

Niklas Lente

# Spielend Programmiersprache lernen

Mit Serious Games IT-Kenntnisse im bibliothekswissenschaftlichen Studium vertiefen

---

## TYP DES DOKUMENTS | TYPE OF THE DOCUMENT

Magazinartikel | Journal Article

## Nachnutzung | Reuse

Diese Publikation steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International (CC BY 4.0 International). Sofern die Namen der Autor\*innen/ Rechteinhaber\*innen genannt werden, kann der Inhalt vervielfältigt, verbreitet, öffentlich aufgeführt und kommerziell genutzt werden. Außerdem dürfen Bearbeitungen angefertigt und verbreitet werden. Weitere Informationen und die vollständigen Bedingungen der Lizenz finden Sie hier: <https://creativecommons.org/licenses/by/4.0/deed.de>.



**Zeitschriftenartikel**

Begutachtet

**Begutachtet:**Jessica Korschewski 

HAW Hamburg

Deutschland

Prof. Dr. Ulrike Verch 

HAW Hamburg

Deutschland

**Erhalten:** 03. Dezember 2025**Akzeptiert:** 07. Januar 2026**Publiziert:** 11. Februar 2026**Copyright:**

© Niklas Lente.

Dieses Werk steht unter der Lizenz

Creative Commons Namens-

nennung 4.0 International (CC BY 4.0).

**Empfohlene Zitierung:**

Lente, Niklas (2026): Spielend Programmiersprache lernen: Mit Serious Games IT-Kenntnisse im bibliothekswissenschaftlichen Studium vertiefen, in: *API Magazin*, Jg. 7, Nr. 1, doi: [10.15460/apimagazin.2026.7.1.287](https://doi.org/10.15460/apimagazin.2026.7.1.287)

# Spielend Programmiersprache lernen: Mit Serious Games IT-Kenntnisse im bibliothekswissenschaftlichen Studium vertiefen

Niklas Lente<sup>1\*</sup> <sup>1</sup> Hochschule für Angewandte Wissenschaften Hamburg, Deutschland

Studierender im 5. Semester des Bachelorstudiengangs Bibliotheks- und Informationsmanagement

\* Korrespondenz: [redaktion-api@haw-hamburg.de](mailto:redaktion-api@haw-hamburg.de)

## Zusammenfassung

Im Beitrag werden zwei Serious Games vorgestellt, mit denen das Lernen von Auszeichnungs- und Programmiersprachen unterstützt wird. Zuerst wird das Spiel Flexbox Zombies vorgestellt, das sich zum besseren Verständnis von Cascading Style Sheets eignet, und anschließend das Spiel The Farmer was replaced, mit dem man Python lernen kann. Der Autor des Beitrags hat beide Spiele getestet und gibt seine Spielerfahrungen wieder, insbesondere auch in Hinblick auf die Anforderungen aus den IT-Lehrveranstaltungen im Bachelorstudiengang Bibliotheks- und Informationsmanagement an der HAW Hamburg.

**Schlagwörter:** Serious Games, CSS, Python, Programmieren, Bibliotheks- und Informationsmanagement

## Learningprogramming languages through play: Deepening IT skills in library science studies with serious games

### Abstract

This article presents two serious games that support learning markup and programming languages. First, it introduces Flexbox Zombie, a game designed to improve understanding of cascading style sheets, followed by The Farmer was replaced, a game that teaches Python. The author of the article has tested both games and shares his gaming experiences, particularly with regard to the requirements of the IT courses in the bachelor's degree program in Library and Information Management at HAW Hamburg.

**Keywords:** Serious Games, CSS, Python, Life Sciences, Programming, Library and Information Management

## 1 Einleitung

Anders als von vielen meiner Kommiliton\*innen und mir erwartet, werden auch im bibliothekswissenschaftlichen Studium Technikinhalte und IT-Kompetenzen im hohen Umfang vermittelt, unter anderem Grundkenntnisse im Programmieren.

Der Einstieg war für mich durch meine Erfahrungen zum Glück weitestgehend niedrigschwellig. Doch HTML im ersten Semester war meine Nemesis: Ich fand es zunächst sehr unleserlich und fügte auf gut Glück Container in Container ein, kämpfte mit Größeneinstellungen und dem Einbau weiterführender Links. Alles in der Hoffnung, dass es am Ende die Form annimmt, welche ich wollte.

