

Einrichtung einer C-Entwicklungsumgebung für den Gameboy

Heiko Wolf  
h.wolf@skywave-2000.de  
FDL 1.3

Stand: 09.03.2019

## **Inhaltsverzeichnis**

1 Vorwort.....	2
2 Die C-Umgebung GBDK.....	2
3 Der Assembler RGBASM.....	2
4 Make.....	2
5 Dokumentationen und Informationsquellen.....	2
6 Abschluss .....	2

## 1 Vorwort

Retrogaming ist in vielerlei Munde. Neben dem Erhalt alter Computer und Konsolen, das Spielen von genauso alten Spielen auf diesen Geräten oder per Emulation ist ein beliebter, steht wachsender Zeitvertreib oder sogar professionelles Hobby. Interessant ist es nicht nur zu konsumieren, sondern selbstständig neue Inhalte für die betagten Geräte zu erschaffen.

Dieser Artikel will dazu eine Grundlage schaffen. Indem gezeigt wird, wie eine Entwicklungsumgebung für die Sprache C<sup>1</sup> und einige nützliche Tools installiert werden.

## 2 Die C-Umgebung GBDK

Das Gameboy Developer's Kit<sup>2</sup> ist eine betagte Software (Beiträge auf Sourceforge<sup>3</sup> sind von 2000 bis 2002), aber dieses Kit ist die einzig bekannte aktuelle Möglichkeit den Gameboy in der Sprache C zu programmieren. Für diesen Artikel wird die Version 2.95-3<sup>4</sup> für Windows<sup>5</sup> verwendet.

Die Installation gestaltet sich recht einfach. Das Archiv wird in einen Ordner, z. B. C:\GBDK, entpackt und dieser Ordner wird zur PATH-Variable<sup>6</sup> von Windows hinzugefügt. Ist das Ganze erfolgt, kann über Eingabe von `lcc -v` in einem CMD-Konsolenfenster geprüft werden, ob die Einrichtung erfolgreich war:

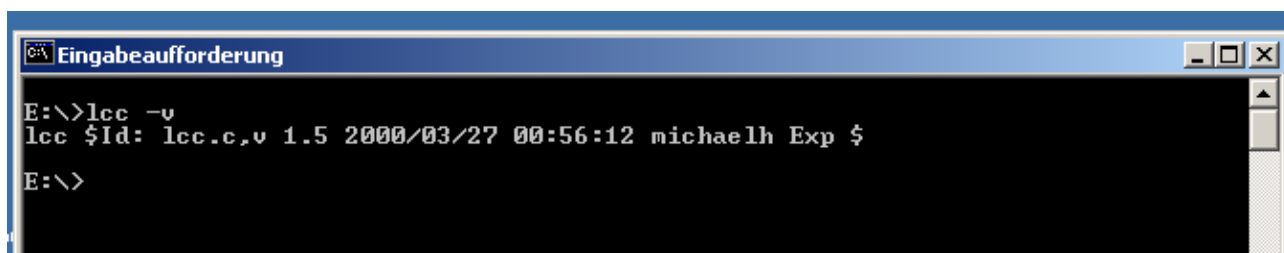


Bild 1: Ausgabe von `lcc -v`

## 3 Der Assembler RGBASM

GBDK verwendet als Compiler den `sdcc`<sup>7</sup> der einen Assembler mitbringt. Der volle Funktionsumfang von GBDK der Version 2.95-3 wird durch diesen allerdings nicht erschlossen. So ist das einfachere Banking<sup>8</sup> mit dem mitgelieferten Assembler nicht möglich. Die Webseite formuliert dazu:

„Banked function support. It is now easier to break the 32k barrier from within C. Functions can live in and be called transparently from any bank. Only works with `rgbds`“<sup>9</sup>.

Auf diese Funktion sollte nicht verzichtet werden, da dieser Punkt das Programmieren mit Banking erheblich erleichtert. Aus diesem Grund wird der Assembler RGBASM und weitere benötigte Programme aus dem RGBDS<sup>10</sup> (Rednex Gameboy Development System) ins GBDK integriert.

