

## **Spiel mir das Spiel vom Code – Über die Möglichkeiten der Nutzung digitaler Spiele in informatischen Bildungskontexten**

Lena-Marie Maisenhölder; Patrick Maisenhölder

### **Abstract**

In diesem Text wird anhand des *Spieles Debugger 3.16: Hack'n'Run* (2017) beispielhaft dargestellt, wie digitale Spiele in informatischen Bildungskontexten genutzt werden können, um problemorientierte Lehr-Lernsituationen zu gestalten. Es wird gezeigt, welche Vorteile dies für die informatische Bildung insbesondere von denjenigen Schüler\*innen mit sich bringt, deren Medienhabitus als unterhaltungsorientiert beschrieben werden kann (vgl. Kutscher 2009).

### **Digitale Spiele und Probleme**

Schaut man sich die Struktur digitaler Spiele als „rechnerbasierte, interaktive Zeichensysteme“ (Ostritsch 2018, 79), die unterhalten sollen (vgl. Tavinor 2009, 46), an, so zeigt sich, dass sie einen Anfangszustand und einen Endzustand aufweisen, wobei Spieler\*inneneingaben notwendig sind, um vom Anfangs- in den Endzustand zu gelangen. Dabei werden auf dem Weg von ersterem in letzteren – je nach Spiel – verschiedene Anforderungen an die Spieler\*innen gestellt. Manche Spiele erfordern Schnelligkeit, andere Genauigkeit, wieder andere Wissen und/oder die Fähigkeit, dieses situationsadäquat anwenden und kombinieren zu können und vieles weiteres. Oftmals sind auch mehrere dieser Aspekte nötig, um das Spiel zu meistern.

Neben anderen wichtigen Aspekten, machen diese Herausforderungen an die Kompetenzbestände der Spieler\*innen das Unterhaltungspotential des Spiels aus.<sup>1</sup> Denn sie erzeugen „*Spannung und Lösung*“ (Klimmt 2006, 81), da sie die

---

<sup>1</sup> Andere wichtige Aspekte, die digitale Spiele unterhaltsam machen, sind beispielsweise die Narration der Spiele (vgl. Neitzel 2018) oder andere Aspekte, wie etwa das freie und ziellose Umherbewegen in der Spielewelt, wie es in der Grand Theft Auto Spielereihe beim *Cruisen* zu beobachten ist. Dort ist es nicht unüblich, dass Spieler\*innen mit virtuellen Fahrzeugen Vergnügungsfahrten durch die Spielwelt unternehmen, ohne dass

Spieler\*innen auffordern, „Tätigkeiten mit Schnelligkeits-Präzisions-Anforderung oder [...] Komplexitäts-Bewältigungs-Anforderungen“ (ebd.) zu bewältigen. Und weil bei diesen Herausforderungen die Möglichkeit des Scheiterns gegeben ist, den Spieler\*innen aber nicht egal ist, ob sie scheitern oder nicht, entsteht eine Spannung. „Die Auflösung des Spannungszustandes kann [...] am Ende einer Computerspiel-Episode im Fall einer erfolgreichen Aufgabenbewältigung in extrem positive Erlebensformen münden oder aber im Fall eines Misserfolgs noch stärkere negative Erlebnisweisen als die vorangegangene Spannung selbst nach sich ziehen“ (ebd., 87).

Manche dieser Herausforderungen können nun den Kenntnis- und/oder Kompetenzstand der Spieler\*innen überschreiten, sodass die Überführung vom Anfangs- in den Endzustand durch die Anforderung als Überforderung nicht gelingt. Dann stehen die Spieler\*innen vor einem Problem. Denn darunter versteht man eine „Barriere, die die Überführung des Anfangszustandes in den Zielzustand im Augenblick verhindert“ (Edelmann, Wittmann 2012, 179).