Als im vierten Semester die Programmiersprache Python auf dem Stundenplan stand, hatte ich Respekt vor dieser Lernaufgabe. In der Vergangenheit hatte ich bereits einmal versucht, Python für ein Projekt zu benutzen, und habe dabei nur sehr schwer Fortschritte gemacht. Auch die im Curriculum vorgegebene Verbindung zur Computerlinguistik wirkte zusätzlich herausfordernd.

Beim Lernen fiel mir auf, dass ich viele Kenntnisse, die ich für das Erlernen von HTML, CSS oder Python benötige, schon in Spielen mitbekommen hatte. Sei es das Konzipieren von Logiken wie in Factorio oder den Aufbau von If-Statements und Schleifen, welches ich durch Spiele wie Autonauts gelernt habe.

Aufgrund dieser Erfahrung, dass mir Serious Games beim Programmieren geholfen haben, möchte ich anderen Studierenden in diesem Beitrag zwei Spiele vorstellen und an die Hand geben. Diese Spiele waren für mich nicht nur unterhaltsam, sondern beschäftigen sich direkt mit den Inhalten aus dem Studium und bieten eine Hilfestellung beim Aneignen von IT-Kompetenzen.

## 2 Was sind Serious Games?

Spiele zum Lernen zu benutzen, ist ein Konzept, welches schon im letzten Jahrhundert angewendet wurde. Die ersten Spiele dieser Art waren Mathelernspiele in den 1970ern wie z. B. Plato (vgl. [Egenfeldt-Nielsen 2005](#): 39-40). Der Begriff Serious Games wurde aber erst um die Jahrtausendwende herum von Ben Sawyer geprägt (vgl. [Jacob und Teuteberg 2017](#): 99). Dennoch ist es schwer, eine einheitliche Definition zu finden. In *Serious Games – An Overview* fassen Susi et al. mehrere Definitionen zusammen und arbeiten den Kernpunkt heraus, dass Serious Games Spieltechnologie nutzen, um ein anderes Ziel als Unterhaltung oder Spaß zu verfolgen. (vgl. [Susi u. a. 2015](#): 7)

Diese Definition umfasst alle Spiele, die mit der Hauptintention, Spielenden etwas beizubringen, programmiert wurden. Darunter fallen z. B. der *Microsoft Flight Simulator* oder *Hidden Codes*, ein Spiel der Bildungsstätte Anne Frank. Spiele wie King-

dom Come Deliverance 2, welches das Leben im Mittelalter realitätsnah nachzustellen versucht und ein umfangreiches Lexikon im Spiel bietet, fallen aber nicht in die Definition, da Wissensvermittlung nur ein Extra darstellt und nicht die Hauptintention ist.

### 3 Das Spiel *Flexbox Zombie* zum Erlernen von CSS

Cascading Style Sheets (CSS) sind eine wichtige Programmiersprache, um Webseiten in ihre gewünschte Form zu bringen. Ein Teil von CSS sind Flexboxes, die einem helfen, den Inhalt einer Webseite flexibel darzustellen. Da diese einige Parameter zum Einstellen haben, benötigt man etwas Übung, bis man ihr volles Potential nutzen kann. Ein Weg, um den Einstieg mit Flexboxes zu erleichtern, ist das Spiel *Flexbox Zombies* von Dave Geddes.

*Flexbox Zombies* ist ein Spiel, das kosten- und werbefrei auf der Webseite [flexbox-zombies.com](https://flexbox-zombies.com) spielbar ist. Jedoch wird zum Spielen ein Konto bei Mastery Games gebraucht. Theoretisch kann es auf PCs, Tablets und Mobiltelefonen gespielt werden, jedoch empfiehlt sich das Spielen auf einem großen Bildschirm und mit Tastatur, da es sonst unübersichtlich und gedrängt wirkt.



