

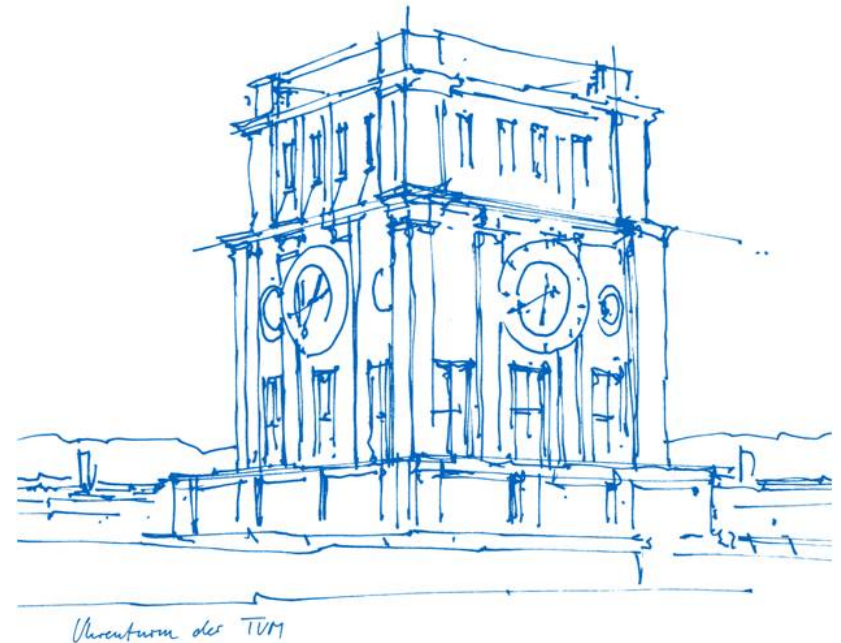
# Aufgabe A203: Run-Length-Encoding

Lukas Legner, Marlo Nickol, Fabian Thomas-Barein

Technische Universität München

Aspekte der systemnahen Programmierung bei der Spieleentwicklung

Garching, 01. Februar 2017



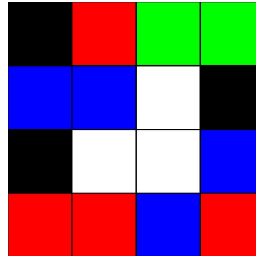
# Strukturierung des Vortrags

- Aufgabenstellung und Spezifikation
- Lösungsfindung
- Implementierung und Dokumentation
- Ergebnisse
- Zusammenfassung

# Aufgabenstellung und Spezifikation

- Lauflängencodierung einer BMP-Grafik

- Beispiel:



RLE ((1,S),(1,R),(2,G),(2,B),(1,W),(2,S),(2,W),(1,B),(2,R),(1,B),(1,R))

- RLE-Encoding speichert Tupel der Form  $T = (L,V)$  mit L hintereinander auftretenden Werten V.

# Aufgabenstellung und Spezifikation

- Bitmaps bestehen aus vier Teilen:
  - Dateikopf
  - Informationsblock
  - Farbtabelle
  - Bilddaten
- Dateikopf: enthält wichtige Dateiinformationen (Format, etc.)
- Informationsblock: spezifiziert Daten, die zum Einlesen der Bitmap benötigt werden (Höhe, Breite, Größe, etc.)
- Farbtabelle (enthält die Farbeinträge)
- Bilddaten

# Aufgabenstellung und Spezifikation

- Bilddaten im unkomprimierten Format
  - Für jeden Pixel des Bildes ist der Eintrag in der Farbtabelle angegeben
- Bilddaten im RLE-8 komprimierten Format
  - Zwei Bytes bilden einen Datensatz
  - „encoded mode“:
    - Pixel des zweiten Byte wird so oft wiederholt wie erstes Byte angibt
  - „absolute mode“
    - Wenn erstes Byte gleich 0 und zweites Byte gleich n, dann sind die nächsten n-Byte einzelne Pixelinformationen und bilden keine Tupel
  - Sonderkombination wie (0,0) oder (0,1), die Zeilenende oder Dateiende markieren

# Lösungsfindung

- C-Programm, dass Bitmap einliest und wieder neu speichert.
- C-Programm, dass ausschließlich „encoded mode“ codiert
- C-Programm, dass sowohl „encoded“ als auch „absolute“ codiert
- Übersetzung des C-Codes, der ausschließlich „encoded mode“ verwendet, in Assembler
- Übersetzung des fertigen Programms in Assembler
- Benchmarking

# Ergebnisse

- Die Implementierung in C benötigt für die Kompression durchschnittlich 0.0245 Sekunden bei 100 Durchläufen
- Die Lauflängencodierung in Assembler umgesetzt benötigt bei 100 Durchläufen durchschnittlich 0.0135 Sekunden
- Reduzierung der Bildgröße von 263 kB auf 248 kB



# Zusammenfassung

- RLE-Kodierung ist besonders geeignet für große Farbflächen mit wenigen Einzelfarben
- Umso ungeeigneter, je detailreicher die Grafik
- Zeichnet sich aus durch Einfachheit und Geschwindigkeit