## FileHandler fileHandler; void penWidthChanged(int width); void changePenColor(); void changeFillColor(); void changeFillColor(); void changeFillProperty(); void saveState(); void dragEnterEvent(QDragEnterEvent\*); void dropEvent(QDropEvent\*); void openFile(); void saveFile(); void closeFile() CopyPaste PaintCanvas QTextEdit \*textEdit; signals: FileManager void imageChanged(); bool fill; QFile file; void copy(); bool drawing; int penWidth; void cut(); void paste(); bool openFile(const QString &fileName); bool saveFile(@paramt); QString readFile(); void drawLineTo(const QPoint &endPoint); void drawRectTo(const QPoint &endPoint, bool ellipse = false); void eraseUnder(const QPoint &topLeft); bool writeFile(const QString &content); QString getOpenFileName(@param); QString getSaveFileName(@param); void resizeImage(QImage \*image, const QSize &newSize); void saveStateForUndo(); DragButton RedoUndo QStack<QImage> undoStack; QStack<QImage> redoStack; QPoint lastPos: QColor buttonColor; QColor getButtonColor() const; void setButtonColor(const QColor &value); void addState(const QImage &state); QImage undo(); QImage redo(); bool canUndo() const; bool canRedo() const;

Mainwindow

Ui::MainWindow \*ui;

QTextEdit \*textEdit; CopyPaste \*copyPaste; RedoUndo \*redoUndo;

On. Mallivillood ut,
PaintCanvas \* canvas;
QPushButton \* penColorButton;
QPushButton \* fillColorButton;
QCheckBox \* fillCheckBox;