# Undo,Redo功能设计文档说明书

## 功能需求

Canvas中元素的各种操作支持有限步骤的撤销、重做功能。具体操作暂定包括：单元素重命名，单元素删除，单元素添加，单元素移动，单条线删除，单条画线。**多元素的操作后续再考虑。**

## 设计方案

### 2.1 数据结构

1，Undo,Redo两个固定大小的栈，目前默认大小各为100，既支持100步撤销、重做，已经能满足大部分需求，无限步骤的撤销、重做业务上没啥意义，还浪费内存。

2，ICommand接口，内含Do, Rollback两个动作，用来封装所有的单元素执行和回撤动作。

3，UndoRedoManager,全局单例，用来管理Undo、Redo两个固定栈和协同界面Undo、Redo按钮事件等操作。

### 2.1 算法设计

1， 所有单元素完成操作的程序点，将相关上下文数据封装到ICommand中，然后Push进入Undo栈（由于是固定栈，所以当元素个数大于固定值时(默认100)，自动替换栈底元素），同时清空Redo栈。

2， 当点击Undo按钮时，Pop出Undo栈栈顶ICommand元素，然后执行Rollback接口，回归对应操作，最后将ICommand元素Push进入Redo栈。

3， 当点击Redo按钮时，Pop出Redo栈栈顶ICommand元素然后执行Do接口，执行对应操作，最后将ICommand元素Push进入Undo栈。

该操作与第二步对称。

1. 当Undo栈由空变不空时，Undo按钮Enable;当Undo栈由不空变空时，Undo按钮Disable;当Redo栈由空变不空时，Redo按钮Enable;当Redo栈由不空变空时，Redo按钮Disable;以上四种操作，由UndoRedoEvent的四种事件通知主界面
2. 在单元素的6种操作完成点，根据不同的操作类型封装ICommand接口所需的各种信息：

|  |
| --- |
| ElementRenameCommand |
| ElementAddCommand |
| ElementDeleteCommand |
| RelationAddCommand |
| RelationDeleteCommand |
| ElementMoveCommand |