

# Quellcodekritik aus der Ferne Distant Viewing und kritische Analyse von Quellcode

## Demleitner, Adrian

adrian@thgie.ch  
Hochschule der Künste Bern, Schweiz  
ORCID: 0000-0001-9918-7300

## Höltgen, Stefan

stefan.hoeltgen@uni-bonn.de  
Universität Bonn, Deutschland  
ORCID: 0000-0001-8669-8507

## Piontkowitz, Vera

piontkowitz@informatik.uni-leipzig.de  
Universität Leipzig, Deutschland  
ORCID: 0000-0003-3605-3609

## Gammenthaler, Daniel

daniel.gammenthaler@unibe.ch  
Universität Bern, Schweiz

## Burghardt, Manuel

burghardt@informatik.uni-leipzig.de  
Universität Leipzig, Deutschland  
ORCID: 0000-0003-1354-9089

## Abstrakt

Die Critical Code Analysis (Marino 2020; Bajohr und Krajewski 2024) verortet Quellcode nicht nur als Funktionsanleitung für Maschinen, sondern auch als kulturelles Artefakt, als Text, der von und für Menschen geschrieben wurde. Es stellt sich die Frage, wie wir in den digitalen Geisteswissenschaften den strukturellen Eigenheiten von Quellcode gerecht werden können? Dieser Workshop geht auf verschiedene Ansätze des distant reading von Quellcode ein, zur Untersuchung auf Bedeutung und der Erarbeitung von Wissen.

## Kontext

Quellcode ist Text, der Anweisungen für den Rechner enthält und nach den Regeln einer Programmiersprache strukturiert wird. Diese Regeln stellen sicher, dass Quellcode

durch verschiedene Verfahren in Maschinencode übersetzt werden kann (Berry 2015). Sie führen aber auch dazu, dass sich Quellcode fundamental in seiner Struktur von anderen Textsorten unterscheiden (Willumsen 2016). So entzieht sich Quellcode einerseits den etablierten Ansätzen zur hermeneutischen Untersuchung durch distant reading in den digitalen Geisteswissenschaften. Andererseits konzentrieren sich Werkzeuge zur Analyse von Quellcode in den Informationswissenschaften auf Code-Qualität, und nicht auf die Erarbeitung von Erkenntnissen. Da sich Quellcode in Umfang und Modularität schnell der kognitiven Leistungsfähigkeit entzieht, ist distant reading ein geeigneter Ansatz, die Forschung an dieser Textsorte zu unterstützen. Dazu braucht es eine Auseinandersetzung in der Anwendung von Methoden und Werkzeugen.

Es reicht nicht aus, nur einzelne Code-Bestandteile im Detail zu betrachten. Oftmals werden lediglich die zugänglicheren Aspekte analysiert, etwa Variabel-Namen, einzelne Subroutinen oder einzelne Code-Kommentare. Um den Quellcode einer Software in seiner Gänze erfassen zu können, müssen wir uns auch strukturellen Aspekten widmen. Dazu gehören Programmfluss, Organisation des Codes, Abhängigkeiten von Programmiersprache und Computer-System und weiteres. Da sich solche Einsichten erst nach längerer und intensiver Auseinandersetzung einstellen oder durch die Menge an vorhandenem Code gar verunmöglicht werden, bietet distant reading methodische Unterstützung an. Distant reading bedeutet hier die computergestützte Methodik zur Erkennung von Strukturen, Mustern, Programmflüssen und grösseren Zusammenhängen in der Form von Visualisierungen, Large Language Models, Netzwerkanalysen, Korpusanalysen und Ähnliches.

## Quellenmaterial: Videospiele aus den 1980er-Jahren

Die frühe Geschichte der Videospieldentwicklung ist durch inhärente Verbindung mit der Ankunft der digitalen Technologie in Europa ein wichtiger Bestandteil der Technikgeschichte (Haddon 1988; Williams 1976; Švelch 2018; Alberts und Oldenziel 2014). Die Übernahme neuer Technologien ist immer mit experimentellen und spielerischen Begegnungen verwoben. Diese lassen sich auf ein Spektrum einordnen, das von Basteln über das Spielerische bis hin zum Erschaffen von Spielen reicht (Swalwell 2021).