Es werden vier Dateien benötigt RGASM<sup>11</sup> in Version 1.11, XLINK<sup>12</sup> Version 1.08d, XLIB<sup>13</sup> und RGBFIX<sup>14</sup>. Nach dem Herunterladen werden die Dateien entpackt und die einzelnen

1 Wikipedia, Die freie Enzyklopädie (Hrsg): „C (Programmiersprache)“, unter: [https://de.wikipedia.org/wiki/C\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/C_(Programmiersprache)) (abgerufen am 08.03.2019)

2 <http://gbdk.sourceforge.net/>, abgerufen am 11.07.2018

3 <https://sourceforge.net/projects/gbdk/files/gbdk/>, abgerufen am 11.07.2018

4 <https://sourceforge.net/projects/gbdk/files/gbdk-win32/2.95-3/>, abgerufen am 11.07.2018

5 Der Autor dieses Artikels selbst benutzt ein Windows 2000 (Lizenzen sind preiswert zu erhalten) in einer virtuellen Maschine unter Linux.

6 Oracle Corporation: „Wie richte ich eine PATH-Systemvariable ein oder ändere diese?“, unter: <https://java.com/de/download/help/path.xml> (abgerufen am 08.03.2019)

7 <http://sdcc.sourceforge.net/>, abgerufen am 11.07.2018

8 Wikipedia, The free Encyclopedia (Publ.): „Bank switching“, unter: [https://en.wikipedia.org/wiki/Bank\\_switching](https://en.wikipedia.org/wiki/Bank_switching) (abgerufen am 11.07.2018)

9 Michael Hope (22. July) : „News“, unter <http://gbdk.sourceforge.net/>, bei News vom 22. Juli (abgerufen am 11.07.2018)

10 Justin Loyd (1999): „REDNEX GAMEBOY DEVELOPMENT SYSTEM“, unter: <http://otakunozoku.com/rednex-gameboy-development-system/> (abgerufen am 11.07.2018)

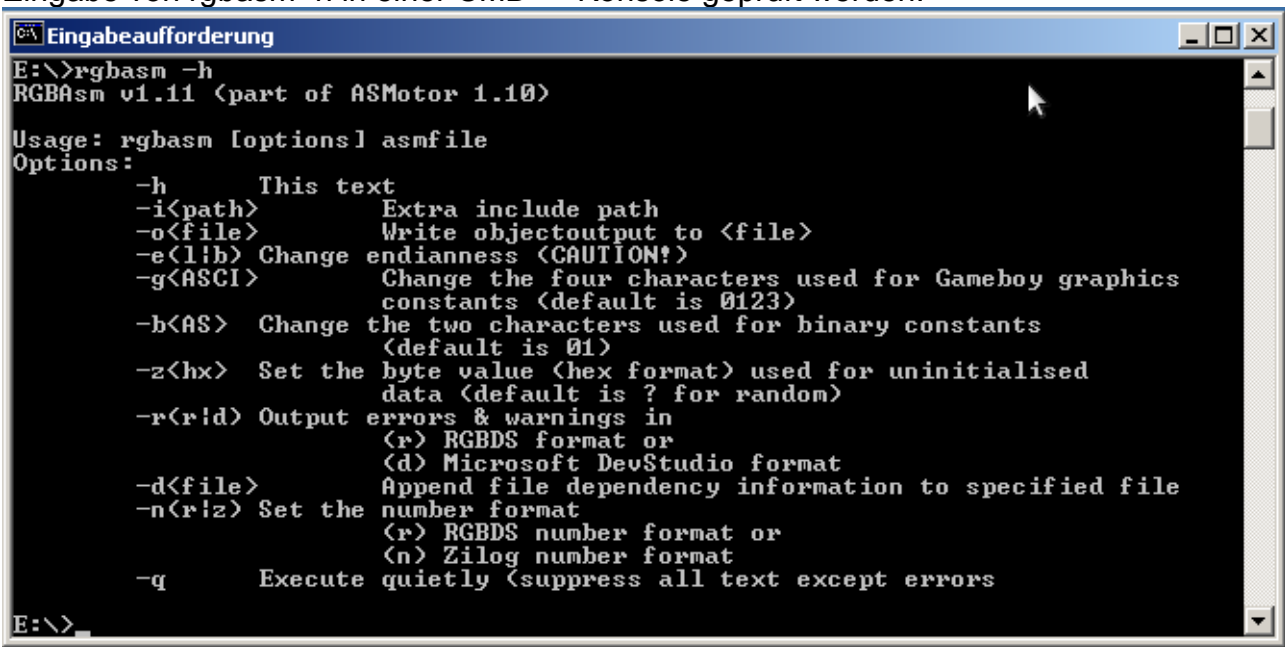
11 <http://otakunozoku.com/wp-content/uploads/2007/05/rgbasm1.zip>, abgerufen am 11.07.2018

