

```
70 c1 f1 b2 35 cf 5e b6 dec 48 c1 db bc ff 41 07 86 33 62 1c c9 e8 f8 97 6f ba e9 3b 67 a^
d0 fa ef f9 f8 b9 23 5a dd 04 dc d2 8b 96 f0 7a b6 14 df 5c a9 cd 6e d4 38 aa 32 6b 9f
05 7f cb 0a cc 79 82 1e d8 0b 67 5f 62 46 c5 e7 9f 7b 66 8a ab 5c c9 41 89 2f 02 // length >= config.items) break;
f8 1e e3 56 71 e4 5c b8 83 2b 66 86 d4 23 23 66 50 26 5c dc 45 df 86 bb 86 b^
d3 50 d9 65 80 eb d5 b3 bc c7 75 07 60 97 f4 48 ac 50 96 fb d5 1a ad 7 // children.length >= config.items) break;
fd c7 6b 76 a1 5e eb 05 92 b7 93 77 bb 75 5a 25 87 03 b5 4d 4^
05 7e 05 30 22 0d 0f 11 3b e8 42 e5 59 6d 4a f8 83 87 84 // return matrix;
6b 06 66 f6 3f dd ff 31 25 21 d2 2c e2 37 8b 4f 94 8c // }
79 12 85 64 73 cd 72 95 35 bf 0f c1 38 3f 26 a^
5f e8 bf 60 62 91 11 f2 6f 0c 1a 32 0d 8c //
4b 55 e8 af c7 f9 95 1b 22 89 6a 4b r^();
6d 30 de 95 f0 08 fe 17 45 // canvas";
86 47 ec 6c 2d e8 87 // 0xfffff);
7d 96 2b 52 91 // pixelRatio(window.devicePixelRatio);
59 3a 9c // setSize(window.innerWidth, window.innerHeight);
eb // document.body.appendChild(renderer.domElement);
document.addEventListener("keypress", onDocumentKeyPress);
scene = new THREE.Scene();
scene.add(new THREE.AmbientLight(0xfffff));
}
camera = new THREE.OrthographicCamera(-window.innerWidth / 2, window.innerWidth / 2, window.innerHeight / 2, window.innerHeight / 2);
camera.position.set(0, 0, 0);
camera.lookAt(v3(0));
if (display == "main") {
  scene.add(mainObject(1, -window.innerWidth / 2, window.innerHeight / 2));
  setInitialMatrix(alphabet, "matrix");
}
else if (display == "mobile") {
  scene.add(mainObject());
  let camerabox = new THREE.Object3D();
  camerabox.add(camera);
  camerabox.name = "camerabox";
  scene.add(camerabox);
  control = new THREE.DeviceOrientationControl(scene);
}
else if (display == "history") {
  scene.add(ABCMatrix(hms));
  setInitialMatrix("H", "history");
}
window.addEventListener("resize", function() {
  let oCamera = scene.getObjectByName("camerabox").clone();
  oCamera.right = window.innerWidth / 2;
  oCamera.left = -window.innerWidth / 2;
  oCamera.top = window.innerHeight / 2;
  oCamera.bottom = -window.innerHeight / 2;
  oCamera.updateProjectionMatrix();
  renderer.setSize(window.innerWidth, window.innerHeight);
}, false);
// End of init() function
function onDocumentKeyPress(event) {
  let ch = String.fromCharCode(event.which);
  if (alphabet.indexOf(ch) + 1) {
    if (display == "main") rotateTo(ch, next_letter_space);
    if (display == "history") updateHMatrix(ch, 0);
  }
}
function ABCMatrix(config) {
  let matrix = new THREE.Object3D();
  matrix.name = config.name;
  for (let i = 0; i < config.length; i++) {
    for (let j = 0; j < config[i].length; j++) {
      let element = new THREE.Object3D();
      element.add(elementMesh.clone());
      element.visible = false;
      element.scale.multiplyScalar(1 / n + d * sum_{j in {1,...,n}} P/R_j);
      element.translateX(0);
    }
  }
}
function mainObject(scale = 1, Xpos = 0, Ypos = 0, name = "matrix") {
  let container = new THREE.Object3D();
  container.name = name;
  container.add(elementMesh.clone());
  container.scale.copy(v3(scale, scale, scale));
  container.translateX(Xpos).translateY(Ypos);
  container.length = container.children[0].length;
  container.tween = new TWEEN.Tween(container);
  return container;
}
function rotateTo(letter, time = 1000, start = true, o) {
  if (i == 0) o.visible = false;
  else {
    o.tween.stop();
    let target = targetRotation(i).normalize();
    let start = o.quaternion.clone().normalize();
    let slerpI = (t: 0);
    o.tween = new TWEEN.Tween(slerpI).to({t: 1}, time);
    o.tween.interpolation(TWEEN.Interpolation.Bezier);
    o.onUpdate(function() {
      THREE.Quaternion.slerp(start, target, o.quaternion);
    });
  }
}
function targetRotation(i) {
  let e = new THREE.Euler(D2r(abcP[1][x]), D2r(abcP[1][y]), 0);
  return new THREE.Quaternion().setFromEuler(e);
}
function updateLetter(letter) {
  if (alphabet.indexOf(letter) >= 0) {
    let matrixObject = scene.getObjectByName("matrix");
    let lPos = letter.charCodeAt(0) - "a".charCodeAt(0);
    blink(matrixObject.children[lPos], 50, 500);
  }
}
function setInitialMatrix(str = alphabet, name = "matrix") {
  var object = scene.getObjectByName(name);
  for (k = 0; k < Math.min(str.length, object.children.length); k++) {
    rotateTo(str.charAt(k), time, true, object.children[k]);
    if (str.charAt(k) == " ") object.children[k].visible = false;
    else object.children[k].visible = true;
  }
}
function updateHMatrix(letter, time = 500, name = "matrix") {
  let object = scene.getObjectByName(name);
  if (H.length >= config.r * config.c) {
    let net_j = sum_{i=1}^n x_i w_ij;
    y_j = phi(net_j - theta_j);
    K(t) = K_0 e^{Mt};
    W_network = [
      w11 w12 w13 w14
      w21 w22 w23 w24
      w31 w32 w33 w34
      w41 w42 w43 w44
    ];
    R_xy(tau) = (x * y)(tau) = integral_{-infinity}^infinity x*(y(t + tau)) dt;
  }
}
```



LEBEN IN DIGITALEN WELTEN

20.10.2017 - 05.08.2018

w ₁₁	w ₁₂	w ₁₃	w ₁₄
w ₂₁	w ₂₂	w ₂₃	w ₂₄
w ₃₁	w ₃₂	w ₃₃	w ₃₄
w ₄₁	w ₄₂	w ₄₃	w ₄₄

$$x_i = \frac{1-a}{n} + d \sum_{j \in \{1, \dots, n\}} \frac{P/R_j}{c_j} \quad R_{xy}(\tau) = (x * y)(\tau) = \int_{-\infty}^{\infty} x^*(y(t + \tau)) dt$$

Diese Ausstellung ist ein ungewöhnliches
bildungspolitisches Experiment!
Starten Sie es gemeinsam mit uns.

Editorial	2
Peter Weibel	
Open Codes 101	9
Ausstellungsmanual	
#GenealogieDesCodes	13
#Binär #Computing #Zahlensystem #Babel	
#Codierung	17
#Morsecode #Klangprogrammierung	
#Algorithmus #Software #Hardware #Interface	
#Decodierung	
#MaschinellesLernen	20
#KünstlicheIntelligenz #Kybernetik	
#Mustererkennung #AutonomeSysteme	
#SelbstfahrendeAutos #Drohnen #Roboter	
#AlgorithmicGovernance	23
#BigData #QuantifiziertesSelbst	
#Arbeit&Produktion	26
#Industrie4.0 #InternetDerDinge	
#Programmieren #SmartFactories #Automation	
#Arbeit4.0	
#AlgorithmischeÖkonomie	29
#Hochfrequenzhandel #Bitcoin	
#Kryptowährungen #Entschlüsseln #Blockchain	
#VirtuelleRealität	32
#HMD #ComputersimulierteUmgebungen	
#AugmentedReality	
#ComputerGeneriertesDesign #Eskapismus	
#GenetischerCode	35
#DNA #Quellcode #Bioengineering #Phänotyp	
#DNADatenspeicher #Genotyp	
Signal- und Maschinencodes	38
Franz Pichler	
Werkliste	42

Editorial

**DIE WELT VERSTEHEN, IN DER WIR LEBEN.
DIE WELT VERSTEHEN, VON DER WIR LEBEN.
DIE WELT VERSTEHEN, DIE WIR BEWOHNEN.**

Unsere Lebenswelt besteht heute zu wesentlichen Teilen aus einer künstlichen, von Menschen gemachten Datenwelt. Digitale Codes bilden den Zugang zu dieser Welt. So erscheint etwa beim Einschalten eines Mobiltelefons zunächst die Aufforderung: „Code eingeben ...“ und in Paris ist es üblich, sich durch die Eingabe eines Nummerncodes an der Tür Einlass in ein Haus oder einen Raum zu verschaffen. Codes bilden zentrale Schlüssel für den Zugang zu unserer analogen wie digitalen gegenwärtigen Welt.

Zu den ältesten Codes unserer Kultur zählen Alphabete und Zahlensysteme. In der Kommunikationswissenschaft bezeichnet ein Code im weitesten Sinne eine Sprache. Jegliche Kommunikation beruht auf dem Austausch von Informationen, die vom Sender gemäß einem bestimmten Code erzeugt werden und auf Empfängerseite gemäß demselben Code interpretiert werden. Allgemeiner gefasst basiert ein Code also auf einem Zeichenvorrat und bildet eine Abbildungsvorschrift für die eindeutige Zuordnung der Zeichen eines Zeichenvorrats zu denjenigen eines anderen. Beispielsweise werden dem Lautstrom der gesprochenen Sprache des Deutschen die 26 Buchstaben des lateinischen Alphabets zugeordnet, um die gesprochenen Vokale und Konsonanten durch Schrift abzubilden. Dieser visuelle alphabetische Code aus 26 Buchstaben kann wiederum, unter Verwendung kurzer und langer Tonsignale, in einen Morsecode übertragen werden. Das wesentliche Merkmal eines Codes ist also die Übersetzbarkeit von einem Code in einen anderen. Erstaunlich jedoch ist vor allem, dass sich mit einer relativ kleinen Menge an Zeichen, also sowohl mit den 26 Buchstaben des lateinischen Alphabets, als auch mit den drei Zeichen des Morsecodes (kurzes Signal, langes Signal, Pause) eine nahezu unendliche Menge an Sätzen produzieren lässt, das heißt eine potenziell unendlich große Informationsmenge codiert werden kann.

Morsecode

Die Signale des Morsealphabets werden mittels eines elektromagnetischen Schreibtelegrafen übertragen. Dabei lassen sich die Zeichen als Ton- oder Funksignal, als elektrischer Impuls über eine Telefonleitung durch Unterbrechung eines konstanten Signals mithilfe einer Taste, oder auch optisch, durch das abwechselnde Ein- und Ausschalten von Licht, senden. Der Morsecode besteht

grundlegend aus zwei Zuständen (Signal oder Pause) und einer zeitlich variierbaren Länge der Signale. Dieses Übertragungsverfahren wird Morsetelegrafie genannt, nach dem bildenden Künstler und Erfinder Samuel Morse, der 1833 das erste Modell eines funktionsfähigen elektromagnetischen Schreiblegrafen konstruierte. Anfangs ließen sich nur zehn Ziffern übertragen, die entsprechend einer Codierungstabelle in Buchstaben und Wörter übersetzt wurden ($a = \cdot -$). In einer weiterentwickelten Form bot der standardisierte Morsecode die entscheidende Funktechnologie für die Seefahrt.

Zahlencode

Während der alphabetische Code als Primärkode der menschlichen Kommunikation und Kultur für Jahrtausende vorherrschend war, dominiert in der heutigen Welt, wie die eingangs zitierten Beispiele zeigen, der numerische Code. Dieser besteht heute im Wesentlichen aus den zehn Ziffern 1 bis 9 und 0, womit nahezu unendlich viele Zahlen gebildet werden können. Eine ähnliche Leistung wie Morse für den alphabetischen Code erbrachte, vollbrachte Gottfried Wilhelm Leibniz 1697 für den numerischen Code.¹ Leibniz bewies, dass sich alle Zahlen mit lediglich zwei Ziffern, 0 und 1, darstellen lassen. Er nutzte als Entsprechungen für Gegenstände nicht nur wie üblich Wörter, Bilder oder Zahlen, sondern ordnete den Zahlen erstmals auch Ziffern zu. Leibniz (*De progressionem Dyadica*, 1679): „Man kann durch die Zahlen alle Arten von wahren Sätzen und Folgerungen darstellen.“ Leibniz' binäres Zahlensystem, sein binärer Code, mit dem er begann Wörter und Sätze in Ziffern zu übersetzen, war die Voraussetzung für den digitalen Code von heute.

Da in der digitalen Welt jede Information als Zahl verarbeitet wird, werden Buchstaben und Ziffern im Computer durch Bitfolgen dargestellt. Die Kombinationen von 0 und 1 (Bits) können als Ziffern, Zeichen oder Buchstaben (z. B. $a = 1100\ 0001$, $b = 1100\ 0010$) festgelegt sein. In der Kodierungstheorie nennt man die Elemente, aus denen ein Code besteht, „Codewörter“ und die Symbole, aus denen die Codewörter bestehen, das „Alphabet“. Während bis vor Kurzem noch die Codesysteme Sprache und Schrift der Kommunikation zwischen Menschen dienten, steht heute eine Vielzahl von Codesystemen zur Verfügung, mit deren Hilfe der Mensch auch mit Maschinen und Dingen kommunizieren kann, wie etwa der sogenannte Strichcode oder der QR-Code der Warenwirtschaft. Ein weiterer wichtiger Code ist der ASCII-Code (American Standard Code for Information Interchange), der zur Codierung von Zeichensätzen dient.

Als Quellcode, Quelltext oder auch Programmcode wird in der Informatik der für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogramms bezeichnet.

Er wird entsprechend den Regeln der jeweiligen Programmiersprache von Menschen erstellt. Oftmals ist der Quellcode im ASCII-Code verfasst. Damit aus dem Quellcode ein vom Computer ausführbares Programm wird, muss dieser in Maschinensprache umgewandelt werden, das heißt in Befehle übersetzt werden, die von einem Prozessor ausführbar sind.

Zur Geschichte der Digitalisierung

Bedeutende philosophische Bücher des 20. Jahrhunderts tragen Titel wie *Word and Object* (Willard Van Orman Quine, 1960)², oder *Les Mots et les choses* (Michel Foucault, 1966)³. Diese berichten von einer analogen Welt, die vor allem aus Dingen und der Beziehung zwischen Dingen und Wörtern besteht. Die Sprache ist demnach das Instrument, mit dem die Welt geordnet wird.

Entsprechend lautet Ludwig Wittgensteins bekanntes Dictum: „Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt“⁴.

In der Tat war die Sprache das erste Instrument, mit dem die Menschen die Welt erklären und gestalten konnten. Der Mensch gab den Dingen Namen und diese Beziehungen zwischen den Wörtern und Dingen waren für Jahrtausende kultur- und zivilisationsbestimmend. Ebenso wie die Menschen den Dingen Namen gaben, ordneten sie den Dingen Bilder zu, woraus eine zweite Kulturtechnik hervorging: die Kunst der Bildwelten, von der Malerei bis zur Fotografie. Die Dinge erzeugten auch Töne und der Mensch erschuf sogar neue Dinge eigens zur Erzeugung von Tönen.

Zur Welt der Bilder, Wörter und Töne gesellte sich die Welt der Zahlen. Die Mathematik ist die Welt der Zahlen. Man kann die Entwicklung der Digitalisierung in drei Stufen beschreiben. Die erste Stufe der Digitalisierung bzw. der Mathematisierung der Welt begann mit der Mathematisierung der Physik. Galileo Galilei schrieb 1623 „Das Buch der Natur ist in der Sprache der Mathematik geschrieben [...]“⁵. Bereits die Abbildungen der Dinge auf Wörter und Bilder stellen erhebliche Stufen menschlicher Abstraktion dar. Die Abbildung der Welt auf Zahlen und ihre Verselbständigung als Mathematik stellen die bisher höchste Stufe einer abstrakten Kulturtechnik dar, wodurch sich der Mensch von anderen Lebewesen unterscheidet. Mit dieser gesteigerten Abstraktion durch Mathematik und der Entwicklung der Naturwissenschaften als mathematische Disziplinen beginnt im eigentlichen Sinne bereits vor 400 Jahren die Digitalisierung. Mathematik wurde zur Universalsprache.

Schematisch ließe sich sagen: Im 17. und 18. Jahrhundert fand die Mathematisierung der Physik statt (1. Stufe), im 19. und in der ersten Hälfte des 20. Jahrhunderts die Mathematisierung des Denkens (2. Stufe). Im weiteren Verlauf des 20. Jahrhunderts konvergierten beide Tendenzen in der Elektronik (3. Stufe). Isaac Newton legte mit seinem Hauptwerk *Philosophiae Naturalis*

Principia Mathematica von 1686 den Grundstein für die mathematische Beschreibung der Natur. Joseph-Louis de Lagranges Meisterwerk *Mécanique analytique* von 1788 bietet erstmals eine vollständige Beschreibung des Universums rein auf Basis algebraischer Operationen. Er überführte die Physik in die analytische Mathematik. Lagrange algebraisierte die Mathematik und mathematisierte die Physik. Diese Algebraisierung der Physik führte zur 2. Stufe der Digitalisierung: die Algebraisierung der Logik (des formalen Denkens). Logische Tatbestände wurden mithilfe von mathematischen Methoden und Begriffsbildungen erfasst. Als Reaktion auf Newton schrieben daher Bertrand Russell und Alfred North Whitehead ihre dreibändigen *Principia Mathematica*, 1910–1913. Ebenso wie Gottlob Frege, der anhand seines Werkes *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* von 1879 das Denken in mathematische Formeln übersetzte, bildeten auch Russell und Whitehead das Denken und die Logik auf Mathematik ab.

Einen entscheidenden Meilenstein setzte George Boole, indem er die Gesetze des Denkens als Gesetze der formalen Logik und diese, in Fortsetzung von Lagrange, als algebraische Mathematik definierte. In *The Mathematical Analysis of Logic*⁶ (1847) und in *An Investigation of the Laws of Thought* (1854) bewies Boole, dass Logik und Algebra ident sind, indem er logische Aussagen als algebraische Gleichungen darstellte. Alan Turing schließlich trieb diese Tendenzen der Mathematisierung von Welt, Sprache, Logik und Denken in seinem berühmten Essay „On Computable Numbers“ von 1936 auf die Spitze. Turings Darstellung der Berechenbarkeit von Zahlen und Zahlenprozessen gilt als das grundlegende Papier für die Entwicklung des digitalen Computers, für die sogenannte Turing-Maschine. Fortan wird nicht mehr nur mit Zahlen gerechnet, Zahlen werden vielmehr selbst berechenbar. Mit berechenbaren Zahlen wird die Natur errechenbar.

Mit der Weiterentwicklung des Computers von einer reinen Rechenmaschine zu einer Bild-, Ton- und Sprachmaschine entstand eine neue Welt der Daten. Bilder und Texte können errechnet werden und visuelle und akustische Welten simulieren. Mit einem Wort: Alles das, was bisher aus Objekten, Wörtern, Tönen und Bildern bestand, kann auf Zahlen abgebildet und aus Zahlen konstruiert werden. Das entscheidende Moment dieser digitalen Kulturtechnik ist eine bis dato unvorstellbare Reversibilität. In der analogen Welt herrschte in der Beziehung zwischen den Dingen und Wörtern bzw. Bildern das Prinzip der Irreversibilität. Die Dinge können in Wörter verwandelt werden, aber die Wörter nicht in Dinge rückverwandelt, weil das Wort „Stuhl“ eben kein Stuhl ist. Die Dinge können in Bilder verwandelt werden, aber die Bilder nicht in Dinge, weil das Bild einer Pfeife keine Pfeife ist – man denke an René Magrittes Gemälde *La trahison des images* von 1929, auf dem eine Pfeife abgebildet ist, darunter der Schriftzug

Ceci n'est pas une pipe – Dies ist keine Pfeife. Im Zeitalter der Digitalisierung werden nun aber Wörter, Bilder und Töne in Daten verwandelt, und – erstmals in der Geschichte der Menschheit – können diese Daten auch in Töne, Bilder und Wörter zurückverwandelt werden. Und durch den 3-D-Druck können Daten sogar in Dinge verwandelt werden. Die Beziehungen zwischen Daten und Dingen, Wörtern, Bildern sind reversibel. Die Sprache der Daten, die Algorithmen und Programmiersprachen, sind zu einer universellen Sprache geworden, aus der die Welt der Töne, Bilder, Texte und Dinge entsteht. Die Mathematik ist also längst nicht mehr nur die Sprache der Natur, sie ist zu einer Sprache der Kultur geworden. Das Buch, das die gegenwärtige Welt beschreibt, müsste also den Titel *Die Dinge und die Daten* tragen. Die Beziehung zwischen Dingen, Wörtern und Bildern war irreversibel. Die Beziehungen zwischen Daten und Wörtern, Bildern und Tönen sind in der digitalen Welt reversibel.

Digitale Codes

Die digitale Kulturtechnik bildet aber auch die Grundlage für eine weitere Revolution, die vielleicht ein neues Zeitalter einleitet. Die bisherige Kultur basiert auf einer zweidimensionalen Notation: Ebenso wie die Schrift sind Noten, Zahlen und Zeichen auf Papier notiert und fixiert. Der Computer jedoch ermöglicht die Simulation eines bewegten dreidimensionalen Raums und somit eine zukünftige dreidimensionale Notation, derer sich schon heute Architekten und Designer bedienen. Das 3-D-Kino war der erste Versuch in diese Richtung, aber mit dem 3-D-Druck beginnt diese Zukunft nun Realität zu werden, durch die oben beschriebenen Möglichkeiten der reversiblen Transformationen. Dank der Entwicklung dieser Kulturtechnik, welche die Beziehung zwischen der Ding- und Zeichenwelt reversibel macht, werden wir in einer Umwelt leben, die von Sensoren und intelligenten Agenten gestützt, von Codes und Algorithmen geleitet und mit künstlicher Intelligenz ausgestattet sein wird.

Dass dies möglich wurde, geht zurück auf „The Unreasonable Effectiveness of Mathematics in the Natural Sciences“, die der Nobelpreisträger Eugene Wigner 1960 feststellte. Realität ist, was mathematisch repräsentierbar und elektronisch schaltbar ist. Das beste Beispiel hierfür bietet Claude E. Shannons Masterarbeit *A Symbolic Analysis of Relay and Switching Circuits* von 1937. In dieser Arbeit bewies Shannon, dass die Boolesche Aussagenlogik mit den Wahrheitswerten 0 und 1 verwendet werden kann, um die durch elektrischen Strom betriebenen, elektromagnetisch wirkenden, fernbetätigten Schalter mit zwei Schaltstellungen zu steuern. Wie der Titel besagt, werden Stromkreise und Schaltkreise, Anordnungen von Relais und Schaltern, in einer symbolischen Analyse auf die Boolesche Aussagenlogik

abgebildet. Die Boolesche Algebra wird also zur Schaltalgebra. Die von Shannon vorgeschlagene Verknüpfung der logischen Gesetze mit der Steuerung von Schaltkreisen, das heißt der Gebrauch der binären Eigenschaften elektrischer Schaltkreise (on – off, 1 – 0, Strom – Nicht-Strom) zur Ausführung logischer Operationen, wurde fortan für den Aufbau aller elektronischen digitalen Computer bestimmend. Shannon zeigte, dass die mentalen Formeln der Booleschen Algebra in materielle Schaltalgebra übertragen werden konnten. Formales Denken wurde in elektronische Schaltkreise – nach Regeln der Booleschen Algebra überführt. Die Elektronik wurde zur Physik der Mathematik!

