Circle generation using the Midpoint Circle Algorithm

Consider a circle segment of 45^o running from $x = 0$ until $x = y = \frac{R}{\sqrt{2}}$, where R is the radius of the circle and $(0, 0)$ the center of the circle. Starting at point $(0, R)$ and following the X-axis in positive direction, pixels are generated for increasing x (increment by 1) and by decreasing y by an integer value, if necessary.
NB: Values of x, y and R are integer. By defining the viewport boundaries large enough, this isn't a problem.

The function $F(x, y) = x^2 + y^2 - R^2$ equals 0 on the circle, is larger than 0 outside the circle and less than 0 within the circle. Starting from a pixel in (x_p, y_p), this function is used to compute the value of y at $x_p + 1$. If d equals the value of the function in between $(x_p + 1, y_p)$ and $(x_p + 1, y_p - 1)$ (the midpoint at $(x_p + 1, y_p - \frac{1}{2})$), then:

$$d_{old} = F(x_p + 1, y_p - \frac{1}{2}) = (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2$$

If $d_{old} < 0$, the midpoint lies inside the circle (or the circle lies outside the midpoint). The value of y_p is not decremented and the next pixel on the circle has coordinates $(x_p + 1, y_p)$. Now the next midpoint value $(x_p + 2, y_p - \frac{1}{2})$ is given by:

$$d_{new} = F(x_p + 2, y_p - \frac{1}{2}) = (x_p + 2)^2 + (y_p - \frac{1}{2})^2 - R^2 = d_{old} + (2x_p + 3) = d_{old} + \Delta_{noincy}$$

If $d_{old} >= 0$, The midpoint lies on or outside the circle, so the value of y_p is decremented and the next pixel has coordinates $(x_p + 1, y_p - 1)$. The next midpoint $(x_p + 2, y_p - \frac{3}{2})$ is given by:

$$d_{new} = F(x_p + 2, y_p - \frac{3}{2}) = (x_p + 2)^2 + (y_p - \frac{3}{2})^2 - R^2 = d_{old} + (2x_p - 2y_p + 5) = d_{old} + \Delta_{incy}$$

So depending on the value of d the value of y is decremented or not and the value of d is incremented by Δ_{incy} or Δ_{noincy} to compute the next midpoint.

The initial value of d is computed at point $(0, R)$. The next midpoint lies at $(1, R - \frac{1}{2})$ and this gives:

$$d_{begin} = F(1, R - \frac{1}{2}) = 1 + (R^2 - R + \frac{1}{4}) - R^2 = \frac{5}{4} - R$$

Since x and y are integer values, d is the only floating point variabele in this computation. However, it is possible to convert d to an integer variable to speed up the computation. Defining $h = d - \frac{1}{4}$ and substituting $h + \frac{1}{4}$ for d, gives $h = 1 - R$ as initialization, while the equation $d < 0$ is substituted by $h < \frac{1}{4}$. Because h can be used as integer and because only integer values Δ_{incy} and Δ_{noincy} are added to it, **the whole algorithm can be executed in integer**.

Finally, each pixel that is generated in the segment of 45^o, can be mirrored to generate a full circle.