|  |  |  |
| --- | --- | --- |
|  | FACULTY OF GERMAN ENGINEERING EDUCATION AND INDUSTRIAL MANAGEMENT |  |

**Implementation of a QuadTree Visialisation alghorithm using   
Visual Studio, Windows Forms and C#**

Prepared by:

Martin Vladimirov Gebov

201210026

Technical University of Sofia

Sofia 1000

Bulgaria

April 2016

# What are quadtrees ?

A quadtree is a tree data structure in which each internal node has exactly four children. Quadtrees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions

# Objectives

Create a graphical user interface that given a sample image (generated or imported) should draw the resulting quadtree. The program must contain the following functionality:

1. Two functions that draw a predefined pattern. This will enable the comparison if the generated quadtree to an existing one.
2. A function that clears the image and leaves it blank
3. A function that takes the currently draw image and draws a quadtree out of it.
4. A function that imports an image from the file system and generates its quadtree.
5. A function that draws the treeview of the generated quadtree

# Implementation

First we begin by creating a Square class that would be used to split the imported (generated) image into squares. The Square class holds two private fields - a Point and a side. Those are needed to specify the starting position of the square and are afterwards used to draw the square itself. Following is the implementation of the square class:



We then add a Node class representing each of the nodes in the quadtree The Node class will contain a reference to its underlying children. The children will be represented as four quadrants of the current Node and will be named from First to Fourth in clockwise order. The Node class will know the color of the node and whether it is homogenous or not (homogenous meaning that the Node comprises of just a single color):



First we create the two functions that will generate a predefined heterogeneous pattern. Following is the function that draws a rectangle taking a starting point, width and height as parameters.



We must then devise an algorithm that will enumerate all of the pixels of the source image and draw the corresponding quadtree. The algorithm will use recursion and will traverse all of the pixels of the image by dividing the image into four quadrants every time more than one pixel color is found in the quadrant. Following is the implementation of the algorithm:



Next we need to implement the importing of an image from the file system. For this we first implement a file open dialog mechanism from which to select the image to be imported and afterwards we place the image in a square and fill the square with whitespace in the blank parts. Following is the code:



Lastly we implement the tree view generation algorithm that traverses all of the nodes of the quadtree and displays their color, depth and position:

