Part	Туре	#	Description	Resource	Path	File
rait	FXML		Complete EditorFrame.fxml with all fx:ids and evt some "On Action"	EditorFrame.fxml	application	application/EditorFrame.fxml
	TODO		Figure#hasFillColor	Figure.java	figures	figures/Figure.java
	TODO		Figure#getFillColor	Figure.java Figure.java	figures figures	figures/Figure.java figures/Figure.java
	TODO		Figure#hasEdgeColor	Figure.java	figures	figures/Figure.java
	TODO		Figure#getEdgeColor	Figure.java	figures	figures/Figure.java
	TODO		Figure#setEdgeColor Figure#setLineType	Figure.java Figure.java	figures figures	figures/Figure.java figures/Figure.java
	TODO		Figure#setLineWidth	Figure.java	figures	figures/Figure.java
	TODO		Figure#setSelected	Figure.java	figures	figures/Figure.java
6	TODO		Figure#applyParameters Figure#updateSelectionFrame	Figure.java Figure.java	figures figures	figures/Figure.java figures/Figure.java
eati	TODO		Figure#equals(Object)	Figure.java	figures	figures/Figure.java
ole c	TODO	013	Controller#initialize: setting up #drawingModel	Controller.java	application	application/Controller.java
Initial Circle creation	TODO		Controller#initialize: Binds properties of UI elements to #drawingModel Controller#initialize: Setting up #historyManager	Controller.java Controller.java	application application	application/Controller.java application/Controller.java
nitis	TODO		Setup #shapeTypeComboBox	Controller.java	application	application/Controller.java
	TODO	017 018	Setup #lineTypeCombobox Controller#initialize: Setup #useFillColor, #useEdgeColor, #fillColorPicker and #edgeColorPicker	Controller.java	application	application/Controller.java
	TODO		Controller#initialize: Setup #userincoid, #userugecolor, #inicolorricker and #edgecolorricker Controller#initialize: Setup #lineWidthSpinner with an new SpinnerValueFactory	Controller.java Controller.java	application application	application/Controller.java application/Controller.java
	TODO	020	Controller#initialize: Setup #figuresListView with	Controller.java	application	application/Controller.java
	TODO	021	Controller#initialize: Setup #messagesLabel with empty or null message Controller#setTools	Controller.java Controller.java	application application	application/Controller.java application/Controller.java
	TODO		AbstractCreationTool#createFigure	AbstractCreationTool.java	tools/creation	tools/creation/AbstractCreationTool.java
	TODO		AbstractCreationTool#updateFigure	AbstractCreationTool.java	tools/creation	tools/creation/AbstractCreationTool.java
	TODO		AbstractCreationTool#cancelFigure AbstractCreationTool#terminateFigure	AbstractCreationTool.java AbstractCreationTool.java	tools/creation tools/creation	tools/creation/AbstractCreationTool.java tools/creation/AbstractCreationTool.java
	TODO	027	Drawing#initiateFigure	Drawing.java	figures	figures/Drawing.java
CursorTool	TODO		CursorTool#mouseMoved	CursorTool.java	tools	tools/CursorTool.java
	TODO	029	CursorTool#mouseExited Complete InfoPanel.fxml with all fx:ids	CursorTool.java InfoPanel.fxml	tools application/panels	tools/CursorTool.java application/panels/InfoPanel.fxml
le l	TODO		Drawing#fromShape	Drawing.java	figures	figures/Drawing.java
InfoPane	TODO	031	FocusedFigureTool#mouseEnteredTarget	FocusedFigureTool.java	tools	tools/FocusedFigureTool.java
트	TODO		InfoPanelController#fillInfoPanel Fill this panel's FXML attributes with data from figure InfoPanelController#clearInfoPanel Clears this panel's FXML attributes content	InfoPanelController.java InfoPanelController.java	application/panels application/panels	application/panels/InfoPanelController.java application/panels/InfoPanelController.java
	FXML		insert fx:id deleButton (with Icon) triggering Controller#onDeleteSelectedAction in the ToolBar	EditorFrame.fxml	application	application/EditorFrame.fxml
	FXML		insert a Menultem in #figureListView's ContextMenu triggering Controller#onDeleteSelectedAction	EditorFrame.fxml	application	application/EditorFrame.fxml
	FXML		insert fx:id moveUpButton (with Icon) triggering Controller#onMoveUpAction in the ToolBar	EditorFrame.fxml	application	application/EditorFrame.fxml
a	FXML		insert fx:id moveDownButton (with Icon) triggering Controller#onMoveDownAction in the	EditorFrame.fxml	application	application/EditorFrame.fxml
Toggle Edit Mode	FXML		ToolBar insert fx:id moveTopButton (with Icon) triggering Controller#onMoveTopAction in the ToolBar	EditorFrame.fxml	application	application/EditorFrame.fxml
Edit	FXML		insert fx:id moveBottomButton (with Icon) triggering Controller#onMoveBottomAction in the	EditorFrame.fxml	application	application/EditorFrame.fxml
90 9	I XIVIL		ToolBar	Editorrame.ixiii	аррисации	application, Editor rame.rxim
D,	FXML		insert fx:id applyStyleButton (with Icon) triggering Controller#onApplyStyleAction in the ToolBar	EditorFrame.fxml	application	application/EditorFrame.fxml
	TODO		Controller#initialize: Disable Edit mode buttons until edit mode is on:	Controller.java	application	application/Controller.java
	TODO		Controller#onEditAction: Set Tools according to selected	Controller.java	application	application/Controller.java
	TODO		Controller#onEditAction: if creation mode then turn off fitlering Controller#onEditAction: Enable / Disable edit mode buttons	Controller.java Controller.java	application application	application/Controller.java application/Controller.java
E	TODO	038	TransformTool#mousePressed initialize motion	TransformTool.java	tools	tools/TransformTool.java
Transform	TODO	039	TransformTool#mouseDragged: Apply translation on #figureRoot TransformTool#mouseDragged: Apply rotation on #figureRoot	TransformTool.java TransformTool.java	tools tools	tools/TransformTool.java tools/TransformTool.java
Ta L			Transform Tool#mousebragged. Apply Totation on #figure Root			
Tra	TODO		TransformTool#mouseDragged: Apply scale on #figureRoot	TransformTool.java	tools	tools/TransformTool.java
Tra	TODO FXML		insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar	TransformTool.java EditorFrame.fxml	tools application	tools/TransformTool.java application/EditorFrame.fxml
Tra	TODO	041	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar	TransformTool.java EditorFrame.fxml EditorFrame.fxml	tools application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml
Tra	FXML FXML TODO TODO	041 042 043	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction	TransformTool.java EditorFrame.fxml	tools application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java
	TODO FXML FXML TODO TODO TODO	041 042 043 044	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals	TransformTool.java Editofrrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java	application application application application application history	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java
Manager	FXML FXML TODO TODO	041 042 043 044 045	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java	tools application application application application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java history/HistoryManager.java
Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO	041 042 043 044 045 046 047	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#noNedoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOut HistoryManager#undo	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java	tools application application application application application history history history history	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java history/HistoryManager.java
	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#redo	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java	tools application application application application application history history history history history	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java
Manager	FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pubUndo HistoryManager#pubUndo HistoryManager#pubUndo	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java	tools application application application application application history history history history	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java
Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onNedoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#secord HistoryManager#undo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo HistoryManager#pundo	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java	tools application application application application application history	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java
Manager	FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pubUndo HistoryManager#pubUndo HistoryManager#pubUndo	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java	tools application application application application history	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java
Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pundo HistoryManager#pundo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popNedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java HistoryManager.java HistoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java	tools application application application application history application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java application/Controller.java
History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setO HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction	TransformTool.java Editoframe.fxml Editoframe.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Editoframe.fxml Controller.java	tools application application application application application history application application application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java distory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java application/Controller.java application/Controller.java
History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pundo HistoryManager#pundo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popNedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java HistoryManager.java HistoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java ListoryManager.java	tools application application application application history application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java application/Controller.java
Clear History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushRedo Controller#onClearAction triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection	TransformTool.java Editoframe.fxml Editoframe.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Editoframe.fxml Controller.java	tools application application application application application history application application application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java distory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java application/Controller.java application/Controller.java
Clear History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id redoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pundo HistoryManager#pundo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding	TransformTool.java Editoframe.fxml Editoframe.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Editoframe.fxml Controller.java	tools application application application application application history application application application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java distory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java distory/HistoryManager.java application/Controller.java application/Controller.java
Clear History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPundo HistoryManager#pushPedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) {	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java EditorFrame.fxml Controller.java Drawing.java	tools application application application application application history application application application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java
History Manager	TODO FXML FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	042 043 044 045 046 047 048 049 050 051 052	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); }	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java	tools application application application application application history application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java application/Controller.java application/EditorFrame.fxml application/Controller.java figures/Drawing.java figures/Drawing.java
Clear History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id redoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pundo HistoryManager#pundo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo HistoryManager#popUndo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i);	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java EditorFrame.fxml Controller.java Drawing.java	tools application application application application application history application application application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java
Selection Tool History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 048 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onDudoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOc HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List <figure> getSelectedFigures() utility method to retreive selected figures</figure>	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java SelectionTool.java	tools application application application application history application application application figures figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java history/DistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java
Selection Tool History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	041 042 043 044 045 046 047 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOic HistoryManager#undo HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushEdo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; I < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Drawing.java Drawing.java SelectionTool.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application history figures figures tools figures application application application application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java figures/Drawing.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java
Selection Tool History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	042 043 044 045 046 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#precord HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.IsSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List <figure> getSelectedFigures() utility method to retreive selected figures from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectionModel()</figure>	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java SelectionTool.java Drawing.java	tools application application application application application history figures figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java history/HostoryManager.java history/HostoryManager.java history/HistoryManager.java history/DistoryManager.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java
Selection Tool History Manager	TODO FXML FXML FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	042 043 044 045 046 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#puoNudo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushRedo HistoryManager#pushRedo DistroyManager#pushPopRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanaged Create a private ListsFigure> getSelectedFigures() utility method to retreive selected figures indices from figuresistView.getSelectionModel() Create a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresistView.getSelectionModel() Create a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures and creates a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures from figuresIndices from figuresistView.getSelectionModel() Create a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures a private ListsFigure> getSelectedFiguresIndices() utility method to retreive selected figures	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Drawing.java SelectionTool.java Drawing.java Controller.java Controller.java Controller.java Controller.java Controller.java Controller.java Controller.java	tools application application application application application application history application application application figures tools figures application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java fistory/DistoryManager.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java
