

# Kreativer Einstieg in die Programmierung

## Alice im Informatik-Anfangsunterricht

Michael Dohmen, Johannes Magenheim  
Didaktik der Informatik  
Universität Paderborn  
Fürstenallee 11  
33100 Paderborn  
dohmen@upb.de  
jsm@upb.de

Dieter Engbring  
Gesamtschule Paderborn-Elsen  
Am Schlengerbusch 27  
33106 Paderborn  
didier@upb.de

**Abstract:** In dem hier vorliegenden Artikel werden erste Ergebnisse einer Unterrichtsreihe mit Alice vorgestellt, die in der Jahrgangsstufe 9 an der Gesamtschule Paderborn-Elsen durchgeführt und evaluiert wurde. Dabei haben wir einen Motivationsschub beobachtet. Der Umgang mit Alice allein trägt jedoch nicht dazu bei, dass Schüler grundlegende Kenntnisse über Objektorientierte Programmierung erwerben. Der deklarative Zugang mit vorgefertigten Objekten erleichtert den Einstieg aber sehr.

## 1 Einleitung

Informatikunterricht sieht sich mit vielfältigen Anforderungen konfrontiert: Er soll für die Schüler<sup>1</sup> motivierend und spannend sein, zu Kreativität anregen, an ihre Alltagserfahrungen anknüpfen, wichtige informatische Konzepte vermitteln, Mädchen in besonderer Weise für informatische Inhalte interessieren, in dem er Genderaspekte berücksichtigt, zudem noch lehrplankonform sein und sich an informatischen Bildungsstandards orientieren... In diesem Spannungsfeld vielfältiger und z.T. divergierender Anforderungen fällt es nicht immer leicht, Unterrichtskonzepte zu entwerfen und zielgruppenbezogen umzusetzen, die diesen Ansprüchen gerecht werden. Die Fachgruppe DDI an der Universität Paderborn hat in diesem Kontext ein informatikdidaktisches Forschungsprojekt initiiert, das sich in besonderer Weise mit motivierenden Zugangsweisen zur Informatik in der Sekundarstufe I beschäftigt. Im Projekt MIAU (Modellieren im Informatik-Anfangsunterricht) sollen Formen des explorativen Lernens erprobt werden, die einerseits an das Konzept des Informatik Lernlabors für die Sekundarstufe II anknüpfen [Ma08] und andererseits Schülerinnen und Schülern der Sekundarstufe I kreative und motivierende Zugangsweisen zum Erkunden und Gestalten von Informatiksystemen ermöglichen. Im Mittelpunkt steht dabei das Erlernen von objektorientierten Sichtweisen und Modellierungsmethoden, das u.a. mit visuellen Entwicklungsumgebungen und Sprachkonzepten wie Scratch und Alice gefördert werden soll.

---

<sup>1</sup>Im weiteren Text wird die männliche Form Schüler gleichwertig für Schülerinnen und Schüler verwendet.

## 2 Warum Alice im Anfangsunterricht?

Die Open Source 3D-Programmierungsumgebung Alice [Ba07] ermöglicht es Schülern, erste Erfahrungen mit objektorientiertem Modellieren und Programmieren zu sammeln. Mit dem kreativen Gestalten von grafischen 3D-Objekten, von Animationen und interaktiven Spielen, die auch leicht im Web publiziert werden können, ist ein kreativer und spielerischer Zugang zum informatischen Modellieren möglich. Damit wird einer wesentlichen Forderung neuerer didaktischer Ansätze zur informatischen Bildung Rechnung getragen, die Kreativität [Ro08] und kooperativem Lernen mit digitalen Medien [MHR08] einen hohen Stellenwert beimessen. Ferner wird von Protagonisten von Alice, wie Randy Pausch, die Möglichkeit des motivierenden indirekten Lernens informatischer Inhalte mittels eines „Head-Fake“-Effekts postuliert: „Der beste Weg jemandem etwas beizubringen, ist, ihn glauben zu lassen, dass er etwas anderes lernt. Das habe ich meine ganze Laufbahn über getan. Der „Head-Fake“ ist bei dieser Sache, dass sie lernen, zu programmieren, aber sie glauben, sie würden Filme und Videospiele machen“ [Pa07, S. 16]

Alice wird an der Carnegie Mellon University in Pittsburgh in Computer Science Kursen zur Einführung von Studierenden in die Informatik genutzt. Das System ist durch Komplexitätsreduktion der virtuellen Szenarien und der verwendeten informatischen Methoden allerdings didaktisch gut skalierbar und kann deshalb auch im Anfangsunterricht Informatik sogar in der Sek. I eingesetzt werden. Das Erlernen fundamentaler Konzepte des Programmierens ist mit Alice daher wahrscheinlich einfacher erlernbar. Durch das deklarative Manipulieren von Objekteigenschaften und objektbezogenen Verhaltensweisen (Methoden) werden möglicherweise objektorientierte Sichtweisen gefördert. Bei geeigneter Fortführung dieses Konzepts, bis hin zur Verwendung klassischer objektorientierter Programmiersprachen, wie z.B. Java, könnten sich später in Verbindung mit objektorientierten Modellierungsmethoden, wie Objektspiel und CRC-Karten Einsatz, im Informatikunterricht didaktisch-methodische Prinzipien des „Objects-first“ Ansatzes realisieren lassen [CDP03][Di07].

