Unterrichtsreihe mit eingebettetem Schülerlabormodul zum Thema Medienmanipulation zur Vermittlung von Schleifen und Arrays in Java

Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung, dem Landesprüfungsamt für Erste Staatsprüfung für Lehrämter an Schulen vorgelegt von:

> Alexander Arseniev 273809

Aachen, 06.08.2014

Gutachter: Prof. Dr. Ulrik Schroeder, RWTH Aachen

Lehrstuhl: Lehr- und Forschungsgebiet Informatik 9

Learning Technologies Research Group

Betreuer: Dipl.-Gyml. Nadine Bergner

Learning Technologies LuFG Informatik 9 RWTH Aachen University



Inhaltsverzeichnis

I	Einleitung und Beschreibung der Ausgangssituation	6
1.	Einleitung und Motivation für die Lerneinheit	7
2.	Rahmenvorgaben in NRW	10
	2.1. Kernlehrplan: Übergeordnete Kompetenzerwartungen .	10
	2.1.1. Implementieren	10
	2.1.2. Argumentieren	11
	2.1.3. Kommunizieren und Kooperieren	12
	2.2. Kernlehrplan: Inhaltsfelder	12
	2.2.1. Algorithmen	12
	2.2.2. Daten und ihre Strukturierung	13
	2.2.3. Informatik, Mensch und Gesellschaft	13
	2.3. Vorgaben zum Zentralabitur	14
	2.3.1. I. Objektorientiertes Modellieren und Implemen-	
	tieren von kontextbezogen Anwendungen	14
3.	Medienmanipulation im Informatikunterricht	16
4.	Rahmenbedingungen bei der Ausarbeitung der Lerneinheit	18
	4.1. Lernziele	18
	4.2. Voraussetzungen	19
5.	Ausgangsmaterialien zum Erstellen der Lerneinheit	20
	5.1. Vorbereitung	20
	5.2. Modul	21
	5.3. Nachbereitung	22
II	Die Lerneinheit	23
6.	Die Software zur Medienmanipulation	24
	6.1. Beschreibung der Software zur Medienmanipulation	24

Inhaltsverzeichnis 3

	6.2.	Didaktische Überlegungen für den Entwurf der Medienbearbeitungssoftware	25
7.	Auf	bau der Lerneinheit	28
	7.1.	Überblick über die Lerneinheit	28
		7.1.1. Zielsetzung	28
		7.1.2. Vorbereitung im Rahmen des Unterrichts	29
		7.1.3. Modul	30
		7.1.4. Nachbereitung	31
	7.2.	Didaktische Überlegungen zum Aufbau der Lerneinheit	32
8.	Deta	aillierte didaktische Begründung	35
	8.1.		
		meine Schlussfolgerungen	35
	8.2.	Didaktische Begründung für Unterrichtsmaterialien der	
		vorbereitenden Unterrichtsreihe	36
		8.2.1. Einführung und Grafikformate	36
		8.2.2. Arbeitsblatt zu Arrays und for-Schleifen	38
		8.2.3. Test zu Arrays und for-Schleifen	41
		8.2.4. Arbeitsblatt zur RGB Farbdarstellung	42
		8.2.5. Arbeitsblatt zu zweidimensionalen Arrays und	
		Bildern	44
	8.3.	0 0	51
		8.3.1. Einführungspräsentation	51
		8.3.2. Arbeitsblatt zur Projektplanung	53
		8.3.3. Umgang mit Eclipse	55
		8.3.4. Arbeitsblatt Anweisungen für das Modul	57
		8.3.5. Die Aufgabenblätter zu den Methoden zur Bild-	
		bearbeitung	59
		8.3.6. Arbeitsblatt zum Hintergrundtausch	63
		8.3.7. Arbeitsblatt Soundbearbeitung	65
		8.3.8. Hilfsblatt zur Videobearbeitung	68
	8.4.	0 0	
		bereitung	69
		8.4.1. Unterrichtsstunde zur Glaubwürdigkeit von Me-	
		dien in der Schule	69
9.	Fazi	it	71
10	. Aus	blick	75
		raturverzeichnis	
			77
В.	Unte	errichtsmaterialien	79
C.	Mus	sterlösungen der Programmieraufgaben	143

