

Alexandra Svedkijs<sup>1</sup>  
Jens-Peter Knemeyer<sup>1</sup>  
Nicole Marmé<sup>1</sup>

<sup>1</sup>Pädagogische Hochschule Heidelberg

### **Programmiersprache Snap! im Physikunterricht „Simulation des Bremsweges“**

#### **Informationstechnische Kompetenzen im Schulunterricht**

Viele Lebensbereiche werden von der voranschreitenden Digitalisierung in Deutschland nachhaltig verändert (Bundesministerium für Wirtschaft und Energie, 2016), so dass informationstechnische Kompetenzen für die zukünftige Arbeitswelt und eine gesellschaftliche Teilhabe von essenzieller Bedeutung sind (Zika et al., 2017). Die informationstechnischen Kompetenzen stellen einen fundamentalen Baustein für die Gesellschaft („digital literacy“) dar (Martin, 2006). Der Anschluss an die digitalisierte Infrastruktur setzt starke Ausprägung dieser Kompetenzen voraus (Höhne et al., 2017). Die informations- und kommunikationstechnologische Kompetenz wird beispielsweise durch das World Economic Forum dem Bereich der Schlüsselkompetenzen im 21. Jahrhundert gleichbedeutend gesehen mit Lese-, Schreib-, Rechenfähigkeit, kultureller Bildung, Bürgerkompetenz, wissenschaftlicher Kompetenz und finanzieller Allgemeinbildung (World Economic Forum, 2015).

Um die nächsten Generationen auf diese Herausforderungen vorzubereiten, sind die Schulen gefordert, einen entsprechend zeitgemäßen und technologisch fortschrittlichen Unterricht anzubieten, in dem Digital- und Informatikkompetenzen eine zentrale Rolle spielen. Die Bildungsangebote sollten mit dem technologischen Wandel synchronisiert werden und selbstgesteuert und bedarfsorientiert sein (acatech, 2016). Der Unterricht soll darauf bedacht sein, digitale Kompetenzen, darunter die informationstechnische Kompetenz zu vermitteln. Insbesondere die naturwissenschaftlichen Fächer scheinen für die Vermittlung der informationstechnischen Kompetenz im Anwendungskontext geeignet, da sie eine problembasierte Einbettung und projektorientiertes Lernen (Schulz-Zander & Preussler, 2017), beispielsweise durch eine digitalisierte Auswertung der Messergebnisse oder Simulation der Experimente ermöglichen sollten.

#### **Integration von IT-Inhalten in den Physikunterricht**

Der Einsatz digitaler Medien ist vielfach in den Bildungs- und Lehrplänen im Sinne der Leitperspektive Medienbildung vorgeschrieben. (Kultusministerium Baden-Württemberg, 2016) In den aktuellen Leitperspektiven des Bildungsplanes 2016 des Landes Baden-Württemberg wird ein „fächerintegrierter Unterricht“ unter anderem auch für die informationstechnischen Grundlagen gefordert. Konkret wird es im Profulfach IMP, Klasse 9 „Physikalische Abhängigkeiten und Abläufe modellieren und implementieren (mit visuellen Programmiersprachen)“.

Prinzipiell sollte eine Integration aber auch im regulären Physikunterricht möglich sein. In diesem Beitrag wird mit der Simulation des Bremsweges mit „Snap!“ ein Beispiel für die Kinematik (Physik, Klasse 9/10) vorgestellt, welches auch als Ergänzung möglicher Unterrichtsszenarien für eine interdisziplinäre Förderung digitaler Kompetenzen dienen kann. An der Pädagogischen Hochschule Heidelberg wird es in der Lehramtsausbildung im Rahmen des Projektes Zukunftslabor MINT eingesetzt. Dieses von der Vector Stiftung geförderte Projekt zielt darauf ab, Lehramtsstudierende aller MINT-Fächer grundlegende IT-Kompetenzen zu vermitteln, damit sie IT-Inhalte in ihre Fächer integrieren können.