Verhindert werden kann es dadurch, dass man die Mittel noch nicht kennt oder hat, um die Barriere zu überwinden oder die Mittel vorhanden sind, aber die Einsicht im Sinne der „plötzlich eintretenden Erkenntnis des Zusammenhangs zwischen den Elementen einer Problemsituation“ (Winkel et al 2006, 149) fehlt, man also nicht weiß, wie man das Vorhandene nutzen, verbinden etc. muss, um den Zielzustand zu erreichen. Natürlich können auch hier Mischformen auftreten. Solcherlei Barrieren treten in digitalen Spielen in verschiedener Form auf. Sie können Gegner\*innen sein, die schwierig zu besiegen sind, bestimmte Levelpassagen, in denen Genauigkeit und Schnelligkeit gefragt ist, weil man Gegenständen ausweichen und gleichzeitig auf andere Umgebungsaspekte, wie etwa Abgründe, die man herunterfallen kann, achten muss oder aber Rätsel, bei denen man verschiedene Elemente verbinden muss, man aber nicht weiß, wie. All diese Dinge stellen die Spieler\*innen vor Herausforderungen, die dann zu Problemen werden, wenn die Spieler\*innen noch nicht über die Mittel verfügen, diese zu bewältigen.

Die Probleme können verschiedenartig gelöst werden. Etwa dadurch, dass man den Zielzustand nicht länger als solchen ansieht und das Spiel abbricht. Sie können auch so gelöst werden, dass man durch das Senken des

---

dies etwas zum Weiterführen der Narration oder zum Bewältigen von Herausforderungen beiträgt oder selbst eine Herausforderung ist (vgl. Welsh 2006).

Schwierigkeitsgrades die Barriere verkleinert, sodass man mit den gegebenen Wissens- und Kompetenzbeständen die Barriere meistern kann. Oder indem man sie anderweitig überspringt, beispielsweise durch Eingabe von Cheats, die die Spielfigur unsterblich machen oder einen das Level im wahrsten Sinne des Wortes überspringen lassen.

Daneben gibt es aber eine weitere Möglichkeit, die von Spieler\*innen sogar aktiv angestrebt wird, indem sie sich immer schwierigere Herausforderungen suchen<sup>2</sup>: Den eigenen Kenntnis- und Kompetenzstand erweitern, damit die sich stellenden spielerischen Probleme gelöst werden können<sup>3</sup>. Und genau dieser Aspekt, dass digitale Spiele Spieler\*innen vor Probleme stellen, die sie lösen wollen – weil sie den Zielzustand erreichen, also, das Spiel durchspielen und meistern wollen – und sie damit motivieren, an ihrem Kompetenzstand zu arbeiten, kann in Bildungskontexten genutzt werden, um Kompetenzen zu vermitteln, die in den jeweiligen Fächern vermittelt werden sollen. Dies gilt ebenso für informatische Bildungskontexte.

### **Das Spiel *Debugger 3.16: Hack'n'Run***

Ein solches, für informatische Bildungskontexte nutzbares Spiel stellt *Debugger 3.16: Hack'n'Run* (2017) dar, das von Spiderwork Games entwickelt und herausgegeben wurde. Das Spiel ist, wie man im Titel schon erkennen kann, ein Jump'n'Run-Spiel. In diesen wird die Spielfigur

„von Spielabschnitt zu Spielabschnitt (Level) durch eine virtuelle Landschaft mit zahlreichen Hindernissen gesteuert; Hüpfen, Rennen, Gegnern ausweichen, sie behindern oder ausschalten verlangt Steuerungsgeschick und Reaktionsschnelligkeit“ (Pohlmann 2007, 11).

---

<sup>2</sup> So gibt es Spiele, die extra schwer zu meistern sind und gerade deshalb gespielt werden, weil sie ständig Herausforderungen stellen, die nur durch hohe Kompetenzen und weiteren Kompetenzzuwachs überstiegen werden können, wie etwa die Dark-Souls-Spiele (2011; 2014; 2016). Hierbei gibt es sogar Spieler\*innen, die den sowieso schon hohen Schwierigkeitsgrad dadurch erhöhen, dass sie sich selbst weitere Herausforderungen stellen, wie das Steuern der Spielfigur mit einem Schlagzeug statt dem Controller oder Tastatur und Maus (vgl. Haffelder, Maisenhölder 2019): Dies zeigt, dass sich Spieler\*innen durchaus (spielerische) Probleme suchen, um selbst Kompetenzzuwächse zu initiieren, die sie befähigen, diese Probleme zu lösen.

