



Irgendetwas mit ASCII-Art

Uwe Berger

bergeruw@gmx.net

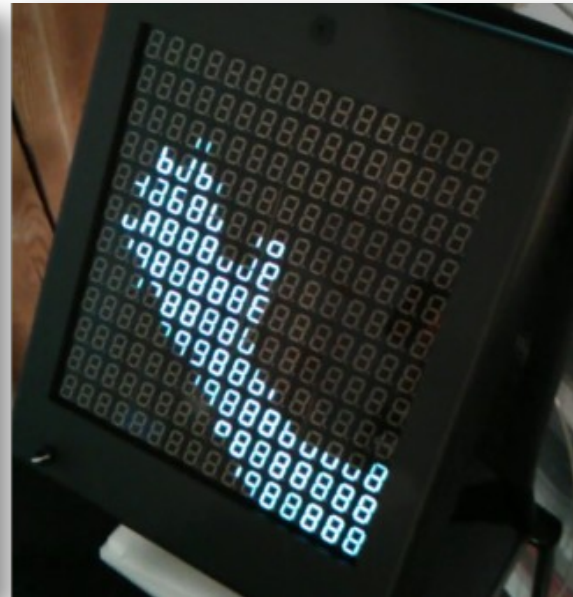
Uwe Berger



Motivation

Hackaday 2013

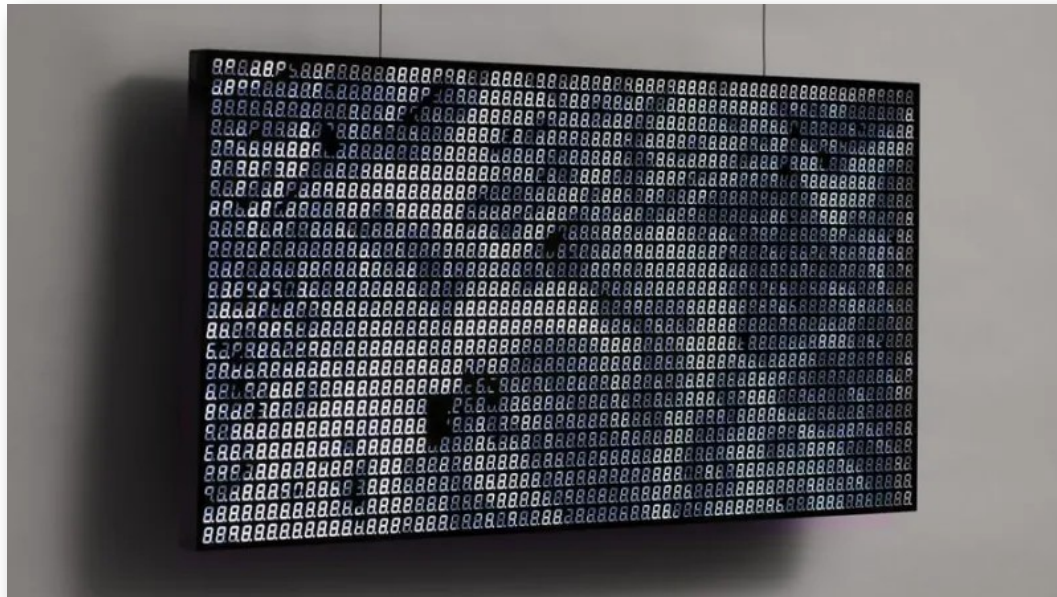
→ <https://hackaday.com/2013/11/21/7-segment-display-matrix-visualizes-more-than-numbers/>



Motivation

Hackaday 2023

→ <https://hackaday.com/2023/02/23/sailing-on-a-sea-of-seven-segment-displays/>



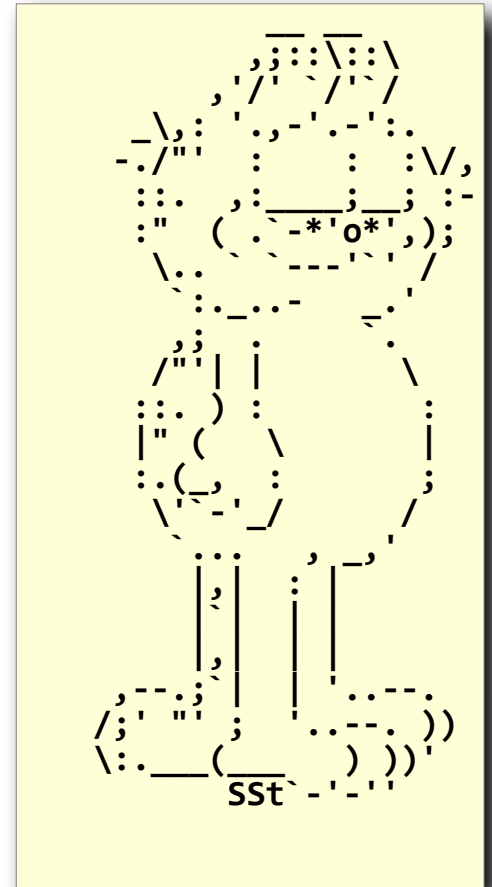
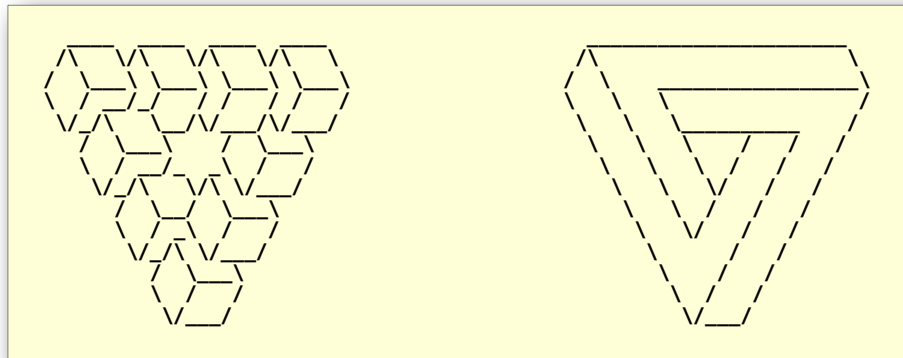
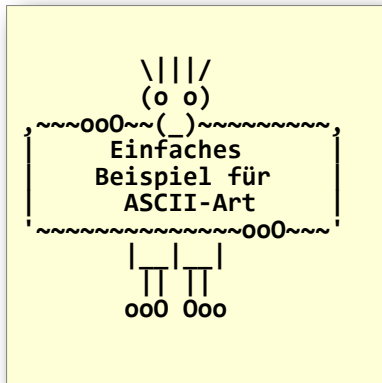
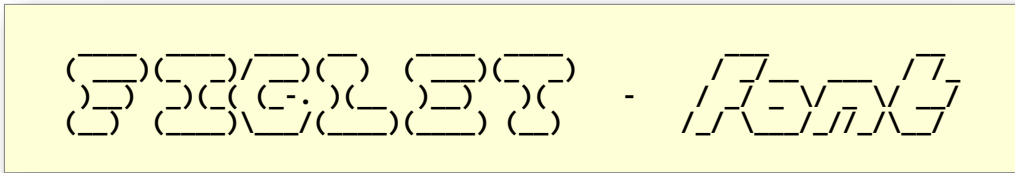
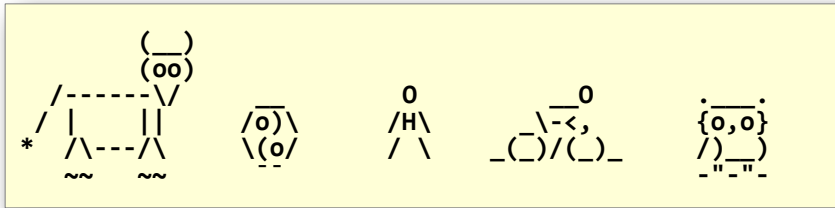
Was erzähle ich heute?

- ASCII-Art
 - Was ist das?
 - Programme, Tools etc.
- Eine 7-Segment Display Matrix
 - Wie funktioniert das?
 - 7s-Matrix-Simulator
 - Was kommt jetzt (vielleicht)?

