

# Informatik und Mathematik – kombiniert im Schülerlabor-Modul „Einstieg in die Computergrafik“

Nadine Bergner, Tim Schellartz, Ulrik Schroeder

Lehr- und Forschungsgebiet Informatik 9

RWTH Aachen

Ahornstr. 55

52074 Aachen

bergner@informatik.rwth-aachen.de

schroeder@informatik.rwth-aachen.de

tim.schellartz@rwth-aachen.de

**Abstract:** Das Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ bietet informatikbegeisterten Schülerinnen und Schülern (SuS) der Sekundarstufe II die Möglichkeit bereits vor Studienbeginn experimentell zu erfahren, warum und inwiefern die Mathematik in der Informatik eine wesentliche Rolle spielt. Im Rahmen eines sechsständigen Moduls werden unter anderem die Inhalte Objektdarstellung mittels Polygonnetzen, Transformation von Objekten, Winkelbestimmung zwischen zwei Ebenen, RGBA-Farbsystem, Phong-Beleuchtung und 3D-Stereosehen von den SuS selbstständig erforscht. Im Folgenden wird aufbauend auf der Motivation für die Thematik Computergrafik für SuS und der Analyse der theoretischen Grundlagen das Modul insgesamt dargestellt und anschließend die didaktische und methodische Ausgestaltung detailliert beleuchtet.

## 1 Einleitung und Motivation

Die Antwort auf die Frage, was genau ein Informatikstudium umfasst, fällt vielen SuS auch kurz vor dem (Fach-)Abitur noch schwer (vgl. [ELK10], [He11]). Darüber hinaus haben die wenigsten Abiturienten und auch nicht alle Studienanfänger eine Vorstellung davon, warum der Mathematik gerade in den Grundvorlesungen eine so große Bedeutung zukommt (vgl. [MW06]). Um diesen und anderen Fehlvorstellungen frühzeitig entgegen zu wirken, bietet das InfoSphere – Schülerlabor Informatik<sup>1</sup> der RWTH Aachen für SuS der Sekundarstufe II das Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ an. Das Thema Computergrafik eignet sich hervorragend, um die SuS in ihrer Lebenswirklichkeit abzuholen und ihnen gleichzeitig einen realistischen Einblick in ein Informatikstudium zu ermöglichen.

Computergrafik findet man heute überall, so zum Beispiel bei vielen Spezialeffekten im Kino, der eingeblendeten Wetterkarte in den Nachrichten oder auch bei animierten

---

<sup>1</sup> <http://schuelerlabor.informatik.rwth-aachen.de>

Hintergründen im Smartphone. Neben dem motivierenden Alltagsbezug ermöglicht es die Computergrafik auf anschauliche Weise die starke Verknüpfung der Mathematik mit der Informatik bewusst zu machen.

## 2 Hintergründe und Related Work

Auch 2013 ist das Schulfach Informatik an weiterführenden Schulen in NRW immer noch stark unterrepräsentiert und wird höchstens in Form von ITG-Kursen (Informationstechnische Grundbildung) in der Unterstufe, Wahlpflichtkursen in der Mittelstufe und als Grundkurs (selten auch als Leistungskurs) in der Oberstufe angeboten (siehe [Re10]). Diese Situation erfordert es, SuS auch außerhalb des Schulunterrichts auf diese Fachwissenschaft aufmerksam zu machen, da der Informatik heute und in Zukunft eine große Bedeutung zukommt. Dazu wurde 2010 an der RWTH Aachen das InfoSphere – Schülerlabor Informatik gegründet. Im Rahmen dieses außerschulischen Lernorts werden aktuell 23 verschiedene Module für SuS aller Schulstufen angeboten, die den Kindern und Jugendlichen einen ersten oder auch vertiefenden Einblick in die Welt der Informatik ermöglichen. Das Lernen an außerschulischen Lernorten ist bereits seit 1994 Thema der fachdidaktischen Forschung (siehe [Gr94], [Eu05] und [Gu07]). In diesen Arbeiten und weiteren informatisch-didaktischen Forschungen (siehe [Be06], [En04] und [GP08]) kristallisierten sich das aktive, selbstgesteuerte und selbstentdeckende Lernen als eine erfolgreiche Methode für das Informatiklernen heraus, nicht nur in außerschulischen Lernorten. Für eine tiefere Analyse der didaktischen Ausgestaltung des InfoSphere sei auf [BHS12] verwiesen. Die Leitidee des selbstentdeckenden Lernens verfolgt auch das hier beschriebene Modul zur Einführung in die Computergrafik.

Neben der fachdidaktischen Forschung gibt es bereits verschiedene Ansätze das Thema Computergrafik für SuS didaktisch aufzuarbeiten, um erste Schritte auf diesem Feld zu ermöglichen und bei den Lernern ein grundlegendes Verständnis zu entwickeln. Als einen Ansatz ist als erstes die Website NeHe<sup>2</sup> zu nennen, welche mehrere sehr umfangreiche Tutorials zum Selbststudium der plattformunabhängigen Programmiersprache OpenGL Shading Language anbietet. Da diese Website nicht nur Einblicke gewährt, sondern mehrere Wochen umfassende Kurse anbietet, eignet sie sich sehr gut, interessierten SuS nach dem Modul eine eigenständige Vertiefungsmöglichkeit zu ermöglichen und wird daher gerne weiter empfohlen. Des Weiteren gibt es eine von Gabriel Zachmann entwickelte Ideensammlung „Computergrafik in Schule“<sup>3</sup>, deren Inhalte ebenfalls weit über die des Moduls hinausgehen. Da die Ideensammlung jedoch nur Inhalte und keinerlei didaktische Konzepte beinhaltet, ist diese höchstens engagierten Lehrkräften für den Aufbau einer eigenen Unterrichtssequenz zu empfehlen. Darüber hinaus existiert mit dem Shader-Maker (siehe [Un12]) bereits ein Programm zur Erstellung und Testung von Shaderprogrammen, welches als Vorbild des hier verwendeten Shadermanagers (für detaillierte Beschreibungen siehe Abschnitt 4.2 Didaktische und methodische Entscheidungen) diene.

---

<sup>2</sup> <http://nehe.gamedev.net/>

<sup>3</sup> [http://zach.in.tu-clausthal.de/cg\\_in\\_schule/](http://zach.in.tu-clausthal.de/cg_in_schule/)

## **3 Überblick über das Modul**

### **3.1 Rahmenbedingungen und Vorwissen**

Da dieses Modul insbesondere Studieninteressierten verdeutlichen soll welche Rolle die Mathematik in der Fachdisziplin Informatik und damit auch im Informatikstudium spielt, richtet es sich vorrangig an SuS des zweiten und dritten Oberstufenjahres (entspricht den Schulstufen Q1 und Q2 in NRW). Der zeitliche Umfang dieses Moduls liegt bei 6 Zeitstunden inkl. Pausen, was jedoch abhängig vom Vorwissen und Können der Lerngruppe variieren kann.

Entsprechend der Zielsetzung setzt das Modul neben grundlegenden Programmierkenntnissen (bspw. in Java) auch Vorwissen aus der Mathematik voraus. Dies betrifft insbesondere die Vektorrechnung (u.a. Bestimmung von Normalenvektoren, Berechnung von Skalarprodukten) und den sicheren Umgang mit Matrizen (u.a. Multiplikation von Matrizen). Da diese informatischen und mathematischen Vorkenntnisse nur auf niedrigem Niveau benötigt werden, sind sie bereits durch Grundkurse abgedeckt.

### **3.2 Inhalte des Moduls**

Das Modul bietet einen ersten Einstieg in das weite Feld der Computergrafik. Ohne allzu weit in die Tiefe einzudringen, ist es dennoch möglich, dass am Ende des Moduls ein echtes, vollständiges Ergebnis in Form einer animierten Szene entsteht, welches im Anschluss weiter verfeinert, vertieft oder ausgebaut werden kann. Folgende Themen werden (in dieser Reihenfolge) behandelt:

1. Objektdarstellung mittels Polygonnetzen
2. Transformation der erstellten Objekte mittels Matrix-Vektor-Rechnung
3. Winkelbestimmung zwischen zwei Ebenen über das Skalarprodukt der Normalenvektoren
4. RGBA-Farbsystem
5. Phong-Beleuchtung
6. 3D-Stereosehen

Im Folgenden wird das gesamte Modul inklusive Ablauf, Materialien und Software detailliert beschrieben und wichtige didaktische Entscheidungen werden dargelegt.