<sup>3</sup> Die Senkung des Schwierigkeitsgrads ist deshalb, wenn das Spiel es ermöglicht, in manchen Fällen auch nicht von Dauer, sondern dient der Kompetenzschulung bis zu dem Punkt, dass man die Herausforderungen, durch Übung auf niedrigerem Schwierigkeitsgrad, auf dem höheren Schwierigkeitsgrad meistern kann. Dies kann also Teil der Kompetenzzuwachsstrategie sein.

Entgegen einem „normalen“ Jump'n'Run wie *Super Mario World* (1992), ist das Besondere an *Debugger 3.16: Hack'n'Run*, dass man nicht nur Schnelligkeit und Genauigkeit benötigt, um die sich in der Spielwelt stellenden Herausforderungen zu bewältigen. Vielmehr muss selbige inklusive der eigenen Figur verändert werden. Etwa indem sich durch das Eingeben von Programmierbefehlen in die Konsole die Sprunghöhe der eigenen Spielfigur verändert wird, sodass man Hindernisse überwinden kann oder sich computergesteuerte Gegner\*innen verkleinern, sodass sie leichter zu besiegen sind. Entgegen „normaler“ Jump'n'Run-Spiele wird die Spielfigur also nicht nur durch Drücken festgelegter Tasten (z.B. Leertaste für Springen) bewegt. Vielmehr sind zusätzlich dazu, Programmierbefehle einzusetzen, um die Spielwelt und die Figuren in ihr zu verändern.

Bei der Programmiersprache handelt es sich um *Lite-C*, die eine vereinfachte Version der Programmiersprache *C* bzw. *C++* darstellt. Diese müssen die Spieler\*innen in Teilen lernen, um die Herausforderungen bewältigen zu können, die sich ihnen im Spiel stellen. Wie in anderen digitalen Spielen mag es hier so sein, dass am Anfang keine bis wenig entsprechende Kompetenzen vorhanden sind, sodass das Spiel die Spieler\*innen nicht nur vor Herausforderungen, sondern gar Probleme stellt, die den Spieler\*innen einen Kompetenzzuwachs abverlangen, wenn sie diese lösen wollen. In diesem Spiel heißt das: Programmierspracheneingaben lernen und richtig anwenden können, um vorliegende Probleme zu lösen. Etwa dann, wenn Spieler\*innen, wie beschrieben, Hindernisse überwinden müssen, diese aber zu groß sind, sodass die Spielfigur es nicht schafft, über sie zu springen.



Abbildung 1: Programm-Code-Eingabe zur Erhöhung der Sprunghöhe (eigener Screenshot)<sup>4</sup>

Ist dies der Fall, stehen die Spieler\*innen vor einem Problem, da die Barriere sie daran hindert, den Zielzustand zu erreichen und sie diese Barriere nicht überwinden können. Lernen sie dann die entsprechenden Programmierspracheneingaben, mit denen sie die Sprunghöhe ihrer Spielfigur erhöhen können, haben sie ein Mittel parat, die Barriere zu überwinden und das Problem verschwindet.

<sup>4</sup> Wir danken Michael Schaefer von Spidework Games für die freundliche Genehmigung, die Screenshots aus dem Spiel *Debugger 3.16: Hack'n'Run* verwenden zu können.



Abbildung 2: Programm-Code-Eingabe zur Erhöhung ist erfolgreich (eigener Screenshot).

Was das Spiel also von den Spieler\*innen verlangt, ist nicht nur Zuwachs an Kompetenzen, die in digitalen Spielen wichtig sind, wie schnelle Reaktionsfähigkeit und Genauigkeit beim Ausweichen und Springen, sondern Kompetenzen, die in informatischen Bildungskontexten angestrebt werden sollen.

### **Vorgehensweisen und Ziele informatischer Bildung**

Auch wenn der Informatikunterricht in den verschiedenen Bundesländern unterschiedlich ausgestaltet ist, finden sich Zielbeschreibungen informatischer Bildungsprozesse, die bundeslandübergreifend Gültigkeit beanspruchen können. Beispielsweise in den *Bildungsstandards Informatik der Gesellschaft für Informatik* (<https://informatikstandards.de/>), aber auch in fachdidaktischer Literatur, in denen die Ziele und Vorgehensweisen informatischer Bildung benannt, reflektiert und begründet werden. In Letzterer finden sich auch Überlegungen und Forschungen zu effektiven Methoden, also Vorgehensweisen, in informatischen Bildungskontexten, um die Zielkategorien zu erreichen.