Im Verbund mit der Entdeckung der elektromagnetischen Wellen durch Heinrich Hertz (1886–1888), das heißt der Erfindung der Telekommunikation (Telegrafie, Telefonie, Television, Radar, Rundfunk, Satellit, Internet) sowie der Entwicklung von Transistoren (1947), integrierten Schaltkreisen und Mikrochips wurde die Mathematisierung der Welt in den letzten einhundert Jahren in die materielle Welt der Elektronik übertragbar. Daher muss die Gleichung „Machinery, Materials, and Men“ (Frank Lloyd Wright, 1930), die für das 19. und 20. Jahrhundert gültig war, für das 21. Jahrhundert um die Gleichung „Medien, Daten und Menschen“ (Peter Weibel, 2011) erweitert werden. Seitdem der alphabetische Code durch den numerischen Code ergänzt wurde, stellen Algorithmen ein fundamentales Element unserer sozialen Ordnung dar.

Das Konzept der Ausstellung – ein bildungspolitisches Experiment

Mit circa 200 künstlerischen und wissenschaftlichen Arbeiten soll die Welt der digitalen Codes und der von ihnen beeinflussten künftigen Lebensformen in acht Bereichen dargestellt werden: [#GenealogieDesCodes](#), [#Codierung](#), [#MaschinellesLernen](#), [#AlgorithmicGovernance](#), [#AlgorithmischeÖkonomie](#), [#Virtuelle-Realität](#), [#Arbeit&Produktion](#) und [#GenetischerCode](#). Die präsentierten Werke bieten Ihnen die Möglichkeit, eine ungewöhnliche Form der Auseinandersetzung mit der Kunst zu erproben und den Ausstellungsbesuch selbst ein Stück weit neu zu definieren. Anders als bei der vertrauten Rezeption von analogen Gemälden, Skulpturen oder Installationen, erschließt sich der Bedeutungshorizont der Werke erst im Prozess physischer Interaktion zwischen BetrachterIn und Werk. Die Partizipation des Publikums ist der Moment, in dem die Werke materiell entstehen. Somit schließt die partizipatorische und analytische Auseinandersetzung mit den Werken neue Formen der Konzentration, Meditation, aber auch der Zerstreung mit ein. Der *Discours* der Ausstellung ist als architektonischer *Parcours* angelegt, um Ihnen die Gelegenheit zu bieten, selbstbestimmt sowohl zwischen Inseln der Kunst und des Wissens zu wandeln als auch an den sogenannten Work Stations aktiv und

kreativ zu werden. Neben den Co-Working-Stations können Sie auch Orte der Ruhe und Rekreation finden. Da sich die Werke erst bei längerer Verweildauer erschließen, stehen Ihnen zudem freie Getränke und Snacks zur Verfügung. Die Ausstellung wird – bei freiem Eintritt! – als eine Mischung aus Labor und Lounge erlebbar, als Lernumgebung und Parkoase zugleich.

So weichen in dieser Ausstellung architektonisches Konzept und Szenografie stark von der gewöhnlichen Museumsarchitektur als White-Cube ab. Studio-, Labor-, Büro- und Wohnelemente wechseln einander ab. Das Museum als Commons: Das Museum wird zu einer Open-Source-Community, in der die Menschen gemeinsam kompetenter, kreativer und kenntnisreicher werden. Einerseits soll die Architektur eine Maker- und Co-Working-Space-Atmosphäre hervorrufen, andererseits sind die Wände so gestellt, dass sich organische Formen ergeben.

Das Museum wird zum Ort von BürgerInnenbildung, in dem die Aneignung von Wissen nicht nur lohnenswert ist, sondern auch belohnt wird. Denn die eigentliche Botschaft des digitalen Wandels lautet: Die Gesellschaft von morgen wird sich von einer Arbeits- zu einer Wissensgesellschaft wandeln (müssen). Daher fordern wir für das 21. Jahrhundert bezahlte BürgerInnenbildung! Wir brauchen in Zukunft kulturell kompetente BürgerInnen, um die Demokratie verteidigen zu können.

Die ZKM | Museumskommunikation hat daher innovative Konzepte des Lernens mit dem Ziel entwickelt, die spannende Welt des digitalen Codierens allen Altersgruppen zu eröffnen. Kleinkindern, Groß(-Eltern), HackerInnen, KünstlerInnen, InformatikerInnen wie Coding-AmateurlInnen bietet das weitgefächerte Vermittlungsprogramm das passende Format: Direkt in den Ausstellungsräumen können Sie das digitale Codieren zusammen mit engagierten AkteurInnen aus Karlsruhe und ZKM-MitarbeiterInnen theoretisch und praktisch erkunden – sei es in Workshops, Partys, Camps, Algoraves, Science-Slams, experimentellen Führungen oder Programmierkursen.

Peter Weibel

1 G. W. Leibniz, in einem Brief an Rudolph August, Herzog zu Braunschweig und Lüneburg, sog. Neujahrsbrief, 12. Januar 1697.

2 Deutsch: *Wort und Gegenstand*, 1980.

3 Deutsch: *Die Ordnung der Dinge*, 2003.

4 Ludwig Wittgenstein, *Tractatus logico-philosophicus*, 1921, Satz 5.6.

5 Galileo Galilei, *Il Saggiatore* (1623), Edition Nazionale, Bd. 6, Florenz 1896, S. 232.

6 Deutsch: *Die mathematische Analyse der Logik*, 2001.

Ausstellungsmニュアル

Wir leben in einer Zeit, in der sich durch die beständige Weiterentwicklung von auf Code basierten Technologien, die Produktion, Verbreitung und Aneignung von Wissen global und grundlegend ändert. Von zentraler Bedeutung für die damit einhergehenden Debatten sind die Stellung und die Funktion des Museums in der heutigen Zeit. Bei dem Wort „Museum“ denken Sie vielleicht an dessen institutionelle Aufgabe, historische Artefakte oder Kunstwerke zu sammeln, zu erhalten und auszustellen. Mit der Ausstellung *Open Codes* möchten wir jedoch eine Neudefinition des Museums für das 21. Jahrhundert formulieren und damit starre Strukturen und überholte Denkweisen aufbrechen: Unser Ziel ist es, den Anforderungen und Bedürfnissen unserer heutigen Zeit Rechnung zu tragen und sie in den Museumsraum zu integrieren. Es ist ein Versuch, sich mit den aktuellen Gegebenheiten auseinanderzusetzen und neue Perspektiven und Entwicklungslinien für die Zukunft aufzuzeigen, um damit die Welt, in der wir leben, besser zu verstehen.

Aus diesem Grund haben wir für die Ausstellung ein deinstitutionalisiertes Format entwickelt, eine Wissensplattform, die stets frei zugänglich und unserer tatsächlichen Lebens- und Arbeitsumwelt sehr ähnlich ist. *Open Codes* ist als gemeinschaftlicher Raum angelegt, als eine Umgebung, in der man zusammenkommen und sich austauschen kann. Arbeiten und Lernen werden als kollaborative Prozesse verstanden, um damit Synergien zwischen unterschiedlichen Berufen und verschiedenen Formen des Wissens und von Wissensinhalten zu schaffen. Als BesucherIn sind Sie dazu eingeladen, gemeinsam mit anderen zu arbeiten, zu produzieren und zu lernen. Die Ausstellungsgestaltung, in der sich parkähnliche Oasen der Entspannung mit Work Stations der Konzentration, Laborsituationen mit Wohnlandschaften, Büroplätze mit Spielplätzen abwechseln, soll ermöglichen, gemeinsam kompetenter, kreativer und kenntnisreicher zu werden. Somit ist *Open Codes* ein Forum, das Kollaboration und Co-Kreation ermöglicht und Sie dazu einlädt, sich an einem offenen Austausch in einer sich stets wandelnden Umgebung zu beteiligen. In der Ausstellung werden verschiedene Ansätze erprobt, um neue Formen der Begegnung und der kritischen Auseinandersetzung zu ermöglichen. Auf diese Weise lernen Sie eine neuartige Wissensumgebung kennen, einen wandelbaren Ort, der sich stets

neu konfiguriert, sich weiterentwickelt und verändert. Teil dieses Konzepts sind verschiedene Werkzeuge, mit denen man sich der Ausstellung nähern kann, und die im Folgenden vorgestellt werden:

Hashtags

Die vielen Themen, die in der Ausstellung aufgegriffen werden, sind in acht übergeordnete Bereiche gegliedert: [#Genealogie](#), [DesCodes](#), [#Codierung](#), [#MaschinellesLernen](#), [#Algorithmic Governance](#), [#Arbeit&Produktion](#), [#AlgorithmischeÖkonomie](#), [#VirtuelleRealität](#) und [#GenetischerCode](#).

Die Ausstellung *Open Codes* zeichnet sich vor allem dadurch aus, dass die Kunstwerke im Ausstellungsraum nicht thematisch gruppiert sind. Die übergeordneten Themen werden mit den oben genannten titelgebenden Hashtags dargestellt sowie mit weiteren, damit inhaltlich in Zusammenhang stehenden Hashtags näher bestimmt. Jedem Werk sind mehrere dieser Hashtags zugeordnet. So können Sie gedankliche Verbindungslinien zwischen verschiedenen Fragestellungen zu den einzelnen Themen ziehen.

Das Hashtag-System lehnt sich an die mediale Verwendung von Hashtags an und steht damit für die offenen, fließenden und dynamischen Verbindungen zwischen Themen, die für unsere vernetzte Welt zentral sind.

Das #-Zeichen wurde bis 2007 vor allem zur Auszeichnung von Zahlen verwendet, bis NutzerInnen sozialer Medien begannen, es als Metatag zu verwenden und Inhalte mit selbst erdachten Schlagwörtern zu versehen. Es handelt sich um ein dynamisches, assoziatives Klassifikationssystem, das heute in den sozialen Medien weit verbreitet ist, um Inhalte, Diskussionen und Themen bestimmten Kategorien zuzuordnen. Dadurch, dass Inhalte wie Bilder, Videos usw. mit Hashtags versehen werden, entsteht ein nicht-hierarchisches System, das an eine Wolke erinnert: Inhalten können hier die unterschiedlichsten Schlagwörter zugeordnet werden, sie fallen damit nicht unter nur eine bestimmte und abschließliche Kategorie.

Der Siegeszug des Hashtags begann vor zehn Jahren auf Twitter, als die Plattform anfang, alle ghashtagten Begriffe, die in Tweets vorkamen, mit den Suchergebnissen in der Twitter-Suche zu verlinken. Als 2010 Instagram auf den Markt kam, wurde der Hashtag zur Lingua franca, um Content auf beiden Plattformen mit einem Schlagwort zu versehen. In jüngster Zeit findet man den Hashtag auf jeder beliebigen Onlineplattform, und er verändert die Art und Weise, wie wir online nach Informationen suchen und auf sie zugreifen.

Broschüre

Die Broschüre, die Sie in den Händen halten, dient als eines der Hauptwerkzeuge, mit dem man sich den physischen Ausstellungsraum erschließen kann. Sie enthält neben dem, die Ausstellung und ihre Komponenten erläuternden Text den einleitenden Text des Kurators Peter Weibel, Texte, die die acht Themenbereiche beschreiben, eine Werkliste sowie einen Text zum Thema Signal- und Maschinencodes von Franz Pichler.

Werkbeschreibungen hingegen finden sich auf Schildern direkt neben den Werken im Ausstellungsraum und auf der Ausstellungswebsite im Internet.

Wenn Sie sich mit einem bestimmten Themenbereich (titelgebenden Hashtag) der Ausstellung ausführlich beschäftigen möchten, können Sie das zusätzliche Faltblatt mit den Ausstellungsplänen zu Hilfe nehmen. Für jeden Themenbereich bietet es einen eigenen Ausstellungsplan sowie eine komplette Übersicht über die gesamte Ausstellung. Mit dieser Methode gewinnen Sie schnell einen Überblick, wo im Ausstellungsraum die entsprechenden Werke platziert sind.

App

Sie können sich Ihren Weg durch die Ausstellung auch mithilfe der App *experience_zkm* erschließen. Die App kann sowohl für Android- wie auch iOS-Geräte heruntergeladen werden.

Mit der App können Sie nach einem der titelgebenden Hashtags suchen und über den Audioguide Informationen zum jeweiligen Bereich der Ausstellung hören. Darüberhinaus werden Ihnen die sich in diesem Bereich befindlichen Werke im Ausstellungsplan angezeigt. Somit liefert die App nicht nur weiterführende Inhalte, sondern kann auch zur Navigation durch *Open Codes* genutzt werden.

Infopoints und Website

Zur Ausstellung wurde eine umfassende Onlinesammlung von Informationen zusammengetragen, die Werktexte, Bildmaterial sowie ausführliches Hintergrundmaterial – Zeitungsartikel, Videos, weiterführende Literatur – umfasst. Neben der Broschüre und der App können Sie über die Website des ZKM (<https://open-codes.zkm.de>) diese Informationen abrufen. Darüberhinaus finden Sie hier weitere künstlerische Arbeiten, die exklusiv online präsentiert werden. Auf die Website kann dabei nicht nur über eigene Geräte zugegriffen werden, sondern auch über die Infopoints in der Ausstellung.

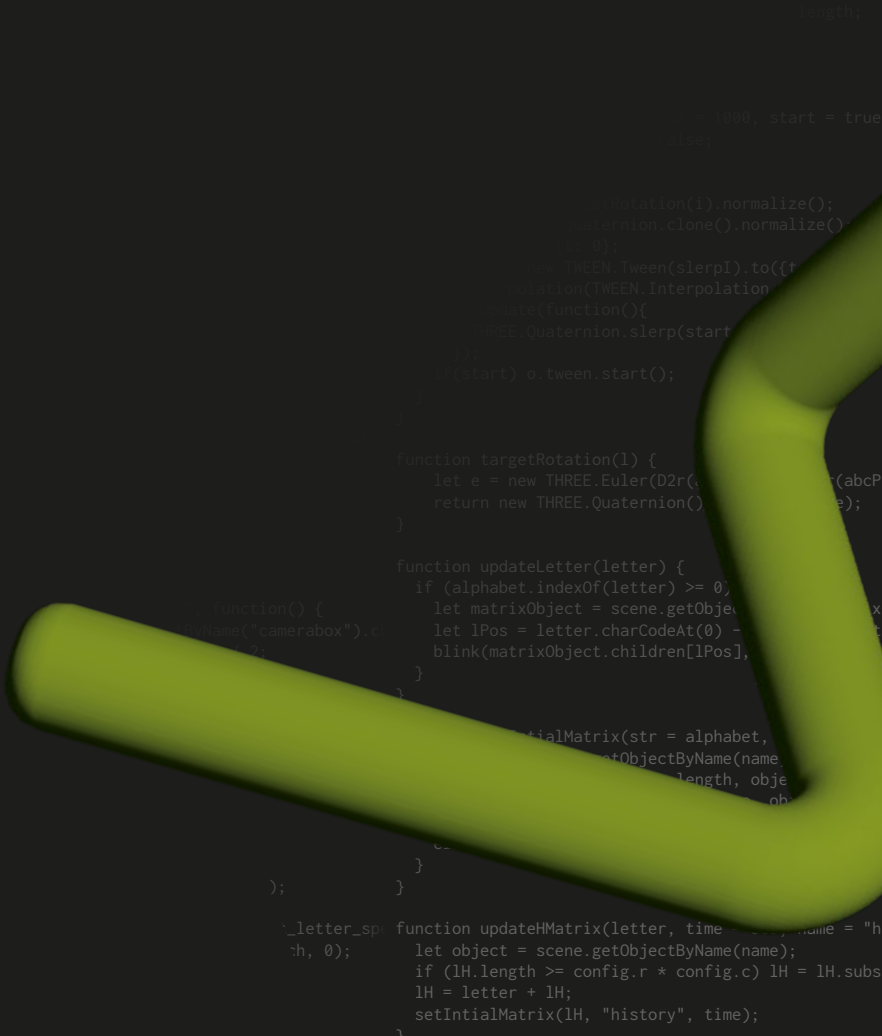
Work Stations

Über die ganze Ausstellung verteilt finden Sie Tische und Arbeitsplätze vor, an denen Sie jederzeit Platz nehmen können, um zu lesen, etwas aufzuschreiben, zu programmieren oder das zu tun, worauf Sie durch den Ausstellungsbesuch Lust bekommen haben. Diese Arbeitsplätze sollen es Ihnen ermöglichen, spontan Ihren Ideen freien Lauf zu lassen, sodass Sie nicht erst lange nach einem Ort suchen müssen, an dem Sie Ihre Gedanken festhalten können. In diesen Bereichen können Sie allein oder auch in der Gruppe arbeiten. So wurden einige Arbeitsplätze als Co-Working-Bereiche konzipiert, um den Austausch zwischen den unterschiedlichsten Menschen und Interessen zu ermöglichen. Andere Work Stations hingegen erlauben konzentriertes Arbeiten in einer ruhigeren Atmosphäre. Zur Verstärkung Ihrer Lust auf Lernen, zur Belohnung Ihrer Bildungskompetenz stehen auch Automaten mit kostenlosen Getränken, Snacks, Äpfeln etc. zur Verfügung.

Der Arbeitsbereich in Lichthof 8 ist so konzipiert, dass dort auch verschiedene öffentliche Veranstaltungen stattfinden können. Es werden Workshops, Vorträge und Diskussionsrunden veranstaltet und alle sind dazu eingeladen, teilzunehmen und sich daran zu beteiligen. Entscheidend ist dabei, dass dies ein Raum ist, den *Sie* mit Inhalt füllen können. Wir stellen die Infrastruktur bereit – *Sie* können den Raum mit Ideen füllen. Wenn Sie selbst eine Veranstaltung anbieten oder organisieren möchten, besuchen Sie einfach unsere Website und buchen Sie den Raum an Ihrem Wunschtermin.

#GenealogieDesCodes

#Binär
#Computing
#Zahlensystem
#Babel



Die Geschichte des maschinellen Rechnens beginnt nicht erst mit dem Aufkommen des Personal Computers oder seinen direkten Vorläufern im 20. Jahrhundert. Um den Grundlagen, auf denen heutige Berechnungsverfahren beruhen, auf die Spur zu kommen, muss man zumindest bis ins Mittelalter zurückgehen.

Ramon Llull (1232–1316), ein mallorquinischer Gelehrter, versuchte ein System zu entwickeln, mit dem man grundlegende theologische und philosophische Fragen lösen konnte. Mithilfe dynamischer Diagramme wollte er alle existierenden Kombinationsmöglichkeiten verschiedener Konzepte finden und untersuchen. Diese Vorgehensweise, seine sogenannte *Ars Magna* oder „Große Kunst“, beschrieb er in seinem gleichnamigen Werk *Ars magna* (1274–1308).¹ Angeregt durch Llull formulierte Gottfried Wilhelm Leibniz (1646–1716) 1666 in seiner Schrift *Dissertatio de arte combinatoria* [Dissertation über die Kunst der Kombinatorik] Parallelen zwischen Logik und Metaphysik.²

1679 beschrieb Leibniz in einem unveröffentlichten Brief ein binäres System („*Arithmetica binaria*“ bzw. „*dyadica*“), in dem nur die Zahlen 0 und 1 verwendet werden (#Binär). Im selben Jahr erdachte er die erste Rechenmaschine (#Computing), die mit dem Binärsystem arbeitet. Die Beschreibung wurde nie veröffentlicht und die Maschine zu seinen Lebzeiten nicht gebaut.³

Zwei Jahrhunderte später arbeitete Charles Babbage an seiner *Difference Engine* (Differenzmaschine), auf welche die *Analytical Engine* (analytische Maschine) folgte. Aufgrund fehlender finanzieller Mittel wurde keiner der beiden Entwürfe unter seiner Leitung je vollständig realisiert. Die *Analytical Engine* wäre der erste universell einsetzbare Computer gewesen, aber immer noch ein mechanischer. „Die Grenzen der *Arithmetik* wurden jedoch in dem Augenblick überschritten, als die Idee aufkam, die Karten zu verwenden; und die *Analytical Engine* steht nicht auf einer Stufe mit bloßen ‚Rechenmaschinen‘“, schrieb Ada Lovelace (1815–1852), die heute als die erste programmierende Person gilt, in ihren Anmerkungen zu Babbages Rechenautomat. Dieser frühe Apparat arbeitete mit einem dezimalen #Zahlensystem. Für heutige Computer wird ausschließlich ein binäres Zahlensystem verwendet, wie es zuerst von Leibniz benutzt und dann von George Boole (1815–1864) wieder eingeführt wurde.

Boole war der Erste, welcher der Logik eine mathematische Form gab: Mit seinem Buch *Die mathematische Analyse der Logik* (1847) führte er die Boolesche Algebra ein.⁵ Das Binärsystem Booles basiert auf drei grundlegenden Verknüpfungen, die als logische Operatoren verwendet werden: UND, ODER und NICHT.⁶

Benutzt wurde dieses System aber erst, als „Claude Shannon (1916–2001) 1937 in der wohl folgenreichsten Masterarbeit, die je geschrieben wurde, bewies, dass sich durch einfache Telegrafengeräte und ihre verschiedenen Verbindungen untereinander die gesamte Boolesche Algebra umsetzen lässt.“⁷

Ebenfalls 1937 realisierte Alan Turing (1912–1954) ein Schaltnetz auf Grundlage der Booleschen Algebra und formulierte eine Berechenbarkeitstheorie in seinem Aufsatz über das „Entscheidungsproblem“.⁸ Die Abhandlung hatte er bereits im Jahr zuvor geschrieben, während er an seiner berühmten Turing-Maschine arbeitete. Dabei handelt es sich nicht um einen Computer, der physisch existiert, sondern um ein mathematisches Rechenmodell. Mit seinem an der Booleschen Logik orientierten Schaltnetz versuchte Turing „der logischen Gestalt einer Turing-Maschine mit einem Netzwerk aus relaisgesteuerten Schaltern eine konkrete Form zu geben“⁹. Kurz darauf, ab den 1940er-Jahren, entstanden mit dem Aufkommen elektronischer Computer unterschiedliche Programmiersprachen. Zu den ersten zählte Assembler (asm). Diese einfache Programmiersprache kann in einem Schritt in ausführbaren Maschinencode übersetzt werden, da sich diese Sprache und die Maschinensprache stark ähneln. Ab den 1950er-Jahren begannen höhere Programmiersprachen ihre „niederen“ Vorgänger zu ersetzen. Dutzende Programmiersprachen wurden seitdem geschrieben und entwickelt, angefangen mit ALGOL (ALGOritmic Language), gefolgt von Fortran, Pascal, C++, Java und Python, um nur einige zu nennen. „Dieser Turm von Babel reicht von einfachen Befehls-Codes, deren linguistische Extension noch eine Hardwarekonfiguration ist, über Assembler, dessen Extension genau jene Befehls-Codes sind, bis zu sogenannten Hochsprachen, deren Extension nach allen möglichen Umwegen über Interpreter, Compiler und Linker wiederum Assembler heißt.“¹⁰ (#Babel)

All diese Sprachen basieren auf einem binären Zahlensystem, einer Abfolge von „an“ und „aus“, wodurch der Strom in den Schaltkreisen fließen kann oder unterbrochen wird. Obwohl ihre Grundbestandteile so einfach sind, können Programmiersprachen äußerst komplexe Vorgänge beschreiben. Alle oben erwähnten Rechenmaschinen funktionieren mit binärem Code – bis auf die Maschinen von Babbage, für die das Dezimalsystem genutzt wurde. Durch Shannons Arbeit und den Einsatz von Transistoren wurde das Binärsystem im Bereich des maschinengestützten Rechnens allgegenwärtig.

Neben Algorithmen (siehe #Algorithmus Themenbereich #Codierung) und Berechnungen kann alles, was decodiert werden kann, durch binären Code beschrieben werden. Das beste Beispiel dafür ist wohl der ASCII-Code (American Standard Code for Information Interchange), mit dem Buchstaben auf einem Computerbildschirm dargestellt werden. Er wurde ab 1960 auf der Basis von Codes aus dem Bereich der Telegrafie entwickelt.

