**int** **Vgascreen::draw\_ellipse**(**int** x\_mp, **int** y\_mp, **int** x\_rad, **int** y\_rad, **int** color, **int** fill)

{

**int** w = **int**(x\_rad/2);

**int** h = **int**(y\_rad/2);

**int** hh = h \* h;

**int** ww = w \* w;

**int** hhww = hh \* ww;

**int** x0 = w;

**int** dx = 0;

// do the horizontal diameter

**for** (**int** x = -w; x <= w; x++)

{

**if** (fill != **true**)

{

**if** (x == -w || x == w)

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp, color);

}

**else**

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp, color);

}

// now do both halves at the same time, away from the diameter

**for** (**int** y = 1; y <= h; y++)

{

**int** x1 = x0 - (dx - 1); // try slopes of dx - 1 or more

**for** (; x1 > 0; x1--)

**if** (x1 \* x1 \* hh + y \* y \* ww <= hhww)

**break**;

dx = x0 - x1; // current approximation of the slope

x0 = x1;

**for** (**int** x = -x0; x <= x0; x++)

{

**if** (fill != **true**)

{

**if** (x == -x0 || x == x0)

{

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp - y, color);

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp + y, color);

}

}

**else**

{

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp - y, color);

**UB\_VGA\_SetPixel**(x\_mp + x, y\_mp + y, color);

}

}

}

oude code werkte beter

std::**swap**(h, w);

std::**swap**(hh, ww);

// do the vertical diameter

**for** (**int** x = -w; x <= w; x++)

{

**if** (fill != **true**)

{

**if** (x == -w || x == w)

**UB\_VGA\_SetPixel**(x\_mp, y\_mp + x, color);

}

**else**

**UB\_VGA\_SetPixel**(x\_mp, y\_mp + x, color);

}

// now do both halves at the same time, away from the diameter

**for** (**int** y = 1; y <= w; y++)

{

**int** x1 = x0 - (dx - 1); // try slopes of dx - 1 or more

**for** (; x1 > 0; x1--)

**if** (x1 \* x1 \* hh + y \* y \* ww <= hhww)

**break**;

dx = x0 - x1; // current approximation of the slope

x0 = x1;

**for** (**int** x = -x0; x <= x0; x++)

{

**if** (x == -x0 || x == x0)

{

**UB\_VGA\_SetPixel**(x\_mp - y, y\_mp + x, color);

**UB\_VGA\_SetPixel**(x\_mp + y, y\_mp + x, color);

}

}

}

**return** 0;

}