Eine Möglichkeit, sich der frühen Geschichte der Videospieldentwicklung zu widmen, bieten die Critical Code Studies. Die kritische Codeanalyse behandelt Code nicht nur als Funktionsanweisung an eine Maschine, sondern auch als Text, der hermeneutisch angegangen und semiotisch gelesen werden kann (Marino 2020). Dabei wird der Code nicht durch gängige Analysemethoden der Informatik auf seine Qualität hin kontrolliert, sondern vor allem kontextualisiert und in Beziehung gesetzt. Aus Quellcode wird so eine Linse, durch welche sich Arbeitsprozesse, soziale Begebenheiten oder materielle Bedingungen betrachten lassen.

Wir bringen für den Workshop geeigneten Quellcode mit. Dabei handelt es sich um einfachere Spiele aus den 1980ern, welche in verschiedenen BASIC-Dialekten geschrieben wurden. Diesem Quellcode werden wir durch die weiter unten kontextualisierten Verfahren begegnen.

## Ziel

Als Tutorial gedacht, geht der Workshop drei Zielen nach.

1. Einführung in die kritische Codeanalyse
2. Austausch und gemeinsame Erkundung des vorgestellten Ansatzes
3. Praktische Auseinandersetzung mit distant reading von Quellcode

Der Fokus des Workshops liegt auf der kritischen Codeanalyse. Nach einer grundlegenden Einführung des Ansatzes sowie theoretischen Inputs sehen wir uns verschiedenen Quellcode an, wie dieser strukturiert ist, welche Teile davon interessant sein können und wie wir diesen begegnen könnten. In der Besprechung von konkreten Fällen versuchen wir uns in einer gemeinsamen Annäherung an Quellcode als geisteswissenschaftliches Forschungsmaterial.

Diskussionen und ein Intermezzo sollen den Ansatz zur kritischen Hinterfragung und Erweiterung öffnen. Distant reading von Quellcode ist kein etablierter Ansatz und kann von einer gemeinsamen Besprechung profitieren. Dabei dürfen die Teilnehmenden auch ihre eigenen Erfahrungen in der hermeneutischen Analyse von Quellcode einbringen.

Im letzten und optionalen Block können die besprochenen und vorgestellten Prozesse und Werkzeuge erprobt werden. Dazu gehören Aufbereitungen von Code durch Python-Scripts, Netzwerkanalysen mit Gephi, Veranschaulichungen durch Programmfluss- und Parametrisierungs-Visualisierungen, die Auseinandersetzung mit Fragestellungen an den Quellcode durch Large Language Models und weitere. Dieser Block ist als praxisorientierte Ergänzung zu den vorgängigen Präsentationen und Besprechungen zu verstehen.

## Publikum und Vorwissen

Der Workshop richtet sich generell an Forschende, welche sich in ihrer Arbeit mit Quellcode auseinandersetzen oder sich dafür interessieren. Verwandte Textarten an der Schnittstelle von Mensch und Maschine, wie Hyper- oder Cybertext, sowie Forschende in den Video Game Studies sind ebenfalls herzlich eingeladen. Vorwissen sowie Einreichung eines eigenen Beitrags sind nicht notwendig.

Teilnehmende, welche am letzten und optionalen Block teilnehmen möchten, müssen ein eigenes Arbeitsgerät mitbringen. Eine gewisse Frustrationstoleranz sowie Grundkenntnisse in Python sind ebenfalls von Vorteil. Es sind Gruppenarbeiten vorgesehen, sodass sich Teilnehmende mit verschiedenen Wissensständen gegenseitig unterstützen und ergänzen können. Wir versenden vorab einen Tech-Reader, welcher hilft, die für den Workshop notwendige

Software zu installieren. Die Workshopleitung steht eine halbe Stunde vor Workshopbeginn für technischen Support zur Verfügung. Weiter bringt die Workshopleitung auch geeigneten Quellcode zur Arbeit mit. Teilnehmende dürfen jedoch gerne eigene, zu untersuchende Fälle einreichen oder mitbringen.