12 <http://otakunozoku.com/wp-content/uploads/2007/05/xlink1.zip>, abgerufen am 11.07.2018

13 <http://otakunozoku.com/wp-content/uploads/2007/05/xlib1.zip>, abgerufen am 11.07.2018

14 <http://otakunozoku.com/wp-content/uploads/2007/05/rgbfix1.zip>, abgerufen am 11.07.2018

aufführbaren Dateien: RGBASM.EXE, RGBFIX.EXE, XLIB.EXE und XLINK.EXE ins bin - Verzeichnis von GBDK (z. B. C:\GBDK\bin) kopiert. Ob alles in Ordnung ist, kann durch Eingabe von `rgbasm -h` in einer CMD<sup>15</sup> - Konsole geprüft werden:



```
E:\>rgbasm -h
RGBAsm v1.11 (part of ASMotor 1.10)

Usage: rgbasm [options] asmfile
Options:
  -h          This text
  -i<path>   Extra include path
  -o<file>   Write objectoutput to <file>
  -e<lib>    Change endianness (CAUTION!)
  -g<ASCII>  Change the four characters used for Gameboy graphics
             constants (default is 0123)
  -b<AS>    Change the two characters used for binary constants
             (default is 01)
  -z<hx>    Set the byte value (hex format) used for uninitialised
             data (default is ? for random)
  -r<r|d>    Output errors & warnings in
             (r) RGBDS format or
             (d) Microsoft DevStudio format
  -d<file>   Append file dependency information to specified file
  -n<r|z>    Set the number format
             (r) RGBDS number format or
             (n) Zilog number format
  -q          Execute quietly (suppress all text except errors)

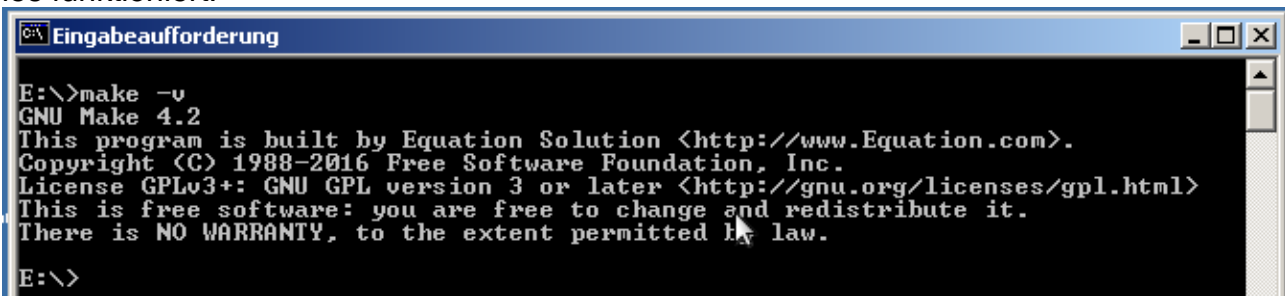
E:\>
```

Bild 2: Die Ausgabe von `rgbasm -h`

## 4 Make

Um die Kompilierung bequemer zu gestalten, wird das Tool `gnu make`<sup>16</sup> empfohlen. Eine Windowsvariante in Version 4.2 gibt es im Netz.<sup>17 18</sup> Nach dem Download wird die entsprechende Datei in das bin - Verzeichnis des GBDK kopiert.

Durch die Eingabe von `make -v` in einem CMD - Fenster kann nachgesehen werden, ob alles funktioniert.



```
E:\>make -v
GNU Make 4.2
This program is built by Equation Solution <http://www.Equation.com>.
Copyright (C) 1988-2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

E:\>
```

Bild 3: Die Ausgabe von `make -v`

## 5 Tools

Weitere sinnvolle Tools sind im Bereich der Gameboy-GBDK-Entwicklung werden hier vorgestellt.

Der Gameboy Tile Designer<sup>19</sup> unterstützt den Entwickler beim Entwickeln von Tiles für den Background-, Window- und Spritelayer. Exportfunktion für GBDK ist grundsätzlich integriert (C-Array<sup>20</sup>). Der generierte Quellcode muss allerdings trotzdem minimal editiert werden, z. B. Bank-Angabe und `const` vor dem Array.

