

Übung 5: Snap To Grid

Der Grafikeditor soll um (mindestens) ein Grid erweitert werden, d.h. man soll den Editor so einstellen können, dass beim Konstruieren, Vergrössern und Verkleinern von Figuren nur noch bestimmte Punkte angesprungen werden können.

Im Interface DrawView sind Methoden setConstrainer und getConstrainer vorgesehen, mit denen ein Point-Constrainer gesetzt und abgefragt werden kann. Das Interface `jdraw.framework.PointConstrainer` ist gegeben:

```
public interface PointConstrainer {
    // 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 point constrainer. This method is called whenever the
    // method setConstrainer is called on a draw view.
    void activate();

    // Deactivates the point constrainer. This method is called whenever another point constrainer
    // is installed. Use this method to do some clean-up when the constrainer 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();
}
```

In der Klasse StdDrawView werden die Mauskoordinaten mit dem installierten „PointConstrainer“ abgebildet bevor die Methoden mouseDown, mouseDrag und mouseUp aus dem aktuell gesetzten Draw-Tool aufgerufen werden. Beim Verschieben einer Selektion mit den Cursor-Tasten werden ebenfalls die vom „Point-Constrainer“ bestimmten Grössen getStepX und getStepY verwendet.

Wie die Mauskoordinaten gerundet werden, ist den konkreten Constrainer-Implementierungen überlassen. Mögliche Implementierungen sind, dass nur noch bestimmte Koordinaten auf einem Grid angesprungen werden, oder, dass in der Nähe von Figuren die Koordinaten dieser Figuren oder ihrer Handles verwendet werden (*Snap Grid*).

Die Implementierung muss in der Klasse StdContext im Menu registriert werden (vorgesehen ist das *Edit Menu*). Da Sie möglicherweise mehrere Varianten realisieren werden, haben wir für die Grids ein Untermenü vorgesehen.

Das Problem beim Entwurf der Schnittstelle für den Point-Constrainer ist, dass diese mächtig genug sein muss, um alle möglichen Strategien realisieren zu können. Wenn Sie eine Snap Grid Strategie implementieren, so werden Sie möglicherweise an Grenzen stossen. Falls ja, so schlagen Sie doch eine Erweiterung der Strategieschnittstelle für das Interface PointConstrainer vor. Was müssten bei den Methoden constrainPoint, getStepX und getStepY und allenfalls auch bei den Methoden mouseDown und mouseUp für zusätzliche Parameter vorgesehen werden?