

## Aufgabenblatt 11

Julian Bertol

Alle Code Dateien sind auf Felix!

### Aufgabe 1

Fbset ausgabe:

Host:

```
mode "800x600"  
geometry 800 600 2048 2048 32  
timings 0 0 0 0 0 0  
rgba 8/16,8/8,8/0,0/0  
endmode  
hfu@host085:~$
```

Das sind Informationen zur Auflösung und Farbtiefe

Host:



Target:



## Aufgabe 2

Put\_pixel funktion:

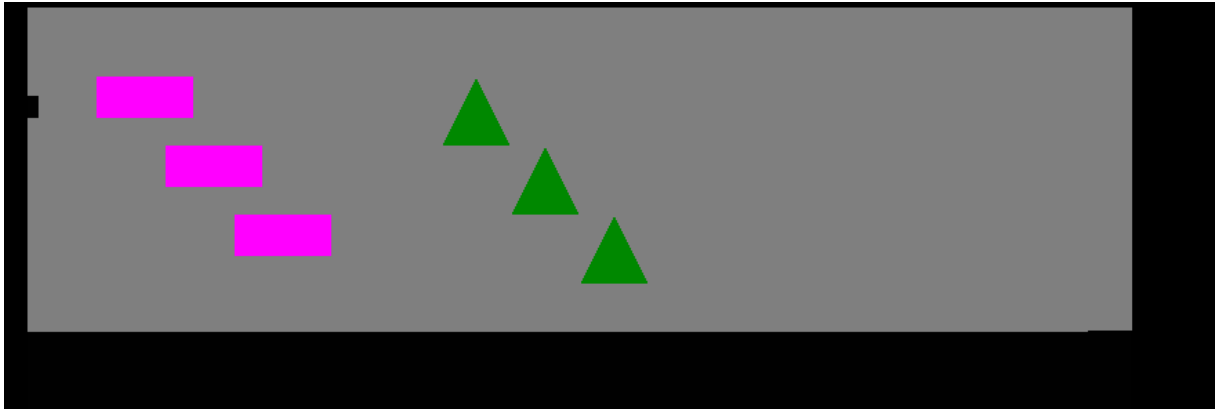
```
void put_pixel_RGB24(int x, int y, int r, int g, int b)
{
    // calculate the pixel's byte offset inside the buffer
    // note: x * 3 as every pixel is 3 consecutive bytes
    // line length: in byte
    unsigned int pix_offset = x * 3 + y * finfo.line_length;

    // now this is about the same as 'fbp[pix_offset] = value'
    // Achtung: cast nach char* für byte-weisen Speicherzugriff!!
    *((char*)(fbp + pix_offset)) = r;
    *((char*)(fbp + pix_offset + 1)) = g;
    *((char*)(fbp + pix_offset + 2)) = b;
}

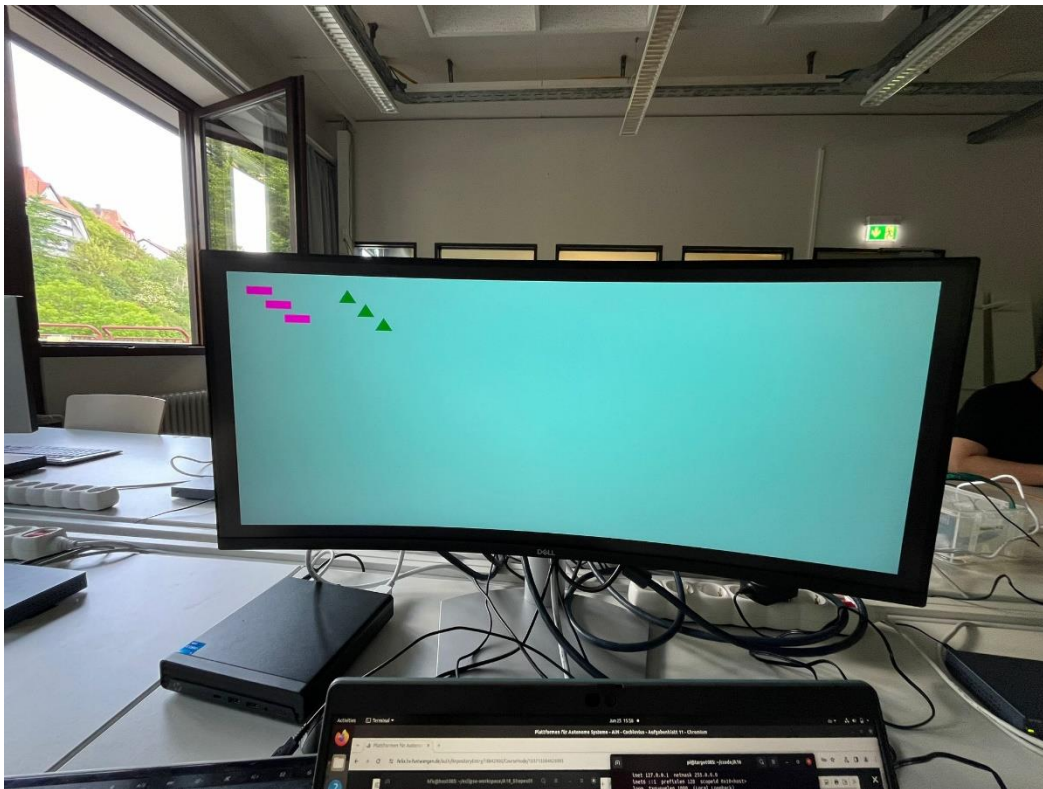
// C: Farbmodell RGB565, d.h. 1 Pixel beansprucht 2 Byte
void put_pixel_RGB565(int x, int y, int r, int g, int b)
{
    unsigned int pix_offset = x * 2 + y * finfo.line_length;
    unsigned int short c = ((r/8) << 11) + ((g/4) << 5) + (b/8);
    *((unsigned short*)(fbp + pix_offset)) = c;
}
```

