

# Programmieren im Vorschulalter mit Hilfe von Tangicons

Claudia Hahn, Christian Wolters, Thomas Winkler, Michael Herczeg

Institut für Multimediale und Interaktive Systeme, Universität zu Lübeck

## Zusammenfassung

In diesem Beitrag diskutieren wir unsere Vorgehensweise zur Konzeption und zum Co-Design-Prozess einer neuen Variante der be-greifbaren, interaktiven Lern-Applikation Tangicons. Tangicons sollen durch spielerisches Erlernen wesentliche Konzepte der Programmierung vermitteln helfen. Vor dem Hintergrund von Vorgängersystemen wurde ein neues Spiel entwickelt, das auch für Kinder im Alter von vier, möglicherweise auch schon von drei Jahren geeignet ist. Es wird ein Prozess vorgestellt, in dem das Spielkonzept formativ, unter Einbezug der Zielgruppe, optimiert und evaluiert wird.

## 1 Einleitung

Ziel des Beitrags ist die Vorstellung und Entwicklung eines Lernspiels, in dem sich 4-jährige Kinder mit wesentlichen Konzepten des Programmierens auseinandersetzen. Dieses Spiel geht aus den Tangicons hervor, die seit 2006 entwickelt und untersucht werden (Scharf et al. 2008; Winkler et al. 2011; Scharf et al. 2012). Bei den Tangicons handelt es sich um eine Klasse von be-greifbaren Lernspielen, mit denen Kinder bis zum Alter von zehn Jahren programmieren und algorithmisch denken. Um mit diesem Spiel der Altersgruppe der 4-Jährigen gerecht zu werden, müssen deren spezifische Fähigkeiten und Grenzen berücksichtigt werden. Deshalb wird die Spielidee unter Berücksichtigung kognitiver Kompetenzen entwickelt.

Die Grundidee der Tangicons umfasst verschiedene Elemente: In einem Prozess des Begreifens sollen Kinder spielerisch das Programmieren erlernen. Dabei bewegen sie sich im Raum, wodurch auch der Körper relevant ist. Sie spielen miteinander, um über ihr Handeln im Spiel Kommunikations- und Entscheidungsprozesse auszulösen. Grundsätzlich sind im Spiel Input und Output getrennt, so dass keine direkte Manipulation stattfindet und Kommunikationsprozesse angeregt werden.

Die Entwicklung dieser Lernumgebungen orientierte sich an modernen pädagogischen Konzepten (Kösel 2007), die die Bedeutung gemeinsamen Lernens, die feinmotorische Handha-

bung (be-greifbarer) physisch-ikonischer Objekte (tangiblen Icons) sowie grobmotorische Bewegung der Lernenden berücksichtigen. In kollaborativen Lernsituationen wird durch gemeinsame Diskussion und Reflexion eine erfolgreiche Lösung erarbeitet (Scharf et al. 2012). Es fand bereits eine weitgehende Auseinandersetzung mit ähnlichen wissenschaftlichen Vorhaben statt (vgl. Scharf et al. 2008).

Die erste Systemgeneration der Tangicons (Scharf et al. 2008) wurde bereits 2007 in einem partizipativen Entwicklungsprozess mit Kindern im Alter von sechs bis neun Jahren fertig gestellt. Die zweite Systemgeneration wurde dahingehend verändert, dass Sensoren und Funkmodule in die Programmierbausteine des Spiels eingebaut wurden. Dies vereinfachte die Übertragung einer Programmiersequenz sowie die Ausführung des Programms und verbesserte das Erleben des Spiels. Hier gab es erstmals qualitative Hinweise auf eine altersbedingte Untergrenze für den Einsatz der Tangicons (Winkler et al. 2011). Eine wesentliche Neuerung bei der dritten Systemgeneration der Tangicons ist, dass die Spieler einen Avatar über Spielfelder bewegen. Dabei können wir vollständig auf handelsübliche Standardhardware (Laptop und Sifteo Cubes, siehe Merrill et al. 2007) zurückgreifen.