Aktuelle Computer sind womöglich nicht leistungsfähig genug, um die Menge und Komplexität zukünftiger Berechnungsaufgaben zu bewältigen. Moderne Computer haben sich jedoch erstaunlich schnell entwickelt, wenn man beispielsweise die

Entwicklung ihrer Leistungsfähigkeit mit der von Autos vergleicht. Wenn Autos sich seit 1971 im selben Tempo wie Computerchips verbessert hätten, dann könnten die aktuellen Modelle eine Höchstgeschwindigkeit von rund 680 Millionen km/h erreichen.¹¹

„Quantencomputing“ könnte die Antwort auf die aktuelle und scheinbar unausweichliche Bedarfssteigerung sein. Ein Quantencomputer führt logische Operationen auf atomarer Ebene aus. Und Atome können mehr leisten als Bits: Da sie den Zustand von 0 und 1 gleichzeitig annehmen und damit zwei Berechnungen zur selben Zeit ausführen können, sind Quantenbits (oder Qubits) leistungsfähiger als die klassischen Bits.¹²

„Wie lange kann das maschinengestützte Rechnen noch im Universum bestehen? Aktuelle Beobachtungsergebnisse legen nahe, dass sich das Universum bis in alle Ewigkeit ausdehnen wird. Mit seiner Ausdehnung wird sich die Menge der ausgeführten Operationen und die Menge der Bits, die bis zum Ende des Horizonts verfügbar sind, weiter wachsen.“¹³

1 Raimundus Llullus, *Opera*, 2 Bde., Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1996, S. 228–663. Siehe auch Anthony Bonner, *The Art and Logic of Ramon Llull: A User's Guide*, Brill, Leiden, Boston, 2007.

2 Vgl. Ana H. Maróstica, „Ars Combinatoria and Time: Llull, Leibniz and Peirce“, in: *Studia Lulliana*, Bd. 32, 1992, S. 105–134, hier S. 111.

3 Vgl. Hermann J. Greve, „Entdeckung der binären Welt“, in: *Herrn von Leibniz' Rechnung mit Null und Eins*, Siemens Aktiengesellschaft, Berlin, München, 1966, S. 21–31.

4 L. F. Menabrea, *Sketch of the Analytical Engine Invented by Charles Babbage*, übersetzt aus dem Französischen und mit Anmerkungen von Ada Lovelace, Richard und John E. Taylor, London, 1843, S. 696f. Übersetzt aus dem Englischen von PK.

5 Vgl. George Boole, *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, Macmillan, Cambridge, 1847.

6 Vgl. Paul J. Nahin, *The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age*, Princeton University Press, Princeton (NJ), Oxford, 2013.

7 Friedrich Kittler, „There Is No Software“, in: *Stanford Literature Review*, Bd. 9, Nr. 1, Frühjahr 1992, S. 81–90, hier S. 88. Übersetzt aus dem Englischen von PK.

8 Vgl. Alan Turing, „On Computable Numbers, with an Application to the Entscheidungsproblem“, in: *Proceedings of the London Mathematical Society*, Reihe 2, Band 42, Nr. 1, Januar 1937, S. 230–265.

9 Vgl. Andrew Hodges, *Alan Turing: The Enigma*, Princeton University Press, Princeton (NJ), Oxford, 2014, S. 177.

10 Kittler 1992, S. 82.

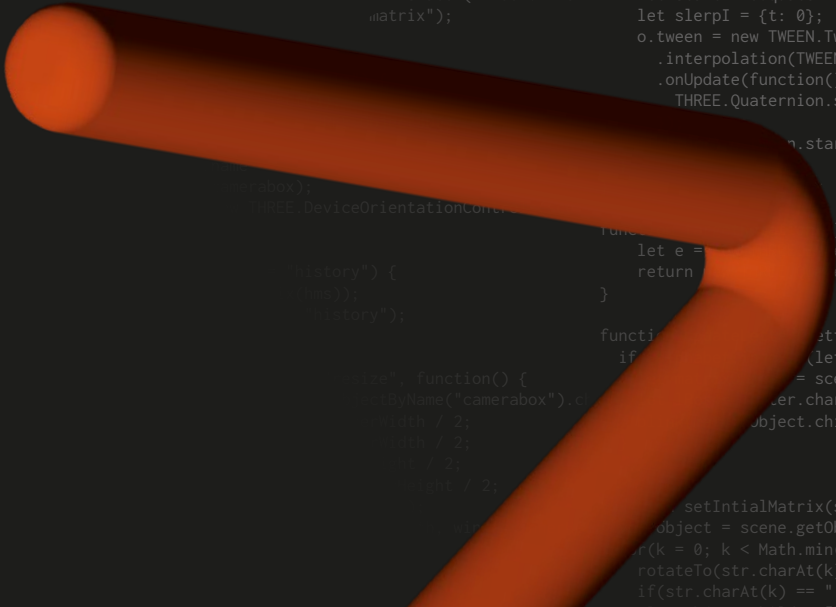
11 Vgl. Tim Cross, „Vanishing point: The rise of the invisible computer“, in: *The Guardian*, 26.01.2017, online: <https://www.theguardian.com/technology/2017/jan/26/vanishing-point-rise-invisible-computer>, 13.09.2017.

12 Vgl. Seth Lloyd, *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*, Vintage Books, New York, 2007, S. 136–139.

13 Ebd., S. 206. Übersetzt aus dem Englischen von PK.

#Codierung

#Morsecode
#Klangprogrammierung
#Algorithmus
#Software
#Hardware
#Interface
#Decodierung



Von genetischem Code (siehe Themenbereich #GenetischerCode) bis zur Notation von Musik, von Kommunikationssystemen für sinnesbehinderte Menschen, wie beispielsweise der Gebärdensprache, bis zum #Morsecode, von ethischen Richtlinien bis zu sozialen Codes – der Begriff „Code“ mag oberflächlich betrachtet wiedererkennbare und vertraute Vorgänge beschreiben, aber was bedeutet er in Hinblick auf #Programmieren (siehe Themenbereich #Arbeit&Produktion) und Datenverarbeitung?

Das *Lexikon der Informatik* definiert Code als „eine Regel, mit der eine Nachricht von einer symbolischen Form (dem Quellalphabet) in eine andere (das Zielalphabet) überführt wird“.¹ Demzufolge kann Code also als ein Regelsatz beschrieben werden, „der das, was eingegeben wird – den Input – von einem Zustand in einen anderen überführt und dann eine Operation ausführt“.² Diese Funktionsweise stellt ein Hauptmerkmal des Codes dar: Er ist sowohl lesbar als auch ausführbar; er ist gleichzeitig Medium und Anweisung. Diese grundlegende Eigenschaft macht den Unterschied zwischen Code und herkömmlichen Sprachen aus: Zweitere können gelesen oder geschrieben werden, aber alleine dadurch bewirken sie keine Veränderung. In diesem Sinne ist Computercode „die erste Sprache, die tatsächlich tut, was sie sagt – er ist eine Maschine, durch die Bedeutung in die Tat umgesetzt wird.“³

Ein weiteres entscheidendes Merkmal des Computercodes ist seine trügerische Unsichtbarkeit. Code bleibt für gewöhnlich im Verborgenen; er selbst ist immateriell und normalerweise unsichtbarer Bestandteil einer Maschine, hat aber sichtbare, konkrete und greifbare Auswirkungen auf die Welt. So wären bei einem programmierten Klangwerk beispielsweise die verschiedenen Klänge oder die Komposition der Output bzw. das Ergebnis einer einzelnen Zeile oder vieler Zeilen Codes (#Klangprogrammierung).

Ähnlich wie der Begriff Code wird auch das Wort #Algorithmus oft mit Datenverarbeitung und Programmierung in Verbindung gebracht, obwohl die Definition eines Algorithmus, als einer Abfolge von Handlungsanweisungen, in vielen Bereichen Anwendung finden könnte. Ein Algorithmus ist ein Satz von Regeln, mit denen definiert wird, wie eine Problemstellung zu lösen oder ein Vorgang auszuführen ist. In diesem Sinne könnten ein Kochrezept oder eine Bauanleitung ebenfalls als Algorithmus betrachtet werden. Im Computerbereich werden solche Regelsätze bzw. die Abfolge einzelner Schritte dazu benutzt, Daten zu verarbeiten und, wie bereits beschrieben, einen Output zu produzieren.

Algorithmen und Code machen auch den unsichtbaren Teil eines Computersystems aus, der für gewöhnlich unter dem Begriff #Software zusammengefasst wird, „einer allgemeinen Bezeichnung für die Komponenten eines Computersystems, die

immateriell und nicht physisch greifbar sind.“⁴ Im Gegensatz dazu, versteht man unter #Hardware die technischen Bauteile, aus denen sich ein Computer zusammensetzt, darunter beispielsweise das Mainboard. Damit zwischen Software und Hardware Informationen ausgetauscht werden können, ist ein drittes Element notwendig: das #Interface. Es kann auch als Verbindung zwischen Software, Hardware und dem Menschen fungieren. Um zu verstehen, wie dieser Austausch funktioniert, muss man nur an den Schalter zum Ein- und Ausschalten eines Geräts denken: Der Schalter ist sozusagen das Interface zwischen dem Menschen und den elektrischen Schaltkreisen des Geräts. Wenn man ihn betätigt, schaltet sich das Gerät ein oder aus.

Selbst bei weitverbreiteten, einfachen Anwendungen, wie beispielsweise dem Verschicken einer SMS, führt der Code eine Vielzahl algorithmischer Vorgänge aus. Im Computerbereich steht das Codieren (#Codierung) für einen Vorgang, bei dem eine Zeichenfolge in ein bestimmtes Format umgewandelt wird, mit dem Daten zweckmäßig übertragen oder gespeichert werden können. Um codierte Daten wieder in die ursprüngliche Zeichenfolge umzuwandeln, muss der umgekehrte Prozess angewendet werden, die #Decodierung. Beides sind gängige Verfahren, die zur Übertragung, Vernetzung und Speicherung von Daten sowie vor allem im Zusammenhang mit drahtlosen Kommunikationssystemen eingesetzt werden. Mit diesen und auch anderen Prozessen, die auf Computercode basieren, ist es heute möglich, eine Vielzahl unterschiedlicher Vorgänge in Sekundenschnelle zu verarbeiten und zu steuern und so neue Handlungsfelder im gesellschaftlichen, wirtschaftlichen und kulturellen Bereich zu erschließen und zu gestalten.

1 Andrew Butterfield und Gerard Ekembe Ngondi, *A Dictionary of Computer Science*, 7. Aufl., Oxford University Press, Oxford, 2016, S. 93. Übersetzt aus dem Englischen von PK.

2 Rob Kitchin und Martin Dodge, *Code/Space: Software and Everyday Life*, The MIT Press, Cambridge (MA), London, 2011, S. 25. Übersetzt aus dem Englischen von PK.

3 Alexander R. Galloway, *Protocol: How Control Exists after Decentralization*, The MIT Press, Cambridge (MA), London, 2004, S. 166. Übersetzt aus dem Englischen von PK.

4 Susan M. Hockey, *A Dictionary of Computing*, 2. Aufl., Oxford University Press, Oxford, 1986, S. 352. Übersetzt aus dem Englischen von PK.

#MaschinellesLernen

#KünstlicheIntelligenz

#Kybernetik

#Mustererkennung

#AutonomeSysteme

#SelbstfahrendeAutos

#Drohnen

#Roboter

```
function  
renderer = ...  
renderer.domElement  
renderer.setClearColor(0x0, ...  
renderer.setPixelRatio(window.devic  
renderer.setSize(window.innerWidth, window  
document.body.appendChild(renderer.domElem  
document.addEventListener("key  
scene = new THREE.Scene()  
scene.add(new THREE.Amb
```

```
camera = new THREE.  
camera.position.set(0, 0, 10);  
camera.lookAt(ya
```

```
low.innerWidth / 4));  
translateX(window.innerWidth  
"trix");
```

```
renderer.render(scene, camera);  
// ...  
// ...
```

```
function rotateTo(i = "a", time = 1000, ...  
if (i == " ") o.visible = false;  
else {  
o.tween.stop();  
let target = targetRotation(i).normalize();  
let start = o.quaternion.clone().normalize  
let slerpI = {t: 0};  
o.tween = new TWEEN.Tween(slerpI).to({t  
.interpolation(TWEEN.Interpolation.P  
.onUpdate(function(){  
THREE.Quaternion.slerp(start, t  
});  
if(start) o.tween.start();  
}  
}
```

```
function targetRotation(l) {  
let e = new THREE.Euler(l  
return new THREE.Quater
```

```
function updateLetter(i) {  
let index = ...  
let object = ...
```

```
function rotateTo(i = "a", time = 1000, ...  
if (i == " ") o.visible = false;  
else {  
o.tween.stop();  
let target = targetRotation(i).normalize();  
let start = o.quaternion.clone().normalize  
let slerpI = {t: 0};  
o.tween = new TWEEN.Tween(slerpI).to({t  
.interpolation(TWEEN.Interpolation.P  
.onUpdate(function(){  
THREE.Quaternion.slerp(start, t  
});  
if(start) o.tween.start();  
}  
}
```

```
function targetRotation(l) {  
let e = new THREE.Euler(l  
return new THREE.Quater
```

```
function updateLetter(i) {  
let index = ...  
let object = ...
```



In der Informatik bezeichnet #KünstlicheIntelligenz (KI) die Operationen intelligenter Agenten durch die Anwendung mechanischer oder „formaler“ Logik. Die KI gründet auf der Ansicht, dass eine Maschine die menschliche Intelligenz simulieren kann. In seiner Computertheorie behauptete Alan Turing (1912–1954), dass es möglich sei, logische Vorgänge durch die Veränderung einfacher Größen wie 0 und 1 darstellen zu können. Turing nahm an, dass Denken, als eine spezifische Abfolge mechanischer Operationen, auf der Grundlage des Kausalitätsprinzips formalisiert werden kann, oder – anders ausgedrückt – als diskrete Abfolge logischer Schritte, die auf einem Regelsatz (#Algorithmus, siehe Themenbereich #Codierung) basieren.¹ Diese Herangehensweise, d. h. der klassische symbolische Ansatz der KI, betrachtet maschinenbasierte Wissensgenerierung als eine regelbasierte Manipulation formaler Zeichen durch eine übergeordnete Kontrollinstanz. Dieser Ansatz war der Versuch, Wissen über die Welt in der formalen Sprache der Mathematik zu codieren. Bewährt hatte sich dieser Ansatz für sogenannte Expertensysteme, die komplexe Aufgabenstellungen, wie etwa Diagnostik in der Medizin, bewältigen oder auch Plan- und Entwurfsaufgaben auf dem Niveau menschlicher Experten ausführen konnten. Aber, wie sich herausstellte, waren solche Systeme schwierig zu programmieren, da zuweilen ein einfacher Fehler zum Versagen des gesamten Systems führen konnte. Vor allem aber waren diese Systeme nicht lernfähig.² Um 1980 war klar, dass die bloße Simulation des Denkens nicht zu tatsächlichem Verstehen führt, dass also die syntaktische Symbolmanipulation zur Erkenntnisgewinnung nicht ausreicht, und so wurde dieser Ansatz nicht weiterverfolgt.³

Eine offenerere und flexiblere Herangehensweise an die maschinelle Kognition kam aus dem Bereich der Neurowissenschaften und der #Kybernetik, in welchem künstliche Intelligenz nicht von Regeln und Repräsentationen ausgehend gedacht wurde, sondern als dynamisches System. In der bahnbrechenden Forschungsarbeit von Warren S. McCulloch (1898–1969) und Walter Pitts (1923–1969) wurde das Gehirn erstmals als Rechenmaschine betrachtet.⁴ Ausgehend von Donald O. Hebb's (1904–1985) Arbeit zum assoziativen Lernen, das auf durch neuronale Signalübertragung entstehenden synaptischen Verknüpfungen basiert⁵, entwickelte Frank Rosenblatt (1928–1971) die Grundlage für das maschinelle Lernen.⁶ #MaschinellesLernen ist ein Teilgebiet der KI, das Berechnungsmethoden erforscht, durch die Programme zur Verarbeitung von Daten ihre internen Parameter automatisch, d. h. ohne Modifikation der Algorithmen durch ein Individuum, ändern und anpassen können. Die algorithmische Struktur ist als künstliches neuronales Netzwerk angelegt, in dem Folgerungsprozesse von Tausenden von Neuronen ausgeführt werden, die in Hunderten komplex miteinander vernetzten Ebenen organisiert sind und Kausalrelationen aufschlüsseln. Künstliche neuronale

Netze basieren auf der Modellierung adaptiver Systeme, die sich durch das Erfassen von Umweltdaten, die in das System eingespeist werden, kontinuierlich weiterentwickeln.⁷ Entscheidend ist hierbei, dass es sich bei den errechneten Resultaten um Näherungswerte handelt, folglich um eine statistische Einschätzung des wahrscheinlichsten Ergebnisses.

Seit 2006 wurden auf dem Gebiet des maschinellen Lernens immense Fortschritte erzielt. Die Gründe dafür liegen in der stetig wachsenden Rechenleistung und der damit einhergehenden gewaltigen Zunahme von Verfahren zur Datensammlung sowie dem physischen Ausbau der informationstechnischen Infrastruktur.⁸ In der Praxis werden Algorithmen aus dem Bereich des maschinellen Lernens vor allem zur #Mustererkennung eingesetzt, zur Objekterkennung und -detektion, die insbesondere für #AutonomeSysteme wie #SelbstfahrendeAutos, #Drohnen und #Roboter von Bedeutung sind. Im Grunde genommen werden also mit dem maschinellen Lernen Denkprozesse neu definiert und damit auch zahlreiche ethische und rechtliche Fragen in Hinblick auf automatisierte Entscheidungsfindung, algorithmische Voreingenommenheit, Haftung und Verantwortlichkeit aufgeworfen.

1 Vgl. Alan Turing, „On Computable Numbers, with an Application to the Entscheidungsproblem“, in: *Proceedings of the London Mathematical Society*, Reihe 2, Bd. 42, Nr. 1, Januar 1937, S. 230–265.

2 Vgl. David Davenport, „The Two (Computational) Faces of AI“, in: Vincent C. Müller (Hg.), *Philosophy and Theory of Artificial Intelligence*, Studies in Applied Philosophy, Epistemology and Rational Ethics, Bd. 5, Springer, Heidelberg, 2013, S. 43–58, hier S. 44.

3 Vgl. John R. Searle, „Minds, Brains, and Programs“, in: *Behavioral and Brain Sciences*, Bd. 3, Nr. 3, September 1980, S. 417–424.

4 Warren S. McCulloch und Walter Pitts, „A Logical Calculus of the Ideas Immanent in Nervous Activity“, in: *Bulletin of Mathematical Biophysics*, Bd. 5, Nr. 4, Dezember 1943, S. 115–133.

5 Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, Chapman and Hall, London, 1949.

6 Frank Rosenblatt, „The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain“, in: *Psychological Review*, Bd. 65, Nr. 6, 1958, S. 386–408.

7 Vgl. Yann LeCun, Yoshua Bengio und Geoffrey Hinton, „Deep Learning“, in: *Nature*, Bd. 521, Mai 2015, S. 436–444.

8 Vgl. Geoffrey E. Hinton, Simon Osindero und Yee-Whye Teh, „A Fast Learning Algorithm for Deep Belief Nets“, in: *Neural Computation*, Bd. 18, Nr. 7, Juli 2006, S. 1527–1554.

#AlgorithmicGovernance

#BigData

#QuantifiziertesSelbst



```
... Xpos = 0, Ypos = ...  
... Object3D();  
... Mesh.clone();  
... (y3(scale, scale, scale));  
... (xpos).translateY(Ypos);  
... container.children[0].length;  
... new TWEEN.Tween();  
...  
... rotateTo(i = "a", time = 1000, start = ...  
... " ") o.visible = false;  
... stop();  
... = targetRotation(i).normalize();  
... .quaternion.clone().normalize();  
... 0);  
... TWEEN.Tween(slerpI).to({t: 1});  
... TWEEN.Interpolation.Bezier);  
... (start, target, o);  
... });  
... if(start) }  
... }  
... functionControls(scene);  
... function targetRotation(l, ...  
... let e = new THREE.Euler(..., D2r...  
... return new THREE.Quatern... Euler(...  
... }  
... function updateLetter(lette...  
... if (alphabet.indexOf(let...  
... let matrixObject = sce... Name("ma...  
... let lPos = letter.char... a".charC...  
... blink(matrixObject.ch... 50, 500);  
... }  
... }  
... function setInitialMatrix... st, name = ...  
... var object = scene.ge... name);  
... for(k = 0; k < Math.n... object.ch...  
... rotateTo(str.charA... ue, object...  
... if(str.charAt(k) =... children[k]...  
... else object.child... = true;  
... }  
... }  
... function updateHMatrix... e = 500, nam...
```

Unter *Governance* versteht man Steuerungs- und Regelungsprozesse – also die Art und Weise wie Normen, Gesetze und Maßnahmen strukturiert, aufrechterhalten und in einer Rechenschaftspflicht verankert werden, sei es durch den Staat, die Gesellschaft oder die freie Wirtschaft. Im Wesentlichen bezieht sich *Governance* auf die konkrete Vorgehensweise, mit der Gesellschaften organisiert werden, also auf die Logik oder Sprache von Steuerungsprozessen. *Governance* schließt somit auch die Machtausübung über jemanden oder etwas mit ein.¹

#AlgorithmicGovernance bezieht sich auf die Untersuchung expliziter und impliziter Regeln, mit denen die Lebenswelt durch Algorithmen (siehe **#Algorithmus** im Themenbereich **#Codierung**) organisiert wird. Sie bezieht sich auf eine Ausprägung von *Soft Power*, einer „sanften Macht“, mit der Gewohnheiten durchbrochen und Handlungspotenziale neu ausgerichtet werden.

#AlgorithmicGovernance ist die Wirkmacht eines spezifischen Verhaltens, das erzielt werden soll; sie ist eine Kraft, die wirksam ist, bevor sie überhaupt erst sichtbar wird.² In diesem Sinne ermöglicht **#AlgorithmicGovernance** eine grundlegend neue Herangehensweise an die Steuerung aller Bereiche des Lebens, sowohl in der Sphäre der Gesellschaft, der Politik, der Wirtschaft oder auch im ökologischen Bereich. Damit geht notwendigerweise die Frage einher, wie algorithmische Verarbeitungsprozesse gesteuert und gesetzlich reguliert werden sollten.

Die Grundlage aktueller Steuerungs- und Regelungsprozesse bilden Verfahren zur Sammlung und Analyse von Daten, durch die Wertzuschreibungen vorgenommen werden. In den letzten zehn Jahren hat die Menge der Daten, die in Echtzeit erfasst und ausgewertet werden kann, explosionsartig zugenommen. Unsere Umwelt wird in zunehmendem Maße in Code umgewandelt (siehe Themenbereich **#Codierung**), somit maschinenlesbar und durch die ungeheure Zahl miteinander verbundener Endgeräte und Sensoren eindeutig zuordnungs- und identifizierbar gemacht. Unser tägliches Leben wird mehr und mehr von digitalen Geräten gestaltet und durch eine computerbasierte Infrastruktur erleichtert. **#BigData** zielt u. a. darauf ab, die Gesellschaft als Ganzes zu erfassen, die gesamte Bevölkerung und ihre Aktivitäten.³ Hinter all den Anstrengungen, die zur Datensammlung und der Quantifizierung des Selbst (**#QuantifiziertesSelbst**) unternommen werden, steht das Ziel, avancierte statistische Modelle zu generieren, mit denen das Leben der Menschen beschrieben, simuliert und vorausgesagt werden kann. Um all diese Daten zusammenführen zu können, ist die Art und Weise, mit der Informationen zueinander in Beziehung gesetzt werden, entscheidend: Der Schlüssel liegt in der Verarbeitung von Daten durch statistische Analysen und Algorithmen aus dem Bereich des maschinellen Lernens (siehe Themenbereich **#MaschinellesLernen**), die Muster erkennen und zwischen den einzelnen Daten Beziehungen herstellen.

