



## 1. Motivation

---

Das Spektrum von Förderungsangeboten für Programmieranfänger ist dank innovativer und erschwinglicher Soft- und Hardware in den letzten Jahren beträchtlich gewachsen. Insbesondere aus dem Bereich Robotik kommt heute eine Vielzahl von Systemen, die in den Informatikunterricht an Schulen und Hochschulen eingebunden werden können (z. B. LEGO Mindstorms , NAO ). Der eher hohe Aufwand, den Lehrende bei der Einbindung derartiger Systeme in ihren Unterricht haben, sowie die entstehenden Kosten bremsen jedoch deren Verbreitung. Stärker verbreitet sind hingegen kostenlose und offene Systeme wie vereinfachte Entwicklungsumgebungen (IDEs), mit denen grundlegende Programmierkonzepte leicht vermittelt und motivierend erlernt werden können. Beispielsweise ist die grafische IDE Scratch [RMMH+09] an vielen Schulen im Einsatz und erfreut sich großer Beliebtheit.

Scratch setzt auf die Abstraktion typischer Programmierkonzepte in Form von grafischen Bausteinen. Programmieranfänger, die ein derartiges Programmierkonzept nutzen, müssen sich weniger auf die Syntax konzentrieren. Der Cognitive Load-Theorie folgend verfügen sie so über mehr lernbezogene kognitive Kapazität für den primären Lerngegenstand, die Struktur und Semantik einer Programmiersprache [BTZS12]. Zudem werden Computersysteme mit physikalischen Sensoren und Aktoren (Physical Computing [SLMR05]) zunehmend alltagsrelevant. Die durch den Alltagsbezug entstehende Motivation wird in der Scratch-Community bereits durch eine Reihe von Erweiterungen zur Einbindung physikalischer Systeme nutzbar gemacht (z. B. für Orbotix Sphero und den LEAP Motion Controller ).

Dieser Beitrag stellt das Framework ScratchDrone zur Einbindung der in Abbildung 1 dargestellten AR.Drone 2.0 – einer modernen, kostengünstigen Flugdrohne – als attraktive Laufzeitumgebung in die Scratch IDE vor. Im Vergleich zu anderen Einstiegsplattformen für Programmieranfänger werden die "begreifbare" Zielplattform [UI00], die natürliche menschliche Faszination für das Fliegen sowie die Herausforderung der Bewegung im 3D-Raum als Motivation ausgenutzt.

Zudem wird mit ScratchDrone ein Unterrichtskonzept in mehreren didaktischen Stufen unterstützt. Lehrer werden in die Lage versetzt, ihre Schüler von der reinen Bedienerseite über die Zusammenstellung einfacher und komplexer werdender Flugmanöver in Scratch bis hin zur Programmierung einer Flugdrohne in einer marktrelevanten Programmiersprache (z. B. Java, Javascript, PHP) zu begleiten. Die Lernziele der Schüler reichen von dem Kennenlernen grundlegender Programmierkonzepte in der einfachen visuellen Programmiersprache Scratch über die Nutzung fertiger REST-Kommandos bis hin zur Vertiefung der Programmierfähigkeiten in einer modernen Programmiersprache. Eine modulare Systemarchitektur stellt die dafür notwendigen Programmierschnittstellen auf drei verschiedenen Abstraktionsebenen zur Verfügung. Weiterhin wird durch das ergänzend bereit gestellte Unterrichtsmaterial Basiswissen über die Drohnentechnologie vermittelt und diese kritisch beleuchtet.

Nach einer Vorstellung der relevanten verwandten Arbeiten wird im Folgenden ScratchDrone im Detail mit seinem didaktischen Konzept und der technischen Umsetzung erläutert.



**Outdoor-Hülle**

**Indoor-Hülle**

*Abbildung 1: Die AR.Drone 2.0 kann im Indoor- und Outdoor-Bereich mit jeweils angepassten Schutzhüllen eingesetzt werden. [Bildquelle: Parrot SA]*

Zudem werden erste Evaluierungsergebnisse präsentiert. Abschließend werden die Darlegungen zusammengefasst, und es wird ein Ausblick auf weitere Arbeiten mit ScratchDrone gegeben.

## 2. Verwandte Arbeiten

---

Die Einbettung von fachlichen Lehrinhalten in realistische, physische Settings ist eine etablierte Methode zur Förderung der Informatik- und Programmierausbildung. Kreative Projekte wie das Computer-Freundebuch [Luc11] zur Vermittlung von Computerwissen unter der Metapher eines klassischen Freundebuches, E-Mail (nur?) für Dich [GHW11] zur Erarbeitung von Grundlagen der Kommunikation in Computernetzen sowie die physische Herstellung programmierbarer, interaktiver Objekte aus dem Informatikunterricht [PR13] zeigen das breite Anwendungsspektrum dieser Lehrstrategie.

Aufgrund der Popularität von Scratch als Entwicklungsumgebung für Programmieranfänger findet sich heute bereits eine Reihe von daraus ansteuerbaren Laufzeitumgebungen, die in der Regel unter einer OpenSource-Lizenz veröffentlicht wurden. Zudem ergibt die Fachliteratur vielfältige Ansätze zur Nutzung derartiger Medien im Bildungsprozess – beispielsweise zur Gestaltung von Lehreinheiten im Bereich Robotik [GSRLLO13]. Insbesondere durch derartige Projekte ist die Scratch-Erweiterung Enchanting [LMM12] zur Programmierung von LEGO Mindstorm-Systemen bekannt, die eine angepasste Firmware nutzt. Auch Scratch-Derivate zur Einbindung von Mikrocontrollern und Gestenerfassungssystemen in Scratch-Programmen sind verfügbar. Dabei kommt häufig die Scratch-Erweiterung Build your Own Blocks (BYOB) bzw. Snap! des MIT Media Lab zum Einsatz. Diese erleichtert die Erstellung eigener Scratch-Blöcke, in denen dann die Kommunikation mit der speziellen Hardware gekapselt wird.

Auch die im Folgenden verwendete Laufzeitumgebung, die AR.Drone 2.0, ist aufgrund des relativ geringen Anschaffungspreises (derzeit ca. 280 Euro), der Verfügbarkeit offener Schnittstellen sowie einer aktiven Entwickler-Community Gegenstand verschiedener Forschungsprojekte. Für den Bereich der Aus- und Weiterbildung ist beispielsweise das

Projekt Catdroid [Sla12] von Interesse. Die Android-Software ermöglicht ebenfalls die Programmierung von Flugdrohnen über eine Scratch-ähnliche grafische Entwicklungsoberfläche.

Sämtliche Lösungen für Scratch und einzelne Laufzeitumgebungen sind als Einzellösungen zu betrachten. Das erschwert den Einsatz in größeren, zusammenhängenden Unterrichtssequenzen bzw. verschiedenen Schwierigkeits-/Abstraktionsstufen. Das im Folgenden beschriebene Framework setzt im Unterschied dazu auf einen flexibleren und systematischen Ansatz durch eine modulare Architektur. Diese ermöglicht beispielsweise mehrstufige didaktische Szenarien für aufeinander aufbauende Unterrichtseinheiten sowie eine freie Technologiewahl durch den Lehrenden bei konsistenter Infrastruktur und Bedienung.

