Aufgabe 2: St. Egano

Team-ID: 00129

Team-Name: VGxSRlowMTZTV2RPYW...

Bearbeiter/-innen dieser Aufgabe:
Felix Mölle

16. Oktober 2023

Inhaltsverzeichnis

Kompletter Teamname:
VGxSRlowMTZTV2RPYWxGblRrUlpaMDVVVVdkT1JGVm5Ua2RGWjA1dFVXZE9ha2xuVFhwQlow
MTZWV2RPUkVsblRsUlZaMDVVVldkT1ZGbG5Ua2RSWjA1VVRXZE9SR05uVG1wUlowNTZXV2RPV
kVWblRsUkZaMDB5VVdkTk1sRTk=

1 Lösungsidee

Die Idee besteht darin zu erst das Bild einzulesen um Zugriff auf die RGB-Werte der einzelnen Pixel zu erhalten.
Man beginnt mit dem Pixel oben links (Aufgabenstellung).
Nun muss nach Aufgabenstellung der R-Wert ausgelesen werden.
Dieser wird in ein ASCII Zeichen umgewandelt und an die Gesamtzeichenkette angehangen.
Nun wird der G- und B- Wert ausgelesen und als x bzw. y Verschiebung des neuen Pixels genommen.
Die x Kooridnate lässt sich nun über $new\_x = (old\_x + x\_move)\%picture\_width$ berechnen Gleiches gilt für y nur das dabei modulo picture_height gerechnet werden muss. Durch berechnen des modulo wird erreicht, dass x bzw. y nie größer sind als das echte Bild und die Verschiebung richtig berechnet wird.
Nun wird das Pixel an der Stelle (x|y) abgerufen.
Mit diesem Pixel wiederholt man nun den Ablauf.
Das Ende hat man erreicht wenn $G = B = 0$ gilt.
Als letztes wird die nun finale Zeichenkette ausgegeben.

2 Umsetzung

Die Umsetzung dieser Aufgabe erfolgte in Rust, einer Programmiersprache, die für ihre Sicherheit, Geschwindigkeit und Nebenläufigkeit bekannt ist.

2.1 Einlesen der Eingabedaten

Die Funktion `read_data()` liest das Bild als `DynamicImage` aus einer Datei deren Pfad als Argument vorgegeben ist oder von der Benutzereingabe stammt. Das DynamicImage gibt z.B. Breite, Höhe und Pixel in eine gut verarbeitbare Form.
Dies geschieht über die Nutzung der `image` Bibliothek von Rust, welche sich für das Laden von allen gängigen Bildformaten eignet.
Dokumentation und SourceCode der Bibliothek: `https://docs.rs/image/latest/image/`

2.2 Analysieren der Pixel

Das Analysieren der Pixel geschieht in der Funktion `analyze_pixel(pixel: Rgba<u8>) -> (char, u32, u32)`
Dabei wird der R-Wert über eine simple `as` Konvertierung in ein ASCII Zeichen umgewandelt (`pixel[0] as char`) in dem der u8 Wert als ASCII Zeichenindex interpretiert wird.
Die G (pixel[1]) und B (pixel[2]) Werte werden zusammen mit dem Zeichen als dreigliedriges Tupel zurückgegeben

2.3 Algorithmus

Der Algorithmus beginnt damit, dass man den Output-String `out` definiert und initialisiert. Nun geht man wie in 1. beschrieben vor und startet oben links, wobei x = 0 und y = 0 gilt.

Folgendes geschieht nun solange $x \neq 0$ und $y \neq 0$:

- Das aktuelle Pixel wird anhand der Koordinaten aus dem Bild abgerufen
- Das aktuelle Pixel wird nun siehe 2.2 analysiert.
- Nun wird das neue Zeichen an den Output-String angehangen. → `out.push(ch);`
- Als letztes wird wie in 1. die x bzw. y Koordinate des neuen Pixels berechent.
  Die Höhe bzw. Breite des Bildes stammt dabei vom Einlesen der Daten

Schlussendlich wird noch der Gesamt-String `out` ausgegeben.

3 Beispiele

3.1 Bild 1

```
Hallo Welt
```

3.2 Bild 2

```
Hallo Gloria

Wie treffen uns am Freitag um 15:00 Uhr vor der Eisdiele am Markplatz.

Alle Liebe,
Juliane
```

3.3 Bild 3

```
Hallo Juliane,
Super, ich werde da sein! Ich freue mich schon auf den riesen Eisbecher mit Erdbeeren.

Bis bald,
Gloria
```

3.4 Bild 4

Der Jugendwettbewerb Informatik ist ein Programmierwettbewerb für alle, die erste Programmiererfahrungen sammeln und vertiefen möchten. Programmiert wird mit Blockly, einer Bausteinorientierten Programmiersprache. Vorkenntnisse sind nicht nötig. Um sich mit den Aufgaben des Wettbewerbs vertraut zu machen, empfehlen wir unsere Trainingsseite . Er richtet sich an Schülerinnen und Schüler der Jahrgangsstufen 5 - 13, prinzipiell ist aber eine Teilnahme ab Jahrgangsstufe 3 möglich. Der Wettbewerb besteht aus drei Runden. Die ersten beiden Runden erfolgen online. In der 3. Runde werden zwei Aufgaben gestellt, diese gilt es mit eigenen Programmierwerkzeugen zuhause zu bearbeiten.