## 2 Programmieren als Problemlöseprozess

Im Folgenden wird Programmieren als ein Problemlöseprozess aufgefasst, das in einfacheren Varianten bereits von Kindern erlernt werden kann. Prozesse des Problemlösens zeichnen sich durch drei wesentliche Merkmale aus (vgl. Anderson 2001): (1) sie sind zielgerichtet, d. h. das Verhalten ist auf ein bestimmtes Ziel hin orientiert; (2) dieses Ziel wird in Teilziele zerlegt; (3) es werden Operatoren zur Transformation von Information angewendet. Dies sind Handlungen, die den vorliegenden Problemzustand in einen anderen Problemzustand transformieren. Das Gesamtproblem, die Aufgabenstellung, wird durch eine Sequenz von Operatoren gelöst.

Ein Programm kann als Sequenz von Befehlen betrachtet werden, die in der Reihenfolge ausgeführt werden, in der sie aufgestellt wurden (Scharf 2007). Zunächst muss also die Bedeutung einer Sequenz verstanden und dann eine Verbindung zwischen Input und Output hergestellt werden. Dabei muss erkannt werden, dass Veränderungen in der Programmierung mit Transformation des Inputs und damit Veränderungen im Output verbunden sind. Folglich wird die Verbindung zwischen Input und Output als Sequenz erkennbar.

Programmieren kann demnach als ein Problemlöseprozess aufgefasst werden, bei dem ein übergeordnetes Ziel in Teilziele zerlegt wird und mit Hilfe einer Sequenz von Operatoren ausgeführt wird. Wesentlich ist hier, dass die kognitiven Fähigkeiten der Zielgruppe berücksichtigt werden, weshalb diese nachfolgend betrachtet werden.

### 3 Kognitive Fähigkeiten

Um für die Altersgruppe der 3- und 4-Jährigen mit einem normalen Entwicklungsverlauf ein adäquates Lernspiel zu entwickeln, ist es erforderlich, deren kognitiven Kompetenzen zu berücksichtigen. Deshalb müssen relevante Konzepte identifiziert werden.

Das Problemlösen und Denken von Kindern verbessert sich, wenn sich eine Reihe von Komponenten mit ihrer kognitiven Entwicklung verändert (vgl. Oerter & Montada 2002): Sie nutzen ihren Arbeitsspeicher besser, sie bilden abstrakter werdende Repräsentationen und nutzen verbesserte Strategien. Bereits Kinder unter drei Jahren gehen beim Problemlösen mit Absicht und Organisiertheit vor, so dass von definierten Problemlösestrategien gesprochen werden kann. Der Wechsel einer Strategie kann als klares Zeichen zielgerichteten Problemlösens gewertet werden (vgl. Oerter & Montada 2002 und Siegler et al. 2005).

Weiterhin ist beim Umgang mit Symbolen die Fähigkeit zur dualen Repräsentation von Bedeutung (DeLoache 2004): Symbolische Artefakte sind sowohl reales Objekt als auch die Repräsentation (Zeichen) von etwas anderem als sich selbst. Diese Fähigkeit nimmt mit dem Alter zu und ist unter bestimmten Umständen ab einem Alter von 2,5 Jahren zu beobachten. Sie ist auch von Eigenschaften der Objekte abhängig, so dass attraktive Objekte von jungen Kindern weniger leicht als Symbole begriffen werden. Da die Sifteo Cubes als Symbole dienen, ist die Fähigkeit zur dualen Repräsentation eine Voraussetzung für erfolgreiches Spielen.