Selection Tool History Manager	TODO FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	042 043 044 045 046 049 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#precord HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.IsSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List <figure> getSelectedFigures() utility method to retreive selected figures from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectionModel()</figure>	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Drawing.java Drawing.java SelectionTool.java Drawing.java Controller.java Drawing.java Controller.java Drawing.java	tools application application application application application application history figures figures tools figures application application application application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/Memento.java history/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java fistory/HistoryManager.java figures/Drawing.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java
Selection Tool History Manager	TODO	041 042 043 044 045 046 047 048 050 051 052 053 054	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPod HistoryManager#pushPod Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Drawing#onChanged Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures from figuresListView.getSelectionModel() Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures indices in figures.istView getSelectionModel() Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures indices in figures.istView getSelectionModel() Create a private Vister figures indices (utility method to retreive selected figures indices in figures.istView (because these indices might change when moving figures up or down) Drawing#onChanged	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java SelectionTool.java Drawing.java Controller.java Controller.java Controller.java Controller.java Controller.java	tools application application application application application application history application application application figures tools figures application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java fistory/DistoryManager.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java
Selection Tool History Manager	TODO	041 042 043 044 045 046 047 048 050 051 052 053 054 055 056 056	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onDundoAction Memento#equals Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOi HistoryManager#undo HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushEdo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Figure figure = get(i); updateSelection (i, figure.isSelected(i); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> petSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> petSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> petSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private List-Integer> petSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelection model() Create a private List-Integer> petSelectedFiguresIndices(TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Drawing.java Controller.java Drawing.java Controller.java Controller.java Controller.java Controller.java Controller.java Controller.java Controller.java Controller.java	tools application application application application application application history application application application figures tools figures application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java
Selection Tool History Manager	TODO TODO	041 042 043 044 045 046 047 050 050 053 054 055 056 056 057 056 057 059	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo FistoryManager#pushRedo Frawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures from figuresListView.getSelectionModel() Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private List-Figure> getSelectedFiguresIndices() utility method to retreive selected figures indices in figures utility method to retreive selected figures indices in figures utility method to retreive selected figures indices in figures utility method to retreive selected figures indices in figures utility method to retreive se	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java SelectionTool.java Drawing.java Controller.java	tools application application application application application application history application figures application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java
Selection Tool History Manager	TODO	041 042 043 044 045 046 047 050 051 052 053 054 055 056 057 058 059 060	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onDundoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Prawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; I < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures from figuresListView_getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView_getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresUndoChanged Controller#onDeleteSelectedAction Controller#onDeleteSelectedAction Controller#onMoveD	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java Drawing.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application application history application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
Selection Tool History Manager	TODO	041 042 043 044 045 048 049 050 051 052 053 054 055 056 057 058 059 060 060 060	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#bonUndoAction Controller#bonBedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#bushundo HistoryManager#pushundo HistoryManager#pushundo HistoryManager#pushundo HistoryManager#popNedo HistoryManager#popNedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.isze(); i++) { Figure figure = get(i); yudateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List <figure> getSelectedFigures() utility method to retreive selected figures from figuresListView.getSelectionModel() Create a private List<figure> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFigures indices() utility method to retreive selected figures indices from figuresListView.getSelectedFigures indices() utility method to retreive selected figures indices from figuresListView.getSelectedFigures indices() utility method to retreive selected figures indices in figuresListView.getSelectedFigures indices() utility method to ret</figure></figure>	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application history application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
Clear History Manager	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 058 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onDudoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOi HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushVado HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures from figuresIstview_getSelectionModel() Create a private List-integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresIstview_getSelectofingures(indices() utility method to retreive selected figures indices from figuresUstview getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresUstview getSelectedFiguresindices() utility method to retreive selected figures indices from figuresUstview getSelectofines(instinteger> indices) to update selected figures indices in figuresUstview decause theses indices might change when moving figures up or down) Drawing#lonChanged Controller#onDeleteSelectedAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Drawing#delearSelection Drawing#delearSelection Drawing#delearSelection Drawing#delearSelection Drawi	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java HistoryManager.java HistoryManager.java HistoryManager.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application history application figures	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java figures/Drawing.java