Für einen leichteren Einstieg in Alice können Schüler auf bereits vorhandene Objekte in einer „Galerie“ der virtuellen 3D-Welt zurückgreifen, deren Verhalten sie dann mittels Programmierung in einem virtuellen Szenario bestimmen können. Die interaktive Entwicklungsumgebung von Alice ermöglicht ein einfaches Drag-and Drop Positionieren von Objekten im virtuellen 3D-Szenario. Die Zuweisung von Attributwerten und das Programmieren von Verhaltensweisen der Objekte erfolgt ebenfalls in einer visuell strukturierten, einfach handhabbaren Schnittstelle.

Alice bietet ähnlich wie Scratch, das eine 2-D Programmierungsumgebung für die Manipulation grafischer Sprites zur Verfügung stellt, und wie andere visuelle Programmierungsumgebungen, den Vorteil, dass die Schüler nicht nur eine unmittelbare Rückmeldung über eventuelle syntaktische Fehler erhalten, sondern auch sofort anhand des Verhaltens ihrer grafischen Objekte erkennen können, ob auf „semantischer Ebene“ gewünschtes und tatsächliches Objektverhalten in der virtuellen Welt übereinstimmt [Ba08]. Auf diese Weise können notwendige Programmmodifikationen sofort erkannt und zielgerichtet vorgenommen werden. Die primär auf die Gestaltung von Animationen und interaktiven Spielen ausgerichtete Konzeption von Alice erfordert für Einsteiger neben objektorien-

tiertem Denken zunächst eine Fokussierung auf die Gestaltung von virtuellen 3D-Welten. Dazu sind Kompetenzen, wie räumliches Vorstellungsvermögen oder die kreative Entwicklung von anschaulichen Spielszenarien (Stories) mit Kamerapositionierung und -führung erforderlich. In fortgeschrittenen interaktiven Animationsszenarien werden über die rein sequentielle Programmierung hinaus Verständnis für Nebenläufigkeit, Ereignisorientierung, Klasse-Objektbeziehungen sowie Vererbung gefördert.

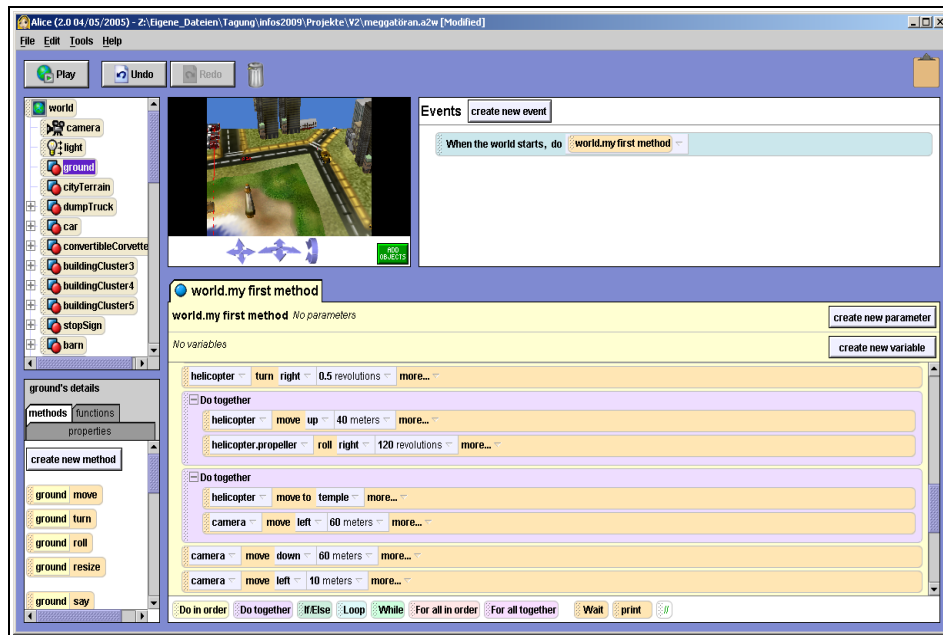


Abb. 1: Screenshot eines Schülerprojekts mit Objektfenster, 3D-Sicht, Event-, Methoden- und Programmierfenster

Um dem Bedürfnis von Mädchen nach einem motivierenden, spielerischen Zugang zur Informatik entgegen zukommen, der nicht zu stark programmiertechnisch ausgerichtet ist, wurde speziell für diese Zielgruppe die Programmierumgebung „Story Telling Alice“ entwickelt. In Erweiterung des klassischen Alice-Konzepts wird in dieser Umgebung eine Komponente bereitgestellt, die es ermöglicht, Handlungsszenarien in Form von gezeichneten Geschichten vorzuplanen, bevor sie im System mit geeigneten Objekten und Spielszenen aus einer Galerie umgesetzt werden [Ke06]. Hier ergeben sich wiederum methodische Querverbindungen zum „Objects first“ Ansatz mit seinen Story Boards [Di07].

Abschließend kann zur Beantwortung der in der Überschrift dieses Abschnitts gestellten Frage auch noch darauf verwiesen werden, dass die mit Alice erlernbaren informatischen Konzepte im Inhaltsbereich von Informatiklehrplänen verschiedener Bundesländer und in den GI „Bildungsstandards Informatik in der Sek. I“ besonders im Inhaltsbereich „Algorithmen“ sowie in den Prozessbereichen „Kommunizieren und Kooperieren“ und „Modellieren und Implementieren“ wiederzufinden sind [GI08].