### Aufgabe 3

Host:

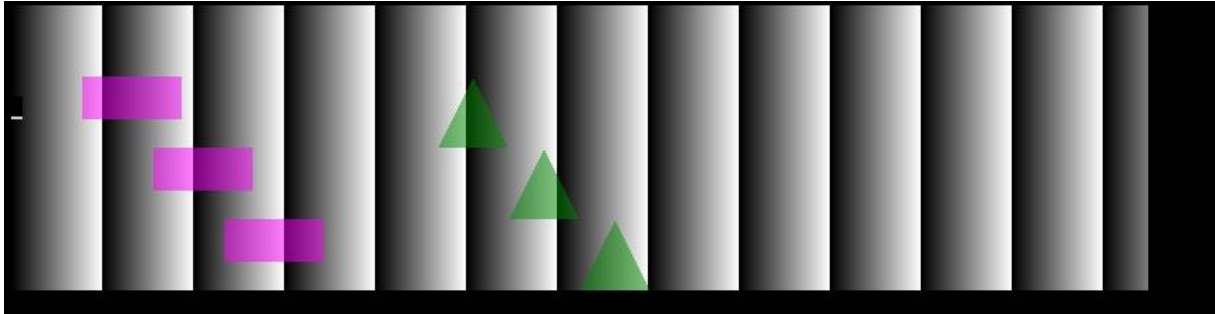


Target:



