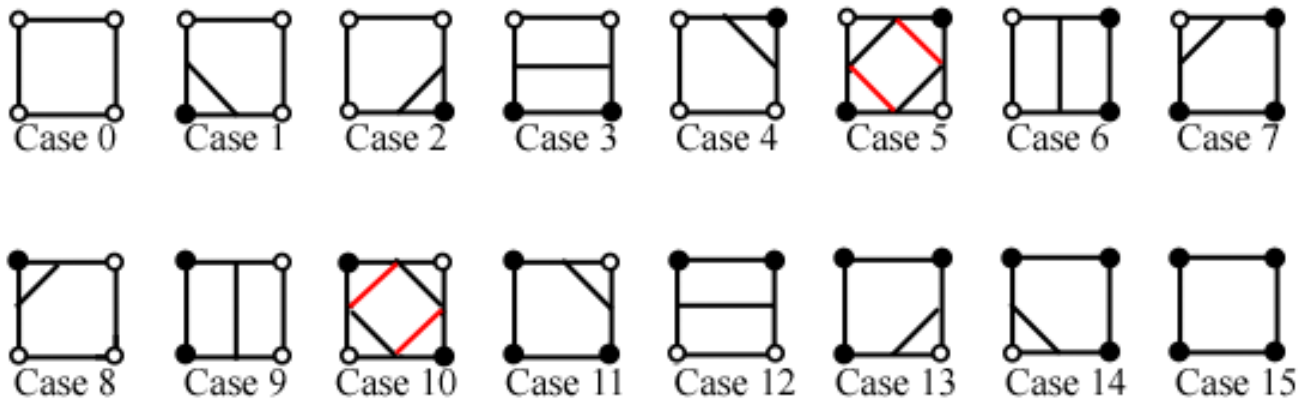
Programming Assignment #6 (Optional)

In this programming assignment you will implement the marching squares algorithm in order to draw approximations of the 2D boundary of 2D implicit equations.



Requirements:

Requirement 1 (20%) – Grid: You will use a grid to determine the size of each square. The grid has to be implemented with the following key controls:

- **z:** will switch between drawing the grid or hiding it
- **q:** will increase the resolution of the grid
- **a:** will reduce the resolution of the grid

Requirement 2 (20%) – Evaluation: Each cell of the grid represents an evaluation square of the marching squares algorithm. Therefore each vertex of the grid represents one evaluation point. In order to show the result of the point evaluations, at each grid vertex you will have to draw a small point, circle or square around the grid vertex with two possible colors. One color meaning that the point was evaluated as inside the curve, and another color for when the point is evaluated outside.

- **x:** will switch between drawing the evaluation points or hiding them

Requirement 3 (25%) – Curve Approximations: You will then apply the marching square rules to connect lines approximating the boundary of the implicit curve being evaluated. Automatically re-compute and update these lines every time a key is pressed for changing the grid resolution.

Requirement 4 (25%) – Curve Selection: You will have to implement in your code at least five different implicit curves, and allow the user to switch between them by pressing keys 1, 2, 3, 4, and 5. *At least two curves have to generate more than one connected component* (more than one "separate boundary").

Here are some curves you may use to test your code:
x^2-y^2=1
x^2 + y^2 + sin4x + sin4y = 1

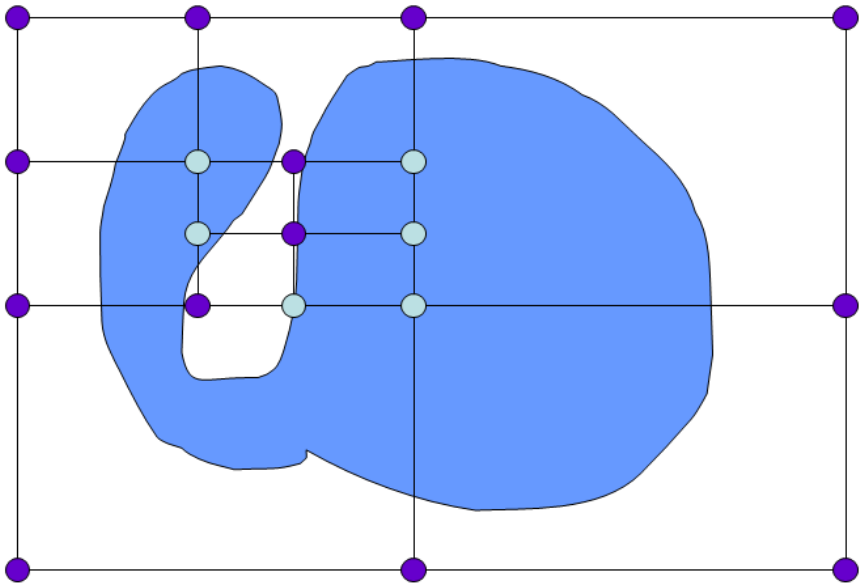
Requirement 5 (10%) – Quadtree Adaptive Subdivision: For this last requirement you will implement in your code an adaptive subdivision of your grid, following the quadtree hierarchical decomposition scheme. Starting from your initial grid, the adaptive resolution is achieved by only further subdividing the cells with mixed evaluations. Given the current global resolution (Requirement 1), you will then advance in adaptive resolution increases whenever a key is pressed. The idea is to first perform global evaluations with the global grid resolution, and then proceed with adaptive resolution.

Key controls:

w: will increase the maximum depth of the adaptive quadtree, increasing the overall resolution of the approximation,

s: will decrease the maximum depth of the adaptive quadtree, decreasing the overall resolution of the approximation.

Here is an example illustrating the quadtree subdivision scheme applied at two quadrants:



This project is entirely in 2D and all the required display can be accomplished with just 2D line segments. You may use several line objects, one for each type of information being displayed.

Submission:

Please follow the instructions in parules.txt (uploaded to CatCourses). In particular: please do not include any third-party support code and do not forget to Clean Solution before preparing your project for submission! Also, check for hidden folders (such as .vs) which can sometimes balloon to hundreds of megabytes!

Deadline lab for presentation and submission: your last lab.