#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

int main(int argc, char\*\* argv) {

if (argc != 2) {

printf("Usage: %s image\_file\_name\n", argv[0]);

return EXIT\_FAILURE;

}

int nCanal = 4;

int i = 0;

//CV\_LOAD\_IMAGE\_COLOR = 1 forces the resultant IplImage to be colour.

//CV\_LOAD\_IMAGE\_GRAYSCALE = 0 forces a greyscale IplImage.

//CV\_LOAD\_IMAGE\_UNCHANGED = -1

IplImage\* Img1 = cvLoadImage(argv[1], CV\_LOAD\_IMAGE\_UNCHANGED);

IplImage\* ImgChess = cvCreateImage(cvSize(512, 512), Img1->depth, nCanal);

// Always check if the program can find a file

if (!Img1) {

printf("Error: fichero %s no leido\n", argv[1]);

return EXIT\_FAILURE;

}

// a visualization window is created with title 'image'

cvNamedWindow("chess", CV\_WINDOW\_NORMAL);

// img is shown in 'image' window

cvShowImage("chess", ImgChess);

cvWaitKey(0);

for (int vert = 0; vert < 8; vert++) {

for (int fila = 0; fila < ImgChess->height / 8; fila++) {

char \*pImgChess = ImgChess->imageData + (fila \* ImgChess->widthStep)+ (vert \* ImgChess->widthStep \* (ImgChess->height / 8));

for (int casilla = 0; casilla < 8; casilla++) {

for (int col = 0; col < ImgChess->width / 8; col++) {

for (int canal = 0; canal < nCanal; canal++) {

if (i % 2 == 0) {

\*pImgChess = 100;

} else {

\*pImgChess = 0;

}

pImgChess++;

}

}

i++;

}

cvShowImage("chess", ImgChess);

cvWaitKey(5);

}

i++;

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}