Labor

Exercise 13 „Platforms for Embedded Systems“ Prof. Cochlovius

- Rasterization with Bresenham und Wu -

Goal:
During this exercise you learn how to map continuous, geometric shapes to a discrete pixel space. We will use the line algorithms of Bresenham and the antialiasing approach of Wu as examples.

Exercise 13.1: Bresenham for the first Octant

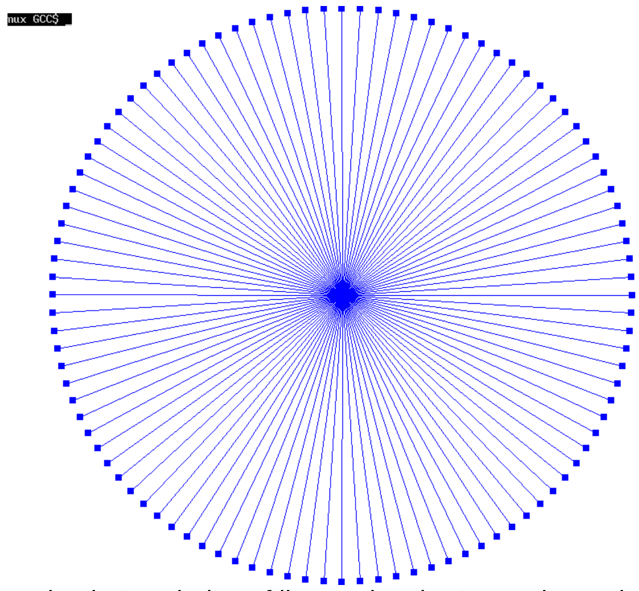
Create a new C-Project in Eclipse (either as Managed-Make or as a Makefile project). Copy the quickstart program **`fbMove2_bsh.c`** into your project. Create the executables for host and target. Walk through the code to understand its functionality: Which steps are required to access the frame buffer?

Note: Again, when running the X11 window system, the access to the frame buffer device is blocked. To avoid conflicts with X11, we switch to a virtual console to run our graphics programs. This can be done with Ctl+Alt+F1. Switching back to X11 is done by Ctl+Alt+F7.

After switching to a virtual console, we run the executable (manually) and check the result:

Exercise 13.2: Now we extend to eight Octands

Copy your code into a new project (or *.c-file). Extend the draw() routine to cover the remaining seven octands.
Hint: http://rosettacode.org/wiki/Bitmap/Bresenham%27s_line_algorithm#C



```
nux GCC$ _
```

Fig.: simple Rendering of lines using the Bresenham Algorithm

Exercise 13.3: And now lines with Alphablending

Copy your code into a new project (or *.c-file). Extend the program to render the Gray scale background in the beginning. Then extend the **`put_pixel()`** routines to include an additional parameter to define the alpha value and to process it properly.

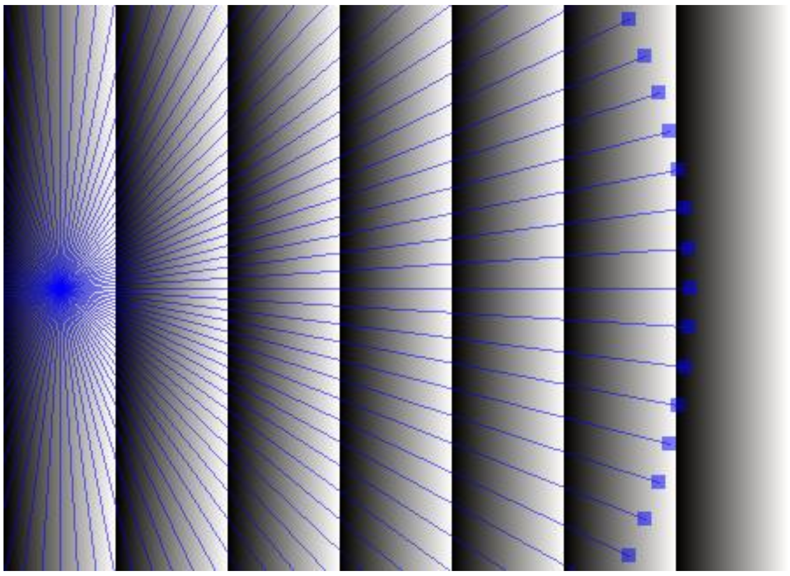


Fig.: Rendering of Lines based on Bresenham; this time including Alphablending

Exercise 13.4: Antialiasing with Wu’s Line Algorithm

On http://rosettacode.org/wiki/Xiaolin_Wu%27s_line_algorithm#C you find the raw C code of Wu’s line algorithm. Study its implementation. (**Note:** this is a pretty straightforward implementation. A much more sophisticated and efficient implementation can be found on http://www.drdobbs.com/database/graphics-programming/184408790?pgno=3). Import and adpt the routine into your setting: Create a routine **`draw_line_wu()`** which utilizes your **`put_pixel()`** functions.

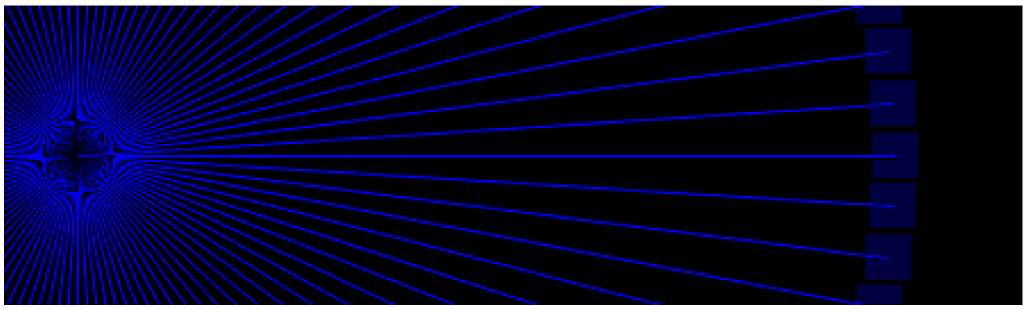


Fig.: Rendering of Lines using Wu's Algorithm, i.e. including Antialiasing