Außerdem sind noch die Entwicklung des Zahlenverständnisses und die Repräsentation von Zahlen durch Symbole von Interesse. Das Phänomen des *Subitizing* (vgl. Siegler et al. 2005) bezeichnet die Kompetenz, Mengen mit bis zu drei Elementen unmittelbar zu erkennen und kann schon im ersten Lebensjahr geleistet werden. Jedoch ist unklar, inwiefern dies auf ein arithmetisches Grundverständnis oder einen Wahrnehmungsprozess zurückzuführen ist. Erst im Alter von drei oder vier Jahren können Kinder Mengen unterscheiden, die nur wenig größer als drei sind (vgl. DeLoache 2004). Die meisten Kinder erlernen das Zählen im Alter von drei Jahren, erst mit fünf Jahren kennen die meisten Kinder die relative Größe (fast) aller Zahlen zwischen 1 und 10 (vgl. Siegler et al. 2005).

Ein weiteres Problem könnte im Verständnis von Ursache und Wirkung liegen. Die meisten 3-Jährigen wissen noch nicht, dass Effekte eine Ursache haben müssen, Vorläufer eines kausalen Verständnisses existieren schon vorher (vgl. Siegler et al. 2005).

### 4 Spiel

Basierend auf dem Spielaufbau der Tangicons der dritten Systemgeneration wurde eine neue Spielidee entwickelt. Es spielen jeweils drei Kinder gemeinsam, damit eine überschaubare Lernsituation geschaffen wird, die Kommunikationsprozesse zulässt.

Die Aufgabe der Spieler ist es, einen Frosch von links nach rechts durch eine Teichlandschaft zu bewegen. Dazu müssen sie eine regelbasierte Sequenz aus Sifteo Cubes erstellen.

Die Darstellung des Teichs dient als Aufgabenstellung für die Spieler, indem sie Spielfelder anzeigt und angemessene Fortbewegungsarten impliziert. Die Abbildungen 1 und 2 visualisieren die Spielidee und gleichzeitig beispielhaft eine Variante des Untersuchungsmaterials, das im Co-Design-Prozess eingesetzt wird.

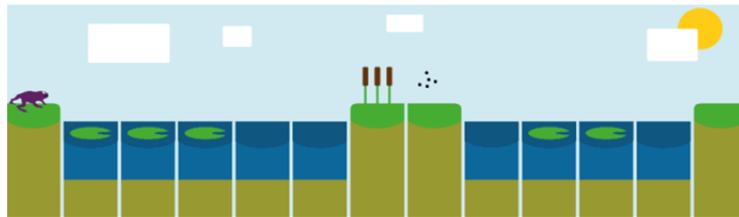


Abbildung 1: Mögliche Visualisierung der Spielfelder

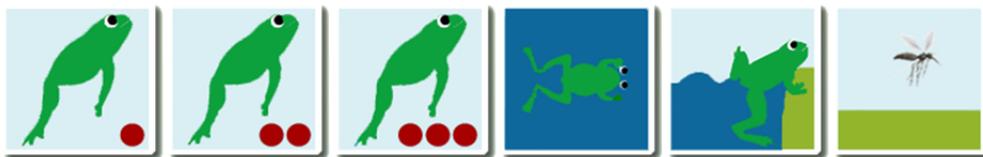


Abbildung 2: Grafischer Entwurf für die Spielsteine (die Anzahl der roten Punkte entspricht der Anzahl der zurückgelegten Spielfelder) und des „Ladesteins“

Der Frosch kann hüpfen, schwimmen, gehen und klettern. Die Spieler müssen die Bewegungsart und die Entfernung, d.h. die Anzahl an Spielfeldern, auswählen. Dazu erhält jedes Kind einen Spielstein. An Land kann der Frosch hüpfen oder gehen - auf Seerosen gelangt er nur durch Hüpfen. Fällt oder springt er ins Wasser, so muss er bis ans nächste Ufer schwimmen, um dort wieder an Land zu klettern. Trifft der Frosch auf eine Mücke, will er sie fressen. Dann müssen die Kinder mit ihrem Spielstein zu einem entfernten „Ladestein“ laufen, um den Frosch zu füttern, damit sie weiterspielen können. Die Kinder lernen das Spiel, indem sie es zunächst gemeinsam mit einem kundigen Erwachsenen spielen. Sie beginnen mit einer einfachen Teichlandschaft, die sukzessive komplexer wird.