### 3 Unterrichtsversuch und Beobachtungsergebnisse

#### Das Setting

Für den Unterrichtsversuch mit Alice musste ein Kurs gefunden werden, für den Programmieren Neuland war und mit dem der Umgang mit Alice im Anfangsunterricht stattfinden konnte. Dazu wurde die Gesamtschule Paderborn-Elsen ausgesucht, die Informatik in der Sek. I im Wahlpflichtbereich II anbietet. Im hausinternen Curriculum – den sogenannten Partituren – für den Wahlpflichtbereich II Jahrgang 9 und 10 im Fach Informatik gab es für einen solchen Unterrichtsversuch Freiräume. Wesentliches Ziel dieses Kurses ist es in wichtige Anwendungen der Informatik einzuführen. Im Jahrgang 10 ist darüber hinaus aber immer wieder die Gelegenheit genutzt worden, auch Grundlagen der Programmierung zu unterrichten. Es bot sich an, diese Unterrichtsreihe vorzuziehen, da sich der Kurs im Umgang mit der Tabellenkalkulation, vorbereitet durch die Mathematik-Kollegen des Jahrgangs, durchaus geschickt anstellte und die gestellten Aufgaben, einschließlich der Fallunterscheidung, bearbeiten konnte. In jedem Jahrgang konnten während der letzten Jahre zwei Kurse angeboten werden, so dass Informatik geschlechterdifferenziert unterrichtet werden konnte. Bei dem Kurs, bei dem wir Alice eingesetzt haben, handelt es sich um den Kurs mit den männlichen Schülern<sup>2</sup>. Während der Durchführung des Projektes haben wir uns dazu entschieden, zu zweit im Klassenraum anwesend zu sein, um das Unterrichtsgeschehen möglichst genau beobachten zu können. Dadurch konnten wir auch intensiver auf Schülerfragen eingehen. Es zeigte sich jedoch, dass auch ein Lehrer allein mit der Beantwortung der Fragen nicht überfordert gewesen wäre.

In dem aktuellen Kurs befanden sich zum Projektstart 19 und nach dem Halbjahreswechsel 20 Schüler. Diese Schüler, die an an der Gesamtschule Paderborn-Elsen Informatik wählten, gehören leider nicht zu den leistungsfähigsten ihres Jahrgangs, so dass wir insbesondere bei den in Englisch verfassten Tutorials Probleme erwartet haben.

Für die Unterrichtsreihe haben wir unsere Planung darauf ausgerichtet, dass die Schüler möglichst selbstständig arbeiten und dass sie selbst bestimmen, welche Produkte hergestellt werden. Zu enge Vorgaben wären hinderlich gewesen. Außerdem konnte damit ein methodischer Kontrast zu den vorangegangenen Unterrichtsreihen zur Textverarbeitung und Tabellenkalkulation geschaffen werden. Dies könnte bei den Schülern möglicherweise zu einer höheren Motivation geführt haben, sich mit Alice zu befassen. Außerdem wollten wir untersuchen, ob der von Randy Pausch beabsichtigte „Head-Fake“ tatsächlich Wirkung zeigt.

Die Reihe war auf fünf Doppelstunden ausgerichtet. Die Schüler haben sich nach der Durchführung der Eingangsbefragung mit den mitgelieferten Tutorials beschäftigt. Danach hatten die die Gelegenheit Fragen zu stellen, die von den anwesenden Lehrern bzw. auch von Mitschülern beantwortet wurden. Eigentlich hatten wir an dieser Stelle größere Probleme mit der englischen Sprache erwartet, diese taten aber nur am Rande in Erscheinung. So wurden gelegentlich bei den anwesenden Lehrern Vokabeln nach-

---

<sup>2</sup> Zeitgleich Alice auch im „Mädchenkurs“ einzusetzen, war organisatorisch leider nicht möglich.

gefragt. Danach erhielten die Schüler die Aufgabe, sich ein Szenario zu überlegen, in dem sie verschiedene Objekte – analog dem Keyboard in einem Film – miteinander zu einer kleinen Geschichte verknüpfen sollten.

Diese Ideen wurden zu Beginn der zweiten Doppelstunde noch einmal intern und mit dem hauptverantwortlichen Lehrer des Kurses diskutiert. Dabei wurden Anregungen der Schüler und des Kurslehrers aufgenommen. Dann gingen die Schüler an die Umsetzung der Ideen und in die „Produktion“. Mitte der dritten Doppelstunde haben wir eine Zwischenpräsentation der bis dahin erzielten Arbeitsergebnisse eingeschoben, bei der Schüler vorstellen sollten, wo sie sich Tipps geholt haben, was nicht funktioniert hat und was sie gerne umsetzen würden, bisher aber nicht konnten. In einer offenen Diskussion kamen aus den anderen Gruppen konstruktive Anregungen und Anmerkungen. Jede Gruppe sollte für die restliche Zeit Ziele abstecken, die dann von den Mitschülern kommentiert wurden. Die vierte Doppelstunde war, ebenso wie die erste Hälfte der fünften Doppelstunde, erneut der Produktion gewidmet. Der zweite Teil der letzten Doppelstunde diente der Abschlusspräsentation und einer Abschlussbefragung.

