

model

April 22, 2023

```
[ ]: !pip install tflearn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: tflearn in /usr/local/lib/python3.9/dist-packages
(0.5.0)
Requirement already satisfied: Pillow in /usr/local/lib/python3.9/dist-packages
(from tflearn) (8.4.0)
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages
(from tflearn) (1.16.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages
(from tflearn) (1.22.4)
```

```
[ ]: import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()

import numpy
import tflearn
import tensorflow as tf
import random
```

```
WARNING:tensorflow:From /usr/local/lib/python3.9/dist-
packages/tensorflow/python/compat/v2_compat.py:107: disable_resource_variables
(from tensorflow.python.ops.variable_scope) is deprecated and will be removed in
a future version.
Instructions for updating:
non-resource variables are not supported in the long term
```

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

```
[ ]: import json
with open('/content/drive/MyDrive/intents.json') as file:
    kb = json.load(file)
```

```
[ ]: print(kb)
```

```
{'intents': [{'tag': 'greeting', 'patterns': ['Hi', 'How are you', 'Is anyone  
there?', 'Hello', 'Good day'], 'responses': ['Hello, thanks for visiting', 'Good  
to see you again', 'Hi there, how can I help?'], 'context_set': ''}, {'tag':  
'thanks', 'patterns': ['Thanks', 'Thank you', "That's helpful"], 'responses':  
['Happy to help!', 'Any time!', 'My pleasure']}, {'tag': 'gameboy', 'patterns':  
['gameboy', 'Gameboy', 'game boy', 'Game Boy', 'Game-Boy', 'gbc', 'gba'],  
'responses': ['creating photo realistic images with neural networks and a  
gameboy camera the game boy printer - an in-depth technical document about the  
printer hardware , the communication protocol and the usual routine that games  
used for implementing the print feature .', 'game boy cartridge pcb photos #  
custom cartridges emulating a gameboy cartridge - emulating the functionality  
of a game boy cartridge with the development board stm32f4 .', 'homebrew-  
gameboy-cartridge - eagle library , schematic , and board files for a cartridge  
pcb using an atmel at49f040 as rom .', 'homebrew gameboy color cartridge -  
board layout for an eeprom powered cartridge .', 'gameboy-mbc5-mbc1-hybrid -  
cpld implementation of a mbc5/mbc1 hybrid cartridge .', "differences between the  
z80 and the gameboy 's processor gameboy 2bpp graphics format - information on  
how the game boy interprets vram tile data to color pixels .", 'dmg-01 - an  
educational gameboy emulator in rust and a companion book explaining its  
development .', 'why did i spend 1.5 months creating a gameboy emulator ?',  
'decoding gameboy z80 opcodes - how to algorithmically decode game boy  
instructions ( as opposed to writing one huge switch-case statement ) .',  
'gameboy-linkup - game boy linkup serial cable networking project .',  
'soundsystem - a modernized audio driver for gameboy tracker ( aka the paragon  
5 music player ) .', 'gameboy programming tutorial : hello world !',  
"deadscroll - a detailed tutorial on how to make the screen wobble , among  
other `` raster effects ' ' # timings nitty gritty gameboy cycle timing mode3  
sprite timing gameboy color dma-transfers v0.0.1 stat interrupt timings video  
timing # boilerplates and libraries rgbds-template - basic hello-world example  
for game boy using rgbds .', 'gameboy boilerplate - boilerplate project to move  
quicker into the actual assembly code for your game .', "dangan # c flappyboy  
flappybird-gameboy fbgb novascape squishy the turtle quadratino doctor how  
super princess ' 2092 exodus - ( zgb engine ) .", 'reverse engineering a  
gameboy rom with radare2 - a walkthrough to reverse engineer a game boy rom  
challenge using radare2 .', 'reverse engineering the gameboy tetris dma  
hijacking - a simple technique that allows you to run custom code in most  
gb/sgb/cgb games , provided you have an ace exploit .', 'to retrieve images from  
the camera ) : arduino gameboy printer emulator - emulate a gameboy printer via  
the gameboy link cable .', 'esp8266 game boy printer - a device that emulates  
the gameboy printer and lets you retrieve images using wifi powered by an  
esp8266 .', 'wifi gbp emulator - a gameboy printer emulator which provides the  
received data over a wifi connection .', 'arduino gameboy - arduino based game  
boy cartridge reader and writer .', 'gameboy-brainfuck - brainf * ck  
interpreter .', 'gbcamcorder - lo-fi portable video recorder using a gameboy  
camera cartridge .', 'gbcartread - read rom , read ram or write ram from/to a
```

gameboy cartridge .', "this site uses the google analytics by monsterinsights plugin v7.0.5 - using analytics tracking - [https : //www.monsterinsights.com/](https://www.monsterinsights.com/) / google analytics by monsterinsights pinch of intelligence blogging artificial intelligence , roland meertens main menu skip to content portfolio about me ai meet vr creating photorealistic images with neural networks and a gameboy camera by roland meertens on february 1 , 2017 < div id= ' ' notebook ' ' class= ' ' border-box-sizing ' ' tabindex= ' ' -1 ' ' > < div id= ' ' notebook-container ' ' class= ' ' container ' ' > < div class= ' ' cell border-box-sizing text_cell rendered ' ' > < div class= ' ' prompt input_prompt ' ' > < /div > < div class= ' ' inner_cell ' ' > < div class= ' ' text_cell_render border-box-sizing rendered_html ' ' > in 1998 nintendo released the gameboy camera .', 'this last example inspired me to take gameboy camera images of faces and turn them into photorealistic images .', 'summary of my experiment back in 1998 , the gameboy camera got the world record as " smallest digital camera " in the guinness book of records .', 'these are some images that were shot with the gameboy camera , and uploaded by random people .', 'and... my face , shot with a gameboy camera .', 'training data unfortunately , there is no training data set with gameboy-camera images of faces together with the real picture of the person .', 'if you look at original gameboy camera images you can see that they create gradients by alternating pixels to give the illusion of more colors .', 'note that if you want to experiment you can change the apply_effect_on_folder function to create images from sketches instead of gameboy camera images .', "http : //mmlab.ie.cuhk.edu.hk/projects/celeba.html test_image_path = '/home/roland/workspace/gameboy_camera/test_images ' picture_dataset = os .", 'thresh_trunc) return img_blend def add_gameboy_camera_effect (image) : img_gray = cv2 .', 'imread (path_file) image = resize_image_by_cropping (image) effect = add_gameboy_camera_effect (image) write_path_original = os .', 'as you can see the random noise on top of the image creates the " gradients " you see in the gameboy camera images that give the illusion of more than 4 colors .', 'hidden is the code that loads 30.000 training images and calculates the mean and standard deviation of the gameboy images and the original images .', 'testdata to test my algorithm i tried to convert the following data using the trained network : testdata from the celebrity dataset images from people i found using google images by typing in " gameboy camera " faces that are in the gameboy camera (also found online) pictures of my face testdata from the celebrity dataset in [8] : for index , generated_image in enumerate (colored_images) : show_progress (source_x [index] , test_y [index] , generated_image) this all looks pretty good to me .', 'images from the gameboy itself when trying to display an empty animation the gameboy camera has several faces it can display warning you that you have to create an animation first .', 'images i took a big problem trying to create color images from my own face was getting them off the gameboy camera .', 'buying the camera was easy , but finding a gameboy printer was impossible .', 'conclusion create photorealistic color images from gameboy camera images is a possibility !', 'update : somebody wanted to try out my network , and sent me an image taken with his gameboy camera .', 'this is the result : the colorized result of an image sent to me taken with his gameboy camera share list .entry-content .entry # post-401 post navigation ← explaining neural machine translation to gala members visiting the

clin 2017 conference - computational linguistics in the netherlands → # posts-
 pagination i love to receive your (anonymous) comments !', '(2) 2 comments
 for " creating photorealistic images with neural networks and a gameboy camera "
 jiangpeng november 20 , 2020 at 2:16 am great , very well , i need programe like
 this .', '[endif] projects recently updated most likes newest project lists
 discover contests courses stack more courses tutorials events hackerspaces
 profiles hackaday.com tindie marketplace sign up log in close 0 % 0 % nekocart -
 cpld gameboy cartridge open-source gameboy flash cartridge using an xilinx cpld
 as mbc wenting zhang following follow project liked like project become a
 hackaday.io member not a member ?', "skip join this project similar projects
 worth following 3.5k views 7 comments 21 followers 17 likes description details
 files 0 components 4 view all logs 3 view all instructions 0 discussion 7 view
 gallery 3.5k 7 21 17 team (1) wenting zhang join this project 's team github
 repo completed project hardware gameboy cpld related lists video games both
 retro and novelty .", 'about us contact hackaday.io give feedback terms of use
 privacy policy hackaday api © 2023 hackaday google tag (gtag.js) yes , delete
 it cancel report project as inappropriate you are about to report the project ``
 nekocart - cpld gameboy cartridge `` , please tell us the reason .', '" |
 thoughts on future decoded 2014 - part 1 : the keynotes how to check for
 visibility on a html control using microsoft coded ui tests recent comments
 archives june 2015 april 2015 november 2014 september 2014 may 2014 march 2014
 categories technical uncategorized meta register log in entries feed comments
 feed wordpress.com search for : peterwynroberts.com menu # menu-toggle widgets #
 sidebar-toggle search # search-toggle .toggles # masthead gameboy programming
 tutorial : hello world !', '.entry-header this tutorial will guide you through
 the steps of programming a rom which , when run using a gameboy emulator , will
 write the words " hello world " on the screen .', 'the first one , gbhw.inc ,
 attaches names to a load of locations in the gameboy ' s memory - meaning that
 we don ' t have to remember exact addresses for all of this stuff .', 'the
 gameboy can store and display two different kinds of images : " tiles " , which
 are 8×8 images ; and " sprites " , which are either 8×8 pixel images , or 8×16
 pixel images .', 'the gameboy can ' t display anything other than images - there
 ' s no text output - so any writing we ' re storing has to be displayed as
 images .', 'tiles the gameboy draws its graphics to a " buffer " , which is then
 displayed on the screen .', 'there are multiple buffers used by the gameboy ,
 but the one we ' ll be worrying about is the background tile buffer .', 'because
 the gameboy only has a 160×144 pixel screen , this means that only part of the
 background can be displayed at a time .', 'they ' re there because of a
 particularly important concept in gameboy programming , called interrupts .',
 "setting up the rom header the next bit of code after that is this : section ``
 start ' ' , home [\$ 0100] nop jp begin rom_headerrom_nombc , rom_size_32kbyte ,
 ram_size_0kbyte on startup , the first thing an actual gameboy does is execute a
 small (256 byte) program , which reads the memory locations from \$ 100 to \$
 14d - an area also known as the ' rom header ' - and uses the information there
 to perform various bits of setup , which i won ' t go into too much detail about
 here because it ' s not particularly important for this program .", 'the three
 arguments we ' ve put in to that macro , rom_nombc , rom_size_32kbyte , and
 ram_size_0kbyte , tell the gameboy that our rom is just a simple , 32 kilobyte

rom - 32 kilobytes is the minimum size for a gameboy rom .', 'the first part , the ' mnemonic ' , tells the gameboy what kind of instruction it ' s executing , while if there are more parts then those are the ' arguments ' or ' parameters ' , which are bits of data the instruction uses .', 'in other words , the mnemonic tells the gameboy what to do , and the parameters tell the gameboy what to do stuff to .', 'it ' s a ' no-op ' instruction : all the gameboy does is wait for 4 cpu cycles , before moving on to the next thing .', 'what it does is set the stack pointer to \$ ffff , which is one more than the highest memory address the gameboy can access .', 'the ' stack ' is the bit of memory the gameboy uses to store things when it ' s running , including arguments , variables , and return addresses - this is something it has in common with most processors .', 'there are two things the gameboy can do with the stack - it can ' push ' a piece of data onto it , and it can ' pop ' the last bit of data added to the stack and use it .', 'setting the stack pointer higher than where the gameboy can access might seem counter-intuitive , but it ' s because of how the gameboy pushes values onto the stack .', 'when a gameboy pushes a piece of data onto the stack , it performs two operations : it subtracts 1 from the stack pointer .', 'to explain that , let ' s take a look at the gameboy ' s cpu .', 'like any cpu , a gameboy cpu has several ' registers ' , which are parts of the processor that can store a value so it can be accessed quickly later on .', 'the gameboy has eight 8-bit registers , which can store values up to 255 , called a , b , c , d , e , f , h and l. there are also two 16-bit registers , called sp and pc .', '\$ ff47 is the memory location that stores the background tile palette , which are the shades of grey used by the gameboy when it draws its graphics .', 'so , first , we make sure that wherever we ' re pulling tile data from actually has the tiles in it , like so : ldhl , tiledata next , we set up de so that it ' s storing the location in ram where tile memory starts on a gameboy .', 'it should then compile and link , and your shiny new rom will appear in the same folder , ready to be run in whatever emulator you like (provided that it runs gameboy games , anyway) .', 'a lot of it was also borrowed from the sample code john harrison wrote for a course in gameboy programming at wichita state university - you can see more about that here .', 'links a course in gameboy programming at wichita state university .', 'everything you always wanted to know about gameboy : a complete gameboy specification , including info about important memory locations , the cpu ' s design , and definitions for the complete z80 instruction set .', 'rgbds tools and vim syntax file example code share this : twitter facebook like this : like loadingentry-content may 11 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80 .entry-meta # post- # # post navigation ← → .nav-links .navigation leave a reply cancel reply enter your comment here ...\nfill in your details below or click an icon to log in : email (required) (address never made public) name (required) website you are commenting using your wordpress.com account .', '[endif] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header gameboy gbdk tutorial rom code !', 'i hope this helps you to get started on your gameboy development journeys !', 'gbdkdemo .zip file , 6kb crayon syntax

```

highlighter v_2.7.2_beta //////////////////////////////////////
//////////////////////////////////// demogame.c// -- -- -- -- --
-- -- -- -- -- //
fairly minimal game demo for getting started with gbdk// by refresh games// --
-- -- -- -- --
-- -- -- -- -- // this is intended to be a base point for
anyone to customise and add to so// they can make their own gameboy games ,
hence the comments !', '// -- -- -- -- --
-- -- -- -- -- // web :
refreshgames.co.uk// twitter : @ refreshgamesuk////////////////////////////////
//////////////////////////////////// # include < rand.h > // include
random functions # include < gb/gb.h > ; // include gbdk function library #
include < gb/hardware.h > ; // include handy hardware references// function
declarationsvoid initgame ( ) ; // initialise our gamevoid updateplayer ( ) ; //
update our playeruint8 collisioncheck ( uint8 , uint8 , uint8 , uint8 , uint8 ,
uint8 , uint8 , uint8 ) ; // simple rect to rect check// variable declarations -
stored in ramuint8 i , j ; // generic looping variableuint8 playerx , playery ;
// player co-ordinatesuint8 ex [ 10 ] , ey [ 10 ] ; // enemy co-ordinatesuint8
lastkeys ; // holds keys for the previous frameuint8 randombkggtiles [ 20 ] ; //
contains random data for our bkg// sprite data - 14 sprites - stored in romconst
unsigned char sprites [ ] = { 0x3c,0x3c,0x42,0x42,0xa9,0xa9,0xa9,0xa9,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x3c,0x3c,0x42,0x42,0x81,0x81,0xed,0xed,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x99,0x81,0x42,0x5a,0x24,0x3c,0x99,0x7e,0x99,0x7e,
0x24,0x3c,0x42,0x5a,0x99,0x81,0x38,0x04,0x7c,0x02,0x5c,0x22,0x5c,0x22,0x5c,0x22,
0x5c,0x22,0x7c,0x02,0x38,0x04,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x1c,0x1c,0x3c,0x3c,0x6c,0x6c,0x0c,0x0c,
0x0c,0x0c,0x7f,0x7f,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x03,0x03,0x1e,0x1e,
0x70,0x70,0x7f,0x7f,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x3e,0x3e,0x07,0x07,
0x43,0x43,0x3e,0x3e,0x00,0x00,0x00,0x00,0x60,0x60,0x6c,0x6c,0x6c,0x6c,0x7f,0x7f,
0x0c,0x0c,0x0c,0x0c,0x00,0x00,0x00,0x00,0x7f,0x7f,0x60,0x60,0x7e,0x7e,0x03,0x03,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3f,0x3f,0x60,0x60,0x7e,0x7e,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x06,0x06,0x0c,0x0c,
0x18,0x18,0x30,0x30,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3e,0x3e,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3f,0x3f,
0x03,0x03,0x7e,0x7e,0x00,0x00 } ; //background data - 4 sprites - stored in
romconst unsigned char mybkgdata [ ] = { 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x00,0x08,0x00,0x36,0x00,
0x08,0x00,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x00,0x50,0x00,
0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x04,0x00,0x0e,0x00,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x50,0x00,0x20,0x00,0x50,0x00,0x00,0x00 } ; //////////////////////////////////
//////////////////////////////////// main functionvoid main ( ) {
initgame ( ) ; // ran once at startup// game loop startwhile ( 1 ) {
updateplayer ( ) ; // update playerhide_win ; // show window layersshow_sprites ;
// hide sprites layersshow_bkg ; // show background layerlastkeys = joypad ( ) ;
// update our keypresseswait_vbl_done ( ) ; // wait for vblank to finish -
ensures 60 frames per second maximum } // game loop end } //////////////////////////////////
//////////////////////////////////// init gamevoid

```

```

initgame ( ) { display_on ; // turns on the gameboy lcdnr52_reg = 0x8f ; // turn
sound onnr51_reg = 0x11 ; // enable sound channelsnr50_reg = 0x1f ; // volume
max = 0x77 , min = 0x00initrand ( div_reg ) ; // seed our
randomizerset_sprite_data ( 0 , 14 , sprites ) ; // store our sprite data at the
start of sprite vramset_bkg_data ( 0 , 4 , mybkgdata ) ; // store our bkg data
at the start of bkg vram - note , the window layer shared the bkg vram by
defaultplayerx = 64 ; // players initial x positionplayery = 64 ; // players
initial y positionset_sprite_tile ( 0,0 ) ; //players sprite tile - 0for ( i=0 ;
i != 10 ; i++ ) { // loop through our 10 enemiesif ( collisioncheck (
playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1 ) { j= 1 ; // if
we collided , set our flag variable , j to 1 } } // adjust our player sprites
frame if they hit somethingswitch ( j ) { // no hitcase 0 : set_sprite_tile (
0,0 ) ; break ; // hit enemydefault : set_sprite_tile ( 0,1 ) ; break ; } } ////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
collision checker - simple rectangle collision checking// returns 1 if
overlappinguint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 <= x2 ) & amp
; & amp ; ( ( x1+w1 ) <= x2 ) & amp ; & amp ; ( y1 <= y2 ) & amp
; & amp ; ( ( y1+h1 ) <= y2 ) ) { return 1 ; } else { return 0 ; } } 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149
150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169
170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198 199
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// demogame.c // -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // fairly minimal game
demo for getting started with gbdk // by refresh games // -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- // this is intended to be a base point for anyone to customise
and add to so // they can make their own gameboy games , hence the comments !',
'// -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- // web : refreshgames.co.uk // twitter
: @ refreshgamesuk
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
# include < rand.h > // include random functions # include < gb/gb.h > ; //
include gbdk function library # include < gb/hardware.h > ; // include handy
hardware references // function declarations void initgame ( ) ; // initialise
our game void updateplayer ( ) ; // update our player uint8 collisioncheck (
uint8 , uint8 , uint8 , uint8 , uint8 , uint8 , uint8 , uint8 ) ; // simple rect
to rect check // variable declarations - stored in ram uint8 i , j ; // generic
looping variable uint8 playerx , playery ; // player co-ordinates uint8 ex [ 10
] , ey [ 10 ] ; // enemy co-ordinates uint8 lastkeys ; // holds keys for the
previous frame uint8 randombkgtiles [ 20 ] ; // contains random data for our bkg