### **Blockbasierte Programmiersprache Snap!**

Snap! ist eine blockbasierte (visuelle) Programmiersprache, mit der direkt im Browser programmiert werden kann. Durch die Übertragung der Konzepte in visuelle Programme eignen sich blockbasierte Programmiersprachen für die Einführung in die Programmierung auch für Lernende ohne jegliche Vorkenntnisse (Maloney et al., 2010). Um ein Programm zu erstellen, werden Befehle aus einer vorhandenen Bibliothek verwendet und miteinander, wie Puzzle zu einem Programm verknüpft. Das Ergebnis der Programmierung ist unmittelbar auf einer Darstellungsfläche, der „Bühne“, zu sehen. Im Vergleich zu gewöhnlichen Programmiersprachen, die eine komplexe Syntax einsetzen, ermöglichen sie eine einfachere Interaktion mit dem Computer: Die Lernenden können sich verstärkt auf Programmierlogik konzentrieren, anstatt sich mit syntaktischen Fehlern auseinanderzusetzen.

### **Simulation des Bremsweges mit Snap!**

Im Rahmen dieses Projektes wird der Bremsvorgang eines Fahrzeuges simuliert und visualisiert. Neben der Festigung physikalischer Inhalte und Training von IT-Fähigkeiten sollen Schüler:innen auf Gefahrensituation sensibilisiert werden: das Fahrzeug fährt während der Reaktionszeit ungebremst weiter und der eigentliche Bremsweg verlängert sich exponentiell mit der Geschwindigkeit. In der Programmierung wird dieser Abschnitt des Bewegungsvorganges explizit programmiert. Für die Implementierung des Projektes ist eine sinnvolle Festlegung verschiedener Parameter, wie Beschleunigung und Reaktionszeit notwendig. Damit soll ein realitätsnaher Bezug zu physikalischen Prozessen und Gegebenheiten hergestellt werden.

### **Durchführung und Aufgaben**

Für die Simulation (Visualisierung) des Bremsvorgang eines Fahrzeuges benötigen die Schüler:innen ihr Wissen über die gleichförmige und beschleunigte Bewegung, um die Simulation des Anhaltewegs zu programmieren. Dabei soll das Objekt Auto zunächst mit einer vorgegebenen Geschwindigkeit fahren und nach einem Bremssignal anhalten. Die Schüler:innen programmieren den Reaktionsweg mit einer Reaktionszeit von einer Sekunde und anschließend den Bremsweg, der (zur Vereinfachung) aus linearisierten Teilabschnitten für die Änderung der Geschwindigkeit besteht. Damit wird eine Annäherung an die verzögerte Bewegung programmiert. In der Darstellung bewegt sich das Auto immer langsamer und hält anschließend an.



Abb. 1: Bühnenbild



Abb. 2: Quellcodebeispiel: Initialisierung

Das Projekt besteht aus vier Objekten: das Auto, die Ampel, der Mensch und die Bühne. Die vollständige Projektlogik wird im Objekt Auto implementiert. Der Quellcode im Objekt Auto besteht aus drei Abschnitten: (1) Initialisierung der Variablen, (2) gleichförmige Bewegung, (3) Bremsvorgang. Für die Initialisierung werden sowohl die Geschwindigkeit, die Zeit und die Beschleunigung auf Beispielwerte gesetzt. An dieser Stelle können die Beispielwerte auch als Benutzereingaben implementiert werden. Außerdem wird für die gleichförmige Bewegung ein Streckenabschnitt (*deltaStrecke*) pro Sekunde ( $s = v \cdot t$ ) und die notwendige Bremszeit ( $t = v/a$ ) berechnet. Die Umsetzung der gleichförmigen Bewegung spiegelt sich hauptsächlich in der Bewegung des Objektes Auto auf der Bühne. Dazu wird eine Startposition, eine Schleife für die Bewegung und ein Abbruchkriterium für die Schleife implementiert. Das Auto bewegt sich so lange gleichförmig, bis das rote Licht der Ampel wahrgenommen wird. Es folgt noch eine Reaktionszeit von einer Sekunde, in der das Objekt ungebremst fährt. Erst nach der Reaktionszeit wird die Nachricht „Bremsen“ geschickt. Der Bremsvorgang beginnt mit dem Empfang einer Nachricht. Dazu wird die zuvor berechnete Bremszeit innerhalb der Schleife heruntergerechnet. Die Änderung der Geschwindigkeit erfolgt in linearisierten Abschnitten mit einem Delta von einer Sekunde. Dabei wird pro Schleifendurchlauf ein fester Wert der Geschwindigkeit abgezogen. Der zurückgelegte Streckenabschnitt wird in jedem Schleifendurchlauf neu berechnet und auf die aktuelle Position des Autos aufaddiert. Damit entsteht eine Simulation einer verzögerten Bewegung mit einem zeitlichen Maßstab von einer Sekunde. Für eine feinere Darstellung kann die zeitliche Auflösung auch deutlich kleiner (bis ca. 0,1 Sekunde) gesetzt werden.

