```
for( int y=0; y<h; y++ ) {
    Color startColor = getRandomColor();
    Color endColor = getRandomColor();

    for( int x=0; x<w; x++ ) {

        Color *blend;

        if( x < w/2 ) {
            blend = Color::linearSum( &endColor, &startColor,
                                      x / (float) (w/2) );
            }
        else {
            blend = Color::linearSum( &startColor, &endColor,
                                      ( x - w/2 ) / (float) (w/2) );
            }

        setPixel( x, y, *blend );

        delete [] blend;
        }
    }

Color tempColors[h][w];

for( int y=0; y<h; y++ ) {
    for( int x=0; x<w; x++ ) {

        Color sumColor( 0, 0, 0 );

        for( int dy = -1; dy <= 1; dy++ ) {
            for( int dx = -1; dx <= 1; dx++ ) {

                int ny = y + dy;
                int nx = x + dx;

                if( ny >= h ) {
                    ny -= h;
                    }
                else if( ny < 0 ) {
                    ny += h;
                    }

                if( nx >= w ) {
                    nx -= w;
                    }
                else if( nx < 0 ) {
                    nx += w;
                    }



                Color neighborColor = getPixelColor( nx, ny );

                for( int c=0; c<3; c++ ) {
                    sumColor[c] += neighborColor[c];
                    }
                }
            }
        for( int c=0; c<3; c++ ) {
            sumColor[c] /= 9;
            }


        tempColors[y][x] = sumColor;
        }
    }

for( int y=0; y<h; y++ ) {
    for( int x=0; x<w; x++ ) {
        setPixel( x, y, tempColors[y][x] );
        }
    }

char markedPixels[h][w];

for( int y=0; y<h; y++ ) {
    for( int x=0; x<w; x++ ) {
        markedPixels[y][x] = false;
        }
    }

for( int j=0; j<3; j++ ) {

    int x = w/2;
    int y = h/2;

    int hitEdge = false;
    for( int s=0; s<2 *w && !hitEdge; s++ ) {

        markedPixels[y][x] = true;
        markedPixels[y][w-x-1] = true;

        if( coinFlip() ) {
            x += randomMove();
            }
        else {
            y += randomMove();
            }

        if( x < 0 || x >= w || y < 0 || y >= h ) {
            hitEdge = true;
            }
        }

    }

for( int y=0; y<h; y++ ) {
    for( int x=0; x<w; x++ ) {

        if( ! markedPixels[y][x] ) {
            setPixel( x, y, white );
            }
        }
    }
```