<sup>15</sup> Markus Kasanmascheff: „Windows 10: So startet ihr die Eingabeaufforderung (CMD)“ unter: <https://www.netzwelt.de/tutorial/163585-windows-10-so-startet-eingabeaufforderung-cmd.html> (abgerufen am 08.03.2019)

<sup>16</sup> Wikipedia, Die freie Enzyklopädie (Hrsg): „GNU Make“, unter: [https://de.wikipedia.org/wiki/GNU\\_Make](https://de.wikipedia.org/wiki/GNU_Make) (abgerufen am 08.03.2019)

<sup>17</sup> <ftp://ftp.equation.com/make/32/make.exe> (abgerufen am 11.07.2018)

<sup>18</sup> <ftp://ftp.equation.com/make/64/make.exe> (abgerufen am 11.07.2018)

<sup>19</sup> Harry Mulder (1999): „Gameboy Tile Designer“, unter: <https://www.devrs.com/gb/hmgd/gbtd.html> (abgerufen am 11.02.2019)

<sup>20</sup> Elias Fischer: „C-HowTo“, unter: <http://www.c-howto.de/tutorial/arrays-felder/> (abgerufen am 11.02.2019)

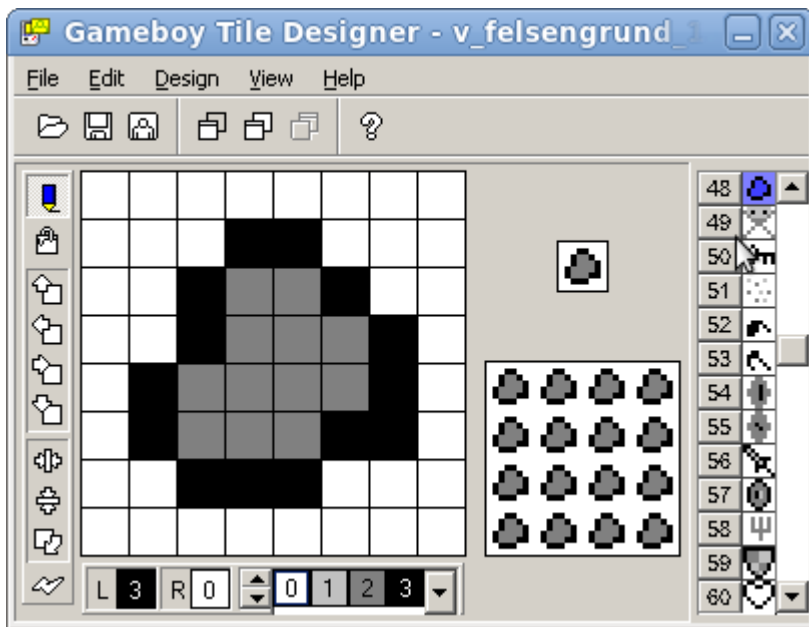


Bild 4: Der Gameboy Tile Designer

Die erstellten Tilesets können mit dem Gameboy Map Builder<sup>21</sup> zur Erstellung von Tilemaps weiterverarbeitet werden. Auch bei diesem Tool gibt es eine Exportfunktion für GBDK (C-Array). Auch hier muss der automatisch erstellte Quellcode etwas bearbeitet werden (z. B. const vor dem Array).

