gore' mobs have been removed and some powerups like the invisible sphere are present in the game but have no effect when picked up. Some of the HUD data are hard coded into the ROM. The levels feature a new data lump called 'LEAFS' which is responsible for each linedef definition. These data lumps however are hard coded as well.

Finding the memory address:

Finding the memory address is simple, just take the ROM offset in your hex editor and add it by 0x8000000 (which is 134217728 in decimal).

Location of the data:

Believe it or not, most of Doom's data is not in the iwad but actually hard coded, not a problem though but this means that the actual code will need to be modified. Since I think that no one will hardly hack for GBA Doom, I will only mention a few of the data that are hard coded in the ROM.

ROM offsets:

0x4E38: This is the pointer to the actual IWAD data when loaded in the memory address (subtract the value by 0x8000000 and you should get the ROM offset for the iwad)

0x17090: This points to the memory address of where the main status bar is saved at. (Value should read 0x70D04F08)

0x6BEE9C: This marks the beginning of the Mobj frame definitions, if you are familiar with decompiled then this shouldn't be hard to edit.

0x6C1DD8: This marks the beginning of the Mobj definitions (like health, speed, frame number etc)

0x92A5C: This marks the beginning of the sound definitions. Each value points to the sound in the memory address.

0x1E514: This is a pointer that points to the sound definition table. Change this value if you ever add additional sounds to the table. (If you do add more sounds to the table, you will need to move it to avoid overriding existing data)

0x207CC: Music directory?

0x20FC4: Music data?

0x6C3A10: This contains a table of sprite information. Each value defines a specific sprite and what lump order they belong in the wad. So if you ever change the order of a sprite (lets say you move the sprite POL5A0 to the end of the wad directory, then you must change the value to the new position).

0x6C73F0: This table calculates the # of times that sprite is used in the wad. Both 0x6C3A10 and 0x6C73F0 must be used together. You can get more information on these by looking at the Jaguar Doom source code.

0x14658 and 0x14834, 0x1F30: All of these points to the sprite table (0x6C73F0).

0x5DA4D8: This is the table for the map leafs. Each word points to the beginning of the Leaf's data.

0x6C76D0: This points to 0x5DA4D8 in the memory address.

Hacking GBA Doom contents

Getting Started

In order to edit GBA Doom, we need to dump the main IWAD from the ROM. In order to do so, you must first dump the ROM. You can buy extraction utilities online.

Using Jagedit

Jagedit is an old hacking utility for almost all console Doom platforms that I made a while back, but the main focus is using the tool for GBA Doom. Open the command prompt and use the following commands:

```
jagedit.exe -gbawadformat <Rom location> -dumpwad
```

This should dump the main iwad to the same location as the executable.

Dump data contents

The next step is that we need to dump all the data contents from the wad. First download either Deepsea, XWE, or WinTex and then create a folder where you will do all the editing at. **NOTE:** I really recommend staying away from XWE, since the tool has the tendency to destroy the iwad and corrupt data. It also saves certain files as bmp which you need to store all data files as .lmp files.

Then in that directory create the following folders:
/Data/, /Sprites/, /Maps/, /Textures/, /Flats/, /Patches/

For /Maps/ creating separate directories for each map called /MapXX/ and so on. XX is the map number. So when dumping map01 it should go in /Maps/Map01/ and so on.



First: dump all the lump contents starting from T_START to T_END to the /Textures/ directory. Then everything from F_START to F_END to the /Flats/ directory. The lumps, Playpal, Texture1, Colormap, and Crypal goes in the /Data/ directory. (NOTE: Crypal isn't really needed so you don't have to dump this lump). And then if you want, dump the maps, but is not really required. Next dump everything from M_TITLE to M_SKULL2, SMCH_033 to MMCH_096 to /Patches/.

Don't worry about the sprites, we will need a special tool to extract them (which saves the time rename the '.' files).



Using LMPDMP.exe

Make sure that LMPDMP.exe and the iwad are in the main editing directory. Open the command prompt and call the following command

```
LMPDMP.exe <IWAD> -dumpgbasprites
```

If done correctly, everything should go in /Sprites/

Time to start editing

Once you got all the data extracted, put the main iwad you dumped into another folder like

/IWAD/ for example. Now copy the gbatextures folder included with the kit in the /IWAD/ folder. Copy and paste the following files into your main editing folder:

GBADLINK.EXE
GBADMAKE.EXE
JAGEDIT.EXE
LMPDMP.EXE
LMPMAKE.EXE
MIMVIEWER.EXE
SPRMAKE.EXE
ENDIANS.DLL
SPRLIST.LMP
DOOM.ACT
LBSP.EXE

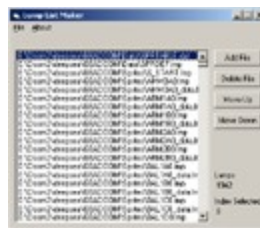
Using LMPMAKE.EXE

First in order to make our own custom IWAD, we need to assemble the data back together. LMPMAKE is a lump organizer that tells GBADMAKE what files to assemble for the wad.

How to use

Load Lmpmake, the buttons are pretty obvious to what they do. Load all the lump files together. Remember that all textures goes between T_START and T_END, flats goes between F_START and F_END etc.

NOTE: There are two files called SPRDEF and SPRTABLE.dat included in the kit. Make sure you include SPRTABLE as the first lump followed by SPRDEF. Its important that you do so.