```

```

// sprite data - 14 sprites - stored in rom const unsigned char sprites [ ] = {
0x3c , 0x3c , 0x42 , 0x42 , 0xa9 , 0xa9 , 0xa9 , 0xa9 , 0x81 , 0x81 , 0x91 ,
0x91 , 0x42 , 0x42 , 0x3c , 0x3c , 0x3c , 0x3c , 0x42 , 0x42 , 0x81 , 0x81 ,
0xed , 0xed , 0x81 , 0x81 , 0x91 , 0x91 , 0x42 , 0x42 , 0x3c , 0x3c , 0x99 ,
0x81 , 0x42 , 0x5a , 0x24 , 0x3c , 0x99 , 0x7e , 0x99 , 0x7e , 0x24 , 0x3c ,
0x42 , 0x5a , 0x99 , 0x81 , 0x38 , 0x04 , 0x7c , 0x02 , 0x5c , 0x22 , 0x5c ,
0x22 , 0x5c , 0x22 , 0x5c , 0x22 , 0x7c , 0x02 , 0x38 , 0x04 , 0x00 , 0x00 ,
0x3e , 0x3e , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x3e ,
0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x1c , 0x1c , 0x3c , 0x3c , 0x6c , 0x6c ,
0x0c , 0x0c , 0x0c , 0x0c , 0x7f , 0x7f , 0x00 , 0x00 , 0x00 , 0x00 , 0x3e ,
0x3e , 0x63 , 0x63 , 0x03 , 0x03 , 0x1e , 0x1e , 0x70 , 0x70 , 0x7f , 0x7f ,
0x00 , 0x00 , 0x00 , 0x00 , 0x7f , 0x7f , 0x03 , 0x03 , 0x3e , 0x3e , 0x07 ,
0x07 , 0x43 , 0x43 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x60 , 0x60 ,
0x6c , 0x6c , 0x6c , 0x6c , 0x7f , 0x7f , 0x0c , 0x0c , 0x0c , 0x0c , 0x00 ,
0x00 , 0x00 , 0x00 , 0x7f , 0x7f , 0x60 , 0x60 , 0x7e , 0x7e , 0x03 , 0x03 ,
0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x3f , 0x3f , 0x60 ,
0x60 , 0x7e , 0x7e , 0x63 , 0x63 , 0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 ,
0x00 , 0x00 , 0x7f , 0x7f , 0x03 , 0x03 , 0x06 , 0x06 , 0x0c , 0x0c , 0x18 ,
0x18 , 0x30 , 0x30 , 0x00 , 0x00 , 0x00 , 0x00 , 0x3e , 0x3e , 0x63 , 0x63 ,
0x63 , 0x63 , 0x3e , 0x3e , 0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 ,
0x00 , 0x3e , 0x3e , 0x63 , 0x63 , 0x63 , 0x63 , 0x3f , 0x3f , 0x03 , 0x03 ,
0x7e , 0x7e , 0x00 , 0x00 } ; //background data - 4 sprites - stored in rom
const unsigned char mybkgdata [ ] = { 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x08 , 0x00 , 0x08 , 0x00 , 0x36 , 0x00 , 0x08 , 0x00 , 0x08 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x20 , 0x00 , 0x50 ,
0x00 , 0x20 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x04 , 0x00 , 0x0e ,
0x00 , 0x04 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x50 , 0x00 , 0x20 , 0x00 , 0x50 , 0x00 , 0x00 , 0x00 } ;
/////////////////////////////////////////////////////////////////
// main function void main ( ) { initgame ( ) ; // ran once at startup // game
loop start while ( 1 ) { updateplayer ( ) ; // update player hide_win ; // show
window layer show_sprites ; // hide sprites layer show_bkg ; // show background
layer lastkeys = joypad ( ) ; // update our keypresses wait_vbl_done ( ) ; //
wait for vblank to finish - ensures 60 frames per second maximum } // game loop
end }
/////////////////////////////////////////////////////////////////
// init game void initgame ( ) { display_on ; // turns on the gameboy lcd
nr52_reg = 0x8f ; // turn sound on nr51_reg = 0x11 ; // enable sound channels
nr50_reg = 0x1f ; // volume max = 0x77 , min = 0x00 initrand ( div_reg ) ; //
seed our randomizer set_sprite_data ( 0 , 14 , sprites ) ; // store our sprite
data at the start of sprite vram set_bkg_data ( 0 , 4 , mybkgdata ) ; // store
our bkg data at the start of bkg vram - note , the window layer shared the bkg
vram by default playerx = 64 ; // players initial x position playery = 64 ; //
players initial y position set_sprite_tile ( 0 , 0 ) ; //players sprite tile - 0
for ( i = 0 ; i != 10 ; i ++ ) { // loop through our 10 enemies if (
collisioncheck ( playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1

```



```

) { j = 1 ; // if we collided , set our flag variable , j to 1 } } // adjust our
player sprites frame if they hit something switch ( j ) { // no hit case 0 :
set_sprite_tile ( 0 , 0 ) ; break ; // hit enemy default : set_sprite_tile ( 0 ,
1 ) ; break ; } }
////////////////////////////////////
// collision checker - simple rectangle collision checking // returns 1 if
overlapping uint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 <= ( x2 + w2 ) ) &
& ; & ; ( ( x1 + w1 ) <= x2 ) & ; & ; ( y1 <= ( h2 + y2 )
) & ; & ; ( ( y1 + h1 ) <= y2 ) ) { return 1 ; } else { return 0 ; }
} [ format time : 0.0040 seconds ] .entry-content author rychan posted on 18th
april 2016 18th april 2016 categories uncategorised .entry-footer # post- # # 1
thought on " gameboy gbdk tutorial rom code !", '.comments-area post navigation
previous previous post : a sudden jolt , of gameboy goodness !', 'next next post
: ion - a gameboy action puzzle game in gbdk in development .site-main .content-
area search me do search for : search twitter woos tweets by @ refreshgamesuk
recent posts februarys nes game - coffee run !', '15th january 2023 more
homebrew nes games 8th january 2023 old creative software still rocks 15 years
later in 2022 2nd november 2022 retro game projects update me do 1st november
2022 archives archives select month february 2023 ( 1 ) january 2023 ( 2 )
november 2022 ( 2 ) june 2022 ( 2 ) september 2021 ( 1 ) may 2021 ( 3 ) april
2021 ( 11 ) march 2021 ( 1 ) january 2021 ( 2 ) march 2020 ( 1 ) february 2020 (
2 ) february 2018 ( 1 ) january 2018 ( 4 ) april 2017 ( 2 ) january 2017 ( 1 )
november 2016 ( 1 ) october 2016 ( 2 ) september 2016 ( 2 ) july 2016 ( 1 ) june
2016 ( 2 ) may 2016 ( 2 ) april 2016 ( 3 ) march 2016 ( 3 ) february 2016 ( 15 )
january 2016 ( 3 ) december 2015 ( 6 ) categories 2018 nes dev atari vcs 2600
finished games game jams gbdk gameboy development how to gbdk nes dev tips nes
game development one game a month ponderings uncategorised refresh games side
ads .sidebar .widget-area .site-content about finished games game jams gbdk
gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-
info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 -
2013.01.24 - message : 0', 'welcome to the gameboy home of flavor ( aka .',
'here you will find all my gameboy files and creations .', "if you're
interested in gameboy advance development , check out my gba dev page .", 'one
will work on bw and color gameboy , and the other will only work on gameboy
color .', 'the gameboy color version allows your picture to have more than 256
tiles , while the other version requires you to reduce your number of tiles with
an application like gameman .', "note : if you need `` gameboy image quantizer
'' which was , at one time , called `` gb color tiler , '' go to formula 's page
.", '8/25/00 pocketvoice-drmario.zip this is another patch for the black and
white dr. mario gameboy game.i created this patch so that i could take my
pocketvoice out of my closet .', "8/24/00 dr Mario-cgb.zip this is a patch for the
black and white dr. mario gameboy game.i created this because when i played dr.
mario on my gb smart cart , it would n't colorize the game properly .", "8/23/00
gman039a.zip `` gameman gameboy tile & map optimizer/exporter '' i do n't know
who to credit for this , but i got it from harry mulder 's site .",
'ring=gameboydev the gameboydev webring .', "http : //www.devrs.com/gb/ great
page for everything gameboy http : //www.tripmode.com/gbdev/ jason 's gameboy

```

info and files [http : //www.devrs.com/gb/hmgd/intro.html](http://www.devrs.com/gb/hmgd/intro.html) necessary tools .",
 "this is a mirror of the old `` harry mulder 's gameboy development ' ' site .",
 'http : //www.subport.org/ interesting scene info [http : //www.bung.com.hk/](http://www.bung.com.hk/)
 hardware and competitions [http : //www.formula.nl/](http://www.formula.nl/) some cool gameboy stuff .',
 "check their tools page for `` gameboy image quantizer `` which was previously
 referred to as `` gb color tiler . ''", 'start bravenet.com service code end
 bravenet.com service code counter added 01-28-2000 the gameboy development ring
 [join now | ring hub | random | < < prev | next > >]', 'email * # secondary #
 sidebar prologue crackmes competition getting radare2 installation updating
 playing with gameboy rom epilogue reverse engineering a gameboy rom with radare2
 standard .entry-header prologue a month ago in barcelona i was attending to
 r2con for the first time .', 'let ' s use radare2 to reverse engineer a gameboy
 rom !', 'check it out @ [https : //t.co/g1kyjushze](https://t.co/g1kyjushze) @ radareorg # radare2
pic.twitter.com/7fjob3sdks - itay cohen (@ megabeets_) october 9 , 2017
 playing with gameboy rom this post will describe how i solved simple.gb , a
 gameboy rom challenge written by @ condret .', 'it was actually my first time
 reversing a gameboy rom - and it was awesome !', 'first thing i did was to open
 the binary in radare2 and check for its architecture and format : \$ r2 simple.gb
 - for a full list of commands see ` strings /dev/urandom ` [0x00000100] >
 i~format format ningb [0x00000100] > i~machine machine gameboy the i command
 gives us i nformation about the binary .', 'surprise , surprise , it is a
 gameboy rom - dah .', 'the obvious thing to do is open the rom in an gameboy
 emulator .', 'don ' t hesitate to share your thoughts with me eat veggies
 related .entry-content date october 9 , 2017 tags ctf , gameboy , radare2 ,
 tutorial , writeups comments 9 comments .entry-meta # post- # # post navigation
 a journey into radare 2 - part 2 : exploitation reversing a self-modifying
 binary with radare2 # nav-below 9 thoughts on " reverse engineering a gameboy
 rom with radare2 " condret .comment-author .comment-meta are u using vbam from
 git ?', ' ' | thoughts on future decoded 2014 - part 1 : the keynotes how to
 check for visibility on a html control using microsoft coded ui tests recent
 comments archives june 2015 april 2015 november 2014 september 2014 may 2014
 march 2014 categories technical uncategorized meta register log in entries feed
 comments feed wordpress.com search for : peterwynroberts.com menu # menu-toggle
 widgets # sidebar-toggle search # search-toggle .toggles # masthead gameboy
 .page-header gameboy programming tutorial : hello world !', '.entry-header this
 tutorial will guide you through the steps of programming a rom which , when run
 using a gameboy emulator , will write the words " hello world " on the screen
 .', 'the first one , gbhw.inc , attaches names to a load of locations in the
 gameboy ' s memory - meaning that we don ' t have to remember exact addresses
 for all of this stuff .', 'the gameboy can store and display two different kinds
 of images : " tiles " , which are 8x8 images ; and " sprites " , which are
 either 8x8 pixel images , or 8x16 pixel images .', 'the gameboy can ' t display
 anything other than images - there ' s no text output - so any writing we ' re
 storing has to be displayed as images .', 'tiles the gameboy draws its graphics
 to a " buffer " , which is then displayed on the screen .', 'there are multiple
 buffers used by the gameboy , but the one we ' ll be worrying about is the
 background tile buffer .', 'because the gameboy only has a 160x144 pixel screen
 , this means that only part of the background can be displayed at a time .',

'they ' re there because of a particularly important concept in gameboy programming , called interrupts .', "setting up the rom header the next bit of code after that is this : section `` start ' , home [\$ 0100] nop jp begin rom_headerrom_nombc , rom_size_32kbyte , ram_size_0kbyte on startup , the first thing an actual gameboy does is execute a small (256 byte) program , which reads the memory locations from \$ 100 to \$ 14d - an area also known as the ' rom header ' - and uses the information there to perform various bits of setup , which i won ' t go into too much detail about here because it ' s not particularly important for this program .", 'the three arguments we ' ve put in to that macro , rom_nombc , rom_size_32kbyte , and ram_size_0kbyte , tell the gameboy that our rom is just a simple , 32 kilobyte rom - 32 kilobytes is the minimum size for a gameboy rom .', 'the first part , the ' mnemonic ' , tells the gameboy what kind of instruction it ' s executing , while if there are more parts then those are the ' arguments ' or ' parameters , which are bits of data the instruction uses .', 'in other words , the mnemonic tells the gameboy what to do , and the parameters tell the gameboy what to do stuff to .', 'it ' s a ' no-op ' instruction : all the gameboy does is wait for 4 cpu cycles , before moving on to the next thing .', 'what it does is set the stack pointer to \$ ffff , which is one more than the highest memory address the gameboy can access .', 'the ' stack ' is the bit of memory the gameboy uses to store things when it ' s running , including arguments , variables , and return addresses - this is something it has in common with most processors .', 'there are two things the gameboy can do with the stack - it can ' push ' a piece of data onto it , and it can ' pop ' the last bit of data added to the stack and use it .', 'setting the stack pointer higher than where the gameboy can access might seem counter-intuitive , but it ' s because of how the gameboy pushes values onto the stack .', 'when a gameboy pushes a piece of data onto the stack , it performs two operations : it subtracts 1 from the stack pointer .', 'to explain that , let ' s take a look at the gameboy ' s cpu .', 'like any cpu , a gameboy cpu has several ' registers ' , which are parts of the processor that can store a value so it can be accessed quickly later on .', 'the gameboy has eight 8-bit registers , which can store values up to 255 , called a , b , c , d , e , f , h and l. there are also two 16-bit registers , called sp and pc .', '\$ ff47 is the memory location that stores the background tile palette , which are the shades of grey used by the gameboy when it draws its graphics .', 'so , first , we make sure that wherever we ' re pulling tile data from actually has the tiles in it , like so : ldhl , tiledata next , we set up de so that it ' s storing the location in ram where tile memory starts on a gameboy .', 'it should then compile and link , and your shiny new rom will appear in the same folder , ready to be run in whatever emulator you like (provided that it runs gameboy games , anyway) .', 'a lot of it was also borrowed from the sample code john harrison wrote for a course in gameboy programming at wichita state university - you can see more about that here .', 'links a course in gameboy programming at wichita state university .', 'everything you always wanted to know about gameboy : a complete gameboy specification , including info about important memory locations , the cpu ' s design , and definitions for the complete z80 instruction set .', 'rgbds tools and vim syntax file example code .entry-content may 11 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80 leave a

comment .entry-meta # post- # # # main # primary # content create a free website or blog at wordpress.com .', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'i hope this helps someone out there who has a similar issue regarding setting up high-score boards on the gameboy as this seems to be a safe an fairly quick method .', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header a sudden jolt , of gameboy goodness !', '.entry-header so , i noticed that gamejolt now allow you to play gameboy roms directly from your browser , that ' s pretty cool i thought , best upload some games i ' ve made to see how it runs then .', 'and , lo and behold , 5 games uploaded and they all seem to be emulated rather well , they even included cgb (gameboy colour) support on there also .', 'view , play & download these lil gameboy beasties

here !', '.entry-content author rychan posted on 9th april 2016 9th april 2016 categories finished games , gbdk gameboy development .entry-footer # post- # # post navigation previous previous post : why the long delay ?', 'next next post : gameboy gbdk tutorial rom code !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header ion - a gameboy action puzzle game in gbdk in development .entry-header despite not having many hour last weekend for ludum dare , i started to put together the pieces for another lil gameboy game called ion .', '.entry-content author rychan posted on 6th may 2016 categories uncategorised .entry-footer # post- # # post navigation previous previous post : gameboy gbdk tutorial rom code !', 'next next post : formula racing gameboy homebrew developments .', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of

a few things of late , the usual combination of commitments has shifted my focus around a lot .', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'i hope this helps someone out there who has a similar issue regarding setting up high-score boards on the gameboy as this seems to be a safe an fairly quick method .', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitterly woo tweets by @ refreshgamesuk recent posts februrarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'i hope this helps someone out there who has a similar issue regarding setting up high-score boards on the gameboy as this seems to be a safe an fairly quick method .', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really

happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far
 from it .', '.entry-content author rychan posted on 1st september 2016
 categories game jams , gbdk gameboy development .entry-footer # post- # # posts
 navigation page 1 page 2 ... page 7 next page .site-main .content-area search me
 do search for : search twitter woos tweets by @ refreshgamesuk recent posts
 februarys nes game - coffee run !', '15th january 2023 more homebrew nes games
 8th january 2023 old creative software still rocks 15 years later in 2022 2nd
 november 2022 retro game projects update me do 1st november 2022 archives
 archives select month february 2023 (1) january 2023 (2) november 2022 (2)
 june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021
 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1
) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1)
 october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016
 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3)
 december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game
 jams gbdk gameboy development how to gbdk nes dev tips nes game development one
 game a month ponderings uncategorised refresh games side ads .sidebar .widget-
 area .site-content about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation refresh games privacy policy / .site-info .site-footer .site-
 inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',
 "sign in hubringinfo.html the gameboy development ring home > computers &
 internet > software > development > game programming hubnavsection.html about
 this ring dedicated to the development and programming of games , applications ,
 and hardware using nintendo 's gameboy , and gameboy advance handheld gaming
 systems and/or gameboy emulators .", "ringmaster · gbdevring · contact
 ringmaster ring stats · founded on 12/31/1996 · 62 sites hub1_text.html sites in
 this ring rs.html in this ring all webring navbar.html showing : 1-20 of 62 next
 20 > sitelist.html hell child games by killerslut its my little company , which
 i 'm using for my gameboy games i 'm hoping to produce .", 'gameboy cartridge
 guts by gameboy cartsuk photos (decent quality) of the insides of all my
 gameboy carts .', 's i x n i n e f i v e f i v e gameboy music by dj6955
 6955-making music with a gameboy as the sole instrument .', 'gameboy rawk !',
 "dreamer 's dreamscape by gbdevring webring maintainer & general gameboy
 development site gameboy tech central gameboy hardware & software info .",
 'wzonka-lad , gameboy emulator for amiga machines is also under these pages .',
 'the free vengeance group hosting the firts ever gameboy camera film festival
 ...\nsubmit , view and vote .', 't : h : e : o : r : ys website on my site you
 will find things under the topics of : art , music , programming and gameboy
 colour programming .', "stephen 's gameboy page by sgstair this site aims to
 bring the resources of assembly game boy development into one , findable area
 .", "old bits some example gameboy code : - rand () implementations in assembly
 , now used in gbdk- conway 's life- sub-pixel accuracy scrolling on the cgb herg
 's gameboy development contains the games i have written for the gameboy
 including gbdk source code .", 'gamejack help by reubenhelms my experiences with
 using the ufo gb gamejack gameboy online by ldstyler this is an ezboard .', 'it
 is dedicated to gameboy development and has the rarest programs , software and
 information .', 'gameboy developement tutorials my gameboy developement site
 will soon contain many thorough tutorials on making games on the gameboy color

gamepire its a site which hasnt been finished yet about gameboy .', 'gbdev
gbdev.co.uk , the new website for beginner gameboy developers - by a beginner
freestylez we are game boy developers that hav been working within the industry
for many years.our site is not yet fully functional , but will feature
information concerning the cgb , and the products we are completing for it .']],
{'tag': 'game', 'patterns': ['game', 'make a game', 'create a game', 'code a
game'], 'responses': ["[[gbdev.io [] [[] games [] [] pan docs []
community community [[] chat [] [] contribute [] [] gb competition 2021 []
[] [] resources [] guides guides [[] choosing development tools [] [] asm
style recomendations [] [] the timing of lyc stat handlers [] [] dead c
scroll [] [] the wonders of prehistorik man [] [] dma hijacking [] [] []
[[[] games [] [] pan docs [] community community [[] chat [] []
contribute [] [] gb competition 2021 [] [] [] resources [] guides guides [[]
[] choosing development tools [] [] asm style recomendations [] [] the timing
of lyc stat handlers [] [] dead c scroll [] [] the wonders of prehistorik
man [] [] dma hijacking [] [] [] [resources [[] introduction [] [[]
the game boy , a hardware autopsy [] [] the ultimate game boy talk [] [] []
documentation [] [[] game boy color [] [] hardware [] [] peripherals []
[] cartridges [] [] [] emulator development [] [[] testing [] [] []
software development [] [[] assemblers [] [] compilers [] [] emulators []
[] [] tools [] [] [] programming [] [[] asm [] [] c [] [] [] homebrews []
[[] asm [] [] c [] [] gb studio [] [] demos [] [] [] reverse
engineering [] [[] game disassemblies [] [] [] game boy camera [] [[]
retrieving images [] [] changing the camera 's behavior [] [] post
processing [] [] [] related projects [] [[] directories [] [] websites []
[] [] [] [] [] [] awesome game boy development a curated list of awesome
game boy (color) development resources , tools , docs , related projects and
homebrews .", '# introduction # the game boy , a hardware autopsy # the
ultimate game boy talk # disambiguation # game boy advance game boy advance
development is covered by another project , the awesome-gbadev list .', 'gba ,
however , can run gb/gbc games .', '# game boy color and super game boy this
list is focused on the original (1989) game boy (dmg) , the game boy color (gbc)
and super game boy (sgb) are very similar systems , with a few important
distinctions , such as : different hardware specifications specific hardware and
software features specific registers specific bugs , quirks and exploitable
behaviours if you aim to develop your software for sgb or gbc , or you want to
know how it runs on the other systems , you may want to take advantage and adapt
to these differences , check the game boy color category and look for specific
references to gbc/cgb and sgb .', '# documentation pan docs - the single , most
comprehensive technical reference to game boy available to the public .', 'the
cycle-accurate game boy docs - a precise documentation by antoniond to make a
cycle-accurate game boy emulator .', 'game boy project report - report of an
hardware emulator (on a terasic del-soc board) developed as final project for
the csee4840 embedded systems design course at columbia university .', '# game
boy color bootstrap rom unused palettes colorization palettes in the bios rom
disassembly # hardware dmg schematics - hardware schematics .', 'the game boy
project - provides a study on the hardware and detailed constructional
information for the implementation of three 8-bit bidirectional parallel ports