Hinsichtlich der in informatischen Lehr-Lernkontexten genutzten Methoden führten etwa Zendler und Klaudt (2015) eine Studie durch, die sich mit der Betrachtung und Bewertung von Unterrichtsmethoden durch Informatiklehrer\*innen beschäftigte. Eine repräsentative Stichprobe von

Informatiklehrer\*innen bewerteten dabei, inwiefern sie die Auswahl an Unterrichtsmethoden für unterschiedliche Wissensprozesse geeignet hielten und bewerteten diese mit Punkten von 1 bis 5, wobei ein Punkt meint, dass die Methode *ungeeignet* ist und fünf Punkte bedeuten, dass sie für *sehr geeignet* gehalten wird. Das Ergebnis der Untersuchung ist gut in der folgenden *Heatmap* zu erkennen:

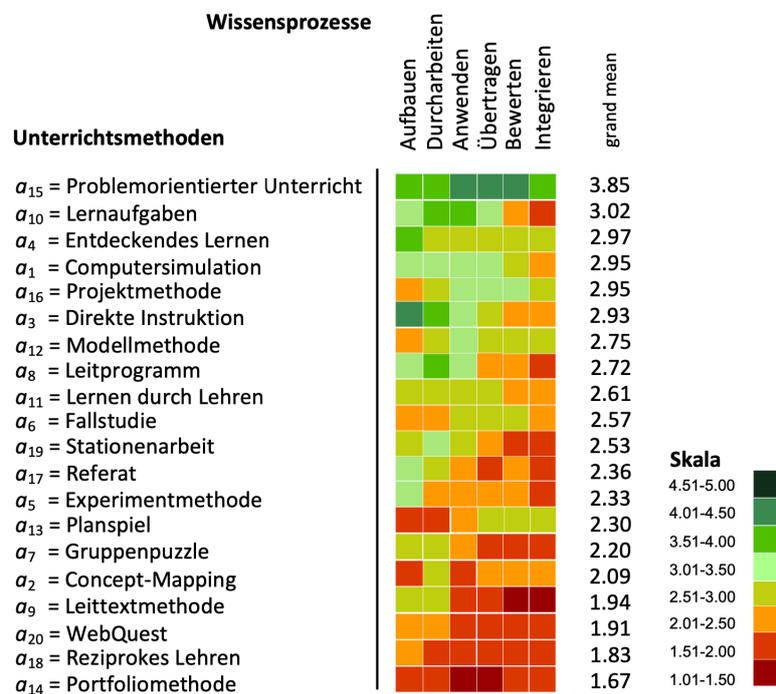


Abbildung 3: Visualisierte Mittelwerte der Einschätzung von Informatiklehrkräften von Unterrichtsmethoden für Wissensprozesse (*Heatmap*) (Übernommen aus: Zendler, Klaudt 2015, 45).

Angeführt wird die Liste der Unterrichtsmethoden vom *problemorientierten Unterricht*. Bei detaillierterer Betrachtung der Heatmap fällt auf, „dass der problemorientierte Unterricht in Bezug zu fast allen Wissensprozessen am besten abschneidet“ (Zendler, Klaudt 2015, 49). In dieser Studie wird *unter problemorientiertem Unterricht* „eine Unterrichtsmethode [verstanden], die Lernenden bei der Lösung eines exemplarischen Problemfalls Kompetenzen erwerben lassen, die in übertragbare Problemfelder transferiert werden können“ (Zendler, Klaudt 2015, 60). Digitale Spiele sind demnach mit der von Informatiklehrenden bestbewerteten Methode kombinierbar. Denn das Spiel stellt die Spieler\*innen vor Probleme, motiviert sie aber gleichzeitig dazu, diesen durch Kompetenzzuwachs zu begegnen, sodass problemorientierte informatische