Die schwierigste Entscheidung war es, die Anzahl der Gruppen festzulegen. Diese wurde stark beeinflusst durch die vorhandene Hardware. Die Gesamtschule Paderborn-Elsen ist, wie alle allgemeinbildenden Paderborner Schulen, Teil des sogenannten Lernstätt-Netzes. Dies ist eine zentral administrierte Infrastruktur auf der Basis von Sun-Servern und „thin“ Sun-Ray-Clients an den Schulen. Die Gesamtschule Paderborn-Elsen verfügt derzeit über 80 solcher Arbeitsplätze in drei Computerräumen: im Selbstlernzentrum der Oberstufe, als Medienecken in den Jahrgangsstufen 5 bis 7 und in den fünf Lehrerstützpunkten. Im Computerraum, in dem der Kurs stattfand, befinden sich 17 davon. Leider ist die Alice-Umgebung nur auf Windows-Rechnern lauffähig und eben nicht unter Unix/Solaris. So haben wir zunächst den Versuch gestartet, Alice im Gruppenverzeichnis des Kurses abzulegen und dann über den Zugriff auf den Windows-Terminalserver, der von jeder Sun-Ray gestartet werden kann, zu benutzen. Dies erwies sich aber als wenig performant. Da wir an der Schule nur dauerhaften Zugriff auf drei Laptops erhalten konnten und im Computerraum nur ein Multimedia-PC zur Verfügung steht, entschieden wir uns für vier Gruppen, die wir in den ersten beiden Sitzungen allerdings an den Sun-Ray-Stations arbeiten ließen, da die Laptops an diesen Tagen anderweitig im Einsatz waren.

Die durch diese Einschränkungen entstandenen Gruppen waren leider zu groß. Es wäre besser gewesen, die Gruppengröße auf jeweils drei Schüler zu beschränken. Diese Entscheidung hat naturgemäß Einfluss auf die Ergebnisse der Unterrichtseinheit. Sie sollen im folgenden Abschnitt zunächst auf der Basis von Beobachtungen dargestellt werden, um sie im dann später mittels empirisch gesicherter Daten ein wenig von ihrer Subjektivität zu befreien.

### **Befunde und Beobachtungen**

Die Zwischen- und Abschlusspräsentation der Schüler am Ende der Unterrichtsreihe und die Analyse der fertigen Projekte zeigen viele interessante Aspekte und Probleme der Schüler im Umgang mit Alice auf. Hier die wichtigsten Ergebnisse:

*Gruppe 1* erstellte ein sehr aufwändiges Szenario in einer Stadt, bei der mit Hilfe von Polizeiwagen und Helikoptern eine Verfolgungsjagd und eine Geiselnahme simuliert wurden. Insgesamt haben die Schüler mehrere Minuten „Spielfilm“ realisiert. Da sie das gesamte Geschehen in einer Methode „World.myFirstMethod“ implementiert haben, war es nicht möglich, einzelne Teile des Szenarios getrennt zu testen, so dass das beschleunigte Abspielen eine besondere Bedeutung für das Testen der Szene bekam. Die wichtigsten Probleme der Gruppe waren die korrekte Ausrichtung der Objekte, damit diese bei der Bewegung nicht im Boden verschwanden und die Realisierung einer dauernden Nebenläufigkeit, um die Rotoren der Helikopter zu drehen.

Das von dieser Gruppe und hauptsächlich vom Schüler A (siehe unten) realisierte Projekt ist zwar wenig strukturiert, zeigt aber eine spontane Umsetzung von Ideen und erforderte viele Stunden Programmierarbeit, die fast vollständig zu Hause ohne explizite Hausaufgabe erbracht wurde.

*Gruppe 2* realisierte einen Skaterpark, in dem zwei Hasen auf Skateboards eine Rampe hochfahren sollten. Dieses recht einfache Szenario erwies sich dann als wesentlich diffiziler als ursprünglich gedacht. Zunächst besaßen die beiden verschiedenen Hasen-Objekte einen unterschiedlichen Koordinatenursprung und damit Körpermittelpunkt, so dass ein Hase beim Sprung auf das Skateboard im Boden verschwand. Dieses Problem konnte nur durch Lehrerintervention gelöst werden, denn hier liegt offenbar mangelnde Transparenz in wichtigen Attributen der vorgefertigten Objekten vor. Die gemeinsame Bewegung von Hase und Skateboard erforderte eine Ausrichtung der beiden Objekte, um eine gemeinsame Bewegungsrichtung zu haben. Da die Schüler dies versäumten, bewegten sich die Objekte in unterschiedliche Richtungen. Die Bewegung des Skateboards in der Halfpipe erforderte eine Zerlegung der Bewegung in viele kleine Abschnitte, da es sich hier um eine nichtlineare Bewegung handelte. Somit verwendete diese Gruppe Schleifenstrukturen mit jeweils 40 Wiederholungen, die aber insgesamt zu langsam für eine realistische Fahrt in der Halfpipe waren.

*Gruppe 3* nutzte ein fertiges City-Terrain und platzierte dort die Gebäude einer Stadt. Zunächst hatten die Schüler Probleme, das Terrain ebenerdig anzulegen. Da die Straßen nur graphische Repräsentationen und keine einzelnen Objekte sind, konnten sie die Orientierung der Fahrzeuge an der Straße nur durch Probieren realisieren. Zwar ist ein fertiger Untergrund sicher praktisch zur schnellen Realisierung des Projekts, jedoch wird dadurch die objektorientierte Sicht auf Untergrundobjekte nicht umgesetzt. Ein großes Problem der Schüler war der zu nahe Hintergrund, der einen Großteil des Cityterrains verschwinden ließ und dadurch viele Kamerafahrten notwendig machte. Der Hintergrund ist notwendig, um die Zahl der Details zu beschränken, jedoch reduziert er die Darstellungsmöglichkeiten komplexer Szenarien.