Entsprechend den kognitiven Fähigkeiten werden die konzeptuellen und grafischen Entwürfe gestaltet. Deshalb werden nur Zahlen bis 3 berücksichtigt und neben Ziffern auch Symbole in der entsprechenden Anzahl verwendet. Ein weiterer Aspekt ist, dass die Symbole vorerst nicht zu attraktiv gestaltet werden, um eine duale Repräsentation zu ermöglichen. Ein wichtiger Gesichtspunkt ist die Verbindung zwischen der gelegten Sequenz und der Ausführung auf dem PC. Wir fassen das Spielen als Programmieren auf, insofern die Spieler das Gesamtproblem in Teilschritte zerlegen und geeignete Bewegungsarten und die Anzahl an Spielfeldern auswählen. Der daraus resultierende Ablauf entspricht einfachen Programmsequenzen in Computerprogrammen. Die vier Bewegungsarten einer Sequenz entsprechen Funktions- oder Methodenaufrufen, denen als Parameter die Reichweite der Bewegung übergeben wird. Um dieses Konzept für die Spieler zu visualisieren, geben die Sifteo Cubes ein

grafisches Feedback über den Fortschritt der Ausführung. Mit dem be-greifbaren Spielmaterial sollen kognitive Prozesse externalisiert werden und so das Problemlösen der Kinder unterstützt und kommunizierbar werden.

## 5 Co-Design-Prozess

Mit dem folgenden formativen Ansatz soll die Eignung eines neuen Spiels für 4-jährige Kinder untersucht und optimiert werden. Eine generelle Voraussetzung für ein erfolgreiches Spiel ist, dass die Sifteo Cubes mit den angezeigten Bildern als Symbole begriffen werden und eine duale Repräsentation zulassen.

Gegenwärtig werden vier Gegenstandsbereiche als zentral für einen Co-Design-Prozess betrachtet. Erstens müssen geeignete Symbole für die Spielsteine identifiziert werden. Dazu werden verschiedene Varianten entwickelt, die dieselbe Information enthalten, sich jedoch in ihrem Abstraktionsgrad unterscheiden. Verschiedene Symbole können auch kombiniert werden, um die ideale Darstellungsweise zu ermitteln. Dazu werden Kindern Entwürfe für Symbole zur Interpretation vorgelegt, um mögliche Fehlinterpretationen aufzudecken. In einem weiterführenden Schritt werden die verschiedenen Varianten der Symbole in einfache Levels des Spiels integriert und das Spiel von einem der Autoren gemeinsam mit je einem 4-jährigen Kind gespielt. Dies erlaubt einen qualitativen Beobachtungs- und Befragungsprozess, in dem Probleme aufgedeckt werden können. Zweitens sollen verschiedene Varianten getestet werden, mit denen die Symbole auf den Steinen ausgewählt werden können. Eine Variante besteht darin, Modifikationen der „Befehle“ der Programmierbausteine durch Drücken der Bildschirme oder durch eine Rotationsbewegung der Programmierbausteine im Raum hervorzurufen. Eine weitere Variante besteht in der Verfügbarkeit von zwei Modifikator-Steinen, wobei einer die Art der Bewegung (Hüpfen, Schwimmen, Klettern, Gehen) und einer die zurückgelegte Entfernung (ein, zwei oder drei Spielfelder) modifiziert. Dazu wird ein Modifikator-Stein neben einen Spielstein gelegt. Ein weiterer Aspekt ist die Ausgangsanzeige der Spielsteine. Der Stein kann zunächst „leer“ sein und nur einen Hintergrund ohne Symbole anzeigen. Diese Variante kann mit einer standardmäßigen Anzeige von einem „Sprung“ verglichen werden. Drittens soll die Landschaft, in der sich der Frosch bewegt, optimiert werden, da diese auch die Aufgabenstellung für die Spieler enthält. Hier ist es notwendig, dass die Anzahl an Spielfeldern fehlerfrei zu erkennen ist. Es ist aber auch wünschenswert, dass die Landschaft ansprechend gestaltet ist. Schließlich wird optimiert, wie eine aus drei Spielsteinen gelegte Sequenz ausgelöst wird. Eine Sequenz kann ausgelöst werden, indem die drei Spielsteine nebeneinander gelegt werden oder durch das Betätigen eines Start-Steines.