Inhaltsverzeichnis 4

D. Quelltext der Medienmanipulationssoftware (ausgenommen Musterlösungen)156

Zusammenfassung

Im Rahmen dieser Hausarbeit wurde eine Lerneinheit entworfen, in der Schülerinnen und Schüler im ersten Jahr der Oberstufe Arrays, Schleifen sowie die Speicherung von Bildern und Tönen kennen lernen, projektartig arbeiten und über die gesellschaftlichen Auswirkungen der Informatik diskutieren. Das alles geschieht im Kontext der Medienmanipulation. Die Lerneinheit hat einen Umfang von etwa 12 Schulstunden¹ und besteht aus drei Phasen. Zunächst wird eine vorbereitende Unterrichtsreihe mit Materialien für sechs Schulstunden in der Schule durchgeführt. Hier sollen die Grundkenntnisse zu Arrays, Schleifen und Bildspeicherung vermittelt werden. Darauf folgt eine Lerneinheit für das Schülerlabor Informatik² der RWTH Aachen, die etwa 4:30 dauert. Hier wird eine Software zur Medienmanipulation entwickelt. Nach dem Besuch des Schülerlabors wird in der Schule eine Stunde lang das Thema Medienmanipulation mit Fokus auf die gesellschaftlichen Aspekte nachbereitet. Die Lerneinheit wurde für Schülerinnen und Schüler entworfen, die erst wenig Erfahrung in der Programmierung mit Java haben. Die Schülerinnen und Schüler erhalten die Möglichkeit, ein Softwareprojekt durchzuführen. Sie implementieren Methoden, die Bilder und Audiodateien in Form von Arrays bearbeiten. In diesen Methoden sollen die übergebenen Arrays umsortiert, gekürzt, erweitert oder auch Eintrag für Eintrag verändert werden. Diese werden dann in eine vorgegebene grafische Benutzeroberfläche integriert, die das Laden, Speichern, Konvertieren und Anzeigen der Dateien übernimmt. Am Ende des Projekts haben die Schülerinnen und Schüler also ein Programm zur Bearbeitung von Bildern und Audio-Dateien erstellt. Diese Hausarbeit beschreibt den Bezug der Lerneinheit zum Lehrplan in NRW, die didaktischen Begründungen für Entscheidungen bei der Ausarbeitung der Lerneinheit sowie die fertige Lerneinheit selbst.

¹ Bei Schulstunden wird in dieser Hausarbeit von einer Dauer von 45 Minuten ausgegangen.

² Das Schülerlabor trägt den Namen Infosphere und ist unter der Adresse http://schuelerlabor.informatik.rwth-aachen.de erreichbar.

Teil I

EINLEITUNG UND BESCHREIBUNG DER AUSGANGSSITUATION

Kapitel 1 Einleitung und Motivation für die Lerneinheit

Programmieren macht Spaß. Die meisten, die eine Laufbahn in Richtung Informatik eingeschlagen haben, werden dieser Aussage wahrscheinlich zustimmen. Dass die Schülerinnen und Schüler¹ innerhalb eines Informatikkurses von vorneherein genauso denken, sollte jedoch nicht als selbstverständlich vorausgesetzt werden. Motivation ist nicht zwingend aufgrund einer Sache vorhanden, sondern muss gefördert werden. Einer der Faktoren, die Hubwieser [5] zur Erzeugung von Motivation nennt, ist die Nähe zum Erfahrungsbereich der Schüler. Und genau das ist etwas, das bei dem Erstellen von Aufgaben zur Programmierung vernachlässigt werden könnte. Schüler schreiben im Unterricht dann hauptsächlich Programme, die irgendwelche mathematischen Funktionen berechnen, beispielsweise Fakultäten oder Potenzen, füllen Arrays mit zusammenhanglosen Zahlen oder irgendwelchen Namenslisten. Bei der Erstellung der in dieser Hausarbeit beschriebenen Lerneinheit² wurde dies wenn möglich vermieden und die Aufgaben mit Bezug auf einen lebensnahen Kontext entworfen. Ebenso werden den Schülern unter Umständen kaum Entscheidungsmöglichkeiten bei der Implementierung ihrer Programme gegeben, was auch ein wichtiger Faktor zur Motivierung ist [5, S.16]. Besonders zu Beginn der Oberstufe, in der die Schüler gerade eine neue Programmiersprache erlernen, vielleicht sogar überhaupt das erste Mal programmieren, kann dies sehr leicht passieren. Mit Rücksicht auf ihre geringe Erfahrung sollen die Schüler natürlich nicht überfordert werden. Dies kann jedoch in Aufgabenstellungen resultieren, deren Ziele bis ins kleinste Detail vorgegeben sind und deren Lösungswege sehr kleinschrittig angeleitet werden. Entscheidungsmöglichkeiten gibt es also in diesem Fall keine. Dies ist, wenn ein neues Thema Motivation für kontextorientierten Unterricht

¹ Im weiteren Verlauf der Hausarbeit wird lediglich "Schüler" stellvertretend für Schülerinnen und Schüler verwendet werden.