Bildungsprozesse initiiert werden. Dies ist der Fall, da Probleme hierbei wirklich als solche wahrgenommen werden: Die Schüler\*innen als Spieler\*innen wollen das Spiel durchspielen, stehen dabei aber vor verschiedenen Barrieren, die damit zu tun haben, dass sie die Programmiersprache nicht können oder in dieser Situation nicht wissen, wie sie die Fähigkeiten ihrer Spielfigur oder die Umwelt durch die Nutzung der Programmiersprache modifizieren müssen, um die sich stellenden Herausforderungen zu meistern. Dies motiviert die Spieler\*innen dazu, sie lernen zu wollen bzw. ihre Fähigkeit lösungsorientiert anzuwenden, sodass das, was als Vorteil des problemorientierten Lernens angesehen wird, tatsächlich auftritt – die Förderung von „Lernmotivation, Autonomie, Lernkompetenz und Transferfähigkeit“ (Winkel et al 2006, 262) – und die entsprechenden zu lernenden Kompetenzen erworben werden.

Vergleicht man diese Erkenntnisse mit den sogenannten *GI-Standards* der Gesellschaft für Informatik e.V., so lässt sich auch zeigen, welche Kompetenzen dies sind, die durch digitale Spiele wie *Debugger 3.16: Hack'n'Run* (2017) geschult werden sollen. Diese Standards wurden, wie angedeutet, aufgrund der Differenz in der Gestaltung des Informatikunterrichts in den Bundesländern von der Gesellschaft für Informatik e.V. im Jahr 2008 als Empfehlung herausgegeben, um inhalts- und prozessbezogene Kompetenzen für alle Schulstufen zu formulieren. Für Sekundarstufe I und II lassen sich dabei folgende inhaltsbezogene Kompetenzen finden, die mit dem Spiel *Debugger 3.16: Hack'n'Run* (2017) geschult werden können:

Sekundarstufe I: „Schülerinnen und Schüler der Jahrgangsstufe 8 bis 10 ...

- ... verwenden Variablen und Wertzuweisungen
- ... modifizieren und ergänzen Quelltexte nach Vorgaben
- ... interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv“ (Gesellschaft für Informatik e.V. 2008, 16).

Sekundarstufe II: „Schülerinnen und Schüler...

- ... verwenden algorithmische Grundbausteine (Sequenz, Alternative, Wiederholung) und implementieren diese mithilfe einer Programmiersprache“ (Gesellschaft für Informatik e.V. 2016, S. 10).

Sekundarstufe I: „Schülerinnen und Schüler der Jahrgangsstufe 8 bis 10 ...
--

- |  |
|--|
| <ul style="list-style-type: none"><li>• ... verwenden Variablen und Wertzuweisungen</li><li>• ... modifizieren und ergänzen Quelltexte nach Vorgaben</li><li>• ... interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv“ (Gesellschaft für Informatik e.V. 2008, 16).</li></ul> |
|--|

Sekundarstufe II: „Schülerinnen und Schüler...
--

- |  |
|--|
| <ul style="list-style-type: none"><li>• ... verwenden algorithmische Grundbausteine (Sequenz, Alternative, Wiederholung) und implementieren diese mithilfe einer Programmiersprache“ (Gesellschaft für Informatik e.V. 2016, S. 10).</li></ul> |
|--|

Die hier beschriebenen Kompetenzen sind den in Abbildung 4 dargestellten Wissensprozessen *Anwenden* und *Analysieren* zuzuordnen, sodass sich hier die Verbindung zwischen Zielen und Methoden findet. Denn beim Spielen muss die Spielwelt und das sich stellende Problem analysiert, Manipulationsmöglichkeiten erkannt und mit zu lernenden Befehlen umgesetzt werden, um die sich stellenden Herausforderungen zu meistern. Dadurch werden sowohl die genannten inhaltsbezogenen Kompetenzen am Beispiel von *Lite-C* geschult, während sich durch die Spielumgebung und sich dort stellenden Probleme die Notwendigkeit der Analyse und Anwendung ergeben. Etwa, wenn man beim Springen über einen Abgrund immer an einen Block stößt, der verhindert, dass man auf die andere Seite kommt und man dann die Sprunghöhe der Spielfigur durch Code-Eingaben wieder verkleinert, weil das Hindernis sonst nicht überwunden werden kann.