## **4 Ausgestaltung des Moduls**

### **4.1 Ablauf**

Ziel des Moduls ist eine animierte Szene, die für alle Teilnehmergruppen individuell unterschiedlich und somit einzigartig ist. Zunächst werden Objekte, abstrahiert durch Polygonnetze, mathematisch mittels der Vektoren ihrer Eckpunkte dargestellt und mit

Hilfe eines Steckbrettes und Gummibändern sichtbar gemacht. Daran anknüpfend lernen die SuS verschiedene Möglichkeiten der Transformation (Skalierung, Rotation und Translation) dieser (auch 3-dimensionalen) Objekte mit Hilfe eines speziell dazu entwickelten Applets kennen. Nachdem die Grundlagen in einem Sicherungsblatt gefestigt wurden, erhalten die SuS die Möglichkeit, mit dem Programm „Shadermanager“ erste kleine Vertexshader (Programme zur Veränderung der Form, Farbe und Position der Eckpunkte eines Objekts) zu erstellen. Neben der Farbgebung, die mittels einer kleinen elektrischen Schaltung erlernt wird, animieren die SuS das Objekt mittels den zuvor erlernten Transformationen. Nach Abschluss des Themas Vertexshader folgt die Erarbeitung des Fragmentshaders (Programm zur Veränderung der Oberfläche der Objekte). Um die Objekte möglichst realistisch darzustellen, erwerben die Teilnehmenden ein Grundwissen über lokale Beleuchtung, welches anschließend im Fragmentshader praktisch umgesetzt wird. Darauf aufbauend haben die SuS Gelegenheit, ihre Szene nach eigenen Vorstellungen weiter auszubauen und zu verfeinern. Als Exkurs besteht darüber hinaus die Möglichkeit, auch 3D-Effekte kennenzulernen. Zum Abschluss des Moduls können die Teilnehmenden ihre individuellen Ergebnisse präsentieren und über ihre Umsetzung diskutieren. Damit erhalten die SuS einen umfassenden ersten Einblick in die Welt der Shaderprogrammierung, als einen zentralen Bereich der Computergrafik.

Der zeitliche Ablauf inklusive der jeweiligen Lernmethode und der verwendeten Medien bzw. Materialien kann der folgenden Tabelle entnommen werden.

Zeit	Inhalt	Methode	Medien/Materialien
0.00	Begrüßung und Motivation	Präsentation	Smartboard
0.10	Einstieg in Phase 1: Objektdarstellung	Partnerarbeit	Steckbrett, Arbeitsblatt 1
0.40	Phase 1: Koordinatentransformation	Partnerarbeit	Applet „Koordinatentransformation“
1.20	Sicherung der 1. Phase	Partnerarbeit	Arbeitsblatt 2
1.40	Einstieg in Phase 2: Präsentation des Programms: Shadermanager	Präsentation	Smartboard
2.00	Phase 2: Animation durch Koordinatentransformationen	Partnerarbeit	Arbeitsblatt 3, Programm „Shadermanager“
2.40	PAUSE	----	----
3.00	Phase 3: RGD-Farbdarstellung	Partnerarbeit	Arbeitsblatt 4, elektr. Schaltung
3.20	Phase 4: Comiclook	Partnerarbeit	Arbeitsblatt 5
4.00	PAUSE	----	----
4.15	Phase 5: Lokale Beleuchtung	Partnerarbeit	Arbeitsblatt 6
4.50	Exkurs: 3D-Stereo	Partnerarbeit	Arbeitsblatt 7, rot-türkis-Brille
5.30	Abschluss: Vorstellung der Ergebnisse	SuS-Präsentation	Smartboard
6:00	ENDE	----	----

## 4.2 Didaktische und methodische Entscheidungen

Das gesamte Modul wurde der Leitidee des InfoSphere folgend nach dem Prinzip des selbstentdeckenden Lernens gestaltet (vgl. [BHS12]). Dies bedeutet, dass insbesondere die grundlegenden Themen Objektdarstellung, Koordinatentransformation und RGB-Farbdarstellung auf anschauliche und experimentelle Art und Weise von den Lernern selbst entdeckt werden. Für den Einstieg in die Objektdarstellung wird zur Veranschaulichung ein einfaches Steckbrett mit Gummibändern genutzt; zur Übertragung auf 3-dimensionale Objekte und deren Transformationen dient ein Applet, welches speziell auf die Bedürfnisse der Zielgruppe zugeschnitten ist; und die RGB-Farbdarstellung können die SuS lernerzentriert mittels einer elektronischen Schaltung selbst erfahren.