Abbildung 1: *Flexbox Zombies* Level 3-11. Quelle: *Flexbox Zombies* (2017): PC [Videospiel], o.O.: Dave Geddes

Das Layout des Spiels besteht aus dem Editorfeld, das als Eingabe dient, einem Erzählerfeld, in welchem der Nichtspielercharakter in Form eines Begleiters Erklärungen oder Hinweise gibt, und dem Spielfeld, auf dem sich die Spielerfigur und die Gegner befinden (Siehe Abbildung 1). Ziel ist es, Untote mit einer Armbrust ins Visier zu nehmen, dargestellt durch grau schraffierte Kästchen. Die Waffe steuert das Format und die Ausrichtung einer Flexbox mithilfe der entsprechenden CSS-Parameter wie `flex-direction`, `justify-content` oder `align-items`. Jedes Level dauert, je nachdem wie schnell man auf die Lösung kommt, nur wenige Minuten. Ein Kapitel bildet dabei eine Lerneinheit, besteht aus mehreren Levels und ist nach circa einer halben Stunde durchgespielt.

### 3.1 Meine Spielerfahrungen

Das Spiel ist stimmig aufbereitet. Die Comicgrafik in der düsteren Ästhetik passt sich gut in das Setting ein. Auch das Userinterface fand ich durch die Schlichtheit sehr angenehm und nicht künstlich aufgebläht. Nichts blinkt andauernd, nichts lenkt ab, und ich konnte mich so gut auf den Text und die Aufgabe konzentrieren.

In jedem Level erwartet mich eine Aufgabenstellung, die durch die Auswahl und Platzierung der einzelnen Untoten dargestellt ist. In den ersten Leveln eines Kapitels wird eine neue Spielmechanik in Form eines CSS-Parameters oder einer Story erklärt. Letztere wurde für mich aber sehr schnell nebensächlich und ich begann lieber direkt mit den Aufgabenstellungen.

In den Leveln selbst baut das Spiel keinen Druck auf. Die Untoten kommen nicht näher, es läuft keine Zeit ab und es gibt kein Verlieren. Ich hatte alle Zeit, um auf die Lösung zu kommen. Auch die Hinweisfunktion, welche ich nach mehreren Tagen Pause zwischen einzelnen Modulen brauchte, machte das Angebot für mich sehr niedrigschwellig. Außerdem fiel mir der Wiedereinstieg nach einer Pause dadurch leichter, weil ich vergangene Lektionen nicht wiederholen musste, sondern mein Wissen durch Tipps selbst wieder auffrischen konnte.

Nach vier Kapiteln und circa zwei Stunden Spielzeit bemerkte ich, dass der Inhalt tiefer geht als das, was ich im ersten Semester benutzt hatte. Dennoch spielte ich weiter, da es mit mehr Funktionen auch wieder spannender wurde und auch neue, durch z. B. Gegnertypen wie beispielsweise Schleime, ergänzt wurden.

### 3.2 Meine Spielbewertung

Das Spiel hat mir am Anfang viel Spaß gemacht. Ich fühlte mich durch die Einleitungen und die Hinweisfunktion an die Hand genommen und wusste immer, was ich zu tun hatte. Da liegt jedoch für mich auch das Manko dieses Spiels: jedes Level hat immer genau eine Lösung. Es kann zwar variiert werden, in welcher Reihenfolge die Parameter definiert werden, aber dennoch gibt es nur eine Einstellung, die zum Sieg führt. Wenn ein Zombie links am Bildschirmrand ist, müssen die Eingaben "display: flex", "flex-direction: row-reverse" und "justify-content: flex-end" lauten, sonst ist das Level nicht geschafft. So fühlte es sich für mich irgendwann mehr nach einer Gamification von Karteikartenlernen an als nach einem wirklichen Spiel. Das passt zu den Flexboxes, da es nur einen Weg gibt, etwas darzustellen, es passt aber nicht zu meinem Lernstil.

Dennoch würde ich das Spiel jedem empfehlen, der sich mit Flexboxes auseinandersetzen möchte. Es führt einen angenehm durch alle Funktionen und zeigt, was alles möglich ist. Damit eröffnet es gute Ideen für die Gestaltung mit CSS.

## 4 Das Spiel *The Farmer was replaced* zum Erlernen von Python

Python ist eine weit verbreitete Programmiersprache und durch ihren sehr modularen Aufbau durch sogenannte Libraries für viele Zwecke benutzbar. Im vierten Semester haben wir Python genutzt, um z. B. Textkorpora aufzubauen, diese Texte weiterzubearbeiten und schlussendlich mit Algorithmen eine Auswertung vorzunehmen. Um den Inhalt zu verstehen und die Aufgaben zu bearbeiten ist es hilfreich, die Syntax von Python zu lernen. Eine Hilfestellung dafür bietet *The Farmer was replaced* vom Schweizer Entwickler Timon Herzog.