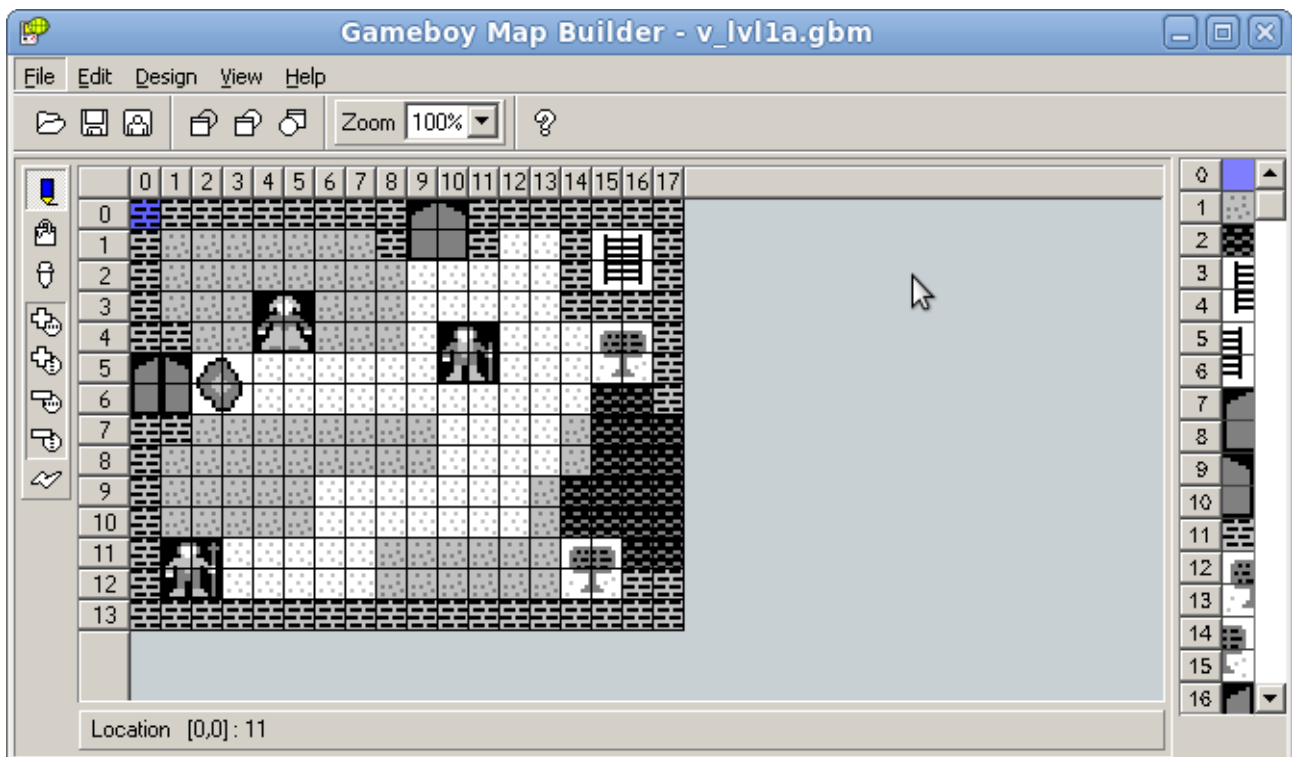


Bild 5: Der Gameboy Map Builder

Beide Tools laufen auch mit Wine<sup>22 23</sup>.

Der BGB<sup>24</sup> ist ein recht genauer Emulator für die Gameboy-Entwicklung. Er bietet nützliche Funktionen wie integrierten Debugger, Videorambetrachter und IO-Map. Der Emulator ist

21 Harry Mulder (1999): „Gameboy Map Designer“, unter: <https://www.devs.com/gb/hmgd/gbmb.html> (abgerufen am 11.02.2019)

22 <https://www.winehq.org/> (abgerufen am 11.02.2019)

23 praktisch im Einsatz mit Version 3.0.x unter Scientific Linux 7.x, Probleme gibt es mit der Hilfe

24 <http://bgb.bircd.org/> (abgerufen am 11.02.2019)

ebenfalls lauffähig mit Wine<sup>25</sup>.

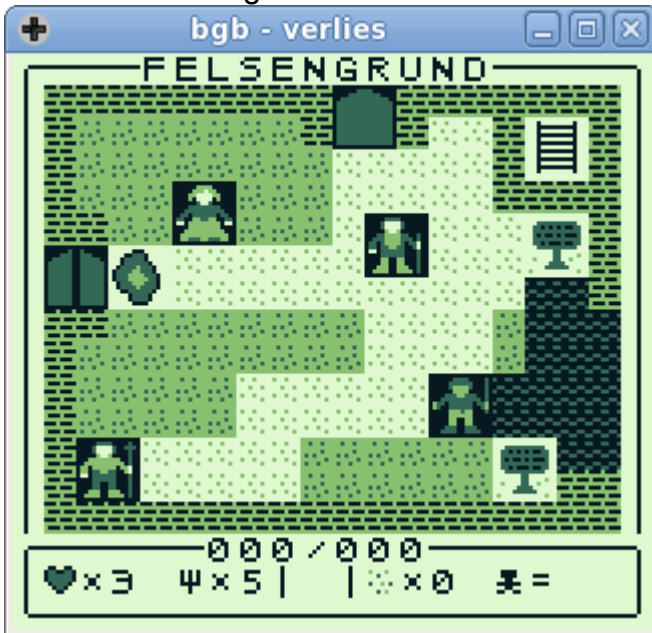


Bild 6: Emulator BGB

## 5 Dokumentationen und Informationsquellen

Die Hauptdokumentation des GBDK-Toolkits ist unter<sup>26</sup> zu finden.