### 3. ScratchDrone

---

#### 3.1. Unterrichtskonzept

ScratchDrone wurde für den Einsatz im Informatikunterricht an Schulen entwickelt und profitiert ggf. von Scratch-Erfahrungen, die in Informatikklassen bereits vorliegen. Das Lernziel variiert je nach Kenntnisstand der Schüler und besteht im Allgemeinen aus der Vertiefung der eigenen Programmierkenntnisse am Beispiel eines konkreten physikalischen Systems. Die Zielgruppe reicht dank des einfachen Scratch-Programmiermodells von Schülern, die noch nie programmiert haben, bis hin zu erfahrenen Schülern aus Informatik Leistungskursen oder Studienanfängern. Möglich ist dies durch ein vierstufiges didaktisches Unterrichtskonzept, das in Abbildung 2 skizziert ist. Es orientiert sich an einem klassischen Curriculum, in dem das Unterrichtsthema von einer abstrakten (Steuerung über eine stark vereinfachte App) bis hin zu einer detaillierten Ebene (Programmierung mit typischer Programmiersprache) erarbeitet wird. Ergänzend wird beim ersten Kontakt der Schüler mit ScratchDrone eine Einführung in nicht-technische Aspekte der Technologie (z. B. gesellschaftliche Aspekte des Drohneneinsatzes, Einsatzszenarien, Risiken und Potentiale) empfohlen.

In der ersten Stufe lernen die Schüler zunächst mit Werkzeugen des Herstellers die Drohne kennen. Insbesondere die Reaktion auf Wind im Freien, auf Hindernisse im Flugbereich sowie die Geschwindigkeit und Präzision der Umsetzung von Steuerbefehlen sollten zu Beginn vermittelt werden. Für Flugdrohnen wie die in diesem Projekt verwendete AR.Drone 2.1 ist vom Hersteller in der Regel eine Steuerung über Smartphone-Apps vorgesehen, die per WLAN mit der Drohne kommunizieren und für die ersten Flugmanöver eingesetzt werden können. Neben den grundlegenden Flugbewegungen können über diese Apps auch Videos aufgenommen und Sensordaten abgerufen werden, sodass den Schülern das technische Potential der Drohnentechnologie vermittelt wird. Diese Stufe kann je nach Klassenstärke in ca. einer Schulstunde absolviert werden.



Abbildung 2: ScratchDrone ist didaktisch in ein vierstufiges Unterrichts konzept eingebunden.

In der zweiten Stufe wird erstmals mit Programmier elementen gearbeitet. Die Schüler können mit vorgefertigten Scratch-Blöcken in dem Scratch-Derivat Snap! einfache Flugsequenzen programmieren. Die entsprechenden Blöcke werden mit ScratchDrone mitgeliefert und ständig erweitert. Derzeit können 19 Flugkommandos genutzt werden. Dazu gehören:

- 1 Startkommando
- 1 Landekommando
- 12 3D-Bewegungen (z. B. Drohne fliegt 50 cm nach links)
- 1 Schwebekommando
- 4 Drehungen (45 Grad nach rechts/links und für eine bestimmte Zeit nach rechts/links)

Außerdem ist es möglich die LED-Lichter an den vier Rotoren grün, rot oder orange leuchten zu lassen. Desweiteren werden Sensorabfragen zum aktuellen Batteriestand und der aktuellen Flughöhe zur Verfügung gestellt.

Abbildung 3 zeigt eine Übersicht über die ScratchDrone-Befehle. Die Flugkommandos finden sich im Scratch-Bereich Bewegung, die LED-Kommandos im Bereich Aussehen und die Sensorabfragen im Bereich Fühlen. Alle Befehle können mit anderen Scratch-

Elementen kombiniert werden (z. B. Schleifen, Verzweigungen, Zufallszahlengenerator). Scratch-erfahrene Schüler können zudem Scratch-Ereignisse (z. B. Tastatureingaben) verwenden, um die asynchrone Flugsequenz mit synchronen Elementen anzureichern (z. B. für eine Live-Steuerung per Tastaturbefehl).

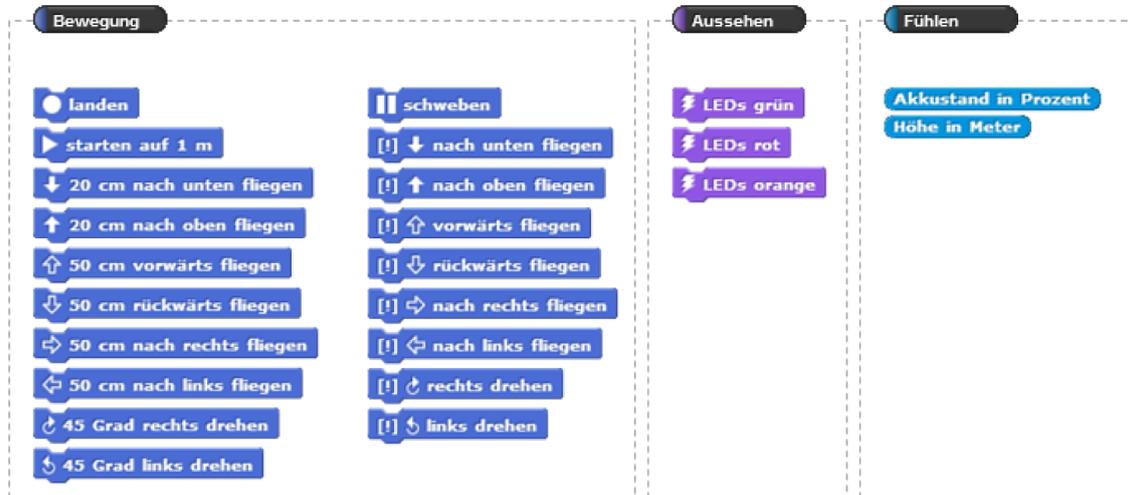


Abbildung 3: Die zusätzlichen ScratchDrone-Befehle sind drei Scratch-Bereichen zugeordnet.

Nachdem die Schüler ihre Scratch-Programme entwickelt haben, werden diese eingesammelt und über den Lehrerrechner mit der Drohne ausgeführt. Dadurch ist sichergestellt, dass der Lehrer die Programme vor ihrer Ausführung prüfen kann, um riskante Manöver zu identifizieren und ggf. zu vermeiden. Dazu gehören beispielsweise Flüge in über 30 m Höhe oder in Bäume, die die Hardware beschädigen würden. Idealerweise werden die Programme je nach Witterung im Freien oder in einer Turnhalle bzw. großen Aula der Schule gemeinsam mit den Schülern ausgeführt. Erfahrungsgemäß wird für diese Stufe bei einer Klassenstärke von 10-15 Schülern eine Doppelschulstunde benötigt.