Das Spiel ist erhältlich auf der Plattform Steam und kostet 9,99 €. Da es leider keine alternative Möglichkeit gibt, das Spiel zu erwerben, benötigt man zum Spielen ein Steam Konto. Auch ist es für Windows PCs ausgelegt, weswegen die Funktionalität auf Mac oder Linux Geräten nur durch Kompatibilitätsmaßnahmen gegeben ist.

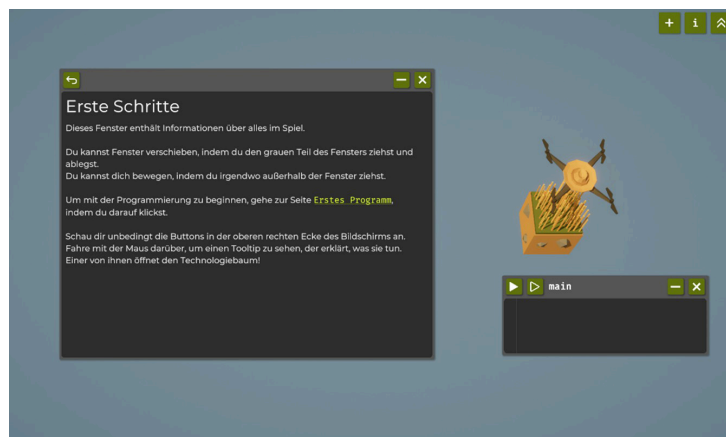


Abbildung 2: Der Startbildschirm von *The Farmer was Replaced*, bestehend aus einem Informationstext, einem Codeeditor und der programmierbaren Drohne über einem Feld. Quelle: *The Farmer was Replaced* (2025): PC [Videospiele], Zürich: Metaroot

Ziel des Spiels ist es, eine Drohne mit Python zu programmieren, um Ressourcen einzusammeln. Damit schaltet man mehr Möglichkeiten frei, wie zum Beispiel die Nutzung weiterer Ressourcen oder neue Funktionen in Python wie Listen, Dictionaries oder Variablen. Die freigeschalteten Inhalte können dann verwendet werden, um schon geschriebenen Code anzupassen oder komplett neue Programme zu schreiben.

Das Spiel ist visuell simpel aufgebaut. Es besteht aus verschiedenen Fenstern, welche einem entweder Mechaniken erklären oder als Codeeditor dienen, einer Bedienoberfläche und einer Drohne, welche die geschriebenen Programme ausführt (siehe Abbildung 2).

### 4.1 Meine Spielerfahrungen

Da ich schon einige Erfahrungen mit Python hatte, war der Anfang von *The Farmer was replaced* niedrigschwellig. Die ersten Schleifen und If-Konditionen hatte ich

schnell geschrieben und schickte die Drohne so über das immer größer werdende Feld.

Für den Spielablauf ist es egal, wie gut ein Code geschrieben ist. Man kann einzelne Schritte in einzelne Programme schreiben und bei Bedarf ausführen, oder alles in If-Konditionen schachteln und zuschauen, wie die Drohne diese automatisiert erledigt. Im Spiel erfolgt keine Bewertung. Die einzigen messbaren Größen sind die Ressourcen, welche sich schneller oder langsamer anhäufen. Für mich funktionierte diese wertungsfreie Sandbox sehr gut und trieb mich an, bessere Lösungen zu finden, um die Drohne schneller und effizienter über das Feld fliegen zu lassen.

Und so habe ich nach jedem neuen Schritt auf dem Technologiebaum, welcher neue Funktionen freischaltet, überlegt, in welches Programm ich die neue Funktion einbauen kann. Nach zwei Stunden saß ich dann vor dem ersten Code, bei dem ich zunächst mehrere Bugfixes ausprobieren musste, damit die Drohne wirklich genau das tat, was ich wollte. Nach einigem Grübeln flog die Drohne übers Feld, pflanzte abwechselnd Bäume und Getreide, wechselte die Pflanzreihe und arbeitete so das ganze Feld ab. Ich hatte mein Vorhaben umgesetzt und dabei mein Verständnis von Python vertieft.