.', 'esp8266 gb dev board - dev board for game boy accessories development ,
 powered by esp8266 .', 'game boy hardware database - data and photos of various
 types of game boy consoles .', '# peripherals dan docs - obscure game boy
 hardware documentation .', 'edge of emulation , a series of articles about
 emulating and investigating game boy accessories .', 'mobile adapter gb -
 internet connectivity and dlc on the game boy color .', 'the game boy printer
 pocket sonar - a blue cart with built-in sonar hardware .', 'infrared madness
 - infrared communication on the game boy color .', 'game boy 4-player adapter -
 dmg-07 .', 'barcode boy - the first game boy card-scanner .', 'dmg-07 technical
 documentation game boy camera re - documentation about gb camera and tools
 used to reverse engineer it by using arduino .', 'creating photo realistic
 images with neural networks and a gameboy camera the game boy printer - an in-
 depth technical document about the printer hardware , the communication protocol
 and the usual routine that games used for implementing the print feature .',
 'ben heck reverse engineers game boy printer (errata : the used thermal paper
 is expired , 4 colors are actually printable) .', 'arduino game boy printer
 emulator - emulating a game boy printer via the game boy link cable with an
 arduino .', "mobile game boy adapter gb kiss link modem # cartridges antoniond
 's docs - corrected schematics and infos on cartridge header data .", "gekkio
 's game boy cartridge types - an overview on existing cartridge types .",
 "pinout , registers descriptions and vhdl code of some cartridge types on
 tauwasser 's wiki : mbc1 mbc2 mmm01 game boy cartridges schematics -
 schematics for mbc2 and mbc3 types .", 'gb rom list - navigable table of every
 game released with details on their cartridges .', 'game boy cartridge pcb
 photos # custom cartridges emulating a gameboy cartridge - emulating the
 functionality of a game boy cartridge with the development board stm32f4 .',
 'wolf - game boy cartridge with co-processor .', 'homebrew-gameboy-cartridge -
 eagle library , schematic , and board files for a cartridge pcb using an atmel
 at49f040 as rom .', 'homebrew gameboy color cartridge - board layout for an
 eeprom powered cartridge .', "reiner ziegler 's game boy page - commercial and
 homemade programmable cartridges and programming systems .", 'gameboy-
 mbc5-mbc1-hybrid - cpld implementation of a mbc5/mbc1 hybrid cartridge .', '#
 misc introduction to game boy hacking - workshop introducing basic assembly ,
 debugging and reverse engineering .', 'gbsound.txt - a document detailing the
 game boy sound engine .', 'game boy bootrom - commented dump of the dmg bootrom
 .', "differences between the z80 and the gameboy 's processor gameboy 2bpp
 graphics format - information on how the game boy interprets vram tile data to
 color pixels .", '# emulator development reverse engineering fine details of
 game boy hardware - 43 minutes talk by gekkio given at disobey 2018 (errata)
 .', 'emulation of nintendo game boy - overview of the game boy hardware with
 the perspective of building an emulator .', 'dmg-01 - an educational gameboy
 emulator in rust and a companion book explaining its development .', 'creating a
 game boy emulator in rust - is a talk given at rust fest 18 about this .',
 'building a game boy emulator in javascript - step by step tutorial .',
 'writing a game boy emulator , cinoop 0dmg - learning rust by building a
 partial game boy emulator .', 'why did i spend 1.5 months creating a gameboy
 emulator ?', 'decoding gameboy z80 opcodes - how to algorithmically decode game
 boy instructions (as opposed to writing one huge switch-case statement) .',

'porting a go game boy emulator to webassembly about swotgb - notes about the development of a game boy emulator in javascript .', "list of open source emulators # testing blargg 's test roms gekkio 's test roms samesuite mealybug tearoom tests gb accuracy tests 144p test suite - port of artemio urbina 's 240p test suite to the game boy .", '# software development the choosing tools for game boy development essay provides an overview of the available development tools for game boy .', 'asmotor - assembler engine and development system targeting game boy , among other cpus .', '# compilers gbdk - maintained and modernized gbdk (game boy development kit) powered by an updated version of the sdcc toolchain .', 'api docs : getting started examples documentation , links and tools turbo rascal syntax error - complete suite (ide , compiler , programming language , resource editor) intended for developing games/demos for 8 / 16-bit line of computers , including the game boy and game boy color .', 'wiz - a high-level assembly language for writing homebrew on retro console platforms (game boy , nes , atari 2600 , and more) .', 'gbforth - a forth-based game boy development kit .', 'gbasm-rs - an opinionated rust based compiler for game boy z80 assembly code .', 'gbasm - a javascript based compiler for game boy z80 assembly code .', 'mgba - modern cross platform gba emulator which also runs gb/gbc games .', 'metroboy - a playable , circuit-level simulation of an entire game boy .', 'complete list of open source emulators # tools # engines zgb - a little engine for creating games for the original game boy (expands gbdk , more info here) .', 'retr0 gb - an engine for creating games (expands gbdk) .', 'gbdk-lib-extension - a small set of sources and tools for the game boy development kit by michael hope .', 'dot matrix game editor - an ide for game boy programming in a c-like language called gbl , with many other features like tile and map extraction , wla-dx assembly , and more .', 'mgbdis - game boy rom disassembler with rgbds compatible output .', 'romusage - command line tool for estimating usage (free space) of game boy roms from a .map , .noi or ihx file .', 'awake - game boy decompiler .', 'game boy text tools - set of tools for text manipulation and translation of game boy roms written in node.js .', 'evscript - a scripting language for the game boy , useful for enemy ai , dialogue , animations , and coroutines .', '# graphics utilities game boy tile data generator - html5 / js web application that will convert bitmap images to hexadecimal data appropriate for use in tile based graphical applications , specifically gb .', "harry mulder 's gb development - some sources and home of game boy tile designer (gbtd) and game boy map builder (gbmb) tools .", "gbtiles - converts .gbr files created with harry mulder 's tile designer (gbtd) and .gbm files created with harry mulder 's map builder (gbmb) to different formats for use with the game boy and gbdk .", 'bmp2cgb - graphics converter for game boy color development providing real time palette adjustments .', 'png2gb - cli tool to convert image file to game boy .c array .', 'gb-convert - game boy tile conversion and map editor tool (converts to assembly) .', 'libstdgb - a c library of useful game boy operations (sdcc) .', 'tilemap studio - a tilemap editor for game boy , color , advance , and snes projects .', 'superfamiconv - flexible and composable tile graphics converter supporting super nintendo , game boy , game boy color , game boy advance , mega drive and pc engine formats .', '# hardware and rom utilities cart-dumper - game boy cartridge dumper rom .', 'gbcamextract

- extracts photos from game boy camera saves .', "game boy lcd sniffing - sniff your game boy 's lcd using a logic analyzer .", 'swapdump - diagnostic utility for game boy flashcarts .', 'gameboy-linkup - game boy linkup serial cable networking project .', 'mmlgb - a mml parser and gbdk sound driver for the nintendo game boy .', 'xpmck - an mml based music compiler with support for game boy & game boy color .', 'soundsystem - a modernized audio driver for gameboy tracker (aka the paragon 5 music player) .', 'hugetracker - a music tracker based on openmpt , focused on ease of use , compact output , and embeddability in homebrew games .', '# programming guides , tutorials and tools to develop software for game boy using the development toolchains described in the software development chapter .', 'hardware.inc - standard include file containing game boy hardware definitions for use in rgbds projects .', 'assemblydigest - exploring game boy programming techniques : making an empty game boy rom (in wiz) making art for the game boy beginner 's guide to reverse engineering gb - some starting tips on disassembling and reverse engineering .', 'flappyboy : making a simple game boy game super game boy development - step by step tutorial to implement super game boy features (frame and palettes) .', 'gameboy programming tutorial : hello world !', 'dmgreport - game programming tutorials in assembly .', 'game boy assembly programming for the modern game developer - an e-book about making game boy games in assembly .', '# sources fragments of code , effects , proof of concepts and generally non complete games .', "deadscroll - a detailed tutorial on how to make the screen wobble , among other `` raster effects " # timings nitty gritty gameboy cycle timing mode3 sprite timing gameboy color dma-transfers v0.0.1 stat interrupt timings video timing # boilerplates and libraries rgbds-template - basic hello-world example for game boy using rgbds .", 'game boy assembly language primer - simple template code with memory defines , copy routines and ibm font tilemap .', 'bootstrap.gb - an example game boy project .', 'gameboy boilerplate - boilerplate project to move quicker into the actual assembly code for your game .', 'gingerbread - a software library for making your own game boy games .', 'it is made to be used alongside the book game boy assembly programming for the modern game developer which also doubles as documentation .', 'gb-boilerplate - a template for starting game boy projects , providing a makefile for infrastructure .', 'vim syntax file for the game boy assembler rgbasm - vim syntax highlighting for rgbds assembly .', '# c 8-bit wonderland - well-written introductory document about how the game boy works and how to start developing working code for it .', 'grooves game boy programming - a complete set of lessons about implementing various game mechanics in a game boy game .', 'how to write a simple side scrolling game - old (but still relevant) tutorial .', 'just another simple tutorial gbdk tutorial - fairly minimal game demo for getting started with gbdk .', 'gbdk color - extends your knowledge of basic spriting on the game boy by adding colors to sprites , backgrounds and the window layer .', 'game boy home of flavor - some full games and sources .', '# homebrews complete and open source games .', 'homebrew hub - every unofficial homebrew ever produced for game boy playable online (mobile/touch too) : a community-lead attempt to collect , archive and save every unofficial game , homebrew , demo , patch , hackrom for game boy (color) produced by the community through years of passionate work

.', '# asm tuff 2048-gb snake lazerpong geometrix pcity carazu snake-gb
 gb303 - gb303 wavetable-based tb-303 style synthesizer for the nintendo game
 boy .', 'adjustris exeman aevilia gbslides - a simple game boy powerpoint-
 like slides viewer .', 'plantboy death planet quartet - puzzle game for the
 game boy (color) and super game boy .', "dangan # c flappyboy flappybird-
 gameboy fbgb novascape squishy the turtle quadratino doctor how super
 princess ' 2092 exodus - (zgb engine) .", 'gbc atari boxing - atari 2600
 boxing clone for the game boy (color) .', 'tobu tobu girl deluxe - an arcade
 platformer for the game boy (color) .', "the mean foxes (gbc port) postbot
 guns & riders dino 's offline adventure - a clone of the google chrome offline
 game .", 'dino-gb - another clone of the chrome game .', 'petris - a puzzle
 game of shapely pets for the game boy color (itch.io) .', 'the game never
 found a publisher and was eventually canceled .', 'black castle - side
 scrolling platformer for the game boy (itch.io) .', 'genesis - shmup for the
 game boy (itch.io) .', ' - side scrolling platformer for the game boy (color
) (zgb engine) .', 'deadeus super impostor bros. # demos back to color
 beach-gbc cute demo 10 print game boy roboto demo matrix-rain-gb - a matrix
 digital rain effect in assembler .', 'gbvideoplayer - a technical demo
 demonstrating how the game boy lcd controller can be hacked to make a game boy
 color play a full motion video in color , together with music .', "# reverse
 engineering reverse engineering kirby 's dreamland 2 pokemontools - a python
 module that provides various reverse engineering components for various pokémon
 games .", 'reverse engineering a gameboy rom with radare2 - a walkthrough to
 reverse engineer a game boy rom challenge using radare2 .', 'reverse engineering
 the gameboy tetris dma hijacking - a simple technique that allows you to run
 custom code in most gb/sgb/cgb games , provided you have an ace exploit .', '#
 game disassemblies pokémon red/blue pokémon crystal pokémon yellow pokémon
 gold and silver pokémon pinball pokémon tcg pokegold-spaceworld - pokémon
 gold and silver 1997 space world demo .', 'fx hammer harvest moon 3 # game boy
 camera # retrieving images game boy printer emulation (e.g .', 'to retrieve
 images from the camera) : arduino gameboy printer emulator - emulate a gameboy
 printer via the gameboy link cable .', 'esp8266 game boy printer - a device
 that emulates the gameboy printer and lets you retrieve images using wifi
 powered by an esp8266 .', 'wifi gbp emulator - a gameboy printer emulator which
 provides the received data over a wifi connection .', 'game boy wifi printer -
 d1 mini shield - game boy printer interface shield for d1 mini/mini pro esp8266
 boards .', 'game boy printer sniffer - sniff packet communications between a
 game boy and the printer .', "# changing the camera 's behavior methods to
 improve and/or manipulate the camera 's quality and behavior : game boy camera
 canon ef lens mount game boy camera to canon lens mount - based on the above
 .", 'game-boy-camera-frame-replacer - manipulate the rom of a camera to include
 custom frames # post processing game boy printer paper simulation - generate
 as-if-printed images of digital printed images .', 'game boy printer web -
 gallery app for to the game boy camera : import pictures from exports or
 cartridge dumps and choose color palettes .', '# related projects gb studio -
 drag and drop game creator with simple , no knowledge required , visual
 scripting .', 'resources to get started dedicated discord arduinoboy - serial
 communication (midi) from an arduino to the game boy for music applications

such as littlesounddj , nanoloop , and mgb .', 'papigb - game boy classic fully functional fpga implementation from scratch .', 'fpgaboy - implementation nintendo 's game boy console on an fpga .', 'piglet - a lua-driven ai that plays classic game boy color games using experimentation .', 'ostrich - a game boy sound system player written in swift .', 'mgb - a game boy cartridge program that enables the game boy to act as a full midi supported sound module .', 'gbvisualizer - demonstrating the use of two undocumented game boy color registers , nicknamed pcm12 (ff76) and pcm34 (ff77) , which can be used to read the current pcm amplitude of the 4 apu channels .', 'arduinogameboy - arduino based game boy cartridge reader and writer .', 'gameboy-brainfuck - brainf * ck interpreter .', 'gb-save-states - patches to add save state support to game boy games when playing on the original hardware .', 'gbcpu - a cpu and peripherals implementing the game boy instruction set and functionality .', 'digitized speech in game boy games sniffing game boy serial traffic with an stm32f4 virtual game boy printer with an stm32f4 printing on the game boy printer using an stm32f4 programming game boy chinese cartridges with an stm32f4 pokemon pocket computer : - what is it and how to use it to make cheat codes .', 'booting the game boy with a custom logo - bypassing the nintendo logo check .', "making a game boy game in 2017 : a `` sheep it up ! '", 'game boy video effects - some little experiments using the stat interrupt to do funny video manipulations .', 'gbos - a basic operating system for the game boy .', 'game boy link cable breakout board gbcartflasher firmware verilogboy - game boy compatible console verilog rtl implementation .', 'gbcamcorder - lo-fi portable video recorder using a gameboy camera cartridge .', 'gbcartread - read rom , read ram or write ram from/to a gameboy cartridge .', 'gbxcart-rw - a device for reading game roms , save games and restoring saves for gb , gbc and gba carts from your pc via usb .', 'dumping the super game boy boot rom # directories archive of related files the game boy archive - a library of game boy related software , hardware and literature .', 'the game boy archive - salvage - historical archive of software , old articles , faqs and various documents .', 'pdroms.de - game boy releases .', 'handheld underground - unlicensed games , blog posts about game boy , home of the hhugboy emulator .', "this site uses the google analytics by monsterinsights plugin v7.0.5 - using analytics tracking - [https : //www.monsterinsights.com/](https://www.monsterinsights.com/) / google analytics by monsterinsights pinch of intelligence blogging artificial intelligence , roland meertens main menu skip to content portfolio about me ai meet vr creating photorealistic images with neural networks and a gameboy camera by roland meertens on february 1 , 2017 < div id= '' notebook '' class= '' border-box-sizing '' tabindex= '' -1 '' > < div id= '' notebook-container '' class= '' container '' > < div class= '' cell border-box-sizing text_cell rendered '' > < div class= '' prompt input_prompt '' > < /div > < div class= '' inner_cell '' > < div class= '' text_cell_render border-box-sizing rendered_html '' > in 1998 nintendo released the gameboy camera .", 'this last example inspired me to take gameboy camera images of faces and turn them into photorealistic images .', 'summary of my experiment back in 1998 , the gameboy camera got the world record as " smallest digital camera " in the guinness book of records .', 'although we did not have the printer , taking pictures , editing them , and playing minigames was a lot of fun .', 'these are some images that were shot with the

gameboy camera , and uploaded by random people .', 'and... my face , shot with a gameboy camera .', 'training data unfortunately , there is no training data set with gameboy-camera images of faces together with the real picture of the person .', 'if you look at original gameboy camera images you can see that they create gradients by alternating pixels to give the illusion of more colors .', 'note that if you want to experiment you can change the apply_effect_on_folder function to create images from sketches instead of gameboy camera images .', 'http : //mmlab.ie.cuhk.edu.hk/projects/celeba.html test_image_path = '/home/roland/workspace/gameboy_camera/test_images ' picture_dataset = os .', 'thresh_trunc) return img_blend def add_gameboy_camera_effect (image) : img_gray = cv2 .', 'imread (path_file) image = resize_image_by_cropping (image) effect = add_gameboy_camera_effect (image) write_path_original = os .', 'as you can see the random noise on top of the image creates the " gradients " you see in the gameboy camera images that give the illusion of more than 4 colors .', 'hidden is the code that loads 30.000 training images and calculates the mean and standard deviation of the gameboy images and the original images .', 'testdata to test my algorithm i tried to convert the following data using the trained network : testdata from the celebrity dataset images from people i found using google images by typing in " gameboy camera " faces that are in the gameboy camera (also found online) pictures of my face testdata from the celebrity dataset in [8] : for index , generated_image in enumerate (colored_images) : show_progress (source_x [index] , test_y [index] , generated_image) this all looks pretty good to me .', 'images from the gameboy itself when trying to display an empty animation the gameboy camera has several faces it can display warning you that you have to create an animation first .', 'images i took a big problem trying to create color images from my own face was getting them off the gameboy camera .', 'buying the camera was easy , but finding a gameboy printer was impossible .', 'conclusion create photorealistic color images from gameboy camera images is a possibility !', 'update : somebody wanted to try out my network , and sent me an image taken with his gameboy camera .', 'this is the result : the colorized result of an image sent to me taken with his gameboy camera share list .entry-content .entry # post-401 post navigation ← explaining neural machine translation to gala members visiting the clin 2017 conference - computational linguistics in the netherlands → # posts-pagination i love to receive your (anonymous) comments !', '(2) 2 comments for " creating photorealistic images with neural networks and a gameboy camera " jiangpeng november 20 , 2020 at 2:16 am great , very well , i need programe like this .', '[endif] projects recently updated most likes newest project lists discover contests courses stack more courses tutorials events hackerspaces profiles hackaday.com tindie marketplace sign up log in close 0 % 0 % nekocart - cpld gameboy cartridge open-source gameboy flash cartridge using an xilinx cpld as mbc wenting zhang following follow project liked like project become a hackaday.io member not a member ?', "skip join this project similar projects worth following 3.5k views 7 comments 21 followers 17 likes description details files 0 components 4 view all logs 3 view all instructions 0 discussion 7 view gallery 3.5k 7 21 17 team (1) wenting zhang join this project 's team github repo completed project hardware gameboy cpld related lists video games both retro and novelty .", 'games , consoles , controllers , gadgets .', 'details a

simple game boy flash cartridge supports mbc1\\5 games .', 'to download games
 via usb and have direct controll over the header and rom from fw .', 'about us
 contact hackaday.io give feedback terms of use privacy policy hackaday api ©
 2023 hackaday google tag (gtag.js) yes , delete it cancel report project as
 inappropriate you are about to report the project `` nekocart - cpld gameboy
 cartridge `` , please tell us the reason .', 'jetpack open graph tags end
 jetpack open graph tags skip to content home about me # site-navigation search
 for : recent posts " there is a missing project subtype... " error on adding a
 windows 10 uap project in monogame 3.4 problem connecting to tfs 2015 rc in
 browser from machine on same domain how to send data to an azure event hub in a
 windows store app " why does my iq feel all small ?', '" | thoughts on future
 decoded 2014 - part 1 : the keynotes how to check for visibility on a html
 control using microsoft coded ui tests recent comments archives june 2015 april
 2015 november 2014 september 2014 may 2014 march 2014 categories technical
 uncategorized meta register log in entries feed comments feed wordpress.com
 search for : recent posts " there is a missing project subtype... " error on
 adding a windows 10 uap project in monogame 3.4 problem connecting to tfs 2015
 rc in browser from machine on same domain how to send data to an azure event hub
 in a windows store app " why does my iq feel all small ?', '" | thoughts on
 future decoded 2014 - part 1 : the keynotes how to check for visibility on a
 html control using microsoft coded ui tests recent comments archives june 2015
 april 2015 november 2014 september 2014 may 2014 march 2014 categories technical
 uncategorized meta register log in entries feed comments feed wordpress.com
 search for : peterwynroberts.com menu # menu-toggle widgets # sidebar-toggle
 search # search-toggle .toggles # masthead gameboy programming tutorial : hello
 world !', '.entry-header this tutorial will guide you through the steps of
 programming a rom which , when run using a gameboy emulator , will write the
 words " hello world " on the screen .', 'the code in this tutorial will all be
 the assembly language used by the game boy processor , which is a slightly
 modified version of the zilog z80 .', 'i also found a vim syntax highlighting
 file for game boy assembler , which you can download from here .', 'the first
 one , gbhw.inc , attaches names to a load of locations in the gameboy ' s memory
 - meaning that we don ' t have to remember exact addresses for all of this stuff
 .', 'the reason we need our letters as tiles is because of how the game boy
 stores and displays graphics .', 'the gameboy can store and display two
 different kinds of images : " tiles " , which are 8×8 images ; and " sprites " ,
 which are either 8×8 pixel images , or 8×16 pixel images .', 'the gameboy can '
 t display anything other than images - there ' s no text output - so any writing
 we ' re storing has to be displayed as images .', 'tiles the gameboy draws its
 graphics to a " buffer " , which is then displayed on the screen .', 'there are
 multiple buffers used by the gameboy , but the one we ' ll be worrying about is
 the background tile buffer .', 'because the gameboy only has a 160×144 pixel
 screen , this means that only part of the background can be displayed at a time
 .', 'they ' re there because of a particularly important concept in gameboy
 programming , called interrupts .', "setting up the rom header the next bit of
 code after that is this : section `` start ' ' , home [\$ 0100] nop jp begin
 rom_headerrom_nombc , rom_size_32kbyte , ram_size_0kbyte on startup , the first
 thing an actual gameboy does is execute a small (256 byte) program , which

