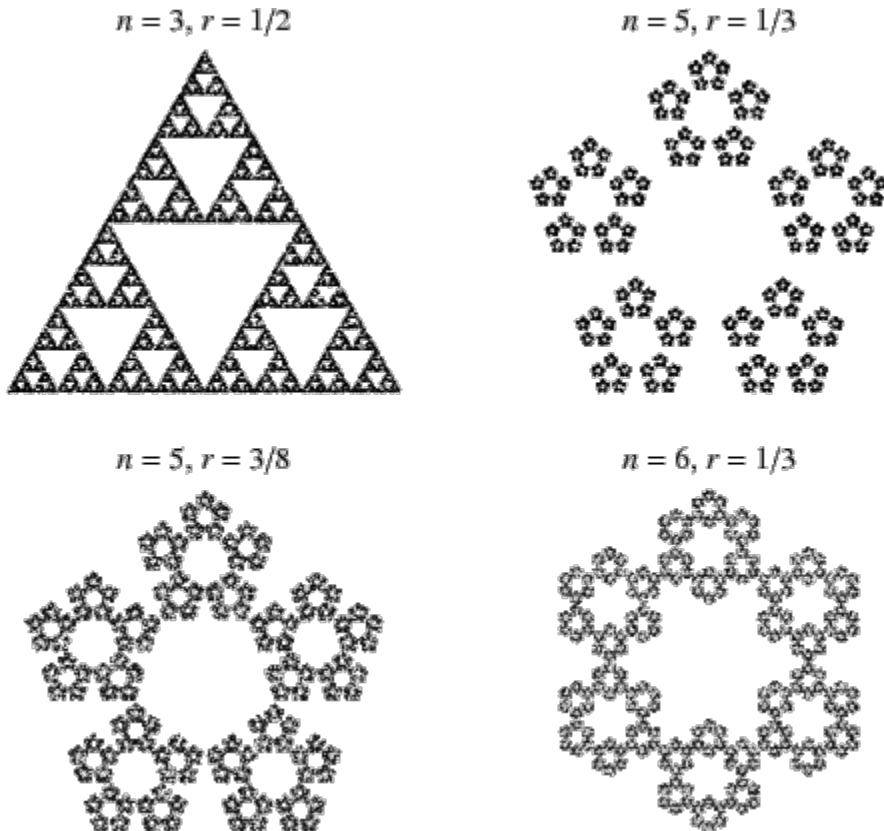




1st Task

Construction of imaging AND moving fractals

The aim of the task is to write a program in C programming language that will draw and move two fractals (one at a time), see the top right and bottom right figure:



For the design (top right fractal) we work as follows:

1. We choose 5 marks, the vertices of a regular pentagon.
2. We take a point q at a random position inside the pentagon
3. We randomly select one of the vertices of the pentagon and find the point p that is at $r=1/3$ of the distance from the current point q to the selected vertex (closest to the vertex).
4. We draw p and consider it a current point q .
5. Repeat steps 3 & 4 up to a number of repetitions. Applying the same algorithm to a regular hexagon with $r=1/3$, we get the bottom right figure in the image above.

The program you write should draw the selected shape in a graphics window using commands from the GL, GLU & GLUT libraries /Freeglut. When the program is started, one fractal (e.g. top right) is drawn which results after 15000 iterations (points). Pressing the right mouse button on the graphics window should open a pop-up menu with the following options:

- 8000 points
- 15000 points, random color
- Hexagon $r=1/3$
- Pentagon $r=1/3$
- Terminate

The first two options redraw the shape using the corresponding number of points with the second option giving a different colour to each point. The third and fourth options determine the fractal to be drawn. The last option terminates the program. Use double buffering so that the shape is drawn in the graphics window complete.

When the mouse is moved with the left button pressed, the shape moves continuously according to the mouse movement (the shape starts from where it is and follows the cursor movement until it stops, when it stays in that position). Implement this movement by appropriately shifting the observation window (defined by `gluOrtho2D`) via a corresponding callback function to handle mouse movement with a pressed key.

To write the program, use as a model Sierpinski's triangle design program, `gasket.c` (from Angel's book) included in the same zip file.