ASCII-Art

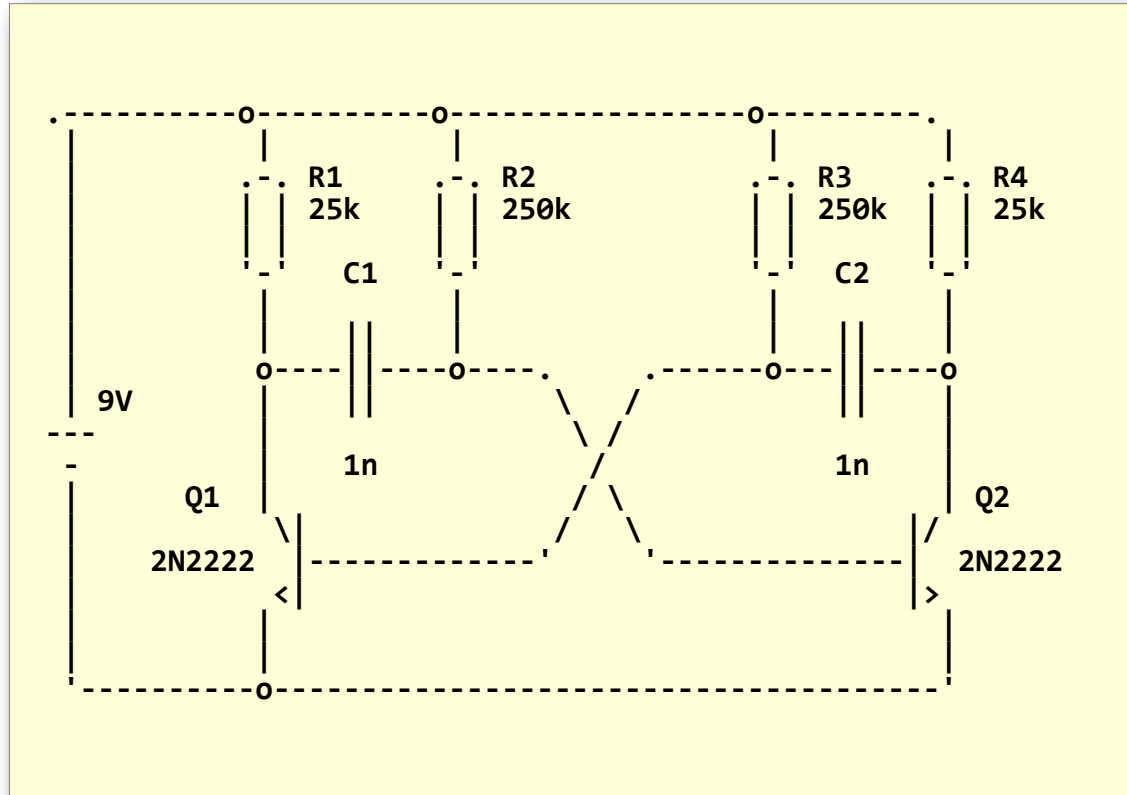
- Kunstrichtung, bei der man mit Buchstaben, Ziffern und Sonderzeichen Piktogramme, Bilder, Videos u.ä. darstellt
- Früher (vor Erfindung des Terminals) gab es auch den Begriff „Typewriter Art“ → Paul Smith

Beispiele für „einfache“ ASCII-Art



Quelle: <https://de.wikipedia.org/wiki/ASCII-Art>

Schaltpläne, Organigramme u.ä.



Quelle: <https://de.wikipedia.org/wiki/ASCII-Art>

„Aufwendige“ ASCII-Art



Bild: <https://aa-project.sourceforge.net/gallery/alfons.png>, (C) 2001 by Jan Hubicka

Programme, Tools ...

...zur Erstellung/Konvertieren von Bildern oder Video-Sequenzen in ASCII-Art basieren meist auf:

- aalib → <https://aa-project.sourceforge.net/aalib/>
- libcaca → <http://caca.zoy.org/wiki/libcaca>

Die zugrundeliegenden Algorithmen versuchen meist Bildbereiche des Originals durch äquivalente ASCII-Zeichen zu ersetzen (Kontur, visuelle Helligkeit/Kontrast, Farben etc.).

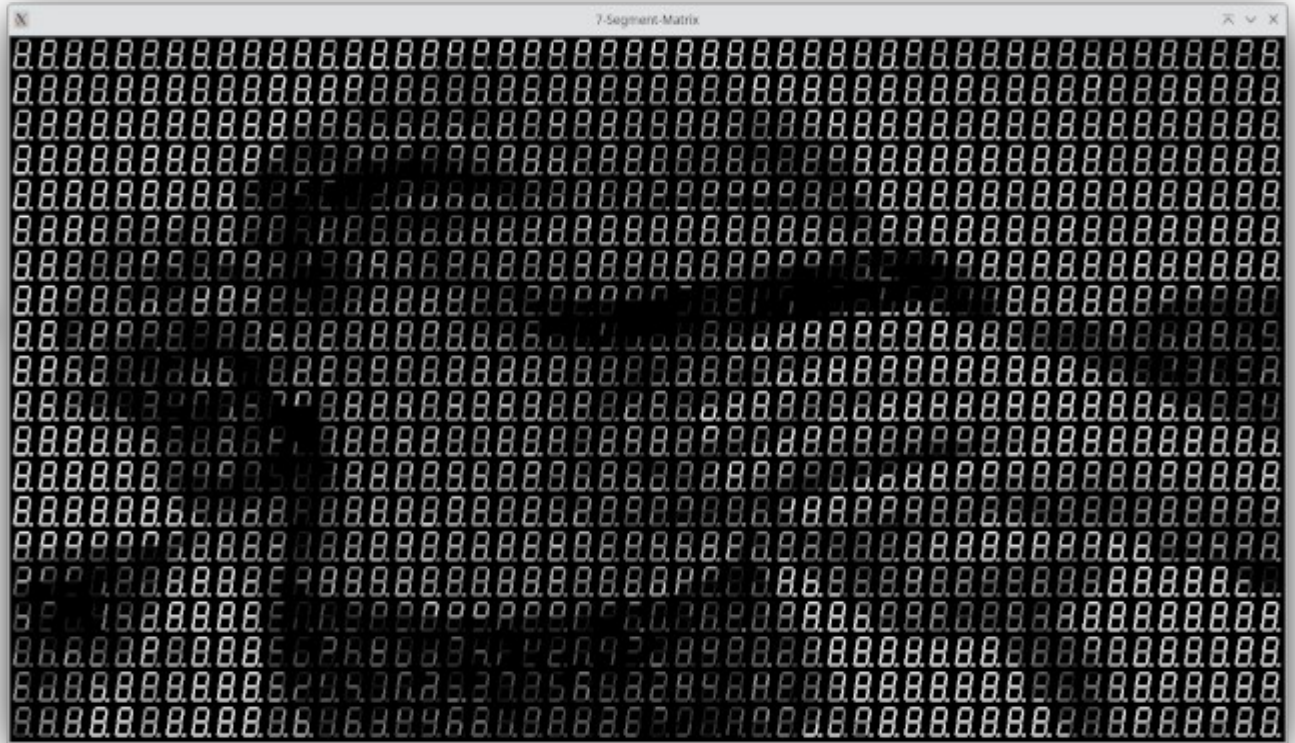
Programme, Tools (Beispiele)

- Bilder in ASCII konvertieren:
 - aview (nur .pnm-Bilder als Input)
 - jp2a (für .jpg, .png)
- Videos als ASCII anzeigen:
 - mplayer -vo aa video.avi
- Webcam-Output (o.ä.) als ASCII-Art-Stream:
 - Hasciicam

ASCII-Art...

...aber das soll eigentlich nicht Thema dieses Vortrages sein!