Controller's actions depending on Selection Selection Tool Beautiful Selection Tool	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 058 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onDundoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.isze(); i++) { Figure figure = get(i);	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application history application	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Controller's actions depending on Selection Selection Tool Beautiful Selection Tool	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 058 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onDudoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOi HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushVado HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures from figuresIstview_getSelectionModel() Create a private List-integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresIstview_getSelectofingures(indices() utility method to retreive selected figures indices from figuresUstview getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresUstview getSelectedFiguresindices() utility method to retreive selected figures indices from figuresUstview getSelectofines(instinteger> indices) to update selected figures indices in figuresUstview decause theses indices might change when moving figures up or down) Drawing#lonChanged Controller#onDeleteSelectedAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Controller#onMoveDownAction Drawing#delearSelection Drawing#delearSelection Drawing#delearSelection Drawing#delearSelection Drawi	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java HistoryManager.java HistoryManager.java HistoryManager.java Drawing.java Controller.java Drawing.java Controller.java	tools application application application application application application history application figures	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java figures/Drawing.java
Controller's actions depending on Selection Selection Tool Beautiful Selection Tool	TODO FXML TODO TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 058 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.issel(); i++) { Figure figure = get(i);	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java SelectionTool.java Drawing.java Controller.java FigureTypeCell.fxml	tools application application application application application application history application application application application application application application application application	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
Controller's actions depending on Selection Selection Tool Beautiful Selection Tool	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 056 057 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onDundoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushPundo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#updateSelection Drawing#updateSelection Drawing#updateSelection in ToolBar Controller#onClearAction Drawing#updateSelection Create a private List<-Figure selectedFigures() utility method to retreive selected figures from figuresListView.getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices in figuresListView (because theses indices might change when moving figures up or down) Drawing#onChanged Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveDpaction Controller#onMoveBottomAction Create a new FigureTypeCell.java by customizing existing FigureCell.fxml Create a new FigureTypeCell.fxml by customizing existing FigureCell.fxml	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fontroller.java	tools application application application application application application history application application/cells application/cells	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java fistory/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/cells/FigureTypeCell.fxml application/cells/FigureTypeCellController.java
igureTypeCell for Controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 056 057 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#popRedo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.issel(); i++) { Figure figure = get(i);	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Controller.java Drawing.java Controller.java Controller.java Drawing.java Controller.java	tools application application application application application application application history application figures application application application figures application application application figures	tools/TransformTool.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
igureTypeCell for Controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 056 057 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Memento#equals ##istoryManager#setSize: Trim #undoStack & #redoStack if required ##istoryManager#setSize: Trim #undoStack & #redoStack if required ##istoryManager#record ##istoryManager#ger#od ##istoryManager#pushUndo ##istoryManager#pushUndo ##istoryManager#pushUndo ##istoryManager#pushUndo ##istoryManager#pushUndo ##istoryManager#pushEdo ##istoryManager#popRedo insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#setMemeto: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) {	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml LineTypeCell.fxml	tools application application application application application application history application application application application application application application application application application application application application application application application application application application application application application application application application application/cells application/cells application application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
igureTypeCell for Controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 056 057 059 060 061 062 063	insert fx:id undoButton (with Icon) triggering onUndoAction in the ToolBar insert fx:id redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Controller#onDundoAction Controller#onRedoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#setOize. HistoryManager#undo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushEdo HistoryManager#popRedo Insert fx:id clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#updateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; I < figures.isze(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#onChanged Create a private List-Figure> getSelectedFigures() utility method to retreive selected figures from figuresListView.getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private List-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private Dist-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private Dist-Integer> getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectionFiguresIndices() utility method to retreive selected figures from figuresListVi	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java Drawing.java Untroller.java Drawing.java Drawing.java Controller.java Drawing.java Controller.java FigureTypeCell.java FigureTypeCell.java FigureTypeCell.java FigureTypeCell.java FigureTypeCell.java	tools application application application application application application history application application/cells application/cells application/cells application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
igureTypeCell for Controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 056 057 060 061 062 063 064	insert fxid redoButton (with Icon) triggering onUndoAction in the ToolBar insert fxid redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Memento#equals ##istoryManager#setSize: Trim #undoStack & #redoStack if required ##istoryManager#record ##istoryManager#pustundo ##istoryManager#p	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.java FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml	tools application application application application application application history application application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java