Das passierte mir auch mit vielen der darauffolgenden Technologien, die ich freischaltete. Und von diesen gibt es viele, wie z. B. zeitgleich gesteuerte Drohnen, durch die das Spiel stark beeinflusst werden können.

#### 4.2. Meine Spielbewertung

Das Spiel geht in die Tiefe von Python und hatte bei mir Erfolg, weil es einen nicht komplett an die Hand nimmt. Es gibt nur einen Werkzeugkasten, eine Aufgabe und viele Möglichkeiten, diese Aufgabe zu lösen. Ich konnte mich somit ausprobieren und dass die Drohne am Ende so flog wie ich wollte, war mir Belohnung genug.

Da ist aber auch gleichzeitig die Kehrseite, weswegen ich glaube, dass es nicht für alle Studierenden geeignet ist. Es gibt keine extrinsische Motivation das Programmieren zu lernen, das Spiel macht meiner Meinung nur dann Spaß, wenn die Person an der Tastatur wirklich programmieren lernen möchte. Dazu muss man bereit sein, Erklärtexte zu lesen und verschiedene Varianten auszuprobieren. Wie beim Programmieren im Studium muss die Frustrationstoleranz in manchen Situationen hoch sein, man darf nicht verzweifeln, wenn der erste Code nicht funktioniert. Sobald man aber das Spiel beherrscht und mit allen Features umgehen kann wird einem Python in dem Umfang, wie es in unserem Bachelorstudium vorkommt, nicht mehr überraschen.

## 5 Fazit

Serious Games können dabei helfen, Inhalte für das Studium leichter zu erschließen. Meine eigenen Erfahrungen mit den beiden vorgestellten Spielen haben mir geholfen meine Programmierkenntnisse zu vertiefen, obwohl ich schon ein wenig Vorwissen hatte. Allerdings gehöre ich auch zu den Personen, die Spaß am Programmieren haben, und damit zur Zielgruppe dieser Spiele. Für Studierende, die nie Kontaktpunkte mit Programmieraufgaben hatten, eignen sich eher Spiele, die stärker die Grundlagen erklären, wie das Spiel **Autonauts**,<sup>1</sup> welches eine Scratch-ähnliche Umgebung benutzt.

Auch sind die Spiele nicht speziell für das bibliothekswissenschaftliche Studium ausgelegt und geben wesentlich mehr Wissen mit, als wir im Studium brauchen. Das kann aber auch hilfreich sein, wenn das Wissen später im Berufsleben gebraucht wird.

Insbesondere die Anfänge der beiden Spiele sind meiner Meinung nach für jeden zu empfehlen. Sie erleichtern den Einstieg in die Materie und können sowohl die Angst vor dem Programmieren nehmen als auch Fortschritte durch die Visualisierungen greifbar machen.

---

<sup>1</sup> Autonauts von Denki ist ein Automationsspiel, in dem die Spielenden eine Scratch-ähnliche Umgebung benutzen, um z.B. Roboter zu programmieren oder Ressourcen abzubauen und weiterzuverarbeiten. Quelle: Autonauts (2019): PC[Videospiel], Dundee: Denki.

## Literatur

EGENFELDT-NIELSEN, Simon (2005): Beyond Edutainment Exploring the Educational Potential of Computer Games, Kopenhagen: IT-University of Copenhagen.

JACOB, Axel; TEUTEBERG, Frank (2017): Game-Based Learning, Serious Games, Business Games und Gamification –Lernförderliche Anwendungsszenarien, gewonnene Erkenntnisse und Handlungsempfehlungen, in: Susanne Strahinger und Christian Leyh (Hrsg.), Gamification und Serious Games, Wiesbaden: Springer Fachmedien Wiesbaden (Edition HMD), S. 97–112, doi: [10.1007/978-3-658-16742-4\\_8](https://doi.org/10.1007/978-3-658-16742-4_8).

SUSI, Tarja; JOHANNESSON, Mikael; BACKLUND, Per (2015): Serious Games - An Overview, Skövde: School of Humanities and Informatics University of Skövde.

## Abbildungsverzeichnis

GEDDES, Dave (2017): Flexbox Zombies, o.O.

O. A. (2025): The Farmer was replaced, Zürich: Metaroot.