

Perspective

by Bitbreaker/Oxyron/Nuance

Best is to calculate the perspective during multiply with the rotation matrix. As soon as you get the value for Z, lookup a corresponding factor in a table and multiply the results for X and Y with that factor:

```

... matrix multiplication for Z ...
tay
lda z_fact,y
sta z1
eor #$ff
sta z2
... matrix multiplication for X ...
tay
;multiply with z_fact
lda (z1),y
sec
sbc (z2),y
sta final_x
... matrix multiplication for Z ...
tay
;multiply with z_fact
lda (z1),y
sec
sbc (z2),y
sta final_y

```

The factor-table you could generate like the following:

```

...
d = 280.0; z0 = 5.0;
for (i = 0; i < 0x100; i++) {
    z = i;
    //make things signed
    if(z > 127) z = z - 256;
    q = round(d/(z0-z/64.0));
    //take care that values are sane
    if(q > 127) q = 127;
    if(q < -127) q = -127;

    if(q < 0) q = 256 + q;

    result[i] = q;
}
...

```

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
<https://codebase64.org/doku.php?id=base:perspective>

Last update: **2015-04-17 04:33**