igureTypeCell for Controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 055 056 057 056 057 060 061 062 063 064	insert fixid redoButton (with Icon) triggering onUndoAction in the ToolBar insert fixid redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo Figure fixid clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#undateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#ionChanged Create a private List <firepres (because="" .<="" change="" controller#onmovedownaction="" down)="" drawing#donchanged="" figures="" figures.istview.getselectedfiguresindices()="" figures.litylew="" from="" getselectedfigures()="" indices="" method="" might="" moving="" or="" retreive="" selected="" td="" these="" to="" up="" utility="" when=""><td>TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Controller.java FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml LineTypeCell.fxml</td><td>tools application application application application application application history application application application application application figures application application figures application application figures application application/cells application/cells application/cells application/cells</td><td>tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/Controller.java</td></firepres>	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Controller.java FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml	tools application application application application application application history application application application application application figures application application figures application application figures application application/cells application/cells application/cells application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java
LineTypeColl for FigureTypeComboBo shapeTypeComboBo shapeTypeComboBo shapeTypeComboBo shapeTypeComboBo shapeTypeComboBox	TODO	041 042 043 044 045 046 050 051 052 053 054 056 056 056 067 068 068 069 060 061 062 063 064	insert fixid redoButton (with Icon) triggering onUndoAction in the ToolBar Insert fixid redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Mementoflequals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#grundo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo Fire trial clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#undateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i): updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#ionChanged Create a private List <integer> getSelectedFigures() utility method to retreive selected figures indices from figuresListView.getSelectionModel() Create a private void setSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelectedFiguresIndices() utility method to retreive selected figures indices from figuresListView.getSelection Controller#onMoveDownAction Controller#onMoveDownBow.setButtonC</integer>	TransformTool.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.fxml FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml LineTypeCell.fxml LineTypeCell.fxml LineTypeCell.fornl LineTypeCell.fornl LineTypeCell.fxml LineTypeCell.fornl	tools application application application application application application history application application application application application application figures application application application application application application application application application application application application application application application application application application application application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java application/Cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/Controller.java application/Controller.java application/Controller.java application/Cells/FigureTypeCell.fxml application/Cells/FigureTypeCell.fxml application/Cells/FigureTypeCell.fxml application/Cells/FigureTypeCell.fxml application/Cells/LineTypeCell.fxml application/Controller.java application/Centroller.java application/Centroller.java application/Centroller.java
ig.uer/ppecell for shaper of controller's actions depending on Selection Tool Box	TODO	041 042 043 044 045 046 050 051 052 053 054 056 056 056 067 068 068 069 060 061 062 063 064	insert fixid redoButton (with Icon) triggering onUndoAction in the ToolBar insert fixid redoButton (with Icon) triggering onRedoAction in the ToolBar Controller#onUndoAction Memento#equals HistoryManager#setSize: Trim #undoStack & #redoStack if required HistoryManager#record HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushUndo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo HistoryManager#pushRedo Figure fixid clearButton triggering Controller#onClearAction in ToolBar Controller#onClearAction Drawing#undateSelection Drawing#setMemento: Refresh Selection when a Memento is set in Drawing by adding for (int i = 0; i < figures.size(); i++) { Figure figure = get(i); updateSelection(i, figure.isSelected()); } SelectionTool#mouseClicked Drawing#ionChanged Create a private List <firepres (because="" .<="" change="" controller#onmovedownaction="" down)="" drawing#donchanged="" figures="" figures.istview.getselectedfiguresindices()="" figures.litylew="" from="" getselectedfigures()="" indices="" method="" might="" moving="" or="" retreive="" selected="" td="" these="" to="" up="" utility="" when=""><td>TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.java FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml LineTypeCell.fxml</td><td>tools application application application application application application history application application/cells application/cells</td><td>tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java</td></firepres>	TransformTool.java EditorFrame.fxml Editofrrame.fxml Controller.java Controller.java Memento.java HistoryManager.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java Controller.java FigureTypeCell.java FigureTypeCell.fxml FigureTypeCell.fxml LineTypeCell.fxml	tools application application application application application application history application application/cells	tools/TransformTool.java application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java history/HistoryManager.java application/Controller.java application/Controller.java figures/Drawing.java tools/SelectionTool.java figures/Drawing.java application/Controller.java application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/FigureTypeCell.fxml application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java application/cells/LineTypeCell.java

Part	Туре	#	Description	Resource	Path	File
useFil s useEc	TODO	068	Controller#onCheckColorsConsistencyAction: edge color case	Controller.java	application	application/Controller.java
					**	
Set History Size	TODO		Read SizeDialogPane documentation Controller#onSetHistorySizeAction pop a SizeDialogPane and set #historyManager's size	SizeDialogPane.java Controller.java	application/panels application	application/panels/SizeDialogPane.java application/Controller.java
	TODO		Controller#onSetLoggerLevelUpToInfoAction Level.INFO	Controller.java	application	application/Controller.java
Set Logger Levels	TODO	071	Controller#onSetLoggerLevelUpToWarningAction Level.WARNING	Controller.java	application	application/Controller.java
	TODO		Controller#onSetLoggerLevelUpToSevereAction Level.SEVERE Controller#onSetLoggerLevelOffAction Level.OFF	Controller.java Controller.java	application application	application/Controller.java application/Controller.java
	TODO		Controller#onDisplayButtonsWithGraphicsOnlyAction	Controller.java	application	application/Controller.java
Set Buttons Display	TODO		Controller#onDisplayButtonsWithTextAndGraphicsAction	Controller.java	application	application/Controller.java
8.0	TODO NEW	076	Controller#onDisplayButtonsWithTextOnlyAction Create class Ellipse	Controller.java Ellipse.java	application figures	application/Controller.java figures/Ellipse.java