7-Segment Display Matrix



7-Segment Display Matrix

Also:

- Einzelne 7-Segment-Anzeigen (7s-Digits) sind in einer Matrix (7s-Matrix) angeordnet
- Die einzelnen Segmente der 7s-Digits der 7s-Matrix sind in ein xy -Koordinatensystem eingeteilt
- In diesem Koordinatensystem ist jeder Punkt einzeln ansteuerbar
- Auf dieser 7s-Matrix soll gezeichnet oder Bilder dargestellt werden können

7-Segment Display Matrix

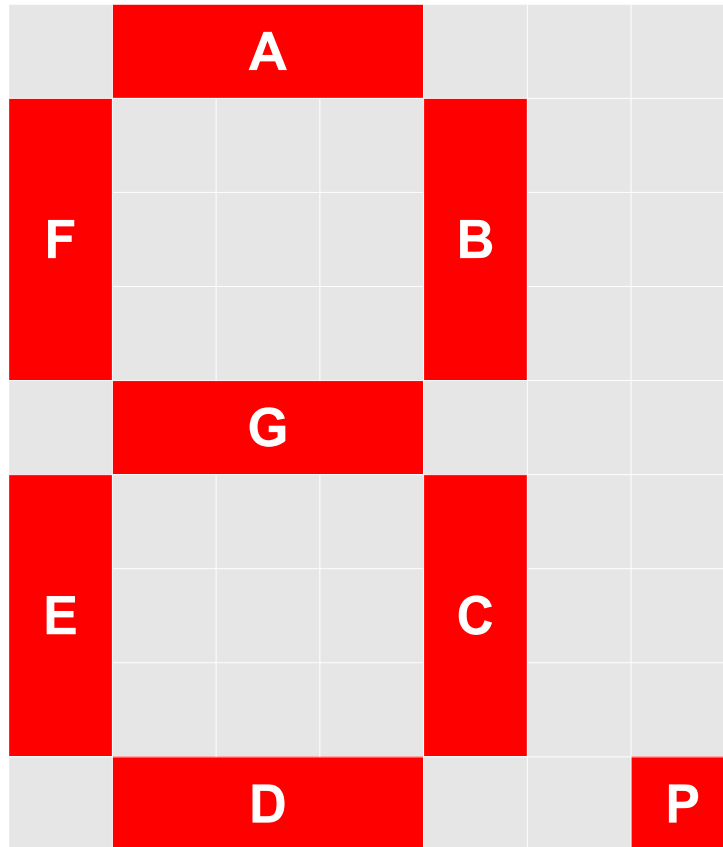
Zu klärende Frage:

- Wie teilt man die einzelnen Digit-Segmente sinnvoll in ein xy-Koordinatensystem ein?

Lösungsvarianten:

- Variante 1: „symmetrische“ Verteilung der Segmente
- Variante 2: „gepackte/komprimierte“ Verteilung der Segmente

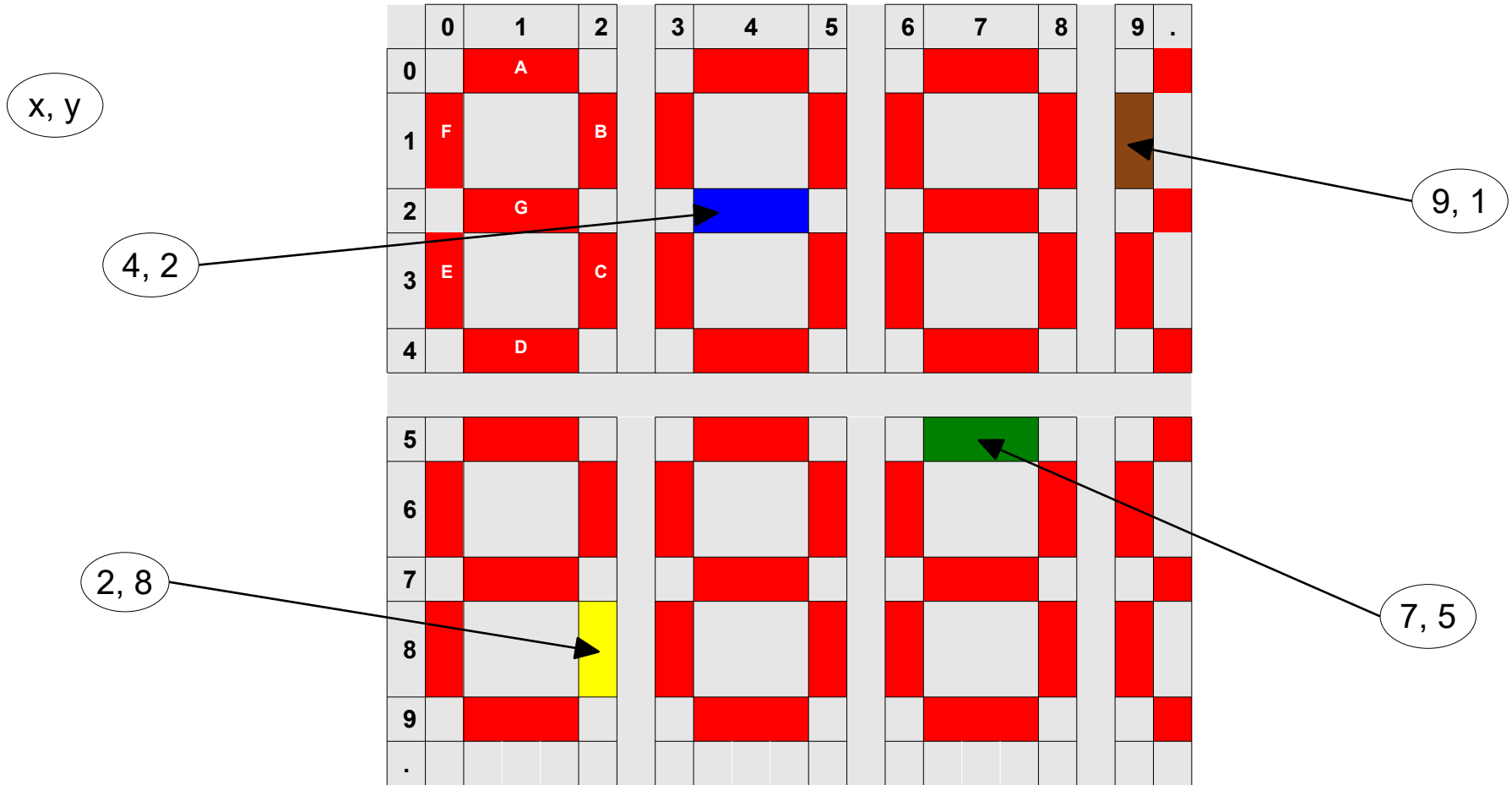
7-Segment-Digit



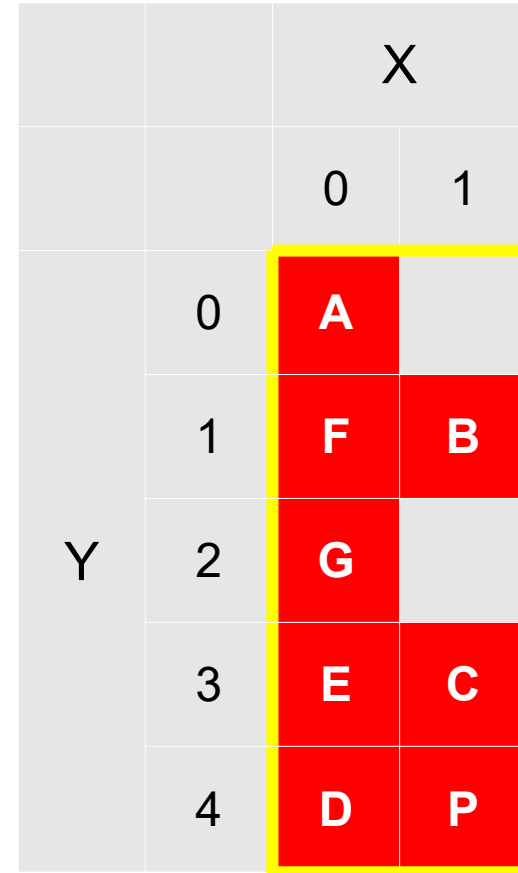
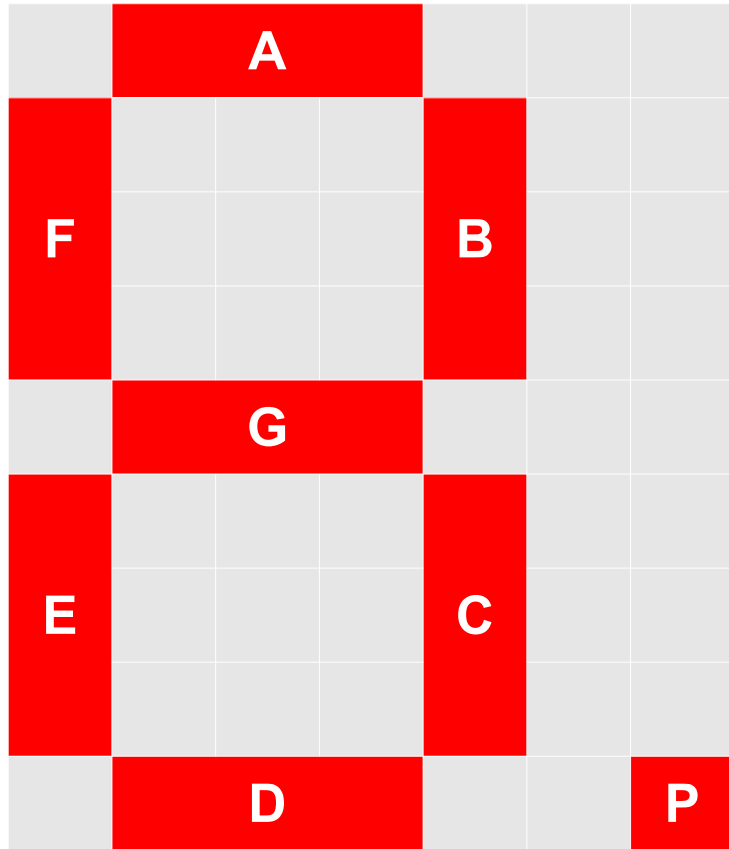
V1: Sym. Verteilung der Segmente