² "Lerneinheit"bezieht sich in der Hausarbeit auf die Unterrichtsreihe und das Schülerlabormodul zusammen.

erarbeitet werden soll, in vielen Fällen sinnvoll, jedoch können mit zunehmendem Wissen der Schüler auch die Aufgabenstellungen offener formuliert werden.

Wirklich problematisch ist, dass diese Faktoren, sowohl der mangelnde Bezug zum Erfahrungsbereich der Schüler als auch die nicht oder kaum vorhandenen Entscheidungsmöglichkeiten also leicht am Anfang der Einführungsphase³ auftreten können. Gerade an diesem Zeitpunkt bilden sich die Schüler eine Meinung über das Schulfach Informatik, und es werden die Grundlagen für den Informatikunterricht in der Qualifikationsphase gelegt.

Genau dann ist also das Motivieren der Schüler besonders wichtig. Die Schüler sollten erleben, dass Programmieren Spaß macht, dass es nicht nur darum geht, realitätsfremde mathematische Probleme zu lösen, sondern dass die Informatik einen reellen und direkten Bezug zur Lebenswelt der Schüler hat.

Das Grundlagenwissen kann nach konstruktivistischer Vorstellung effektiver vermittelt, und die Begeisterung für die Informatik bei den Lernenden nachhaltig gesteigert werden. Dies könnte bereits im Unterricht dazu beitragen, die Schüler für ein eventuelles Informatikstudium besser vorzubereiten, was in Anbetracht der Abbrecherquoten von über 50% [4] bei Informatikstudenten ein Ziel des Informatikunterrichts sein sollte.

Zu diesem Zweck sollten ein Modul⁴ für das Schülerlabor Informatik der RWTH und eine Unterrichtsreihe, in die dieses eingebettet wird, also eine gesamte Lerneinheit erstellt werden. Hierzu wurden ein bereits vorhandenes Modul sowie eine vorhandene Unterrichtsreihe weiterentwickelt und aufeinander abgestimmt. Der Bezug zur Lebenswelt der Schüler sollte hierbei durch den Kontext der Medienmanipulation hergestellt werden. Die Auswirkungen der Medienmanipulation sind im Alltag allgegenwärtig. Bilder, Musik und Filme, denen die Schüler im Alltag begegnen, sind zu einem großen Teil mit Computern bearbeitet.

Diese Lerneinheit sollte so konzipiert werden, dass sie am Anfang der Einführungsphase, bei dem Einstieg in die Programmierung mit Java, eingesetzt werden kann. Der Umgang mit Arrays und Schleifen soll anhand der Lerneinheit erlernt werden können. Die Schüler sollten Programme implementieren, mit denen Bilder und Töne bearbeitet werden können, die in Form von Arrays dargestellt werden. Variablen, primitive Datentypen, Wertzuweisungen und if-Verzweigungen muss den Schülern bereits vor der Lerneinheit bekannt sein, restliches dafür notwendiges Vorwissen sollte in der vorbereitenden Unterrichtsreihe behandelt werden.

Zielsetzung der Unterrichtseinheit

Die Einführungsphase bezeichnet in NRW das erste Schuljahr der Oberstufe.

⁴ Module für das Schülerlabor sind Lerneinheiten unterschiedlicher Länge, die die Schüler im Schülerlabor durchlaufen.

Da dies die Grundlage für die gesamte Lerneinheit ist, wird in Kapitel 2 zunächst auf den Kontext Medienmanipulation eingegangen, wo Informationen zu diesem Kontext bezogen werden können und warum dieser für den Informatikunterricht geeignet ist.

Inhaltliche Strukturierung der Hausarbeit

Um die Lerneinheit mit den schulischen Vorgaben abzugleichen, wird in Kapitel 3 der Bezug der Lerneinheit zu dem Kernlehrplan und den Vorgaben zum Zentralabitur hergestellt.

In der vorliegenden Hausarbeit wird außerdem beschrieben, welche Überlegungen bei dem Entwurf dieser Lerneinheit eine Rolle gespielt haben. Dazu wird im vierten Kapitel zunächst erläutert, welche Lernziele durch die gesamte Lerneinheit erreicht werden sollen. Im Anschluss daran werden kurz die Voraussetzungen zur Durchführung der Lerneinheit erläutert. Die Kompetenz- und Inhaltsbereiche des Lehrplans, die durch die gesamte Lerneinheit abgedeckt werden sollen, werden ebenfalls aufgezählt.

Anschließend wird genauer auf die Lerneinheit eingegangen. Hierzu wird in Kapitel 5 erläutert, welche Materialien zur Erstellung verwendet wurden.