## Format und Ablauf

Der Workshop ist als Tutorial ausgelegt und für die Dauer eines ganzen Tages ausgelegt.

- Grundlagen: Einführung in die kritische Codeanalyse (1h 30min)
  - 10" Einführung und Vorstellung
  - 25" Ansätze zu distant reading und der kritischen Codeanalyse (PhD-Kandidat Adrian Demleitner)
  - 25" Computational Game Studies: Annäherungsperspektiven (PhD-Kandidatin Vera Piontkowitz, Dr. Manuel Burghardt)
  - 25" Quellcode: Relevante Aspekte und Parametrisierung stilistischer Eigenschaften (Dr. Dr. Stefan Höltingen)
- 15" Pause
- Praxis 1: Geisteswissenschaftliche Annäherung an Quellcode (1h 30min)
  - Präsentation und Besprechung von Fallbeispielen
  - 20-30" pro Fall (3-4): Kurzvorstellung Fall mit anschließender Besprechung in Gruppen und Plenum zur Erarbeitung geisteswissenschaftlicher Fragestellungen an Quellcode
- Mittags-Pause (1h)
- Kritische Auseinandersetzung (1h 30min)
  - 25" Im Gespräch mit Quellcode: Input und Demo zu Large Language Models und Code Analyse (PhD-Kandidat Adrian Demleitner)
  - 25" Code Analyse und Hermeneutik? Input zur kritischen Betrachtung der Critical Code Analysis (PhD-Kandidat Daniel Gammenthaler)
  - 25" Moderierte Diskussion und Erfahrungsaustausch
- 15" Pause
- Praxis 2: Optionale technische Vertiefung (2h)
  - Vorgängig besprochene Werkzeuge und Verfahren können in diesem Block vertieft und getestet werden: Dazu gehören Python zur Aufbereitung und ersten Analyse von Quellcode, Netzwerkanalyse mit Gephi, Visualisierungen von Programmfluss, Quantifizierung formaler Code-Aspekte und weitere
  - 30" Einführung der Werkzeuge und Prozesse
  - 1h 30" Begleitetes Arbeiten in Gruppen mit Support und Inputs der Workshopleitung

Da es stellenweise direkten Support der Teilnehmenden brauchen wird, können wir realistisch gesehen nicht mehr als 20 Personen aktiv betreuen.

## Verfahrensweisen

### Visualisierung

Zwei strukturelle Eigenheiten von Quellcode sind die Modularisierung und der Informationsfluss. Ersteres drückt sich dadurch aus, dass sich wiederholende Code-Bausteine in eigenen Dateien oder Funktionen abstrahiert werden. Beim Informationsfluss werden durch Schleifen oder Anweisungen zum Sprung an eine andere Stelle die Linearität des Textes, und somit des Leseflusses, aufgebrochen. Um diese Strukturen besser zu erschliessen, eignen sich Visualisierungen. Hier eignen sich Flussdiagramme im Kleinen oder Graphen im Grossen, um sich bessere Übersichten und Zugänge zu verschaffen oder Erkenntnisse zu erlangen.

### Korpusanalyse

Nebst den strukturellen Vorgaben durch die Programmiersprache werden zur kognitiven Unterstützung bei der Programmierung auch frei wählbare Elemente erlaubt, unter anderem die Benennung von Variablen oder Kommentaren. Diese Elemente eignen sich natürlich für Ansätze der Korpusanalyse. Gerade Kommentare können sehr wertvoll sein. Vor allem bei Quellcode mit mehreren Autor:innen können diese frei beschreibbaren Elemente Hinweise auf Arbeitsorganisation, soziale Prozesse, Schreibstile und weiteres geben.

