

Gestensteuerung mit 3D-Kameradaten

Projektunterricht in der IT-Fachinformatiker/innen-Ausbildung

LARS STUCKMANN & PEER STECHERT

Wie werden wir zukünftig mit den Computern interagieren? Eine Perspektive auf diese Frage steht im Mittelpunkt eines Programmier-Projektes in der IT-Fachinformatiker/innen-Ausbildung am Regionalen Berufsbildungszentrum (RBZ) Technik Kiel. Neben Maus und Tastatur am PC, Wisch-Gesten auf Smartphone und Tablet sowie Spracheingabe kennen wir aus Science-Fiction-Filmen und -Serien auch schon andere Möglichkeiten zur Steuerung von Computern: Mit Gesten in der Luft werden Hologramme und projizierte Inhalte beiseite gewischt, gezoomt und manipuliert. Zur Erkennung solcher Gesten werden 3D-Informationen benötigt, die eine Tiefenkamera wie die Kinect liefert.

1 Problemstellung

In diesem Artikel wird eine zweiwöchige Projektphase beschrieben, in der sich IT-Fachinformatiker/innen im zweiten Ausbildungsjahr in die Programmierung der 3D-Kamera Kinect einarbeiten (Abbildung 1).

Die IT-Fachinformatiker/innen und IT-Systemelektroniker/innen werden im Lernfeld 6, Entwickeln und Bereitstellen von Anwendungssystemen, in den Konzepten von objektorientierter Programmierung unterrichtet (KMK, 1997). Speziell geschieht dies am RBZ Technik in der .NET Programmierumgebung von Microsoft mit der Programmiersprache C#. Dabei werden Anwendungen meist von den Schüler/innen von Grund auf neu konzipiert und entwickelt.

In der Praxis werden Anwendungen jedoch in den seltensten Fällen von Grund auf neu programmiert, sondern es wird auf bestehende Algorithmen zurückgegriffen. So kann auf der Arbeit von anderen aufgebaut werden und größere und komplexere Software entstehen. Gerade bei der Ansteuerung von Hardware greift man in der Regel auf vom Hardwarehersteller kommende Treiberpakete und Bibliotheken zurück.

Die Ansteuerung von Hardware ist für einen Großteil der Fachinformatiker unabdingbar: In beinahe jeder App für Smartphones wird die gegebene Peripherie verwendet, um die Anforderung der Applikation zu erfüllen. Darunter fällt die Nutzung der Kamera, des GPS-Sensors, der Lage- und Beschleunigungssensoren und vieles mehr. Aber nicht nur in der App-Entwick-

lung wird Hardware benutzt. In vielen Produktionsstätten mit automatisierten Fertigungsstationen müssen Programme zur Fertigung geschrieben werden, welche zwangsläufig die vorhandene Hardware wie Roboterarme ansprechen müssen.

In dem vorliegenden Projekt arbeiten die Schüler/innen mit hochaktueller Technik, welche die meisten aus dem privaten Umfeld her kennen. Sie werden die Programmiersprache C++ statt C# nutzen, welche sie somit erst erlernen müssen. Das eigene Erstellen eines großen Softwareprojektes mit Einbindung aller nötigen Bibliotheken bis hin zur Fertigstellung und Testen des Programms wird nach dem Konzept der vollständigen Handlung geübt.

2 Unterrichtsgegenstand und Rahmenbedingungen

2.1 Curriculare Vorgaben

Im Laufe des zweiten Ausbildungsjahres (Mittelstufe) wird die im Jahr zuvor eingeführte objektorientierte Programmierung anhand der Entwicklung von grafischen Oberflächen mit Trennung von Programmlogik und Programmvisualisierung weiter vertieft. Außerdem werden Grundlagen des relationalen Datenbankmodells behandelt und ihre Nutzung und Einbindung anhand von MySQL und PHP/HTML geübt. Diese Inhalte sind nach dem schulinternen Curriculum für alle IT-Berufe vorgegeben und deckungsgleich.

2.2 Unterrichtliche Voraussetzungen

In den letzten zwei Wochen der Mittelstufe findet für alle IT-Schüler/innen ein fächerübergreifendes IT-Projekt statt. Der Stundenplan wird für diese Zeit außer Kraft gesetzt, der Klassenverband aufgelöst und die Schüler/innen arbeiten in Projektgruppen ganztägig an einem zuvor gewählten Projekt. Angeboten werden dabei



Abb. 1. Die Kinect-Kamera

Projekte mit inhaltlichem Bezug zu den jeweiligen IT-Ausbildungsgängen (Systemintegration, Anwendungsentwicklung und Systemelektronik). Die Schüler/innen können aus den Angeboten ein beliebiges Projekt auswählen, wobei in der Regel dem Ausbildungsgang entsprechende Projekte gewählt werden.

Für die Motivation der Schüler/innen und unter Berücksichtigung der verschiedenen Leistungsniveaus und Ausbildungsgänge werden mit den verschiedenen Arbeitsgruppen unterschiedliche Projektziele vereinbart, sodass alle Teilnehmer/innen gefordert, aber nicht überfordert werden. Gewisse Grundlagen müssen jedoch von allen erfüllt werden.

2.3 Einbindung der Einheit in den laufenden Unterricht

Im Kontext des laufenden Unterrichts der verschiedenen Klassen wird eine Vertiefung des bisher vorhandenen Verständnisses für objektorientierte Programmierung angestrebt. Des Weiteren soll eine Transferleistung erfolgen, bei der die Schüler/innen die ihnen bekannten Konzepte auf eine neue Programmiersprache übertragen. Ziel dabei ist es, dass die Schüler/innen erkennen, dass die objektorientierten Prinzipien allen modernen Sprachumgebungen zugrunde liegen und dass bekannte Modellierungs- und Planungswerkzeuge auch in weiteren Sprachumgebungen angewendet werden können, unabhängig von der konkret verwendeten Technologie. Außerdem lernen die Schüler/innen Methoden, um bereits vorhandene Softwarebibliotheken in ihre Projekte einzubinden.

2.4 Gegenstand des Unterrichts

Für das Erreichen der gesteckten Ziele werden verschiedene Hard- und Softwarelösungen zum Einsatz kommen, die es zu einer Gesamt-Anwendung zusammenzuführen gilt. Dabei soll der Anwendungsnutzer von einer Tiefenkamera aufgenommen werden und mit seinen Gesten oder seiner Positionierung relativ zur Kamera bestimmte Ereignisse auslösen, die softwareseitig weiterverarbeitet werden.

Im Folgenden werden die einzelnen Komponenten beschrieben:

2.4.1 Objektorientierte Programmiersprache C++

Wenngleich eine der frühen objektorientierten Programmiersprachen, ist C++ auch heute noch sehr weit verbreitet. Die vergleichsweise aufwändige Speicherverwaltung und Objektadressierung ist für viele eine Einstiegshürde und auch für fortgeschrittene Programmierer sehr fehlereinladend. Seine Daseinsberechtigung zieht C++ aus seiner außerordentlich guten Performance, da der Benutzer mehr Kontrolle als bei anderen objektorientierten Programmiersprachen über das Programm besitzt. Gerade bei der Verarbeitung von Bilddaten, bei denen eine Operation auf jedes Pixel angewendet wird, spielt die Performance eine große Rolle.

Die Syntax und Semantik, also die formalen Regeln über die zulässigen Sprachelemente und die Bedeutung der Sprachelemente und der zulässigen Kombinationen von Elementen einer Programmiersprache in einem Programm, sind in C++ denen der bereits gelernten Programmiersprache C# sehr ähnlich. Alle den Schülern/innen bekannte Datentypen, Kontrollstrukturen und Prinzipien existieren auch in der jeweils

anderen Sprache, nur unter anderen Namen oder Symbolen, mit Ausnahme der oben genannten Speicherverwaltung und -adressierung.

Als integrierte Entwicklungsumgebung (IDE) kommt Microsoft Visual Studio zum Einsatz. Dies ist den Schülern/innen bereits aus der Entwicklung mit C# bekannt. Die IDE ist eine Softwarelösung, die durch Bereitstellung von Formatierungsvorlagen, Autokorrektur und Autovervollständigung, Syntaxhervorhebung, Projektmanagement, Versionskontrolle, integriertem Compiler, Werkzeugen zur Fehlersuche und vielen weiteren Werkzeugen die Entwicklung von Anwendungen erleichtert und unterstützt.

2.4.2 Microsoft Kinect V2 Sensor

Das Kernstück der von den Schülern/innen erstellten Projekte ist der Kinect V2 Sensor, welcher die Bilddaten liefert, die grundlegend für die Anwendungsfunktionalität sind.

Die Microsoft Kinect V2 kam 2013 in Verbindung mit der Microsoft XboxOne als Spielzeug auf den Markt. Kurze Zeit später, ab 2014, brachte Microsoft einen Adapter für die Nutzung am PC heraus und stellte nötige Softwarebibliotheken und Treiber zur Verfügung (Kinect SDK 2.0). Daraufhin wurde die Kinect weltweit von vielen Instituten, die sich mit der Verarbeitung von Tiefenbilddaten beschäftigen, zur Forschung verwendet. Dies ist auf den niedrigen Preis von nur ca. 150€ zurückzuführen, da Kameras mit ähnlicher Technik derzeit Anschaffungskosten von ca. 8000€ nach sich zogen.

Die Kinect ist eine Kombination aus Farb- und Tiefenkamera mit einem zusätzlichen Array an Mikrofonen. Der Tiefenmessung liegt hierbei die Time-of-Flight-Technik zugrunde.

Bei der Lichtlaufzeitmessung ist der Signalwert eines einzelnen Bildpunktes nicht proportional zur Lichtintensität der betrachteten Szene, wie es bei klassischen Bildsensoren der Fall ist, sondern proportional zur Entfernung des abgebildeten Punktes in der Szene. Das Funktionsprinzip ist in Abbildung 2 zu sehen: Time-of Flight (ToF) Sensoren emittieren einen diffusen Lichtkegel mit modulierter Intensität, beispielsweise in Form eines Sinus, welcher die betrachtete Szene beleuchtet und von dieser reflektiert wird.

Die Wellenlänge des ausgesendeten Signals liegt im Bereich des nicht sichtbaren nahen Infrarotlichts. Das reflektierte Signal wird von einem Detektor empfangen, der in seiner Struktur einem konventionellen Bildsensor ähnelt. Durch eine Korrelation des emittierten und empfangenen Lichtsignals kann eine Phasenverschiebung ermittelt werden, welche einer Distanzinformation entspricht. Hierzu werden die von dem Detektor empfangenen Photonen im photosensitiven Halbleiterbereich in Elektronen umgewandelt und entfernungsabhängig in unterschiedlichen Ladungsschaukeln getrennt. Somit stellt das resultierende Ausgangssignal eines jeden Bildpunktes eine direkte Beziehung zur eigentlichen Tiefeninformation der betrachteten Szene dar (ALBRECHT, 2007).

2.4.3 CMake

CMake4 ist ein plattformunabhängiges Programmierwerkzeug für die Entwicklung und Erstellung von Software. Mit CMake

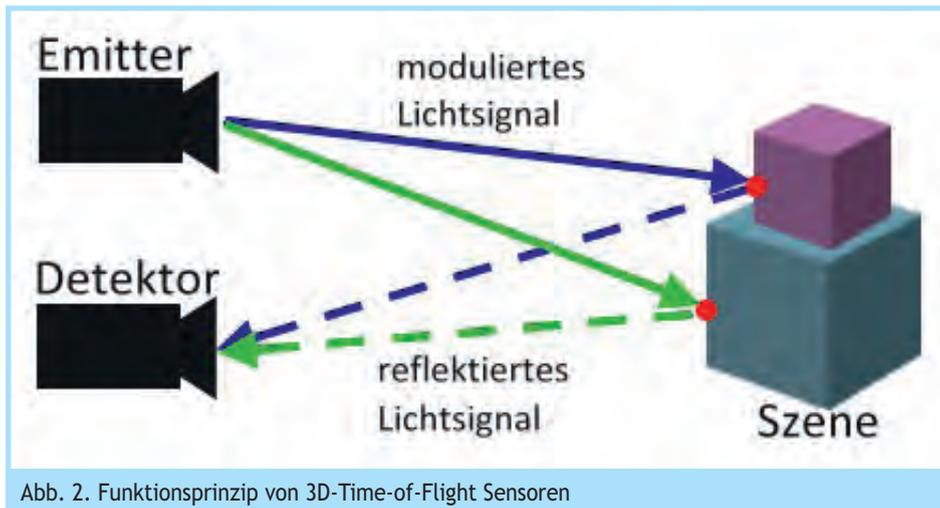


Abb. 2. Funktionsprinzip von 3D-Time-of-Flight Sensoren

Auch spiegelt hier der Projektunterricht das Modell der vollständigen Handlung wider, nach welchem die erworbenen Handlungskompetenzen im späteren Berufsleben selbständig auf andere Arbeitsprozesse übertragen werden können. Das Modell der vollständigen Handlung geht auf die von VOLPERT (1974) entwickelte Handlungsregulationstheorie zurück.

Für die Projektarbeit sind 68 Unterrichtsstunden zu je 45 Minuten vorgesehen, welche auf zwei Wochen aufgeteilt werden.

werden aus Skriptdateien (CMakeLists.txt) Makefiles und Projekte für viele integrierte Entwicklungsumgebungen und Compiler erzeugt. Makefiles werden beispielsweise genutzt, um in Projekten, die aus vielen verschiedenen Dateien mit Quellcode bestehen, automatisiert alle Arbeitsschritte (Übersetzung, Linken, Dateien kopieren etc.) zu steuern, bis hin zum fertigen, ausführbaren Programm.

2.4.4 Weitere Bibliotheken

Für die Darstellung und Bearbeitung der Bilder kann die freie Softwarebibliothek OpenCV oder alternativ das von Microsoft bereitgestellte Direct2D genutzt werden.

Für Interaktionen mit anderen Windowsfunktionen und für Zugriffe auf andere laufende Programme wird die Windows API verwendet.

3 Didaktische Entscheidungen und methodische Überlegungen

Ein wichtiger Teil zur Erreichung der gesteckten Ziele ist die Didaktische Reduktion. Wie GRÜNER (1967) schon darlegte, gelten als Prinzipien der Didaktischen Reduktion fachliche Richtigkeit, fachliche Ausbaufähigkeit und Angemessenheit. In diesem Sinne wird auf die genaue Funktionsweise von der vorliegenden Time-of-Flight Technik verzichtet und das Prinzip nur kurz vorgestellt. Auch Darstellungsformen und Umrechnungsarten zur Bestimmung einer Raumkoordinate anhand eines einfachen Tiefenwerts eines Pixels werden nur prinzipiell erläutert und nicht im Detail erklärt, da diese von bereits vorhandenen Softwarebibliotheken übernommen werden und der Fokus auf die Verwendung solcher Bibliotheken gelegt werden soll.

Im beruflichen Alltag arbeiten die IT-Fachinformatiker/innen häufig in Teams zusammen, was die Fähigkeit erfordert, sowohl eigenständig abgesprochene Arbeitspakete zu bearbeiten und Probleme zu lösen als auch kommunikativ und im respektvollen Umgang miteinander Schnittstellen abzusprechen und eine Arbeitsteilung vorzunehmen. Daher bietet sich in diesem Fall der Projektunterricht an, welcher nach GUDJONS (2001) die „Hochform“ handlungsorientierten Lehrens und Lernens ist.

3.1 Didaktische Entscheidungen

In der Informatik ist die Nutzung bereits existierender Softwarebibliotheken unverzichtbar. Die getrennte Einbindung in den Quellcode und das Linken der existierenden Dateien gegen das eigene Programm ist zu Beginn nicht intuitiv und die Funktionsweise muss erst verinnerlicht werden. Danach ist das Prinzip aber bei fast allen anderen Bibliotheken gleich.

Auch die Erstellung großer Softwareprojekte erfordert eine gewisse Grundstruktur und muss verwaltet werden. Außerdem sollen selbst programmierte Bibliotheken für andere nutzbar gemacht werden. Dies kann in Form des Quellcodes oder vorzugsweise in kompilierten DLL Dateien geschehen. Zur Erstellung und Verwaltung wird das weit verbreitete CMake genutzt. Die Projekte werden in der Programmiersprache C++ geschrieben. Dafür wurden anfangs die Unterschiede zu der ansonsten im Unterricht verwendeten Sprache C# aufgezeigt und auf weitere Besonderheiten hingewiesen. C++ ist auch heute noch die Sprache der Wahl, wenn es um die Ansteuerung von Hardware, z. B. die hier genutzten Kinect Sensoren, geht. Auch in der Bildverarbeitung wird meist C++ verwendet, da es im Gegensatz zu anderen Programmiersprachen schneller ist in Bezug auf die Laufzeit der programmierten Algorithmen.

Die Schüler/innen dürfen im Rahmen der Thematik selbst eine Projektidee entwickeln, was zur Förderung der Motivation durch eine stärkere Identifizierung mit dem Arbeitsergebnis führt. Auf diese Weise wird auch eine Binnendifferenzierung über den Inhalt vorgenommen. Alle Projekte müssen jedoch gewisse Grundlagen erfüllen und vor der Durchführung mit der Lehrkraft abgesprochen werden.

3.2 Methodische Überlegungen

Bevor mit dem Projektunterricht begonnen wird, werden in kurzen Input-Phasen Grundlagen vermittelt. Dazu zählen Unterschiede von C# zu C++, Umgang mit CMake, Verwendung des Kinect Sensors und Einbinden und Nutzung von fremden Programmibibliotheken. Zu Beginn einer Input-Phase gibt es eine Problemstellung: „Was möchte ich erreichen und warum?“. Dies geschieht im Lehrer-Schüler-Gespräch. Da zu den hier benutzten Themen wenig bis gar kein Vorwissen vorhanden ist, wird in einem kurzen Lehrervortrag geschildert, wie man das Problem lösen kann. Anschließend verschriftlichen alle

Schüler/innen die für sich wesentlichen Informationen und verwenden diese, um kleine Aufgaben zu dem entsprechenden Thema zu lösen. Danach sind die Schüler/innen in der Lage, die Herangehensweise auf ihre Projekte zu übertragen.

Nach den Input-Phasen wechselt die Lehrerrolle aus der „vorgabenden Rolle“ zur „begleitenden Rolle“, die während des gesamten Projektunterrichts beibehalten wird. Dabei greift die Lehrperson nur noch unterstützend ein oder gibt gezieltes Feedback, so wie von TSCHEKAN (2005) beschrieben.

Der Projektunterricht orientiert sich an der von FREY (1998) beschriebene Methodik.

Tabelle 1 zeigt den Ablauf des Projekts.

4 Ergebnisse

Abschließend werden kurz exemplarisch vier Projektergebnisse vorgestellt.

Spotify-Gestensteuerung: „Spotify ist ein digitaler Musikdienst, der dir Zugriff auf Millionen von Songs ermöglicht“ (<https://www.spotify.com/de/>). Dieser Dienst ist ein fester Bestandteil des privaten Lebens der meisten Schüler/innen. Aus diesem Grund hat eine Gruppe es sich zur Aufgabe gemacht, den Spotify Windows Client mit vor der Kinect ausgeführten Gesten zu steuern. Dazu haben sie sich ein Konzept zur Erkennung der Gesten überlegt und festgesetzt, welche Funktionen umgesetzt

Grundlagenphase	1h Plenum	
	Kick Off/Organisatorisches	<ul style="list-style-type: none"> • Organisatorisches • Vorstellung der Bewertungskriterien • Schüler*innenfragen • Fragebogen vor Projektbeginn
	6h Lehrervortrag, Einzelarbeit / Partnerarbeit	
	Grundlagen: Input-Phasen mit anschließender Übung	<ul style="list-style-type: none"> • Unterschiede C# / C++ • Einbinden und verwenden fremder Bibliotheken • CMake • Microsoft Kinect • Bildverarbeitung mit OpenCV • Weiteres (z. B. Windows API)
	1h Einzelarbeit, Gruppenarbeit, Plenum	
	Mögliche Arbeitsaufträge	<ul style="list-style-type: none"> • Vorstellung von Beispielprojekten • Diskussion über Möglichkeiten • Themenfindung für Projekte der Kleingruppen
Projektphase	50h Gruppenarbeit	
	Projektinitiative	<ul style="list-style-type: none"> • Projektidee • Meilensteine • Grobplanung
	Projektplanung	<ul style="list-style-type: none"> • Beschreiben der Anforderungen • Arbeitspakete und Verantwortlichkeiten • Netzplan • Testplan
	Durchführung	<ul style="list-style-type: none"> • Eigenständiges Arbeiten in Kleingruppen
	Reflexionsprozess (parallel zur Durchführung)	<ul style="list-style-type: none"> • Überprüfung der Meilensteine • Dokumentation Arbeitsprozess • Dokumentation Arbeitsergebnis
Abschluss	10h Gruppenarbeit, Plenum	
	Abnahmegespräche	<ul style="list-style-type: none"> • Vorstellung des Arbeitsergebnisses • Technisches Fachgespräch unter Beteiligung aller Gruppenmitglieder
	Präsentation	<ul style="list-style-type: none"> • Jede Gruppe stellt ihr Arbeitsergebnis und ihren Arbeitsprozess vor • ca. 20 Minuten je Gruppe
	Reflexion/Feedback	<ul style="list-style-type: none"> • Fragebogen nach dem Projekt • Diskussion im Plenum

Tab. 1. Projekttablauf

werden sollen. Man kann beispielsweise, wenn man die linke Hand zur Faust ballt mit der rechten Hand die Lautstärke verändern: die rechte Hand nach oben hebt die Lautstärke an, nach unten senkt sie. Weiterhin wurden noch Funktionen zum Vorwärtsschalten und Zurückschalten von Liedern, Start/Stop und eine Art Tastensperre umgesetzt. Angesteuert wird der Client über die Windows API und verwaltet wird er über ein System Tray im Infobereich von Windows. Abbildung 3 zeigt, wie eine gehobene Hand als Geste erkannt wird. Beachtenswert ist, dass mehrere Messpunkte an der Hand liegen und somit auch feine Hand- und Fingerbewegungen zuverlässig erkannt werden.



Abb. 3. Schematische Erkennung einer Geste

Greenscreen: Greenscreen- oder oft auch Bluescreen-Technik ermöglicht, Personen oder Objekte nachträglich vor einen Hintergrund zu setzen. Die Person oder das Objekt befindet sich dabei vor einer grünen oder blauen Wand, welche über Farberkennung mit einem Bild oder Video ausgetauscht wird. Eine Gruppe von Schüler/innen haben auf dieser Funktionsweise aufbauend statt einer Farberkennung eine Tiefenerkennung eingeführt. So wird alles, was sich hinter einem Schwellwert von beispielsweise zwei Metern befindet ausgeblendet und durch einen vorher festgelegten Hintergrund ersetzt. Abbildung 4 zeigt ein Tiefenfoto in der die Tiefeninformation mit Farben codiert ist.



Abb. 4. Tiefenfoto von nah (rot) bis fern (über grün bis violett)

Virtual Input Device „Gamecontroller“: Eingabegeräte werden unter Windows als Input Devices bezeichnet. Zu Ihnen gehören beispielsweise Maus, Tastatur, Touchscreen, Joystick, Gamecontroller und Ähnliche. Eine Gruppe hat ein solches Input Device virtuell als Gamecontroller angelegt. Dieser Controller wird über Gesten gesteuert, die vor der Kinect ausgeführt werden. Dabei lassen sich Gesten über eine Software trainieren und in einer Datenbank abspeichern. Man kann also jede bestimmte Bewegung mit den Händen, Armen, Beinen, Füßen, dem Kopf, dem Körperrumpf einzeln oder auch in Kombination zu einer Geste definieren. Über eine weitere Software sind erkannte Gesten bestimmten Aktionen auf dem Input Device zuordenbar. Zum Beispiel könnte das Ballen einer Faust den linken Mausklick simulieren oder, wie diese Gruppe zu Demonstrationszwecken erstellt hat, die Bewegung der flachen Hand als Neigung der Joystickachsen erkannt werden. Damit kann nun jede Software gesteuert werden, die auch von den realen Input Devices gesteuert werden kann. Für die Demonstration konnte man ein Rennspiel mit nur einer Hand steuern (links/rechts: lenken, hoch/unter beschleunigen/bremsen). Abbildung 5 zeigt die interne Klassifizierung einer Geste, z. B. als „Druck des Knopf 3“ auf einem Controller.

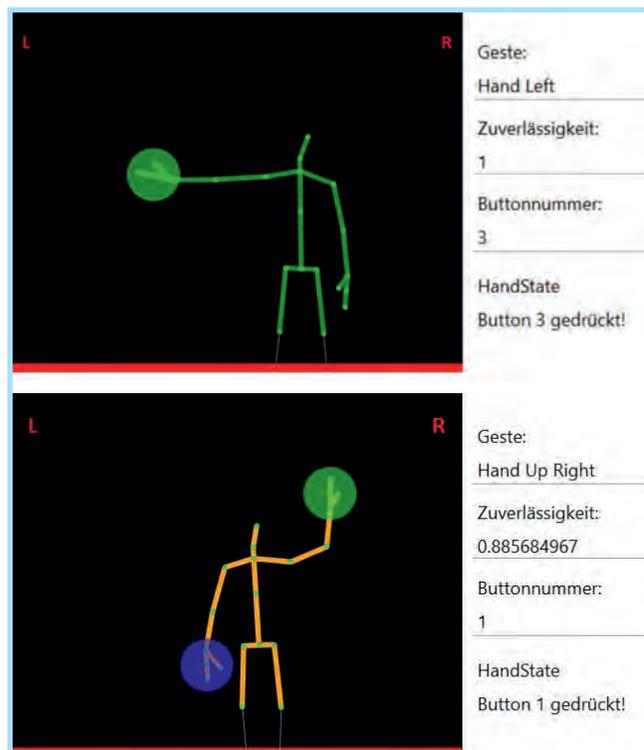


Abb. 5. Interne Klassifizierung von Gesten

Snapchat-Filter: Eine weitere Anwendung funktionierte so, dass die Gesichter von Personen automatisch erkannt und mit einem möglichst witzigen Bild überdeckt werden, z. B. Smiley oder Hasengesicht. Mit der Gestensteuerung konnte zwischen diesen Filtern umgeschaltet werden (Abb. 6).

Paint: Eine Schülergruppe hat ein Malprogramm entwickelt, das durch Gestensteuerung funktioniert (Abb. 7).



Abb. 6. Die Geste mit jeweils zwei erhobenen Fingern pro Hand blendet einen neuen Gesichts-Filter ein, in diesem Fall Smileys

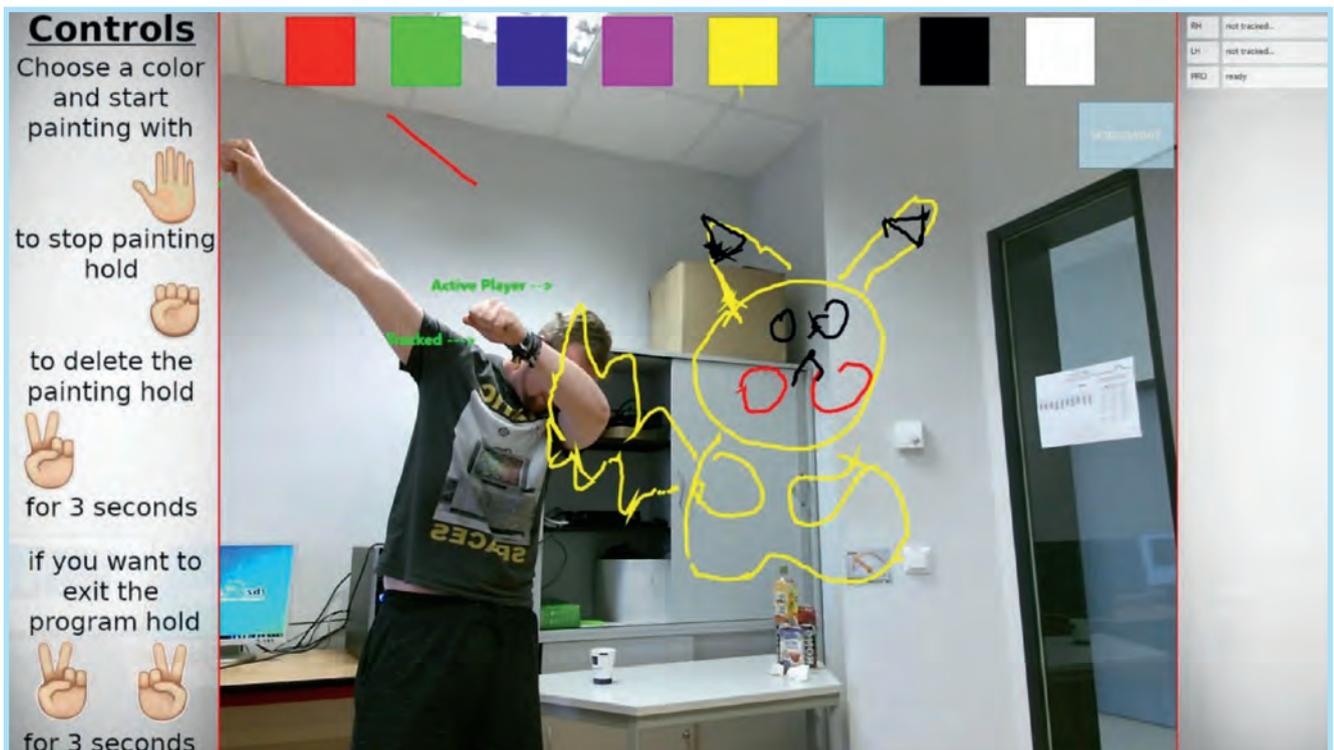


Abb. 7. Durch unterschiedliche Gesten an unterschiedlichen Positionen wird die Farbe ausgewählt und es kann gemalt werden.

5 Fazit

Es war während der gesamten Projektphase eine hohe Motivation bei allen Schülern/innen zu erkennen. Es hat ihnen sicht-

lich Spaß gemacht, die für sie neue Hardware zu benutzen, um Ereignisse in ihrer Software auszulösen. Ein Schüler hat sich sogar nach dem zweiten Projekttag privat eine Kinect gekauft, weil er in jedem Falle auch nach Projektende noch damit arbei-

ten möchte. Durch die Komplexität der Aufgabenstellung und den hohen Anspruch, den die Schüler/innen an sich selbst haben, sind zwischendurch viele Probleme aufgetreten. Doch anstatt sich davon entmutigen zu lassen, hat es die Schüler angespornt zum Teil sehr kreative Lösung zu finden. Oft haben die Schüler/innen in Gesprächen berichtet, dass sie sich auch nach dem Unterricht noch mit dem Projekt auseinandergesetzt hatten und neue Dinge ausprobiert haben. Meistens wurden die gut geplanten Meilensteine vorzeitig erreicht und selbstständig um Zusätze erweitert.

Zusammenfassend kann man sagen, dass sowohl die Schüler/innen als auch die Lehrkräfte ein breit gefächertes Spektrum an Kompetenzen während des Projektes erworben haben. Im Sommer 2019 wurde das Projekt erneut durchgeführt, wobei abweichend mit der Programmiersprache C# gearbeitet wurde.

Literaturverzeichnis

Kultusministerkonferenz (KMK) (1997). *Rahmenplan für den Ausbildungsberuf Fachinformatikerin/Fachinformatiker*. Bonn.

ALBRECHT, M. (2007). *Untersuchung von Photogate-PMD-Sensoren hinsichtlich qualifizierender Charakterisierungsparameter und -methoden*. Universität Siegen, Dissertation.

GRÜNER, G. (1967). *Die Entwicklung der höheren technischen Fachschulen im deutschen Sprachgebiet: Ein Beitrag zur historischen und zur angewandten Berufspädagogik*. Braunschweig: Westermann.

GUDJONS, H. (2001). *Handlungsorientiert lehren und lernen. Projektunterricht und Schüleraktivität*. 6. Auflage. Bad Heilbrunn: Klinkhardt.

VOLPERT, W. (1974). *Handlungsstrukturanalyse als Beitrag zur Qualifikationsforschung*, Köln: Pahl-Rugenstein.

TSCHEKAN, K. (2005). *Guter Unterricht und der Weg dorthin*. <http://www.schul-physik.de/downloads/GuterUnterricht.pdf> (18.10.2019)

FREY, K. (1998). *Die Projektmethode - Der Weg zum bildenden Tun*. 8. Auflage, Weinheim: Beltz.

LARS STUCKMANN, lars.stuckmann@rbz-technik.de, ist Lehrer für Informationstechnik und Mathematik am RBZ Technik Kiel. Seine Unterrichtsschwerpunkte sind Anwendungsentwicklung und insbesondere Computergraphik.

PEER STECHERT, peer.stechert@mnu.de, ist ebenfalls Lehrer am RBZ Technik Kiel. Am Landesseminar Berufliche Bildung in Schleswig-Holstein ist er als Studienleiter für die Ausbildung von Informatik-Lehrkräften im Vorbereitungsdienst zuständig. ■□