Die dritte Stufe entspricht vom organisatorischen Ablauf der zweiten Stufe. Allerdings sind die Schüler nun aufgefordert, eigene Blöcke zur Drohnensteuerung zu entwerfen. Dabei können sowohl existierende ScratchDrone-Blöcke wiederverwendet als auch von Grund auf neue Blöcke erstellt werden. Abbildung 4 zeigt den in dieser Stufe verwendeten Blockeditor mit einer beispielhaften Blockdefinition. Die Schüler nähern sich somit etwas weiter der eigentlichen Programmierung über eine marktrelevante Programmiersprache. Stufe 2 und 3 können zudem während des Unterrichts parallel laufen, um gezielt besonders leistungsstarke Schüler zu fördern (Stufe 3).

In Stufe 4 kann die Flugdrohne ohne Scratch in einer vom Lehrer zu wählenden Programmiersprache programmiert werden. ScratchDrone umfasst zu diesem Zweck eine Kapselung der hardwarenahen Steuerung über UDP-Befehle durch REST-Services, die von nahezu beliebigen Programmiersprachen angesteuert werden können. Sofern der Lehrer sich mit seinem Unterricht inhaltlich weiter der Hardware nähern möchte, besteht zudem die Möglichkeit, die UDP-Kommandos der AR.Drone 2.0 unmittelbar zu nutzen. In

diesem Fall ist das ScratchDrone-Framework nicht mehr erforderlich, da diese Option vom Hersteller selbst unterstützt wird. Die zeitliche Dauer dieser Phase ist stark von der ausgewählten Didaktik abhängig.



Abbildung 4: Die Schüler können existierende Blöcke zu neuen Blöcken zusammenfügen.

Je nach verfügbarer Zeit und Wissensstand werden die Stufen für einen konkreten Unterricht ausgewählt. Das komplette Unterrichtskonzept eignet sich eher für 1-2 Projektstage, vor allem während der ersten Programmiererfahrungen. Sinnvoll für den kompakten ScratchDrone-Einsatz ist hingegen eine Doppelschulstunde, in der Stufe 1 mit einer weiteren der Stufen 2-4 kombiniert wird. Ferner ist dieses Konzept nur ein Vorschlag der Verwendung von ScratchDrone. Die Informatiklehrer an den verschiedenen Schulen haben jede Freiheit, dieses Konzept zu modifizieren, um es den Lernzielen einer konkreten Klasse sowie dem jeweiligen Fortschritt anzupassen.

### 3.2. Technische Realisierung

Das vierstufige Unterrichtskonzept erfordert ein modulares Framework, das die Programmier- und Benutzungsschnittstellen für die einzelnen didaktischen Stufen 2-4 anbietet. Dieses wurde an der Universität Potsdam im Rahmen von mehreren, aufeinander aufbauenden Bachelorarbeiten realisiert. Die Grobarchitektur ist in Abbildung 5 dargestellt.

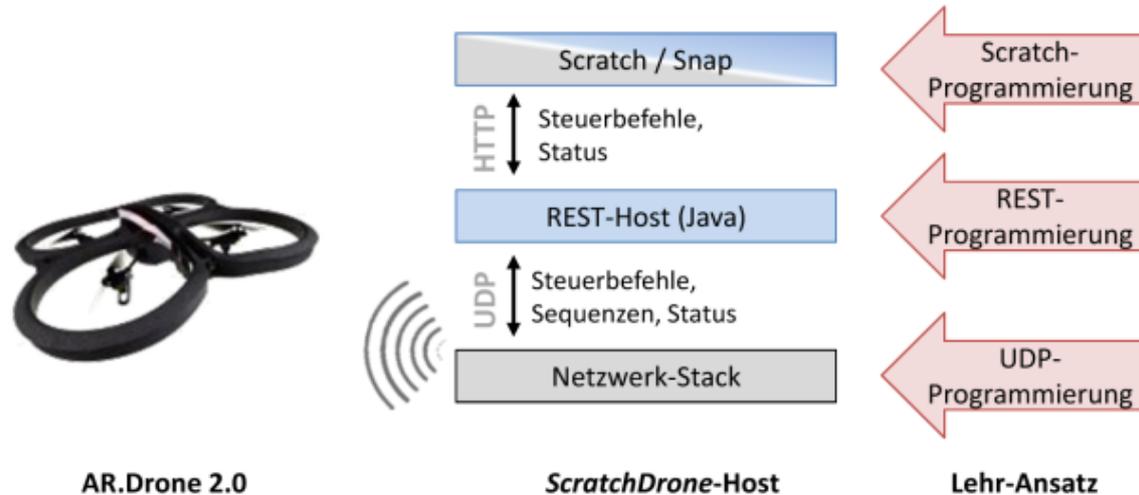


Abbildung 5: Die mehrstufige ScratchDrone-Architektur bietet verschiedene Abstraktionsstufen für den Programmierunterricht.

Der AR.Drone 2.0 werden über WLAN Steuerkommandos gesendet. Zu diesem Zweck wurden vom Hersteller Parrot eine Reihe einzelner UDP-Kommandos spezifiziert. Zudem kann über die UDP-Schnittstelle auf Sensoren und Kamerabilder der AR.Drone 2.0 zugegriffen werden [PBED12]. Die Sensorinformationen zur Flughöhe und zum Batteriefüllstand wurden teilweise bereits zur Verwendung in ScratchDrone implementiert. Die Integration der Kamerabilder ist für künftige Updates vorgesehen.

Mit den UDP-Kommandos kann ein großes Spektrum von Steuerkommandos in verschiedenen Granularitätsstufen genutzt werden. Für die Zielgruppe der Programmieranfänger besteht bei der Umsetzung des Gesamtspektrums jedoch die Gefahr der Überforderung durch zu hohe Komplexität. Aufgrund dessen musste im Rahmen der Systemkonzeption eine sinnvolle Auswahl von vordefinierten Kommandos getroffen werden. Dabei war bei den grundlegenden Flugbewegung – neben Start und Landung – das Hauptziel, die Bewegung im 3D-Raum mit allen drei Freiheitsgraden zu ermöglichen. Ergänzt wurde diese Bewegung durch die Möglichkeit, die Drohne um ihre eigene Achse zu drehen um das räumliche Denken durch Variation der Bezugspunkte im Raum zu fordern.

Aufgrund der Heterogenität der Zielgruppe wurden zwei Befehlssätze umgesetzt. Der Basic-Befehlssatz enthält einerseits nur Flugbewegungen, die eine vordefinierte Distanz zurücklegen und die Drohne im Anschluss schweben lassen. Dies ermöglicht absoluten Anfängern zuverlässig, im Voraus die Drohnenbewegung abzuschätzen. Andererseits erzeugen die vielen Schwebepausen kurze Unterbrechungen in den Flugbewegungen. Der Advanced-Befehlssatz ergänzt den Basic-Befehlssatz um acht weitere Befehle, die so lange ausgeführt werden, bis sie durch einen anderen Flugbefehl unterbrochen werden (in Abbildung 3 durch den Präfix [!] erkennbar). Dadurch kann auf jeden Flugbefehl sofort ein neuer Befehl folgen, und die Bewegungen wirken deutlich flüssiger. Dieser Befehlssatz ist aber nur fortgeschrittenen ScratchDrone-Anwendern zu empfehlen, da die Drohnenbewegungen nur schwer im Vorfeld abgeschätzt werden können. Die Befehle zur Steuerung der LEDs, Abfrage der Höhe sowie des Batteriestandes sind in beiden Befehlssätzen verfügbar.

