Università della Svizzera italiana

Faculty of Informatics

# Computer Graphics (Fall 2021)

## Assignment 7: Rasterization

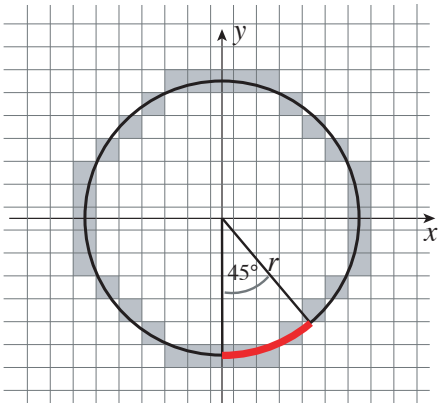
November 3, 2021

### Exercise 1 [7 points]

Let the camera opening angle be $\frac{3}{4}\pi$ radians and the window be $15 \times 15$ pixels large. Which pixels does the midpoint algorithm (without anti-aliasing) set for the line from $p_1 = (1, 1, 4)$ to $p_2 = (2, 1, 1)$, given in global coordinates? And which pixels are set if we replace $p_2$ by $p_2' = (3, 1, -2)$ ? Is that correct, and if not, then which pixels should be set instead?

### Exercise 2 [8 points]

Consider the problem of rasterizing a circle. Derive a version of Bresenham's algorithm for this task, and sketch a pseudo-code for it. Assume that your circle is already defined using pixel indices. To simplify your task, assume that the pixel indices can be negative, and the center of the circle is located at $(0, 0)$. The radius of the circle is $r$. Note that in your derivation, you only have to consider $1/8$ of the circle marked in red (see picture on the right). The rest of the pixels can be computed using symmetries. More precisely, if you color pixel $(x, y)$, you should also color: $(-x, y)$, $(x, -y)$, $(-x, -y)$, $(y, x)$, $(-y, x)$, $(y, -x)$, $(-y, -x)$. By deriving the algorithm for this small piece of the circle, you also can assume that you have to color only one pixel per column. This is similar to the assumption about $45°$ slope in the original line drawing algorithm. The derivation of the algorithm for the circle follows the same approach as for the line and uses a mid-point idea together with an implicit circle equation, which should have different sign depending on whether you are inside or outside of the circle.




### Submission

Submit a single PDF including all the calculations you did do solve the assignments to iCorsi. You are allowed to use calculator. Indicate on top of the PDF all the authors of the solutions.

**Solutions must be returned on November 10, 2021 via iCorsi3**