reads the memory locations from \$ 100 to \$ 14d - an area also known as the ' rom header ' - and uses the information there to perform various bits of setup , which i won ' t go into too much detail about here because it ' s not particularly important for this program .", 'the three arguments we ' ve put in to that macro , rom_nombc , rom_size_32kbyte , and ram_size_0kbyte , tell the gameboy that our rom is just a simple , 32 kilobyte rom - 32 kilobytes is the minimum size for a gameboy rom .', 'the first part , the ' mnemonic ' , tells the gameboy what kind of instruction it ' s executing , while if there are more parts then those are the ' arguments ' or ' parameters , which are bits of data the instruction uses .', 'in other words , the mnemonic tells the gameboy what to do , and the parameters tell the gameboy what to do stuff to .', 'it ' s a ' no-op ' instruction : all the gameboy does is wait for 4 cpu cycles , before moving on to the next thing .', 'what it does is set the stack pointer to \$ ffff , which is one more than the highest memory address the gameboy can access .', 'the ' stack ' is the bit of memory the gameboy uses to store things when it ' s running , including arguments , variables , and return addresses - this is something it has in common with most processors .', 'there are two things the gameboy can do with the stack - it can ' push ' a piece of data onto it , and it can ' pop ' the last bit of data added to the stack and use it .', 'setting the stack pointer higher than where the gameboy can access might seem counter-intuitive , but it ' s because of how the gameboy pushes values onto the stack .', 'when a gameboy pushes a piece of data onto the stack , it performs two operations : it subtracts 1 from the stack pointer .', 'to explain that , let ' s take a look at the gameboy ' s cpu .', 'like any cpu , a gameboy cpu has several ' registers ' , which are parts of the processor that can store a value so it can be accessed quickly later on .', 'the gameboy has eight 8-bit registers , which can store values up to 255 , called a , b , c , d , e , f , h and l. there are also two 16-bit registers , called sp and pc .', '\$ ff47 is the memory location that stores the background tile palette , which are the shades of grey used by the gameboy when it draws its graphics .', 'so , first , we make sure that wherever we ' re pulling tile data from actually has the tiles in it , like so : ldhl , tiledata next , we set up de so that it ' s storing the location in ram where tile memory starts on a gameboy .', 'it should then compile and link , and your shiny new rom will appear in the same folder , ready to be run in whatever emulator you like (provided that it runs gameboy games , anyway) .', 'a lot of it was also borrowed from the sample code john harrison wrote for a course in gameboy programming at wichita state university - you can see more about that here .', 'links a course in gameboy programming at wichita state university .', 'everything you always wanted to know about gameboy : a complete gameboy specification , including info about important memory locations , the cpu ' s design , and definitions for the complete z80 instruction set .', 'rgbds tools and vim syntax file example code share this : twitter facebook like this : like loadingentry-content may 11 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80 .entry-meta # post- # # post navigation ← → .nav-links .navigation leave a reply cancel reply enter your comment here ... \nfill in your details below or click an icon to log in : email (required) (address never made public) name (required) website you are commenting using your wordpress.com account .', "[if lt ie 10] > < link


```

rel='stylesheet ' id='twentysixteen-ie-css ' href='https :
//refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?', "[ endif ] [
if lt ie 9 ] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https
: //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?', "[ endif
] [ if lt ie 8 ] > < link rel='stylesheet ' id='twentysixteen-ie7-css '
href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css
?', "[ endif ] [ if lt ie 9 ] > < script type='text/javascript ' src='https :
//refreshgames.co.uk/wp-content/themes/twentysixteen/js/html5.js ?', '[ endif ]
analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to
content refresh games gameboy game development in gbdk & other game dev things
.site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation .site-header-menu .site-header-main .header-image .site-header
gameboy gbdk tutorial rom code !', 'i hope this helps you to get started on your
gameboy development journeys !', 'gbdkdemo .zip file , 6kb crayon syntax
highlighter v_2.7.2_beta //////////////////////////////////////
//////////////////////////////////// demogame.c// -- -- -- -- --
-- -- -- -- -- //
fairly minimal game demo for getting started with gbdk// by refresh games// --
-- -- -- -- --
-- -- -- -- -- // this is intended to be a base point for
anyone to customise and add to so// they can make their own gameboy games ,
hence the comments !', '// -- -- -- -- --
-- -- -- -- -- // web :
refreshgames.co.uk// twitter : @ refreshgamesuk////////////////////////////////
//////////////////////////////// # include < rand.h > // include
random functions # include < gb/gb.h > ; // include gbdk function library #
include < gb/hardware.h > ; // include handy hardware references// function
declarationsvoid initgame ( ) ; // initialise our gamevoid updateplayer ( ) ; //
update our playeruint8 collisioncheck ( uint8 , uint8 , uint8 , uint8 , uint8 ,
uint8 , uint8 , uint8 ) ; // simple rect to rect check// variable declarations -
stored in ramuint8 i , j ; // generic looping variableuint8 playerx , playery ;
// player co-ordinatesuint8 ex [ 10 ] , ey [ 10 ] ; // enemy co-ordinatesuint8
lastkeys ; // holds keys for the previous frameuint8 randombkgtiles [ 20 ] ; //
contains random data for our bkg// sprite data - 14 sprites - stored in romconst
unsigned char sprites [ ] = { 0x3c,0x3c,0x42,0x42,0xa9,0xa9,0xa9,0xa9,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x3c,0x3c,0x42,0x42,0x81,0x81,0xed,0xed,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x99,0x81,0x42,0x5a,0x24,0x3c,0x99,0x7e,0x99,0x7e,
0x24,0x3c,0x42,0x5a,0x99,0x81,0x38,0x04,0x7c,0x02,0x5c,0x22,0x5c,0x22,0x5c,0x22,
0x5c,0x22,0x7c,0x02,0x38,0x04,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x1c,0x1c,0x3c,0x3c,0x6c,0x6c,0x0c,0x0c,
0x0c,0x0c,0x7f,0x7f,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x03,0x03,0x1e,0x1e,
0x70,0x70,0x7f,0x7f,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x3e,0x3e,0x07,0x07,
0x43,0x43,0x3e,0x3e,0x00,0x00,0x00,0x00,0x60,0x60,0x6c,0x6c,0x6c,0x6c,0x7f,0x7f,
0x0c,0x0c,0x0c,0x0c,0x00,0x00,0x00,0x00,0x7f,0x7f,0x60,0x60,0x7e,0x7e,0x03,0x03,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3f,0x3f,0x60,0x60,0x7e,0x7e,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x06,0x06,0x0c,0x0c,
0x18,0x18,0x30,0x30,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3e,0x3e,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3f,0x3f,

```

```

0x03,0x03,0x7e,0x7e,0x00,0x00 } ; //background data - 4 sprites - stored in
romconst unsigned char mybkgdata [ ] = { 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x00,0x08,0x00,0x36,0x00
,0x08,0x00,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x00,0x50,0x00
,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x04,0x00,0x0e,0x00,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x50,0x00,0x20,0x00,0x50,0x00,0x00,0x00 } ; //////////////////////////////////////////////////
//////////////////////////////////////////////// main functionvoid main ( ) {
initgame ( ) ; // ran once at startup// game loop startwhile ( 1 ) {
updateplayer ( ) ; // update playerhide_win ; // show window layersshow_sprites ;
// hide sprites layersshow_bkg ; // show background layerlastkeys = joypad ( ) ;
// update our keypresseswait_vbl_done ( ) ; // wait for vblank to finish -
ensures 60 frames per second maximum } // game loop end } //////////////////////////////////
//////////////////////////////////////////////// init gamevoid
initgame ( ) { display_on ; // turns on the gameboy lcdnr52_reg = 0x8f ; // turn
sound onnr51_reg = 0x11 ; // enable sound channelsnr50_reg = 0x1f ; // volume
max = 0x77 , min = 0x00initrand ( div_reg ) ; // seed our
randomizerset_sprite_data ( 0 , 14 , sprites ) ; // store our sprite data at the
start of sprite vramset_bkg_data ( 0 , 4 , mybkgdata ) ; // store our bkg data
at the start of bkg vram - note , the window layer shared the bkg vram by
defaultplayerx = 64 ; // players initial x positionplayery = 64 ; // players
initial y positionset_sprite_tile ( 0,0 ) ; //players sprite tile - 0for ( i=0 ;
i != 10 ; i++ ) { // loop through our 10 enemiesif ( collisioncheck (
playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1 ) { j= 1 ; // if
we collided , set our flag variable , j to 1 } } // adjust our player sprites
frame if they hit somethingswitch ( j ) { // no hitcase 0 : set_sprite_tile (
0,0 ) ; break ; // hit enemydefault : set_sprite_tile ( 0,1 ) ; break ; } } ////
////////////////////////////////////////////////
collision checker - simple rectangle collision checking// returns 1 if
overlappinguint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 <= x2 ) & amp
; & amp ; ( ( x1+w1 ) <= x2 ) & amp ; & amp ; ( y1 <= y2 ) & amp
; & amp ; ( ( y1+h1 ) <= y2 ) ) { return 1 ; } else { return 0 ; } } 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149
150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169
170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198 199
////////////////////////////////////////////////
// demogame.c // -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // fairly minimal game
demo for getting started with gbdk // by refresh games // -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- // this is intended to be a base point for anyone to customise

```



```

layer lastkeys = joypad ( ) ; // update our keypresses wait_vbl_done ( ) ; //
wait for vblank to finish - ensures 60 frames per second maximum } // game loop
end }

////////////////////////////////////
// init game void initgame ( ) { display_on ; // turns on the gameboy lcd
nr52_reg = 0x8f ; // turn sound on nr51_reg = 0x11 ; // enable sound channels
nr50_reg = 0x1f ; // volume max = 0x77 , min = 0x00 initrand ( div_reg ) ; //
seed our randomizer set_sprite_data ( 0 , 14 , sprites ) ; // store our sprite
data at the start of sprite vram set_bkg_data ( 0 , 4 , mybkgdata ) ; // store
our bkg data at the start of bkg vram - note , the window layer shared the bkg
vram by default playerx = 64 ; // players initial x position playery = 64 ; //
players initial y position set_sprite_tile ( 0 , 0 ) ; //players sprite tile - 0
for ( i = 0 ; i != 10 ; i ++ ) { // loop through our 10 enemies if (
collisioncheck ( playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1
) { j = 1 ; // if we collided , set our flag variable , j to 1 } } // adjust our
player sprites frame if they hit something switch ( j ) { // no hit case 0 :
set_sprite_tile ( 0 , 0 ) ; break ; // hit enemy default : set_sprite_tile ( 0 ,
1 ) ; break ; } }

////////////////////////////////////
// collision checker - simple rectangle collision checking // returns 1 if
overlapping uint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 < ( x2 + w2 ) ) &
& ( x1 + w1 > x2 ) & ( y1 < ( h2 + y2 ) ) & ( y1 + h1 > y2 ) ) { return 1 ; } else { return 0 ; }
} [ format time : 0.0040 seconds ]

.entry-content author rychan posted on 18th
april 2016 18th april 2016 categories uncategorised .entry-footer # post- # # 1
thought on " gameboy gbdk tutorial rom code !", " pingback : awesome game boy
development - massive collection of resources - learn practice & share #
comment- # # .comment-list comments are closed .', '.comments-area post
navigation previous previous post : a sudden jolt , of gameboy goodness !',
'next next post : ion - a gameboy action puzzle game in gbdk in development
.site-main .content-area search me do search for : search twitter woos tweets by
@ refreshgamesuk recent posts februarys nes game - coffee run !', '8th february
2023 secret hidden mini game complete !', '15th january 2023 more homebrew nes
games 8th january 2023 old creative software still rocks 15 years later in 2022
2nd november 2022 retro game projects update me do 1st november 2022 archives
archives select month february 2023 ( 1 ) january 2023 ( 2 ) november 2022 ( 2 )
june 2022 ( 2 ) september 2021 ( 1 ) may 2021 ( 3 ) april 2021 ( 11 ) march 2021
( 1 ) january 2021 ( 2 ) march 2020 ( 1 ) february 2020 ( 2 ) february 2018 ( 1
) january 2018 ( 4 ) april 2017 ( 2 ) january 2017 ( 1 ) november 2016 ( 1 )
october 2016 ( 2 ) september 2016 ( 2 ) july 2016 ( 1 ) june 2016 ( 2 ) may 2016
( 2 ) april 2016 ( 3 ) march 2016 ( 3 ) february 2016 ( 15 ) january 2016 ( 3 )
december 2015 ( 6 ) categories 2018 nes dev atari vcs 2600 finished games game
jams gbdk gameboy development how to gbdk nes dev tips nes game development one
game a month ponderings uncategorised refresh games side ads .sidebar .widget-
area .site-content about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation refresh games privacy policy / .site-info .site-footer .site-
inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',

```

'welcome to the gameboy home of flavor (aka .', 'here you will find all my gameboy files and creations .', "if you 're interested in gameboy advance development , check out my gba dev page .", "please post comments/suggestions to the forum .oh , it 's also worth noting that this new `` automatic sram '' should allow you to have more savegames (in most cases) than version 1.1 did .", 'one will work on bw and color gameboy , and the other will only work on gameboy color .', 'the gameboy color version allows your picture to have more than 256 tiles , while the other version requires you to reduce your number of tiles with an application like gameman .', "note : if you need `` gameboy image quantizer '' which was , at one time , called `` gb color tiler , '' go to formula 's page .", '8/25/00 pocketvoice-drmario.zip this is another patch for the black and white dr. mario gameboy game.i created this patch so that i could take my pocketvoice out of my closet .', "8/24/00 drmario-cgb.zip this is a patch for the black and white dr. mario gameboy game.i created this because when i played dr. mario on my gb smart cart , it would n't colorize the game properly .", "8/23/00 gman039a.zip `` gameman gameboy tile & map optimizer/exporter '' i do n't know who to credit for this , but i got it from harry mulder 's site .", 'ring=gameboydev the gameboydev webring .', "http : //www.devrs.com/gb/ great page for everything gameboy http : //www.tripmode.com/gbdev/ jason 's gameboy info and files http : //www.devrs.com/gb/hmgd/intro.html necessary tools .", "this is a mirror of the old `` harry mulder 's gameboy development '' site .", 'http : //www.subport.org/ interesting scene info http : //www.bung.com.hk/ hardware and competitions http : //www.formula.nl/ some cool gameboy stuff .', "check their tools page for `` gameboy image quantizer `` which was previously referred to as `` gb color tiler . ''", 'start bravenet.com service code end bravenet.com service code counter added 01-28-2000 the gameboy development ring [join now | ring hub | random | < < prev | next > >]', 'email * # secondary # sidebar prologue crackmes competition getting radare2 installation updating playing with gameboy rom epilogue reverse engineering a gameboy rom with radare2 standard .entry-header prologue a month ago in barcelona i was attending to r2con for the first time .', 'let ' s use radare2 to reverse engineer a gameboy rom !', 'check it out @ https : //t.co/g1kyjushze @ radareorg # radare2 pic.twitter.com/7fjob3sdks - itay cohen (@ megabeets_) october 9 , 2017 playing with gameboy rom this post will describe how i solved simple.gb , a gameboy rom challenge written by @ condret .', 'it was actually my first time reversing a gameboy rom - and it was awesome !', 'first thing i did was to open the binary in radare2 and check for its architecture and format : \$ r2 simple.gb - for a full list of commands see ` strings /dev/urandom ` [0x00000100] > i~format format ningb [0x00000100] > i~machine machine gameboy the i command gives us i nformation about the binary .', 'surprise , surprise , it is a gameboy rom - dah .', 'the obvious thing to do is open the rom in an gameboy emulator .', 'it ' s a simple game where , by using the arrow keys , you increase/decrease 5 digits .', 'don ' t hesitate to share your thoughts with me eat veggies related .entry-content date october 9 , 2017 tags ctf , gameboy , radare2 , tutorial , writeups comments 9 comments .entry-meta # post- # # post navigation a journey into radare 2 - part 2 : exploitation reversing a self-modifying binary with radare2 # nav-below 9 thoughts on " reverse engineering a gameboy rom with radare2 " condret .comment-author .comment-meta are u using

vbam from git?', 'but it doesn't work if you're dealing with real game boy
 roms since all text is non-ascii and each letter is referencing different
 character table offsets.', 'jetpack open graph tags end jetpack open graph tags
 skip to content home about me # site-navigation search for : recent posts "
 there is a missing project subtype... " error on adding a windows 10 uap project
 in monogame 3.4 problem connecting to tfs 2015 rc in browser from machine on
 same domain how to send data to an azure event hub in a windows store app " why
 does my iq feel all small?', '" | thoughts on future decoded 2014 - part 1 :
 the keynotes how to check for visibility on a html control using microsoft coded
 ui tests recent comments archives june 2015 april 2015 november 2014 september
 2014 may 2014 march 2014 categories technical uncategorized meta register log in
 entries feed comments feed wordpress.com search for : recent posts " there is a
 missing project subtype... " error on adding a windows 10 uap project in monogame
 3.4 problem connecting to tfs 2015 rc in browser from machine on same domain how
 to send data to an azure event hub in a windows store app " why does my iq feel
 all small?', '" | thoughts on future decoded 2014 - part 1 : the keynotes how
 to check for visibility on a html control using microsoft coded ui tests recent
 comments archives june 2015 april 2015 november 2014 september 2014 may 2014
 march 2014 categories technical uncategorized meta register log in entries feed
 comments feed wordpress.com search for : peterwynroberts.com menu # menu-toggle
 widgets # sidebar-toggle search # search-toggle .toggles # masthead gameboy
 .page-header gameboy programming tutorial : hello world !', '.entry-header this
 tutorial will guide you through the steps of programming a rom which , when run
 using a gameboy emulator , will write the words " hello world " on the screen
 .', 'the code in this tutorial will all be the assembly language used by the
 game boy processor , which is a slightly modified version of the zilog z80 .',
 'i also found a vim syntax highlighting file for game boy assembler , which you
 can download from here .', 'the first one , gbhw.inc , attaches names to a load
 of locations in the gameboy 's memory - meaning that we don't have to
 remember exact addresses for all of this stuff .', 'the reason we need our
 letters as tiles is because of how the game boy stores and displays graphics .',
 'the gameboy can store and display two different kinds of images : " tiles " ,
 which are 8x8 images ; and " sprites " , which are either 8x8 pixel images , or
 8x16 pixel images .', 'the gameboy can't display anything other than images -
 there's no text output - so any writing we're storing has to be displayed as
 images .', 'tiles the gameboy draws its graphics to a " buffer " , which is then
 displayed on the screen .', 'there are multiple buffers used by the gameboy ,
 but the one we'll be worrying about is the background tile buffer .', 'because
 the gameboy only has a 160x144 pixel screen , this means that only part of the
 background can be displayed at a time .', 'they're there because of a
 particularly important concept in gameboy programming , called interrupts .',
 "setting up the rom header the next bit of code after that is this : section ``
 start ' , home [\$ 0100] nop jp begin rom_headerrom_nombc , rom_size_32kbyte ,
 ram_size_0kbyte on startup , the first thing an actual gameboy does is execute a
 small (256 byte) program , which reads the memory locations from \$ 100 to \$
 14d - an area also known as the ' rom header ' - and uses the information there
 to perform various bits of setup , which i won't go into too much detail about
 here because it's not particularly important for this program .", 'the three

arguments we ' ve put in to that macro , rom_nombc , rom_size_32kbyte , and ram_size_0kbyte , tell the gameboy that our rom is just a simple , 32 kilobyte rom - 32 kilobytes is the minimum size for a gameboy rom .', 'the first part , the ' mnemonic ' , tells the gameboy what kind of instruction it ' s executing , while if there are more parts then those are the ' arguments ' or ' parameters ' , which are bits of data the instruction uses .', 'in other words , the mnemonic tells the gameboy what to do , and the parameters tell the gameboy what to do stuff to .', 'it ' s a ' no-op ' instruction : all the gameboy does is wait for 4 cpu cycles , before moving on to the next thing .', 'what it does is set the stack pointer to \$ ffff , which is one more than the highest memory address the gameboy can access .', 'the ' stack ' is the bit of memory the gameboy uses to store things when it ' s running , including arguments , variables , and return addresses - this is something it has in common with most processors .', 'there are two things the gameboy can do with the stack - it can ' push ' a piece of data onto it , and it can ' pop ' the last bit of data added to the stack and use it .', 'setting the stack pointer higher than where the gameboy can access might seem counter-intuitive , but it ' s because of how the gameboy pushes values onto the stack .', 'when a gameboy pushes a piece of data onto the stack , it performs two operations : it subtracts 1 from the stack pointer .', 'to explain that , let ' s take a look at the gameboy ' s cpu .', 'like any cpu , a gameboy cpu has several ' registers ' , which are parts of the processor that can store a value so it can be accessed quickly later on .', 'the gameboy has eight 8-bit registers , which can store values up to 255 , called a , b , c , d , e , f , h and l. there are also two 16-bit registers , called sp and pc .', '\$ ff47 is the memory location that stores the background tile palette , which are the shades of grey used by the gameboy when it draws its graphics .', 'so , first , we make sure that wherever we ' re pulling tile data from actually has the tiles in it , like so : ldhl , tiledata next , we set up de so that it ' s storing the location in ram where tile memory starts on a gameboy .', 'it should then compile and link , and your shiny new rom will appear in the same folder , ready to be run in whatever emulator you like (provided that it runs gameboy games , anyway) .', 'a lot of it was also borrowed from the sample code john harrison wrote for a course in gameboy programming at wichita state university - you can see more about that here .', 'links a course in gameboy programming at wichita state university .', 'everything you always wanted to know about gameboy : a complete gameboy specification , including info about important memory locations , the cpu ' s design , and definitions for the complete z80 instruction set .', 'rgbds tools and vim syntax file example code .entry-content may 11 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80 leave a comment .entry-meta # post- # # # main # primary # content create a free website or blog at wordpress.com .', "[if lt ie 10] > < link rel='stylesheet ' id='twentysixteen-ie-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?' , "[endif] [if lt ie 9] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?' , "[endif] [if lt ie 8] > < link rel='stylesheet ' id='twentysixteen-ie7-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css ?' , "[endif] [if lt ie 9] > < script type='text/javascript ' src='https :

```
//refreshgames.co.uk/wp-content/themes/twentyseven/js/html5.js ?", 'skip to
content refresh games gameboy game development in gbdk & other game dev things
.site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation .site-header-menu .site-header-main .header-image .site-header
category : gbdk gameboy development .page-header picking nes & game boy
development back up again .entry-header life has certainly gotten in the way of
a few things of late , the usual combination of commitments has shifted my focus
around a lot .', 'i ' ve not really felt settled at home to get many decent game
development sessions in for quite a while .', 'bee happy ( nes game development
) this game is at that weird stage in development where it ' s fairly playable
but needs some extra elements inside it for a good difficulty progression .',
'versus mode music sound effects transitions between screens / levels an ending
sequence flap happy and fancy free ( game boy & game boy colour dev ) flap happy
is further back in terms of development , i ' ve attempted to convert what i
made to utilize bank switching , but ran into issues .', 'the game is in a
similar position to bee happy and needs the following features to be considered
complete .', 'music sound effects level transitions ( easy fade ) ending
sequence cpu usage is still fine for the game boy which is good news for future
development bits .', 'keep your eyes peeled for updates later on this week for
more details on my retro game development adventures !', '.entry-content author
rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy
development , nes game development .entry-footer # post- # # how high is your
highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as
arrays as scores up to 255 would be rather boring... .', 'crayon syntax
highlighter v_2.7.2_beta c score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; 1 2 3 4 score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; [ format time : 0.0006 seconds ] in a game i ' m making
currently , scores can extend beyond the limits of a 16 bit integer also , so we
' re using slightly longer arrays to store things in , no biggie .', 'i hope
this helps someone out there who has a similar issue regarding setting up high-
score boards on the gameboy as this seems to be a safe an fairly quick method
.', 'i may well be missing another way that ' s quicker all-around but this
seems to work for me anyway expect a new game posted around the end of this
month .entry-content author rychan posted on 19th january 2017 categories gbdk
gameboy development .entry-footer # post- # # so , august and me coding in gbdk
didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy
development , far from it .', 'there ' s plenty of games i need to finish off
before i move onto tackling 6502 asm again in october / november .', 'it ' s a
small game i attempted to make for ludum dare 36 during the bank holiday weekend
, but life and family fun was way more important .', 'a quick whack-a-mole type
game .', '.entry-content author rychan posted on 1st september 2016 categories
game jams , gbdk gameboy development .entry-footer # post- # # posts navigation
page 1 page 2 ... page 7 next page .site-main .content-area search me do search
for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes
game - coffee run !', '8th february 2023 secret hidden mini game complete !',
'15th january 2023 more homebrew nes games 8th january 2023 old creative
software still rocks 15 years later in 2022 2nd november 2022 retro game
projects update me do 1st november 2022 archives archives select month february
```