Selbst die mit konkreten Distanzen versehenen Flugbefehle sind als Schätzwerte zu betrachten. Sie wurden ermittelt, indem mehrmals gemessen wurde, wie lange ein UDP-Befehl gesendet werden muss, um z. B. 50 cm in der Horizontalen zurückzulegen. Wind und andere physikalische Umgebungseigenschaften tragen dazu bei, dass diese Distanzen im Feldtest nicht immer exakt zurückgelegt werden. In geschlossenen Räumen (z. B. Turnhallen) sind die Angaben deutlich zuverlässiger.

Nachdem in früheren ScratchDrone-Versionen noch auf eine installierbare Scratch-Anwendung für Schüler und Lehrer gesetzt wurde [ZHWL14], erfolgt die Programmierung in Scratch inzwischen direkt im Browser über das Scratch-Derivat Snap!. Über drohnenspezifische Scratch-Blöcke werden die REST-Kommandos per HTTP an den REST-Host versendet. Abbildung 6 zeigt diese Benutzungsschnittstelle auf der obersten Schicht von ScratchDrone.

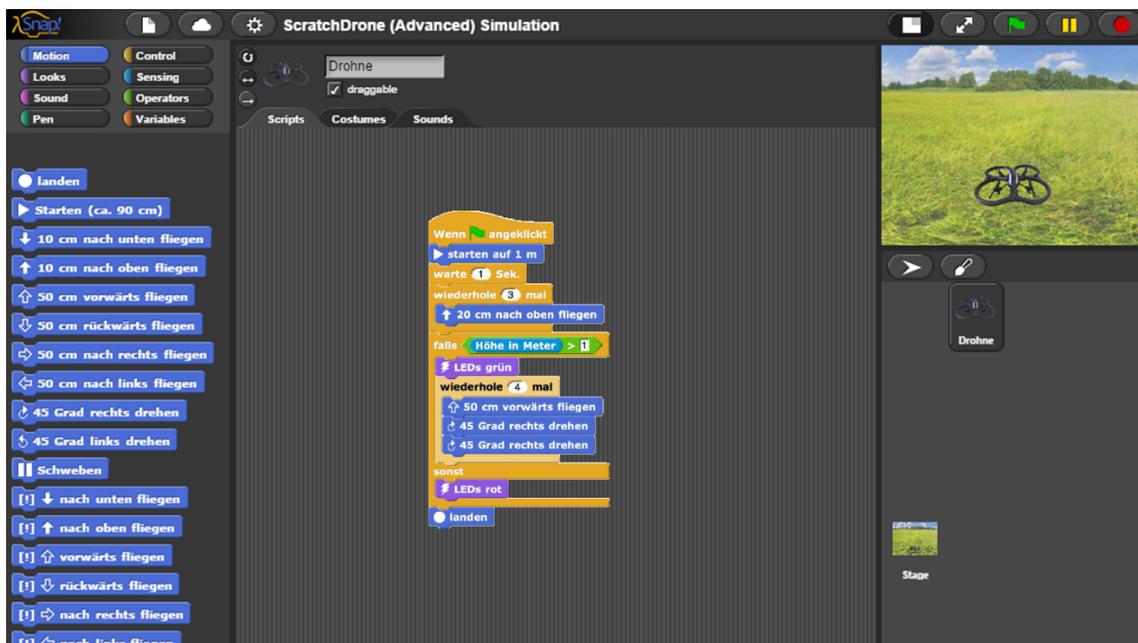


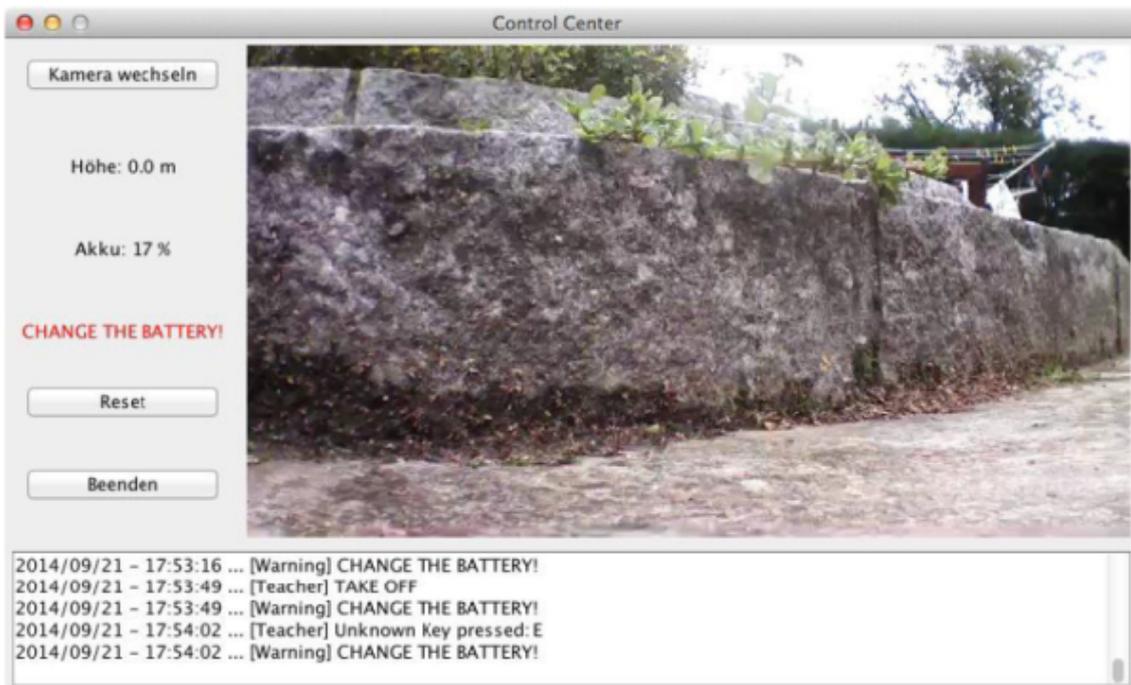
Abbildung 6: Das Scratch-Derivat Snap! wurde um Befehle zur Drohnensteuerung erweitert.

Neben der klassischen Drag-and-Drop-GUI von Scratch wurde auch das Vorschaubild durch ein Foto der Flugdrohne, welches pseudo-dreidimensional vor einem Naturhintergrund transformiert wird, auf das ScratchDrone-Szenario angepasst. Zumindest einfache Manöver können so bereits vor der physischen Ausführung als Scratch-Animation simuliert werden.

