M @ z k ! n 3 n . k u n z t . m2cht . fr3!

Die Netzkunst und ihr(e) Code(s)

Inke Arns

1. "mezangelle" und "Kroperom" als künstlerische Appropriationen von Programmiercode

Die oft formulierte These vom “Verlust der Einschreibung” mit ihrem ausschließlichen Fokus auf den Oberflächentext als *dem* "Text" von Netzkunst bzw. -literatur geht von einer falschen Fragestellung aus. Es reicht meiner Ansicht nach nicht aus, hinsichtlich der “Oberflächeneffekte der Software” von einem “*performative turn* graphischer Benutzerschnittstellen” zu sprechen[1], denn diese Sichtweise bleibt zu sehr einer unterstellten Performativität eben jener Oberflächen verhaftet. Vielmehr muss man bei der Betrachtung von Netzkunst- und -literaturprojekten (wie auch bei Software allgemein) von mindestens zwei Texten ausgehen, einem "Phäno"- und einem "Genotext". Die Oberflächeneffekte des Phänotextes, z.B. sich bewegende Texte, werden durch andere, unter den Oberflächen liegende "effektive" Texte, den Programmcodes oder Quelltexten, hervorgerufen und gesteuert. Man könnte sogar behaupten, dass es sich bei Programmiercodes um illokutionäre Sprechakte[2] handelt, insofern, als hier "Sagen" und "Tun" zusammenfallen, diese "handlungsmächtigen" Sprechakte also keine Beschreibung oder Repräsentation von etwas sind, sondern direkt affizieren, in Bewegung setzen, Effekte zeitigen.

Bewegung und Stillstand, Dynamik und Statik, Linearität und Non-Linearität gehen im Code ein paradoxales Verhältnis ein: Während der invariable Code als Schrift sich “an einer Geraden [...] entfaltet”[3] - laut Burckhardt heißt *scriptum* im Lateinischen nicht nur Schrift, sondern auch Linie -, dieser Code somit Linearität und auch eine gewisse Statik verspricht, kann es doch bereits schon auf dieser Ebene nonlineare, dynamische, zyklische Zustände geben " und zwar noch bevor eine Oberfläche ins Spiel kommt, die sich bewegen könnte. Diese Kopplung von Statik und Dynamik im Code ist weniger paradox, wenn man mitdenkt, dass es die Performativität des (statischen, linearen) Textes ist, die den Text zum Abarbeiten textinterner Rekursionen und Schleifen anhält. Es handelt sich bei Programmcodes nicht um ein Aufzeichnungssystem von Mobilitäten oder Dynamiken, sondern um ein "Mobilisierungs-" bzw. "Immobilisierungssystem".

Die Privilegierung des Programmcodes gegenüber den Oberflächen, der *Poiesis* gegegenüber der *Aisthesis* führt in den ASCII-Arbeiten bzw. *Codeworks* von mez, Jodi und Netochka Nezvanova zu einer Befreiung insofern, als diese Fokussierung auf das "Virtuell-Unbewusste" eine Ent-Täuschung ermöglicht. Eine Ent-Täuschung darüber, bzw. Verabschiedung des Glaubens daran, dass auch heutzutage nur dann, wenn eine Kamera anwesend ist, Überwachung stattfindet (so die problematische These der Ausstellung ctrl_space am ZKM in Karlsruhe). Die *Codeworks* lenken unsere Aufmerksamkeit auf die zunehmende Codiertheit und Programmiertheit unserer medialen Umgebung. Sie bedienen sich einer "armen" Schrift im Medium E-Mail, die aber gleichzeitig im Kontext der Kommandozeile performativ bzw. ausführbar (*executable*) ist. Indem sie genau mit dieser Ambivalenz von Simplizität und Totalität der Ausführung arbeiten, verweisen sie auf die potentiell totalitäre Dimension des algorithmischen Genotextes.

Notes

1. [Matussek: 2001]
2. [John Langshaw Austin: How to Do Things with Words] (dt.: [Zur Theorie der Sprechakte, Stuttgart 1979])
3. [Burckhardt 2002], 35