Korrelationen von Daten werden zu Quellen der Erkenntnis bzw. zu Information.

Als Folge daraus scheint sich *Governance* in ein Ringen darum verwandelt zu haben, *wie* Daten ausgewertet werden und *von wem*. Im Wesentlichen lassen sich durch die Korrelation von Daten Profile von Individuen oder Gruppen erstellen, um sogenanntes normales Verhalten zu definieren und davon abweichende Verhaltensweisen zu bestimmen. Individuen werden so zu *Dividuen* – numerischen Datenkörpern, die aus gesammelten Informationen bestehen.⁴ Ausgehend von diesen Profilen setzen Regierungen und Unternehmen ihre Ziele um. Während Letztere Strategien zur Kapitalbildung anwenden, die große Profite abwerfen, ist für Erstere der Schutz des Staates von Belang. Auf der einen Seite stehen die Unternehmen, die durch zunehmend invasivere Vorgehensweisen Nutzerprofile generieren und versuchen, das Kundenverhalten durch Mikromarketing immer weiter personalisiert zu kontrollieren. Auf der anderen Seite steht der Staat, der mit neuen Technologien Informationen sammelt, die zur Verhinderung von Straftaten beitragen bzw. durch Mikrotargeting das Wahlverhalten der BürgerInnen beeinflussen sollen. In beiden Fällen werden leistungsfähige Algorithmen zusammen mit prädikativen Analyseverfahren eingesetzt, um die Voraussetzungen zukünftiger Ereignisse zu ermitteln. Kontrolle wird fast unmerklich ausgeübt, sodass der Eindruck entsteht, das *Dividuum* würde autonom agieren, wobei es jedoch nicht über die Fähigkeit verfügt, Entscheidungen aus freien Stücken zu treffen.

1 Vgl. Isabell Lorey, *States of Insecurity: Government of the Precarious*, Verso Futures, Verso, London, New York, 2015, S. 23ff.

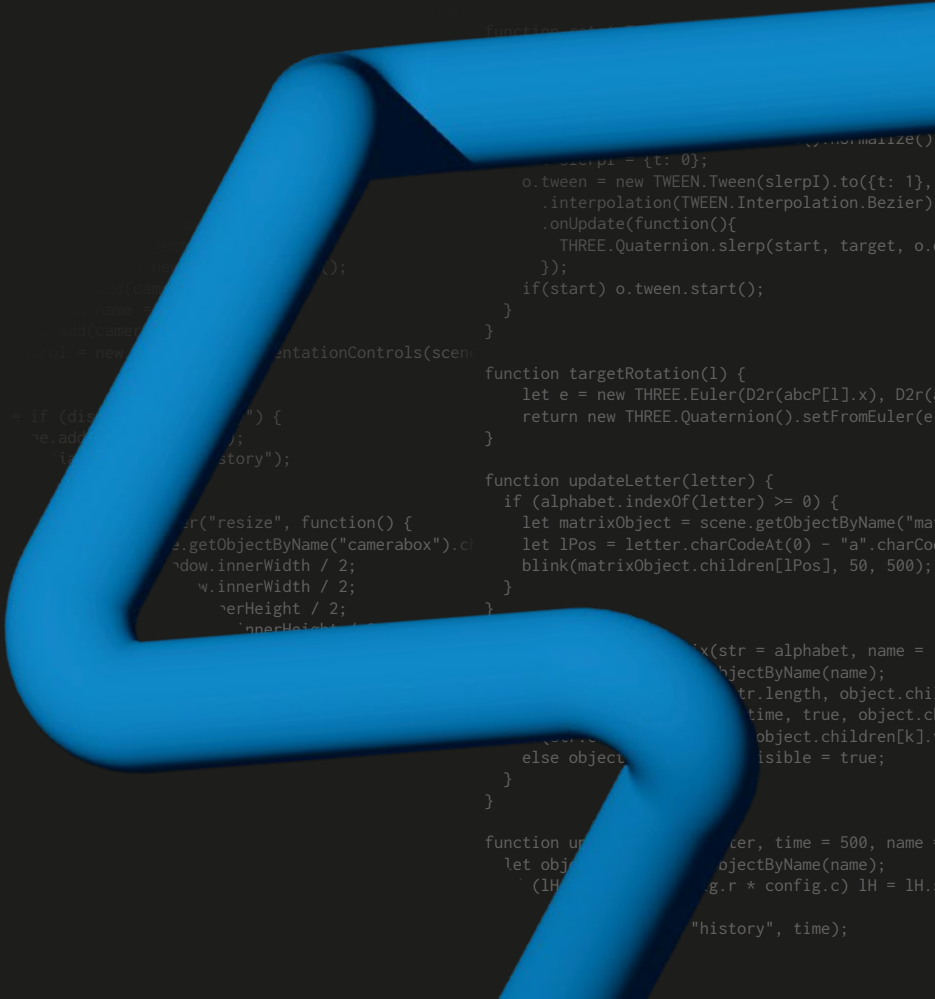
2 Vgl. Luciana Parisi, *Contagious Architecture: Computation, Aesthetics, and Space*, The MIT Press, Cambridge (MA), London, 2013, S. 169ff.

3 Vgl. Rob Kitchin, *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*, Sage Publications, Los Angeles, London, 2014, S. 67ff.

4 Vgl. Gilles Deleuze, „Postscript on the Societies of Control“, in: *October*, Bd. 59, Winter 1992, S. 3–7.

#Arbeit&Produktion

#Industrie4.0
#InternetDerDinge
#Programmieren
#SmartFactories
#Automation
#Arbeit4.0



Die Nachfrage nach Gütern und Dienstleistungen on demand, die auf individuelle Vorlieben zugeschnitten und sieben Tage die Woche rund um die Uhr erhältlich sind, steigt stetig. Dieses Phänomen der digitalen Ökonomie ist ein Geschäftsmodell, das sich über alle Wirtschaftszweige hinweg erstreckt – von der verarbeitenden Industrie und dem Dienstleistungssektor, bis hin zum Transportwesen und der Telekommunikationsbranche – und das in hohem Maße von Informationstechnologien abhängig ist.¹ Die Organisation und das Management der gesamten Wertschöpfungskette im Verbrauchsgüterbereich wandeln sich dadurch grundlegend und eine neue Infrastruktur wird etabliert. Möglich wird dieses Wirtschaftsmodell durch die Echtzeitvernetzung von Waren, Abläufen und Infrastrukturen sowie durch die Vernetzung mit der internetbasierten Kundenbetreuung. Starre Wertschöpfungsketten werden so zu hochflexiblen Wertnetzen.

Für diesen Ansatz, der auch als die vierte industrielle Revolution eingestuft wird, wurde der Begriff #Industrie4.0 geprägt. Ihr zentrales Merkmal ist Interoperabilität, d. h. die Vernetzung von Maschinen, Geräten, Sensoren und Menschen und der damit einhergehende Informationsaustausch in Echtzeit über das #InternetDerDinge. Durch diese Transparenz können dynamische, effiziente Produktionsprozesse etabliert werden, die sich beispielsweise im Hinblick auf Kosten, Verfügbarkeit und Ressourcenverbrauch optimieren lassen. Da Software und Maschinen autonom agieren, ist es nicht notwendig, sie aufwändig neu zu #Programmieren, wenn veränderten Anforderungen Rechnung getragen werden muss. Dadurch kann auf individuelle Kundenwünsche schnell reagiert werden. Die einzelnen Teile der Kette „wissen“, was sie sind, wo sie hingehören und wie sie verfahren müssen. Auch können sie mit der Produktionsstätte interagieren. Der Betrieb entscheidet dann selbst, was unter Berücksichtigung der gesetzten Prioritäten und des Zeitrahmens geschehen soll. In diesen modular aufgebauten #SmartFactories [intelligente Fabriken] erkennt die eingesetzte Software Mängel und Fehler bereits im Anfangsstadium und kann entsprechend gegensteuern.²

Die Industrie 4.0 befindet sich aktuell noch in einem fortlaufenden Entwicklungsprozess. Um erfolgreich zu sein, sind auf globaler Ebene noch große Anstrengungen im Hinblick auf Standardisierung und Vereinheitlichung erforderlich. Sowohl national als auch international müssen branchenübergreifend neue Formen der Unternehmenskooperation entwickelt werden. Die höchst flexiblen Wertnetzwerke der Smart Factories machen die Angleichung von Interfaces (siehe #Interface im Themenbereich #Codierung), d. h. eine Referenzarchitektur sowie einheitliche Definitionen und Methoden, erforderlich. Notwendig sind eine gemeinsame Struktur und Sprache zur standardisierten Beschreibung und Spezifikation von Systemen. Die Industrie 4.0 bringt

viele Herausforderungen für die Informationstechnologie und den Datenschutz mit sich, die sich negativ auf die Integrität von Produktionsprozessen auswirken können. Auch werden dadurch juristische Fragen aufgeworfen, die gesetzlich geregelt werden müssen. Dazu zählen etwa Datenschutz (auf Firmen-, Angestellten- und Kundenseite) und Haftungsfragen in Bezug auf automatisierte Systeme.³

Die tief greifendste Veränderung jedoch, welche die neuen Geschäftsmodelle mit sich bringen, ist die Art und Weise, wie Arbeit organisiert wird. Die #Automation bedroht immer stärker Routinetätigkeiten und unqualifizierte Arbeitsplätze, da sie von intelligenten Maschinen und Robotern (siehe #Roboter im Themenbereich #MaschinellesLernen) übernommen werden können. Die ArbeitnehmerInnen sind gezwungen, sich ein viel breiteres Spektrum an Fähigkeiten anzueignen, damit sie Handlungen ausführen und Entscheidungen treffen können, zu denen Algorithmen (siehe #Algorithmus im Themenbereich #Codierung) nicht in der Lage sind. Aufgrund dieser Entwicklungen müssen die ArbeitnehmerInnen für neue Aufgaben ausgebildet und qualifiziert werden, sie müssen flexibler und mobiler werden. Diese Veränderungsprozesse werden mit dem Begriff #Arbeit4.0 zusammengefasst.⁴ In einem Wirtschaftssystem, das gänzlich auf Innovation ausgerichtet ist, wird Wissen zur zentralen Ressource. Es steht der Wandel von einer Arbeitsgesellschaft zu einer Wissensgesellschaft bevor. Nicht nur immer weniger Menschen werden immer mehr verdienen, sondern auch immer mehr Menschen wird immer weniger (Routine-)Arbeit zur Verfügung stehen und immer weniger Menschen werden immer mehr (hochtechnisierte/hochqualifizierte) Arbeit leisten. Wissen wird das neue Gold bzw. Öl. Am einen Ende des Spektrums sind Arbeitsplätze angesiedelt, die sich durch immer flexiblere und dynamischere Strukturen auszeichnen und damit an individuelle Bedürfnisse angepasst werden können. Das Ziel ist dabei, Kreativität wirtschaftlich nutzbar zu machen. In den Genuss dieser Vorteile kommen jedoch nur Spitzenkräfte, die auch von gesundheitsfördernden und familienfreundlichen Initiativen der Arbeitgeber profitieren. Am andere Ende des Spektrums steht die fabrikmäßige Entwicklung von Software, für die etwa die Akzeleratoren des Silicon Valley stehen.

1 Vgl. Nick Srnicek, *Platform Capitalism*, Theory Redux-Reihe, Polity Press, Cambridge (UK), 2017, S. 4–5.

2 Vgl. Bundesministerium für Wirtschaft und Energie, *Industrie 4.0 und Digitale Wirtschaft: Impulse für Wachstum, Beschäftigung und Innovation*, Berlin, 2015.

3 Vgl. Bundesministerium für Bildung und Forschung, *Zukunftsbild Industrie 4.0*, Berlin, 2015.

4 Vgl. Ned Rossiter, *Software, Infrastructure, Labor: A Media Theory of Logistical Nightmares*, Routledge, New York, 2016, S. 109.

#AlgorithmischeÖkonomie

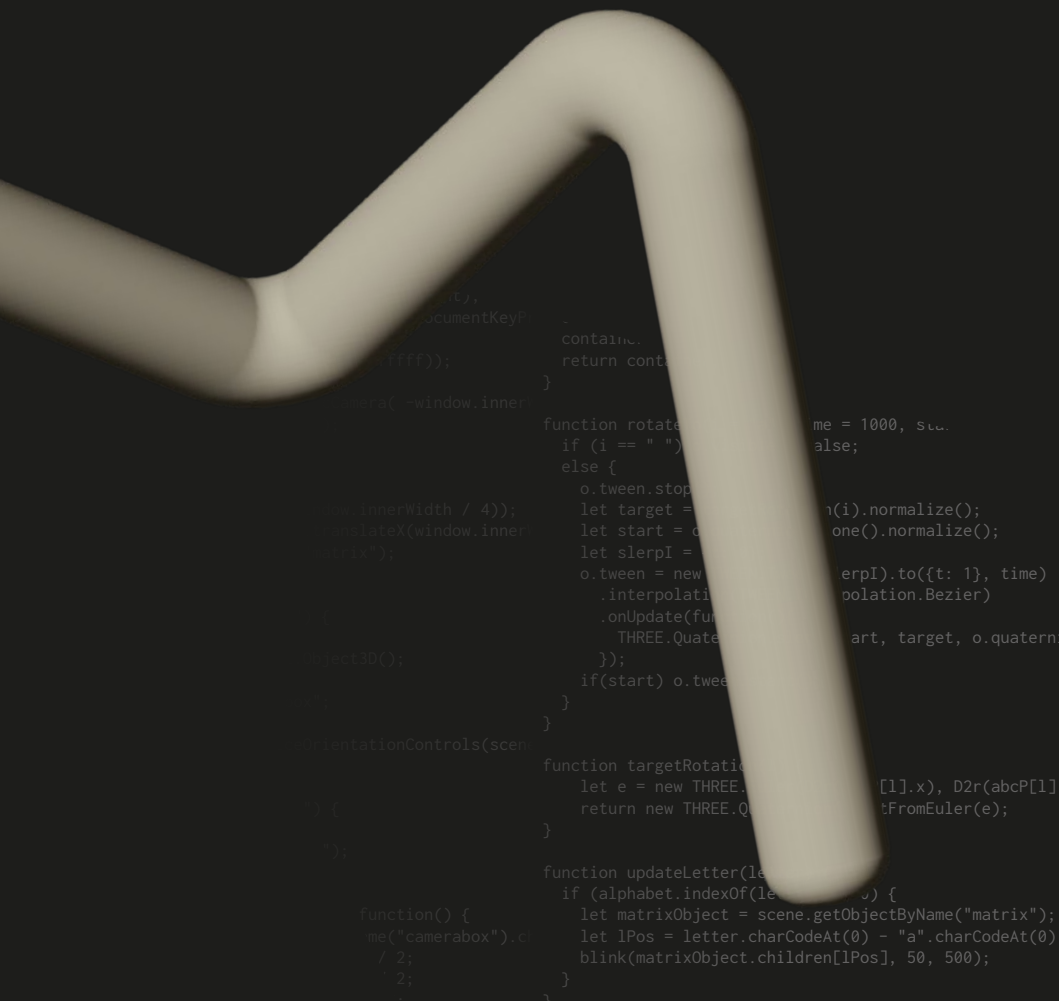
#Hochfrequenzhandel

#Bitcoin

#Kryptowährungen

#Entschlüsseln

#Blockchain



In einer Welt, in der alles zunehmend digital wird (unsere Art zu kommunizieren, unsere Werbung, unsere Freizeit und unser Arbeitsleben), war es nur eine Frage der Zeit, bis auch Geld digital hergestellt werden konnte. Banken und Märkte nutzen digitale Algorithmen seit Jahrzehnten und auch viele EndverbraucherInnen greifen bereits seit einiger Zeit digital auf ihr Geld zu. Aber der Sachverhalt, der hier betrachtet werden soll – die #AlgorithmischeÖkonomie – geht weit darüber hinaus und ist wesentlich vielschichtiger: Welche Auswirkungen hat der Einsatz von Computercodes auf unsere globalisierte Wirtschaft? Welche Systeme haben sich herausgebildet – oder werden in Zukunft entstehen?

Wenn man sich die Verbindung von Wirtschaft, Mathematik und Informatik anschaut, ist einer der ersten Begriffe, die in diesem Kontext auftauchen, der algorithmische Handel. Dabei handelt es sich um eine gängige Praxis im Bereich des Investmentbankings und der Rentenmärkte, bei der automatisiert eine Abfolge von Rechenbefehlen zum Einsatz kommt, mit denen an den Finanzmärkten Entscheidungen getroffen und Transaktionen durchgeführt werden. Das heißt, dass der Handel mit Aktien heute größtenteils durch Algorithmen (siehe #Algorithmus im Themenbereich #Codierung) erfolgt. Viele Systeme, die solche automatisierten Vorgänge ausführen, fallen in die Kategorie #Hochfrequenzhandel (HFT). Bezeichnend für sie ist die immense Geschwindigkeit, mit der sie agieren, und die von Menschen nie – nicht einmal annähernd – erreicht werden könnte.

Um gegen die Vormachtstellung dieses Systems und die von ihm ausgehende Finanzialisierung anzugehen, wurde 2009 im Internet eine neue Währung geschaffen: #Bitcoin. Ihr folgten noch viele andere digitale Währungen wie beispielsweise Ethereum oder Litecoin. Worin aber unterscheiden sich #Kryptowährungen nun von herkömmlichen Währungen? Wie der Name schon andeutet, basieren sie auf einem kryptografischen System¹, was bedeutet, dass der ihnen zugrunde liegende Code in ein System implementiert ist, das Informationen geheim hält. Nur die Personen – oder, genauer gesagt, die Programme –, die sie dechiffrieren bzw. #Entschlüsseln können, haben Zugriff auf diese Informationen. Außerdem haben Kryptowährungen keine materielle Entsprechung und sind dezentral organisiert. Anders als im Fall von zentralisierten Bankensystemen, bei denen Regierungen den Geldwert durch das Drucken von Geld kontrollieren, haben Regierungen über Kryptowährungen keine Kontrolle: Ihr Wert zirkuliert im Internet, ohne dass eine Zwischeninstanz regulierend eingreift.

Um die Zusammenhänge erfassen zu können, ist es notwendig, das System zu verstehen, das hinter den Kryptowährungen steht: #Blockchain. Hierbei handelt es sich um eine offene Datenbank,

mit der im Fall der Kryptowährungen Finanztransaktionen in ihrer Zeitfolge aufgezeichnet werden. Blockchain [engl. für Blockkette] besteht aus einzelnen Datenblöcken, in denen verschiedene Transaktionen zusammengefasst werden und die jeweils mit einem vorhergehenden Eintrag in der Kette verbunden sind. Wenn eine Person etwas mit Bitcoins kauft, dann wird an alle Computer (die sogenannten Miner), die gleichberechtigt in dem Peer-to-Peer-Netzwerk des Bitcoin-Systems zusammengeschaltet sind, eine schwer zu lösende, kryptografische Aufgabe geschickt. Wenn nun ein Miner eine solche Aufgabe löst, dann wird der Blockkette ein neuer Block hinzugefügt und mit einer bestimmten Anzahl von Bitcoins vergütet. Aber der Erwerb von Bitcoins ist nicht der alleinige Zweck des Minings: Die gestellten Aufgaben sind so komplex, dass mit jedem Block, der neu hinzukommt, die vorhergehenden Blöcke sicherer werden und damit die gesamte Kette. Um die Blockkette zu hacken, wäre zur Veränderung selbst nur einer Transaktion eine enorme Rechengeschwindigkeit notwendig. Dadurch, dass viele Miner stetig neue Blöcke hinzufügen, wäre dazu eine immense Rechenleistung erforderlich.

Wie auch andere disruptive Technologien, die im digitalen Zeitalter aufkommen, stellen Kryptowährungen für die etablierten Wirtschaftsstrukturen und -abläufe eine Herausforderung dar, da sie auf eine Zukunft zielen, in der es keine Mittlerinstanzen mehr gibt. In einer von Blockchain-Technologien verwalteten Welt, würden neue Geschäftsstrategien und Instrumente für den Finanzhandel entwickelt, wodurch die „Kontrolle von Geld und Information nicht mehr bei den mächtigen Eliten läge [...], sondern in der Hand der Menschen, von denen sie eigentlich ausgehen sollte.“² Während von mancher Seite angeführt wird, dass solche Modelle die zentralisierte Wirtschaft und das politische Establishment zerschlagen könnten, ist andererseits zu hören, dass die Auswirkungen auf den Arbeitsmarkt gravierend wären und nur die Spitzenkräfte in der Arbeitnehmerschaft davon profitieren würden. Zwar ist es nicht möglich, die Zukunft genau vorauszusagen, um jedoch die Welt, in der wir leben und die Wirtschaft, die wir aufbauen, verstehen zu können, müssen wir uns der Macht, die Algorithmen und computerbasierter Berechnungsverfahren haben, bewusst werden und sie genau untersuchen.

1 Der Begriff „Kryptografie“

leitet sich vom griechischen κρυπτός bzw. *kryptós* ab, was „verborgen“ oder „geheim“ bedeutet, sowie von γράφειν bzw. *graphein*, dem griechischen Wort für „schreiben“. (Siehe dazu: Henry George Liddell und Robert Scott, *A Greek-English Lexicon*, Oxford University Press, 1984.)

2 Paul Vigna und Michael

J. Casey, *The Age of Cryptocurrency: How Bitcoin and Blockchain are Challenging the Global Economic Order*, St. Martin's Press, New York, 2015, S. 6. Übersetzt aus dem Englischen von PK.