Als Sicherheitsmechanismus kann der Lehrer zudem über das Java-Programm des REST-Hosts jederzeit per Tastatursteuerung in eine laufende Flugsequenz eingreifen und die Drohne so z. B. landen oder Hindernisse umfliegen. Abbildung 7 zeigt das Lehrer-Interface im Screenshot. Ferner bietet die AR.Drone 2.0 selbst bereits eine Reihe von Sicherheitsmechanismen, wie beispielsweise das automatische Absinken auf 6 m beim Verbindungsabbruch und die rudimentäre Kompensierung von Windböen. Daher ist diese Plattform auch für Unerfahrene unkompliziert einsetzbar.

Durch die modulare Systemarchitektur können Lehrer ScratchDrone flexibel gemäß ihren eigenen Lehrplänen in den Unterricht einbinden. Sie werden technologisch nicht eingeschränkt, sobald sie die einfache Scratch-Ebene verlassen und eine marktrelevante Programmiersprache einführen möchten. Dank der REST-Kapselung der konkreten Drohnensteuerung ist zudem der Wechsel auf eine andere Flugdrohnen-Hardware mit vergleichsweise geringem Programmieraufwand möglich. Weiterhin ermöglicht diese Abstraktionsschicht das Steuern der Drohne über

mehr als einen Client. Somit kann der Lehrer jederzeit das aktuell laufende Programm abbrechen oder anderweitig einschreiten, falls dies beispielsweise aus Sicherheitsgründen erforderlich ist.



*Abbildung 7: Der Lehrer kann über eine eigene Java-Anwendung die Ausführung der Scratch-Programme überwachen und jederzeit eingreifen.*

### 3.3 Unterrichtsmaterial

Neben der technischen Umsetzung von ScratchDrone sind in Zusammenarbeit mit Studierenden des Lehramts Informatik verschiedene Unterrichtsmaterialien entstanden, die Lehrern für ihre ScratchDrone-Unterrichtseinheiten zur Verfügung gestellt werden.

Abbildung 8 zeigt eine Auswahl des verfügbaren Unterrichtsmaterials. Folgende Artefakte stehen derzeit zur Verfügung:

- Einführung in ScratchDrone (Präsentation)
- Einführung in die Drohnentechnologie, ihre Einsatzfelder, Risiken, Potentiale und gesellschaftlichen Implikationen (Präsentation)

- Anleitung zur Inbetriebnahme von ScratchDrone
- Handout für Schüler zur Übersicht über die einzelnen Kommandos inkl. Hinweisen und Beispielprogrammen
- Unterrichtskonzept für eine einmalige Doppelstunde
- Unterrichtskonzept für eine Reihenplanung von sechs Schulstunden (z. B. im Rahmen eines Projekttages)

**ScratchDrone**  
Bedienungsanleitung

**Vorschlag Doppelstunde**

Stunde	Thema	Didaktische Schwerpunktsetzung
1	Einführung Drohnen	<ul style="list-style-type: none"> <li>Einführung in die Thematik (siehe Präsentationsmaterial, je nach dem, wie umfangreich dieser Teil behandelt werden soll, wird nicht unbedingt eine ganze Unterrichtsstunde benötigt)</li> </ul>
2+3	Drohnen-Steuerung	<ul style="list-style-type: none"> <li>Motivation: kurze Vorführung der Drohne</li> <li>Programmier-Anfänger:                             <ul style="list-style-type: none"> <li>ScratchDrone (Safe) als Variante zur Programmierung der Steuerung vorstellen</li> <li>einfache Sequenzen mit den SchülerInnen zusammen entwickeln und vorführen (Start - Landung, einfache geometrische Formen: "Fliege ein Quadrat." etc.) sowie die einfachen Bewegungsbefehle erklären</li> <li>das Steuerungselement <i>Auswahl</i> vorstellen und vorführen (z. B. "Zufallszahl &gt; 5 drehe links, sonst rechts") sowie wichtige Operatoren erklären (vor allem Vergleichsoperatoren)</li> <li>Wiederholungen vorstellen: z. B. "Wie kann ich drei Quadrate fliegen lassen?", dann die elegantere Form mit Zählschleifen vorstellen und -führen; Schleifen mit Abbruchbedingungen z. B. "Fliege nach oben bis Höhe &gt; 150 cm" und Sensorabfragebefehle der Drohne vorstellen</li> <li>Übungseinheit für die SchülerInnen je nach Bedarf und Zeit: zwischendurch für die einzelnen Strukturen (Sequenz, Auswahl und Wiederholung) und/oder am Ende, Lösungen vorstellen und einige ausführen</li> </ul> </li> <li>Fortgeschrittene:                             <ul style="list-style-type: none"> <li>ScratchDrone (Advanced) als Variante zur Programmierung der Steuerung vorstellen</li> <li>vorführen einfacher Sequenzen, <i>Auswahlen</i> und <i>Wiederholungen</i></li> <li>Übungseinheit für die SchülerInnen (z. B. "Fliege ein Quadrat", "Fliege drei Quadrate mit Hilfe einer Schleife" usw.)</li> <li>Problem: Schrägen z. B. im rechteckigen Dreieck -&gt; kontinuierliche Bewegungsbefehle</li> <li>SchülerInnen Ideen für ein rechteckiges Dreieck entwickeln lassen und gemeinsam auswerten/vorführen</li> <li>Problem: Kreisbahn -&gt; kontinuierliche Rotationsbefehle</li> <li>SchülerInnen Ideen für einen Kreis entwickeln lassen und gemeinsam auswerten/vorführen</li> <li>Übungseinheit für die SchülerInnen (z. B. "Fliege ein gleichseitiges Dreieck", "Fliege eine Ellipse" - 2 Halbkreise + 2 Geraden usw.)</li> <li>je nach Leistung der SchülerInnen  und/oder Freestyle als Abschluss (vielleicht sogar eine sich nicht verjüngende Spirale)</li> </ul> </li> </ul>
6	Parcours fliegen	<ul style="list-style-type: none"> <li>"Drehung"</li> <li>als Wiederholung/Festigung verschiedene Bewegungsmuster als eigene Blöcke in Snap speichern und am Ende als Parcours aneinander reihen</li> <li>Flugparcours vorführen (am besten in der Turnhalle)</li> </ul>

**Zivile Nutzung**

**ScratchDrone 2.0**

**Handout**

**Hinweise**  
**Dringen:**  
Max. 2,5m in die Höhe und max. in einem 5m-Radius um den Startplatz fliegen lassen

Abbildung 8: Lehrern wird ein breites Spektrum an Unterrichtsmaterial für ScratchDrone zur Verfügung gestellt.

Die einzelnen Materialien wurden im Laufe der Tests zusammen mit Informatiklehrern optimiert und an die Bedürfnisse von Unterrichtssituationen an Schulen angepasst.