3.5 Bild 5

Der Bundeswettbewerb Informatik richtet sich an Jugendliche bis 21 Jahre, vor dem Studium oder einer Berufstätigkeit. Der Wettbewerb beginnt am 1. September, dauert etwa ein Jahr und besteht aus drei Runden. Dabei können die Aufgaben der 1. Runde ohne größere Informatikkenntnisse gelöst werden; die Aufgaben der 2. Runde sind deutlich schwieriger.
Der Bundeswettbewerb ist fachlich so anspruchsvoll, dass die Gewinner i.d.R. in die Studienstiftung des deutschen Volkes aufgenommen werden. Aus den Besten werden die TeilnehmerInnen für die Internationale Informatik-Olympiade ermittelt. Der Bundeswettbewerb ermöglicht den Teilnehmenden, ihr Wissen zu vertiefen und ihre Begabung weiterzuentwickeln. So trägt der Wettbewerb dazu bei, Jugendliche mit besonderem fachlichen Potenzial zu erkennen.

3.6 Bild 6

Bonn
Die Bundesstadt Bonn (im Latein der Humanisten Bonna) ist eine kreisfreie Großstadt im Regierungsbezirk Köln im Süden des Landes Nordrhein-Westfalen und Zweitregierungssitz der Bundesrepublik Deutschland. Mit 336.465 Einwohnern (31. Dezember 2022) zählt Bonn zu den zwanzig größten Städten Deutschlands. Bonn gehört zu den Metropolregionen Rheinland und Rhein-Ruhr sowie zur Region Köln/Bonn.
...
Arbeitsmarktbehörden Bonn ist außerdem Standort der Zentralen Auslands- und Fachvermittlung (ZAV) der Bundesagentur für Arbeit (BA). Im Stadtteil Duisdorf befindet sich der Hauptsitz der ZAV mit ihren bundesweit 18 Standorten. Quelle: https://de.wikipedia.org/wiki/Bonn

3.7 Bild 7

Es hatte ein Mann einen Esel, der schon lange Jahre die Säcke unverdrossen zur Mühle getragen hatte, dessen Kräfte aber nun zu Ende giengen, so daß er zur Arbeit immer untauglicher ward. Da dachte der Herr daran, ihn aus dem Futter zu schaffen, aber der Esel merkte daß kein guter Wind wehte, lief fort und machte sich auf den Weg nach Bremen: dort, meinte er, könnte er ja Stadtmusikant werden.
...
Da machte ich daß ich fortkam." Von nun an getrauten sich die Räuber nicht weiter in das Haus, den vier Bremer Musikanten gefiels aber so wohl darin, daß sie nicht wieder heraus wollten. Und der das zuletzt erzählt hat, dem ist der Mund noch warm.

4 Quellcode

4.1 main.rs: Hauptprogramm

```rust
use std::env;
use image::{DynamicImage, GenericImageView, Rgba};

fn main() {
    let args: Vec<String> = env::args().collect();
    let mut fixed_path = String::new();

    if args.len() >= 2 {
        fixed_path = args[1].clone();
    }

    let picture: DynamicImage = read_data(fixed_path);

    let mut out = String::new();

    let mut x = 0;
    let mut y = 0;

    let mut pix: Rgba<u8> = picture.get_pixel(x, y);

    while !(pix[1] == 0 && pix[2] == 0) {
        pix = picture.get_pixel(x, y);

        let (ch, x_d, y_d) = analyze_pixel(pix);
        out.push(ch);

        x += x_d;
        x %= picture.width();

        y += y_d;
        y %= picture.height();
    }

    println!("{}", out);
}

fn read_data(fixed_path: String) -> DynamicImage {
    let mut input = String::new();

    if fixed_path.is_empty() {
        println!("Enter␣path␣to␣file:");
        std::io::stdin().read_line(&mut input).unwrap();
    } else {
        input = fixed_path;
    }

    let file_path = input.trim();
    println!("{}", file_path);

    let img = image::open(file_path).unwrap();

    img
}
```

```
55      fn analyze_pixel(pixel: Rgba<u8>) -> (char, u32, u32) {
            let ch: char = pixel[0] as char;
57          let x: u32 = pixel[1] as u32;
            let y: u32 = pixel[2] as u32;
59
            (ch, x, y)
61      }
```