	0	1	2	3	4	5	6	7	8	9	.
0		A									
1	F		B								
2		G									
3	E		C								
4		D									
5											
6											
7											
8											
9											
.											

V1: Sym. Verteilung der Segmente

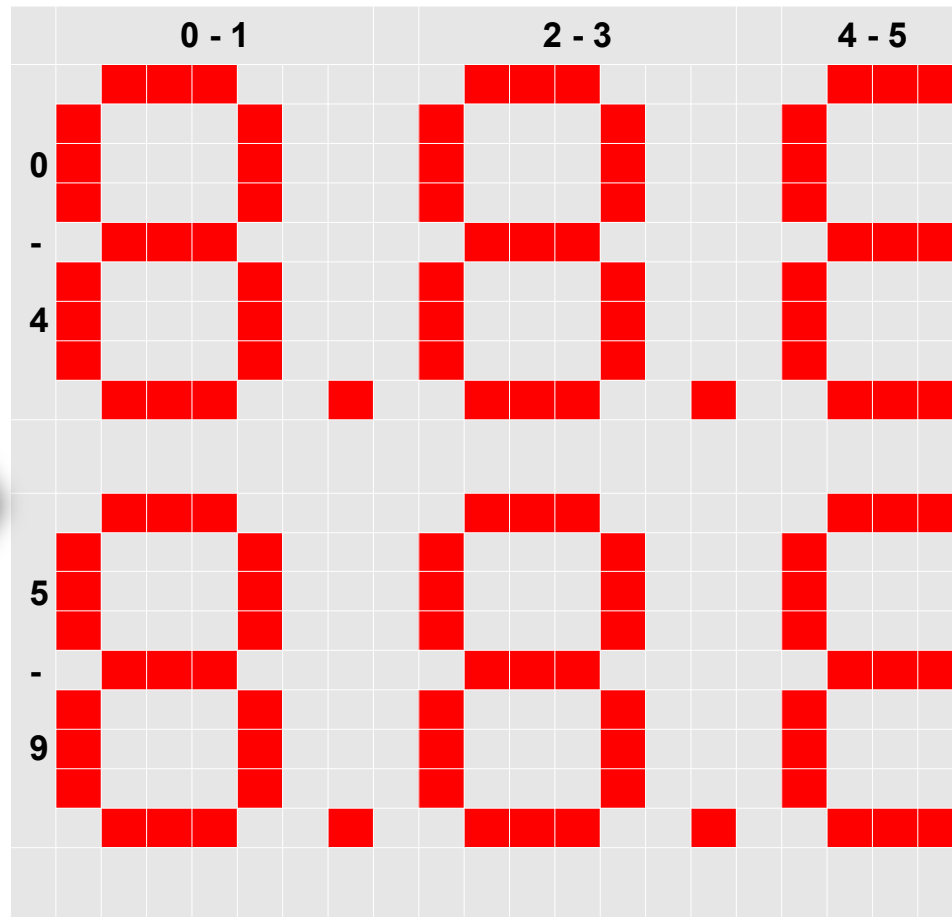


V2: gepackte Verteilung der Segm.

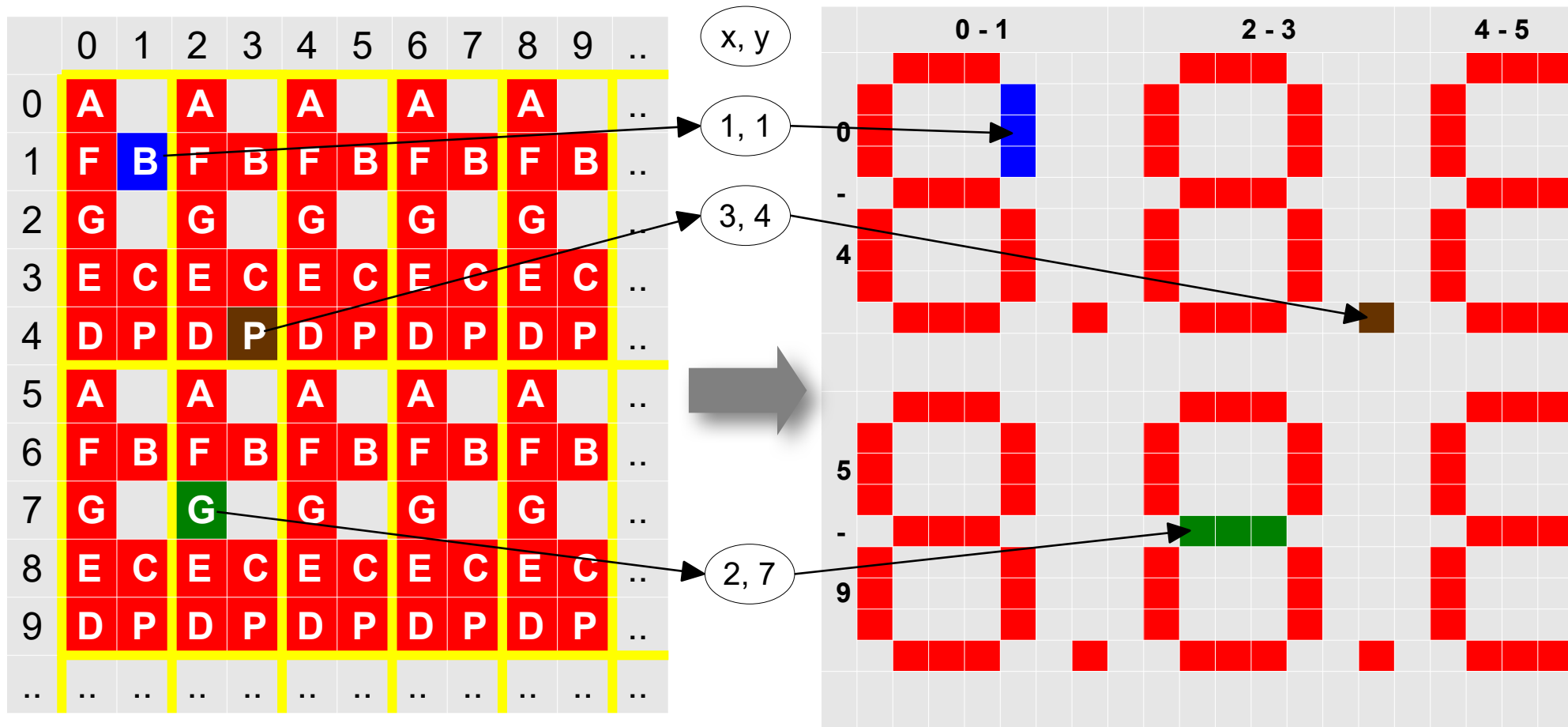


V2: gepackte Verteilung der Segm.

	0	1	2	3	4	5	6	7	8	9	..
0	A		A		A		A		A		..
1	F	B	F	B	F	B	F	B	F	B	..
2	G		G		G		G		G		..
3	E	C	E	C	E	C	E	C	E	C	..
4	D	P	D	P	D	P	D	P	D	P	..
5	A		A		A		A		A		..
6	F	B	F	B	F	B	F	B	F	B	..
7	G		G		G		G		G		..
8	E	C	E	C	E	C	E	C	E	C	..
9	D	P	D	P	D	P	D	P	D	P	..
..