Ein weiteres grundsätzliches Merkmal der Phasen 2 bis 5 ist das Prinzip der abnehmenden Hilfestellung und schrittweisen Öffnung der Arbeitsaufträge (vgl. [MN11]). Die relativ kleinschrittige Anleitung in Phase 2 stellt sicher, dass alle SuS mit ersten Erfolgserlebnissen in der Shaderprogrammierung starten. Anschließend bieten die Aufträge immer mehr Spielraum zur Entfaltung der eigenen Ideen.

Im Folgenden werden die wichtigsten didaktischen und methodischen Entscheidungen der einzelnen Phasen genauer erläutert.

### *Einstieg in Phase 1: Objektdarstellung mittels Polygonnetzen*

Als Einstieg in das Modul entdecken die SuS selbstständig den Zusammenhang zwischen einfachen Strich-Grafiken (Polygonnetzen) und den zugrundeliegenden mathematischen Konstrukten (Vektoren und Geraden), indem sie einfache Dreiecksflächen mithilfe gegebener Vektoren auf einem Nagelbrett abstecken (siehe dazu Abbildung 1). Diese recht einfache praktische Aufgabe ermöglicht es den SuS im Modul anzukommen, was wichtig für weitere Lernphasen ist, da die Moduldurchführungen außerhalb des regulären Schulunterrichts stattfinden (siehe [Gr94]).



Abbildung 1: Aufgabe zur Objektdarstellung und zugehörige Lösung

Zur direkten Sicherung der Vektordarstellung können sich die SuS nun eigene Objekte ausdenken, durch Vektoren darstellen und diese mit dem Nachbarsteam austauschen. Mittels solcher Austausch-Aufgaben ist es möglich, die Kontrolle ganz in die Hand der

Lerner zu legen, welche so selbst feststellen, ob die Vektordarstellung verstanden wurde oder ob Fehler unterlaufen sind. Dieses Vorgehen ermöglicht es den SuS, die Betreuer des Moduls von Beginn an als Unterstützer und nicht als Lehrer wahrzunehmen.

Direkt im Anschluss an diese erste praktische Aufgabe erfolgt der Übergang zur Transformation (erst einmal nur die Skalierung) solcher Objekte. Dies geschieht indem diese zunächst per Hand auf dem Nagelbrett vergrößert werden, dann die neuen Vektoren abgelesen werden und zum Schluss die entsprechende Übergangsmatrix schrittweise zusammengesetzt wird. Auf diese Art und Weise können die Lerner die Transformationsmatrix selbstständig erarbeiten, was wesentlich motivierender und somit auch effektiver ist, als nur gegebene Informationen aufzunehmen (vgl. [Gu07]).

### *Phase 1: Koordinatentransformation*

Nachdem die Vergrößerung von Grafiken per Hand auf dem Steckbrett erlernt wurde, werden nun auch die anderen Arten der Transformation (Rotation und Translation) mithilfe eines zu diesem Zweck entwickelten Applets (siehe dazu Abbildung 2) erforscht. Auch hier verfolgt das Modul wieder die Leitidee des selbstentdeckenden Lernens, indem die Aufgabenstellungen (siehe grauer Block links) absichtlich sehr kurz gehalten sind und das Applet viel Raum zum Ausprobieren bietet. Die Lerner haben die Möglichkeit, die Matrixeinträge (oben links) zu verändern und die resultierenden Effekte direkt in der Animationsfläche (rechts) zu sehen oder auch die Grafik direkt per Maus zu verändern, woraufhin die Einträge der Matrix automatisch angepasst werden. Neben dem freien Ausprobieren bietet das Applet insgesamt 13 kleine Aufgaben zur Transformation 2- und 3-dimensionaler Objekte. Für jede Aufgabe gibt es einen Fortschrittsbalken, der den SuS sichtbar macht, wie weit sie noch vom Ziel entfernt sind, und nach Abgabe der Lösung ein Feedback (siehe Box). Um auf die zu Beginn erwähnte große Heterogenität von Informatik-Kursen zu reagieren, steht es den SuS frei, wie viele der vorgegebenen Aufgaben sie absolvieren bzw. inwieweit sie frei ausprobieren möchten. Da jedoch nach dieser Phase ein bestimmtes Mindestwissen bei allen SuS vorhanden sein muss, um das Modul sinnvoll fortzusetzen, folgt eine explizite Sicherungsphase.