#VirtuelleRealität

#HMD

#ComputersimulierteUmgebungen

#AugmentedReality

#ComputerGeneriertesDesign

#Eskapismus

```
function init() {
  scene = new THREE.Scene();
  camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
  camera.position.z = 5;
  renderer = new THREE.WebGLRenderer({ antialias: true });
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);

  // Create a plane
  planeGeometry = new THREE.PlaneGeometry(10, 10);
  planeMaterial = new THREE.MeshBasicMaterial({ color: 0x000000 });
  plane = new THREE.Mesh(planeGeometry, planeMaterial);
  plane.position.y = 0.5;
  scene.add(plane);

  // Create a camera box
  cameraBoxGeometry = new THREE.BoxGeometry(2, 2, 2);
  cameraBoxMaterial = new THREE.MeshBasicMaterial({ color: 0x000000 });
  cameraBox = new THREE.Mesh(cameraBoxGeometry, cameraBoxMaterial);
  cameraBox.position.z = 1.5;
  scene.add(cameraBox);

  // Create a camera
  camera.position.z = 5;
  camera.lookAt(0, 0, 0);

  // Create a main object
  mainObjectGeometry = new THREE.BoxGeometry(1, 1, 1);
  mainObjectMaterial = new THREE.MeshBasicMaterial({ color: 0x000000 });
  mainObject = new THREE.Mesh(mainObjectGeometry, mainObjectMaterial);
  mainObject.position.z = 1.5;
  scene.add(mainObject);

  // Create an ABC matrix
  ABCMatrix({ name: 'ABCMatrix', r: 1, c: 1 });

  // Add event listeners
  window.addEventListener('resize', function() {
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize(window.innerWidth, window.innerHeight);
  });

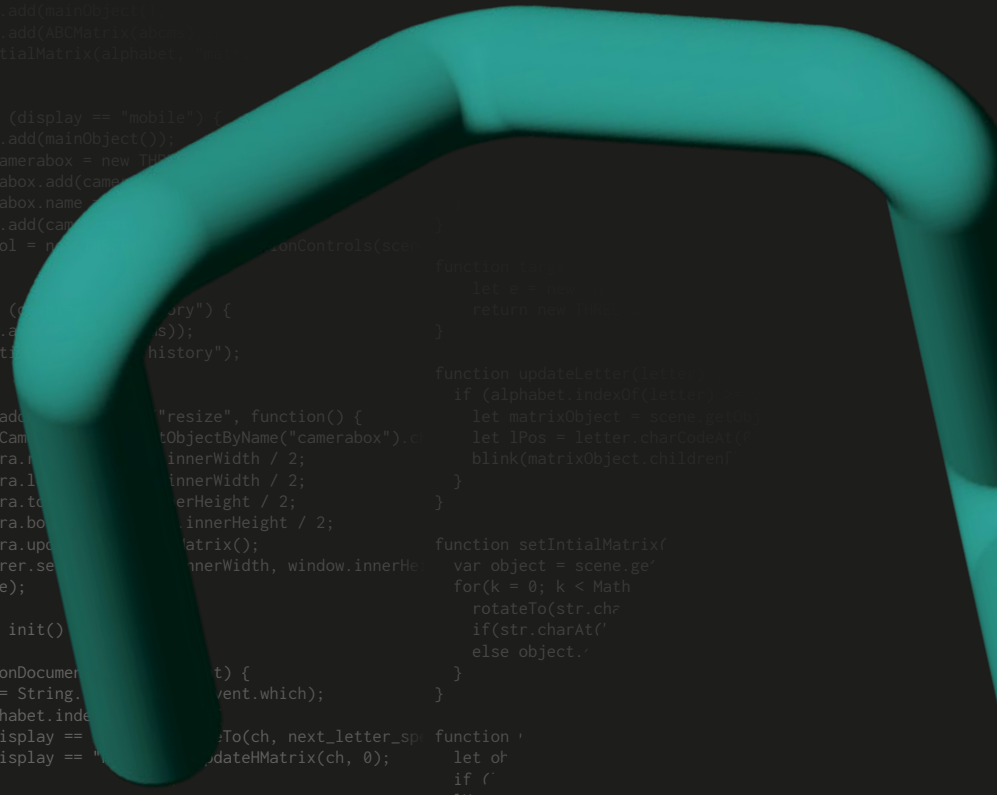
  document.addEventListener('keydown', function(event) {
    let ch = String.fromCharCode(event.which);
    if (alphabet.indexOf(ch) !== -1) {
      if (display == "mobile") {
        updateHMatrix(ch, 0);
      }
    }
  });
}
```

```
function updateLetterMatrix() {
  let letter = document.getElementById('letter');
  let matrixObject = scene.getObjectByName("camerabox").camera;
  let lPos = letter.getBoundingClientRect();
  matrixObject.position.x = lPos.x + window.innerWidth / 2;
  matrixObject.position.y = lPos.y + window.innerHeight / 2;
  matrixObject.rotation.z = 0;
  matrixObject.rotation.y = 0;
  matrixObject.rotation.x = 0;
  scene.add(matrixObject);
}

function setInitialMatrix() {
  let object = scene.getObjectByName("camerabox");
  for(k = 0; k < Math.PI; k += 0.1) {
    object.rotateTo(str.charCodeAt(k));
    if(str.charCodeAt(k) === 0) {
      object.rotateTo(str.charCodeAt(k));
    }
  }
}

function updateHMatrix(ch, 0) {
  let object = scene.getObjectByName("camerabox");
  if (ch === ' ') {
    object.rotateTo(0);
  } else {
    object.rotateTo(str.charCodeAt(ch));
  }
}

function ABCMatrix(config) {
  let matrix = new THREE.Object3D();
  matrix.name = config.name;
  for(i = config.r; i > 0; i--) {
    for(j = 0; j < config.c; j++) {
      let element = new THREE.Object3D();
      element.name = config.name + "_" + i + "_" + j;
      element.position.x = j;
      element.position.y = i;
      element.position.z = 0;
      scene.add(element);
    }
  }
}
```



Um die aktuelle Bedeutung der Begriffe virtuell und #Virtuelle Realität zu verstehen, ist die Entstehungsgeschichte ihres heutigen Gebrauchs zu betrachten. Virtuelle Realität (VR) wird als technischer Begriff verstanden, als ein Medium, das für die BetrachterInnen die Erfahrung von Raum reproduziert – und zwar eines Raumes, der physisch nicht vorhanden ist, d. h. der beispielsweise taktil nicht erfahrbar ist. Letzteres trifft insbesondere dann zu, wenn andere Reize ausgespart bleiben (was z. B. beim Einsatz von Head-Mounted-Displays (#HDMs) der Fall ist). In der Kunst werden seit ca. 1920 Volumen, die nur scheinbar existieren, virtuell genannt (z. B. Naum Gabo, *Konstruktion*, 1921). Die motorbasierte und elektrisch betriebene Drehung eines Drahtes erzeugt beispielsweise auf einer Scheibe ein scheinbares Volumen. Experimentalpsychologie und Gestalttheorie der 1920er-Jahre befassten sich mit diesen Phänomenen, z. B. dem stereokinetischen Effekt. Kinetik und Op-Art sind deren Produkte. Seit der Ausdruck im Zusammenhang mit Computertechnologie gebraucht wird, fand die Bedeutung, die sich auf #ComputersimulierteUmgebungen bezieht, auch Eingang in Wörterbuchdefinitionen: simuliert von einem Computer oder einem Computernetzwerk oder in einer virtuellen Realität existierend.¹

Daraus können wir folgern, dass „das Virtuelle ein Substitut ist – , es agiert ohne materielle Handlungsträgerschaft‘ – , ein immaterieller Stellvertreter des Materiellen.“²

Im Bereich der Philosophie haben Henri Bergson (1859–1941), Gilles Deleuze (1925–1995), Félix Guattari (1930–1992) und Pierre Lévy (*1956) verschiedene Konzepte des Virtuellen entwickelt. Bergson beschreibt die Immaterialität der Erinnerung als virtuell.³ Für Deleuze steht virtuell nicht im Gegensatz zu real, sondern zu aktuell, sodass virtuell in diesem Ansatz einen Modus der Realität bezeichnet.⁴ Für Guattari ist das Virtuelle eine von „vier ontologischen Faktoren“⁵, zu denen zudem, das Aktuelle, das Reale und das Mögliche zählen.

Der Begriff „Virtuelle Realität“ ist recht jung und wurde wahrscheinlich von Antonin Artaud (1896–1948) in seinem 1938 auf Französisch erschienenen Buch *Das Theater und sein Double* geprägt.⁶ Unser heutiges Verständnis entspricht nicht mehr der Begriffsverwendung Artauds. Während der letzten Jahrzehnte hat sich seine Bedeutung geändert und der Begriff wird heute – wie auch der Begriff #AugmentedReality – vor allem für computer-gestützte interaktive und immersive Umgebungen gebraucht, auf die man über projizierte Bilder zugreift sowie in vielen Fällen über zusätzliche technische Geräte (wie etwa HMDs).

KünstlerInnen und TechnikerInnen begannen mit dem Medium in den 1980er-Jahren (Myron W. Krueger, *Artificial Reality*, 1983) zu experimentieren und trugen mit zur Entwicklung des #Computer GeneriertesDesign bei. Vor allem in den 1990er-Jahren entstanden zahlreiche Anwendungen und künstlerische Experimente auf dem

Gebiet der VR, aus denen Kunstwerke hervorgingen. Seit der wachsenden Verfügbarkeit von Hardware und Software für VR-Anwendungen nutzen KünstlerInnen zunehmend Virtual Reality als künstlerisches Medium.

Das Medium ermöglicht völlige visuelle Immersion und öffnet nicht nur, wie Leon Battista Alberti (1404–1472) in seinem Traktat *Über die Malkunst* (1435) formulierte, ein Fenster, wie die Malerei und gerahmte Bilder es tun. Vielmehr zieht *The Art of Immersion* die BetrachterInnen in das Bild hinein. VR bildet eine Tür, durch welche die BetrachterInnen aus der realen Welt in die virtuelle Welt ein- und austreten können. VR öffnet buchstäblich die Tür zu einer anderen Realität.

Nicht nur in der Spieleindustrie, sondern auch in der Medizin wird die Technologie häufig eingesetzt. So können beispielsweise ChirurgInnen mithilfe virtueller Modelle den sichersten und effizientesten Weg zur Lokalisation und Operation von Tumoren bestimmen. PsychologInnen und andere medizinische Berufsgruppen nutzen VR, um traditionelle Therapiemethoden weiterzuentwickeln und effektive Behandlungsmethoden für Posttraumatische Belastungsstörungen (PTBS), Angststörungen und soziale Phobien zu finden. MaklerInnen und ArchitektInnen begleiten potenzielle KäuferInnen, MieterInnen oder BauunternehmerInnen auf Rundgängen durch noch nicht existierende Gebäude.

Virtual-Reality-Technologien werden langsam allgegenwärtig, aber nicht nur aufgrund des Höchstmaßes an #Eskapismus, den das Medium bietet, sondern auch wegen seiner praktischen und wirtschaftlichen Einsatzmöglichkeiten.

1 Vgl. *Webster's Third New International Dictionary*, ungekürzte Auflage, Merriam Webster, 1993.

2 Anne Friedberg, *The Virtual Window: From Alberti to Microsoft*, The MIT Press, Cambridge (MA), London, 2006. S. 8. Übersetzt aus dem Englischen von PK.

3 Vgl. Henri Bergson, *Materie und Gedächtnis. Eine Abhandlung über die Beziehung zwischen Körper und Geist*, mit einer Einleitung von Erik Oger, Verlag Felix Meiner, Hamburg, 1991, S. 127.

4 Manuel DeLanda liefert eine umfassende Erklärung für den Vorgang, den Deleuze als „Gegenverwirklichung“ bezeichnet (der Übergang vom Aktuellen zum Virtuellen) –, siehe dazu Manuel DeLanda, *Intensive Science and Virtual Philosophy*, Continuum, London, 2002.

5 Felix Guattari, *Chaosmosis: An Ethico-aesthetic Paradigm*, Indiana University Press, Bloomington, 1995. Übersetzt aus dem Englischen von PK.

6 Vgl. Antonin Artaud, *Das Theater und sein Double*, Matthes & Seitz, München, 1996, S. 52.

#GenetischerCode

#DNA
#Quellcode
#Bioengineering
#Phänotyp
#DNADatenspeicher
#Genotyp

```

    .setFromEuler(e);
    scene.add(object);
    scene.update();
    }
    }
    }
    }

function setInitialMatrix(scene, str, name = "abcP", time = 1000){
  scene.remove(object);
  var object = scene.generateMatrixObject(name);
  for(k = 0; k < Math.PI; k += 0.1, object.children.length > 0){
    rotateTo(str.charAt(k), object.children[k], true, object.children[k].parent);
    if(str.charAt(k) === " " || str.charAt(k) === ",") object.children[k].visible = false;
    else object.children[k].visible = true;
  }
}

function rotateTo(letter, object, visible, parent) {
  const letterIndex = alphabet.indexOf(letter);
  const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
  const quaternion = new THREE.Quaternion().setFromEuler(e);

  object.rotation.setFromQuaternion(quaternion, Math.PI);
  object.visible = visible;
  parent.add(object);
  scene.update();
}

function updateLetter(letter) {
  const letterIndex = alphabet.indexOf(letter);
  const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
  const quaternion = new THREE.Quaternion().setFromEuler(e);

  object.rotation.setFromQuaternion(quaternion, Math.PI);
  scene.update();
}

function setInitialMatrix(scene, str, name = "abcP", time = 1000, start = true, o) {
  scene.remove(object);
  var object = scene.generateMatrixObject(name);

  function rotateTo(letter, object, visible, parent) {
    const letterIndex = alphabet.indexOf(letter);
    const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
    const quaternion = new THREE.Quaternion().setFromEuler(e);

    object.rotation.setFromQuaternion(quaternion, Math.PI);
    object.visible = visible;
    parent.add(object);
    scene.update();
  }

  function updateLetter(letter) {
    const letterIndex = alphabet.indexOf(letter);
    const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
    const quaternion = new THREE.Quaternion().setFromEuler(e);

    object.rotation.setFromQuaternion(quaternion, Math.PI);
    scene.update();
  }

  scene.update();
  scene.clock.start();
}

function generateMatrixObject(scene, str, name = "abcP", time = 1000, start = true, o) {
  scene.remove(object);
  var object = scene.generateMatrixObject(name);

  function rotateTo(letter, object, visible, parent) {
    const letterIndex = alphabet.indexOf(letter);
    const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
    const quaternion = new THREE.Quaternion().setFromEuler(e);

    object.rotation.setFromQuaternion(quaternion, Math.PI);
    object.visible = visible;
    parent.add(object);
    scene.update();
  }

  function updateLetter(letter) {
    const letterIndex = alphabet.indexOf(letter);
    const euler = new THREE.Euler(D2r(abcP[1].x), D2r(abcP[1].y), D2r(abcP[1].z));
    const quaternion = new THREE.Quaternion().setFromEuler(e);

    object.rotation.setFromQuaternion(quaternion, Math.PI);
    scene.update();
  }

  scene.update();
  scene.clock.start();
}

```

Man weiß über die #DNA (Desoxyribonukleinsäure), dass sie #Quellcode beinhaltet, den sogenannten genetischen Code (#GenetischerCode). Dieser ist ein Satz von Regeln, mit dem im genetischen Material (in DNA- oder mRNA-Sequenzen) codierte Informationen von lebenden Zellen in Proteine übersetzt werden.

Die Beschreibung des genetischen Codes begann in den 1950er-Jahren. 1953 wusste man, dass die genetische Information der DNA, einem Makromolekül in Form einer Doppelhelix (James Watson, Francis Crick), aus vier chemischen Basen besteht: Adenin (A), Guanin (G), Cytosin (C) und Thymin (T). Zu dieser Zeit lautete der oberste Grundsatz der Molekularbiologie, dass die DNA den Code für die Herstellung von Proteinen beinhaltet, die als Katalysatoren die Strukturen des Lebens „ausführen“.¹

Die Sprachbilder und Formulierungen, die damals in der Molekularbiologie verwendet wurden, waren stark von der #Kybernetik (siehe Themenbereich #MaschinellesLernen) und der Informationstheorie geprägt. Diese Disziplinen gewannen in den späten 1940er- und den 1950er-Jahren an Einfluss, zu der Zeit also, als sich die Genetik etablierte.²

DNA und RNA wurden als „informationelle Moleküle“ und „Bänder“ bezeichnet, die den Regeln der Informationsverarbeitung unterliegen.³ Der genetische Code wurde auch mit einem Computerprogramm verglichen, denn es hieß, dass „Organe, Zellen und Moleküle durch ein Kommunikationsnetzwerk verbunden sind.“⁴

Den biologischen Code des „Buchs des Lebens“ zu entschlüsseln war ein zentrales Anliegen der Molekularbiologie und die Forscher wetteiferten darum, ihn zu knacken. Das Humangenomprojekt (Human Genome Project (HGP), 1990–2003) war ein internationales Forschungsprojekt, das sich zum Ziel gesetzt hatte, den menschlichen #Genotyp⁵, also die Abfolge der paarweise angeordneten Nukleinbasen, aus denen sich die menschliche DNA zusammensetzt, zu bestimmen. Das private Forschungsprojekt von Craig Venter (Celera Corporation) arbeitete parallel seit 1998 mit automatisierter Sequenzierung.

Durch die Fortschritte auf dem Gebiet der Molekularbiologie entstanden neue Wissenschaftszweige, so z. B. das Bioingenieurwesen bzw. #Bioengineering. Darunter versteht man „die Manipulation eines Organismus zur Produktion nicht-nativer Moleküle (wie z. B. medizinische Wirkstoffe oder Proteine)“.⁶

Ein Schlüsselverfahren dieser Disziplin ist die sogenannte rekombinante DNA-Technik, eine Methode die ursprünglich in den 1970er-Jahren von Stanley N. Cohen (*1935) und Herbert Boyer (*1936) entwickelt wurde. Die beiden Wissenschaftler schleusten damit menschliche DNA in Bakterien ein, um eine rekombinante Form des Insulins zur Behandlung von Diabetes herzustellen. Zu den neuesten Entwicklungen auf diesem Gebiet gehören Verfahren zur Editierung des Genoms, wie etwa die CRISPR/

Cas9-Methode, die jüngst große mediale Aufmerksamkeit erfuhr.⁷ Aus dem Genome Editing, der Genomchirurgie bzw. Genomschere ergeben sich zahlreiche Anwendungsmöglichkeiten, da WissenschaftlerInnen damit jedes – auch das menschliche – Genom verändern können. Diese Methode, genauso wie die Gentechnik im Allgemeinen, wirft ethische Fragen auf.

Jüngst fand man heraus, dass DNA-Moleküle jede beliebige Form von Daten speichern können (#DNADatenspeicher). Text- oder Bildinformationen, selbst Bewegtbilder, können in Binärcode und dann in genetischen Code umgewandelt werden.⁸ WissenschaftlerInnen gelang es beispielsweise, eine Folge von Einzelbildern aus Eadweard Muybridges (1830–1904) *Bewegungsstudien von Mensch und Tier* in der DNA von Bakterien zu speichern, also so zu codieren, dass sie aus dem DNA-Speicher auch wieder ausgelesen werden können.

1 Adrian Mackenzie und Theo Vurdubakis, „Codes and Codings in Crisis: Signification, Performativity and Excess“, in: *Theory, Culture & Society*, Bd. 28, Nr. 6, 2011, S. 3–23, hier S. 7.

2 Vgl. Lily E. Kay, *Who Wrote the Book of Life: A History of Genetic Code*, Stanford University Press, Stanford, 2000, insbes. S. 73–127.

3 Vgl. Carl R. Woese, *The Genetic Code: The Molecular Basis for Genetic Expression*, Harper & Row, New York, 1967, S. 253–254.

4 François Jacob, *The Logic of Life: A History of Heredity* [1970], Pantheon Books, New York, 1973.

5 Unter #Genotyp versteht man die Gesamtheit der Gene eines Organismus, während der Begriff #Phänotyp ihre konkrete Ausprägung bezeichnet.

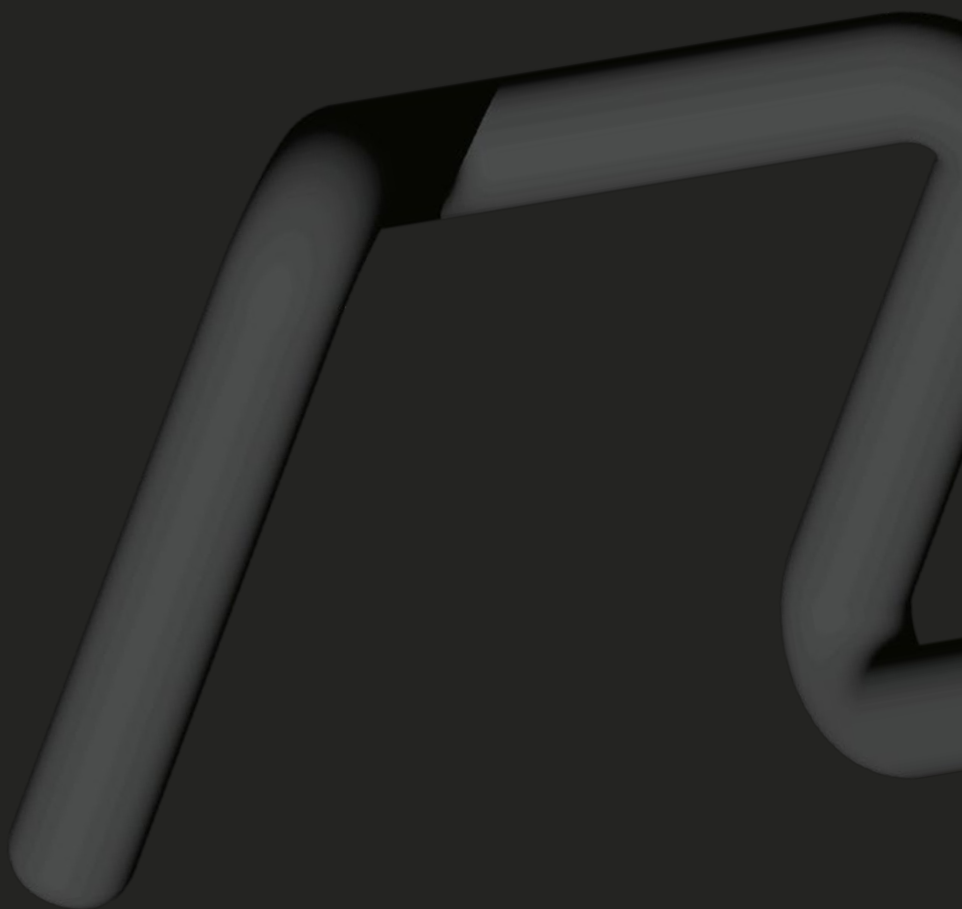
6 Brandon Adkins, *A Future Guide to Bioengineering*, Kindle E-Book, Amazon Distribution, Leipzig, 2016, S. 5. Übersetzt aus dem Englischen von PK.

7 Vgl. Alex Reis, „CRISPR/Cas9 and Targeted Genome Editing: A New Era in Molecular Biology“, in: *NEB expressions*, Nr. 1, 2014, online: <https://www.neb.com/tools-and-resources/feature-articles/crispr-cas9-and-targeted-genome-editing-a-new-era-in-molecular-biology>, 13.09.2017.

8 Vgl. Andy Extance, „How DNA Could Store All the World’s Data?“, in: *Nature*, Bd. 537, Nr. 7618, September 1, 2016, S. 22–24.

Signal- und Maschinencodes

Franz Pichler



Codes: Mathematische Objekte zur Übermittlung von Nachrichten

Signalcodes

Signale dienen zur Kommunikation zwischen Menschen und Maschinen. Sie ermöglichen den Austausch von Nachrichten und vermitteln damit Information. Jedes Signal hat sowohl eine technisch-materielle als auch eine mathematisch-abstrakte Komponente. Zum Beispiel besteht jedes zwischen einem Sender und zugehörigen Empfänger übertragene Morsesignal zum einen aus einer Folge von elektrischen energetischen Zuständen, die zum anderen mathematisch als „Punkte“ und als „Striche“ des Morsealphabets gedeutet werden können und so den Morsecode bilden. Für den Nachrichtenaustausch ist der Morsecode das Wesentliche. Das elektrisch erzeugte Signal hat die Aufgabe, den Morsecode mit den vorhandenen technischen Mitteln zu realisieren und zu übertragen. Zur Erzeugung von Signalen und zu deren Übertragung und Speicherung und zu deren Empfang sind bestimmte technisch-physikalische Geräte erforderlich. Für optische, akustische, mechanische, magnetische und elektrische Signale werden dazu technischen Mittel eingesetzt, die mittels der Physik, der Optik, der Akustik, der Mechanik, des Magnetismus, der Elektrizität und der elektromagnetischen Wellen erzeugt werden können. Alle diese Signale können im Prinzip zur Darstellung ein und desselben Signalcodes eingesetzt werden.

Die Zuordnung des zu einem Signal gehörenden Codes nennt man Codierung. Die Transformation eines Signals in ein anderes Signal nennt man Signalwandlung. Von analogen Signalen spricht man, wenn die zugehörige mathematische Beschreibung die Verwendung des Systems der reellen Zahlen verlangt. Analoge Signale erhält man vorwiegend bei der Messung physikalischer Zustände. Auch die Signale, die von Musikinstrumenten oder durch die menschliche Stimme erzeugt werden, sind Beispiele für analoge Signale. Digitale Signale sind solche, die mathematisch mittels des Systems der ganzen Zahlen beschrieben werden können. Digitale Signale zusammen mit den zugeordneten digitalen Codes ermöglichen heute mit den Mitteln der Mikroelektronik, der Computertechnik und der Informatik einen effektiven Einsatz zum Nachrichtenaustausch und sind auch die Basis für Daten zur Verarbeitung durch Computer.