*Gruppe 4* setzte eine Szene auf dem Meer mit mehreren Booten und einem Flugzeugträger um. Dabei gab es ebenfalls Probleme mit dem zu nahen Hintergrund und der Steuerung der Kamera. Es sollte eine geradlinige Bootsbewegung dargestellt werden, die auf den ersten Blick zunächst auch geradlinig aussah. Leider zeigte sich beim Drehen der Kamera, dass die Bewegung nicht korrekt war. Auch hier würde man sich den gleichzeitigen Blick aus mehreren Perspektiven beim Erstellen der Landschaft wünschen, um die Objekte präziser positionieren zu können.

Ein wesentliches Problem aller Gruppen war es, dass Bewegungen „eckig“ wirkten. Hierfür konnte keine der Gruppen eine Lösungsstrategie entwickeln. Nachdem wir mit einem entsprechenden Tipp (kürzere Treppenzugbewegungen, die dann im Ablauf rund aussehen) nachgeholfen haben, wurde die Idee in verschiedenen Szenarien umgesetzt. Ein weiteres grundsätzliches Problem bestand darin, dass die tatsächliche Ausrichtung der Objekte nicht genügend nachvollziehbar war. Je nach Perspektive der Kamera, sahen Objekte gut oder ganz schlecht ausgerichtet aus. Insbesondere der Flugzeugträger sah mal funktionstüchtig, mal wie die sinkende Titanic aus. Hier bietet die Alice-Umgebung wenig Hilfe, um eine exakte Positionierung vorzunehmen. Autos oder andere Gefährte, die auf eine Straße gesetzt wurden, waren nur schwer so auszurichten, dass sie langfristig auf der Straße fuhren. Auch die Attribute der Objekte – etwa ein unterschiedlicher Koordinatenursprung verschiedener Objekte – waren für die Schüler und auch für die Lehrer nur nach längerem Suchen erklärbar.



Abb 2: Szenarios der 4 Gruppen

Allen erstellten Projekten ist gemein, dass Sie einen linearen, filmähnlichen Ablauf des Geschehens ohne Benutzerinteraktion darstellen. Der Programmcode enthielt jedoch keine Strukturierung oder Kommentierung. Sicher wäre hierzu noch zusätzlicher Input zur Strukturierung mit Hilfe von Methoden und Eventhandling durch den Lehrer notwendig gewesen. Darauf haben wir zugunsten der Förderung des selbständigen Arbeitens der Schüler bewusst verzichtet.

Aus den 20 Schülern des Kurses haben wir vier Schüler herausgegriffen, die uns besonders aufgefallen sind. Wir nennen diese Schüler A, B, C und D.

Der *Schüler A* war bisher zwar interessiert, aber nicht unmittelbar motiviert, sich mit den bisherigen Themen des Unterrichts zu befassen. Dies änderte sich nach der ersten Stunde mit Alice. Er war der erste, der sich Alice auf seinem heimischen PC installierte. Er arbeitete zu Hause mehrere Stunden daran und brachte seine Ergebnisse mit in den Unterricht, nicht nur, um sie seiner Gruppe zu zeigen, sondern damit alle daran weiterarbeiten konnten.<sup>3</sup> Dabei nahm sich A durchaus zurück und ließ die anderen machen.

Der *Schüler B*, ein eher, auch in Bezug auf Störungen, auffälliger Schüler, nahm „das Heft“ in seiner Gruppe in die Hand. Ihm fehlte, wie auch den anderen aus der Gruppe, zum einen die Ausdauer, Ideen auch gegen Widerstände umzusetzen. Ein Grund dafür war das oben beschriebene Hasenproblem. Bei diesem Schüler konnten wir feststellen, wie schnell anfängliche Motivation schwinden kann. Obschon dieser Schüler mit einer Strukturierung in Abschnitte gearbeitet hat, ist hier das simpelste Produkt entstanden.

Der *Schüler C* hatte sich bei der zuvor durchgeführten Unterrichtsreihe zur Tabellenkalkulation sehr in den Vordergrund gearbeitet. Er war der Schnellste bei den Problemlösungen und konnte auch die schwierigen Aufgaben ohne Schwierigkeiten lösen. Auch beim Einstieg in Alice zeigte er sich sehr motiviert. Ein Großteil dieser Motivation war aber schnell verfliegen, da er die „Umgebung“ als zu langsam, als zu kompliziert zu bedienen und als zu instabil empfand. Dennoch war es C, der die ganze Zeit Maus und Tastatur unter Kontrolle hielt und sich dort mit den Ideen der Mitschüler auseinandersetzte. Hier war eine hierarchische Form der Teamarbeit zu beobachten. Gerade bei den Mitgliedern dieser Gruppe wäre es interessant gewesen zu beobachten, welche Ideen sie unter anderen Voraussetzungen umgesetzt hätten.

Der *Schüler D* kam erst zum Halbjahreswechsel (dritte Doppelstunde) in den Kurs. Er wurde von uns in die 4-er Gruppe (mit B) eingewiesen und fragte sehr bald, ob er nicht die Gruppe wechseln dürfe. Nach einer Ablehnung probierte er Alice alleine aus. Zur nächsten Stunde brachte er ein eigenes Projekt mit, welches einen Ritter und Hexenkampf im Mittelalter als Spielfilmszene simuliert. Dabei benutzt er eine Vielzahl von Objekten, um die Umgebung möglichst realistisch zu gestalten. Er entwickelte dieses komplexe Projekt in den nächsten Stunden weiter und besprach es mit einem Teil der Gruppenmitglieder.