Rounded Rectangle	TODO		FigureType#getFigure ELLIPSE case	FigureType.java	figures/enums	figures/enums/FigureType.java
	TODO		FigureType#fromFigure add all other figures types Controller#onShapeChangedAction	FigureType.java Controller.java	figures/enums application	figures/enums/FigureType.java application/Controller.java
	NEW		Create class Rectangle	Rectangle.java	figures	figures/Rectangle.java
	TODO		FigureType#getFigure RECTANGLE case FigureType#fromFigure add all other figures types	FigureType.java FigureType.java	figures/enums figures/enums	figures/enums/FigureType.java figures/enums/FigureType.java
	NEW	0,0	Create class RoundedRectangle	RoundedRectangle.java	figures	figures/RoundedRectangle.java
	NEW		Create class RoundedRectangularShapeCreationTool to manage RoundedRectangle drawing with 3 steps (instead of 2 with RectangularShapeCreationTool) 1. Initiate Figure 2. Drag to set Size and release 3. Do something to set Arc size	RoundedRectangularShapeCreationTool.java	tools/creation	tools/creation/RoundedRectangularShapeCreationTool.java
onno	TODO	081	FigureType#getFigure ROUNDED_RECTANGLE case	FigureType.java	figures/enums	figures/enums/FigureType.java
~	TODO		FigureType#getCreationTool ROUNDED_RECTANGLE case	FigureType.java	figures/enums	figures/enums/FigureType.java
	TODO	0/8	FigureType#fromFigure add all other figures types Create class Polygon	FigureType.java Polygon.java	figures/enums figures	figures/enums/FigureType.java figures/Polygon.java
Polygon	NEW		Create class PolygonCreationTool to manage Polygon drawing with 2 steps 1. Click to initiate an new Polygon 2. Click to add a point to the Polygon or Right click to terminate mouseMoved moves the last added point	PolygonCreationTool.java	tools/creation	tools/creation/PolygonCreationTool.java
	TODO		FigureType#getFigure POLYGON case FigureType#getCreationTool POLYGON case	FigureType.java FigureType.java	figures/enums figures/enums	figures/enums/FigureType.java figures/enums/FigureType.java
	TODO		FigureType#fromFigure add all other figures types	FigureType.java	figures/enums	figures/enums/FigureType.java
NGon	NEW		Create class NGon Create class RegularPolygonCreationTool to manage NGon drawing with 3 steps 1. Press to initiate Figure 2. Drag to set NGon's radius and release 3. Scroll to set number of sides and click to terminate	NGon.java RegularPolygonCreationTool.java	figures tools/creation	figures/NGon.java tools/creation/RegularPolygonCreationTool.java
	TODO		FigureType#getCreationTool NGON case FigureType#getCreationTool NGON case	FigureType.java FigureType.java	figures/enums figures/enums	figures/enums/FigureType.java figures/enums/FigureType.java
	TODO		FigureType#fromFigure add all other figures types	FigureType.java	figures/enums	figures/enums/FigureType.java
	NEW		Create class Star	Star.java Star.java	figure	figure/Star.java
Star	NEW		Create class StarCreationTool to manage Star drawing with 4 steps 1. Press to initiate Figure 2. Drag and release to set Star's outer radius 3. Drag and release to set Star's inner radius or Right Click to cancel inner radius step 4. Scroll to set number of branches and click to terminate	StarCreationTool.java	tools/creation	tools/creation/StarCreationTool.java
	TODO		FigureType#getCreationTool STAR case FigureType#getCreationTool STAR case	FigureType.java FigureType.java	figures/enums figures/enums	figures/enums/FigureType.java figures/enums/FigureType.java
	TODO					
	TODO TODO		FigureType#fromFigure add all other figures types	FigureType.java	figures/enums	figures/enums/FigureType.java
			FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge</color>			
	TODO NEW	078	FigureType#fromFigure add all other figures types	FigureType.java EdgeColorFilter.java	figures/enums figures/filters	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java
	TODO	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color</color></color>	FigureType.java	figures/enums	figures/enums/FigureType.java
	TODO NEW	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type</linetype></color></color>	FigureType.java EdgeColorFilter.java	figures/enums figures/filters	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java
	NEW NEW	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<couble> to filter figures based on their</couble></linetype></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java	figures/enums figures/filters figures/filters figures/filters	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java
	NEW NEW NEW	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller;). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event)</double></linetype></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java	figures/enums figures/filters figures/filters figures/filters	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java
	NEW NEW NEW	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in ControllerJ. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangeAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in ControllerJ. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event)</double></linetype></linetype></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java	figures/enums figures/filters figures/filters figures/filters figures/filters	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java
	NEW NEW NEW FXML	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill ine type Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller/DeliterChangeAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangeAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangeAction(ActionEvent event)</double></linetype></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml	figures/enums figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml
	NEW NEW NEW FXML	078	FigureType#fromFigure add all other figures types Create new class EdgecolorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#infiltalize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added</double></double></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml	figures/enums figures/filters figures/filters figures/filters figures/filters application application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml
	NEW NEW NEW FXML FXML TODO	078	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill ine type Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangeAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem should have "On Action" registered to Controller#nonFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem needs to be processed AFTER TODO 019 because</double></linetype></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml	figures/enums figures/filters figures/filters figures/filters figures/filters application application application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml
	NEW NEW NEW FXML FXML TODO	078	FigureType#fromFigure add all other figures types Create new class EdgecolorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#infiltalize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added</double></double></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application application application application application application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java
80	NEW NEW NEW FXML FXML TODO	078 089 089b	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<clor> to filter figures based on their fill expe Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangeAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterGuelCorCheckMenultem in Controller,). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem needs to be processed AFTER TODO 019 because #lineWidthSpinner needs to be set up first Add a #filterSyelectedMenultem Menultem FXML attributes corresponding to the Menultem in ContextMenu of #listView below</double></linetype></clor></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters application application application application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml
iltering	NEW NEW NEW FXML FXML TODO	078 089 089b	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill ine type Create new class LineWidthFilter extends FigureFilter<color> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of fine type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#intialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem needs to be processed AFTER TODO 019 because #filterLineWidthCheckMenultem meeds to be processed AFTER TODO 019 because #filterLineWidthCheckMenultem meeds to be set up first Add a #filterIneWidthCheckMenultem Menultem FXML attributes corresponding to the Menultem in</double></color></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application application application application application application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO	078 089 089b	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter<linetype> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller.] Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultems and #filterEdgeColorCheckMenultem in Controller.). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem needs to be processed AFTER TODO 019 because #lineWidthSpinner needs to be set up first Add a #filterSyelectedMenultem Menultem FXML attributes corresponding to the Menultem in ContextMenu of #listView below Add a fx:id filterBySelected Menultem in #figuresFilterSistView's</double></linetype></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO FXML TODO	089 089b 090	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter<color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter<color> to filter figures based on their fill ine type Create new class LineWidthFilter extends FigureFilter<color> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter<double> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems and #filterLineTypeCheckMenultem in Controller. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Cortoller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem needs to be processed AFTER TODO 019 becaus</double></color></color></color></color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java EditorFrame.fxml	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/EditorFrame.fxml
Filtering	NEW NEW FXML FXML TODO TODO FXML	089 089 090	FigureType#fromFigure add all other figures types Create new class EdgecolorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter LineType> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterCaleColorCheckMenultem in Controller;). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultem according to selected states of the CheckMenultem in expanding the filter in the filter in the filter in t</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Controller.java EditorFrame.fxml	figures/enums figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java gplication/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java
Filtering	NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO	089 089b 090 091 092	FigureType#fromFigure add all other figures types Create new class EdgecolorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter To filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter LineType> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter LineType> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterGelColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems width (see #filterLineTypeCheckMenultem and #filterIneTypeCheckMenultem meeds to be processed AFTER TODO 019 because #filterLineWidthCheckMenultem meeds to be processed AFTER TODO 019 because #filterIneWidthCheckMenultem meeds to be processed AFTER TODO 019 because #filterIneWidthCheckMenultem meeds to be processed AFTER TODO 019 because #filterInePySelectedMenultem</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application figures figures	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java
Filtering	NEW NEW FXML FXML TODO TODO TODO TODO TODO	089 089 090 091	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter Tolar figures based on their fill color Create new class LineTypeFilter extends FigureFilter FigureFilter FigureFilter Figures based on their fill fill fill fill fill fill fill fi</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java EditorFrame.fxml Controller.java Controller.java Controller.java Dawing.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application figures	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java
Filtering	NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO	089 089b 090 091 092	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter In the figures based on their fill color Create new class LineTypeFilter extends FigureFilter In the figures based on their fill color Create new class LineTypeFilter extends FigureFilter In the figures based on their line type Create new class LineTypeFilter extends FigureFilter In the figures based on their line type Create new class LineWidthFilter extends FigureFilter In the figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of film type and line width (see #filterLineTypeCheckMenultem in Guttorler. Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineTypeCheckMenultem neds to be processed AFTER TODO 019 because #lineWidthCheckMenultem neds to be processed AFTER TODO 019 because #lineWidthSpinner needs to be set up first Add a #filterEdySelectedMenultem Menultem FXML attributes corresponding to the Menultem in Controller#onerHilterChangedAction in the Controller Controller#initia</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Controller.java EditorFrame.fxml Controller.java Drawing.java Drawing.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application figures figures	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO	089 089b 090 091 092 093	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter toliter figures based on their fill color Create new class LineTypeFilter extends FigureFilter to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter LineType> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultem and #filterIneTypeCheckMenultem meeds to be processed AFTER TODO 019 because #filterLineWi</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java EditorFrame.fxml Controller.java Controller.java Controller.java Controller.java Drawing.java Controller.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application application application application application application application application application figures figures application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	089 089 090 091 092 093 094	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of film type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller.). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initalize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineWidthCheckMenultem meeds to be processed AFTER TODO 019 because #filterLineWidthCheckMenultem Menultem FXML attributes corresponding to the Menultem in Controller#initalize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenulter Menultem Menultem FXML attributes corresponding to the Men</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Controller.