V2: gepackte Verteilung der Segm.



7s-Matrix-Simulator...

Da es relativ aufwändig wäre, die Hardware „einfach mal so“ aufzubauen, wurde zuerst ein Hardware-Simulator in Software implementiert:

- ...um überhaupt mal anzufangen und die Machbarkeit zu prüfen
- ...um diverse Algorithmen ausprobieren
- ...um den Spieltrieb zu befriedigen

7s-Matrix-Simulator

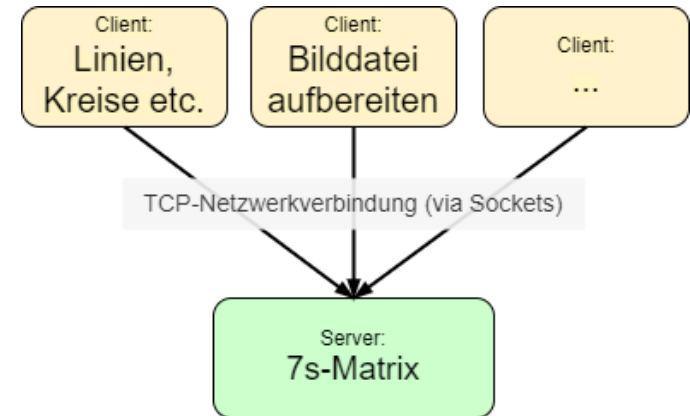
- Client/Server-Architektur

- Client

- generiert/konvertiert Bilder
 - sendet das Ergebnis zum Server

- Server

- generiert initial die 7s-Matrix in einem Grafikfenster
 - stellt empfangene Bilddaten auf dieser dar
 - ...ist der eigentliche Simulator und soll später durch Hardware ersetzt werden

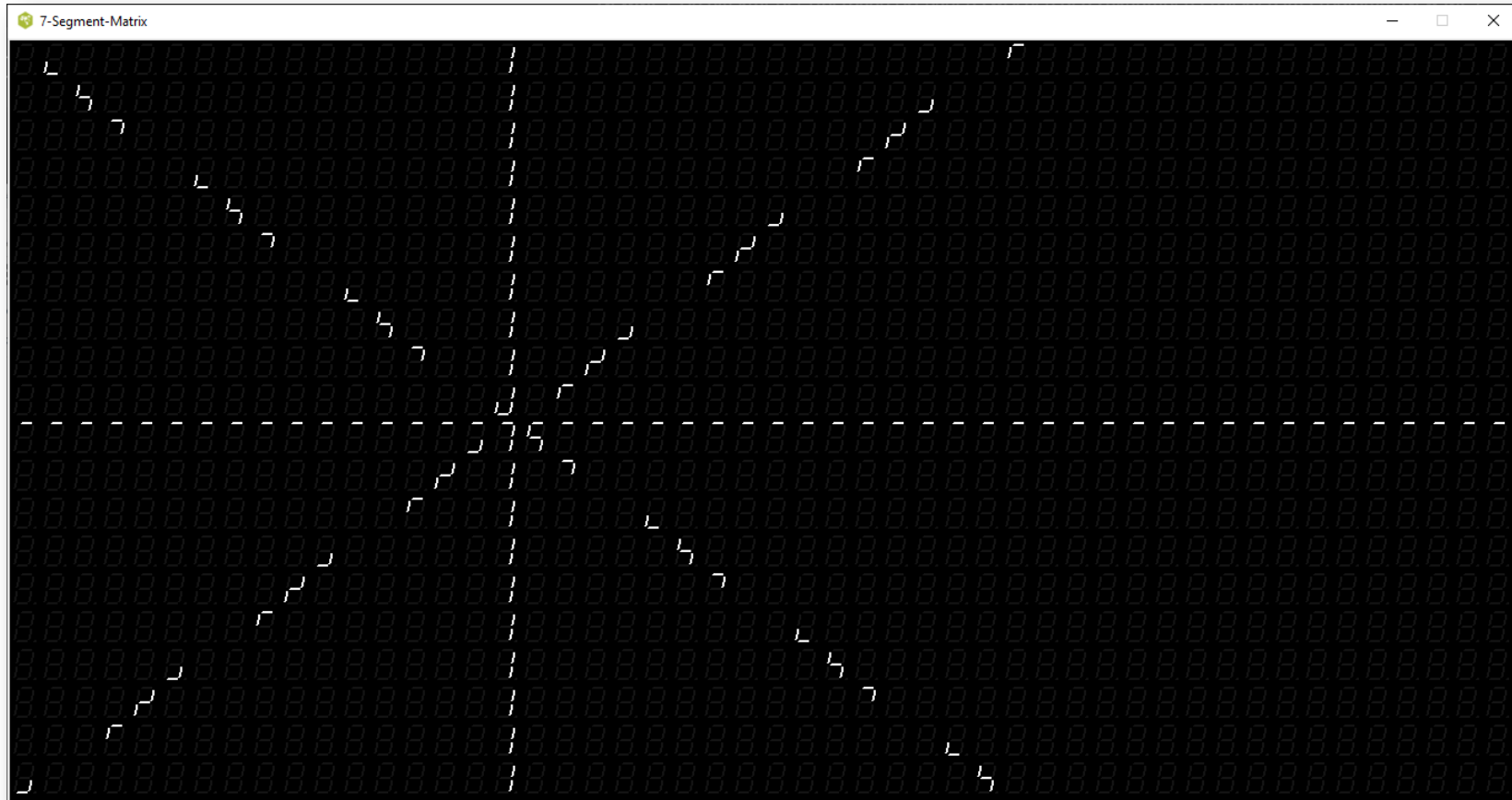


7s-Matrix-Simulator

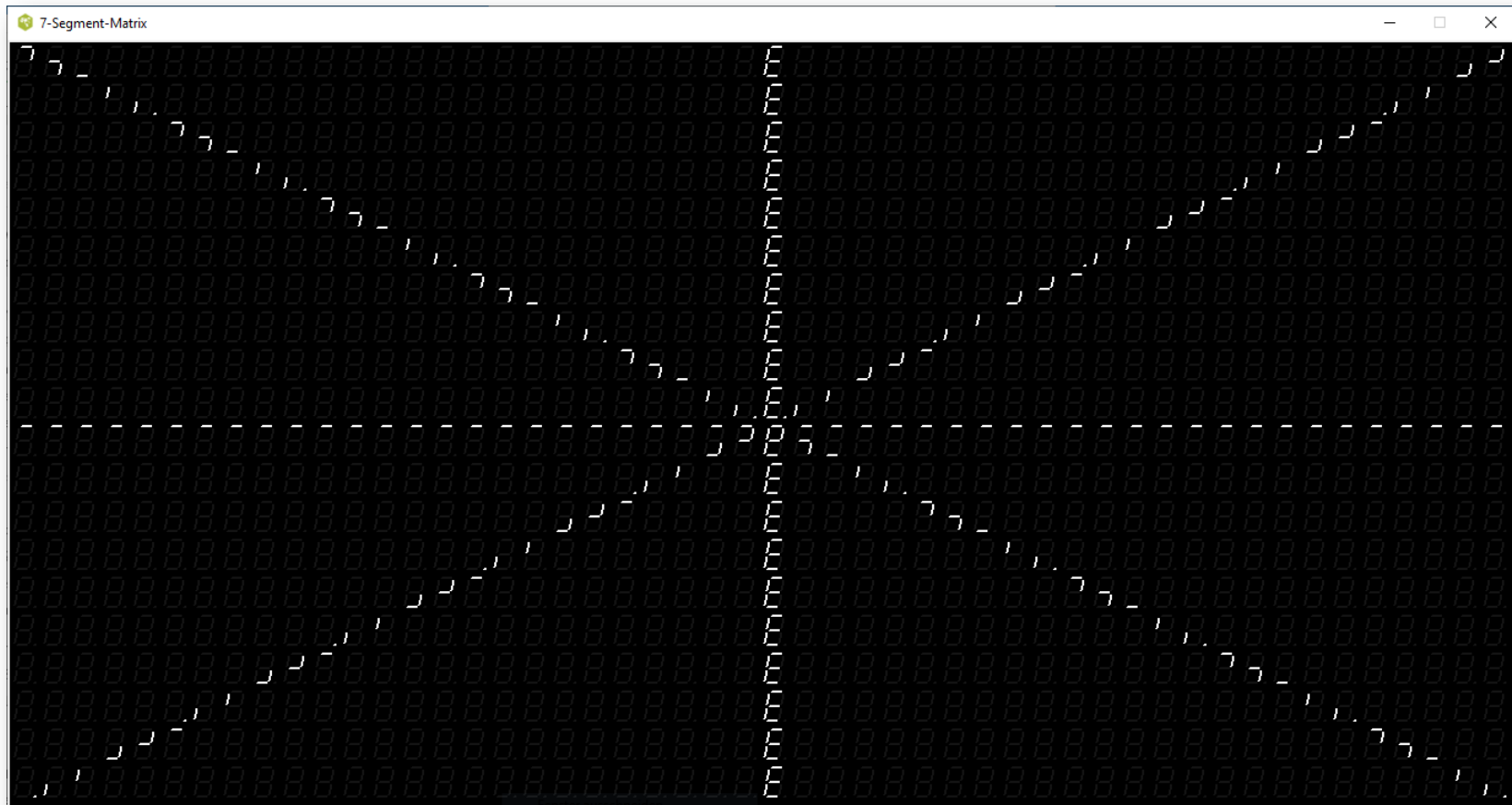
Evaluationsstufen:

- Erste Version in Tcl/Tk geschrieben...
- Wegen relativ geringer Performance → Portierung nach Python:
 - Kommunikation zwischen Client/Server via UNIX-FIFO-File
 - Weil Pythons tkinter auch nicht umwerfend schneller → Experimente mit pygame (später mehr dazu...)

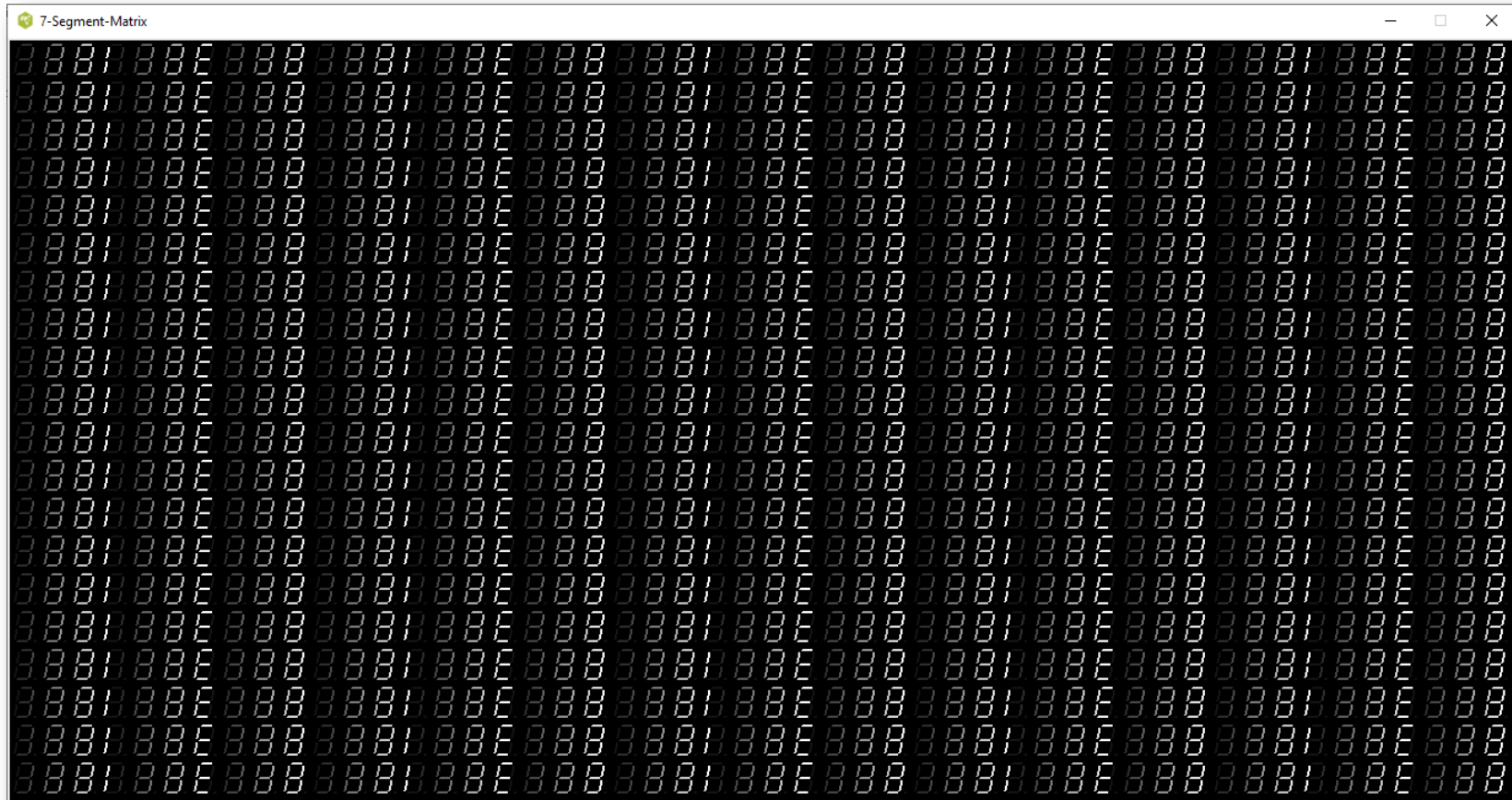
7s-Matrix-Simulator (mit V1)



7s-Matrix-Simulator (mit V2)



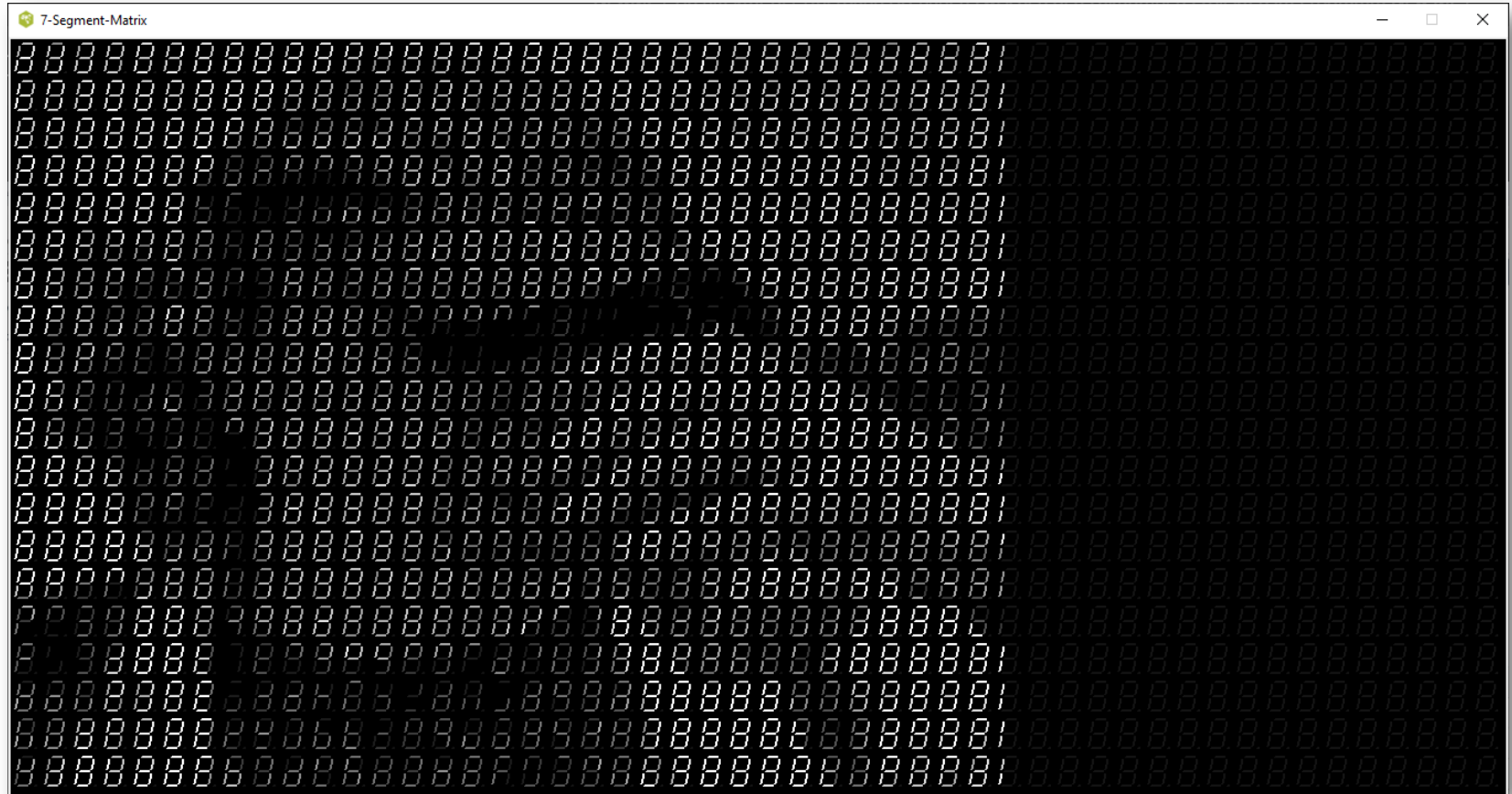
7s-Matrix-Simulator (mit V1)



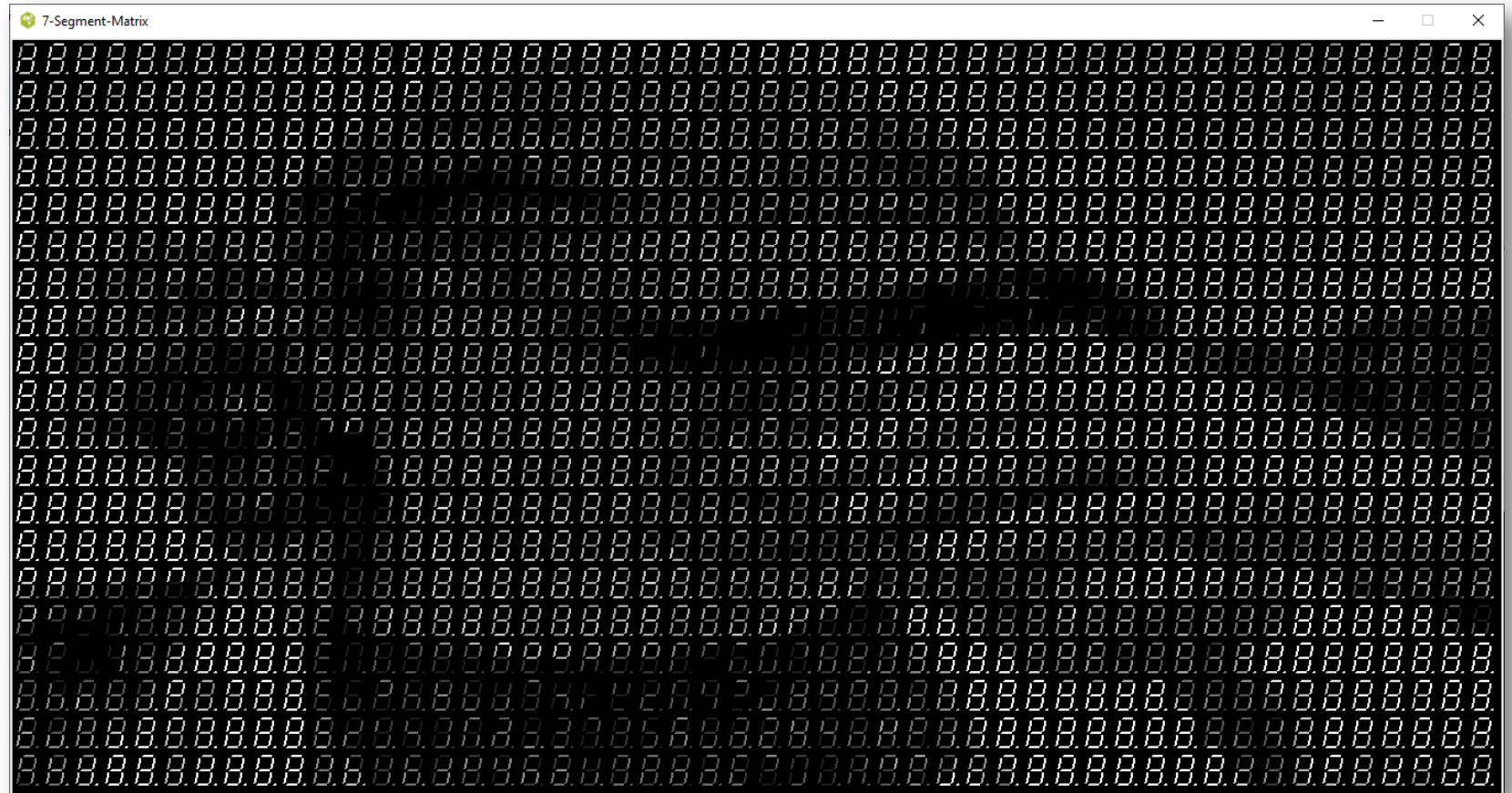
7s-Matrix-Simulator (mit V2)



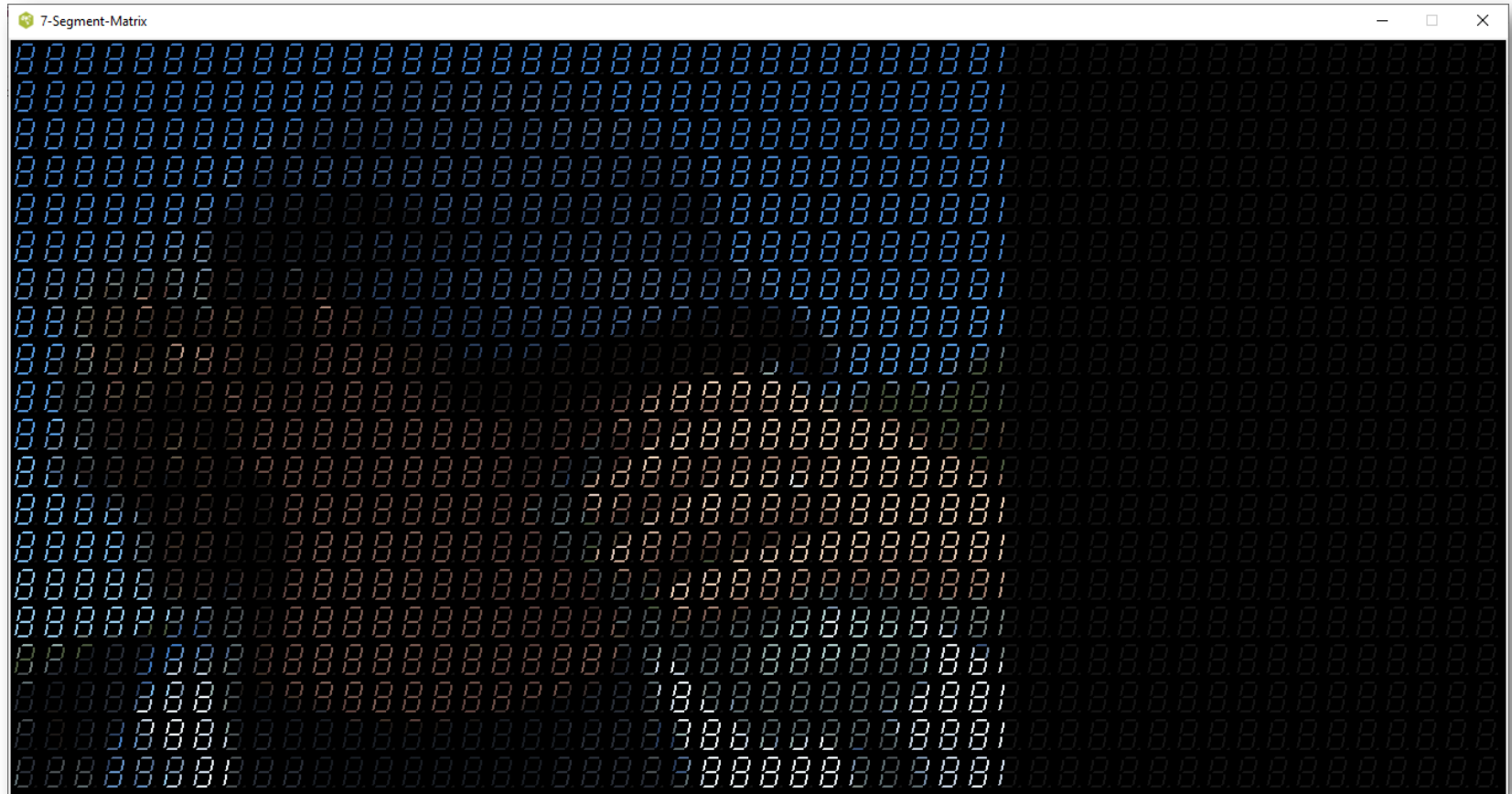
7s-Matrix-Simulator (mit V1)