## 4 Empirische Ergebnisse

Zur Unterstützung und zur Objektivierung der Beobachtungen durch die beiden anwesenden Lehrer wurde zusätzlich vor und nach der Unterrichtsreihe eine Befragung der Schüler durchgeführt.<sup>4</sup>

Die *Eingangsbefragung* sollte die Verfügbarkeit und Einsatzgebiete des Computers im Umfeld der Schüler, die Vorstellungen von Informatik, Gründe für die Wahl des Kurses und Erwartungen an das Fach sowie die Sicherheit im Umgang mit Computern und Computeranwendungen und damit wichtige Lernvoraussetzungen untersuchen.

---

<sup>3</sup>Angesichts des Ganztagsbetriebes an der Gesamtschule Paderborn-Elsen ist das durchaus bemerkenswert.

<sup>4</sup>Die Fragebögen und der Wissenstest sind unter <http://ddi.upb.de/didaktik/material/alice> zugänglich.



Jeder der 19 befragten männlichen Schüler hat Zugang zu einem Computer, 2/3 der Befragten besitzen einen eigenen Computer, die Übrigen können einen gemeinsamen Computer nutzen. Dabei ist die durchschnittliche Nutzungszeit von 3 Stunden pro Tag erschreckend hoch, zumal man berücksichtigen muss, dass alle häuslichen Computer einen Internetzugang besitzen. Die Befragten geben an, dass sie von 14 ausgewählten Standardanwendungen des Internets etwa 8 Programme regelmäßig nutzen. Dabei stehen neben Surfen, E-Mail, SchülerVZ und Chat auch Onlinespiele hoch im Kurs. Aus diesen Angaben erklärt sich auch die tägliche Nutzungszeit. Insgesamt kommen die Schüler aus wenig computeraffinen Haushalten, nutzen das Gerät aber trotzdem für private Zwecke recht ausgiebig.

Für den Erfolg der Unterrichtsreihe sind u.a. die motivationalen Voraussetzungen und die Erwartungen an das Fach bzw. die Unterrichtseinheit ausschlaggebend. Die Ergebnisse in Abb. 3 zeigen, dass das allgemeine Interesse am Computer und vermutete berufliche Vorteile für die Schüler wichtiger sind als das Interesse am Programmieren. Gerade bei dieser Untersuchungsgruppe, die eigentlich nicht das Programmieren erlernen möchte, sind die Ergebnisse des Nachtests besonders interessant.

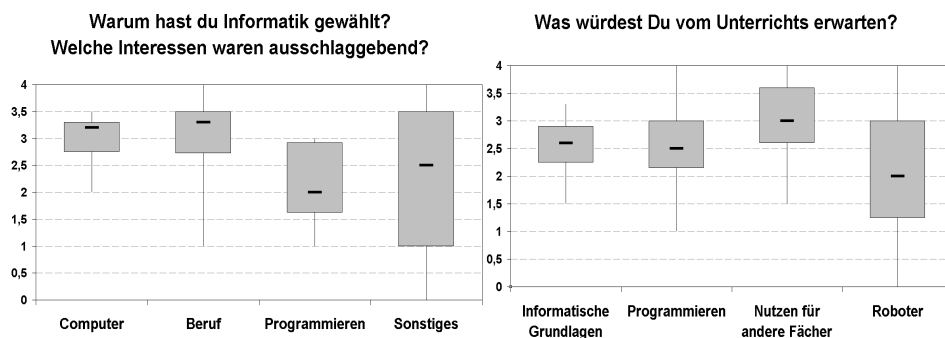


Abb. 3: Gründe für die Wahl des Faches Informatik und Erwartungen an den Unterricht auf einer Skala von 0 (stimme nicht zu) bis 4 (stimme zu)<sup>5</sup>

In einer weiteren Frage wurden die Erwartungen der Schüler an den Unterricht untersucht. Hier zeigen sich ähnliche Ergebnisse; die Schüler erwarten vom Informatikunterricht die Vermittlung von informatischen Grundlagen, die dann hauptsächlich Nutzen für andere Schulfächer bringen. Auch das Thema Roboterprogrammierung, welches häufig zur Steigerung der Motivation im Fach eingesetzt wird, wurde von den Schüler nicht als wichtig erachtet.

In einem letzten Schritt wurde die Sicherheit im Umgang mit Computern untersucht. Dieser schon 10 Jahre alte und validierte Test zeigt, dass die Untersuchungsgruppe keine Unsicherheiten bei der Bedienung des Rechners haben. Insgesamt haben wir es mit computererfahrenen, mit vielen Anwendungen und dem Internet vertrauten Probanden zu tun.

<sup>5</sup>Die dargestellten Boxplots geben neben dem Median (schwarze Markierung) auch Angaben über die Quartile und Extrema der Verteilung an. Somit enthalten sie im Unterschied zum Mittelwert zusätzliche Informationen zur Verteilung.

Der *Nachtest* am Ende der Unterrichtsreihe sollte zum Einen die subjektive Qualität der Unterrichtsreihe aus Schülersicht untersuchen, zum Anderen auch das erworbene Wissen der Teilnehmer zum Programm Alice und zur Objektorientierung festhalten. Zur Bewertung sollten die Schüler sowohl den Lernerfolg der Unterrichtsreihe für den gesamten Kurs als auch ihren individuellen Lernerfolg mit Fragen zu methodischen und medialen Hilfen und zu möglichen Problemen.

Hier die wesentlichen Ergebnisse: Der Lernerfolg des Kurses wurde von den Versuchsteilnehmern recht differenziert gesehen. So überwog die Einschätzungen, dass eine große Leistungsdiskrepanz zwischen den Leistungsstarken und Leistungsschwachen vorlag, die auch während der Beobachtung der Projektphase durch die Lehrer bestätigt werden konnte. Ein Drittel der Schüler war der Ansicht, dass alle Schüler die Inhalte gut bis sehr gut verstanden haben. Diese Einschätzung wurde zumeist von leistungsstärkeren Schülern vertreten.

Der eigene Lernerfolg wurde sogar noch besser als der Kurserfolg und zwar als hoch oder eher hoch eingeschätzt. Dabei waren aus Sicht der Schüler die Hauptlernziele „Programmieren“ (12 Nennungen), „nicht sofort aufzugeben, wenn ich bei Problemen nicht weiterkommen“ (9 Nennungen) und „Probleme zu besprechen und schließlich zu lösen“ (8 Nennungen). Methodisch haben dabei die Arbeit am Projekt (12 Nennungen) und die Erklärungen des Lehrers im Unterricht (10 Nennungen) geholfen. Aus diesen Ergebnissen kann man schließen, dass die Schüler zwar erkannt haben, dass sie einen Programmierkurs durchlaufen haben, jedoch auch der Erwerb sozialer Kompetenzen wie Teamfähigkeit durch Gruppenarbeit einen ähnlich hohen Stellenwert besaßen. Eine differenzierte Einschätzung der Unterrichtsreihe nach verschiedenen Aspekten in Abb. 4 zeichnet ein ähnliches Bild.

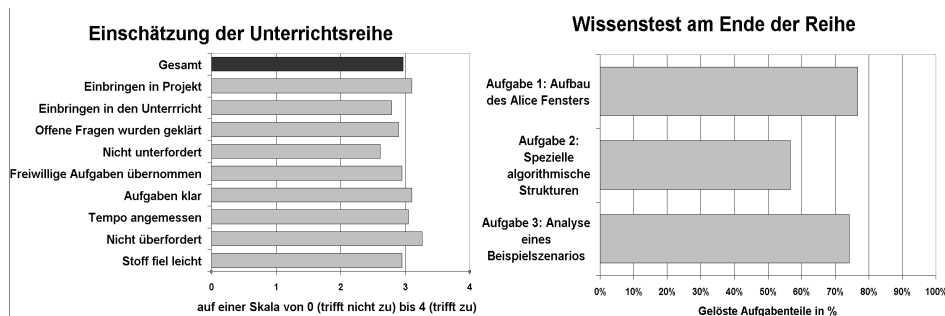


Abb. 4 Einschätzung verschiedener Aspekte der Unterrichtsreihe durch die Teilnehmer und Ergebnisse des Wissenstests

Der abschließende *Wissenstest* bestand aus Fragen zum Aufbau des Alice-Fensters zur Überprüfung von Bedienfertigkeiten, einer Aufgabe zu algorithmischen Grundstrukturen, wie Wiederholung und Nebenläufigkeit und einer kleineren Modellierungsaufgabe für ein vorgegebenes Beispielszenarios, um objektorientierte Kompetenzen zu messen.

Die Ergebnisse dieses Tests zeigen eine große Schwankungsbreite, da einige Schüler, etwa die Aufgabe 2 zu algorithmischen Grundstrukturen, aus zeitlichen Gründen nicht ausgefüllt hatten. Die Benutzerschnittstelle von Alice wurde von fast allen Schülern in

wesentlichen Teilen korrekt beschrieben, Probleme gab es nur beim Fenster für das Event-Handling, da sie dies aufgrund ihrer linearen Projekte nicht benötigten und deshalb die Funktion nicht kannten. Bei der Aufgabe 2 wurden im Schnitt 3 von 5 Strukturen richtig zugeordnet, hierbei handelte es sich um die bei den Projekten fast ausschließlich benutzten „DO IN ORDER“, „DO TOGETHER“ und „WAIT“. In der Aufgabe 3 sollten die Schüler für ein Beispielszenario beschreiben, welche Objekte agieren und welche Methoden in welcher Reihenfolge aufgerufen werden müssen. Hier wird das Basiswissen zur Objektorientierung abgefragt. Dieser Aufgabe wurde insgesamt am Besten bearbeitet. Die Lösung von drei Schülern war sehr detailliert mit einer genauen Befehlsfolge dargestellt. Alle Schüler fanden auch die passenden Objekte und die verwendeten Methoden, konnten aber den genauen Ablauf und die bei der Realisierung in Alice auftretenden Probleme nur oberflächlich beschreiben. Insgesamt zeigt der Wissenstest im Rahmen der kurzen Unterrichtsreihe bei den Schülern fundierte Kenntnisse von Alice, die von der Mehrzahl der Schüler in einem konkreten Projekt mit einer objektorientierten Sichtweise, d.h. Identifikation von Objekten über Attribute und deren Methoden angewandt werden konnten.

## 5 Fazit und Perspektive

Was haben die Schüler also in der Unterrichtsreihe gelernt? Die Schüler waren sehr kreativ und haben umfangreiche Szenarien trotz der Kürze der Zeit erstellt. Dabei gab es motivationale Unterschiede, der von Randy Pausch angesprochene „Head-Fake“ hat nicht für die gesamte Lerngruppe funktioniert. Vielleicht lag dies teilweise auch an der Gruppengröße von jeweils 5 Schülern, die aufgrund der Leistungsheterogenität nicht gut kooperiert haben. Somit ist auf geeignete Gruppengrößen, ausreichende Betreuung und Begleitung durch den Lehrer zu achten, um soziale Prozesse wie Teamfähigkeit zu fördern.