Codes: Mathematische Objekte zur Erzeugung von dynamischen Prozessen

Maschinencodes

Wir sind im Alltag und in der Arbeit stets mit dynamischen Prozessen konfrontiert. Als solche bezeichnen wir Vorgänge, die in zeitlicher Folge verschiedene Zustände annehmen. Denken wir zum Beispiel an den dynamischen Prozess, der in Zusammenhang mit dem Kauf einer Fahrkarte in einem Fahrkartenautomat abläuft. Dieser beginnt mit dem Startzustand. Mit den schrittweise, in zeitlicher Folge durch Anklicken, gegebenen Signalen wird eine Steuerung des Automaten in einen Endzustand, dem Ziel, und nach Bezahlung die Ausgabe der entsprechenden Fahrkarte erreicht. Ein weiteres Beispiel für einen dynamischen Prozess ist auch mit der im Fernsehen gezeigten Wetterprognose gegeben, in der die geografischen Wetterzustände in zeitlicher Folge an verschiedenen Tagen der Woche gezeigt werden. Auch die Durchführung von Arbeitsplänen, in denen der jeweilige Stand einer Arbeit gezeigt wird, kann als ein Beispiel für einen dynamischen Prozess gelten. Wir leben in einer Welt, umgeben von einer Vielzahl von dynamischen Prozessen. Schließlich kann man sich selbst als einen, allerdings sehr komplexen, dynamischen Prozess betrachten.

Allgemein betrachtet ist jeder dynamische Prozess ein Konglomerat, das sich aus Anteilen von Materie, Energie und Information zusammensetzt. Im Fokus der Ausstellung *Open Codes* stehen jedoch dynamische Prozesse, die vor allem mit Information zu tun haben. Es geht dabei um die Behandlung der mathematischen Basis, um das mathematische System, das wesentlich für die Generierung von dynamischen Prozessen ist. Im Rahmen der Ausstellung sind dafür die Maschinencodes verantwortlich. Maschinencodes in diesem Sinne führen uns in die Welt der von Isaac Newton und Gottfried Wilhelm Leibniz geschaffenen mathematischen Analysis, in die Welt der Mechanik von Leonhard Euler bis Joseph-Louis de Lagrange und Pierre-Simon Laplace, und auch in die Welt der Automaten, Rechenmaschinen und Computer der Neuzeit sowie der mittels Programmierung geschaffenen Welt der Algorithmen. Charles Babbage, Alan Turing, Kurt Gödel, Konrad Zuse, Howard Aiken, John von Neumann, Claude E. Shannon, Gustav Knuth, um wichtige Pioniere zu nennen, sind hier von besonderer Bedeutung. Mathematisch haben wir es bei Maschinencodes mit Gebieten wie Differenzialgleichungen, Differenzengleichungen, Mathematische Logik, Boolesche Algebra, Endliche Automaten, Zellulare Automaten, Petrinetze oder Algorithmen zu tun. Zu allen diesen Gebieten gibt es zugehörige mathematische Theorien, die in Zusammenhang mit der Behandlung dynamischer Prozesse und zugehöriger Maschinencodes

gesetzt werden können. Aktuelle praktische Beispiele von Maschinencodes sind mit den Programmsystemen für die Sozialen Netzwerke wie Facebook und andere, sowie für Betriebssysteme für persönlich betriebene Computer (z. B. Smartphones) und deren Anwendungssystemen (z. B. Google) gegeben. Aufgrund der damit verbundenen Geschäftsmodelle stellen diese vielfach Gegenbeispiele für den Begriff der „Open Codes“ im Sinne der Informatik, dar. In der Ausstellung *Open Codes* werden sie aber thematisiert.

Codes: Historische technische Bausteine und Schriften

Bausteine für „Open Codes“

Die Ausstellung *Open Codes* behandelt neben dem Blick auf Kunst, Wissenschaft und Gesellschaft auch die historische Entwicklung der für die Erzeugung, Übertragung und Verarbeitung von Codes erforderlichen technischen Bausteine und Systeme. Es geht dabei um Zählwerke in mechanischer oder elektrischer Art, um Speicherbausteine, realisiert in mechanischer, magnetischer oder elektronischer Technologie, um Schaltnetze, von einfachen Kreuzschiebenschaltern mit Kontaktstößeln bis zu den heute eingesetzten mikroelektronischen Gatterbausteinen und Mikroprozessoren. Wegen der historischen Bedeutung des Morsecodes erfährt dieser in der Ausstellung eine besondere Beachtung. Zum Thema Computer werden mechanische Rechenmaschinen verschiedener Art gezeigt. Die technische und mathematische Entwicklung von Codes und deren Verwendung als Signalcodes oder als Maschinencodes geschieht durch die Präsentation von wichtigen Schriften, beginnend mit der *Ars Magna* von Raimundus Lullus und dem Buch von Friedrich von Knaus zu den von ihm konstruierten selbstschreibenden Wundermaschinen bis hin zu den mathematischen Arbeiten von George Boole, Claude E. Shannon und John von Neumann.

Franz Pichler

Open Codes

Werkliste A - Z



A

Jean-Michel Alberola

*1953 in Saïda (DZ), lebt und arbeitet in Paris (FR)

001 *A Mathematical Sky - Henri Poincaré*

2011, Wandinstallation, 2 mathematische Modelle; Collection Fondation Cartier pour l'art contemporain, Paris; Mathematisches Institut, Ruprecht-Karls-Universität Heidelberg; Karlsruher Institut für Technologie Archiv, Sammlung Mathematische Modelle, Karlsruhe; Konzipiert in Zusammenarbeit mit Giancarlo Lucchini, mit Unterstützung des Instituts Henri Poincaré

#GenealogieDesCodes

Morehshin Allahyari

*1985 in Teheran (IR), lebt und arbeitet in Boston (US)

002 *Lamassu* aus der Serie *Material Speculation: ISIS* (2015)

Ebu aus der Serie *Material Speculation: ISIS* (2015)

South Ivan Human Heads: Bearded River God (2017)

Plastikfiguren in 3-D-Druck mit elektronischen Komponenten, 22,2 × 20,3 × 6,4 cm, Edition von 3; Mit freundlicher Genehmigung der Upfor Gallery, Portland

#Codierung

#Decodierung #Eskapismus

AppSphere AG

gegründet 2010 in Ettlingen (DE)

003 *Digitale Transformation. Die Kunst des modernen Arbeitslebens*

Präsentation von Ideen für moderne und zukunftsfähige IT-Arbeitsplatzmodelle; Microsoft Surface Hub, Microsoft Studio, Microsoft Surface Book, Microsoft Hololens; Mit freundlicher Unterstützung von Microsoft

#Arbeit&Produktion

#Programmieren #Arbeit4.0

B

Lisa Bergmann

*1979 in Nürnberg (DE), lebt und arbeitet in Karlsruhe (DE)

004 *All We Know We Know from Light* 2017, HD-Video, Farbe, Ton, 45 Min.

#Codierung

#VirtuelleRealität

#ComputersimulierteUmgebungen

Michael Bielicky, Kamila B. Richter

*1954 in Prag (CZ), lebt und arbeitet in Karlsruhe und Düsseldorf (DE)

*1976 in Olmouc (CZ), lebt und arbeitet in Karlsruhe und Düsseldorf (DE)

005 *Narzisstische Maschine*

2017, Interaktive Installation, Kamera, Computer, Software, Moskitonetze; Softwareentwicklung: Lukas Böhm, Lukas Feller; Sound: Lorenz Schwarz

#Codierung

#QuantifiziertesSelbst

#Algorithmus

Patrick Borgeat

*1985 in Öhringen (DE), lebt und arbeitet in Karlsruhe (DE)

006 *Notation. Prozess. Musik.*

2017; Videopräsentation

#Codierung

#Klangprogrammierung #Interface

James Bridle

*1980 in London (GB), lebt und arbeitet in Athen (GR)

007 *Autonomous Trap 001*

2017, Performance-Dokumentation, Archiv-Pigmentdruck, 150 × 200 cm; Mit freundlicher Genehmigung von NOME, Berlin

#Arbeit&Produktion

#Industrie4.0 #Selbstfahrende-Autos #Automation

Ludger Brümmer (Idee), Benjamin Miller (Programmierung, Interface-Design)

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)

*1986 in Paris (FR), lebt und

arbeitet in Karlsruhe (DE)

008 CellularAutomataExplorer

2017, Interaktive Klanginstallation, Computer, Bildschirm, Maus; Produktion des ZKM_Hertz-Lab

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Elizabeth Pich

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1989 Friedberg (DE), lebt und arbeitet in Karlsruhe (DE)

009 CodeChain

2017, Interaktive Klanginstallation, App, Tablet-PC; Produktion des ZKM_Hertz-Lab

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Dan Wilcox

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1981 in Orange (US), lebt und arbeitet in Karlsruhe (DE)

010 LindemayerExplorer

2017, Interaktive Klanginstallation, Computer, Bildschirm, Maus; Produktion des ZKM_Hertz-Lab

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer (Idee), Benjamin Miller, Sami Chibane

(Programmierung, Interfacedesign)

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1986 in Paris (FR), lebt und arbeitet in Karlsruhe (DE)
*1995 in Échirrolles (FR), lebt und studiert in Grenoble (FR)

011 MarkowKetten Explorer

2017, Interaktive Klanginstallation, Computer, Maus, Bildschirm

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Dan Wilcox

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1981 in Orange (US), lebt und arbeitet in Karlsruhe (DE)

012 MusiCode

2017, Interaktive Klanginstallation, Computer, Maus, Bildschirm; Produktion des ZKM_Hertz-Lab

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Chandrasekhar Ramakrishnan, Götz Dipper

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1975, lebt und arbeitet in Zürich (CH)
*1966 in Stuttgart (DE), lebt und arbeitet in Karlsruhe (DE)

013 Pattern Machine

2004, Interaktive Klanginstallation

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Chandrasekhar Ramakrishnan, Götz Dipper

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1975, lebt und arbeitet in Zürich (CH)
*1966 in Stuttgart (DE), lebt und arbeitet in Karlsruhe (DE)

014 Random Machine

2004, Interaktive Klanginstallation

#Codierung

#Klangprogrammierung #Software
#Interface

Ludger Brümmer, Anton Himstedt (Idee), Chikashi Miyama, Alex Rodrigues (Softwareentwicklung)

*1958 in Werne (DE), lebt und arbeitet in Karlsruhe (DE)
*1952 in Wiesbaden (DE), lebt und arbeitet Geisenheim (DE)
*1979 in Otsu (JP), lebt und arbeitet in Karlsruhe (DE)
*1993 in Covilhã (PT), lebt und arbeitet in Castelo Branco (PT)

015 Rotating Scores

2016, Interaktive Klanginstallation

#Codierung

#Klangprogrammierung #Software #Interface

Butternutten AG

Oliver-Selim Boualam, *1992 in Stühlingen (DE), lebt und arbeitet in Karlsruhe (DE) und Marrakesch (MA) Lukas Marstaller, *1993 in Aalen (DE), lebt und arbeitet in Karlsruhe (DE) und Marrakesch (MA)

016 PLAY

2016, Lackiertes MDF, Holzbalcken, Plexiglas, Tischtennisnetz, 274 x 152 x 76 cm; Das Projekt entstand in Kollaboration mit Louis Kohlmann (Projektraum LOTTE - Land of the Temporary Eternity, Suttgart)

#Codierung

#Arbeit4.0

Can Büyükerberber, Yagmur Uyanik

*1987 in Izmir (TR), lebt und arbeitet in San Francisco (US) *1992 in Antalya (TR), lebt und arbeitet in San Francisco (US)

017 Morphogenesis

2016, Virtual-Reality-Installation, 3 Drucke, jeweils 91,5 x 114,3 cm

#VirtuelleRealität

#Codierung

#Eskapismus #ComputersimulierteUmgebungen #HMD

C

Emma Charles

*1985 in London (GB), lebt und arbeitet in London

018 White Mountain

2016, 16 mm-Film auf HD-Video, Farbe, Ton, 20 Min.

#AlgorithmicGovernance

#BigData

Matthieu Cherubini

*1984 in Aigle (CH), lebt und arbeitet in Shanghai (CN)

019 Ethical Autonomous Vehicles

2014, Touchscreen, 3 Drucke, jeweils 42 x 59,4 cm

#MaschinellesLernen

#KünstlicheIntelligenz #Autonome Systeme #SelbstfahrendeAutos

Tyler Coburn

*1983 in New York (US), lebt und arbeitet in New York

020 NaturallySpeaking

2013-2014, Mixed-Media-Installation, Text, Bildschirmschoner, Monitore, Möbel

#MaschinellesLernen

#Mustererkennung #Interface

Max Cooper, Andy Lomas

*1980 in Belfast (GB), lebt und arbeitet in London (GB) *1967 in Welwyn Garden City (GB), lebt und arbeitet in London (GB)

021 Chromos

2017, HTC Vive, Unreal Engine-Software

#VirtuelleRealität

#GenetischerCode

#HMD #Computersimulierte Umgebungen #DNA

Shane Cooper

*1964 in Yorba Linda (US), lebt und arbeitet in Neuseeland (NZ)

022 Remote Control

1999, Interaktive Netzwerkinstallation

#AlgorithmicGovernance

#Binär

Larry Cuba

*1950 in Atlanta (US), lebt und arbeitet in Santa Cruz (US)

023 3/78 (Objects and Transformations)

1978, 16 mm-Film auf HD-Video, schwarz-weiß, 6 Min.

Two Space

1979, 16 mm-Film auf HD-Video, schwarz-weiß, 8 Min.

Calculated Movements

1985, 16 mm-Film auf HD-Video, schwarz-weiß, 6 Min.

#Codierung

#Algorithmus

024 Animation Notebook 2010

2010, generative Animation,
Video, 70 Min.

Animation Notebook 2012

2012, generative Animation,
Video, 13 Min.

#Codierung

#Algorithmus

#ComputerGeneriertesDesign

D

Frederik De Wilde

*1975 in Brüssel (BE), lebt und
arbeitet in Brüssel

025 RzI-DzI-AI

2016, HD-Video, Farbe, Ton,
7:46 Min.

#MaschinellesLernen

#AlgorithmicGovernance

#KünstlicheIntelligenz

#Mustererkennung #Drohnen

Simon Denny

*1982 in Auckland (NZ), lebt und
arbeitet in Berlin (DE)

026 Blockchain Future States

2016, Mixed-Media-Installation,
Digitaldrucke, HD-Video, 3 Min.;
Mit freundlicher Genehmigung
der Galerie Buchholz, Berlin /
Köln / New York

#AlgorithmicGovernance

#AlgorithmischeÖkonomie

#Bitcoin #Kryptowährungen

#Blockchain

Götz Dipper

*1966 in Stuttgart (DE), lebt und
arbeitet in Karlsruhe (DE)

027 Add_Synth

2017, Interaktive Klangins-
tallation, Computer, Software,
Bildschirm, Maus, Kopfhörer;
Produktion des ZKM_Hertz-Lab

028 algoRhythm Machine

2017, Interaktive Klanginstal-
lation, Computer, Bildschirm,
Maus, Kopfhörer; Produktion des
ZKM_Hertz-Lab

029 FM_Synth

2017, Interaktive Klanginstal-
lation, Computer, Bildschirm,
Maus, Kopfhörer;
Produktion des ZKM_Hertz-Lab

030 ...wie der Computer Musik macht

2017, Interaktive Klanginstal-
lation, Computer, Bildschirm,
Maus, Kopfhörer; Idee: Peter
Weibel; Beratung: Ludger
Brümmer, Benjamin Miller;
Produktion des ZKM_Hertz-Lab

#Codierung

#Klangprogrammierung #Software
#Interface

Harm van den Dorpel

*1981 in Zaandam (NL), lebt und
arbeitet in Berlin (DE)

031 Death Imitates Language

2016/2017, Website, zwei Drucke
(einmalige Auflage), jeweils
100 × 100 cm, 70 × 70 cm

#MaschinellesLernen

#GenetischerCode

#Codierung

#Mustererkennung #Autonome-
Systeme #Algorithmus #Software

Constant Dullaart

*1979 in Leiderdorp (NL), lebt und
arbeitet in Amsterdam (NL)

032 DullDream

2015, Neuronales Netzwerk;
Mit freundlicher Genehmigung
von DullTechTM

#MaschinellesLernen

#Mustererkennung #Software

E

Margret Eicher

*1955 in Viersen (DE), lebt und
arbeitet in Berlin und Mannheim (DE)

033 Das Große Rasenstück

2013, Tapisserie, digitale
Collage, Jacquardgewebe,
275 × 425 cm

#Arbeit&Produktion

#ComputerGeneriertesDesign
#Programmieren

Jonas Eltes/Fabrica

*1993 in Kungsbacka (SE), lives
and works in Treviso (IT)

034 Lost in Computation

2017, Mixed-Media-Installation,
2 Displays, 2 Raspberry Pi-
Computer; Mit freundlicher
Genehmigung der Fabrica, Catena

di Villorba

#MaschinellesLernen

#Codierung

#Mustererkennung #Autonome-Systeme

2017, UV-Direktdruck auf Plexiglas, Schaumstoff, LED-Leuchten, 210 × 65 × 15 cm

#MaschinellesLernen

#Mustererkennung

César Escudero Andaluz, Martín Nadal

*1983 in Ávila (ES), lebt und arbeitet in Linz (AT)

*1978 in Madrid (ES), lebt und arbeitet in Linz (AT)

035 BitterCoin

2016, Installation, Tischrechner

#AlgorithmischeÖkonomie

#Bitcoin #Kryptowährungen #Blockchain

Fraunhofer IOSB und ZKM | Karlsruhe

039 Autonome Fahrzeuge

2017, HD-Video, Ton, Farbe, ca. 8 Min.

#MaschinellesLernen

#Arbeit&Produktion

#AutonomeSysteme #Drohnen

#Roboter #Industrie4.0

040 Industrie 4.0

2017, HD-Video, Ton, Farbe, ca. 8 Min.

#Arbeit&Produktion

#MaschinellesLernen

#AlgorithmicGovernance

#InternetDerDinge #SmartFactories #Automation #Industrie4.0

Claire L. Evans

*1984 in Swindon (GB), lebt und arbeitet in Los Angeles (US)

036 2001 100011

2011, Drehbuch, 21,6 × 27,9 cm

#Codierung

#Binär

F

Harun Farocki

*1944 in Nový Jičín (CZ),

†2014 in Berlin (DE)

037 Parallele

2012-2014, 4 Videos

Parallele I

2012, 2-Kanal-Videoinstallation, Farbe, Ton, 16 Min.

Parallele II

2014, 1-Kanal-Videoinstallation, Farbe, Ton, 9 Min.

Parallele III

2014, 2-Kanal-Videoinstallation, Farbe, Ton, 7 Min.

Parallele IV

2014, 1-Kanal-Videoinstallation, Farbe, Ton, 11 Min.

Harun Farocki GbR, Berlin

#VirtuelleRealität

#Codierung

#ComputersimulierteUmgebungen

FZI Forschungszentrum Informatik am Karlsruher Institut für Technologie (KIT)

gegründet 1985 in Karlsruhe (DE)

041 The Human Brain Project

2017, HD-Video, Ton, Farbe, ca. 8 Min.

#GenetischerCode

#Codierung

#Computing #Roboter

G

Kristof Gavrielides

*1973 in Köln (DE), lebt und arbeitet in Stuttgart (DE) und Paris (FR)

042 Spatial Code Lab

2017, Mixed-Media-Installation;

Sponsoren: ZKM | Zentrum für

Kunst und Medien Karlsruhe,

Ministerium für Wissenschaft,

Forschung und Kunst Baden-

Württemberg, BW-Stipendium /

Cité des Arts Paris, msa /

mediaspaceagency, sam / studio-

advancedmedia

#VirtuelleRealität

#Arbeit&Produktion

#Codierung

#Roboter #Algorithmus

#Software #Hardware

Thierry Fournier

*1960 in Oullins (FR), lebt und arbeitet in Paris (FR)

038 Oracles

#ComputersimulierteUmgebungen
#Automation

Melanie Gilligan

*1979 in Toronto (CA), lebt und arbeitet in New York (US) und London (GB)

043 *The Common Sense*

Phase 1, 5 Episoden
2014/2015, 5-Kanal-Videoinstallation, ortsspezifische Installation, 5 LED-TV-Bildschirme, pulverbeschichtete Stahlrohre, Funkkopfhörer, HD-Video, Farbe, Ton, je 6-7 Min.; Julia Stoschek Foundation e.V., Düsseldorf

#AlgorithmicGovernance

#Eskapismus #Quantifiziertes-Selbst

Fabien Giraud, Raphaël Siboni

*1980 in Caen (FR), lebt und arbeitet in Paris (FR)
*1981 in Romorantin-Lathenay (FR), lebt und arbeitet in Paris (FR)

044 *The Unmanned*

1997 - *The Brute Force* (2014)
1759 - *Mil Troi Cens Quarante Huyt* (2017)
2-Kanal-Videoinstallation, HD, Farbe, Ton, Loop, 26 Min.; Mit Unterstützung des Casino Luxembourg - Forum d'Art Contemporain, des Palais de Tokyo und von Le Fresnoy, Studio national des arts contemporains.

#GenealogieDesCodes

H

Daniel Heiss

*1978 in München (DE), lebt und arbeitet in Karlsruhe (DE)

045 *KryptoLab*

2017, Bitcoin ASIC Miner, diverse Computer, *Times* Zeitschrift (mit freundlicher Genehmigung der Universitätsbibliothek Regensburg)

#AlgorithmischeÖkonomie

#AlgorithmicGovernance

#Bitcoin #Kryptowährungen
#Blockchain

046 *S2T2T2M2L*

2017, Computer, Bildschirme,

LED-Strip

#MaschinellesLernen

#Codierung

#Morsecode #Binär #Algorithmus
#Software

Yannick Hofmann

*1988 in Offenbach am Main (DE), lebt und arbeitet in Karlsruhe (DE)

047 *Monocause. Dialectics of the Post-Truth Era*

2017, Interaktive Klanginstallation, iOS-App; Illustration und Produktionsassistentz: Fiona Marten

#Codierung

#AlgorithmicGovernance

#Binär

I

Simon Ingram

*1971 in Wellington (NZ), lebt und arbeitet in Auckland (NZ)

048 *Looking for the Waterhole*

2017, Installation, malende Maschine

#Arbeit&Produktion

#Industrie4.0 #Roboter #Computing

ICD, ITKE, ITECH

Institut für Computerbasiertes Entwerfen und Baufertigung (ICD) und Institut für Tragkonstruktionen und Konstruktives Entwerfen (ITKE) in Zusammenarbeit mit Integrative Technologies and Architectural Design Research Program (ITECH) an der Universität Stuttgart

049 *ICD/ITKE Research Pavilion 2016-17*

2016-2017, Struktur aus Glas- und Carbonfasern; ICD Institut für Computerbasiertes Entwerfen und Baufertigung (Prof. Achim Menges), ITKE Institut für Tragkonstruktionen und Konstruktives Entwerfen (Prof. Jan Knippers), Universität Stuttgart

#MaschinellesLernen

#Arbeit&Produktion

#Industrie4.0 #Mustererkennung

#AutonomeSysteme #Drohnen
#Roboter #Automation

Institut für Theoretische Informatik, DebateLab, KIT

Gregor Betz, *1976 in Peine (DE); Michael Hamann, *1988 in Mühlacker (DE); Tamara Mchedlidze, *1981 in Tbilisi (GE); Sophie von Schmettow, *1992 in Aachen (DE); Christian Voigt, *1979 in Hamburg (DE); alle leben und arbeiten in Karlsruhe (DE)
050 *OpinionMap: What Should One Eat?* 2017, Software; KIT, Institut für Theoretische Informatik, DebateLab

#Codierung

#AlgorithmicGovernance

#BigData #Algorithmus #Software
#Interface #QuantifiziertesSelbst

K

Eduardo Kac

*1965 in Rio de Janeiro (BR), lebt und arbeitet in Chicago (US)

051 *Transcription Jewels*

2001, Objekte, Glas, gereinigte „Genesis“-DNA, Gold, Holz

#GenetischerCode

#DNA #Phänotyp #DNADatenspeicher
#Genotyp

Helen Knowles

*1975 in London (GB), lebt und arbeitet in London und Manchester (GB)

052 *The Trial of Superdebthunterbot*

2016, Installation, HD-Video, Farbe, Ton, 45 Min., Geschworenenbank aus laminiertem Birkenperrholz und Kunstleder, 5 Zeichnungen

#AlgorithmicGovernance

#MaschinellesLernen

#Algorithmus #Mustererkennung
#BigData #KünstlicheIntelligenz

Beryl Korot

*1945 in New York (US), lebt und arbeitet in New York

053 *Babel 1*

1980, Pigment auf handgewebtem Leinen, fotografische Reproduktion, 77,7 × 58,9 × 6,4 cm;

Mit freundlicher Genehmigung der bitforms gallery, New York

Babel 2

1980, Pigment auf handgewebtem Leinen, fotografische Reproduktion, 183 × 98,4 cm; Mit freundlicher Genehmigung der bitforms gallery, New York

#Codierung

#Decodierung #Babel

Anton Kossjanenko

*in Kerch (SU), lebt und arbeitet in Karlsruhe (DE)

054 *Sacrophonie*

2017, Interaktive Klanginstallation; Programmierung: Alexandre Rodrigues

#Codierung

#Klangprogrammierung #Interface

Brigitte Kowanz

*1957 in Wien (AT), lebt und arbeitet in Wien

055 *Morsealphabet*

1998, Lichtinstallation

#Codierung

#Decodierung #Morsecode

L

Marc Lee

*1969 in Knutwil (CH), lebt und arbeitet in Eglisau (CH)

056 *The Show Must Go On.*

2017 fortlaufend, Online-Nachrichtensender

#AlgorithmicGovernance

#BigData

Jan Robert Leegte

*1973 in Assen (NL), lebt und arbeitet in Amsterdam (NL)

057 *Portrait of a Web Server*

2013, JavaScript, HTML, CSS, Apache HTTP Server-Quelltext (programmiert in C)

#Codierung

#Software #Quellcode #Computing

Donna Legault

*in Ottawa (CA), lebt und arbeitet in Ottawa und Montreal (CA);

Die Künstlerin ist Teil des internationalen Netzwerkes Hexagram. Das in Montreal (CA) gegründete Kollektiv widmet sich der Forschung in den Bereichen Medienkunst, Design, Technologie und digitaler Kultur und besteht aus über 80 Mitgliedern.

