Assignment 10: Undo & Redo

In this assignment we extend the graphics editor with an undo/redo functionality. In order to do that, we have to provide all operations such as insertion and deletion of a figure as well as moving and resizing of a figure as commands. In the JDraw framework, commands are defined with the interface `DrawCommand`:

```
public interface DrawCommand {
     // Executes a command
     void redo();

     // Undoes the action performed by execute.
     void undo();
}
```

Method `redo` executes a command and method `undo` makes the command undone. As an example, for the insertion operation an `InsertCommand` has to be implemented whose `redo` method inserts the figure and whose `undo` method removes the figure from the model. The remove command is slightly more complicated as its `undo` method has to insert the removed figure at the same place in the ordered list of figures as it was before removal.

The model delegates the management of the commands to a separate object which implements the interface `DrawCommandHandler`. This draw command handler is provided by the draw model and can be accessed with method `getDrawCommandHandler`. In your current draw model implementation this method returns an empty handler (an instance of class `EmptyDrawCommandHandler`).

```
public interface DrawCommandHandler {

     // adds a command to the command chain
     public void addCommand(DrawCommand cmd);

     // clears the command chain
     public void clearHistory();

     // undoes or redoes commands in the command chain
     public void undo();
     public void redo();

     // Indicates whether an undo or redo operation is possible.
     public boolean undoPossible();
     public boolean redoPossible();

     // Method to open/close a script.
     public void beginScript();
     public void endScript();
}
```

With the Move- and Resize-Operations we have the problem that they consist of several small mouse movements (and thus we end with many move or resize commands). It would be desirable to collect these operations to one macro operation. This functionality is provided in the interface `DrawCommandHandeler` with the methods `beginScript` and `endScript`. Operations which are added between the calls to `beginScript` and `endScript` are internally combined to *one* operation which can be undone with one undo invocation.

The Undo and Redo menu entries are already provided and class `StdDrawView` registers commands in the draw command handler provided by the model for the removal and the move operations. Methods `beginScript` and `endScript` are called in the `MouseMotionListener` methods which are implemented in class `StdDrawView` (e.g. for the move operation).

Tasks:

1. Extend your tool implementations to add a command to the draw command handler whenever a new figures is added to the model.
2. The handles have to register the changes they perform on a figure as commands as well.
3. Provide an implementation of the `DrawCommandHandler` interface in your model.

Deadline: January 08, 2019