## Aufgabe 4

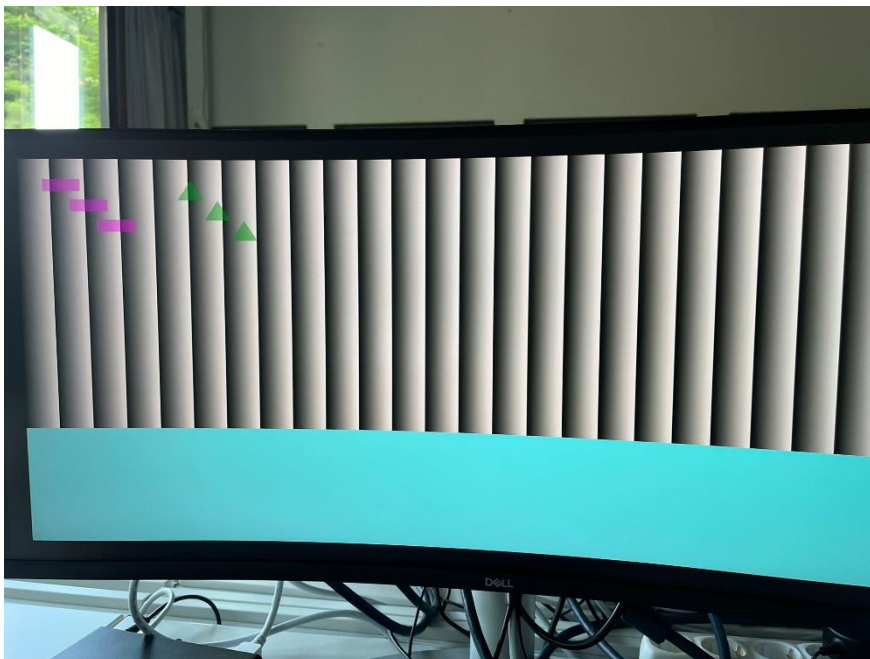
Host:



Für das Target muss man die Funktion put\_pixel\_RGB565 anpassen:

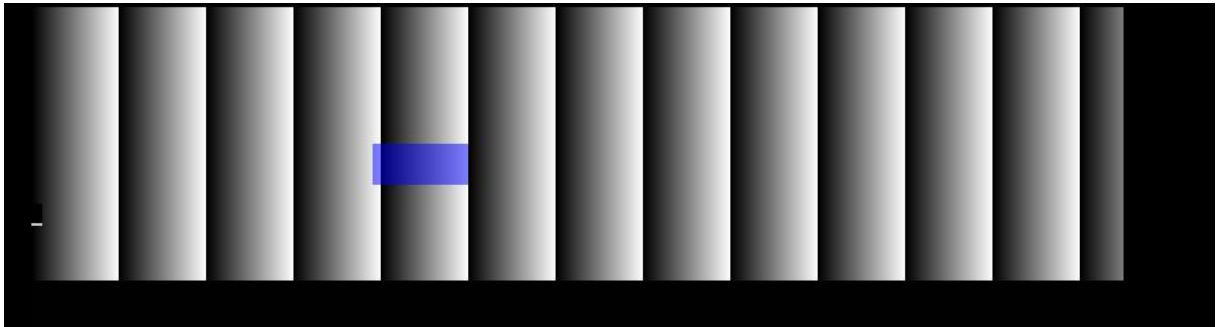
```
void put_pixel_RGB565(int x, int y, int r, int g, int b, float a) {  
    //Berechnet den Byteversatz eines Pixels innerhalb des buffers. finfo.line_length gibt die Breite einer Zeile in Bytes  
    unsigned int pix_offset = x * 2 + y * finfo.line_length;  
  
    //Hintergrundwerte der Farben bekommen, fuer Softwareberechnung der Transparenz a:  
    unsigned short c = *((unsigned short*) (fbp + pix_offset));  
    int backgroundB = (c & 0x1F) * 8;  
    int backgroundG = ((c >> 5) & 0x3F) * 4;  
    int backgroundR = ((c >> 11) & 0x1F) * 8;  
  
    //Berechnen des alpha-blendings:  
    r = (a * r + (1 - a) * backgroundR);  
    g = (a * g + (1 - a) * backgroundG);  
    b = (a * b + (1 - a) * backgroundB);  
  
    //Werte setzen:  
    c = ((r / 8) << 11) + ((g / 4) << 5) + (b / 8); //Rot, Gruen und Blau werden entsprechend RGB565 umgewandelt  
    *((unsigned short*) (fbp + pix_offset)) = c; //Zuerst wird auf einen Zeiger auf short dereferenziert und dann auf s  
}
```

Target:



## Aufgabe 5

Host:



Für das Target muss man die Funktion `put_pixel_RGB565` anpassen. Siehe Aufgabe zuvor.

Target:

