

Dokumentation zu dem PRG_Projekt

Übersicht

1. Beschreibung der Aufgaben der Software
2. Installationsanleitung
3. Bedienungsanleitung
4. Systemarchitektur
5. Beschreibung technischer Teilaspekte
6. Fazit

1. Beschreibung der Aufgaben der Software

Bei der Software handelt es sich um ein Bildbearbeitungsprogramm geschrieben in Java.

Das Programm sollte Bilder öffnen und speichern, sowie einen Filter drüber legen können. Außerdem sollte es eine Paint-Funktion ermöglichen.

Wir nennen es CnC.Painto und es sind uns einige Funktionen gelungen, jedoch ist es noch nicht ganz fertig. Mit Hilfe eines JFileChoosers ist es uns gelungen, dass man mit dem Programm in der Ordnerstruktur des Rechners nach Dateien suchen kann. Jedoch lassen sich die Bilder noch nicht öffnen.

Eine fertige und ausführbare ist die Paint-Funktion, in der wir uns über den JColorChooser eine Farbe aussuchen und über einen JSlider die Strichdicke von 1 bis 20 einstellen können.

2. Installationsanleitung

Man doppelklickt mit der linken maustaste auf die Ausführungs-Datei "CnC-Painto.exe". Anschließend öffnet sich unser Paint Java-Programm und man kann direkt mit dem malen/zeichnen oder dem Bild bearbeiten loslegen.

3. Bedienungsanleitung

Die Bedienung des Programms ist ziemlich überschaubar und dadurch relativ selbsterklärend.

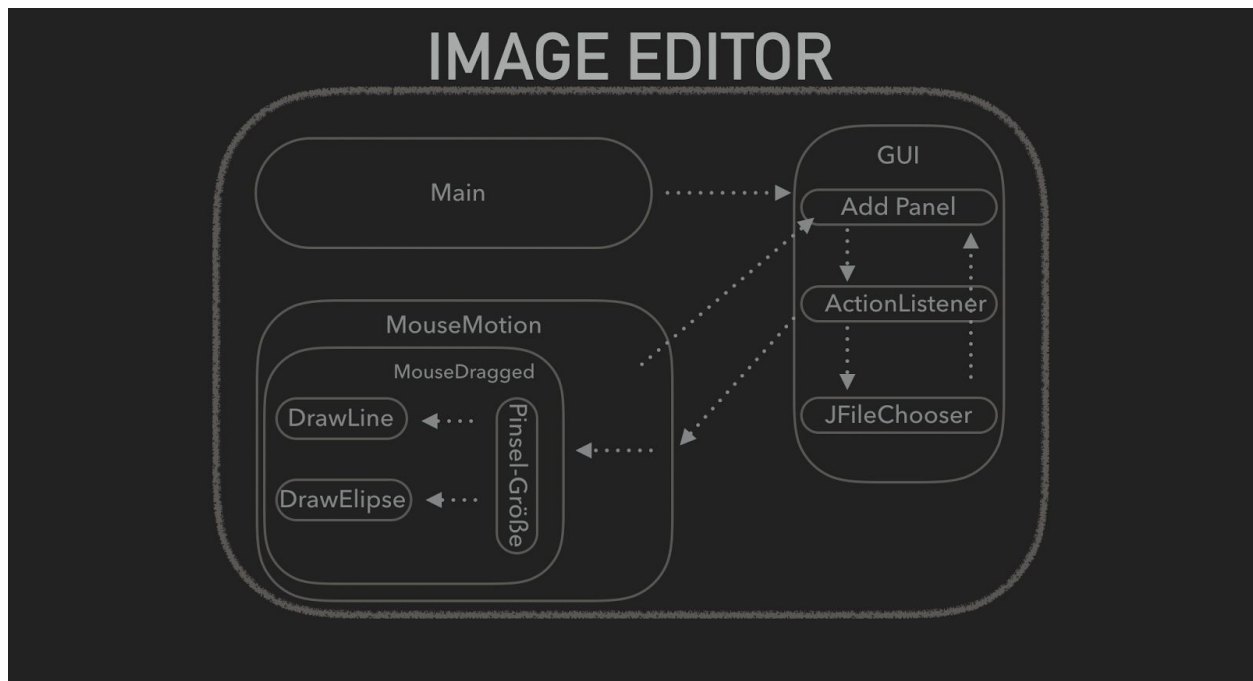
Es gibt oben links eine Menüleiste mit einem drop-down-Menü, die wir als "File" bezeichnen und klickt man darauf öffnet sich ein Dropdown-Menü mit zwei weiteren Feldern. Durch "Open" öffnet sich ein neues Fenster, das uns in die Ordnerstruktur des Rechners einblick gewährt und uns Dateien auswählen lässt. Da sich noch keine Bilder öffnen lassen, hat das "Save" Feld bis auf weiteres keine Funktion.