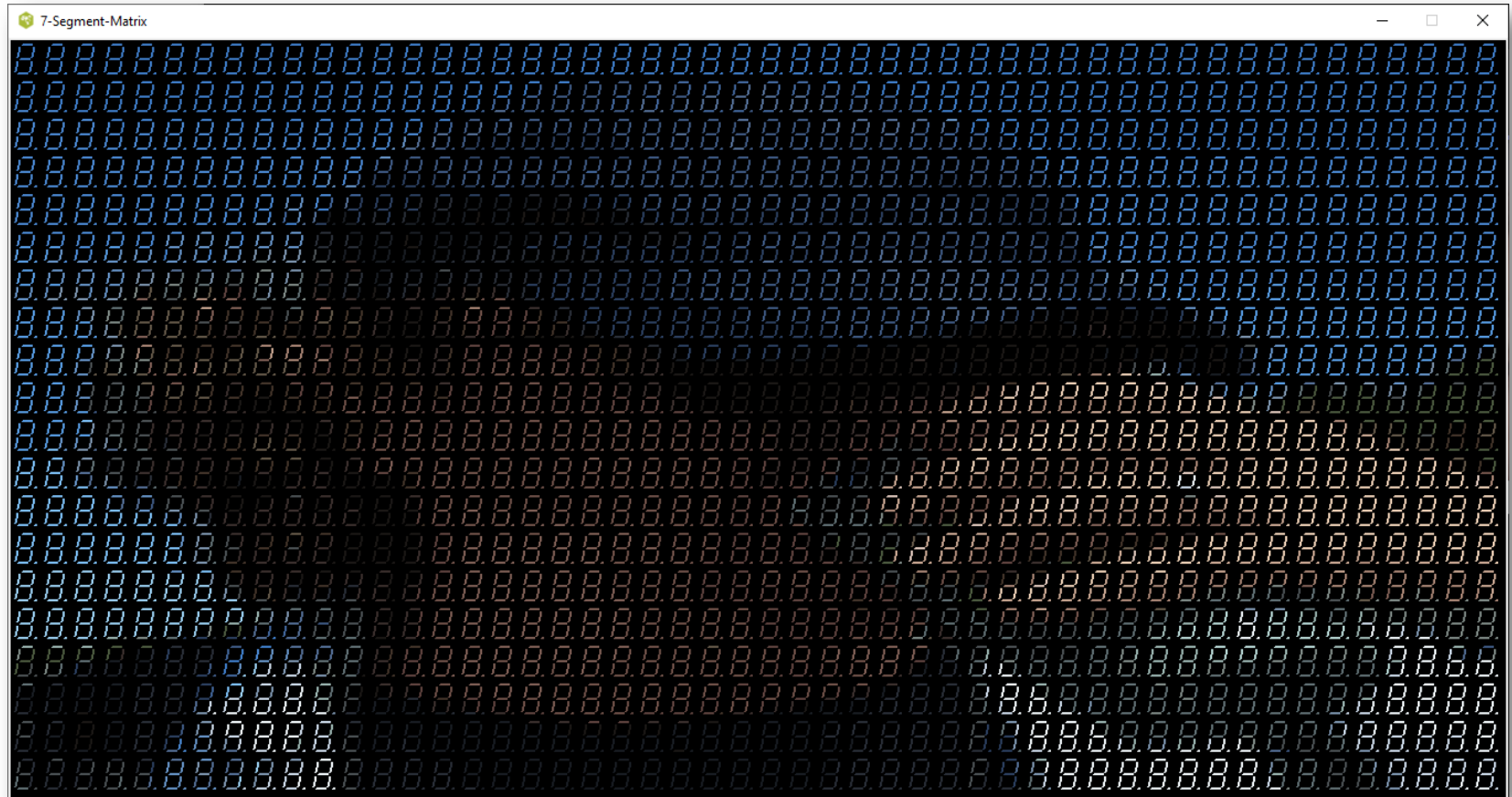
7s-Matrix-Simulator (mit V2)



7s-Matrix-Simulator (mit V1)



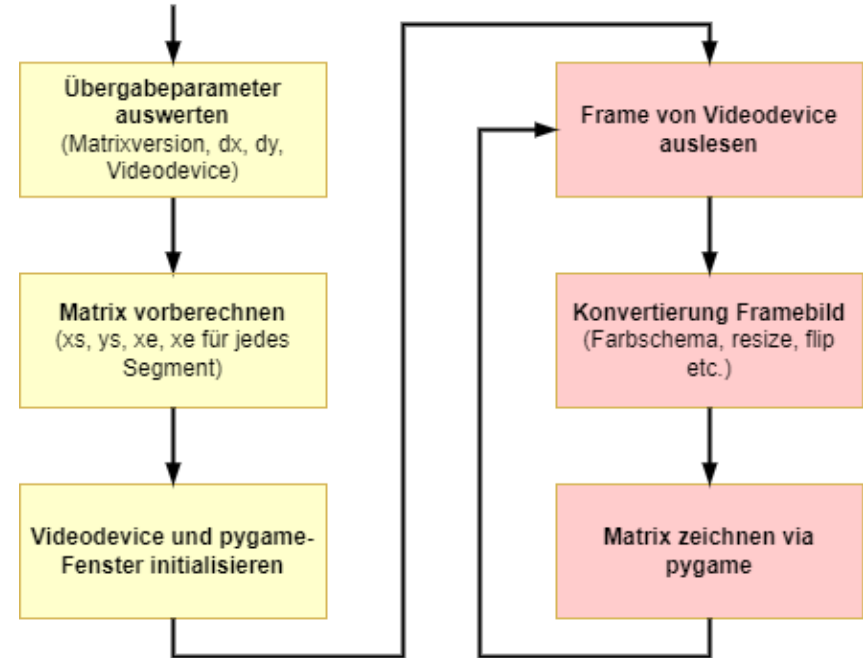
7s-Matrix-Simulator (mit V2)



7s-Cam...

7s-Cam:

- ...aus reinem Spieltrieb entstanden
- Ziel → testen, ob ein Live-Cam-Stream darstellbar ist
- Deshalb auch (erst mal) keine Client/Server-Architektur



Vergleich Varianten (Matrix-Koord.)

	Variante 1	Variante 2
Abbildbare „Pixel-Fläche“ auf einem Digit	<ul style="list-style-type: none">• dx=3• dy=5	<ul style="list-style-type: none">• dx=2• dy=5
Nicht anzeigbare Pixel pro Digit	8 (7 Segmente bei 15 Pixel; Digit-Punkt wird nicht verwendet)	2 (8 Segmente bei 10 Pixel; Digit-Punkt wird verwendet)
Pro	<ul style="list-style-type: none">• Symmetrisches Abbild• Digit-Verbrauch geringer (z.B. 48x50 Pixel: 160 Digits)	<ul style="list-style-type: none">• Guter Ausnutzungsgrad
Contra	<ul style="list-style-type: none">• Schlechter Ausnutzungsgrad	<ul style="list-style-type: none">• Unsymmetrisches Abbild• Höherer Digit-Verbrauch (z.B. 48x50 Pixel: 240 Digits)

Was kommt jetzt (vielleicht)?

7s-Matrix in Hardware...(?)....:

- Größe z.B. 48x50 Pixel (bei V2) entsprechen 24x10 Einzel-Digits bzw. 6x10 4er-Digits
- Zu lösende Probleme:
 - Ansteuerung: Graustufen (Helligkeit) via PWM...
 - Stromverbrauch: für 48x50 Pixel, bei V2 → ca. 38A!
 - Mechanischer Aufbau ...

Weiterführende Informationen

- <https://github.com/boerge42/Tcl-Magie>
- https://github.com/boerge42/7s_matrix/



Fragen?

...ansonsten Danke & Ende!