058 Drone

2017, 2-Kanal-Videoinstallation, Farbe, Ton, Loop, 5 Min.; Sponsoren: Hexagram, Milieux: Institute for Arts, Culture and Technology

#Arbeit&Produktion

#Industrie4.0 #Drohnen
#Automation #Programmieren

Lawrence Lek

*1982 in Frankfurt a.M. (DE), lebt und arbeitet in London (GB)

059 Sinofuturism (1839-2046 AD)

2016, HD-Videoessay, 60 Min.

#MaschinellesLernen

#GenealogieDesCodes

#KünstlicheIntelligenz
#Computing

Armin Linke

*1966 in Mailand (IT), lebt und arbeitet in Berlin (DE)

060 Phenotypes/Limited Forms

2007, Interaktive Installation, Fotografien, RFID-Tags, 16 RFID-Lesegeräte, 2 Touchscreens, 2 PCs, 2 BOCA Micro Ticketdrucker, Thermopapier-Tickets, Videoprojektor

#Codierung

#Decodierung #Phänotyp
#Interface

Bernd Lintermann, Torsten Belschner, Mahsa Jenabi, Werner A. König

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe (DE)

*1966 in Freiburg i. B. (DE), lebt und arbeitet in Freiburg i. B.

*1982 in Teheran (IR)

*1978 Ravensburg (DE), lebt und arbeitet in Worms (DE)

061 CloudBrowsing: Open Codes

2009/2017, Interaktive Installation für den PanoramaScreen;

Gesamtkonzept, visuelles Konzept, Produktionsleitung, Realisierung: Bernd Lintermann; Audio-Konzept, Realisierung: Torsten Belschner; Interaktionsdesign, Realisierung: Mahsa Jenabi, Markus Nitsche, Werner A. König; Interface-Design: Matthias Gommel; Projektmanagement: Jan Gerigk, Petra Kaiser; Technische Realisierung: Manfred Hauffen, Jan Gerigk, Nikolaus Völzow, Arne Gräber, Joachim Tesch; Produktion: ZKM | Institut für Bildmedien; In Zusammenarbeit mit AG Mensch-Computer Interaktion, Universität Konstanz; Ein Forschungsprojekt im Rahmen des Forschungsverbundes „Information at your fingertips – Interaktive Visualisierung für Gigapixel Displays“; gefördert durch das Förderprogramm Informationstechnik des Landes Baden-Württemberg (BW-FIT).

#GenealogieDesCodes

#Computing #Interface

Bernd Lintermann, Nikolaus Völzow

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe (DE)

*1980 in Koblenz (DE), lebt und arbeitet in Karlsruhe (DE)

062 Drei Phasen der Digitalisierung

2017, Interaktive Installation mit polarisiertem Licht und Augmented-Reality-Technik; Idee: Peter Weibel; Konzept: Bernd Lintermann, Nikolaus Völzow; Softwareentwicklung: Nikolaus Völzow; Buchgestaltung: Jan Zappe; Design: Matthias Gommel; Technische Mitarbeit: Jan Gerigk, Manfred Hauffen

#Codierung

#GenealogieDesCodes

#Interface #Software

Bernd Lintermann, Jan Gerigk

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe (DE)

*1963 in Pforzheim (DE), lebt und arbeitet in Karlsruhe (DE)

063 Site Map: Open Codes

2017, Interaktive Augmented-Reality-Installation für iPad und HoloLens; Konzept, Projekt-
leitung: Bernd Lintermann, Jan Gerigk; Applikationssoftware: Bernd Lintermann; Produktions-
leitung: Jan Gerigk; Technik: Manfred Hauffen; Produktion: ZKM | Institut für Bildmedien und ZKM | Medienmuseum; Basierend auf der Augmented-Reality Installation Traffic, 2011

#Codierung

#Interface #AugmentedReality

Bernd Lintermann, Julia Gerlach, Peter Weibel

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe (DE)

*1967 in Hannover (DE), lebt und arbeitet in Frankfurt a. M. (DE) und Berlin (DE)

*1944 in Odessa (UA), lebt und arbeitet in Karlsruhe (DE)

064 SoundART IDEAMA

2012, Interaktive Augmented-Reality-Installation, AR Audio-Datenbankbrowser für iPad; Konzept: Bernd Lintermann, Julia Gerlach, Peter Weibel; Kurator: Hartmut Jörg; Software: Bernd Lintermann; Technische Koordination: Manfred Hauffen; Produktion: ZKM | Institut für Bildmedien

#Codierung

#Klangprogrammierung #Interface, #AugmentedReality

Bernd Lintermann, Manfred Hauffen, Peter Weibel

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe

*1956 in Karlsruhe (DE), lebt und arbeitet in Karlsruhe

*1944 in Odessa (UA), lebt und arbeitet in Karlsruhe (DE)

065 SynSeeThis

2013, iOS-App für iPad; Idee, Konzeption, Software: Bernd Lintermann; Performance: Peter Weibel; Technische Unterstützung: Manfred Hauffen; Klang: Manfred Hauffen, Hartmut Bruckner; Produktion: ZKM | Institut für Bildmedien

#Codierung

#Klangprogrammierung #Interface

Bernd Lintermann

*1967 in Düsseldorf (DE), lebt und arbeitet in Karlsruhe (DE)

066 VRMe

2017, Interaktive Installation für VR-Brille; Produktion des ZKM_Hertz-Lab

#VirtuelleRealität

#Eskapismus #HMD

#ComputersimulierteUmgebungen

067 YOU:R:CODE

2017, Interaktive Installation mit Mehrkanalprojektion; Idee: Peter Weibel; Konzept, Realisierung: Bernd Lintermann; Audiodesign: Ludger Brümmer, Yannick Hofmann; Flip-Dot-Display: Christian Lölkes; Technische Unterstützung: Manfred Hauffen, Jan Gerigk; Aufbau, Planung: Thomas Schwab; Produktion des ZKM_Hertz-Lab

#Codierung

#GenetischerCode

#Decodierung #Software

#Hardware #Interface #Quellcode

#QuantifiziertesSelbst

Fei Liu

*1986 in Harbin (CN), lebt und arbeitet in New York (US)

068 The Qualified Life

2014, Interaktive Installation; In Kooperation mit der Akademie Schloss Solitude

#Arbeit&Produktion

#Industrie4.0 #SmartFactories

#BigData #QuantifiziertesSelbst

#Interface #Arbeit4.0

Christian Lölkes

*1990 in White Plains (US), lebt und studiert in Karlsruhe (DE)

069 Arcibo-Nachricht

2017, Installation

#GenealogieDesCodes

#Codierung

#Decodierung

070 Code Styles

2017, Installation

#Codierung

#Software

071 Codierte Informationen

2017, Installation

#Codierung

#Decodierung

072 Game of Life

2017, Installation

#Codierung**#AlgorithmicGovernance****#GenealogieDesCodes****073 Sound of Sorting**

2017, Installation

#Codierung

#Algorithmus

Solimán López

*1981 in Burgos (ES), lebt und arbeitet in Valencia (ES)

074 Column 1-02016-2017, Videoinstallation, Digitaldruck auf Leinwand, Bild-
daten, Videoprojektion, Loop,
ca. 120 × 200 cm; Assistenz:
Toni Vaca**#Codierung**

#Interface #Zahlensystem

M**Shawn Maximo**

*1975 in Toronto (CA), lebt und arbeitet in New York (US)

075 Open Doors

2017, Digitaldruck

#Arbeit&Produktion

#Arbeit4.0 #Automation

#Programmieren #Algorithmus

#Software

Rosa Menkman

*1983 in Arnhem (NL), lebt und arbeitet in Berlin (DE)

076 DCT:SYPHONING.The 1000000th (64th) Interval

2017, Virtuelle 3-D-Umgebung

#VirtuelleRealität**#Codierung**

#Eskapismus #HMD

#ComputersimulierteUmgebungen

#Binär

Chikashi Miyama

*1979 in Otsu (JP), lebt und arbeitet in Karlsruhe (DE)

077 Rhythm of Shapes2016, Interaktive Klanginstal-
lation**078 Sonorama - Karlsruhe**2017, Klanginstallation,
ZKM_PanoramaLabor**#Codierung**#Klangprogrammierung #Software
#Interface**Andreas Müller-Pohle**

*1951 in Braunschweig (DE), lebt und arbeitet in Berlin (DE)

079 Blind Genes2002, Digitaler Lambda-Print,
Cibachrome auf Aluminium unter
Acrylglas, 11 × 100 × 5 cm,
45 × 100 × 5 cm; Bettina and
Thomas Hebell**#GenetischerCode****#Codierung**

#DNA #Quellcode #Genotyp

080 Digitale Partituren I (nach Nicéphore Niépce)1995, 8 Tafeln, Iris 3047 Ink-
jet-Print auf Aquarell-Papier
(Arches grain Satiné 300 g/m²),
je 68,5 × 68,5 × 4 cm**#GenealogieDesCodes**

#ComputerGeneriertesDesign

Jörn Müller-Quade

*1967 in Darmstadt (DE), lebt und arbeitet in Karlsruhe (DE)

081 Code beautiful like a clock2017, Monitor, Software,
Schablone; KASTEL (Kompetenz-
zentrum für angewandte
Sicherheitstechnologie), KIT
(Karlsruher Institut für Tech-
nologie); Wissenschaftliche
Mitarbeit: Jeremias Mechler**#GenealogieDesCodes****#Codierung**

#Interface

N**Greg Niemeyer**

*1967 in der Schweiz, lebt und arbeitet in Berkeley (US)

082 Sonic Web Instrument2017, Code in JavaScript,
Touchscreen, Ton**#AlgorithmicGovernance****#Codierung**

#Algorithmus #Software
#Klangprogrammierung

Helena Nikonole

*1982 in Moskau (RU), lebt und arbeitet in Moskau

083 *deus X mchn*
2017, Multimedia-Installation

#MaschinellesLernen

#AlgorithmicGovernance

#KünstlicheIntelligenz
#InternetDerDinge

P

Julian Palacz

*1983 in Leoben (AT), lebt und arbeitet in Wien (AT)

084 *Fragmentierungen*
2015, Glasgravierungen auf Festplatten

#Codierung

#Interface #Hardware

Matthew Plummer-Fernandez

*1982 in London (GB), lebt und arbeitet in London

085 *Vertigo in the Face of the Infinite*
2017 fortlaufend, Webanwendung und Onlineshop, 3-D-Druck-Plastikfiguren, 3-D-Drucke, Tablets mit Ständern, Monitore

#Codierung

#Computing #Software
#Interface #BigData
#ComputerGeneriertesDesign

Julien Prévieux

*1974 in Grenoble (FR), lebt und arbeitet in Paris (FR)

086 *Patterns of Life*
2015, HD-Video, Farbe, Ton, 15:30 Min.

#MaschinellesLernen

#AlgorithmicGovernance

#Codierung

#Mustererkennung
#QuantifiziertesSelbst

087 *What Shall We Do Next?*
(Sequence #2)

2014, HD-Video, Farbe, Ton, 16:47 Min.

#MaschinellesLernen

#Codierung

#Mustererkennung

R

Peter Reichard, Manfred Kraft, Michael Volkmer

*1969 in Mainz (DE), lebt und arbeitet in Wiesbaden (DE)

*1966 in Heidelberg (DE), lebt und arbeitet in Berlin (DE)

*1965 in Augsburg (DE), lebt und arbeitet in Wiesbaden (DE)

088 *NOx STADT_LUFT_ANZEIGER*
2016-2017, Outdoor: LED-Installation im urbanen Raum; Indoor: Interaktives LED-Display; Mitarbeit: Tom Kresin

#Codierung

#Software #Binär

Matthias Richter, Josef N. Patoprsty

*1986 in Bonn (DE), lebt und arbeitet in Karlsruhe (DE)

*1987 in Wien (AT), lebt und arbeitet in Austin (US)

089 *Die Leidmaschine*
2017, PC, Arduino, Software: Python (OpenCV, DLib), Lua (LÖVE)

#MaschinellesLernen

#Codierung

#KünstlicheIntelligenz
#Mustererkennung
#AutonomeSysteme #Interface

Betty Rieckmann

*1986 in Palo Alto (US), lebt und arbeitet in Karlsruhe (DE)

090 *Silent Communications*
2017, Ortsspezifische Installation, LED-Leuchten, Smartphone-App

#Codierung

#GenealogieDesCodes

#Decodierung #Software
#Interface #Morsecode #Computing
#Babel

robotlab

gegründet 2000 von Matthias Gommel, Martina Haitz und Jan Zappe, tätig

in Karlsruhe (DE)

091 **manifest**

2008/2017, Industrieroboter, Schreibpult, Computer, Software, Papier, Stift; angeregt durch Peter Weibel

#MaschinellesLernen

#Industrie4.0 **#Künstliche-Intelligenz** **#Roboter** **#Automation**

Curtis Roth

*1986 in Portland (US), lebt und arbeitet in Columbus (US)

092 **Real Time**

2017, Video-Livestream

#AlgorithmicGovernance

#BigData **#QuantifiziertesSelbst**
#Arbeit4.0

RYBN.ORG

Künstlerkollektiv, gegründet 1999 in Paris (FR), die Künstler leben und arbeiten in Paris

093 **ADM XI**

2017, Multimedia-Installation; An der Plattform **ADM XI** sind beteiligt: b01, Femke Herregraven, Brendan Howell, Martin Howse, JoDi, Nicolas Montgermont, Horia Cosmin Samoila, Antoine Schmitt, Marc Swynghedauw, Suzanne Treister. **ADM XI** ist Teil der 2006 begonnenen Werkserie **Antidatamining**. Das Projekt ist der dritte und letzte Teil der von RYBN.ORG 2011 mit **ADM 8** initiierten und mit **ADM X: The Algorithmic Trading Freakshow** 2013 weitergeführten Trilogie zum algorithmenbasierten Finanzwesen. **ADM XI** wird von Inke Arns kuratiert und ist eine Koproduktion mit Jeu de Paume (Paris), mit Unterstützung von DICRéaAM, Labomedia und Espace Multimedia Gantner.

#AlgorithmischeÖkonomie

#AlgorithmicGovernance

#Hochfrequenzhandel

S

saai, KIT

Südwestdeutsches Archiv für Architektur und Ingenieurbau, Karlsruher

Institut für Technologie

094

Dokumente zu Frei Ottos Multi-halle Mannheim 1974-1975, Plan: digitaler Print; Bücher: digitalisierte Bücher; saai | Südwestdeutsches Archiv für Architektur und Ingenieurbau, Karlsruhe

#Codierung

#GenealogieDesCodes

Chris Salter

*1967 in Beaumont (US), lebt und arbeitet in Montreal (CA) und Berlin (DE); Der Künstler ist Teil des internationalen Netzwerkes Hexagram. Das in Montreal (CA) gegründete Kollektiv widmet sich der Forschung in den Bereichen Medienkunst, Design, Technologie und digitaler Kultur und besteht aus über 80 Mitgliedern.

095 **N-Polytope: Behaviors in Light and Sound After Iannis Xenakis**

2012/2017, Stahlseile, Mikroelektronik, LED-Lampen, Lautsprecher, Software; In Zusammenarbeit mit: Sofian Audry, Adam Basanta, Marije Baalman, Elio Bidinost, Thomas Spier

#MaschinellesLernen

#Codierung

#KünstlicheIntelligenz **#Software**

Karin Sander

*1957 in Bensberg (DE), lebt und arbeitet in Berlin (DE) und Zürich (CH)

096 **XML-SVG CODE / Quellcode des Ausstellungsraums**

2010/2017, Oracal 638, Plotterfolie matt, 3-farbig

#Codierung

#Quellcode **#Computing**

Signal- und Maschinencodes

Block 1: Morsetelegrafie

097 Morsetaste von Siemens & Halske, Berlin, lautlose Ausführung 1890; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

098 Morsetelegraf, deutscher

Normalschreiber mit Papierlade,
Modell von Siemens & Halske,
Berlin
1870; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 099** Franz Schmid, *Die Telegraphen-Alphabete und Zeichen Österreichs in ihrer Historischen Entwicklung*, Wien
1891; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 100** Seekabelempfänger von Lauritzen, „Große Nordische Telegraphen Company“ Kopenhagen
ca. 1890; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 101** Morse-Knatterfunktensender mit Induktionsspule der Firma Max Kohl, Chemnitz; Oszillator-Funkenstrecke nach Heinrich Hertz (Selbstbau) und Morsetaste
ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 102** Morse-Kohärer-Empfänger mit Empfangsklingel (Selbstbau mit historischen Bausteinen)
ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 103** Morse-Farbschreiber-Klopfer, Demonstrationsmodell. Das Modell beinhaltet: Farbschreiber von Siemens & Halske, Berlin; Farbschreiber von Gebr. Naglo, Berlin; Morsetaste; Umschalter; Klopfergehäuse mit Klopfer; Museum für Kommunikation, Frankfurt am Main

#GenealogieDesCodes

#Morsecode

Block 2: Signalcodes

1. Werke zur Schaltalgebra

- 104** Erich Hochstetter, *Herrn von Leibniz' Rechnung mit Null und Eins*, Siemens-Aktiengesellschaft, Berlin
1966; Mit freundlicher Genehmigung von Peter Weibel

#GenealogieDesCodes

#Zahlensystem #Computing #Binär

- 105** George Boole, *The Mathematical Analysis of Logic*, Basil

Blackwell, Oxford
1969 (Reprint); Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Zahlensystem #Computing #Binär

- 106** Claude E. Shannon, „The Synthesis of Two-Terminal Switching Circuits“, in: *The Bell System Technical Journal*, 28,1,
S. 59-98
Januar 1949; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Zahlensystem #Computing #Binär

- 107** *Rudolf Mosse-Code Buch*, Verlag Rudolf Mosse, Berlin ohne Jahr; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Morsecode

2. Speicherbausteine

- 108** Lochkarten des EDV-Zentrums der Universität Linz für einen Philipps Bürocomputer und Lochkarten aus der Bibliothek des IBM Laboratorium Wien
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Software

- 109** Impulswiederholer, Telefon-Weitverkehrstechnik von Siemens & Halske, Berlin
1951; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 110** Matrizen mit Ferritspeicherringen
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 111** Ringkernspeicher-Block aus einem IBM 705 Computer, USA
1955; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 112** Ringkernspeicher - Platine von Nixdorf Computer AG, Paderborn
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 113** Magnetband, Multi-System-Tape von IBM
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 114** Magnetplattenstation
1970er-Jahre; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 115** Festplatte aus einem PC
Mit freundlicher Genehmigung von
Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 116** Festplatte eines Laptops von
Toshiba, 80 GB
Mit freundlicher Genehmigung von
Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 117** Röhren-Flip-Flop einer IBM 650
ca. 1958; Mit freundlicher
Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 118** Röhren-Flip-Flop einer ZUSE Z22
ca. 1958; Mit freundlicher
Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 119** Optische Speicherscheibe,
Compact Disc (CD)
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 120** Mikroelektronischer Halbleiter-
speicher, USB-Stick
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

3. Vermittlungsbausteine

- 121** Drehrelais aus der Telefon-
Wähltechnik
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 122** Strowger-Wähler mit Relaisatz
von Mix & Genest, Stuttgart
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 123** Kreuzschienenschalter für
Telegrafie von Western Electric
Inc., USA
ZKM | Karlsruhe

#GenealogieDesCodes

#MorseCode

- 124** Mini-Koordinatenschalter,
Telefonnebenstellentechnik,

ITT Corporation

Mit freundlicher Genehmigung
von Franz Pichler**#GenealogieDesCodes**

#Computing #Hardware

- 125** ESK-Koppelfeld von Siemens
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

4. Codierbausteine

- 126** Relaisatz für ein Telefon aus
der Wählvermittlungstechnik
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 127** Relaisplatte eines elektromag-
netischen Schaltnetzes
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 128** Platine mit mikroelektronischen
Bausteinen
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

**Block 3: ENIGMA-Maschine und
Zeigertelegrafie**

- 129** ENIGMA K Verschlüsselungs-
maschine erzeugt für Schweizer
Armee
ca. 1939; Karlsruher Institut
für Technologie, Kompetenzzent-
rum für angewandte Sicherheit-
stechnologie (KASTEL)

#GenealogieDesCodes

#Babel

- 130** Johannes Trithemius, *Polygra-
phia*, Frankfurt
1550; ZKM | Karlsruhe

#GenealogieDesCodes

- 131** Giambattista della Porta, *De
Furtivis Literarum Notis*, Neapel
1563; ZKM | Karlsruhe

#GenealogieDesCodes

- 132** Claude E. Shannon, „Communica-
tion Theory of Secrecy Systems“,
in: *The Bell System Technical
Journal*, 28, 4, S. 656-715
Oktober 1949; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing

- 133** Geber zum Zeigertelegrafen der
Firma Breguet, Paris
ca. 1855; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

- 134** Empfänger zum Zeigertelegrafen der Firma Breguet, Paris ca. 1855; ZKM | Karlsruhe

#GenealogieDesCodes

#Morsecode

Block 4: Mechanische Automaten, Androiden und logische Maschinen

- 135** P. Gaspare Schotto, *Magiae Universalis. Naturae et Artis, Pars II Acustica, Bambergae, MDCLXXIV* 1674

Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

- 136** Friedrich von Knauß, *Selbstschreibende Wundermaschinen*, Wien

1780; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 137** Siegfried Richter, *Wunderbares Menschenwerk*, Edition Leipzig 1989; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 138** Alfred Chapuis, Edmond-Droz, *Mechanical Puppets*, Neuchâtel (mit Originalschreiben des „Schreibers“) 1956; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 139** M. Raymundus Lullus, *Ars Magna, generalis et ultima Francofurti MDXCVI*

