Fachhochschule
Nordwestschweiz

Module Design Patterns
Prof. Dr. Dominik Gruntz
Prof. Dr. Wolfgang Weck

Assignment 5: Snap To Grid

In this assignment we want to add (at least) one grid to the graphics editor. It should be possible to set a grid which asserts that while constructing and resizing figures only selected points can be used.

In the interface `DrawView` methods `setGrid` and `getGrid` are defined, these methods can be used to set and get a grid. Such a grid is used to constrain the coordinates. The interface `jdraw.framework.DrawGrid` is provided:

```
public interface DrawGrid {
   // Returns constrained coordinates for p, e.g. rounded to a grid.
   Point constrainPoint(Point p);

   // Returns the horizontal step size when the selection is moved with the arrow keys.
   int getStepX(boolean right);

   // Returns the vertical step size when the selection is moved with the arrow keys.
   int getStepY(boolean down);

   // Activates the grid. This method is called whenever method setGrid is called on a draw view.
   void activate();

   // Deactivates the grid. This method is called whenever another grid
   // is installed. Use this method to do some clean-up when the grid is switched.
   void deactivate();

   // Indicates that a mouse interaction was just started. This method might be used to
   // setup data used during this mouse interaction.
   void mouseDown();

   // Indicates that a mouse interaction has been finished. This method might be used to
   // clean-up data created in method mouseDown.
   void mouseUp();
}
```

`StdDrawView` calls the registered grid to map the mouse coordinates before calling methods `mouseDown`, `mouseDrag` and `mouseUp` respectively on the current draw tool. When the selection is moved with the arrow keys, the horizontal and the vertical sizes `getStepX` and `getStepY` are used.

How the mouse coordinates are mapped is to be defined by the concrete implementation of the grid. Possible implementations map the coordinates to a fixed grid or alternatively the mouse coordinates may be rounded to the coordinates of handle anchors of existing figures (a so called snap grid).

The implementations of your grids have to be registered in the menu (it is intended that the grids are added to the Edit menu). If you have implemented many variants it is recommended to define a separate sub menu.

The Problem of the design of the interface for the grid is that this interface has to be powerful enough so that all possible strategies can be implemented. If you implement a snap grid strategy, you will probably reach some limits. If this is the case, then propose an extension of the strategy interface `DrawGrid`. Which additional parameters would have to be added to the methods `constrainPoint`, `getStepX` and `getStepY` and also (if needed) to the methods `mouseDown` and `mouseUp`?

Deadline: October 30, 2018