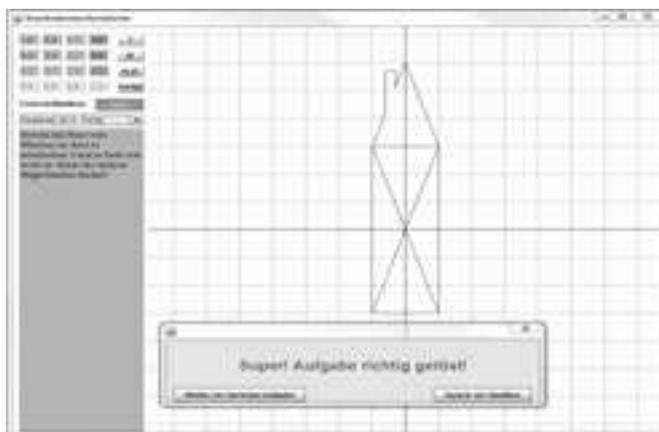


Abbildung 2: Applet "Koordinatentransformation" mit Feedback

### *Sicherung der 1. Phase*

In dieser Phase wird von den Lernern zunächst in Partnerarbeit ein Sicherungsblatt bearbeitet. Dabei ist zu einer bestimmten textuell beschriebenen Transformation die entsprechende Matrix aufzustellen und auch aus einer gegebenen Matrix die entsprechenden Transformationen abzuleiten. In der letzten Aufgabe werden die Matrizen für Skalierung, Rotation und Translation noch einmal allgemein festgehalten (die Musterlösung ist in Abbildung 3 dargestellt). Dies fördert zum einen das abstrakte Denken der SuS und soll zum anderen als Nachschlagemöglichkeit bei der späteren Programmierung dienen. Anschließend werden die Ergebnisse im Plenum besprochen, um sicherzustellen, dass die Grundlagen von allen SuS korrekt verstanden wurden.

$$S = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Abbildung 3: Allgemeine Transformationsmatrizen für Skalierung (S), Rotation um die z-Achse (R) und Translation (T)

### *Einstieg in Phase 2: Präsentation des Programms „Shadermanager“*

Nachdem in der ersten Phase die Grundlagen der Objektdarstellung und Transformation inklusive der entsprechenden mathematischen Methoden gelegt wurden, geht es in Phase 2 um die Programmierung dieser Effekte. Speziell für diese Phase wurde das Programm „Shadermanager“ nach dem Vorbild des Shader-Maker (siehe Abschnitt 2 ) entwickelt, welches ohne tiefgehendes Wissen die Erstellung einfacher Grafikprogramme ermöglicht. Der Shadermanager stellt bewusst nur eine Teilmenge der Funktionalitäten des Shader-Makers zur Verfügung, punktet dabei allerdings durch ein aufgeräumtes Design, einfache Bedienung, umfangreiche Hilfe und verbesserte Fehlerausgaben, so dass die Lernenden ohne ablenkende Einstellungsmöglichkeiten ihre ersten eigenen Programme im Shadermanager implementieren können. Zur Unterstützung erfolgt an dieser Stelle eine kurze Präsentation der Software und der nötigen Grundlagen in der Programmiersprache OpenGL Shading Language durch die Betreuer.

### *Phase 2: Animation durch Koordinatentransformationen*

In dieser Phase geht es nun darum, die im Einstieg erlernten Kenntnisse selbstständig an einfachen Grafikprogrammen zu erproben und zu festigen. Wie bereits oben erwähnt, sind die Arbeitsaufträge in dieser Phase recht konkret und es gibt einige Tipps und Hilfen, um allen SuS einen erfolgreichen Start in die Shaderprogrammierung zu ermöglichen. So erlernen die SuS schrittweise die Gestaltung erster Animationen von Objekten. Dabei finden immer wieder Verweise zur (Schul-)Mathematik statt, damit die Lerner das neuerworbene Wissen in bekannte Strukturen integrieren können. Auch in dieser ersten Phase der Shaderprogrammierung können die SuS sich bereits kreativ ausleben, da nicht festgelegt ist, welches Objekt animiert werden soll und wie die Animation exakt aussehen soll. Dies ermöglicht es auf das individuelle Lerntempo der

SuS einzugehen, da die Animationen beliebig weit ausgebaut und verfeinert werden können.