## **Unterschiedliche Mediennutzung der Schüler\*innen**

Zwar kann durch das Gesagte die Nutzung digitaler Spiele in informatischen Bildungskontexten allgemein empfohlen werden. Doch durch die Gegebenheit „situationaler und milieuspezifischer Problem- und Motivations- bzw. Interessenslagen“ (Kutscher 2009, 9f.) kann die Nutzung digitaler Spiele zur Schulung informatischer Kompetenzbestände und weiterer Aspekte, wie der Lernkompetenz, insbesondere für die Schüler\*innen sinnvoll sein, die aus sozioökonomisch gesehen niedrigeren Milieus kommen. Denn es konnte gezeigt werden, dass deren Mediennutzung eher *unterhaltungs-*, als *informationsorientiert* ist (vgl. ebd.). Das heißt, dass Medien eher zu Unterhaltungszwecken als zu Informationszwecken genutzt werden, vor allem

Bildschirmmedien. Selbstverständlich heißt das nicht, dass es nie vorkommt, dass informationsorientierte Mediennutzung stattfindet. Doch überwiegen bei „[...] der Informationssuche [...] Themen aus dem Unterhaltungs- bzw. Fernsehsektor wie z.B. Informationen zu Lieblingsserien, Lieblingsstars etc.“ (Kutscher 2009, 2), während informationsorientierte Nutzer\*innen Medien eher (auch) zum Lernen von Inhalten nutzen, die in schulischen Kontexten als inkorporiertes kulturelles Kapitel (vgl. Bourdieu 1992) genutzt werden können und ihnen dort Vorteile verschaffen.

Für die Schüler\*innen mit eher unterhaltungsorientierter Mediennutzung können Aufgabenformate darüber hinaus eine große Herausforderung darstellen, wenn diese mit informationsorientierter Mediennutzung verbunden sind. Denn dabei müssen sie nicht nur die angestrebten Kompetenzen erwerben, sondern gleichzeitig (oder erst einmal) die Kompetenzen erwerben, die notwendig sind, um Medien informationsorientiert zu nutzen. „Dieses Problem ist insbesondere an Hauptschul- und Migrationsmilieus ein Benachteiligungsfaktor“ (Niesyto 2009, 872). Um dieser Gruppe von Schülerinnen und Schülern aber ebenfalls eine gute informatische Grundbildung zu ermöglichen, ist es sinnvoll, deren „mediale Habitusformen“ (ebd.) aufzugreifen und für Lehr-Lernprozesse fruchtbar zu machen. Und dies kann eben auch heißen, digitale Spiele als Hauptmedium informatischer Bildungsprozesse zu verwenden, um informatische Kompetenzen zu vermitteln. Denn dadurch müssen sie nicht gleichzeitig informatische Inhalte *und* die informationsorientierte Mediennutzung lernen, während andere gleichaltrige Schüler\*innen Letzteres bereits können und sich auf ersteres fokussieren können. Vielmehr können sie sich so ebenfalls auf die zu lernenden Inhalte fokussieren und haben gegenüber Schüler\*innen mit anderem Medienhabitus keinen Nachteil.<sup>5</sup>

---

<sup>5</sup> Man könnte einerseits einwenden, dass es ungut wäre, sich *immer* am Medienhabitus der Schüler\*innen zu orientieren. Erstens, weil er in Klassen, aufgrund der unterschiedlichen sozioökonomischen Verortung der Schüler\*innen, sowieso nie einheitlich ist und es deshalb nicht möglich ist, sich an *dem* Habitus zu orientieren. Zweitens würden die Schüler\*innen, wenn man dies täte, keine Mediennutzungskompetenzen erwerben, die außerhalb ihres eigenen Medienhabitus liegen, was aber erklärtes Ziel von Bildungsprozessen ist (z.B. als Teil der Lernkompetenz). Darauf antworten wir, dass das stimmt und man nicht immer und überall Bildungsprozesse gestalten soll, die zu eher unterhaltungsorientierter Mediennutzung passen. Man sollte dies aber eben auch nicht vernachlässigen und, wo es passt, Unterrichtseinheiten so gestalten, dass die verschiedenen Medienhabitus beachtet werden, sodass die Schüler\*innen in ihrer Unterschiedlichkeit bedacht werden.