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089b 090 091 092 093 094 095 096	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter To filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter LineType> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter Loud Filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Unes" to menu "Filtering" containing CheckMenultems indicating the filtering state of film type and line width (see #filterLineTypeCheckMenultem indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineTypeCheckMenultem in Controller*Controller*Ontroller*Controller*DilterMenultem Nenultem FXML</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML FXML TODO TODO TODO TODO TODO OPT TODO DONE TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089 090 091 092 093 094 095 096 097 098	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter Tolor Create new class LineTypeFilter extends FigureFilter FigureFilter Create new class LineWidthFilter extends FigureFilter Foliter figures based on their line type Create new class LineWidthFilter extends FigureFilter Foliter figures based on their line width Mad sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterGriclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterGreater to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems whould have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction flaterOngetPilter according to selected states of the CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction in the Controller fla</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089b 090 091 092 093 094 095 096 097 098 099	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter To filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter LineType> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter Loud Filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller*, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller*. Each of these CheckMenultem should have "On Action" registered to Controller*MonFilterChangedAction(ActionEvent event) Add sub-menu "Unes" to menu "Filtering" containing CheckMenultems indicating the filtering state of film type and line width (see #filterLineTypeCheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem should have "On Action" registered to Controller*MonFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultem in the Controller*</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML FXML TODO TODO TODO TODO TODO OPT TODO DONE TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089 090 091 092 093 094 095 096 097 098 099 100	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter <color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter Create new class FillColorFilter extends FigureFilter Tolor Create new class LineTypeFilter extends FigureFilter FigureFilter Create new class LineWidthFilter extends FigureFilter Foliter figures based on their line type Create new class LineWidthFilter extends FigureFilter Foliter figures based on their line width Mad sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterGriclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterGreater to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems whould have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultem and #filterLineTypeCheckMenultem in Controller#onFilterChangedAction flaterOngetPilter according to selected states of the CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction in the Controller fla</color>	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089b 090 091 092 093 094 095 096 097 099 099 100 101 102	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter-Color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter-Color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter-Color> to filter figures based on their filme type Create new class LineTypeFilter extends FigureFilter-Couble> to filter figures based on their filme vight Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCiclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller/BonFilterChangedAction[ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller/BonFilterChangedAction[ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultems should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineVybeCheckMenultem meds to be processed AFTER TODO 019 because #filterineWidthCheckMenultem meds to be processed AFTER TODO 019 because #filterineWidthCheckMenultem meds to be set up first Add a #filterBySelectedMenultem Menultem FXML attributes corresponding to the Menultem in Controller#onFilterChangedAction in the Controller Controller#onFilterChangedAction in the Controller Controller#onFilterChangedAc	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089b 090 091 092 093 094 095 096 097 099 100 101 102 103	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter-Color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter-Color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter-Color> to filter figures based on their line type Create new class LineTypeFilter extends FigureFilter-Couble> to filter figures based on their line type Create new class LineWidthFilter extends FigureFilter-Couble> to filter figures based on their line width Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCirclesCheckMenultem #filterStarsCheckMenultem in Controller, Each of these CheckMenultem should have "On Action" registered to Controller, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction[ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction[ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller, Each of these CheckMenultem should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineTypeCheckMenultem meeds to be processed AFTER TODO 019 because #lineWidthSpinner needs to be set up first Add a #filterSpSelectedMenultem hemotypesFilter (see the CheckMenultem and #filteringProperty propenties so that when one chan	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java
Filtering	NEW NEW NEW FXML FXML TODO TODO TODO TODO TODO TODO TODO TOD	078 089 089b 090 091 092 093 094 095 096 097 099 099 100 101 102	FigureType#fromFigure add all other figures types Create new class EdgeColorFilter extends FigureFilter-Color> to filter figures based on their edge color (see existing FigureTypeFilter class for inspiration) Create new class FillColorFilter extends FigureFilter-Color> to filter figures based on their fill color Create new class LineTypeFilter extends FigureFilter-Color> to filter figures based on their filme type Create new class LineTypeFilter extends FigureFilter-Couble> to filter figures based on their filme vight Add sub-menu "Shapes" to menu "Filtering" containing CheckMenultems indicating the filtering state of each figure type (see #filterCiclesCheckMenultem #filterStarsCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller/BonFilterChangedAction[ActionEvent event) Add sub-menu "Colors" to menu "Filtering" containing CheckMenultems indicating the filtering state of fill and edge colors (see #filterFillColorCheckMenultem and #filterEdgeColorCheckMenultem in Controller). Each of these CheckMenultem should have "On Action" registered to Controller/BonFilterChangedAction[ActionEvent event) Add sub-menu "Lines" to menu "Filtering" containing CheckMenultems indicating the filtering state of line type and line width (see #filterLineTypeCheckMenultem and #filterLineTypeCheckMenultem in Controller). Each of these CheckMenultems should have "On Action" registered to Controller#onFilterChangedAction(ActionEvent event) Controller#initialize Setup #figureTypesFilter and #figuresFilter according to selected states of the CheckMenultems we just added #filterLineVybeCheckMenultem meds to be processed AFTER TODO 019 because #filterineWidthCheckMenultem meds to be processed AFTER TODO 019 because #filterineWidthCheckMenultem meds to be set up first Add a #filterBySelectedMenultem Menultem FXML attributes corresponding to the Menultem in Controller#onFilterChangedAction in the Controller Controller#onFilterChangedAction in the Controller Controller#onFilterChangedAc	FigureType.java EdgeColorFilter.java FillColorFilter.java LineTypeFilter.java LineWidthFilter.java EditorFrame.fxml EditorFrame.fxml EditorFrame.fxml Controller.java Controller.java Drawing.java Drawing.java Controller.java	figures/enums figures/filters figures/filters figures/filters figures/filters figures/filters figures/filters application	figures/enums/FigureType.java figures/filters/EdgeColorFilter.java figures/filters/FillColorFilter.java figures/filters/LineTypeFilter.java figures/filters/LineTypeFilter.java figures/filters/LineWidthFilter.java application/EditorFrame.fxml application/EditorFrame.fxml application/EditorFrame.fxml application/Controller.java application/Controller.java application/Controller.java figures/Drawing.java figures/Drawing.java application/Controller.java