Sind diese grundlegenden Fragen des Spiel-Designs geklärt, kann das Spiel eingesetzt und weiterführende Fragen können geklärt werden. Beispielsweise kann dann untersucht werden, ob das Lernspiel gemeinsam gespielt werden kann und wenn, ob die Kinder sich gegenseitig durch Erklärungen beim Spielen helfen. Weiterhin kann die Rolle des Greifens untersucht werden, indem alle Elemente des Spiels auf einem Tablet PC präsentiert werden. Schließlich ist es entwicklungspsychologisch mit Fokus auf Programmierfähigkeiten auch interessant, ob

Kinder im Alter von drei Jahren schon in der Lage sind, das Spiel erfolgreich zu spielen. Dabei soll ebenfalls untersucht werden, ob mögliche Probleme für 3- und 4-Jährige gleichermaßen bestehen.

## 6 Zusammenfassung

Der vorliegende Beitrag skizziert die Entwicklung eines be-greifbaren Lernspiels für 4-Jährige. Ziel ist das spielerische Erlernen des Programmierens. Aufbauend auf drei Systemgenerationen wird das Spiel unter Berücksichtigung kognitiver Fähigkeiten in einem Co-Design-Prozess entwickelt. Wichtige Aspekte sind dabei die Gestaltung der Symbole und der Interaktion der Kinder mit den Steinen um die Symbole auszuwählen. Ist das Spielkonzept durch iterative Entwicklung hinreichend ausgereift, können weiterführende Fragestellungen bearbeitet werden. Eine spezielle Frage ist die Eignung des Spiels für 3-Jährige.

### Literaturverzeichnis

- Anderson, J. R. (2001). *Kognitive Psychologie. 3. Auflage*. Heidelberg: Spektrum.
- DeLoache, J. S. (2004). Becoming symbol-minded. *Trends in Cognitive Science*, 8(2), 66-70.
- Kösel, E. (1993-2007). *Die Modellierung von Lernwelten*. 3 Bände. SD Verlag, Bahlingen.
- Merrill, D., Kalanithi, J. & Maes, P. (2007). Siftables: towards sensor network user interfaces. In Ullmer, B. & Schmidt, A. (Hrsg.): *Proceedings of the 1st International Conference on Tangible and Embedded Interaction 2007*. New York: ACM, S. 75-78.
- Oerter, R. & Montada, L. (Hrsg.) (2002). *Entwicklungspsychologie. 5. Auflage*: Weinheim: Beltz.
- Scharf, F. (2007). *Tangicons: Algorithmic Reasoning in a Collaborative Game for Preschool Children*, Student Research Project, IMIS, University of Luebeck.
- Scharf, F., Winkler, T., Hahn, C., Wolters, C. & Herczeg, M. (2012). Tangicons 3.0 – An Educational Non-Competitive Collaborative Game. In Schelhowe, H. (Hrsg.): *The 11th International Conference on Interaction Design and Children, IDC '12*. Bremen: ACM, S. 144-151.
- Siegler, R., DeLoache, J. & Eisenberg, N. (2005). *Entwicklungspsychologie im Kindes- und Jugendalter. 1. Auflage*. München: Elsevier.
- Winkler, T., Scharf, F., Peters, J. & Herczeg M. (2011). Tangicons - Programmieren im Kindergarten. In Eibl, M. & Ritter, M. (Hrsg.): *Workshop Proceedings der Tagung Mensch & Computer 2011 - überMEDIEN ÜBERmorgen*. Universitätsverlag: Chemnitz, S. 23-24.