1596; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 140** Alexander Fidora, Carles Sierra (Hg.), *Ramon Lull, From the Ars Magna to Artificial Intelligence*, Barcelona

2011; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

Block 5: Zählwerke und mechanische Rechner

1. Zählwerke

- 141** Mechanisches dekadisches Zählwerk
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Zahlensystem #Computing

#Hardware

- 142** Zählmagnet 53 von Standard Elektrik Lorenz AG, Stuttgart 1953; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 143** Mechanische Zahnradwerk mit elektromagnetischem Zähler, vermutlich aus einer Büromaschine
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 144** Elektromagnetischer Zählerbaustein
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 145** Elektronischer Zählerbaustein mit Philips Zählröhre E1T von VEB Vokutronik Dresden
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

2. Mechanische Rechner

- 146** Rechenwalze von Albert Nestler AG, Lahr
1,6 m; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 147** Kugelplanimeter der Firma Coradi, Zürich
ca. 1910; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 148** „Air Data Computer“, elektromechanischer Analogrechner der US-Luftwaffe
ca. 1965; ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 149** Mechanische Rechenmaschine Comptometer von Felt & Tarrant, Chicago

ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

- 150** Mechanischer Hebelrechner, *Consul the Educated Monkey* kennt das Kleine Einmaleins Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 151** Mechanische Rechenmaschine Mercedes-Euklid Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

- 152** Mechanische Rechenmaschine CURTA II von Contina Ltd., Mauren, Liechtenstein 1948; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

Block 6: Historische Werke zu Maschinencodes, Automaten und Programmierung

- 153** C. E. Shannon, J. McCarthy (Hg.), *Automata Studies*, Princeton University Press, Princeton (NJ) 1956; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 154** Carl Adam Petri, *Kommunikation mit Automaten*, Dissertation, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn 1962; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 155** John von Neumann, *Theory of Selfreproducing Automata*, University of Illinois Press, Urbana (IL) 1966; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 156** Stephen Wolfram, *Theory and Applications of Cellular Automata*, World Scientific Publishing Co., Singapur 1986; Mit freundlicher

Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 157** Konrad Zuse, *Rechnender Raum*, Springer Fachmedien Wiesbaden GmbH 1969 (Reprint); ZKM | Karlsruhe

#GenealogieDesCodes

#Computing

- 158** Benoît Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York (NY) 1977; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 159** Konrad Zuse, *The Plankalkül*, Gesellschaft für Mathematik und Datenverarbeitung Bonn, St. Augustin-Birlinghoven 1976; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 160** Edsger W. Dijkstra, *A Primer of ALGOL 60 Programming*, Academic Press, London 1962; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 161** Heinz Rutishauser, *Description of ALGOL 60*, Springer-Verlag, Heidelberg 1967; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Babel

- 162** Heinz Zemanek et al., *Extension of the Algorithmic Language ALGOL*, Mailüfterl - Vienna, Final Report 31. Juli 1961; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

- 163** P. Lucas, *Introduction to the Method used for the formal Definition of PL/1*, Technical Report, IBM Laboratory, Vienna 1967; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

Block 7: Mikroelektronik und Digitalisierung

1. Halbleiterbausteine

164 Entwicklung vom Relais bis zur integrierten Schaltung, Nixdorf Computer AG, Paderborn
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

165 Schaustück Reinsilizium, Wacker Chemie AG, Burghausen
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

166 Silizium-Wafer mit Prozessorchips, Siemens Forschung ZTE, München
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

167 Analog-Digitalwandler in Mikrochiptechnologie, angefertigt von Siemens München als „Münchner Frauenkirche“ anlässlich des IFIP Congress 1989
1989, Plexiglas mit Vergrößerungslupe; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

168 Mikroprozessor Z80 (3 Stück) von Zilog, USA
1975; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

169 Schnittmodell eines Prozessor-Chips
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

170 Platine mit Mikroprozessor von Texas Instruments, USA
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

171 Platine eines IBM-Rechners mit Chip, IBM14 und IBM52
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing #Hardware

172 Single-Chip-Computer, „Supercomputer“ der Firma n-Cube, USA
ZKM | Karlsruhe

#GenealogieDesCodes

#Computing #Hardware

2. Werke zur Mikroelektronik

173 Carver Mead, Lynn Conway, *Introduction to VLSI Design*, Addison-Wesley Publishing Company, 1980
Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

174 E. Hörbst, C. Müller-Schloer, H. Schwärtzel, *Design of VLSI Circuits*, Springer-Verlag, Heidelberg
1986; Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

175 Franz Pichler, *Historische Meilensteine der Mikroelektronik*, Trauner Verlag, Linz
2012, Mit freundlicher Genehmigung von Franz Pichler

#GenealogieDesCodes

#Computing

Karl Sims

*1962 in Boston (US), lebt und arbeitet in den USA

176 *Evolved Virtual Creatures*
1994; Computeranimation, Video, 5 Min.

#GenetischerCode

#DNA #ComputerGeneriertesDesign

Adam Słowik, Christian Lölkes, Peter Weibel

*1980, in Skierniewice (PL), lebt und arbeitet in Berlin (DE)

*1990 in White Plains (US), lebt und studiert in Karlsruhe (DE)

*1944 in Odessa (UA), lebt und arbeitet in Karlsruhe (DE)

177 *Alphabet-Space*
Mixed-Media-Installation, 2017

#Codierung

#GenealogieDesCodes

#ComputerGeneriertesDesign

#Software #Hardware #Interface

Rasa Smite, Raitis Smits

*1969 in Riga (LV), lebt und arbeitet in Riga

*1966 in Riga (LV), lebt und arbeitet in Riga

178 *Biotricity. Fluctuations of Micro-Worlds*
 2014, Mikrobielle Batterie (2 MBZ), Echtzeitsonifikation und -visualisierung von Bioenergie, Video; Die Arbeit wurde mit Unterstützung der Stiftung Kulturhauptstadt Litauens und des Instituts für Festkörperphysik der Universität Litauen realisiert. Die Sonifikation wurde in Zusammenarbeit mit dem Künstler Voldemärs Johansons realisiert.

#Codierung
 #Bioengineering

Space Caviar

Simone C. Niquille, *1987 in Zug (CH), lebt und arbeitet in in Amsterdam (NL)
 Joseph Grima, *1977, lebt und arbeitet in Genua (IT)

179 *Fortress of Solitude*
 2014, HD-Video, Farbe, Ton, 20:25 Min.; Eine Auftragsarbeit der Biennale Interieur Foundation; Soundtrack: M.E.S.H.

#MaschinellesLernen
 #AlgorithmicGovernance
 #KünstlicheIntelligenz
 #Mustererkennung
 #AutonomeSysteme
 #InternetDerDinge
 #Automation #Industrie4.0

Barry Stone

*1971 in Lubbock (US), lebt und arbeitet in Austin (US)
DAILY, IN A NIMBLE SEA

180 *20150714-DSCF9064_3.tif Bailey Island, Maine (soft sun), 20150714-DSCF9064_3.txt Bailey Island, Maine (soft sun)*
 2016, 2 archivfeste Inkjet-Drucke auf ungestrichenem Polar-Matte-Papier, 101 × 150,5 cm

181 *20150714-DSCF9296_3.tif, Bailey Island, Maine (seascape), 20150714-DSCF9296_3.txt, Bailey Island, Maine (seascape)*
 2016, 2 archivfeste Inkjet-Drucke auf ungestrichenem Polar-Matte-Papier, 101 × 67 cm

182 *20150714-DSCF9297_4.tif, Bailey Island, Maine (seascape),*

20150714-DSCF9297_4.tif, Bailey Island, Maine (seascape)
 2016, 2 archivfeste Inkjet-Drucke auf ungestrichenem Polar-Matte-Papier, 101 × 67 cm; Mit freundlicher Genehmigung der Klaus von Nichtssagend Gallery, New York

#Codierung
 #Zahlensystem

Monica Studer, Christoph van den Berg

*1960 in Zürich (CH), lebt und arbeitet in Basel (CH)
 *1962 in Basel (CH), lebt und arbeitet in Basel

183 *Dark Matter - One Million Years Later*
 2016-2017, Computergenerierte Animation, Loop, 10 Min.

#Codierung
 #Algorithmus
 #ComputerGeneriertesDesign

184 *Passage Park #7: relocate*
 2017, Interaktive Echtzeit-Animation, Projektion, Interface

#Codierung
 #Algorithmus
 #ComputersimulierteUmgebungen
 #Interface

T

The Critical Engineering Working Group

Julian Oliver, *1974 in Neuseeland, lebt und arbeitet in Berlin (DE)
 Gordan Savičić, *1980 in Wien (AT), lebt und arbeitet in Lausanne (CH)
 Daniil Vasiliev, *1978 in Russland, lebt und arbeitet in Berlin (DE)

185 *The Critical Engineering Manifesto*
 2011, Druck, 84,1 × 118,9 cm

#AlgorithmicGovernance
 #Computing

JoI Thomson

*1981 in Toronto (CA), lebt und arbeitet in Berlin (DE) und London (GB)

186 *Deep Time Machine Learning*
 2017, 3-Kanal-Projektion, Farbe, Ton, 12 Min.; Die Videoarbeit

entstand im Rahmen des „Wimmelforschungs-Stipendiums“ im Kontext von „Plattform 12“, einem Gemeinschaftsprojekt von Akademie Schloss Solitude, Robert Bosch GmbH und Wimmelforschung. Mit zusätzlicher Unterstützung von: Staatliches Museum für Naturkunde Stuttgart; Freundeskreis Philipp Matthäus Hahn e.V., Kornwestheim

#MaschinellesLernen

#GenealogieDesCodes

#KünstlicheIntelligenz

#Mustererkennung

#AutonomeSysteme

Suzanne Treister

*1958 in London (GB), lebt und arbeitet in London

187 Hexen 2.0 / Macy Conferences Attendees

2009-2011, Giclée-Drucke;

Mit freundlicher Genehmigung

von Anny Juda Fine Art, London

#GenealogieDesCodes #Kybernetik

U

UBERMORGEN.COM

gegründet 1995, tätig in Wien (AT) und St. Moritz (CH)

188 Chinese Coin (Red Blood)

2015; Mixed-Media-Installation, Full HD-Video mit Dolby Surround 5.1, rote Bank, Video und Ton: Mike Huntermann

#AlgorithmischeÖkonomie

#AlgorithmicGovernance

#Bitcoin #Kryptowährungen

V

Ruben van de Ven

*1989 in Lelystad (NL), lebt und arbeitet in Rotterdam (NL)

189 Emotion Hero

2016, Android-App, Server-Software, browser-basierte Projektion

#MaschinellesLernen

#Mustererkennung

Koen Vanmechelen

*1965 in Sint-Truiden (BE), lebt und arbeitet in Genk (BE)

190 Book of Genome - PCC

2016, 3 ledergebundene Bücher mit vergleichenden DNA-Sequenzanalysen, jeweils 8 x 35 x 31 cm

191 DECODE - PCC

2016, Video, Farbe, Ton, 60 Min.

#GenetischerCode

#DNA #Genotyp

Ivar Veermäe

*1982 in Tallinn (EE), lebt und arbeitet in Berlin (DE)

192 Center of Doubt

2012-2015, Künstlerisches Forschungsprojekt, 3-Kanal-Videoinstallation bestehend aus: *Crystal Computing* (Google Inc., St. Ghislain), HD-Video, 9:20 Min.

Patent Application Data,

HD-Video, 8:06 Min.

The Formation of Clouds,

HD-Video, 7:24 Min.

#AlgorithmicGovernance

#BigData #Hardware

::vtol::

*1986 in Moskau (RU), lebt und arbeitet in Moskau

193 IVY

2017, Step-Sequencer

#Codierung

#Klangprogrammierung

W

Clemens Wallrath, Felix Held

*1992 in Karlsruhe (DE), lebt und studiert in Karlsruhe

*1990 in Rinteln (DE), lebt und arbeitet in Karlsruhe (DE)

194 keine zahl ist illegal

2017, Installation, 40 x 40 RGB-LED-Matrix-Panel

#Codierung

#Computing #Interface

#Entschlüsseln

Clemens von Wedemeyer

*1974 in Göttingen (DE), lebt und arbeitet in Leipzig (DE)

195 *ESIOD 2015*

2016, HD-Video, Farbe, Ton,
39 Min.; Mit freundlicher
Genehmigung von KOW, Berlin

#AlgorithmicGovernance

#AlgorithmischeÖkonomie

#BigData #QuantifiziertesSelbst

Peter Weibel (Idee), **Ludger
Brümmer** (Computeranimationen),
Götz Dipper (Interaktives
Environment)

*1944 in Odessa (UA), lebt und
arbeitet in Karlsruhe (DE)

*1958 in Werne (DE), lebt und
arbeitet in Karlsruhe (DE)

*1966 in Stuttgart (DE), lebt
und arbeitet in Karlsruhe (DE)

Produktion: ZKM | Karlsruhe

196 *Monochord*

2012, Interaktive audiovisuelle
Installation für Computer und
Bildschirm; Sponsor: Genesis,
physical Modeling Environment:
ACROE, Grenoble

#Codierung

#Klangprogrammierung #Software

#Interface

Alex Wenger, Max-Gerd Retzlaff

*1975 im Kanton Zug (CH), lebt und
arbeitet in Ettlingen (DE)

*1981 in Warendorf (DE), lebt und
arbeitet in Karlsruhe (DE)

197 *Daten/Spuren*

2015, Multimedia-Installation

#AlgorithmicGovernance

#BigData #QuantifiziertesSelbst

#Entschlüsseln #Programmieren

Where Dogs Run

gegründet 2000 in Jekaterinburg (RU)
Alexey Korzukhin, *1973 in Sverd-
lowsk (RU); Olga Inozemtseva, *1977
in Jalutorowsk (RU); Natalia
Grekhova, *1976 in Kamensk-Uralski
(RU); Vladislav Bulatov, *1974
in Sverdlowk (RU);

Die KünstlerInnen leben und
arbeiten in Jekaterinburg (RU)

198 *Symbolism in Circuit Diagrams*

2006 fortlaufend, Mixed-Media-
Installation

#Codierung

#Decodierung #Software

#Hardware #Interface

#ComputersimulierteUmgebungen

**Wibu-Systems AG und FZI
Forschungszentrum Informatik,
Karlsruher Institut für
Technologie (KIT)**

199 *Blurry Box®*

2014, USB-Dongle, Software
Vom 20.10.2017 bis 25.03.2018
in der Ausstellung

#Codierung

#Entschlüsseln #Software

Stephen Willats

*1943 in London (GB), lebt und
arbeitet in London

200 *A State of Agreement*

1975, 4 Tafeln, Gouache, foto-
grafische Drucke, Letraset auf
Karton, Tinte; Mit freundlicher
Genehmigung der Victoria Miro
Gallery, London

201 *Meta Filter*

1973-1975, HD-Film, Farbe, Ton,
6 Min., Digitaldrucke

202 *Six Levels of Interpersonal
Organisation*

1974, Fotografische Drucke,
Gouache, Tinte, Letraset auf
Karton

#GenealogieDesCodes

#AlgorithmicGovernance

#Kybernetik #Computing

**Manfred Wolff-Plottegg,
Wolfgang Maass**

*1946 in Schöder (AT), lebt und
arbeitet in Graz und Wien (AT)

*1949 in Frankfurt a. M. (DE),
lebt und arbeitet in Graz (AT)

203 *Neuronaler Architektur Generator*

1999, Computerinstallation,
2 PC (CPU 686), 2 Projektionen

#MaschinellesLernen

#Codierung

#ComputersimulierteUmgebungen

World-Information Institute

gegründet 1999, tätig in Wien (AT)

204 *Painted by Numbers*

2016, 6-Kanal-Videoinstallation

#AlgorithmicGovernance

#AlgorithmischeÖkonomie

Cerith Wyn Evans

*1958 in Llanelli (GB), lebt und arbeitet in London (GB)

205 *“Astrophotography - Stages of Photographic Development” by Siegfried Marx (1987)*

2007, Kronleuchter (Luce Italia), Glühlampen, Flachbildschirm, Computer mit Morsezeichen-Einheit; Thyssen-Bornemisza Art Contemporary Collection, Wien

#GenealogieDesCodes

#Codierung

#Decodierung #Morsecode

Z

Julia Zamboni

*1985 in Brasilia (BR), lebt und arbeitet in Montreal (CA); Die Künstlerin ist Teil des internationalen Netzwerkes Hexagram. Das in Montreal (CA) gegründete Kollektiv widmet sich der Forschung in den Bereichen Medienkunst, Design, Technologie und digitaler Kultur und besteht aus über 80 Mitgliedern.

206 *Robot Ludens*

2017, Installation, 2 Touchscreens, spinnenähnlicher Roboter; Sponsoren: Hexagram und TAG (Technoculture, Art, and Games)

#Arbeit&Produktion

#Industrie4.0 #Roboter

#Automation #Programmieren

ZKM | Karlsruhe

207 *Genealogie des digitalen Codes*

2017, Installation

Linear Navigator (1999): Jeffrey Shaw; Idee: Peter Weibel; Konzeption, Realisierung: ZKM | Institut für Bildmedien; Projektleitung: Bernd Lintermann; Redaktion: Livia Nolasco-Rózsás, Magdalena Stöger, Olga Timurgalieva; Software: Bernd Lintermann, Nikolaus Völzow; Videopostproduktion und Grafik: Moritz Büchner, Frenz Jordt, Christina Zartmann; Konstruktion: Nelissen Dekorbouw

Impressum

20. Oktober 2017 – 5. August 2018
Open Codes. Leben in digitalen Welten

Ausstellung

Konzept: Peter Weibel

Kuratoren: Peter Weibel mit
Livia Nolasco-Rózsás,
Yasemin Keskinetepe,
Blanca Giménez

Wissenschaftliche Beratung:
Christian Lölkes

Externe Berater: Natalia Fuchs,
Franz Pichler

Projektassistentz: Magdalena Stöger,
Olga Timurgalieva, Amit Shemma,
Hannah-Maria Winters

**Kuratorische Assistenz für
die Hexagram-Projekte:**
Garrett Lockhard

Leitung kuratorische Abteilung:
Philipp Ziegler

**Leitung Museums- und Ausstellungs-
technik:** Martin Mangold

Szenografie und Raumkonzept:
Peter Weibel, Vitra GmbH,
feco-feederle GmbH

Technische Projektleitung:
Thomas Schwab mit
Andrea Hartinger

Aufbauteam: Andrea Hartinger,
Volker Becker, Claudius Böhm,
Mirco Fraß, Rainer Gabler,
Gregor Gaissmaier, Ronny Haas,
Dirk Heesakker, Daniel Heiss,
Christof Hierholzer, Werner
Hutzenlaub, Gisbert Laaber,
Christian Nainggolan, Marius
Nestler, Marco Preitschopf,
Martin Schläfke, Marc Schütze

Externe Firmen: Artinate, Essential Art
Solutions, Artbeats

Konzept Ausstellungsgrafik:
Peter Weibel, Christian Lölkes

Restauratorische Betreuung:
Nahid Matin Pour, Sophie Bunz,
Ursula Ganß

Logistik, Registrarin: Natascha Daher

Direktion: Anett Holzheid,
Tobias Klingenmayer, Adrian Koop

ZKM_Hertz-Lab: Institut für Bildmedien:
Bernd Lintermann, Jan Gerigk,
Manfred Hauffen, Volker Nowicki
Institut für Musik und Akustik:
Ludger Brümmer, Yannick Hofmann,
Elisabeth Pich, Ben Miller, Götz
Dipper, Dan Wilcox,

Bernhard Sturm, Dorte Becker,
Chikashi Miyama, Sami Chibane

Presse- und Öffentlichkeitsarbeit:
Dominika Szope, Regina Hock,
Alexa Knapp, Stefanie Strigl,
Sophia Wulle, Marie Schmidt

Videostudio: Christina Zartmann,
Moritz Büchner, Frenz Jordt,
Martina Rotzal

Museumskommunikation:
Janine Burger, Banu Beyer,
Sabine Faller, Regine Frisch,
Edgar Guttmann, Barbara Zoé
Kioldbassa, Fanny Kranz

**Veranstaltungsmanagement und
Veranstaltungstechnik:**
Viola Gaiser, Wolfgang Knapp,
Desiree Weiler, Hartmut Bruckner,
Manuel Weber, Manuel Becker,
Hans Gass, Andre Gemmrich,
Niklas Wallbaum

Sekretariate: Ingrid Truxa,
Anna Maganuco, Caroline Mössner,
Sabine Krause, Alexandra Kempf,
Elke Cordell, Silke Sutter,
Dominique Theise

Bibliothek: Petra Zimmermann,
Christiane Minter, Regina Strasser-
Gnädig, Alena Dauth

**Wissen (Sammlung, Archive &
Forschung):** Margit Rosen,
Andreas Brehmer, Claudia Gehrig,
Hartmut Jörg, Felix Mittelberger,
Dorcas Müller, Stephanie Tiede

EDV: Uwe Faber, Elena Lorenz,
Joachim Schütze,
Volker Sommerfeld, Christian Lölkes

Museumsshop und Infotheke:
Petra Koger, Daniela Doermann,
Tatjana Draskovic, Laurine Haller,
Ines Karabuz, Rana Karan,
Susen Schorpp, Jutta Schuhmann,
Marina Siggelkow

Das **ZKM | Karlsruhe** dankt den
KünstlerInnen, LeihgeberInnen
und KooperationspartnerInnen
der Ausstellung für ihre Unter-
stützung. Besonderer Dank an:
Corona Feederle, Marc Frohn,
Eva Hamsch, Matthias Hoffmann,
Jochen Specht, Gerd Wetzel

Broschüre

Herausgeber: Peter Weibel

Redaktion: ZKM | Publikationen
(Jens Lutz, Ulrike Havemann,
Anna Straetmans, Miriam Stürner,
Claudia Voigtländer)

AutorInnen: Blanca Giménez,
Yasemin Keskintepe, Livia Nolasco-
Rózsás, Franz Pichler, Peter Weibel

Lektorat: ZKM | Publikationen (D),
Gloria Custance (E)

Übersetzungen: Petra Kaiser (D);
Gloria Custance, Isaac Custance,
Jane Yager (E)

Covergestaltung: Peter Weibel,
Christian Lölkes

Schriften: Univers, Inconsolata

Papier: Amber Graphic, Cover: 150 g/m²,
Inhalt: 80 g/m²

Druck: Stober GmbH, Eggenstein

Dank an: Tanja Schüz, Anett Holzheid,
Gloria Custance, Martina Hofmann

© 2017 ZKM | Zentrum für Kunst und
Medien Karlsruhe

© der Texte: bei den AutorInnen
Soweit nicht anders vermerkt, befinden
sich die Werke im Besitz der
KünstlerInnen



ZKM | Zentrum für Kunst und Medien
Karlsruhe

Lorenzstraße 19, 76135 Karlsruhe
Germany
info@zkm.de
www.zkm.de

Vorstand ZKM: Peter Weibel

Geschäftsführender Vorstand ZKM:
Christiane Riedel

Verwaltungsleitung: Boris Kirchner

STIFTER DES ZKM



PARTNER DES ZKM



IN KOOPERATION MIT



MIT FREUNDLICHER UNTERSTÜTZUNG VON: WERNER STOBER STIFTUNG, FAM. WÜRTH, FAM. LEIBINGER

GEFÖRDERT VON



Light | Art | Space



KULINARISCHE PARTNER: BÜHLER CATERING ETTLINGEN, RITTER SPORT, LASERZENTRUM LINDAU, FOODLOOSE GMBH & CO. KG, SÜDKOLA LIMONADENWERK, BRATZLER & CO. GMBH, LIFEBRANDS NATURAL FOOD GMBH, BARNHOUSE NATURPRODUKTE GMBH, OBST VOM BODENSEE VERTRIEBSGESELLSCHAFT MBH, SEEZÜNGLE