2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021
 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march
 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017
 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2)
 july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3)
 february 2016 (15) january 2016 (3) december 2015 (6) categories 2018
 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to
 gbdk nes dev tips nes game development one game a month ponderings uncategorised
 refresh games side ads .sidebar .widget-area .site-content about finished games
 game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy
 policy / .site-info .site-footer .site-inner .site lightbox plus colorbox
 v2.7.2/1.5.9 - 2013.01.24 - message : 0', "[if lt ie 10] > < link
 rel='stylesheet ' id='twentysixteen-ie-css ' href='https :
 //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?'", "[endif] [
 if lt ie 9] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https
 : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?'", "[endif
] [if lt ie 8] > < link rel='stylesheet ' id='twentysixteen-ie7-css '
 href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css
 ?'", "[endif] [if lt ie 9] > < script type='text/javascript ' src='https :
 //refreshgames.co.uk/wp-content/themes/twentysixteen/js/html5.js ?'", '[endif]
 analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to
 content refresh games gameboy game development in gbdk & other game dev things
 .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation .site-header-menu .site-header-main .header-image .site-header
 a sudden jolt , of gameboy goodness !', '.entry-header so , i noticed that
 gamejolt now allow you to play gameboy roms directly from your browser , that '
 s pretty cool i thought , best upload some games i ' ve made to see how it runs
 then .', 'and , lo and behold , 5 games uploaded and they all seem to be
 emulated rather well , they even included cgb (gameboy colour) support on
 there also .', 'it seems like a nice place to host the files , they ' re not
 going to take up much space either way , but gamejolt serves as a handy place to
 put them all !', 'view , play & download these lil gameboy beasts here !',
 '.entry-content author rychan posted on 9th april 2016 9th april 2016 categories
 finished games , gbdk gameboy development .entry-footer # post- # # post
 navigation previous previous post : why the long delay ?', 'next next post :
 gameboy gbdk tutorial rom code !', '.site-main .content-area search me do search
 for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes
 game - coffee run !', '8th february 2023 secret hidden mini game complete !',
 '15th january 2023 more homebrew nes games 8th january 2023 old creative
 software still rocks 15 years later in 2022 2nd november 2022 retro game
 projects update me do 1st november 2022 archives archives select month february
 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021
 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march
 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017
 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2)
 july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3)
 february 2016 (15) january 2016 (3) december 2015 (6) categories 2018
 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to

gbdk nes dev tips nes game development one game a month ponderings uncategorised
 refresh games side ads .sidebar .widget-area .site-content about finished games
 game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy
 policy / .site-info .site-footer .site-inner .site lightbox plus colorbox
 v2.7.2/1.5.9 - 2013.01.24 - message : 0', "[if lt ie 10] > < link
 rel='stylesheet ' id='twentysixteen-ie-css ' href='https :
 //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?"', "[endif] [
 if lt ie 9] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https
 : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?"', "[endif
] [if lt ie 8] > < link rel='stylesheet ' id='twentysixteen-ie7-css '
 href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css
 ?"', "[endif] [if lt ie 9] > < script type='text/javascript ' src='https :
 //refreshgames.co.uk/wp-content/themes/twentysixteen/js/html5.js ?"', '[endif]
 analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to
 content refresh games gameboy game development in gbdk & other game dev things
 .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation .site-header-menu .site-header-main .header-image .site-header
 ion - a gameboy action puzzle game in gbdk in development .entry-header despite
 not having many hour last weekend for ludum dare , i started to put together the
 pieces for another lil gameboy game called ion .', 'it ' s a fairly simple game
 in essence and uses ice physics for motion control with state-based a.i .',
 'here ' s a quick 20 second video with some of the game in action .', 'there ' s
 still a way to go , but i ' m planning on adding obstacles , slow and charge
 wells to help mix the gameplay up .', '.entry-content author rychan posted on
 6th may 2016 categories uncategorised .entry-footer # post- # # post navigation
 previous previous post : gameboy gbdk tutorial rom code !', 'next next post :
 formula racing gameboy homebrew developments .', '.site-main .content-area
 search me do search for : search twitterly woo tweets by @ refreshgamesuk recent
 posts februrarys nes game - coffee run !', '8th february 2023 secret hidden mini
 game complete !', '15th january 2023 more homebrew nes games 8th january 2023
 old creative software still rocks 15 years later in 2022 2nd november 2022 retro
 game projects update me do 1st november 2022 archives archives select month
 february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2)
 september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january
 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018
 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2)
 september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3)
 march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6)
 categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy
 development how to gbdk nes dev tips nes game development one game a month
 ponderings uncategorised refresh games side ads .sidebar .widget-area .site-
 content about finished games game jams gbdk gameboy dev how to gbdk .main-
 navigation refresh games privacy policy / .site-info .site-footer .site-inner
 .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "[if lt
 ie 10] > < link rel='stylesheet ' id='twentysixteen-ie-css ' href='https :
 //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?"', "[endif] [
 if lt ie 9] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https
 : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?"', "[endif

```

] [ if lt ie 8 ] > < link rel='stylesheet ' id='twentysixteen-ie7-css '
href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css
?', "[ endif ] [ if lt ie 9 ] > < script type='text/javascript ' src='https :
//refreshgames.co.uk/wp-content/themes/twentysixteen/js/html5.js ?', 'skip to
content refresh games gameboy game development in gbdk & other game dev things
.site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation .site-header-menu .site-header-main .header-image .site-header
category : gbdk gameboy development .page-header picking nes & game boy
development back up again .entry-header life has certainly gotten in the way of
a few things of late , the usual combination of commitments has shifted my focus
around a lot .', 'i ' ve not really felt settled at home to get many decent game
development sessions in for quite a while .', 'bee happy ( nes game development
) this game is at that weird stage in development where it ' s fairly playable
but needs some extra elements inside it for a good difficulty progression .',
'versus mode music sound effects transitions between screens / levels an ending
sequence flap happy and fancy free ( game boy & game boy colour dev ) flap happy
is further back in terms of development , i ' ve attempted to convert what i
made to utilize bank switching , but ran into issues .', 'the game is in a
similar position to bee happy and needs the following features to be considered
complete .', 'music sound effects level transitions ( easy fade ) ending
sequence cpu usage is still fine for the game boy which is good news for future
development bits .', 'keep your eyes peeled for updates later on this week for
more details on my retro game development adventures !', '.entry-content author
rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy
development , nes game development .entry-footer # post- # # how high is your
highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as
arrays as scores up to 255 would be rather boring... .', 'crayon syntax
highlighter v_2.7.2_beta c score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; 1 2 3 4 score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; [ format time : 0.0006 seconds ] in a game i ' m making
currently , scores can extend beyond the limits of a 16 bit integer also , so we
' re using slighlty longer arrays to store things in , no biggie .', 'i hope
this helps someone out there who has a similar issue regarding setting up high-
score boards on the gameboy as this seems to be a safe an fairly quick method
.', 'i may well be missing another way that ' s quicker all-around but this
seems to work for me anyway expect a new game posted around the end of this
month .entry-content author rychan posted on 19th january 2017 categories gbdk
gameboy development .entry-footer # post- # # so , august and me coding in gbdk
didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy
development , far from it .', 'there ' s plenty of games i need to finish off
before i move onto tackling 6502 asm again in october / november .', 'it ' s a
small game i attempted to make for ludum dare 36 during the bank holiday weekend
, but life and family fun was way more important .', 'a quick whack-a-mole type
game .', '.entry-content author rychan posted on 1st september 2016 categories
game jams , gbdk gameboy development .entry-footer # post- # # posts navigation
page 1 page 2 ... page 7 next page .site-main .content-area search me do search
for : search twittery woo tweets by @ refreshgamesuk recent posts februrarys nes
game - coffee run !', '8th february 2023 secret hidden mini game complete !',

```

'15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "[if lt ie 10] > < link rel='stylesheet ' id='twentysixteen-ie-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie.css ?'", "[endif] [if lt ie 9] > < link rel='stylesheet ' id='twentysixteen-ie8-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie8.css ?'", "[endif] [if lt ie 8] > < link rel='stylesheet ' id='twentysixteen-ie7-css ' href='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/css/ie7.css ?'", "[endif] [if lt ie 9] > < script type='text/javascript ' src='https : //refreshgames.co.uk/wp-content/themes/twentysixteen/js/html5.js ?'", 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', 'i ' ve not really felt settled at home to get many decent game development sessions in for quite a while .', 'bee happy (nes game development) this game is at that weird stage in development where it ' s fairly playable but needs some extra elements inside it for a good difficulty progression .', 'versus mode music sound effects transitions between screens / levels an ending sequence flap happy and fancy free (game boy & game boy colour dev) flap happy is further back in terms of development , i ' ve attempted to convert what i made to utilize bank switching , but ran into issues .', 'the game is in a similar position to bee happy and needs the following features to be considered complete .', 'music sound effects level transitions (easy fade) ending sequence cpu usage is still fine for the game boy which is good news for future development bits .', 'keep your eyes peeled for updates later on this week for more details on my retro game development adventures !', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'crayon syntax highlighter v_2.7.2_beta c score [0] = 9 ; score [1] = 0 ; score [2] = 0 ; score [3] = 1 ; 1 2 3 4 score [0] = 9 ; score [1] = 0 ; score [2] = 0 ;

score [3] = 1 ; [format time : 0.0007 seconds] in a game i ' m making currently , scores can extend beyond the limits of a 16 bit integer also , so we ' re using slightly longer arrays to store things in , no biggie .', 'i hope this helps someone out there who has a similar issue regarding setting up high-score boards on the gameboy as this seems to be a safe and fairly quick method .', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', 'there ' s plenty of games i need to finish off before i move onto tackling 6502 asm again in october / november .', 'it ' s a small game i attempted to make for ludum dare 36 during the bank holiday weekend , but life and family fun was way more important .', 'a quick whack-a-mole type game .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '8th february 2023 secret hidden mini game complete !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "sign in hubbringinfobar.html the gameboy development ring home > computers & internet > software > development > game programming hubnavsection.html about this ring dedicated to the development and programming of games , applications , and hardware using nintendo 's gameboy , and gameboy advance handheld gaming systems and/or gameboy emulators .", "ringmaster · gbdevring · contact ringmaster ring stats · founded on 12/31/1996 · 62 sites hub1_text.html sites in this ring rs.html in this ring all webring navbar.html showing : 1-20 of 62 next 20 > sitelist.html hell child games by killerslut its my little company , which i 'm using for my gameboy games i 'm hoping to produce .", 'game boy demon by thruster86 everything a game boy gamer needs is right here , end of discussion !', 'gameboy cartridge guts by gameboycartsuk photos (decent quality) of the insides of all my gameboy carts .', "amatuer developer software or a.d. software by amatuerdeveloper a.d. software is now making a game called `` the dark world '' for gbc and are looking for new people to join are team .", 's i x n i n e f i v e f i v e

gameboy music by dj6955 6955-making music with a gameboy as the sole instrument .', 'gameboy rawk !', "dreamer 's dreamscape by gbdevring webring maintainer & general gameboy development site gameboy tech central gameboy hardware & software info .", 'wzonka-lad , gameboy emulator for amiga machines is also under these pages .', 'the free vengeance group hosting the firts ever gameboy camera film festival ...\nsubmit , view and vote .', 't : h : e : o : r : ys website on my site you will find things under the topics of : art , music , programming and gameboy colour programming .', "stephen 's gameboy page by sgstair this site aims to bring the resources of assembly game boy development into one , findable area .", "old bits some example gameboy code : - rand () implementations in assembly , now used in gbdk- conway 's life- sub-pixel accuracy scrolling on the cgb herg 's gameboy development contains the games i have written for the gameboy including gbdk source code .", 'gamejack help by reubenhelms my experiences with using the ufo gb gamejack gameboy online by ldstyler this is an ezboard .', 'it is dedicated to gameboy development and has the rarest programs , software and information .', 'gameboy developement tutorials my gameboy developement site will soon contain many thorough tutorials on making games on the gameboy color gamempire its a site which hasnt been finished yet about gameboy .', 'gbdev gbdev.co.uk , the new website for beginner gameboy developers - by a beginner freestylez we are game boy developers that hav been working within the industry for many years.our site is not yet fully functional , but will feature information concerning the cgb , and the products we are completing for it .', 'navbar.html showing : 1-20 of 62 next 20 > formbottom.html address book · alerts · auctions · bill pay · bookmarks · briefcase · broadcast · calendar · chat · classifieds · clubs · companion · domains · experts · games · greetings · home pages · invites · mail · maps · member directory · messenger · my yahoo !]], {'tag': 'score', 'patterns': ['score'], 'responses': ['.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', '.entry-header right , i recently came into a little problem regarding storing highscores and comparing between them .', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'crayon syntax highlighter v_2.7.2_beta c score [0] = 9 ; score [1] = 0 ; score [2] = 0 ; score [3] = 1 ; 1 2 3 4 score [0] = 9 ; score [1] = 0 ; score [2] = 0 ; score [3] = 1 ; [format time : 0.0006 seconds] in a game i ' m making currently , scores can extend beyond the limits of a 16 bit integer also , so we ' re using slighlty longer arrays to store things in , no biggie .', 'i ' ve found a handy solution to my problem , along the lines of the following : crayon syntax highlighter v_2.7.2_beta c //set up our test scoreshiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2] = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ; score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ; score [5] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than highscorei = 0 ; while (higher == 10) { if (hiscore1 [i] > score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher = 20 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) { break ; } i++ ; } 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 //set up our test scores

```

hiscore1 [ 0 ] = 0 ; hiscore1 [ 1 ] = 0 ; hiscore1 [ 2 ] = 9 ; hiscore1 [ 3 ] =
6 ; hiscore1 [ 4 ] = 9 ; hiscore1 [ 5 ] = 9 ; score [ 0 ] = 0 ; score [ 1 ] = 0
; score [ 2 ] = 6 ; score [ 3 ] = 9 ; score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher
= 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than
highscore i = 0 ; while ( higher == 10 ) { if ( hiscore1 [ i ] > score [ i ] ) {
higher = 0 ; } if ( hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if (
hiscore1 [ i ] == score [ i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i ++
; } [ format time : 0.0006 seconds ] what the code above does is compare the
arrays on an integer by integer basis and set another variable with the result ,
reliably !', 'i hope this helps someone out there who has a similar issue
regarding setting up high-score boards on the gameboy as this seems to be a safe
an fairly quick method .', 'where you take the role of a charged particle that
needs to attain certain levels of charge to proceed to further levels and boost
the players score .', '.entry-content author rychan posted on 14th june 2022
14th june 2022 categories gbdk gameboy development , nes game development
.entry-footer # post- # # how high is your highscore ?', '.entry-header right ,
i recently came into a little problem regarding storing highscores and comparing
between them .', 'but with the gameboy being 8 bit , i ' m storing my scores as
arrays as scores up to 255 would be rather boring... .', 'crayon syntax
highlighter v_2.7.2_beta c score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; 1 2 3 4 score [ 0 ] = 9 ; score [ 1 ] = 0 ; score [ 2 ] = 0 ;
score [ 3 ] = 1 ; [ format time : 0.0006 seconds ] in a game i ' m making
currently , scores can extend beyond the limits of a 16 bit integer also , so we
' re using slighlty longer arrays to store things in , no biggie .', 'i ' ve
found a handy solution to my problem , along the lines of the following : crayon
syntax highlighter v_2.7.2_beta c //set up our test scoreshiscore1 [ 0 ] = 0 ;
hiscore1 [ 1 ] = 0 ; hiscore1 [ 2 ] = 9 ; hiscore1 [ 3 ] = 6 ; hiscore1 [ 4 ] =
9 ; hiscore1 [ 5 ] = 9 ; score [ 0 ] = 0 ; score [ 1 ] = 0 ; score [ 2 ] = 6 ;
score [ 3 ] = 9 ; score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher = 10 ; // 10 = same
score , 0 = lower than highscore , 20 = higher than highscorei = 0 ; while (
higher == 10 ) { if ( hiscore1 [ i ] > score [ i ] ) { higher = 0 ; } if (
hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if ( hiscore1 [ i ] == score [
i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i++ ; } 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 //set up our test scores
hiscore1 [ 0 ] = 0 ; hiscore1 [ 1 ] = 0 ; hiscore1 [ 2 ] = 9 ; hiscore1 [ 3 ] =
6 ; hiscore1 [ 4 ] = 9 ; hiscore1 [ 5 ] = 9 ; score [ 0 ] = 0 ; score [ 1 ] = 0
; score [ 2 ] = 6 ; score [ 3 ] = 9 ; score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher
= 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than
highscore i = 0 ; while ( higher == 10 ) { if ( hiscore1 [ i ] > score [ i ] ) {
higher = 0 ; } if ( hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if (
hiscore1 [ i ] == score [ i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i ++
; } [ format time : 0.0006 seconds ] what the code above does is compare the
arrays on an integer by integer basis and set another variable with the result ,
reliably !', 'i hope this helps someone out there who has a similar issue
regarding setting up high-score boards on the gameboy as this seems to be a safe
an fairly quick method .', '.entry-content author rychan posted on 14th june
2022 14th june 2022 categories gbdk gameboy development , nes game development
.entry-footer # post- # # how high is your highscore ?', '.entry-header right ,