## Fazit

Als Fazit lässt sich deshalb festhalten: Digitale Spiele, wie *Debugger 3.16 Hack'n'Run* (2017), für informatische Bildungsprozesse zu nutzen, kann aus lerntheoretischen, fachdidaktischen und nicht zuletzt auch Gerechtigkeitsgründen empfohlen werden. Sie stellen eine gute Möglichkeit dar, einen sonst in schulischen Kontexten wenig(er) beachteten Medienhabitus mit ein zu beziehen und sind eine Chance zur Umsetzung problemorientierter informatischer Bildungsprozesse, die das Prädikat *Problemorientierung* wirklich verdienen. Denn aufgrund dessen, dass sie Spieler\*innen vor Herausforderungen stellen, gleichzeitig aber – als Spiele – über eine motivationale Kraft verfügen, die in Spieler\*innen den Wunsch erzeugt, diese zu lösen, also den Zielzustand zu erreichen, werden die Herausforderungen zu Problemen. Diese motivieren sie, die Kompetenzen zu erwerben, die für ihre Lösung notwendig sind. Sind die dafür zu erwerbenden Kompetenzen, wie in *Debugger 3.16 Hack'n'Run* (2017), solche, die in informatischen Bildungsprozessen erworben werden sollen, ist ihre Nutzung dringend empfohlen.

---

Andererseits könnte man einwenden, dass die Fokussierung auf unterhaltungsorientierte Mediennutzung diejenigen Schüler\*innen benachteiligt, die Medien eher informationsorientiert nutzen. Dieser Vorwurf läuft aber zum einen ins Leere, wenn man, wie beispielsweise die JIM-Studie 2019 zeigt (S. 44), bedenkt, dass sich die Häufigkeit der Nutzung digitaler Spiele bei Gymnasiast\*innen und anderen Sekundarstufenschüler\*innen nicht allzu sehr unterscheidet. Soll heißen: Auch Schüler\*innen, denen ein eher informationsorientierter Medienhabitus unterstellt werden kann, sind tendenziell fähig, digitale Spiele zu nutzen, während andersherum – unterhaltungsorientierte Nutzer\*innen sollen Medien informationsorientiert nutzen – weniger Berührungspunkte vorhanden sein dürften. Zum anderen ist zu beachten, was vorher bereits gesagt wurde. Wir plädieren hier nicht dafür, informatische Bildungsprozesse nur noch mit digitalen Spielen zu gestalten. Denn auch unterhaltungsorientierte Nutzer\*innen sollen lernen Medien informationsorientiert zu nutzen. Aber wir behaupten, dass es bedenkenswert ist – und zwar im positiv gemeinten Sinne des Wortes – digitale Spiele in informatischen Bildungskontexten, wo es sich anbietet, zu verwenden, um damit und anderen Aufgabenformaten den jeweiligen in der Klasse vorkommenden Medienhabitus entgegenzukommen und niemanden aufgrund desselbigen zu benachteiligen.

## Verwendete Quellen

Bourdieu, Pierre (1992): Ökonomisches Kapital – Kulturelles Kapital – Soziales Kapital. In: Bourdieu, Pierre: Die verborgenen Mechanismen der Macht. Hamburg: VSA.

Edelmann, Walter; Wittmann, Simone (2012): Lernpsychologie. Weinheim: Beltz Verlag.

From Software (2011) Dark Souls.

From Software (2014) Dark Souls II.

From Software (2016) Dark Souls III.

Gesellschaft für Informatik e.V. (2008): Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I. URL:

[https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/bildungsstandards\\_2008.pdf](https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/bildungsstandards_2008.pdf) [Letzter Zugriff: 30.08.2020].

Gesellschaft für Informatik e.V. (2016): Bildungsstandards Informatik für die Sekundarstufe II. URL:

[https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/Bildungsstandards\\_SII.pdf](https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/Bildungsstandards_SII.pdf) [Letzter Zugriff: 30.08.2020].

Haffelder, Maximilian; Maisenhölder, Patrick (2019): Digitale Selbstgeißelung? Oder: Warum SpielerInnen *Souls*-like Spiele spielen. In: Inderst, Rudolf Thomas / Wagner, Pascal / Zurschmitt, Christof: Prepare to Die. Interdisziplinäre Perspektiven auf *Demon's Souls*, *Dark Souls* und *Bloodbourne*. Glückstadt: Verlag Werner Hülsbusch. S. 33-47.