Desweiteren haben wir noch zwei JButtons, über den oberen "Choose Color" öffnet sich ein neues Fenster, was uns sagt, dass wir eine Farbe auswählen sollen. Nachdem wir uns für eine Farbe entschieden haben, können wir auf "OK" klicken, den Vorgang abbrechen oder aber zur zuletzt genutzten Farbe "Zurücksetzen" und mit dieser Farbe etwas auf den weißen Hintergrund malen.

Mit dem sich darunter befindenden “Choose Filter” Button sollte es möglich sein zum Beispiel einen Graufilter über das Bild zu legen, leider konnten wir dies noch nicht testen.

Unter dem “Choose Filter” Button findet man eine Leiste mit der man die “Stift-Dicke” einstellen kann um je nach verlangen dünner oder dicker in der entsprechenden Farbe zu malen.

4. Systemarchitektur



5. Beschreibung der technischen Teilaspekte:

Als Grundlage dient uns eine Reihe aus vorgefertigten Funktionen von Java zur Erstellung der GUI, so wie das Erkennen von Veränderungen.

Die Mal-Komponente basiert auf einer Reihe von `MausMotionListenern` die bei jeder Bewegung einer gedrückten Maus die X und Y Koordinaten ausgeben. Dann malen wir mit der Funktion `drawline` von einem Punkt $A(ax,ay)$ zu dem nächsten Punkt $B(bx,by)$. A ist dabei der Anfang der Linie und B das Ende. Bei jedem Aufrufen der Funktion wird zunächst A auf den letzten aktuellen Wert gesetzt und B bekommt kurz danach den aktuelleren Wert. So malen wir unsere Linien.

Da wir zum Malen ein `Graphics` Objekt benutzen, haben wir noch ein Problem, da sich dieses an der insgesamt Fenstergröße orientiert, wohingegen die Koordinaten von dem `JPanel` kommen. Was zur Folge hat, dass wir einen Offset einbauen mussten damit der Cursor nicht abseits von dem gemalten Pixel ist.

Dieser Offset wird zur Laufzeit der Methode `drawline` auf die einzelnen Werte addiert. Um zu garantieren dass der Nutzer auch anständig malen kann.

Neben dem eigentlichen Malen benutzen wir noch die Funktion `setStroke` um die Pinselbreite zu bestimmen, da es diese Funktion aber nur in `Graphics2D` gibt, casten wir `Graphics` zu `Graphics2D` und benutzen das um später weiter zu malen. Die Größe wird dann noch über einen Slider bestimmt mit dem man einen Integer Wert verändert. Da die Methode `drawline` nur Viereckige Linien malt wird mit zunehmender Größe die Linie immer kantiger. Darum benutzen wir bei einer Pinselbreite über 10 `drawEllipse` um immer wieder kleine Kreise zu malen um gerade Linien ausgeben zu können.

Zu guter Letzt können wir noch die Farbe des Pinsel ändern, dazu benutzen wir `JColorChooser`, eine in Java Swing vorgefertigte Funktion um eine Farbpalette aufzurufen die dem Nutzer die Möglichkeit gibt frei zu wählen welche Farbe er möchte.

6. Fazit

Im Großen und Ganzen steht das Grundkonzept des Programmes. Wir hatten einen steinigen Weg vor uns, den wir unserer meinung nach gut gemeistert haben da uns allerlei kenntnisse in Java noch fehlen und wir uns durchlesen mussten. Sourceforge stand uns zur seite da viele neulinge und auch Fortgeschrittene viele unserer fragen schon gestellt haben. Als nächster sinnvoller Schritt würden wir kleinere Kinderkrankheiten und unsaubere Code Passagen bereinigen. Dadurch würde letzten endes nicht nur die wartbarkeit, sondern unter umständen auch die Leistung verbessert werden. Außerdem würden wir gerne noch mehr verschiedene Filter und eventuell transformationen hinzufügen.