

Idee: ImageDestroyer

Mittels verschiedenen Tools, die in einem kleinen, dynamisch generierten Menü ausgewählt werden können, kann man ein ausgewähltes Bild auf verschiedenste Art und Weise „zerstören“. Vorerst sind als Tools geplant: Hammer (Dellen), Winkelschleifer (kreisförmige Verwischungen), Schwamm (Verwischungen entlang der Mausspur), Giesskanne (Verschwimmen der einzelnen Pixel, evtl. Glanzeffekt), Spray (nebelartige Verfärbung), Lupe (dunkle Verbrennungen).

Vorgehen

Da wir an der Cursorposition das Bild des jeweils aktivierten Tools anzeigen möchten, können wir das zu verändernde Bild nicht einfach als Hintergrundbild anzeigen und kontinuierlich verändern. Stattdessen müssen wir sämtliche Pixel des Bildes speichern und für jedes angezeigte Frame zuerst aus dem Speicher ausgeben, bevor wir das Tool an der Cursorposition zeichnen. Bei einem Mausklick müssen wir ebenfalls alle Pixel aus dem Speicher ausgeben (das Bild des Tools ist dann verschwunden), die entsprechenden Änderungen am Bild vornehmen und anschliessend wieder sämtliche Pixel speichern damit sie beim nächsten Durchlauf des Loops wieder ausgegeben werden können um das angezeigte Tool verschwinden zu lassen usw..

Alternativ könnte man für jedes Frame zuerst das Hintergrundbild anzeigen lassen (also alles resetten), dann die Änderungen am Bild vornehmen und dann das Bild des Tools an der Cursorposition einfügen. Allerdings hat das den grossen Nachteil, dass erstens alle Änderungen am Bild (nicht in Pixelform sondern mittels Angaben zur Art und Weise wie sie hergestellt wurden) gespeichert werden müssen und zweitens, dass alle diese Änderungen für jedes Frame neu gezeichnet werden müssen.

Pseudocode

```
setup() { // einmal ausführen
  Hintergrundbild anzeigen;
  savepixels();
}

draw() { // im Loop ausführen
  printpixels();
  Aktuellestool an Cursorposition anzeigen;
}

mousePressed() { // bei Mausklick ausführen
  printpixels; // angezeigtes Tool wird „entfernt“
  drawtooltrace();
}

savepixels() {
  alle Pixel des aktuellen Bildes in Array speichern;
}

printpixels() {
  alle Pixel aus Array holen und anzeigen;
}

drawtooltrace() {
  Spur des aktivierten Tools an Curorposition zeichnen;
  savepixels();
}
```