```

i recently came into a little problem regarding storing highscores and comparing between them .', 'but with the gameboy being 8 bit , i ' m storing my scores as arrays as scores up to 255 would be rather boring... .', 'crayon syntax highlighter v_2.7.2_beta c score [0] = 9 ; score [1] = 0 ; score [2] = 0 ; score [3] = 1 ; 1 2 3 4 score [0] = 9 ; score [1] = 0 ; score [2] = 0 ; score [3] = 1 ; [format time : 0.0007 seconds] in a game i ' m making currently , scores can extend beyond the limits of a 16 bit integer also , so we ' re using slighlty longer arrays to store things in , no biggie .', 'i ' ve found a handy solution to my problem , along the lines of the following : crayon syntax highlighter v_2.7.2_beta c //set up our test scores
 hiscore1 [0] = 0 ;
 hiscore1 [1] = 0 ; hiscore1 [2] = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ; score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ; score [5] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than highscore
 i = 0 ; while (higher == 10) { if (hiscore1 [i] > score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher = 20 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) { break ; } i++ ; } 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 //set up our test scores
 hiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2] = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ; score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ; score [5] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than highscore
 i = 0 ; while (higher == 10) { if (hiscore1 [i] > score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher = 20 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) { break ; } i ++ ; } [format time : 0.0006 seconds] what the code above does is compare the arrays on an integer by integer basis and set another variable with the result , reliably !', 'i hope this helps someone out there who has a similar issue regarding setting up high-score boards on the gameboy as this seems to be a safe an fairly quick method .']}, {'tag': 'gbdk', 'patterns': ['gbdk'], 'responses': ['# compilers gbdk - maintained and modernized gbdk (game boy development kit) powered by an updated version of the sdcc toolchain .', 'gbdk-go - a compiler translates go programs to c code .', 'the output c code is built into gb rom by gbdk .', 'complete list of open source emulators # tools # engines zgb - a little engine for creating games for the original game boy (expands gbdk , more info here) .', 'retr0 gb - an engine for creating games (expands gbdk) .', '# development tools gbextended - c library extending gbdk .', 'gbdk-lib-extension - a small set of sources and tools for the game boy development kit by michael hope .', 'works with gbdk-2020 and rgbds .', 'gbtiles - converts .gbr files created with harry mulder 's tile designer (gbtd) and .gbm files created with harry mulder 's map builder (gbmb) to different formats for use with the game boy and gbdk .', 'mmlgb - a mml parser and gbdk sound driver for the nintendo game boy .', 'cbit-fx - a gbdk-2020 sound effect driver compatible with fx-hammer sound effects .', 'just another simple tutorial gbdk tutorial - fairly minimal game demo for getting started with gbdk .', 'gbdk sprite - presents a workflow for getting multiple sprites to display and animate .', 'gbdk color - extends your knowledge of basic spriting on the game boy by adding colors to sprites , backgrounds and the window layer .', 'gbdk joypad -


```

details the use of the joypad with gbdk .', 'gbdk configuring and programming
tutorial - configuring gbdk , using tiles , colliding sprites , gbtd , gbmb ,
memory management and rom banking .', 'simplified gbdk examples gbdk
programming video tutorials - a series of video tutorials introducing beginners
to programming with gbdk .', 'gbsnake gb-mines oranges red hot princess
carnage loderunner hives bubble factory - * vanilla- sdcc ( no gbdk ) .', '[
endif ] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip
to content refresh games gameboy game development in gbdk & other game dev
things .site-branding menu about finished games game jams gbdk gameboy dev how
to gbdk .main-navigation .site-header-menu .site-header-main .header-image
.site-header gameboy gbdk tutorial rom code !', 'gbdkdemo .zip file , 6kb crayon
syntax highlighter v_2.7.2_beta //////////////////////////////////////
//////////////////////////////////// demogame.c// -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- // fairly minimal game demo for getting started with gbdk// by refresh
games// -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // this is intended to be a base
point for anyone to customise and add to so// they can make their own gameboy
games , hence the comments !', '// -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // web :
refreshgames.co.uk// twitter : @ refreshgamesuk////////////////////////////////
//////////////////////////////// # include < rand.h > // include
random functions # include < gb/gb.h > ; // include gbdk function library #
include < gb/hardware.h > ; // include handy hardware references// function
declarationsvoid initgame ( ) ; // initialise our gamevoid updateplayer ( ) ; //
update our playeruint8 collisioncheck ( uint8 , uint8 , uint8 , uint8 , uint8 ,
uint8 , uint8 , uint8 ) ; // simple rect to rect check// variable declarations -
stored in ramuint8 i , j ; // generic looping variableuint8 playerx , playery ;
// player co-ordinatesuint8 ex [ 10 ] , ey [ 10 ] ; // enemy co-ordinatesuint8
lastkeys ; // holds keys for the previous frameuint8 randombkggtiles [ 20 ] ; //
contains random data for our bkg// sprite data - 14 sprites - stored in romconst
unsigned char sprites [ ] = { 0x3c,0x3c,0x42,0x42,0xa9,0xa9,0xa9,0xa9,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x3c,0x3c,0x42,0x42,0x81,0x81,0xed,0xed,0x81,0x81,
0x91,0x91,0x42,0x42,0x3c,0x3c,0x99,0x81,0x42,0x5a,0x24,0x3c,0x99,0x7e,0x99,0x7e,
0x24,0x3c,0x42,0x5a,0x99,0x81,0x38,0x04,0x7c,0x02,0x5c,0x22,0x5c,0x22,0x5c,0x22,
0x5c,0x22,0x7c,0x02,0x38,0x04,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x1c,0x1c,0x3c,0x3c,0x6c,0x6c,0x0c,0x0c,
0x0c,0x0c,0x7f,0x7f,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x03,0x03,0x1e,0x1e,
0x70,0x70,0x7f,0x7f,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x3e,0x3e,0x07,0x07,
0x43,0x43,0x3e,0x3e,0x00,0x00,0x00,0x00,0x60,0x60,0x6c,0x6c,0x6c,0x6c,0x7f,0x7f,
0x0c,0x0c,0x0c,0x0c,0x00,0x00,0x00,0x00,0x7f,0x7f,0x60,0x60,0x7e,0x7e,0x03,0x03,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3f,0x3f,0x60,0x60,0x7e,0x7e,0x63,0x63,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x7f,0x7f,0x03,0x03,0x06,0x06,0x0c,0x0c,
0x18,0x18,0x30,0x30,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3e,0x3e,
0x63,0x63,0x3e,0x3e,0x00,0x00,0x00,0x00,0x3e,0x3e,0x63,0x63,0x63,0x63,0x3f,0x3f,
0x03,0x03,0x7e,0x7e,0x00,0x00 } ; //background data - 4 sprites - stored in
romconst unsigned char mybkgdata [ ] = { 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x00,0x08,0x00,0x36,0x00

```

```

,0x08,0x00,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x00,0x50,0x00
,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x04,0x00,0x0e,0x00,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x50,0x00,0x20,0x00,0x50,0x00,0x00,0x00 } ; ///////////////////////////////////////////////////
///////////////////////////////////////////////// main functionvoid main ( ) {
initgame ( ) ; // ran once at startup// game loop startwhile ( 1 ) {
updateplayer ( ) ; // update playerhide_win ; // show window layersshow_sprites ;
// hide sprites layersshow_bkg ; // show background layerlastkeys = joypad ( ) ;
// update our keypresseswait_vbl_done ( ) ; // wait for vblank to finish -
ensures 60 frames per second maximum } // game loop end } ///////////////////////////////////////////////////
///////////////////////////////////////////////// init gamevoid
initgame ( ) { display_on ; // turns on the gameboy lcdnr52_reg = 0x8f ; // turn
sound onnr51_reg = 0x11 ; // enable sound channelsnr50_reg = 0x1f ; // volume
max = 0x77 , min = 0x00initrand ( div_reg ) ; // seed our
randomizerset_sprite_data ( 0 , 14 , sprites ) ; // store our sprite data at the
start of sprite vramset_bkg_data ( 0 , 4 , mybkgdata ) ; // store our bkg data
at the start of bkg vram - note , the window layer shared the bkg vram by
defaultplayerx = 64 ; // players initial x positionplayery = 64 ; // players
initial y positionset_sprite_tile ( 0,0 ) ; //players sprite tile - 0for ( i=0 ;
i != 10 ; i++ ) { // loop through our 10 enemiesif ( collisioncheck (
playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1 ) { j = 1 ; // if
we collided , set our flag variable , j to 1 } } // adjust our player sprites
frame if they hit somethingswitch ( j ) { // no hitcase 0 : set_sprite_tile (
0,0 ) ; break ; // hit enemydefault : set_sprite_tile ( 0,1 ) ; break ; } } //
/////////////////////////////////////////////////
collision checker - simple rectangle collision checking// returns 1 if
overlappinguint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 <= x2 ) & amp
; & amp ; ( ( x1+w1 ) <= x2 ) & amp ; & amp ; ( y1 <= y2 ) & amp
; & amp ; ( ( y1+h1 ) <= y2 ) ) { return 1 ; } else { return 0 ; } } 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149
150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169
170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198 199
/////////////////////////////////////////////////
// demogame.c // -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // fairly minimal game
demo for getting started with gbdk // by refresh games // -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- // this is intended to be a base point for anyone to customise
and add to so // they can make their own gameboy games , hence the comments !',
'// -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- // web : refreshgames.co.uk // twitter

```

```

: @ refreshgamesuk
/////////////////////////////////////////////////////////////////
# include < rand.h > // include random functions # include < gb/gb.h > ; //
include gbdk function library # include < gb/hardware.h > ; // include handy
hardware references // function declarations void initgame ( ) ; // initialise
our game void updateplayer ( ) ; // update our player uint8 collisioncheck (
uint8 , uint8 , uint8 , uint8 , uint8 , uint8 , uint8 , uint8 ) ; // simple rect
to rect check // variable declarations - stored in ram uint8 i , j ; // generic
looping variable uint8 playerx , playery ; // player co-ordinates uint8 ex [ 10
] , ey [ 10 ] ; // enemy co-ordinates uint8 lastkeys ; // holds keys for the
previous frame uint8 randombkgtiles [ 20 ] ; // contains random data for our bkg
// sprite data - 14 sprites - stored in rom const unsigned char sprites [ ] = {
0x3c , 0x3c , 0x42 , 0x42 , 0xa9 , 0xa9 , 0xa9 , 0xa9 , 0x81 , 0x81 , 0x91 ,
0x91 , 0x42 , 0x42 , 0x3c , 0x3c , 0x3c , 0x3c , 0x42 , 0x42 , 0x81 , 0x81 ,
0xed , 0xed , 0x81 , 0x81 , 0x91 , 0x91 , 0x42 , 0x42 , 0x3c , 0x3c , 0x99 ,
0x81 , 0x42 , 0x5a , 0x24 , 0x3c , 0x99 , 0x7e , 0x99 , 0x7e , 0x24 , 0x3c ,
0x42 , 0x5a , 0x99 , 0x81 , 0x38 , 0x04 , 0x7c , 0x02 , 0x5c , 0x22 , 0x5c ,
0x22 , 0x5c , 0x22 , 0x5c , 0x22 , 0x7c , 0x02 , 0x38 , 0x04 , 0x00 , 0x00 ,
0x3e , 0x3e , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x63 , 0x3e ,
0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x1c , 0x1c , 0x3c , 0x3c , 0x6c , 0x6c ,
0x0c , 0x0c , 0x0c , 0x0c , 0x7f , 0x7f , 0x00 , 0x00 , 0x00 , 0x00 , 0x3e ,
0x3e , 0x63 , 0x63 , 0x03 , 0x03 , 0x1e , 0x1e , 0x70 , 0x70 , 0x7f , 0x7f ,
0x00 , 0x00 , 0x00 , 0x00 , 0x7f , 0x7f , 0x03 , 0x03 , 0x3e , 0x3e , 0x07 ,
0x07 , 0x43 , 0x43 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x60 , 0x60 ,
0x6c , 0x6c , 0x6c , 0x6c , 0x7f , 0x7f , 0x0c , 0x0c , 0x0c , 0x0c , 0x00 ,
0x00 , 0x00 , 0x00 , 0x7f , 0x7f , 0x60 , 0x60 , 0x7e , 0x7e , 0x03 , 0x03 ,
0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 , 0x00 , 0x3f , 0x3f , 0x60 ,
0x60 , 0x7e , 0x7e , 0x63 , 0x63 , 0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 ,
0x00 , 0x00 , 0x7f , 0x7f , 0x03 , 0x03 , 0x06 , 0x06 , 0x0c , 0x0c , 0x18 ,
0x18 , 0x30 , 0x30 , 0x00 , 0x00 , 0x00 , 0x00 , 0x3e , 0x3e , 0x63 , 0x63 ,
0x63 , 0x63 , 0x3e , 0x3e , 0x63 , 0x63 , 0x3e , 0x3e , 0x00 , 0x00 , 0x00 ,
0x00 , 0x3e , 0x3e , 0x63 , 0x63 , 0x63 , 0x63 , 0x3f , 0x3f , 0x03 , 0x03 ,
0x7e , 0x7e , 0x00 , 0x00 } ; //background data - 4 sprites - stored in rom
const unsigned char mybkgdata [ ] = { 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x08 , 0x00 , 0x08 , 0x00 , 0x36 , 0x00 , 0x08 , 0x00 , 0x08 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x20 , 0x00 , 0x50 ,
0x00 , 0x20 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x04 , 0x00 , 0x0e ,
0x00 , 0x04 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 , 0x00 ,
0x50 , 0x00 , 0x20 , 0x00 , 0x50 , 0x00 , 0x00 , 0x00 } ;
/////////////////////////////////////////////////////////////////
// main function void main ( ) { initgame ( ) ; // ran once at startup // game
loop start while ( 1 ) { updateplayer ( ) ; // update player hide_win ; // show
window layer show_sprites ; // hide sprites layer show_bkg ; // show background
layer lastkeys = joypad ( ) ; // update our keypresses wait_vbl_done ( ) ; //
wait for vblank to finish - ensures 60 frames per second maximum } // game loop
end }

```

```

////////////////////////////////////
// init game void initgame ( ) { display_on ; // turns on the gameboy lcd
nr52_reg = 0x8f ; // turn sound on nr51_reg = 0x11 ; // enable sound channels
nr50_reg = 0x1f ; // volume max = 0x77 , min = 0x00 initrand ( div_reg ) ; //
seed our randomizer set_sprite_data ( 0 , 14 , sprites ) ; // store our sprite
data at the start of sprite vram set_bkg_data ( 0 , 4 , mybkgdata ) ; // store
our bkg data at the start of bkg vram - note , the window layer shared the bkg
vram by default playerx = 64 ; // players initial x position playery = 64 ; //
players initial y position set_sprite_tile ( 0 , 0 ) ; //players sprite tile - 0
for ( i = 0 ; i != 10 ; i ++ ) { // loop through our 10 enemies if (
collisioncheck ( playerx , playery , 8 , 8 , ex [ i ] , ey [ i ] , 8 , 8 ) == 1
) { j = 1 ; // if we collided , set our flag variable , j to 1 } } // adjust our
player sprites frame if they hit something switch ( j ) { // no hit case 0 :
set_sprite_tile ( 0 , 0 ) ; break ; // hit enemy default : set_sprite_tile ( 0 ,
1 ) ; break ; } }
////////////////////////////////////
// collision checker - simple rectangle collision checking // returns 1 if
overlapping uint8 collisioncheck ( uint8 x1 , uint8 y1 , uint8 w1 , uint8 h1 ,
uint8 x2 , uint8 y2 , uint8 w2 , uint8 h2 ) { if ( ( x1 <= x2 ) &
( x1 + w1 >= x2 + w2 ) & ( y1 <= y2 ) & ( y1 + h1 >= y2 + h2 ) ) { return 1 ; } else { return 0 ; }
} [ format time : 0.0040 seconds ] .entry-content author rychan posted on 18th
april 2016 18th april 2016 categories uncategorised .entry-footer # post- # # 1
thought on " gameboy gbdk tutorial rom code !", 'next next post : ion - a
gameboy action puzzle game in gbdk in development .site-main .content-area
search me do search for : search twitter woos tweets by @ refreshgamesuk recent
posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes
games 8th january 2023 old creative software still rocks 15 years later in 2022
2nd november 2022 retro game projects update me do 1st november 2022 archives
archives select month february 2023 ( 1 ) january 2023 ( 2 ) november 2022 ( 2 )
june 2022 ( 2 ) september 2021 ( 1 ) may 2021 ( 3 ) april 2021 ( 11 ) march 2021
( 1 ) january 2021 ( 2 ) march 2020 ( 1 ) february 2020 ( 2 ) february 2018 ( 1
) january 2018 ( 4 ) april 2017 ( 2 ) january 2017 ( 1 ) november 2016 ( 1 )
october 2016 ( 2 ) september 2016 ( 2 ) july 2016 ( 1 ) june 2016 ( 2 ) may 2016
( 2 ) april 2016 ( 3 ) march 2016 ( 3 ) february 2016 ( 15 ) january 2016 ( 3 )
december 2015 ( 6 ) categories 2018 nes dev atari vcs 2600 finished games game
jams gbdk gameboy development how to gbdk nes dev tips nes game development one
game a month ponderings uncategorised refresh games side ads .sidebar .widget-
area .site-content about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation refresh games privacy policy / .site-info .site-footer .site-
inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',
'4/16/01 ems_multi-rom_source.zip this is the source code ( gbdk 2.1.5 ) for my
ems multi-rom menu program .', '12/12/00 bung_multi-rom_source.zip this is the
source code ( gbdk 2.1.5 ) for my bung multi-rom menu program .', '10/2/00
pvoice0999.zip this is the new candidate for the final pvoice library ( for gbdk
) .', '9/29/00 splash04.zip gbdk source code to display a 160x144graphic on the
gb and gbc .', '9/13/00 pvoice099.zip this is version 0.99 of my pvoice library
( for gbdk ) .', '5/11/00 pvoice097.zip gbdk library and source code for

```

recording , playback , and rom flashing on the pocketvoice .', "2/3/00
 pvoice095.zip gbdk library for record/playback on the pocketvoice 1/3/00
 arkaid32.zip this is a 32k version of noah 's arkaid .", 'skip to content
 refresh games gameboy game development in gbdk & other game dev things .site-
 branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-
 navigation .site-header-menu .site-header-main .header-image .site-header
 category : gbdk gameboy development .page-header picking nes & game boy
 development back up again .entry-header life has certainly gotten in the way of
 a few things of late , the usual combination of commitments has shifted my focus
 around a lot .', '.entry-content author rychan posted on 14th june 2022 14th
 june 2022 categories gbdk gameboy development , nes game development .entry-
 footer # post- # # how high is your highscore ?', 'i may well be missing another
 way that ' s quicker all-around but this seems to work for me anyway expect a
 new game posted around the end of this month .entry-content author rychan
 posted on 19th january 2017 categories gbdk gameboy development .entry-footer #
 post- # # so , august and me coding in gbdk didn ' t really happen .', '.entry-
 content author rychan posted on 1st september 2016 categories game jams , gbdk
 gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ...
 page 7 next page .site-main .content-area search me do search for : search
 twitter woos tweets by @ refreshgamesuk recent posts februarys nes game -
 coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old
 creative software still rocks 15 years later in 2022 2nd november 2022 retro
 game projects update me do 1st november 2022 archives archives select month
 february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2)
 september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january
 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018
 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2)
 september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3)
 march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6)
 categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy
 development how to gbdk nes dev tips nes game development one game a month
 ponderings uncategorised refresh games side ads .sidebar .widget-area .site-
 content about finished games game jams gbdk gameboy dev how to gbdk .main-
 navigation refresh games privacy policy / .site-info .site-footer .site-inner
 .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif
] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to
 content refresh games gameboy game development in gbdk & other game dev things
 .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation .site-header-menu .site-header-main .header-image .site-header
 a sudden jolt , of gameboy goodness !', '.entry-content author rychan posted on
 9th april 2016 9th april 2016 categories finished games , gbdk gameboy
 development .entry-footer # post- # # post navigation previous previous post :
 why the long delay ?', 'next next post : gameboy gbdk tutorial rom code !',
 '15th january 2023 more homebrew nes games 8th january 2023 old creative
 software still rocks 15 years later in 2022 2nd november 2022 retro game
 projects update me do 1st november 2022 archives archives select month february
 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021
 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march

2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017
 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2)
 july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3)
 february 2016 (15) january 2016 (3) december 2015 (6) categories 2018
 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to
 gbdk nes dev tips nes game development one game a month ponderings uncategorised
 refresh games side ads .sidebar .widget-area .site-content about finished games
 game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy
 policy / .site-info .site-footer .site-inner .site lightbox plus colorbox
 v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif] analytics by wp statistics
 v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy
 game development in gbdk & other game dev things .site-branding menu about
 finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-
 header-menu .site-header-main .header-image .site-header ion - a gameboy action
 puzzle game in gbdk in development .entry-header despite not having many hour
 last weekend for ludum dare , i started to put together the pieces for another
 lil gameboy game called ion .', '.entry-content author rychan posted on 6th may
 2016 categories uncategorised .entry-footer # post- # # post navigation previous
 previous post : gameboy gbdk tutorial rom code !', '15th january 2023 more
 homebrew nes games 8th january 2023 old creative software still rocks 15 years
 later in 2022 2nd november 2022 retro game projects update me do 1st november
 2022 archives archives select month february 2023 (1) january 2023 (2)
 november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april
 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2)
 february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1)
 november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june
 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15)
 january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600
 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes
 game development one game a month ponderings uncategorised refresh games side
 ads .sidebar .widget-area .site-content about finished games game jams gbdk
 gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-
 info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 -
 2013.01.24 - message : 0', 'skip to content refresh games gameboy game
 development in gbdk & other game dev things .site-branding menu about finished
 games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu
 .site-header-main .header-image .site-header category : gbdk gameboy development
 .page-header picking nes & game boy development back up again .entry-header life
 has certainly gotten in the way of a few things of late , the usual combination
 of commitments has shifted my focus around a lot .', '.entry-content author
 rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy
 development , nes game development .entry-footer # post- # # how high is your
 highscore ?', 'i may well be missing another way that ' s quicker all-around but
 this seems to work for me anyway expect a new game posted around the end of
 this month .entry-content author rychan posted on 19th january 2017 categories
 gbdk gameboy development .entry-footer # post- # # so , august and me coding in
 gbdk didn ' t really happen .', '.entry-content author rychan posted on 1st
 september 2016 categories game jams , gbdk gameboy development .entry-footer #