Während der Durchführung im Schülerlabor verwenden die Schüler eine im Rahmen dieser Hausarbeit erstellte Software zur Medienmanipulation, in die sie ihre eigenen Methoden einbinden können. Diese Software wird in Kapitel 6 kurz erläutert sowie die didaktische Überlegungen, die bei der Implementierung der Software eine Rolle spielten.

Zusammen mit einem Überblick in Kapitel 7 über die Lerneinheit wird begründet, warum die Unterrichtsreihe so aufgebaut wurde, wie es in Kapitel 7 beschrieben wird.

Darauf folgt eine detaillierte didaktische Begründung der gesamten Unterrichtsreihe in Kapitel 8. Hierbei wird zu den einzelnen Unterrichtsmaterialien erläutert, welche Lernziele durch diese erreicht werden sollen, welche Voraussetzungen die Schüler erfüllen müssen, um diese Unterrichtsmaterialien zu bearbeiten, welche grundsätzlichen Überlegungen den Aufbau der Materialien beeinflusst haben und letztendlich wie die Materialien inhaltlich gestaltet wurden sowie die Begründung für diese Gestaltung.

Zuletzt wird ein Fazit gezogen und ein Ausblick darauf gegeben, wie die Lerneinheit weiterentwickelt werden könnte.

Kapitel 2 Rahmenvorgaben in NRW

2.1. Kernlehrplan: Übergeordnete Kompetenzerwartungen

Da die Lerneinheit in erster Linie für Schüler der zehnten Klasse gedacht ist, sollten während der Durchführung auch die in der Einführungsphase vorgesehenen Kompetenzbereiche gefördert werden. Diese können dem Kernlehrplan entnommen werden. [7] Zunächst sollen hier die übergeordneten Kompetenzerwartungen betrachtet werden.

2.1.1. Implementieren

Der erste Punkt, der hier im Kernlehrplan genannt wird, ist: "Schüler implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme". Die in dieser Lerneinheit angesprochenen und verwendeten Modelle sind in erster Linie die Modelle zur Darstellung von Bildern und Tönen in digitaler Form. Die hier von Schülern zu implementierenden Computerprogramme verwenden diese Modelle, um Bilder und Töne zu bearbeiten. In der vorbereitenden Unterrichtsreihe werden diese Modelle zur Bildbearbeitung ausführlich besprochen und allmählich entwickelt.

Behandelte Teilaspekte des Kompetenzbereichs Implementieren

Der nächste Unterpunkt zu diesem Kompetenzbereich im Kernlehrplan lautet "Die Schüler modifizieren und erweitern Computerprogramme". Dies ist in der Lerneinheit auch wiederzufinden. Die Schüler erhalten ein unfertiges Computerprogramm, welches bis dahin nur die

¹ Bei den Formulierungen des Kernlehrplans wurde "Die Schülerinnen und Schüler" durch "Die Schüler" ersetzt.

Funktionalitäten "Laden und Speichern von Dateien" sowie das "Anzeigen" beinhaltet. Dieses erweitern sie derart, dass die Manipulation von Medien möglich wird. Während des Moduls müssen ebenfalls bereits geschriebene Methoden zur Medienmanipulation optimiert werden, um gute Ergebnisse zu erzielen. Dies wird dadurch gefördert, dass die Schüler nicht nur fertige Programme erstellen, sondern diese auch angewandt werden sollen, um ein Video zu erstellen. Dies kann die Schüler motivieren, ihre eigenen Programmteile immer weiter zu verbessern und anzupassen, sobald bei einer Bildbearbeitung ein etwas anderes Ergebnis gewünscht ist.

Als letzten Punkt zu diesem Kompetenzbereich nennt der Kernlehrplan "Die Schüler testen und korrigieren Computerprogramme." Auch dieser Punkt ist im Modul wiederzufinden. Schüler werden einerseits aufgefordert, ihre Programmteile zu testen. Andererseits ist es auch in ihrem eigenen Interesse, dass diese funktionieren, um ihre eigenen Bilder in der Art bearbeiten zu können, wie sie es wünschen. Da hier mit visuellen Ausgaben gearbeitet wird, die sofort angezeigt werden, wenn die Programmteile ausgeführt werden, können Schüler auch sofort Fehler im Programmablauf erkennen und korrigieren.

2.1.2. Argumentieren

Ein Unterpunkt zu diesem Kompetenzbereich, der in der Lerneinheit gefördert wird, ist: "Die Schüler analysieren und erläutern informatische Modelle". Auch dies geschieht wieder anhand der Modelle zur Repräsentation von Bildern. Obwohl natürlich die Schüler die ganze Zeit mit diesen Modellen arbeiten, geschieht das Analysieren und Erläutern explizit in der vorbereitenden Unterrichtsreihe. Die Modelle Raster- und Vektorgrafik werden hier besprochen. Hierbei analysieren die Schüler zunächst diese Modelle während der zu bearbeitenden Aufgaben. Anschließend werden die beiden Modelle besprochen, wobei für verschiedene Anwendungsbereiche diskutiert wird, welches Modell geeigneter ist. Hierbei müssen Aussagen natürlich begründet werden, wobei es aus Sicht der Schüler erforderlich sein wird, die Unterschiede der Modelle noch einmal zu erläutern. An genau diesem Beispiel wird auch der Unterpunkt des Kompetenzbereichs Argumentieren: "Die Schüler beurteilen die Angemessenheit informatischer Modelle" gezielt gefördert. Schüler sollen hier die zwei nun bekannten Modelle in Anwendungsfällen gegenüberstellen und die Angemessenheit des jeweiligen Modells beurteilen.

Behandelte Teilaspekte des Kompetenzbereichs Argumentieren

2.1.3. Kommunizieren und Kooperieren

Der erste Unterbereich dieses Kompetenzbereichs lautet: "Die Schüler kommunizieren und kooperieren in Gruppen- und Partnerarbeit". Dieser Aspekt wird in der Lerneinheit ebenfalls gefördert. Die Lerneinheit findet größtenteils in Partnerarbeit statt. Insbesondere bei der Moduldurchführung im Schülerlabor wird jedoch besonderer Wert auf diesen Aspekt gelegt. Die Schüler arbeiten in Projektgruppen, bestehend aus je drei Zweierteams, die die Aufgaben untereinander verteilen, sowie ihre Arbeit miteinander koordinieren müssen. Hinzu kommt, dass die Gruppen gemeinsam mit Dateien arbeiten und diese innerhalb der Gruppe austauschen.

Ein weiterer Aspekt dieses Kompetenzbereichs ist, dass die Schüler Arbeitsabläufe und Ergebnisse präsentieren sollen. Dies geschieht am Ende der Durchführung im Schülerlabor. Es werden hier zwar in erster Linie die Videos der Gruppen vorgestellt, jedoch wird anhand dieser auch ein informatischer Aspekt diskutiert. Wenn Schüler Methoden zur Bildbearbeitung geschrieben haben, die über die vorgegebenen Methoden auf den Arbeitsblättern hinausgehen, sollen sie diese ebenfalls erläutern. Dies geschieht, da sonst andere Gruppen nur das Endergebnis zu sehen bekämen und nicht nachvollziehen könnten, wie die Methoden zur Bildbearbeitung, die dahinterstecken, funktionieren.

Behandelte Teilaspekte des Kompetenzbereichs KuK

2.2. Kernlehrplan: Inhaltsfelder

Abgesehen von den übergeordneten Kompetenzerwartungen werden auch einige Inhaltsfelder abgedeckt. Diese sollen im Folgenden besprochen werden.

2.2.1. Algorithmen

Analyse, Entwurf und Implementierung einfacher Algorithmen

Zu diesem Inhaltsbereich können zwei Unterpunkte in der Lerneinheit wiedergefunden werden. Der erste hiervon ist: "Die Schüler analysieren und erläutern einfache Algorithmen und Programme". Das Kennenlernen der for-Schleife findet in der Art statt, dass die Schüler eine einfache for-Schleife analysieren und beschreiben sollen. Außerdem sollen Schüler ihre selbst ausgedachten Algorithmen zur Bildbearbeitung am Ende des Moduls erläutern.

Während der Durchführung im Schülerlabor kommt auch ein zweiter Aspekt dieses Inhaltsbereichs zur Geltung. "Die Schüler modifizieren einfache Algorithmen und Programme." Die Bildbearbeitungen im Modul sind im Wesentlichen Weiterentwicklungen der Methode, die in der letzten Stunde der Vorbereitung implementiert wird. Das hier eingeführte Programm zum Durchlaufen eines Bildes und Verändern einzelner Pixel wird im Modul immer wieder modifiziert und erweitert, um andere Bildbearbeitungen zu ermöglichen. Außerdem müssen die im Modul geschriebenen Methoden, beziehungsweise die Algorithmen, die dahinterstecken, immer wieder modifiziert werden. Dies muss getan werden, wenn Schüler diese Methoden anwenden und merken, dass die Methoden noch nicht das gewünschte Ergebnis erzielen. Dies kann auch erforderlich sein, wenn einfach für den momentanen Anwendungsfall ein anderes Ergebnis erwünscht ist.

Behandlung des Inhaltsbereichs Algorithmen

2.2.2. Daten und ihre Strukturierung

Objekte und Klassen

Zu diesem Inhaltsfeld ist vor allem zu einem Aspekt in der Unterrichtsreihe etwas wiederzufinden. Es handelt sich um den Aspekt "Die Schüler implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken." Die Schüler implementieren Methoden in den Klassen BildOperationen.java und SoundOperationen.java, wobei sie Klassenbibliotheken verwenden. Bei der Klasse SoundOperationen wird keine Klassenbibliothek benötigt, allerdings schon bei der Klasse BildOperationen. Hierbei ist es notwendig, die dokumentierte Klasse Pixel zu verwenden. Hierzu erhalten Schüler eine Dokumentation der Klasse, die auf die für sie relevanten Methoden der Klasse reduziert ist.

Behandlung des Inhaltsbereichs Daten und ihre Strukturierung

2.2.3. Informatik, Mensch und Gesellschaft

Einsatz von Informatiksystemen

"Die Schüler nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig sicher zielführend und verantwortungsbewusst." In diesem Modul werden verschiedene Informatiksysteme verwendet. Zur Programmierung wird im Schülerlabor Eclipse verwendet, was es für viele Informatikkurse erforderlich macht, sich auf diese neue Entwicklungsumgebung einzustellen. Zum Einstieg in diese Entwicklungsumgebung wird lediglich ein Hilfsblatt ausgegeben, der Rest erfolgt

Behandlung des Inhaltsbereichs IMuG in eigenständiger Arbeit der Schüler. Des Weiteren ist im Rahmen der Projektarbeit im Schülerlabor die gemeinsame Nutzung der Dropbox und das Austauschen von Dateien und Quelltext erforderlich. Im letzten Abschnitt, der im Schülerlabor durchgeführt wird, wird der Windows Movie Maker verwendet, dessen grundsätzliche Funktionen zwar auf einem Hilfsblatt erklärt werden, Schüler sollen jedoch auch hier selbständig arbeiten.

Wirkungen der Automatisierung

"Die Schüler bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen." Dieser Aspekt des Inhaltsfeldes kommt hauptsächlich in der letzten Stunde der Reihe zur Geltung. In dieser Stunde werden diverse Fallbeispiele, unter anderem auch Zeitungsausschnitte und Videos besprochen. Diese sind mit Informatiksystemen bearbeitet worden, wobei ausgehend von diesen Beispielen die allgemeinen gesellschaftlichen Konsequenzen der Medienmanipulation durch Informatiksysteme besprochen werden.

Behandlung des Inhaltsbereichs Wirkungen der Automatisierung

2.3. Vorgaben zum Zentralabitur

²Diese Lerneinheit ist in erster Linie für die Einführungsphase konzipiert. Trotzdem ist es interessant zu vergleichen, inwiefern Inhalte der Lerneinheit sich mit den Vorgaben zum Zentralabitur decken. Erstens weil in der Einführungsphase schon Grundlagen für die Q1³ und Q2⁴ gelegt werden. Zweitens weil es durchaus dazu kommen kann, dass auch Kurse der Qualifikationsphase dieses Modul durchführen.

Behandlung von Grundlagen für das Zentralabitur

2.3.1. I. Objektorientiertes Modellieren und Implementieren von kontextbezogen Anwendungen

I.1 Konzepte des objektorientierten Modellierens: Klasse, Objekt, Attribut, Methode, Geheimnisprinzip

Die Schüler arbeiten hier mit Klassen und Methoden. In einer derart frühen Phase des Lernens von Programmieren wurde noch kein Wert

² Als Grundlage dienen hier die Vorgaben zum Zentralabitur für 2016.[10]

³ Zweites Jahr der Oberstufe.

⁴ Drittes Jahr der Oberstufe.

darauf gelegt, dass die Schüler beim Erstellen von Klassen das Geheimnisprinzip anwenden. Allerdings sind die Schnittstellen der Klassen, mit denen gearbeitet wird, so gestaltet, dass das Geheimnisprinzip berücksichtigt wird. Die Schnittstellen sind Get- und Set-Methoden, die die Schüler in ihren Code einbauen müssen.

I.2 Algorithmen und Datenstrukturen

"Lineare Strukturen mit den Akzenten: Such- und Sortieralgorithmen für Felder und Listen." In diesem Modul werden keine Such- oder Sortieralgorithmen thematisiert, allerdings wird das Verständnis und der kompetente Umgang mit Arrays gefördert. Dies legt die Grundlage, um den Schülern später zu ermöglichen, auch Such- und Sortieralgorithmen für Arrays zu durchdenken und anzuwenden.

⁵ Der Begriff "Felder" entspricht dem Begriff "Arrays"

Kapitel 3 Medienmanipulation im Informatikunterricht

Zum Einsatz von Medienmanipulation im Informatikunterricht ist einiges an Materialien erhältlich. Eine Website, auf der das Thema vorgestellt wird, ist www.mediacomputation.org.[8] Diese Website wird von Mark Guzdial betrieben, der als Professor am Georgia Institute of Technology arbeitet.[2] Guzdial leitet hier eine Gruppe, die sich mit dem kontextorientierten Lehren von Programmierung beschäftigt. Auf der Website www.mediacomputation.org wird ebenfalls Literatur vorgestellt. Hier werden mehrere Titel genannt. Unter anderem wird ein Buch genannt, das die Grundlagen zur Programmierung in Python in einem multimedialen Kontext vermittelt, sowie ein anderes, in dem die Grundlagen der Programmierung mit Java in demselben Kontext vermittelt werden. Zu diesen Büchern können Materialien auf der Website heruntergeladen werden. Die Materialien zu Python dienten als Grundlage für das Schülerlabormodul, auf dem das im Rahmen dieser Hausarbeit erstellte Modul basiert. Gerade das Buch zu Java ist sicherlich interessant, da die Grundlagen der Programmierung mit Java vollständig in dem Kontext Medienmanipulation vermittelt werden, was im Unterricht in der Einführungsphase in der Oberstufe verwendet werden kann.

Ressourcen zur Medienmanipulation im Unterricht

In den Bildungsstandards[1] der Gesellschaft für Informatik wird darauf eingegangen, warum kontextorientiertes Unterrichten sinnvoll ist. Nach konstruktivistischer Vorstellung des Lernens konstruieren sich Lernende ihr Wissen selbst und können von außen nur zu dieser Konstruktion angeregt werden. Da neues Wissen immer mit Bezug auf bereits vorhandenes Wissen konstruiert wird, ist es sinnvoll, dass Kontexte gewählt werden, die einen Bezug zur Lebenswelt der Schüler haben. Außerdem ist es ein Grundsatz der Bildungsstandards, dass Informatik so vermittelt wird, dass die Alltagskompetenzen der Schüler im Umgang mit Informatik gefördert werden. Auch hierfür ist kontextorientierter Unterricht von Vorteil, da Unterrichtsinhalte direkt mit Bezug auf die Lebenswelt der Schüler vermittelt werden. So ist es für

Argumente für kontextorientierten Unterricht die Schüler später leichter, die Lerninhalte auch in Ihrer persönlichen Lebenswelt wieder abzurufen. [1, S. 6,7] Da die Medienmanipulation einen starken Bezug zur Lebenswelt der Schüler besitzt, ist dieser Kontext für den Informatikunterricht geeignet.

Kapitel 4 Rahmenbedingungen bei der Ausarbeitung der Lerneinheit

4.1. Lernziele

Die Lernziele, die bei der Durchführung der Lerneinheit erreicht werden sollen, sind:

Die Schülerinnen und Schüler...

Auflistung der Lernziele

- ... kennen den Unterschied zwischen Raster- und Vektorgrafiken und sind in der Lage, für Anwendungsgebiete das passende Grafikformat auszuwählen.
- ... wissen, wie Farbwerte im RGB-Farbraum dargestellt werden.
- ... sind in der Lage, Arrays zu erzeugen, zu bearbeiten und umzustrukturieren.
- ... können for-Schleifen verwenden, unter anderem auch um Arrays zu bearbeiten, zu vergrößern, zu verkleinern und zusammenzufügen.
- ... sind in der Lage, ein vorgegebenes Softwareprojekt mit Hilfe einer Entwicklungsumgebung weiterzuentwickeln.
- ... arbeiten in Projektgruppen, verteilen Aufgabenbereiche und koordinieren ihre gemeinsame Arbeit.
- ... können eine einfache Datenstruktur zur Repräsentation von Bildern nach ihren eigenen Vorstellungen bearbeiten.

- ... kennen das Prinzip der digitalen Speicherung von Tönen.
- ... beurteilen fundiert die gesellschaftlichen Auswirkungen von Medienmanipulation, deren Vorteile sowie deren Gefahren.