## 4. Evaluierung

Die Evaluierung von ScratchDrone erfolgte im Sommer 2014 in Kooperation mit verschiedenen Schulen – vorwiegend Gymnasien – in Berlin und Brandenburg. Der Fokus lag dabei zunächst auf der motivationalen Ebene. Die folgenden Ergebnisse basieren auf dem Einsatz von ScratchDrone an acht Schulen. Insgesamt beteiligten sich 166 Schüler in zwölf unterschiedlichen Schulklassen (7.-12. Klasse). Das Vorwissen der Schüler reichte von ersten Erfahrungen mit Programmierung bis hin zu fortgeschrittener Programmiererfahrung (häufig mit Python oder Scratch). Flugdrohnen waren fast ausschließlich nur aus den Medien und Computerspielen bekannt.

Die Tests wurden stets im Rahmen des Informatikunterrichts und unter Anwesenheit der Informatiklehrer durchgeführt. In der Regel wurde den Projektmitarbeitern jedoch die Leitung der Unterrichtsstunde überlassen. Das Unterrichtsformat war in acht Fällen eine Doppelstunde und in zwei Fällen ein Projekttag.

Zunächst wurde den Schülern eine Einführung in die Drohnentechnologie sowie die Modifikationen der Scratch-Umgebung gegeben und anschließend Stufe 2 (Einführung in die Scratch-Steuerung) des Unterrichtskonzeptes ausgeführt. Je nach verfügbarer Zeit und Lerngeschwindigkeit der Schüler wurden diese aufgefordert, dabei ein oder mehrere der in Abbildung 9 dargestellten Flugmanöver zu programmieren.



Abbildung 9: Die Schüler wurden aufgefordert eine oder mehrere der dargestellten Flugmanöver zu programmieren.

Im Anschluss wurden die Teilnehmer gebeten einen Feedback-Fragebogen auszufüllen. Dieser bestand aus einem User Experience Questionnaire (UEQ) [LHS08], Freitextfragen (z. B. zu bisherigen Programmiererfahrungen und Eindrücken von den ScratchDrone-Tests) sowie Likert-Skala-Abfragen zu

- Spaß an der Arbeit mit ScratchDrone (3: sehr Spaß bis -3: sehr langweilig)
- mehr Lust auf Programmierung (3: auf jeden Fall bis -3: auf keinen Fall)
- Zuwachs an Interesse für Informatik im Allgemeinen (mehr Lust auf . . .)
- mehr Lust auf Programmierung mit Scratch (3: auf jeden Fall bis -3: auf keinen Fall)
- mehr Lust auf die Programmierung von Flugdrohnen (3: auf jeden Fall bis -3: auf keinen Fall)

Abbildung 10 fasst die Ergebnisse der fünf Likert-Skala-Abfragen vergleichend zusammen. Dabei wird zwischen Teilnehmern aus der Sekundarstufe I (Anfänger) und der Sekundarstufe II (Fortgeschrittene) unterschieden.

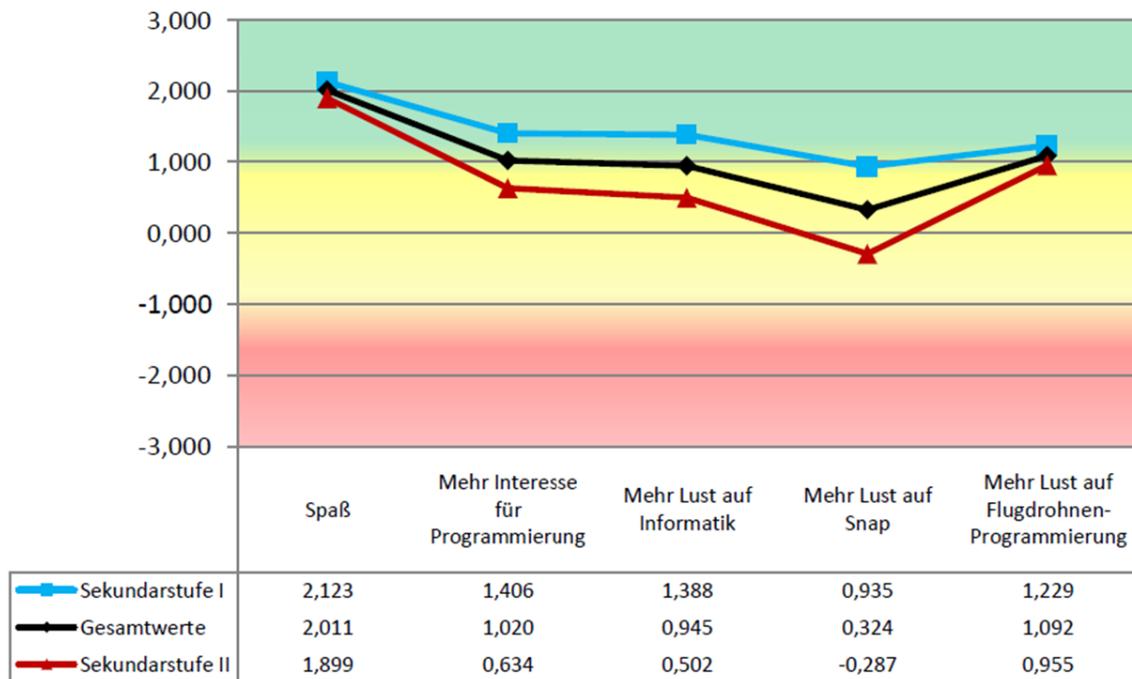


Abbildung 10: Die Evaluierungsergebnisse belegen die motivationale Wirkung von ScratchDrone für Anfänger und fortgeschrittene junge Programmierer.

Beide Zielgruppen hatten erwartungsgemäß großen Spaß an der Drohnenprogrammierung. Vor allem die exotische Plattform, kombiniert mit den schnellen Erfolgserlebnissen dank Scratch, haben dazu beigetragen. Bei beiden Gruppen konnte zudem das Interesse an Programmierung und Informatik gesteigert werden. ScratchDrone eignet sich somit gut für die Motivation beider Themen sowie als Abwechslung zum traditionelleren Informatikunterricht. Die Teilnehmer der Sekundarstufe I waren gegenüber den fortgeschrittenen Programmierern jedoch noch eher zu begeistern. Es kann angenommen werden, dass bei den fortgeschrittenen Teilnehmern schon eine leichte Unterforderung einsetzte.

Die Schüler der Sekundarstufe I haben zudem einen Interessengewinn für die Programmierung mit Scratch/Snap! erfahren. Vor allem die durch Scratch wahrgenommene Leichtigkeit der Programmierung war für diese Gruppe neu und hat sich auch positiv auf das Scratch-Interesse ausgewirkt. Für die Sekundarstufe II wurde das Interesse an Scratch leicht geschwächt. Die Schüler fühlten sich durch Scratch eher beschränkt in ihren Möglichkeiten die Drohne zu steuern. Bestätigt wird diese Erkenntnis durch ein großes Spektrum an Vorschlägen aus dieser Gruppe für weitere ScratchDrone-Flugmanöver und -Features.

Das große Interesse beider Gruppen an der Programmierung von Flugdrohnen untermauert die Annahme, dass vor allem durch die exotische Laufzeitplattform ein deutlicher Motivationsgewinn erzielt werden kann. Weitere Evaluierungen – vor allem der Stufe 4 – müssen zeigen, ob die Motivation über eine einfache Programmierumgebung wie Scratch hinaus auch für komplexere, marktrelevante Programmiersprachen ausgenutzt werden kann.

Die Ergebnisse der UEQ-Abfrage zur allgemeinen Benutzungserfahrung sind in Abbildung 11 einem Benchmark mit 163 anderen UEQ-Studien gegenübergestellt. Dabei kann in vier von sechs Aspekten eine klar positive Wahrnehmung festgestellt werden. Nur die Steuerbarkeit und die Effizienz von ScratchDrone wurde als unterdurchschnittlich bewertet.

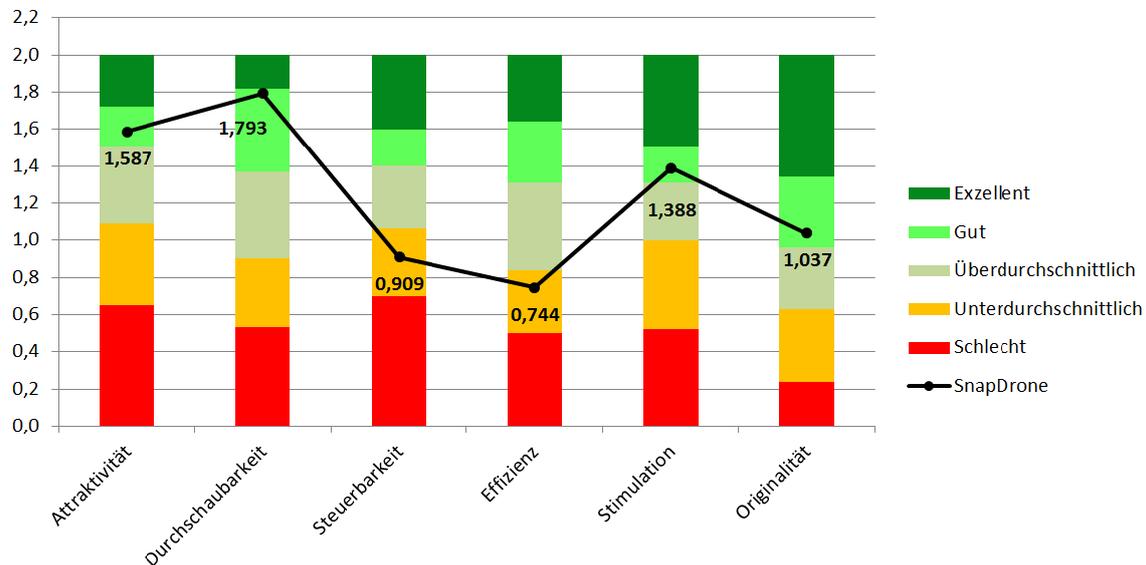


Abbildung 11: Die Benutzungserfahrung wurde im Wesentlichen als positiv wahrgenommen.

Diese Einschätzung ist aufgrund der Scratch-eigenen, kindgerechten Abstraktion von Programmierkonzepten klar nachvollziehbar. In der Benutzungserfahrung wurden keine statistisch signifikanten Unterschiede zwischen beiden Testgruppen festgestellt.

In beiden Gruppen kam zudem vereinzelt der Wunsch auf, tiefer in die Drohnensteuerung einzusteigen, als die vorgefertigten Scratch-Blöcke in Stufe 2 es erlauben. Auch von Seiten der Lehrer wurde die Nutzung der Flugdrohne für andere Programmiersprachen – wie es für Stufe 4 vorgesehen ist – angeregt. Diese Stufen werden in den kommenden Evaluierungen ebenfalls Unterrichtsgegenstand sein, sodass das systematische Unterrichtskonzept in seiner Gänze fundierter bewertet werden kann.

## 5. Zusammenfassung und Ausblick

Der Einstieg in die Programmierung wird heute durch eine Reihe von Förderangeboten erleichtert. Insbesondere durch einfache IDEs wie die grafische Programmierumgebung Scratch können schon früh motivierende Erfolge erzielt werden. Das in diesem Artikel vorgestellte Framework ScratchDrone nutzt die in Schulen breit vorhandene Scratch-Erfahrung, um Schülern die Programmierung einer Flugdrohne als exotische Laufzeitumgebung zu ermöglichen.

Im Vergleich zu ähnlichen Angeboten (z. B. für den Ball Orbotix Sphero) wird durch ScratchDrone die Komplexität der Bewegung im 3D-Raum fokussiert. Zudem ermöglicht die modulare Systemarchitektur den Einsatz im Informatikunterricht auf unterschiedlichen Abstraktionsstufen der Programmierung. Das resultierende didaktische Unterrichtskonzept

nutzt vier aufeinander aufbauende Stufen der Programmierung, um auch Anfänger schrittweise mit der Thematik vertraut zu machen. Einzelne Stufen können weiterhin autonom unter Berücksichtigung der Fachkenntnisse der Schüler ausgewählt und durchgeführt werden. Eine Modifikation der ScratchDrone-Architektur ist dafür nicht erforderlich.

Dank einer Kapselung der eigentlichen Drohnensteuerung hinter einer Reihe von plattformunabhängigen REST-Services bleibt dem Lehrer – über Scratch hinaus – die Entscheidung überlassen, welche Programmiersprache er lehren will. Selbst die Drohnen-Hardware kann ausgetauscht werden, sofern die REST-Kapselung bedient wird. Die grundlegende ScratchDrone-Architektur muss auch dafür nicht geändert werden. Eine exemplarische Umsetzung unter Nutzung kostengünstiger Drohnen (ca. 40-50 Euro/Stk.) des Herstellers Hubsan ist dazu aktuell im Entstehen.

Erste Evaluierungsergebnisse belegen den Erfolg von ScratchDrone bei der Motivation von Anfängern und fortgeschrittenen Programmierschülern. Insbesondere die Attraktivität, Durchschaubarkeit, Stimulation und Originalität der Lösung wurden in mehreren Tests an Schulen in Berlin/Brandenburg belegt. Erst nach breiteren und vor allem über einen längeren Zeitraum durchgeführten Tests scheinen beispielsweise Untersuchungen der Lernwirksamkeit sinnvoll. Diese werden dann mit den beteiligten Schulen durchgeführt.

Neben und in weiteren Evaluierungsläufen, die Informatiklehrer inzwischen eigenverantwortlich durchführen, werden auch eine Reihe organisatorischer und technischer Neuerungen sukzessive konzipiert, implementiert und getestet. Dazu gehören:

- Erstellung neuer ScratchDrone-Blöcke, darunter weitere Sensorabfragen (z. B. Ausrichtung im 3D-Raum) und weitere Flugmanöver (z. B. Salto in der Luft)
- Konzeption von Flugmissionen (z. B. Parcours-Flug) zur weiteren Motivation und um die Programme der Schüler zielgerichtet vergleichen zu können
- Untersuchung weiterführender Evaluierungssitems (z. B. Lernerfolg)
- Integration weiterer, günstigerer Flugdrohnen (mit beschränktem Funktionsumfang)

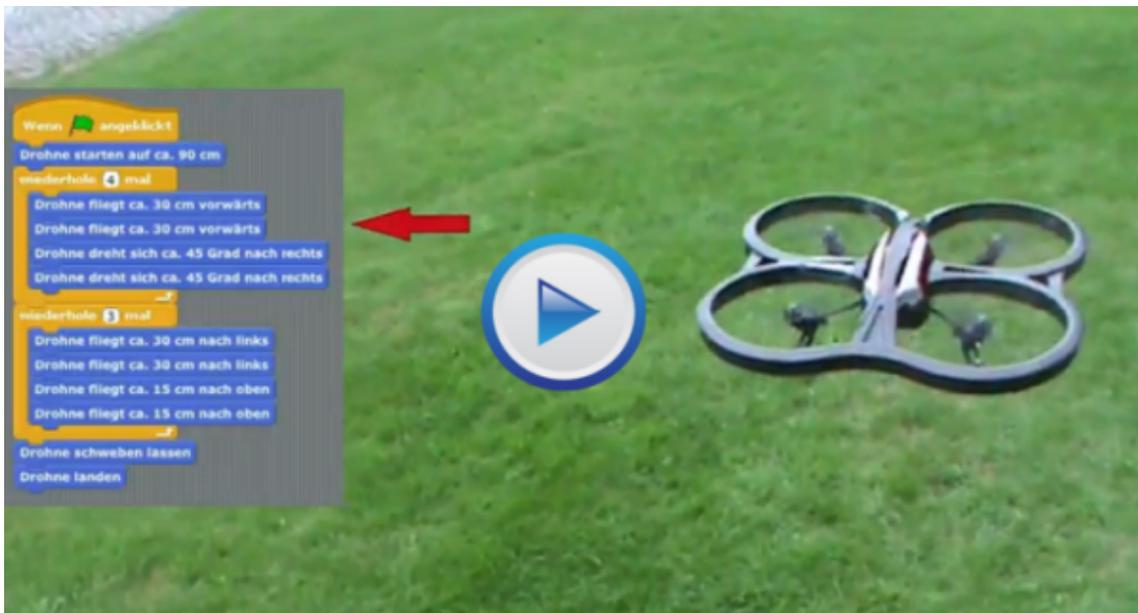
Das derzeit noch prototypische ScratchDrone-Framework wird zudem im Zuge dieser Überarbeitungen unter einer OpenSource-Lizenz für den freien Einsatz im schulischen Programmierunterricht zur Verfügung gestellt. Schon jetzt kann die Software inklusive Unterrichtsmaterial und -konzepten in Einzelfällen von den Autoren bezogen werden, um sich an den Tests vor der öffentlichen Freigabe zu beteiligen.

## Danksagung

---

Die Autoren danken den Schülern und Lehrern aus Berlin/Brandenburg, die bei der Evaluierung mitgewirkt haben, für ihre Aufgeschlossenheit und Kreativität beim Ausprobieren von ScratchDrone sowie zahlreiche hilfreiche Anregungen. Zudem wäre die hohe Qualität der Ergebnisse nicht ohne erste Fallstudien im Bereich der Kapselung von Drohnen-Steuerbefehlen in der Bachelorarbeit von Herrn Patrick Wolfien erreicht worden.

## Multimedia



Video: ScratchDrone im Einsatz

Alternativer Download des Videos "ScratchDrone im Einsatz".

## Literaturverzeichnis

[BTZS12] Brauner, Philipp; Thüs, Hendrik; Ziefle, Martina; Schroeder, Ulrik: ScratchTab - Eine Tablet-basierte Anwendung zum Erlernen von Programmierkonzepten. In: Proc. 2. Workshop Mobile Learning im Rahmen der DeLFI 2012, Fernuniversität Hagen, 2012, pp. 15–22.

[GHW11] Gramm, Andreas; Hornung, Malte; Witten, Helmut: E-Mail (nur?) für Dich – Eine Unterrichtsreihe des Projekts Informatik im Kontext. In: LOG IN, 31, Nov 2011, pp. 169-170.

[GSRLLO13] Sierra, Juan Felipe García; Lera, Francisco J. Rodríguez; Llamas, Camino Fernández; Olivera, Vicente Matellán: Inside the Maze: Who Would Find the Cheese First, a Robot or a Mouse?: Teaching IT Using Robots. In: Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality. TEEM '13, ACM, New York, NY, USA, 2013.

[LHS08] Laugwitz, Bettina; Held, Theo; Schrepp, Martin: Construction and Evaluation of a User Experience Questionnaire. In: Holzinger, Andreas (Hrsg.): HCI and Usability for Education and Work. Jgg. 5298 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 63–76.

[LMM12] Lane, Aidan; Meyer, Bernd; Mullins, Jonathan: Robotics with Enchanting and LEGO NXT - A Project Based Introduction to Programming. Monash University, 2012.

- [Luc11] Lucke, Ulrike: Das Computer-Freundebuch: Ein Ansatz für Informatik in der Grundschule. In: 14. GI-Fachtagung Informatik und Schule (INFOS), Münster, 2011.
- [PBED12] Piskorski, Stephane; Brulez, Nicolas; Eline, Pierre; D'Haeyer, Frederic: AR.Drone Developer Guide SDK 2.0. Parrot S.A., 2012.
- [PR13] Przybylla, Mareen; Romeike, Ralf: Physical Computing mit "My Interactive Garden". In: 15. GI Fachtagung Informatik und Schule (INFOS), Kiel, 2013, pp. 87–91.
- [RMMH+09] Resnick, Mitchel; Maloney, John; Monroy-Hernández, Andrés; Rusk, Natalie; Eastmond, Evelyn; Brennan, Karen; Millner, Amon; Rosenbaum, Eric; Silver, Jay; Silverman, Brian; Kafai, Yasmin: Scratch: Programming for All. *Commun. ACM*, 52(11), November 2009, pp. 60–67.
- [Sla12] Slany, Wolfgang: Catroid: A Mobile Visual Programming System for Children. In: Proceedings of the 11th International Conference on Interaction Design and Children. IDC '12, ACM, Bremen, 2012, pp. 300–303,
- [SLMR05] Stankovic, John A.; Lee, Insup; Mok, Aloysius; Rajkumar, Raj: Opportunities and obligations for physical computing systems. In: *Computer*, 38(11), Nov 2005, pp.23–31.
- [UI00] Ullmer, Brygg; Ishii, Hiroshi: Emerging Frameworks for Tangible User Interfaces. In: *IBM Syst. J.*, 39(3-4), Juli 2000, pp. 915–931.
- [ZHWL14] Zender, Raphael; Höfler, Julius; Wolfien, Patrick; Lucke, Ulrike: ScratchDrone - Systematische Programmierung von Flugdrohnen für den Informatikunterricht. In: Proc. Die 12.