Für Diskussionen und den Austausch existieren einige Chats, z. B.

IRC: efnet<sup>27</sup>, Channel: #gbdev

Discord: gbdev<sup>28</sup> (hat auch eine Verlinkung zum IRC-Channel)

Kleine Diskussionsforen zum Thema existieren ebenfalls im Internet<sup>29 30</sup>.

GBDK wird allerdings in der Community nicht positiv angesehen, daher sollte der Fragesteller eine gewisse Vorsicht walten lassen. Die Assemblerprogrammierung ist bei der Gameboyspieleentwicklung vorherrschend<sup>31</sup>.

Unter dem Github-Repo Awesome Game Boy Development<sup>32</sup> wird eine Liste zu Dokumentationen, Programmen und Quellcode rund um die Gameboy-Entwicklung gepflegt.

Unter der ISBN 3-8311-3087-6<sup>33</sup> existiert ein Buch, welches die Spielentwicklung auf dem Gameboy mit dem GBDK als Thema hat. Ein weiteres Buch ist unter der ISBN 3-7723-4205-1<sup>34</sup> zu finden, welches ebenfalls GBDK-Programmierung verwendet. Das primäre Thema dieses Buches ist allerdings die Hardware. Die beiden Bücher sind aktuell nur noch gebraucht zu bekommen.

Ein Kurs in deutscher Sprache ist unter<sup>35</sup> zu finden.

---

25 Praktisch im Einsatz BGB Version 1.5.6 (32bit), Wineversion 3.0.x, GDI-Grafikausgabe, Stand 11.02.2019

26 <http://gbdk.sourceforge.net/doc/html/index.html> (abgerufen am 11.02.2019)

27 <http://www.efnet.org/?module=servers> (abgerufen am 11.02.2019)

28 <https://discord.gg/D4CPzeU> (abgerufen am 11.02.2019)

29 <http://gbdev.gg8.se/forums/> (abgerufen am 11.02.2019)

30 <https://forums.nesdev.com/viewforum.php?f=20&sid=51261d2b89a1208e6b161ed332553b41> (abgerufen am 12.02.2019)

31 Eldred Habert: „to\_c\_or\_not\_to\_c“, unter: <https://gist.github.com/ISSOtm/4f4d335c3fd258ad0dfc7d4d615409fd> (abgerufen am 09.03.2019)

32 <https://github.com/gbdev/awesome-gbdev#emulator-development> (abgerufen am 11.02.2019)

33 Andreas Stephan Mank: Einführung in die Spieleprogrammierung, Wendel 2001

34 Michael Sebastian Mühlhaus: Messen und Steuern mit dem Game Boy, 85586 Poing 2003 Franzis' Verlag GmbH

35 Gerd Bartel: „Programmierkurs Gameboy (Color)“, unter: <http://www.sebulli.com/gbkurs/kurs.html> (abgerufen am 08.03.2019)

Quellcodes von GBDK-Projekten sind z. B. bei Github<sup>36</sup> zu entdecken.

## **6 Abschluss**

Der Autor bietet ein fertiges Archiv<sup>37</sup> mit alle hier genannten Programmen an. Das Archiv muss nur entpackt und der Ordner zur PATH - Variable von Windows hinzugefügt werden.

Es empfiehlt sich, lokale Kopien der angesprochenen Aspekte und selbst Gefundenes anzulegen. Es ist nicht garantiert, dass dieses historische Wissen (wozu Retro nun mal gehört) im Netz immer an der entsprechenden Stelle wieder auffindbar ist.

---

<sup>36</sup> <https://github.com/topics/gbdk> (abgerufen am 08.03.2019)

<sup>37</sup> <https://der-magister.de/software/spieleentwicklung/gameboy/gbdk.7z> (abgerufen am 09.03.2019)