JIM-Studie 2019: Jugend, Information, Medien. Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger. Herausgegeben von: Medienpädagogischer Forschungsverband Südwest. URL:

[https://www.mfps.de/fileadmin/files/Studien/JIM/2019/JIM\\_2019.pdf](https://www.mfps.de/fileadmin/files/Studien/JIM/2019/JIM_2019.pdf) [Letzter Zugriff: 30.08.2020].

Klimmt, Christoph (2006): Computerspielen als Handlung. Dimensionen und Determinanten des Erlebens interaktiver Unterhaltungsangebote. Köln: Herbert von Halem Verlag.

Kutscher, Nadia (2009): Ungleiche Teilhabe. Überlegungen zur Normativität des Medienkompetenzbegriffs. In: Online-Zeitschrift MedienPädagogik 2009/17. URL <https://www.medienpaed.com/article/view/110> [Letzter Zugriff: 30.08.2020].

Neitzel, Britta (2018): Narration. In: Feige, Daniel M.; Ostritsch, Sebastian; Rautzenberg, Markus: Philosophie des Computerspiels. Theorie – Praxis – Ästhetik. Stuttgart: J. B. Metzler. S. 43-53.

Niesyto, Horst (2009): Medienpädagogik und soziale Benachteiligung. In: Mertens, Gerhard / Frost, Ursula; Böhm, Winfried; Ladenthin, Volker: Handbuch der Erziehungswissenschaft, Band III/2. Paderborn: Verlag Ferdinand Schöningh, S. 871-876.

Nintendo EAD (1992): Super Mario World.

Ostritsch, Sebastian (2018): Ethik. In: Feige, Daniel M.; Ostritsch, Sebastian; Rautzenberg, Markus: Philosophie des Computerspiels. Theorie – Praxis – Ästhetik. Stuttgart: J. B. Metzler. S. 77-96.

Pohlmann, Horst (2007): Überwältigt von der Spieleflut? – Genrekunde. In: Kaminski, Winfred; Witting, Tanja: Digitale Spielräume. Basiswissen Computer- und Videospiele. München: kopaed. S. 9-34.

Spiderwork Games (2017): Debugger 3.16: Hack'n'Run.

Tavinor, Grant (2009): The Art of Videogames. Jefferson, North Carolina: Mc Farland.

Winkel, Sandra; Petermann, Franz; Petermann, Ulrike (2006): Lernpsychologie. Paderborn: Schöningh.

Welsh, Timothy J. (2006): Everyday Play: Cruising for Leisure in San Andreas. In: Garrelts, Nate (Hrsg.): The Meaning and Culture of Grand Theft Auto: Critical Essays. Jefferson: Mc Farland. S. 127-142.

Zendler, Andreas; Klaudt, Dieter (2015): Zwanzig Unterrichtsmethoden für den Informatikunterricht: Bewertung des Lernvorgangs durch Informatiklehrer. In: Notes on Educational Informatics – Section A: Concepts and Techniques 2015/11. S. 37-63. URL:

[https://www.ph-ludwigsburg.de/fileadmin/subsites/2e-imix-t-01/user\\_files/Journal\\_NEI\\_-](https://www.ph-ludwigsburg.de/fileadmin/subsites/2e-imix-t-01/user_files/Journal_NEI_-)

[PDFs fuer Webauftritt/Section A/Volume 8 No 1 2012/NEI Section A Vol. 1 1 No. 1 2015 S.37-63. Zendler Klaudt -](#)

[Zwanzig Unterrichtsmethoden fuer den Informatikunterricht.pdf](#) [Letzter Zugriff: 30.08.2020).

## **Autor\*inneninformationen**

Maisenhölder, Lena-Marie, Jg. 1995, Studentin an der Pädagogischen Hochschule Ludwigsburg, Lehramt PO 2015/Sekundarstufe I: Mathematik, Evangelische Theologie (Master) und Bildungsinformatik (Erweiterungsstudium).

Maisenhölder, Patrick, Jg. 1990, Akademischer Mitarbeiter an der Universität Stuttgart und der Pädagogischen Hochschule Ludwigsburg, jeweils im Institut für Philosophie. Studierte Englisch, Ethik, Politikwissenschaften (Staatsexamen, Sekundarstufe I), Medienpädagogik (Erweiterungsstudium) sowie Empirische Bildungsforschung mit Schwerpunkt Medienbildung (Master).