Graph Editor programmers documentation

Technical data

Program was developed in visual studio 2010 sp1. For 3D visualization I'm using Microsoft XNA Game Studio 4.0 .

Program can be run on windows XP or higher with installed Microsoft framework 4.0.

Solution

Solution is divided in 4 projects

- Graph definition
- Graph editor
- Plugin interface
- Test plugin

Graph definition

In this project are all classes describing graph with positioning and colors. But there is no definition of drawing process.

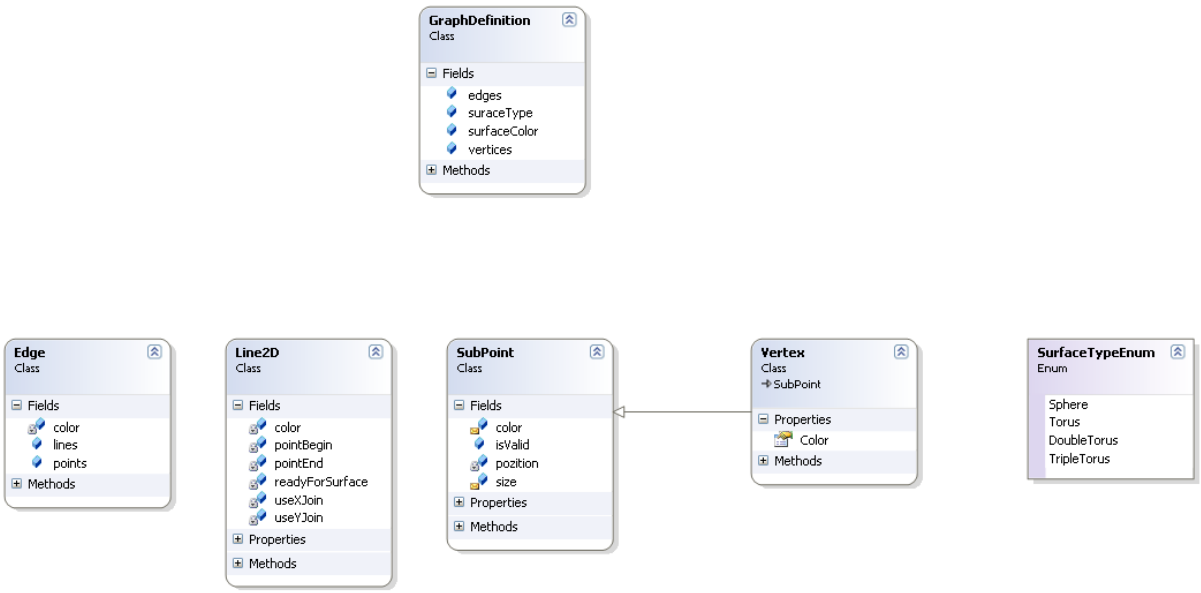

Figure 1: Graph definition class diagram

GraphDefinition

Main class is GraphDefinition which holds:

- List of edges
- List of vertices
- Surface type and surface color

Edge
Edge is descripted in Edge class. Because edge can be an polygonal line, edge is constructed from list of Line2D.

Line
Line 2D always have begin and end point pointer. There are also two Booleans which stores whether lines are strait or going round Torus.

Sub Point
Sub point represents a point on end of line. if it is also end of Edge it is called Vertex(have special inherited class)

It stores a position. Position is stored relatively in interval [0,1] in x and y axes. For position storing is PointF type used.

Graph Editor
This is the main project. The is a Main window description. Main window is described in Xaml. There are two main components on Main window: canvas and Xna hosing component.

Graph
This class holds graph. It stores graph definition and defines the drawing methods. It also caches 3D object witch are necessary for 3D visualization. Two main methods are draw2D and draw3D. Drawing is done by Drawers

Drawers
Drawers defines drawing of all graph components. Ale drawers have IDrawer interface implemented which forces to implement draw2D and draw3D method.

All drawers are almost same. In constructor it takes object to draw as parameter and then defines drawing methods.

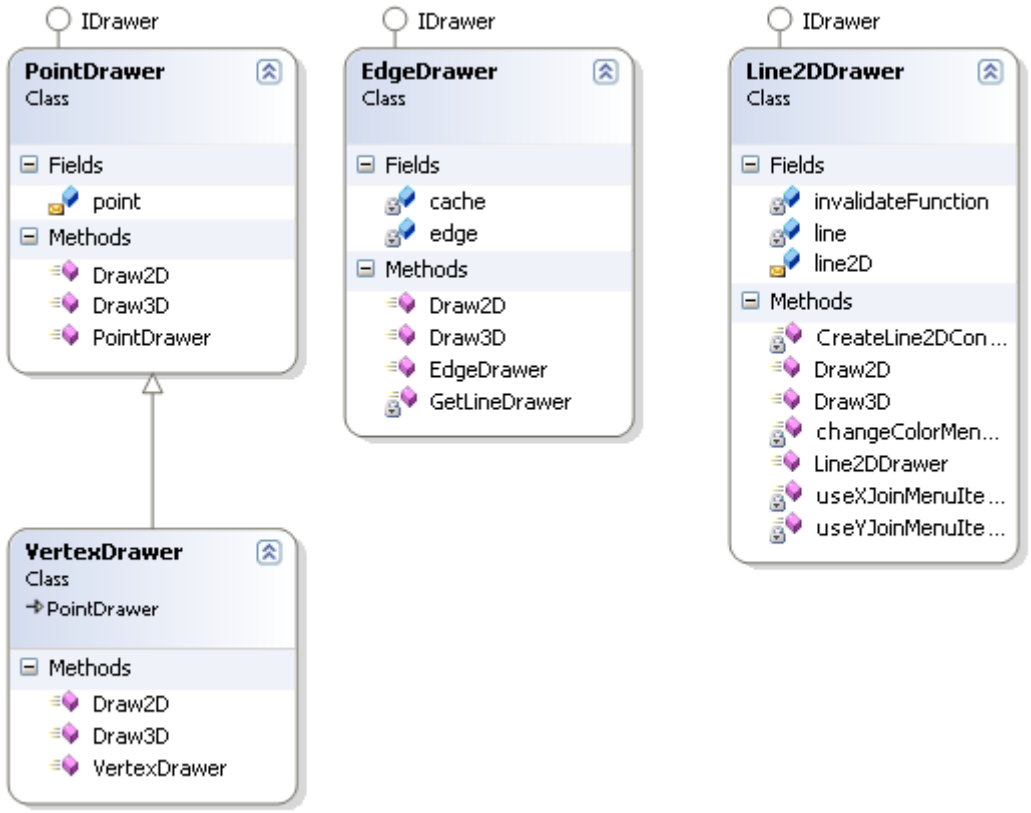

Figure 2: drawers object structure

Plugin interface

This project describes only one but important class. It defines an abstract class. This class must be used as mother class for each plugin.

```
namespace AddInView
{
    public abstract class IPozitioning :MarshalByRefObject
    {
        public abstract GraphDefinition Pozitioning(GraphDefinition sourceDefinition);
    }
}
```

Figure 3: Abstract class for plugins

Test plugin

There is one plugin for load plugin testing. This plugin just take all point in graph and set position to random position. This plugin is loaded to own app domain for security reasons.