

Computer Graphics 1

Project 3

Subject: Rasterization

The goal of this project is to create an application with graphical user interface for creating and editing simple vector graphics - lines, polygons and circles.

Requirements

- implementation assigned algorithms, one of each:
 - line drawing algorithm,
 - thick lines drawing algorithm,
 - circle drawing algorithm,
 - anti-aliasing algorithm,
- drawing lines:
 - drawing lines with a mouse (e.g. by first clicking to select the starting point of a line and then clicking to select the end point),
 - editing existing lines with a mouse (moving end points),
 - deleting existing lines,
 - changing the thickness of individual lines
- drawing circles:
 - drawing circles with a mouse (e.g. by first clicking to select the center of a circle and then clicking on a point on the circle),
 - editing existing circles with a mouse (moving circle and changing its radius),
 - deleting existing circles

- drawing polygons:
 - drawing polygons with a mouse (e.g. by clicking to select the next vertex of a polygon and clicking in the proximity of a first vertex to close the polygon),
 - editing existing polygons with a mouse (moving vertices, edges and entire polygon),
 - deleting existing polygons,
 - changing the thickness of all the edges of individual polygons
- changing the color of individual shapes,
- option to remove all of the shapes clearing the screen,
- option to turn on and turn off anti-aliasing for all the shapes on the screen,
- loading and saving shapes to a single files containing vector information, that is only properties of each shape like vertices position, color etc.,

Remarks

All the drawing must be done using only single pixel operations

Interaction with the application may be handled mainly by the graphical user interface. All actions performed with keyboard shortcuts should be possible to do using application GUI.

All the listed algorithms are described on the 5th lecture slides. The slide number is given in the parenthesis next to each algorithm.

Line drawing algorithms

- DDA Algorithms (s. 6)
- Midpoint Line algorithms (s. 11)
- Symmetric Midpoint Line (s. 12)

Thick line drawing algorithms

Line thickness can be limited to an odd number of pixels.

- Copying pixels - for each pixel drawn by a line drawing algorithm the same color value is copied to the appropriate number of pixels above and below (for more "horizontal lines" i.e. $|dx| > |dy|$) or to the left and to the right (for more "vertical lines" i.e. $|dx| < |dy|$) currently drawn pixel,

- Brush - brush is defined as $n \times n$ array containing the pattern of a brush, where n is a thickness of a line. When drawing the line for each pixel the pattern needs to be pasted around currently drawn pixel. The brush should be in a shape of a circle.

Circle drawing algorithms

- Midpoint Circle (s. 15),
- Alternative Midpoint Circle - using only additions (s. 17)

Anti-Aliasing algorithms

- Gupta-Sproull algorithms (s. 28) - used only for the line anti-aliasing,
- Xiaolin Wu line and circle drawing (s. 30-32) - used for circles and lines. When drawing the lines their thickness is ignored