post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-
 area search me do search for : search twitter woos tweets by @ refreshgamesuk
 recent posts februarys nes game - coffee run !', '15th january 2023 more
 homebrew nes games 8th january 2023 old creative software still rocks 15 years
 later in 2022 2nd november 2022 retro game projects update me do 1st november
 2022 archives archives select month february 2023 (1) january 2023 (2)
 november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april
 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2)
 february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1)
 november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june
 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15)
 january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600
 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes
 game development one game a month ponderings uncategorised refresh games side
 ads .sidebar .widget-area .site-content about finished games game jams gbdk
 gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-
 info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 -
 2013.01.24 - message : 0', 'skip to content refresh games gameboy game
 development in gbdk & other game dev things .site-branding menu about finished
 games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu
 .site-header-main .header-image .site-header category : gbdk gameboy development
 .page-header picking nes & game boy development back up again .entry-header life
 has certainly gotten in the way of a few things of late , the usual combination
 of commitments has shifted my focus around a lot .', '.entry-content author
 rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy
 development , nes game development .entry-footer # post- # # how high is your
 highscore ?', 'i may well be missing another way that ' s quicker all-around but
 this seems to work for me anyway expect a new game posted around the end of
 this month .entry-content author rychan posted on 19th january 2017 categories
 gbdk gameboy development .entry-footer # post- # # so , august and me coding in
 gbdk didn ' t really happen .', '.entry-content author rychan posted on 1st
 september 2016 categories game jams , gbdk gameboy development .entry-footer #
 post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-
 area search me do search for : search twitter woos tweets by @ refreshgamesuk
 recent posts februarys nes game - coffee run !', '15th january 2023 more
 homebrew nes games 8th january 2023 old creative software still rocks 15 years
 later in 2022 2nd november 2022 retro game projects update me do 1st november
 2022 archives archives select month february 2023 (1) january 2023 (2)
 november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april
 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2)
 february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1)
 november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june
 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15)
 january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600
 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes
 game development one game a month ponderings uncategorised refresh games side
 ads .sidebar .widget-area .site-content about finished games game jams gbdk
 gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-

```

info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 -
2013.01.24 - message : 0', "old bits some example gameboy code : - rand ( )
implementations in assembly , now used in gbdk- conway 's life- sub-pixel
accuracy scrolling on the cgb herg 's gameboy development contains the games i
have written for the gameboy including gbdk source code ."}], {'tag':
'development', 'patterns': ['development', 'develop a game', 'dev', 'dev tools',
'dev board'], 'responses': ["[ [ gbdev.io [ ] [ ] games [ ] [ ] pan docs [ ]
community community [ ] chat [ ] [ ] contribute [ ] [ ] gb competition 2021 [
] [ ] resources [ ] guides guides [ ] [ ] choosing development tools [ ] [ ] asm
style recommendations [ ] [ ] the timing of lyc stat handlers [ ] [ ] dead c
scroll [ ] [ ] the wonders of prehistorik man [ ] [ ] dma hijacking [ ] [ ] [ ]
[ [ [ ] games [ ] [ ] pan docs [ ] community community [ ] chat [ ] [ ]
contribute [ ] [ ] gb competition 2021 [ ] [ ] resources [ ] guides guides [ ]
] choosing development tools [ ] [ ] asm style recommendations [ ] [ ] the timing
of lyc stat handlers [ ] [ ] dead c scroll [ ] [ ] the wonders of prehistorik
man [ ] [ ] dma hijacking [ ] [ ] [ ] [ resources [ ] introduction [ ] [ [ ]
the game boy , a hardware autopsy [ ] [ ] the ultimate game boy talk [ ] [ ]
documentation [ ] [ ] game boy color [ ] [ ] hardware [ ] [ ] peripherals [ ]
[ ] cartridges [ ] [ ] emulator development [ ] [ ] testing [ ] [ ]
software development [ ] [ ] assemblers [ ] [ ] compilers [ ] [ ] emulators [
] [ ] tools [ ] [ ] programming [ ] [ ] asm [ ] [ ] c [ ] [ ] homebrews [
] [ ] asm [ ] [ ] c [ ] [ ] gb studio [ ] [ ] demos [ ] [ ] reverse
engineering [ ] [ ] game disassemblies [ ] [ ] game boy camera [ ] [ ]
retrieving images [ ] [ ] changing the camera 's behavior [ ] [ ] post
processing [ ] [ ] related projects [ ] [ ] directories [ ] [ ] websites [ ]
] [ ] [ ] [ ] [ ] awesome game boy development a curated list of awesome
game boy ( color ) development resources , tools , docs , related projects and
homebrews .", '# introduction # the game boy , a hardware autopsy # the
ultimate game boy talk # disambiguation # game boy advance game boy advance
development is covered by another project , the awesome-gbadev list .', 'this
is covered in the emulator development section of this list .', 'esp8266 gb dev
board - dev board for game boy accessories development , powered by esp8266 .',
'game boy cartridge pcb photos # custom cartridges emulating a gameboy
cartridge - emulating the functionality of a game boy cartridge with the
development board stm32f4 .', '# emulator development reverse engineering fine
details of game boy hardware - 43 minutes talk by gekkio given at disobey 2018
( errata ) .', 'dmg-01 - an educational gameboy emulator in rust and a
companion book explaining its development .', 'porting a go game boy emulator to
webassembly about swotgb - notes about the development of a game boy emulator
in javascript .', '# software development the choosing tools for game boy
development essay provides an overview of the available development tools for
game boy .', 'asmotor - assembler engine and development system targeting game
boy , among other cpus .', '# compilers gbdk - maintained and modernized gbdk (
game boy development kit ) powered by an updated version of the sdcc toolchain
.', 'gbforth - a forth-based game boy development kit .', '# development tools
gbextended - c library extending gbdk .', 'gbdk-lib-extension - a small set of
sources and tools for the game boy development kit by michael hope .', "harry
mulder 's gb development - some sources and home of game boy tile designer (

```


gbtd) and game boy map builder (gbmb) tools .", 'bmp2cgb - graphics converter for game boy color development providing real time palette adjustments .', 'brewtool - a collection of primitive editor/converter tools for making assets used with homebrew rom development .', '# programming guides , tutorials and tools to develop software for game boy using the development toolchains described in the software development chapter .', 'flappyboy : making a simple game boy game super game boy development - step by step tutorial to implement super game boy features (frame and palettes) .', 'got recently released with the full source , development tools and workflows .', 'in active development .', '[endif] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header gameboy gbdk tutorial rom code !', 'i hope this helps you to get started on your gameboy development journeys !', '" pingback : awesome game boy development - massive collection of resources - learn practice & share # comment- # # .comment-list comments are closed .', 'next next post : ion - a gameboy action puzzle game in gbdk in development .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "if you 're interested in gameboy advance development , check out my gba dev page .", "this is a mirror of the old `` harry mulder 's gameboy development ' ' site .", 'start bravenet.com service code end bravenet.com service code counter added 01-28-2000 the gameboy development ring [join now | ring hub | random | < < prev | next > >]', 'getting radare2 installation radare2 ' s development is pretty quick - the project evolves every day , therefore it ' s recommended to use the current git version over the stable one .', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', 'i ' ve not really felt settled at home to get many decent game

development sessions in for quite a while .', 'bee happy (nes game development) this game is at that weird stage in development where it ' s fairly playable but needs some extra elements inside it for a good difficulty progression .', 'versus mode music sound effects transitions between screens / levels an ending sequence flap happy and fancy free (game boy & game boy colour dev) flap happy is further back in terms of development , i ' ve attempted to convert what i made to utilize bank switching , but ran into issues .', 'music sound effects level transitions (easy fade) ending sequence cpu usage is still fine for the game boy which is good news for future development bits .', 'keep your eyes peeled for updates later on this week for more details on my retro game development adventures !', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header a sudden jolt , of gameboy goodness !', '.entry-content author rychan posted on 9th april 2016 9th april 2016 categories finished games , gbdk gameboy development .entry-footer # post- # # post navigation previous previous post : why the long delay ?', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2)

september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january
 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018
 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2)
 september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3)
 march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6)
 categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy
 development how to gbdk nes dev tips nes game development one game a month
 ponderings uncategorised refresh games side ads .sidebar .widget-area .site-
 content about finished games game jams gbdk gameboy dev how to gbdk .main-
 navigation refresh games privacy policy / .site-info .site-footer .site-inner
 .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[endif
] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip to
 content refresh games gameboy game development in gbdk & other game dev things
 .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation .site-header-menu .site-header-main .header-image .site-header
 ion - a gameboy action puzzle game in gbdk in development .entry-header despite
 not having many hour last weekend for ludum dare , i started to put together the
 pieces for another lil gameboy game called ion .', 'next next post : formula
 racing gameboy homebrew developments .', '15th january 2023 more homebrew nes
 games 8th january 2023 old creative software still rocks 15 years later in 2022
 2nd november 2022 retro game projects update me do 1st november 2022 archives
 archives select month february 2023 (1) january 2023 (2) november 2022 (2)
 june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021
 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1)
 january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1)
 october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016
 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3)
 december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game
 jams gbdk gameboy development how to gbdk nes dev tips nes game development one
 game a month ponderings uncategorised refresh games side ads .sidebar .widget-
 area .site-content about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation refresh games privacy policy / .site-info .site-footer .site-
 inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',
 'skip to content refresh games gameboy game development in gbdk & other game dev
 things .site-branding menu about finished games game jams gbdk gameboy dev how
 to gbdk .main-navigation .site-header-menu .site-header-main .header-image
 .site-header category : gbdk gameboy development .page-header picking nes & game
 boy development back up again .entry-header life has certainly gotten in the way
 of a few things of late , the usual combination of commitments has shifted my
 focus around a lot .', 'i ' ve not really felt settled at home to get many
 decent game development sessions in for quite a while .', 'bee happy (nes game
 development) this game is at that weird stage in development where it ' s
 fairly playable but needs some extra elements inside it for a good difficulty
 progression .', 'versus mode music sound effects transitions between screens /
 levels an ending sequence flap happy and fancy free (game boy & game boy colour
 dev) flap happy is further back in terms of development , i ' ve attempted to
 convert what i made to utilize bank switching , but ran into issues .', 'music
 sound effects level transitions (easy fade) ending sequence cpu usage is still

fine for the game boy which is good news for future development bits .', 'keep your eyes peeled for updates later on this week for more details on my retro game development adventures !', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', 'i ' ve not really felt settled at home to get many decent game development sessions in for quite a while .', 'bee happy (nes game development) this game is at that weird stage in development where it ' s fairly playable but needs some extra elements inside it for a good difficulty progression .', 'versus mode music sound effects transitions between screens / levels an ending sequence flap happy and fancy free (game boy & game boy colour dev) flap happy is further back in terms of development , i ' ve attempted to convert what i made to utilize bank switching , but ran into issues .', 'music sound effects level transitions (easy fade) ending sequence cpu usage is still fine for the game boy which is good news for future development bits .', 'keep your eyes peeled for updates later on this week for more details on my retro game development adventures !', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game

development .entry-footer # post- # # how high is your highscore ?', 'i may well be missing another way that ' s quicker all-around but this seems to work for me anyway expect a new game posted around the end of this month .entry-content author rychan posted on 19th january 2017 categories gbdk gameboy development .entry-footer # post- # # so , august and me coding in gbdk didn ' t really happen .', 'but , that doesn ' t mean i ' m stopping gameboy development , far from it .', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "sign in hubringinfobar.html the gameboy development ring home > computers & internet > software > development > game programming hubnavsection.html about this ring dedicated to the development and programming of games , applications , and hardware using nintendo 's gameboy , and gameboy advance handheld gaming systems and/or gameboy emulators .", "dreamer 's dreamscape by gbdevring webring maintainer & general gameboy development site gameboy tech central gameboy hardware & software info .", "stephen 's gameboy page by sgstair this site aims to bring the resources of assembly game boy development into one , findable area .", "old bits some example gameboy code : - rand () implementations in assembly , now used in gbdk- conway 's life- sub-pixel accuracy scrolling on the cgb here 's gameboy development contains the games i have written for the gameboy including gbdk source code .", 'it is dedicated to gameboy development and has the rarest programs , software and information .']}, {'tag': 'nes', 'patterns': ['nes', 'snes', 'console'], 'responses': ['infrared madness - infrared communication on the game boy color .', 'wiz - a high-level assembly language for writing homebrew on retro console platforms (game boy , nes , atari 2600 , and more) .', 'complete list of open source emulators # tools # engines zgb - a little engine for creating games for the original game boy (expands gbdk , more info here) .', 'evscript - a scripting language for the game boy , useful for enemy ai , dialogue , animations , and coroutines .', 'tilemap studio - a tilemap editor for game boy , color , advance , and snes projects .', 'game boy assembly language primer - simple template code with memory defines , copy routines and ibm font tilemap .', 'superhappyfunbubbletime minesweepgb libbet

and the magic floor waveform-gb - program visualizing the wave form used by the wave channel .', 'gbsnake gb-mines oranges red hot princess carnage loderunner hives bubble factory - * vanilla- sdcc (no gbdk) .', 'genesis - shmup for the game boy (itch.io) .', 'digitized speech in game boy games sniffing game boy serial traffic with an stm32f4 virtual game boy printer with an stm32f4 printing on the game boy printer using an stm32f4 programming game boy chinese cartridges with an stm32f4 pokemon pocket computer : - what is it and how to use it to make cheat codes .', 'summary of my experiment back in 1998 , the gameboy camera got the world record as " smallest digital camera " in the guinness book of records .', 'especially in the last image you see a brightness gradient over the image .', 'some stuff here might also be a bit difficult to follow if you ' ve never done any programming before - and , to be honest , if you haven ' t programmed before , then jumping right into assembler like this is probably not the best way to learn .', "the first bit of code we ' re going to look at is these two lines : include `` gbhw.inc '' include `` ibmpc1.inc '' these two lines import two include files .", 'the second include file , ibmpc1.inc , defines the ascii font set as a set of 8x8 pixel images , also known as " tiles " .', "interrupts now , the next few bits of code we ' re dealing with are these ten lines , here : section `` vblank '' , home [\$ 0040] ret section '' lcdc '' , home [\$ 0048] ret section '' timer_overflow '' , home [\$ 0050] ret section `` serial '' , home [\$ 0058] ret section `` p1thru4 '' , home [\$ 0060] ret now , these lines don ' t actually do anything in this program .", 'in this case , we won ' t actually be using interrupts at all , so none of these routines do anything - the only instruction in any of them is " ret " , which is used to exit the interrupt handler and go back to where the program was interrupted .', 'essentially , taken together , these 2 lines set up the palette so that any tiles drawn will use these 4 colours .', 'that sounds like a lot (or a few lines , at least) of work to do something we can sum up pretty easily in a few words , so this seems like the perfect time to introduce something new : subroutines .', '.comments-area post navigation previous previous post : a sudden jolt , of gameboy goodness !', 'next next post : ion - a gameboy action puzzle game in gbdk in development .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',

"honestly , i don't know what state this is in , since i have n't touched it for a while .", 'after a while and a deeper understanding of the function and each condition (radare2 isn ' t able to debug the program so it was even harder) i understood that the blocks are not so different from each other , except several lines .', 'although it ' s a simple lines of code , the disassembly looked awful .', 'some stuff here might also be a bit difficult to follow if you ' ve never done any programming before - and , to be honest , if you haven ' t programmed before , then jumping right into assembler like this is probably not the best way to learn .', "the first bit of code we ' re going to look at is these two lines : include `` gbhw.inc ' ' include `` ibmpc1.inc ' ' these two lines import two include files .", 'the second include file , ibmpc1.inc , defines the ascii font set as a set of 8x8 pixel images , also known as " tiles " .', "interrupts now , the next few bits of code we ' re dealing with are these ten lines , here : section `` vblank ' ' , home [\$ 0040] ret section ' ' lcdc ' ' , home [\$ 0048] ret section ' ' timer_overflow ' ' , home [\$ 0050] ret section `` serial ' ' , home [\$ 0058] ret section `` p1thru4 ' ' , home [\$ 0060] ret now , these lines don ' t actually do anything in this program .", 'in this case , we won ' t actually be using interrupts at all , so none of these routines do anything - the only instruction in any of them is " ret " , which is used to exit the interrupt handler and go back to where the program was interrupted .', 'essentially , taken together , these 2 lines set up the palette so that any tiles drawn will use these 4 colours .', 'that sounds like a lot (or a few lines , at least) of work to do something we can sum up pretty easily in a few words , so this seems like the perfect time to introduce something new : subroutines .', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', 'probably every wednesday .', 'bee happy (nes game development) this game is at that weird stage in development where it ' s fairly playable but needs some extra elements inside it for a good difficulty progression .', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'i ' ve found a handy solution to my problem , along the lines of the following : crayon syntax highlighter v_2.7.2_beta c //set up our test scores hiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2] = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ; score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ; score [5] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than highscore i = 0 ; while (higher == 10) { if (hiscore1 [i] > score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher = 20 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) { break ; } i++ ; } 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 //set up our test scores hiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2] = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ; score [1] = 0

```

; score [ 2 ] = 6 ; score [ 3 ] = 9 ; score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher
= 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than
highscore i = 0 ; while ( higher == 10 ) { if ( hiscore1 [ i ] > score [ i ] ) {
higher = 0 ; } if ( hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if (
hiscore1 [ i ] == score [ i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i ++
; } [ format time : 0.0006 seconds ] what the code above does is compare the
arrays on an integer by integer basis and set another variable with the result ,
reliably !', '.entry-content author rychan posted on 1st september 2016
categories game jams , gbdk gameboy development .entry-footer # post- # # posts
navigation page 1 page 2 ... page 7 next page .site-main .content-area search me
do search for : search twitterly woo tweets by @ refreshgamesuk recent posts
februrarys nes game - coffee run !', '15th january 2023 more homebrew nes games
8th january 2023 old creative software still rocks 15 years later in 2022 2nd
november 2022 retro game projects update me do 1st november 2022 archives
archives select month february 2023 ( 1 ) january 2023 ( 2 ) november 2022 ( 2 )
june 2022 ( 2 ) september 2021 ( 1 ) may 2021 ( 3 ) april 2021 ( 11 ) march 2021
( 1 ) january 2021 ( 2 ) march 2020 ( 1 ) february 2020 ( 2 ) february 2018 ( 1
) january 2018 ( 4 ) april 2017 ( 2 ) january 2017 ( 1 ) november 2016 ( 1 )
october 2016 ( 2 ) september 2016 ( 2 ) july 2016 ( 1 ) june 2016 ( 2 ) may 2016
( 2 ) april 2016 ( 3 ) march 2016 ( 3 ) february 2016 ( 15 ) january 2016 ( 3 )
december 2015 ( 6 ) categories 2018 nes dev atari vcs 2600 finished games game
jams gbdk gameboy development how to gbdk nes dev tips nes game development one
game a month ponderings uncategorised refresh games side ads .sidebar .widget-
area .site-content about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation refresh games privacy policy / .site-info .site-footer .site-
inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '[
endif ] analytics by wp statistics v13.2.16 - https : //wp-statistics.com/ skip
to content refresh games gameboy game development in gbdk & other game dev
things .site-branding menu about finished games game jams gbdk gameboy dev how
to gbdk .main-navigation .site-header-menu .site-header-main .header-image
.site-header a sudden jolt , of gameboy goodness !', '.site-main .content-area
search me do search for : search twitterly woo tweets by @ refreshgamesuk recent
posts februrarys nes game - coffee run !', '15th january 2023 more homebrew nes
games 8th january 2023 old creative software still rocks 15 years later in 2022
2nd november 2022 retro game projects update me do 1st november 2022 archives
archives select month february 2023 ( 1 ) january 2023 ( 2 ) november 2022 ( 2 )
june 2022 ( 2 ) september 2021 ( 1 ) may 2021 ( 3 ) april 2021 ( 11 ) march 2021
( 1 ) january 2021 ( 2 ) march 2020 ( 1 ) february 2020 ( 2 ) february 2018 ( 1
) january 2018 ( 4 ) april 2017 ( 2 ) january 2017 ( 1 ) november 2016 ( 1 )
october 2016 ( 2 ) september 2016 ( 2 ) july 2016 ( 1 ) june 2016 ( 2 ) may 2016
( 2 ) april 2016 ( 3 ) march 2016 ( 3 ) february 2016 ( 15 ) january 2016 ( 3 )
december 2015 ( 6 ) categories 2018 nes dev atari vcs 2600 finished games game
jams gbdk gameboy development how to gbdk nes dev tips nes game development one
game a month ponderings uncategorised refresh games side ads .sidebar .widget-
area .site-content about finished games game jams gbdk gameboy dev how to gbdk
.main-navigation refresh games privacy policy / .site-info .site-footer .site-
inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0',
'charges that vary between seeking and fleeing routines to keep the player on

```


their toes .', '.site-main .content-area search me do search for : search
 twitter woos tweets by @refreshgamesuk recent posts february's nes game -
 coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old
 creative software still rocks 15 years later in 2022 2nd november 2022 retro
 game projects update me do 1st november 2022 archives archives select month
 february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2)
 september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january
 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018
 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2)
 september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3)
 march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6)
 categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy
 development how to gbdk nes dev tips nes game development one game a month
 ponderings uncategorised refresh games side ads .sidebar .widget-area .site-
 content about finished games game jams gbdk gameboy dev how to gbdk .main-
 navigation refresh games privacy policy / .site-info .site-footer .site-inner
 .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'skip to
 content refresh games gameboy game development in gbdk & other game dev things
 .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk
 .main-navigation .site-header-menu .site-header-main .header-image .site-header
 category : gbdk gameboy development .page-header picking nes & game boy
 development back up again .entry-header life has certainly gotten in the way of
 a few things of late , the usual combination of commitments has shifted my focus
 around a lot .', 'probably every wednesday .', 'bee happy (nes game development
) this game is at that weird stage in development where it ' s fairly playable
 but needs some extra elements inside it for a good difficulty progression .',
 '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories
 gbdk gameboy development , nes game development .entry-footer # post- # # how
 high is your highscore ?', 'i ' ve found a handy solution to my problem , along
 the lines of the following : crayon syntax highlighter v_2.7.2_beta c //set up
 our test scores hiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2] = 9 ;
 hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0] = 0 ;
 score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ; score [5] = 9 ;
 higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 =
 higher than highscore i = 0 ; while (higher == 10) { if (hiscore1 [i] >
 score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher = 20
 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) { break
 ; } i++ ; } 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 //set up our test scores hiscore1 [0] = 0 ; hiscore1 [1] = 0 ; hiscore1 [2]
 = 9 ; hiscore1 [3] = 6 ; hiscore1 [4] = 9 ; hiscore1 [5] = 9 ; score [0]
 = 0 ; score [1] = 0 ; score [2] = 6 ; score [3] = 9 ; score [4] = 9 ;
 score [5] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore ,
 20 = higher than highscore i = 0 ; while (higher == 10) { if (hiscore1 [i]
 > score [i]) { higher = 0 ; } if (hiscore1 [i] < score [i]) { higher =
 20 ; } if (hiscore1 [i] == score [i]) { higher = 10 ; } if (i == 5) {
 break ; } i++ ; } [format time : 0.0006 seconds] what the code above does is
 compare the arrays on an integer by integer basis and set another variable with
 the result , reliably !', '.entry-content author rychan posted on 1st september