Copy and paste S_START.lmp and S_END.lmp from the kit into the /Sprites/ directory. Put all the sprite data between these two markers.

Note about sprites: you may notice that there are files with the _data.lmp extension at the end. These files are important, so make sure they follow the previous lump with the same name. For example:

pol5a0.lmp
pol5a0_data.lmp

trooa1.lmp
trooa1_data.lmp



Place these two files in the /Data/ folder (overwrite the existing ones). Then use GBADMAKE again to produce the wad.

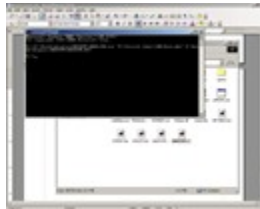
Using GBADLINK.EXE

This tool will link your custom iwad and your gbadoom ROM.

Use the following command

`GBADLINK.exe <path of doom rom> <path of doom iwad>`

This should link and produce a file called GBADOOM.GBA



Use your favorite GBA emulator and load your custom rom. Everything SHOULD load correctly.

Also if you see the Doom title, then everything works. But we have no map loaded! But we will make our own right now.



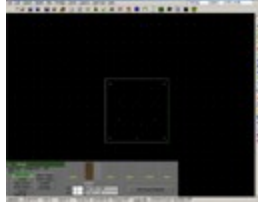
Creating a custom map

Now you should know already how to make a map. So this should be straight forward.

Setting up your editor to map for gbadoom.

Included with the kit (located in /gbatextures/) are two wad files: gbaTex.wad, and gbaFlats.wad. Use these two in order to create your map.

For testing reasons lets just make a box map with a player start and other stuff like so:



Save your map in your gbadoom editing directory.

Using LBSP.EXE

LBSP is a customized BSP compiler (originally by Fraggie, Killough, and the Prboom team) that is used to generate the LEAFS data lump for your map. Parameters are like so:

`LBSP.exe <pwad> <output pwad>`

For now lets just override your current pwad with the new one.

Our next step is to dump the map contents into the /Maps/Map01/ folder. Use `lmpdmp.exe` and use the following commands

`LMPDMP.EXE <pwad> -dumpmaplumps XX` where XX is the map number. Should be 1 since you made a map for Map01.

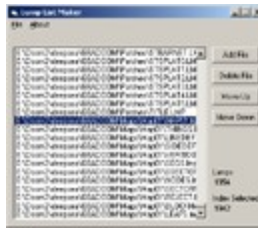
If done right all the map contents should be dumped in /Maps/Map01/



Once again open LMPMAKE.exe and include the map data lumps **MAKE SURE THEY ARE IN THE PROPER ORDER.**

In this case it should be:

- Things
- Linedefs
- Sidedefs
- Vertexes
- Segs
- Sectors
- Nodes
- Sectors
- Reject
- BlockMap
- Leafs



Use GBADMAKE to compile your wad. No need to create another SPRDEF, SPRTABLE lump since you haven't changed the order of the sprites.

Link your wad and rom again and then load the rom in your emulator.

And hopefully if everything has been done correctly, you should see something similar to this:



Troubleshooting:

If the sprites are appearing weird, then try generating a sprdef and sprtable lump again and recompile your wad/rom and try again.

If the game doesn't load AT ALL, then one of the main reasons is that the lump data file size are not in multiples of 4. The reason why is that the GBA can only read words and not bytes, this includes drawing two pixels at a time, reading data two bytes at a time and saving two bytes at a time and so on. But the tools should automatically adjust the file sizes for you, especially map data files.

Heh, and thats all I can say about troubleshooting..

Custom Textures:

I will only be brief on this, again since I am assuming you already know how to edit doom, I won't get into much detail.

First of all the textures1.lmp is exactly the same so edit the lump and implant your custom texture (you may have to edit the pname lump in the gbatex wad as well). Now take your texture image and load it in your paint program such as photoshop. First flip the image horizontally and then rotate the image to the left. After that save it as a raw/binary file. Save the file between the T_START and T_END markers and replace texture1 with your custom texture1 lump. To add flats, just save an image as a binary image and stuff it between F_START and F_END

Custom Sprites:

Here is where you will be sweating, adding sprites is VERY tricky, but should be easy due the tools. Again I won't be detailed about it since you should already know how to do this. Anyways:

Open your image in Photoshop etc. Note that widths are the columns and the height are the rows. If you have a pixel touching the last row, then you must add a new row with just the pixels used for transparency only. Do the same for the textures, flip horizontally, then rotate to the left. Save it as a binary/raw image file and give it the name of the sprite you want to replace.

Open the command prompt and use the command:

`SPRMAKE <raw image file> -i width, height, x offset, y offset.`

This should produce two files:

`yoursprite.lmp`

`yoursprite_data.lmp`

Add them to the lump list and re-generate the sprdef/sprtable lumps if needed then compile your wad/rom and try it out.

Patching:

Of course if you ever want to show your hacked stuff to others, you will need to create a patcher that will patch the rom into your version. Don't EVER include the whole rom.

I included a patch program by ClickTeam with this kit. The setup is pretty straight forward.

Thats pretty much all I got to say, again I didn't get into much detail here since I don't think not that many people will be hacking gbadoom, but if you do find this helpful and want to know more about gbadoom hacking, simply email me and I'll see if I can update the tools or this article or any other questions regarding to this.

Happy Hacking
-kaiser