### Phase 3: RGB-Farbdarstellung

Um den SuS auch die freie Farbgestaltung ihrer Objekte zu ermöglichen, erfolgt an dieser Stelle ein Exkurs in die RGB-Farbdarstellung. Dazu steht eine kleine elektronische Schaltung aus einer 9V-Blockbatterie, einer RGB-LED mit Vorwiderstand und drei Potentiometern (siehe Abbildung 4) zur Verfügung. Die Einstellung der Potentiometer (von 0=aus bis 1=voll) bildet dabei eine Analogie zu den RGB-Farbwerten (jeweils 0-255). Somit erlernen die Teilnehmer allein durch Experimentieren wie die Rot-, Grün- und Blauanteile zu einer Mischfarbe zusammen spielen.

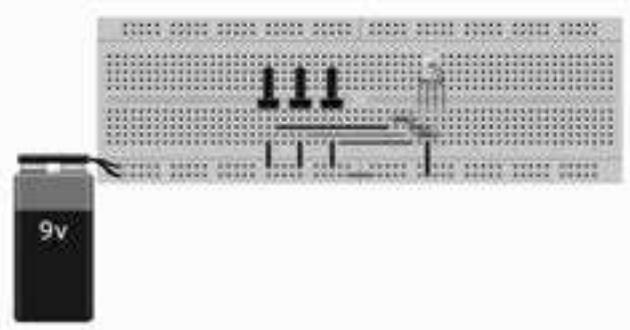


Abbildung 4: Elektrische Schaltung zur RGB-Farbdarstellung

### Phase 4: Comiclook

Diese Phase unterstreicht besonders gut die Verknüpfung der Mathematik und Informatik im Themenbereich Computergrafik, da zur Erzeugung des sogenannten Comiclooks (siehe Abbildung 5) die Berechnung des Winkels zwischen der Blickrichtung und der Oberfläche des Objekts (Skalarprodukt der Richtungs- bzw. Normalenvektoren) notwendig ist, um dann den entsprechenden Teil des Objekts je nach Lage zur Blickrichtung unterschiedlich dunkel einzufärben. Dies kann selbstverständlich nicht nur in Grautönen, sondern auch in beliebigen Farbverläufen umgesetzt werden, indem die SuS das in der vorhergehenden Phase erworbene Wissen über die RGB-Farbdarstellung einsetzen.

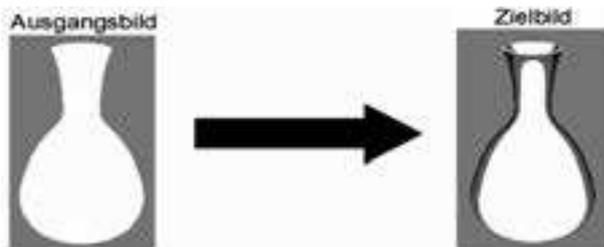


Abbildung 5: Veränderung durch Comiclook

### *Phase 5: Lokale Beleuchtung*

In der letzten obligatorischen Phase geht es um einen besonderen Beleuchtungseffekt, die Phong-Beleuchtung. Dieses Thema eignet sich besonders gut als Abschluss, da das erlernte Wissen über die RGB-Farbdarstellung aus Phase 3 und die Winkelberechnung aus Phase 4 noch einmal in neuem Zusammenhang vertieft wird. Hier stellt das Aufgabenblatt lediglich einen groben Leitfaden für eine mögliche Herangehensweise zur Verfügung, der gleichzeitig den schwächeren Lernern noch etwas Sicherheit gibt und den schnelleren SuS viele Freiheiten aufzeigt. Auch hier wird noch einmal die Notwendigkeit mathematischer Grundlagen für die informatische Umsetzung deutlich.