2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitter woos tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'skip to content refresh games gameboy game development in gbdk & other game dev things .site-branding menu about finished games game jams gbdk gameboy dev how to gbdk .main-navigation .site-header-menu .site-header-main .header-image .site-header category : gbdk gameboy development .page-header picking nes & game boy development back up again .entry-header life has certainly gotten in the way of a few things of late , the usual combination of commitments has shifted my focus around a lot .', 'probably every wednesday .', 'bee happy (nes game development) this game is at that weird stage in development where it ' s fairly playable but needs some extra elements inside it for a good difficulty progression .', '.entry-content author rychan posted on 14th june 2022 14th june 2022 categories gbdk gameboy development , nes game development .entry-footer # post- # # how high is your highscore ?', 'i ' ve found a handy solution to my problem , along the lines of the following : crayon syntax highlighter

```
v_2.7.2_beta c //set up our test scores
hiscore1 [ 0 ] = 0 ; hiscore1 [ 1 ] = 0 ;
hiscore1 [ 2 ] = 9 ; hiscore1 [ 3 ] = 6 ; hiscore1 [ 4 ] = 9 ; hiscore1 [ 5 ] =
9 ; score [ 0 ] = 0 ; score [ 1 ] = 0 ; score [ 2 ] = 6 ; score [ 3 ] = 9 ;
score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher = 10 ; // 10 = same score , 0 = lower
than highscore , 20 = higher than highscore
i = 0 ; while ( higher == 10 ) { if (
hiscore1 [ i ] > score [ i ] ) { higher = 0 ; } if ( hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if ( hiscore1 [ i ] == score [ i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i++ ; }
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 //set up our test scores
hiscore1 [ 0 ] = 0 ; hiscore1 [ 1 ] = 0 ; hiscore1 [ 2 ] = 9 ; hiscore1 [ 3 ] = 6 ; hiscore1 [ 4 ] = 9 ; hiscore1 [ 5 ] = 9 ; score [ 0 ] = 0 ; score [ 1 ] = 0 ; score [ 2 ] = 6 ; score [ 3 ] = 9 ; score [ 4 ] = 9 ; score [ 5 ] = 9 ; higher = 10 ; // 10 = same score , 0 = lower than highscore , 20 = higher than highscore
i = 0 ; while ( higher == 10 ) { if ( hiscore1 [ i ] > score [ i ] ) { higher = 0 ; } if ( hiscore1 [ i ] < score [ i ] ) { higher = 20 ; } if ( hiscore1 [ i ] == score [ i ] ) { higher = 10 ; } if ( i == 5 ) { break ; } i++ ; }
[ format time : 0.0006 seconds ] what the code above does is compare the arrays on an integer by integer basis and set
```

another variable with the result , reliably !', '.entry-content author rychan posted on 1st september 2016 categories game jams , gbdk gameboy development .entry-footer # post- # # posts navigation page 1 page 2 ... page 7 next page .site-main .content-area search me do search for : search twitterly woo tweets by @ refreshgamesuk recent posts februarys nes game - coffee run !', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', 'wzonka-lad , gameboy emulator for amiga machines is also under these pages .'}], {'tag': 'software', 'patterns': ['software', 'script', 'tools'], 'responses': ["[[gbdev.io [] [] games [] [] pan docs [] community community [] [] chat [] [] contribute [] [] gb competition 2021 [] [] resources [] guides guides [] [] choosing development tools [] [] asm style recommendations [] [] the timing of lyc stat handlers [] [] dead c scroll [] [] the wonders of prehistorik man [] [] dma hijacking [] [] [] [] games [] [] pan docs [] community community [] [] chat [] [] contribute [] [] gb competition 2021 [] [] resources [] guides guides [] [] choosing development tools [] [] asm style recommendations [] [] the timing of lyc stat handlers [] [] dead c scroll [] [] the wonders of prehistorik man [] [] dma hijacking [] [] [] resources [] [] introduction [] [] [] the game boy , a hardware autopsy [] [] the ultimate game boy talk [] [] [] documentation [] [] [] game boy color [] [] hardware [] [] peripherals [] [] cartridges [] [] emulator development [] [] [] testing [] [] [] software development [] [] [] assemblers [] [] compilers [] [] emulators [] [] tools [] [] [] programming [] [] [] asm [] [] c [] [] [] homebrews [] [] [] asm [] [] c [] [] [] gb studio [] [] [] demos [] [] [] reverse engineering [] [] [] game disassemblies [] [] [] game boy camera [] [] [] retrieving images [] [] [] changing the camera 's behavior [] [] [] post processing [] [] [] related projects [] [] [] directories [] [] [] websites [] [] [] [] [] [] [] [] awesome game boy development a curated list of awesome game boy (color) development resources , tools , docs , related projects and homebrews .", '# game boy color and super game boy this list is focused on the original (1989) game boy (dmg) , the game boy color (gbc) and super game boy (sgb) are very similar systems , with a few important distinctions , such as : different hardware specifications specific hardware and software features specific registers specific bugs , quirks and exploitable behaviours if you aim to

develop your software for sgb or gbc , or you want to know how it runs on the other systems , you may want to take advantage and adapt to these differences , check the game boy color category and look for specific references to gbc/cgb and sgb .', '# software development the choosing tools for game boy development essay provides an overview of the available development tools for game boy .', '# programming guides , tutorials and tools to develop software for game boy using the development toolchains described in the software development chapter .', 'gingerbread - a software library for making your own game boy games .', 'infinity - rpg developed by affenix software primarily between the years 1999 and 2001 .', 'dumping the super game boy boot rom # directories archive of related files the game boy archive - a library of game boy related software , hardware and literature .', 'the game boy archive - salvage - historical archive of software , old articles , faqs and various documents .', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017

(2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', '15th january 2023 more homebrew nes games 8th january 2023 old creative software still rocks 15 years later in 2022 2nd november 2022 retro game projects update me do 1st november 2022 archives archives select month february 2023 (1) january 2023 (2) november 2022 (2) june 2022 (2) september 2021 (1) may 2021 (3) april 2021 (11) march 2021 (1) january 2021 (2) march 2020 (1) february 2020 (2) february 2018 (1) january 2018 (4) april 2017 (2) january 2017 (1) november 2016 (1) october 2016 (2) september 2016 (2) july 2016 (1) june 2016 (2) may 2016 (2) april 2016 (3) march 2016 (3) february 2016 (15) january 2016 (3) december 2015 (6) categories 2018 nes dev atari vcs 2600 finished games game jams gbdk gameboy development how to gbdk nes dev tips nes game development one

game a month ponderings uncategorised refresh games side ads .sidebar .widget-area .site-content about finished games game jams gbdk gameboy dev how to gbdk .main-navigation refresh games privacy policy / .site-info .site-footer .site-inner .site lightbox plus colorbox v2.7.2/1.5.9 - 2013.01.24 - message : 0', "sign in hubringinfobar.html the gameboy development ring home > computers & internet > software > development > game programming hubnavsection.html about this ring dedicated to the development and programming of games , applications , and hardware using nintendo 's gameboy , and gameboy advance handheld gaming systems and/or gameboy emulators .", "amatuer developer software or a.d. software by amatuerdeveloper a.d. software is now making a game called `` the dark world '' for gbc and are looking for new people to join are team .", "dreamer 's dreamscape by gbdevring webring maintainer & general gameboy development site gameboy tech central gameboy hardware & software info .", 'it is dedicated to gameboy development and has the rarest programs , software and information .']], {'tag': 'programming', 'patterns': ['programming', 'coding', 'writing', 'write'], 'responses': ["[[gbdev.io [] [] games [] [] pan docs [] community community [] [] chat [] [] contribute [] [] gb competition 2021 [] [] resources [] guides guides [] [] choosing development tools [] [] asm style recomendations [] [] the timing of lyc stat handlers [] [] dead c scroll [] [] the wonders of prehistorik man [] [] dma hijacking [] [] [] [] games [] [] pan docs [] community community [] [] chat [] [] contribute [] [] gb competition 2021 [] [] resources [] guides guides [] [] choosing development tools [] [] asm style recomendations [] [] the timing of lyc stat handlers [] [] dead c scroll [] [] the wonders of prehistorik man [] [] dma hijacking [] [] [] resources [] [] introduction [] [] the game boy , a hardware autopsy [] [] the ultimate game boy talk [] [] documentation [] [] game boy color [] [] hardware [] [] peripherals [] [] cartridges [] [] emulator development [] [] testing [] [] software development [] [] assemblers [] [] compilers [] [] emulators [] [] tools [] [] programming [] [] asm [] [] c [] [] homebrews [] [] asm [] [] c [] [] gb studio [] [] demos [] [] reverse engineering [] [] game disassemblies [] [] game boy camera [] [] retrieving images [] [] changing the camera 's behavior [] [] post processing [] [] related projects [] [] directories [] [] websites [] [] [] [] [] [] awesome game boy development a curated list of awesome game boy (color) development resources , tools , docs , related projects and homebrews .", "reiner ziegler 's game boy page - commercial and homemade programmable cartridges and programming systems .", 'api docs : getting started examples documentation , links and tools turbo rascal syntax error - complete suite (ide , compiler , programming language , resource editor) intended for developing games/demos for 8 / 16-bit line of computers , including the game boy and game boy color .', 'dot matrix game editor - an ide for game boy programming in a c-like language called gbl , with many other features like tile and map extraction , wla-dx assembly , and more .', '# programming guides , tutorials and tools to develop software for game boy using the development toolchains described in the software development chapter .', "assemblydigest - exploring game boy programming techniques : making an empty game boy rom (in

wiz) making art for the game boy beginner 's guide to reverse engineering gb
- some starting tips on disassembling and reverse engineering .", 'gameboy
programming tutorial : hello world !', 'dmgreport - game programming tutorials
in assembly .', 'game boy assembly programming for the modern game developer -
an e-book about making game boy games in assembly .', 'it is made to be used
alongside the book game boy assembly programming for the modern game developer
which also doubles as documentation .', 'grooves game boy programming - a
complete set of lessons about implementing various game mechanics in a game boy
game .', 'gbdk configuring and programming tutorial - configuring gbdk , using
tiles , colliding sprites , gbtd , gbmb , memory management and rom banking .',
'simplified gbdk examples gbdk programming video tutorials - a series of video
tutorials introducing beginners to programming with gbdk .', 'digitized speech
in game boy games sniffing game boy serial traffic with an stm32f4 virtual
game boy printer with an stm32f4 printing on the game boy printer using an
stm32f4 programming game boy chinese cartridges with an stm32f4 pokemon pocket
computer : - what is it and how to use it to make cheat codes .', 'show ()
helper functions to ease the programming i created several helper functions that
create layers in my graph .', '" | thoughts on future decoded 2014 - part 1 :
the keynotes how to check for visibility on a html control using microsoft coded ui
tests recent comments archives june 2015 april 2015 november 2014 september
2014 may 2014 march 2014 categories technical uncategorized meta register log in
entries feed comments feed wordpress.com search for : peterwynroberts.com menu #
menu-toggle widgets # sidebar-toggle search # search-toggle .toggles # masthead
gameboy programming tutorial : hello world !', '.entry-header this tutorial will
guide you through the steps of programming a rom which , when run using a
gameboy emulator , will write the words " hello world " on the screen .', 'some
stuff here might also be a bit difficult to follow if you ' ve never done any
programming before - and , to be honest , if you haven ' t programmed before ,
then jumping right into assembler like this is probably not the best way to
learn .', 'they ' re there because of a particularly important concept in
gameboy programming , called interrupts .', 'a lot of it was also borrowed from
the sample code john harrison wrote for a course in gameboy programming at
wichita state university - you can see more about that here .', 'links a course
in gameboy programming at wichita state university .', 'rgbds tools and vim
syntax file example code share this : twitter facebook like this : like loading
... .entry-content may 11 , 2014 peter assembly language , coding , gameboy ,
programming , tutorial , z80 .entry-meta # post- # # post navigation ← → .nav-
links .navigation leave a reply cancel reply enter your comment here ...\nfill
in your details below or click an icon to log in : email (required) (address
never made public) name (required) website you are commenting using your
wordpress.com account .', '" | thoughts on future decoded 2014 - part 1 : the
keynotes how to check for visibility on a html control using microsoft coded ui
tests recent comments archives june 2015 april 2015 november 2014 september 2014
may 2014 march 2014 categories technical uncategorized meta register log in
entries feed comments feed wordpress.com search for : peterwynroberts.com menu #
menu-toggle widgets # sidebar-toggle search # search-toggle .toggles # masthead
gameboy .page-header gameboy programming tutorial : hello world !', '.entry-
header this tutorial will guide you through the steps of programming a rom which

, when run using a gameboy emulator , will write the words " hello world " on the screen .', 'some stuff here might also be a bit difficult to follow if you ' ve never done any programming before - and , to be honest , if you haven ' t programmed before , then jumping right into assembler like this is probably not the best way to learn .', 'they ' re there because of a particularly important concept in gameboy programming , called interrupts .', 'a lot of it was also borrowed from the sample code john harrison wrote for a course in gameboy programming at wichita state university - you can see more about that here .', 'links a course in gameboy programming at wichita state university .', 'rgbds tools and vim syntax file example code .entry-content may 11 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80 leave a comment .entry-meta # post- # # # main # primary # content create a free website or blog at wordpress.com .', 'i feel it needs re-programming now i have a better knowledge of the system as opposed to in october last year .', 'i feel it needs re-programming now i have a better knowledge of the system as opposed to in october last year .', 'i feel it needs re-programming now i have a better knowledge of the system as opposed to in october last year .', "sign in [hubringinfo.com](#) the gameboy development ring home > computers & internet > software > development > game programming [hubnavsection.html](#) about this ring dedicated to the development and programming of games , applications , and hardware using nintendo 's gameboy , and gameboy advance handheld gaming systems and/or gameboy emulators .", 't : h : e : o : r : ys website on my site you will find things under the topics of : art , music , programming and gameboy colour programming .']}, {'tag': 'assembly', 'patterns': ['assembly', 'hex', '0x', 'low level'], 'responses': ['# game boy color bootstrap rom unused palettes colorization palettes in the bios rom disassembly # hardware dmg schematics - hardware schematics .', '# misc introduction to game boy hacking - workshop introducing basic assembly , debugging and reverse engineering .', 'binjgb on the web - porting of the binjgb emulator to web assembly .', 'porting a go game boy emulator to webassembly about swotgb - notes about the development of a game boy emulator in javascript .', 'wiz - a high-level assembly language for writing homebrew on retro console platforms (game boy , nes , atari 2600 , and more) .', 'gbasm-rs - an opinionated rust based compiler for game boy z80 assembly code .', 'gbasm - a javascript based compiler for game boy z80 assembly code .', 'runs in the browser using webassembly .', 'dot matrix game editor - an ide for game boy programming in a c-like language called gbl , with many other features like tile and map extraction , wla-dx assembly , and more .', 'evunit - a unit testing program for assembly code .', 'gb-convert - game boy tile conversion and map editor tool (converts to assembly) .', 'asmschool - a set of lessons by duo about coding in assembly for gb/gbc and disassembling .', 'assembly tutorial by david pello - good document to learn to produce working asm code for gb .', "assemblydigest - exploring game boy programming techniques : making an empty game boy rom (in wiz) making art for the game boy beginner 's guide to reverse engineering gb - some starting tips on disassembling and reverse engineering .", 'dmgreport - game programming tutorials in assembly .', 'oam dma tutorial - example of how to use oam dma in assembly .', 'game boy assembly programming for the modern game developer - an e-book about making game boy games in assembly .', 'game boy assembly language

primer - simple template code with memory defines , copy routines and ibm font
 tilemap .', 'gameboy boilerplate - boilerplate project to move quicker into the
 actual assembly code for your game .', 'it is made to be used alongside the book
 game boy assembly programming for the modern game developer which also doubles
 as documentation .', '# syntax highlighting packages atom language package for
 rgbasm - atom syntax highlighting for rgbds assembly .', 'gbz80-highlight -
 notepad+- and gedit syntax highlighting files for rgbds assembly .', 'vim syntax
 file for the game boy assembler rgbasm - vim syntax highlighting for rgbds
 assembly .', 'vim syntax file for rgbds - another vim syntax highlighting file
 for rgbds assembly .', 'z80 assembly support for visual studio code rgbds-
 vscode - visual studio code language extension for rgbds gbz80 assembly .',
 'rgbds-mode - emacs major mode for rgbds assembly .', "link 's awakening dx
 oracle of ages tetris - complete tetris disassembly .", 'the code in this
 tutorial will all be the assembly language used by the game boy processor ,
 which is a slightly modified version of the zilog z80 .', 'if you don ' t know
 any assembly language , don ' t panic !', 'before you go on , however , you need
 a few things first : an editor so you can actually write some code ; an
 assembler and linker which will allow you to make your assembly language program
 into an actual , working rom ; and finally , an emulator so you can run your
 program .', 'to view and edit assembly language , i use vim , because a) it ' s
 free , and b) it has a few features , like syntax highlighting and line
 numbering , which make it fairly useful for coding .', 'to get a working rom
 from your assembler source file , simply store the rgbds tools in the same
 folder as your assembly code .', 'rgbds tools and vim syntax file example code
 share this : twitter facebook like this : like loadingentry-content may 11
 , 2014 peter assembly language , coding , gameboy , programming , tutorial , z80
 .entry-meta # post- # # post navigation ← → .nav-links .navigation leave a reply
 cancel reply enter your comment here ...\nfill in your details below or click an
 icon to log in : email (required) (address never made public) name (
 required) website you are commenting using your wordpress.com account .', '[
 0x00000100] > pd 1024 ~ 2f3 0x000002e4 21f302 ld hl , 0x02f3 pd stands for p
 rint d isassembly .', '00000274) 0x000002ed c9 ret s is used to s eek to
 address and pdf stands for p rint d isassembly f unction radare recognized that
 our function begins at 0x274 .', 'although it ' s a simple lines of code , the
 disassembly looked awful .', 'the code in this tutorial will all be the assembly
 language used by the game boy processor , which is a slightly modified version
 of the zilog z80 .', 'if you don ' t know any assembly language , don ' t panic
 !', 'before you go on , however , you need a few things first : an editor so you
 can actually write some code ; an assembler and linker which will allow you to
 make your assembly language program into an actual , working rom ; and finally ,
 an emulator so you can run your program .', 'to view and edit assembly language
 , i use vim , because a) it ' s free , and b) it has a few features , like
 syntax highlighting and line numbering , which make it fairly useful for coding
 .', 'to get a working rom from your assembler source file , simply store the
 rgbds tools in the same folder as your assembly code .', 'rgbds tools and vim
 syntax file example code .entry-content may 11 , 2014 peter assembly language ,
 coding , gameboy , programming , tutorial , z80 leave a comment .entry-meta #
 post- # # # main # primary # content create a free website or blog at

```
wordpress.com .', "stephen 's gameboy page by sgstair this site aims to bring
the resources of assembly game boy development into one , findable area .", "old
bits some example gameboy code : - rand ( ) implementations in assembly , now
used in gbdk- conway 's life- sub-pixel accuracy scrolling on the cgb herg 's
gameboy development contains the games i have written for the gameboy including
gbdk source code ."]], {'tag': 'tutorials', 'patterns': ['tutorials',
'tutorial'], 'responses': ['tutorials , wiring and schematics provided .', '#
programming guides , tutorials and tools to develop software for game boy using
the development toolchains described in the software development chapter .',
'dmgreport - game programming tutorials in assembly .', 'simplified gbdk
examples gbdk programming video tutorials - a series of video tutorials
introducing beginners to programming with gbdk .', '# websites devrs.com/gb -
old home of the scene : examples , sources , complete documentation , guides ,
tutorials and various tools .', '[ endif ] projects recently updated most likes
newest project lists discover contests courses stack more courses tutorials
events hackerspaces profiles hackaday.com tindie marketplace sign up log in
close 0 % 0 % nekocart - cpld gameboy cartridge open-source gameboy flash
cartridge using an xilinx cpld as mbc wenting zhang following follow project
liked like project become a hackaday.io member not a member ?', 'lots of
sourcecode , apps and tutorials .', 'gameboy developement tutorials my gameboy
developement site will soon contain many thorough tutorials on making games on
the gameboy color gamempire its a site which hasnt been finished yet about
gameboy .']]]}
```

```
[ ]: nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

```
[ ]: True
```

```
[ ]: words = []
labels = []
docs_x = []
docs_y = []

for intent in kb['intents']:
    for pattern in intent['patterns']:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent['tag'] not in labels:
        labels.append(intent['tag'])

print(intent)
```

```
{'tag': 'tutorials', 'patterns': ['tutorials', 'tutorial'], 'responses':
['tutorials , wiring and schematics provided .', '# programming guides ,
tutorials and tools to develop software for game boy using the development
toolchains described in the software development chapter .', 'dmgreport - game
programming tutorials in assembly .', 'simplified gbdk examples gbdk
programming video tutorials - a series of video tutorials introducing beginners
to programming with gbdk .', '# websites devrs.com/gb - old home of the scene :
examples , sources , complete documentation , guides , tutorials and various
tools .', '[ endif ] projects recently updated most likes newest project lists
discover contests courses stack more courses tutorials events hackerspaces
profiles hackaday.com tindie marketplace sign up log in close 0 % 0 % nekocart -
cpld gameboy cartridge open-source gameboy flash cartridge using an xilinx cpld
as mbc wenting zhang following follow project liked like project become a
hackaday.io member not a member ?', 'lots of sourcecode , apps and tutorials .',
'gameboy developement tutorials my gameboy developement site will soon contain
many thorough tutorials on making games on the gameboy color gamempire its a
site which hasnt been finished yet about gameboy .']}]}
```

```
[ ]: words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))

labels = sorted(labels)

training = []
output = []

out_empty = [0 for _ in range(len(labels))]

for x, doc in enumerate(docs_x):
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1

    training.append(bag)
    output.append(output_row)

training = numpy.array(training)
```

```
output = numpy.array(output)
```

```
[ ]: tf.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)
```

WARNING:tensorflow:From /usr/local/lib/python3.9/dist-packages/tflearn/initializations.py:164: calling TruncatedNormal.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.

Instructions for updating:

Call initializer instance with the dtype argument instead of passing it to the constructor

```
[ ]: model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
```

```
Training Step: 5999 | total loss: 0.49600 | time: 0.024s
| Adam | epoch: 1000 | loss: 0.49600 - acc: 0.9713 -- iter: 40/42
Training Step: 6000 | total loss: 0.44971 | time: 0.028s
| Adam | epoch: 1000 | loss: 0.44971 - acc: 0.9742 -- iter: 42/42
--
```

```
[ ]: model.save("/content/drive/MyDrive/model.tflearn")
```

```
[ ]: import pickle

with open("/content/drive/MyDrive/data.pickle", "wb") as f:
    pickle.dump((words, labels, training, output), f)
```

```
[ ]:
```