User Interface - Bug #3973

Veiwport area coordinates rounding error

03/19/2019 12:50 PM - Eugenie Lyzenko

Status:	New	**Start date:**	03/19/2019
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#1 - 03/19/2019 02:04 PM - Eugenie Lyzenko

The testcase to show the error will be not easy to write. Consider the problematic code in Viewport.draw():

```
   public void draw()
   {
      // need to minimize rounding errors to avoid UI artifacts
      Point origin = null;
      Dimension dim = null;
      if (isChui)
      {
         origin = screenLocation();
         dim = dimension();
      }
      else
      {
         NativePoint np = screenPhysicalLocation();
         NativeDimension nd = physicalDimension();
         // extra pixel to compensate rounding errors
         if (np.x > 0)
         {
            np.x--;
            nd.width++;
         }
         if (np.y > 0)
         {
            np.y--;
            nd.height++;
         }
         origin = cc.pointFromNative(np);
         dim = cc.dimensionFromNative(nd);
      }

      Rectangle rect = new Rectangle(origin, dim, screen().coordinates().baseUnits());

      ScreenBitmap bitmap = getAndSetBitmap(rect);
      scrollWidget.draw();
      setBitmap(bitmap);
   }
```

The FWD uses character based rectangle to define the bitmap to draw in. The unit precision is based on Java double value which is good enough to provide proper rectangle in ChUI mode.

But in GUI mode we have a bit different situation. The measurement units become a integer based pixels. And this is what we have on all modern raster based screens. Here the issue starts. We need to convert double value into integer one by formula: pixelValue = pixelsPerChar * charValue. The result for pixelValue needs some kind of rounding, either classical round() or custom ceil()/floor(). The question is which one to choose in different cases(especially when pixelsPerChar * charValue is small but not absolute 0)

The problem symptoms are missing pixel image on the edges of the drawing area. The current workaround uses additional pixels for the edges to compensate rounding errors effect. But we need to investigate if it is possible to have something better to resolve pixel fraction missing feature(we can not have 0.XX part of the pixel on screen and this is solid constraint of the raster display).