Das Projekt „Bestimmung des Bremsweges“ kann in drei Komplexitätsstufen durchgeführt werden. Auf der einfachsten Stufe werden vorhandene Blöcke für die Berechnungen in die Lücken im Quellcode eingefügt und die Variablen werden auf Startwerte initialisiert. Damit kann das vollständige Projekt ausgeführt werden. Auf der zweiten Komplexitätsstufe müssen vorhanden Anweisungen mit notwendigen Parametern in der richtigen Reihenfolge in das Programm eingesetzt werden. Auf der dritten Komplexitätsstufe muss das vollständige Programm für vorhandene Objekte implementiert werden. Für die Lehrkraft steht eine vollständige Lösung zur Verfügung. Das gesamte Projekt kann unter [www.innovation-tank.de/snap](http://www.innovation-tank.de/snap) abgerufen werden.

### Danksagung

Für die finanzielle Förderung des Projektes Zukunftslabor MINT danken wir der Vector Stiftung.

## Literatur

- acatech. (2016). *Kompetenzen für Industrie 4.0. Qualifizierungsbedarfe und Lösungsansätze (acatech POSITION)*. München: Herbert Utz Verlag.
- Bundesministerium für Wirtschaft und Energie (Hrsg.). (2016). *Digitale Bildung—Der Schlüssel zu einer Welt im Wandel*.
- European Commission. Joint Research Centre. (2022). *DigComp 2.2, The Digital Competence framework for citizens :with new examples of knowledge, skills and attitudes*. Publications Office. <https://data.europa.eu/doi/10.2760/490274>
- Höhne, B. P., Bräutigam, S., Longmuß, J., & Schindler, F. (2017). Agiles Lernen am Arbeitsplatz – Eine neue Lernkultur in Zeiten der Digitalisierung: Agile Methoden, Kompetenzentwicklung, Fachkräftemangel, Digitalisierung, Industrie 4.0. *Zeitschrift für Arbeitswissenschaft*, 71(2), 110–119. <https://doi.org/10.1007/s41449-017-0055-x>
- Kultusministerium Baden-Württemberg (Hrsg.). (2016). *Bildungsplan Baden-Württemberg*.
- Martin, A. (2006). A european framework for digital literacy. *Nordic Journal of Digital Literacy*, 1(2), 151–161. <https://doi.org/10.18261/ISSN1891-943X-2006-02-06>
- Maloney J., Rusk N., Resnick M., Eastmond E. (2010) The Scratch programming Language and Environment. In: ACM Transactions on Computing Education, November 2010
- Schulz-Zander, R., & Preussler, A. (2017). Selbstreguliertes und kooperatives Lernen mit digitalen Medien – Ergebnisse der SITE-Studie und der SelMa-Evaluation. *MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung*, 211–228. <https://doi.org/10.21240/mpaed/retro/2017.08.12.X>
- van Laar, E., van Deursen, A. J. A. M., van Dijk, J. A. G. M., & de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior*, 72, 577–588. <https://doi.org/10.1016/j.chb.2017.03.010>
- World Economic Forum (Hrsg.). (2015). *New Vision for Education—Unlocking the potential of technology*. World Economic Forum.
- Zika, G., Maier, T., Helmrich, R., Hummel, M., Kalinowski, M., Mönning, A., & Wolter, M. I. (2017). *Qualifikations- und Berufsfeldprojektion bis 2035*. 11.