#### *Exkurs: 3D-Stereo*

Je nach Schülergruppe kann dieser Exkurs von der gesamten Gruppe oder auch von einzelnen besonders interessierten und motivierten SuS als Einblick in das Themengebiet 3D-Effekte angeschlossen werden. In dieser Phase erzielen die SuS einen 3D-Tiefeneffekt, indem sie ein und dasselbe Objekt in zwei verschiedenen Farben (Rot und Türkis) leicht versetzt auf dem Bildschirm ausgeben. Dieses wird anschließend für die Person vor dem Bildschirm mit Hilfe einer Rot-Türkis-Brille als 3-dimensionales Objekt sichtbar.

## **5 Fazit**

Mit dem Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ kann das InfoSphere – Schülerlabor Informatik zukünftig allen Studieninteressierten einen realistischen, zielgruppengerechten und experimentellen Einblick in die Welt der Informatik und damit verbunden der Verknüpfung von Mathematik und Informatik, geben.

Während der Entwicklung wurde das Modul mit zwei Lerngruppen (einem Informatik-Kurs der Einführungsphase und einzelnen interessierten Schülerinnen beim Schnupperstudium der RWTH Aachen) testweise durchgeführt. Diese Durchführungen dienten der formativen Evaluation, Verbesserung der Arbeitsmaterialien und letzten Anpassungen an die Zielgruppe.

Weitere Erfahrungen werden wir am 1. März 2013 bei der Durchführung mit einem Informatik-Leistungskurs und während der Schüleruni 2013 in den Sommerferien sammeln. Durch die in diesem Rahmen stattfindende quantitative Evaluation wird es uns möglich sein, auch den Erfolg des Moduls zu messen.

## Literaturverzeichnis

- [Be06] Bell, T.: Schülervorstellungen und Lernen von Physik. Forschendes Lernen, Kiel, 2006.
- [BHS12] Bergner, N.; Holz, J.; Schroeder, U.: Concept of an Extracurricular Learning Environment for Computer Science. In (Knobelsdorf, M.; Romeike, R. Hrsg.): PRE-PROCEEDINGS 7th Workshop in Primary and Secondary Computing Education WiPSCE 2012, 2012; S. 26–33.
- [ELK10] Engeser, S.; Limbert, N.; Kehr, H.: Studienwahl Informatik. Abschlussbericht zur Untersuchung, München, 2010.
- [En04] Engeln, K.: Schülerlabors. authentische, aktivierende Lernumgebungen als Möglichkeit, Interesse an Naturwissenschaften und Technik zu wecken. Dissertation. Logos-Verl, Kiel, 2004.
- [Eu05] Euler, M.: Schülerinnen und Schüler als Forscher: Informelles Lernen im Schülerlabor. In Naturwissenschaften im Unterricht. Physik, 2005, 16; S. 4–12.
- [GP08] Guderian, P.; Priemer, B.: Interessenförderung durch Schülerlaborbesuche, 2008.
- [Gr94] Griffin, J.: Learning to learn in informal science settings. In Research in Science Education, 1994, 24; S. 121–128.
- [Gu07] Guderian, P.: Wirksamkeitsanalyse außerschulischer Lernorte. Der Einfluss mehrmaliger Besuche eines Schülerlabors auf die Entwicklung des Interesses an Physik. Dissertation, Berlin, 2007.
- [He11] Hechenberger, S.: Das informatische Weltbild von Studienanfängern der Informatik. Empirische Untersuchung unter Absolventen der letzten G9-Jahrgänge in Bayern, München, 2011.
- [MN11] MNU Deutscher Verein zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts e.V. Hrsg.: 033.068 MNU BS 2011 X3.indd. Aufgaben im naturwissenschaftlichen Unterricht. Lernen – Prüfen - Diagnostizieren. Verlag Klaus Seeberger, Neuss, 2011.
- [MW06] Maass, S.; Wiesner, H.: Programmieren, Mathe und ein bisschen Hardware ... Wen lockt dies Bild der Informatik? In Informatik-Spektrum, 2006, 29; S. 125–132.
- [Re10] Reher, J.: Dossier über die Informatik in der allgemeinbildenden Schule für die Bundesländer Nordrhein-Westfalen und Niedersachsen. Dossier, Potsdam, 2010.
- [Un12] Universität Bremen: Computer Graphics - Teaching. [http://cgvr.cs.uni-bremen.de/teaching/shader\\_maker/index.shtml](http://cgvr.cs.uni-bremen.de/teaching/shader_maker/index.shtml), 06.02.2013.