Wie oben ausgeführt entzieht sich jedoch ein grosser Anteil von Quellcode angestammten computergestützten Korpusanalysen. Diese müssen in der Folge überdacht werden. Ein erster Ansatz ist der Fokus auf formale Aspekte. Programmier-Stil, Bandbreite an eingesetzten Befehlen oder das Verhältnis von Befehlen zur Programmlogik können Hinweise auf Arbeits- und Organisationsprozesse liefern. Einsichten zu Programmier-Praxen können via der Frage nach welcher und wie viel Code für welche grundlegenden Funktionen aufgewendet wurde, etwa der Darstellung von Grafiken oder der Verwaltung des Spielstandes.

### Kontextualisierung

Quellcode ist in der Regel ein Mittel zum Zweck. Das Endprodukt sind ausführbare Videospiele und andere Software. Über seine textliche Essenz hinaus ist Quellcode weiter in vielfältigen Beziehungen verwoben. Diese Verbindungen umfassen Interpreter, Compiler, Laufzeiten, integrierte Entwicklungsumgebungen (IDEs) und Entwicklungswerkzeuge, Programmiersprachen, daraus resultierende ausgeführte Ausgaben, Frameworks, geliehene Code-Fragmente und schlussendlich, wie erwähnt, den Autor:innen. Für eine hermeneutische Betrachtung von Quellcode ist es unabdingbar, diesen in Kontext mit seiner Produktion sowie seinem Produkt zu setzen. Hier eignen sich

Ansätze in der Modellierung und dem Mapping von Metadaten.

### Large Language Models (LLM)

LLMs können helfen, um einen ersten Zugang zu uns unbekanntem oder unlesbarem Quellcode zu erlangen, um mit diesem ins Gespräch zu kommen. Solche Fälle können eintreten, wenn unter anderem ein kompiliertes Programm zurück in Quellcode übersetzt wird. Dabei gehen die ursprünglichen Strukturen sowie Kommentare der Programmierer:innen verloren. Oder aber die Entwickler:innen bedienen sich einer Programmiersprache, welcher wir nicht mächtig sind. LLMs können hier unterstützend wirken und mit ersten Einsichten kontextualisieren. So können in diesem Verfahren grundlegende Fragen zur Programmiersprache des Quellcodes, dessen technologischer Kontext oder auch die wahrscheinlichen Funktionsweisen einzelner Subroutinen erörtert werden. Besonders interessant werden solche Gespräche, wenn der Quellcode durch nachträgliche Informationen der ursprünglichen Entwickler:innen angereichert wird, wie wir in einem Beispiel erörtern werden.

## Bibliographie

- Alberts, Gerard, and Ruth Oldenziel, eds.** 2014. *Hacking Europe: From Computer Cultures to Demoscenes*. History of Computing. London: Springer. <https://doi.org/10.1007/978-1-4471-5493-8>.
- Bajohr, Hannes, and Markus Krajewski, eds.** 2024. *Quellcodekritik: zur Philologie von Algorithmen*. Erste Auflage. August Akademie. Berlin: August Verlag.
- Berry, David M.** 2015. *The Philosophy of Software: Code and Mediation in the Digital Age*. Basingstoke: Palgrave Macmillan.
- Haddon, Leslie.** 1988. "The Home Computer: The Making of a Consumer Electronic." *Science as Culture* 1 (2): 7–51. <https://doi.org/10.1080/09505438809526198>.
- Marino, Mark C.** 2020. *Critical Code Studies*. Software Studies. Cambridge, Massachusetts: The MIT Press.
- Švelch, Jaroslav.** 2018. *Gaming the Iron Curtain – How Teenagers and Amateurs in Communist Czechoslovakia Claimed the Medium of Computer Games*. Game Histories. Cambridge, Massachusetts: The MIT Press. <https://mitpress.mit.edu/books/gaming-iron-curtain>.
- Swalwell, Melanie.** 2021. *Homebrew Gaming and the Beginnings of Vernacular Digitality*. Cambridge: MIT. <https://mitpress.mit.edu/9780262044776/homebrew-gaming-and-the-beginnings-of-vernacular-digitality/>.
- Williams, Richard.** 1976. "Early Computers in Europe." In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, 21–29. AFIPS '76. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1499799.1499804>.
- Willumsen, Ea Christina.** 2016. "Source Code and Formal Analysis: A Hermeneutic Reading of Passage." In: