

## INTRODUCTION

ColorIt is a script-based technology aimed to colorize black & white Game Boy games (DMG). The goal is to allow any user to colorize their favorite games on their own in a way much easier than ROM hacking, and conserving a good tradeoff between simplicity and possibilities.

The goal of this document is to provide basic instructions on how to integrate the ColorIt technology to an existing emulator. All has been made as simple as possible. If your language of choice is C or C++, then all you should need to do is to add the files to your project and call a few functions from these files when specific events happen.

## DESCRIPTION OF THE PACKAGE

All the code necessary to read and interpret ColorIt script files (.pal.ini file type) is supplied in the 'color-it' folder of this archive. You should copy it to your project folder. It is made of the following components, as a form of .c/.h file pairs (they can compile to C++ if you change the extension to .cpp):

- **crc32**: Optional routine computing the CRC32 of some data. Comes from *zlib* compression library and is not required if you already have your own strictly-equivalent CRC32 function.
- **Parser**: The core of the language. Reads language elements and dispatches them in the form of tokens.
- **Lang**: Analyzes language tokens and checks for the syntax. It allows reading more complex elements, such as colors.
- **System**: Contains routines to be called at specific events.
- **User**: Contains routines called by the system when specific commands are executed or errors detected. The user should know what to do of them, though it may be enough to let what the default implementation does: it stores them into buffers that can be accessed from the emulator itself.

Also example files are put in the 'example' directory. These include:

- **lcd**: an example of colorized LCD implementation using variables defined in **user.c**.

This example code has been made to be easy to understand. Just search for 'ColorIt\_' in the source code and you will see how to ColorIt is integrated. All in all, that's about 20 lines.

## LIST OF FUNCTIONS TO BE CALLED EXTERNALLY

In order to interact correctly with the supplied code, there are a number of functions that your code needs to call in specific situations. They are listed below:

- **ColorIt\_Init(ROMPath, ROMData)**: this functions needs to be called every time a new ROM is loaded. The full path is used to check if there is a corresponding color file on the system and use it if it is the case. The loaded ROM data needs to be passed as a second argument. The title of the game will be extracted from this data.
- **ColorIt\_endFrame()**: call this at the end of each frame (60 per second).
- **ColorIt\_exitingLcdc(VRAMData)**: this has to be called in two different cases:
  - Unconditionally when a special action can potentially modify the VRAM completely (for example loading a save state) but not when the CPU writes a data to the VRAM,
  - When the game enables the LCD (i.e. only if it was disabled before and is now enabled). The game does that by writing a '1' to the bit 7 of the LCDC register. This can be detected by

adding handling code for writes to the LCDC register (\$FF40), as with special registers like DMA (\$FF46), and if `new_value & ~old_value & (1 << 7)` then `ColorIt_exitingLcdc` needs to be called. You should already have such detection code as a hack for some games like Bomb Jack.

- And that's all!

If you want more info, you can download the source of VisualBoyAdvance CE edition or HEIG-Boy (both available on <http://brunni.dev-fr.org>).