Welche Probleme traten mit der Software auf? Im Unterschied zu 2D-Umgebungen wie Scratch hat man in der 3D-Welt eine Vielzahl von Freiheitsgraden nicht nur bei der Positionierung und Steuerung der Objekte, die noch zusätzlich durch die Tastatur und Maus erfolgen kann, sondern auch die Kameraperspektive spielt eine entscheidende Rolle. Diese Freiheitsgrade müssen die Schüler zunächst beherrschen lernen, bevor sie ein eigenes Szenario entwickeln und programmieren können. Die interaktive Steuerung von Objekten wird anfänglich nicht umgesetzt. Auch die Parallelität mehrerer Aktionen ist durch den sequentiellen Ablauf der Programme und ohne den Input des Lehrers nicht genutzt worden. Somit ist der unterrichtliche Overhead beim Einsatz von Alice zunächst etwas größer. Er wird jedoch durch die gesteigerte Motivation der Schüler aufgewogen, bedingt durch die Nähe von Alice-Szenen zu Videospielen und zur eigenen Erfahrung.

Aber haben denn die Schüler Objektorientierung erlernt? Die Schüler haben etwas über Objektorientierung und die objektorientierte Sichtweise gelernt, das zeigt auch das Ergebnis des Wissenstests. Aber sie haben nicht objektorientiert modelliert, da die Objekte durch ein „Drehbuch“ gesteuert werden. Es handelt sich also um einen Ansatz zu „Objects first“ mit Rückgriff auf vorgefertigte Modelle, ohne dass der zweite Schritt, nämlich die Einführung von Klassen erfolgt. Die Schüler arbeiten dabei deklarativ.

Wie geeignet ist das didaktische Material? Das Tutorial bereitete den Schülern auch in der Jahrgangsstufe 9 trotz der englischen Sprache kaum Probleme. Die Umgebung ist so motivierend, so dass viele Details des Tutorials von den Schülern nicht wahrgenommen, sondern schnell eigene Projekte entwickelt wurden.

Und wie weit sind diese Ergebnisse verallgemeinerbar? Es handelt sich bei der Untersuchungsgruppe um recht junge und unerfahrene Schüler. Der Einsatz in einer höheren Jahrgangsstufe und für einen längeren Zeitraum könnte vermutlich den „Head-Fake“ auslösen. Für den Einsatz in Mädchengruppen ist Storytelling-Alice mit speziellen Methoden zur sozialen Kommunikation vielleicht eine sinnvolle Alternative. Bei den beiden letzten Punkten besteht jedoch noch Untersuchungsbedarf.

Alice ist aus unserer Sicht für den Einstiegsunterricht im Fach Informatik sehr geeignet, um objektorientierte Konzepte kreativ zu vermitteln. Ein Wechsel zu einer „klassischen“ Programmiersprache ist aufgrund von Vorgaben für zentrale Abschlussprüfungen allerdings unvermeidlich. Alice erhebt auch nicht den Anspruch, ein Allheilmittel zur Motivationssteigerung zu sein und wird die klassischen Modellierungsansätze über UML bis zur textuellen Programmierung im Anfangsunterricht nicht verdrängen können.

## Literaturverzeichnis

- [Ba07] Baldwin R.G., Learn to Program using Alice, 2007 URL: <http://www.dickbaldwin.com/tocalice.htm> [l.v. 16.02.09]
- [Ba08] Baldwin R.G., The New Face of Computer Science Education – The Scratch Generation, 2008 URL: <http://www.dickbaldwin.com/homeschool/Hs10000.htm> [l.v. 16.02.09]
- [CDP03] Cooper S., Dann W., Pausch R.: Teaching Objects-first In Introductory Computer Science, SIGCSE 2003
- [Di07] Diethelm, I., „Strictly models and objects first“ – Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht, Diss. Uni Kassel 2007 URL: <http://deposit.ddb.de/cgi-bin/dokserv?idn=98668760x> [[l.v. 16.02.09]
- [GI08] Gesellschaft für Informatik (Hrsg.), Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I, Empfehlungen der GI, erarbeitet vom Arbeitskreis „Bildungsstandards“ URL: <http://gi.informatikstandards.de/>
- [Ke06] Kelleher, C. Motivating Programming: Using storytelling to make computer programming attractive to middle school girls. PhD Dissertation, Carnegie Mellon University, School of Computer Science Technical Report CMU-CS-06-171 URL:[http://www.c-s.cmu.edu/~caitlin/kelleherThesis\\_CSD.pdf](http://www.c-s.cmu.edu/~caitlin/kelleherThesis_CSD.pdf) [l.v. 18.02.09]
- [Ma08] Magenheimer, J., Systemorientierte Didaktik der Informatik Sozio-technische Informatiksysteme als Unterrichtsgegenstand? In Kortenkamp; U.; Weigand; H.G.; Weth, T. (Hrsg.): Informatische Ideen im Mathematikunterricht, S. 17 – 36, Hildesheim 2008
- [MHR08] Monroy-Hernández, A. Resnick, M. Empowering kids to create and share programmable media In Interactions Volume 15, Issue 2 (March + April 2008) pp 50-53, MIT Media Lab, ACM DL, New York, NY, USA
- [Ra07] Pausch, R. Last Lecture (German Translation) Pittsburgh 18.09.2007 URL: <http://download.srv.cs.cmu.edu/~pausch/Randy/GermanTranslationPauschLastLecture.pdf> [l.v. 16.02.09]
- [Ro08] Romeike R.: Kreativität im Informatikunterricht, Diss. Universität Potsdam, 2008, <http://www.informatikdidaktik.de/Forschung/Schriften/RomeikeDiss2008.pdf> [l.v. 16.02.09]