4.2. Voraussetzungen

Die Lerneinheit ist zwar für eine sehr frühe Phase innerhalb der Einführungsphase gedacht, allerdings sind einige Kenntnisse im Umgang mit Java erforderlich, um die Aufgabenstellungen in der Lerneinheit erfolgreich und ohne große Hilfestellung bearbeiten zu können. Es ist von Vorteil, wenn die Schüler wissen, was eine Methode, die Parameter einer Methode und ihre Rückgabe ist. Des Weiteren ist es erforderlich, dass die Schüler diese grundlegenden Datentypen kennen:

Die Voraussetzungen zur Durchführung

- Integer
- Double
- String
- Boolean.

Zu diesen Datentypen sollte auch die Verwendung von Variablen dieser Datentypen bekannt sein. Darüber hinaus sollten die Schüler logische Operatoren kennen sowie zumindest if-Verzweigungen. Schleifen und Felder sind für die Durchführung im Schülerlabor ebenfalls erforderlich. Diese werden jedoch in der vorbereitenden Unterrichtsreihe behandelt.

Kapitel 5 Ausgangsmaterialien zum Erstellen der Lerneinheit

Das Schülerlabormodul und die dazugehörige Unterrichtsreihe wurden nicht von Grund auf neu konstruiert. Im Folgenden soll beschrieben werden, welche Ausgangsmaterialien als Grundlage dienten und inwieweit und warum diese abgewandelt werden mussten, um in dem Modul und der Unterrichtsreihe zur Anwendung zu kommen.

5.1. Vorbereitung

Arbeitsmaterialien zu Grafikformaten, Präsentation

Die erste Stunde wurde komplett aus einer Unterrichtsreihe übernommen, die im Rahmen des Seminars "Fachdidaktik Informatik 3" im Sommersemester 2013 erstellt wurde. Lediglich bei der Präsentation zu Bildformaten wurden einige Formulierungen etwas geändert.

Das Arbeitsblatt zu Arrays und for-Schleifen und Test zu diesem Arbeitsblatt

Beim Erstellen dieser Arbeitsmaterialien wurden einige Ideen und Abläufe aus "Einführung in die Programmierung in Java - ein Leitprogramm in Informatik" von Christina Roeckerath, Fassung vom 11.1.12 verwendet.[3]

Ausgangsmaterialien für die vorbereitende Unterrichtseinheit 5.2. Modul 21

Arbeitsblatt zur RGB-Farbdarstellung

Dieses Arbeitsblatt wurde komplett neu erstellt.

Arbeitsblatt zu zweidimensionalen Arrays und Bildern

Dieses Arbeitsblatt wurde ebenfalls vollständig neu erstellt.

5.2. Modul

Bei der Entwicklung der Software für das Modul wurde die Klasse WavFile verwendet.[6] Die Software orientiert sich von der Aufmachung her an der Software, die im Seminar "Fachdidaktik Informatik 3" im Sommersemester 2013 erstellt wurde. Die im Modul verwendete Software wurde jedoch komplett neu programmiert, wobei auch die Möglichkeit Sounds zu bearbeiten hinzugekommen ist.

Ausgangsmaterialien für das Schülerlabormodul

Die Arbeitsmaterialien und Aufgabenstellungen für das Schülerlabor Ablauf orientieren sich an dem Schülerlabormodul "Medienmanipulation leichtgemacht". Dieses wurde im Seminar "Fachdidaktik Informatik 2"im Wintersemester 12/13 erstellt. Es wurde von den Studenten Carsten Klein, Andrea Langer und Alexander Arseniev gemeinsam entwickelt. Ein Unterschied zu dem bestehenden Schülerlabormodul ist der Wechsel der Programmiersprache von Jython zu Java. Dies war erforderlich, da die Verwendung von Jython einige Schwierigkeiten mit sich brachte. Für Schüler war es eine sehr große Herausforderung, sich auf die andere Programmiersprache einzustellen. Dazu kamen diverse Fehler der verwendeten Benutzeroberfläche. Beispielsweise musste korrekter Code, wenn er aus einer Datei als Musterlösung geladen wurde, obwohl dieser richtig eingerückt war¹, komplett neu eingerückt werden, da der Code sonst nicht kompiliert wird. Musterlösungen waren also für Schüler kaum verwertbar und es war problematisch, wenn die Schüler mit einem vorgegebenen Quelltext arbeiten sollten. Im ursprünglichen Modul wurde lediglich in Zweiergruppen gearbeitet. Das Programm des Moduls konnte jedoch aus Zeitgründen nicht vollständig absolviert werden, sodass der Abschnitt, in dem Schüler sich mit der Manipulation von Audiodateien beschäftigen sollten gewöhnlich weggelassen wurde. In dem jetzt neu erstellten Modul wurde, unter anderem um diese Zeitprobleme zu lösen, in Projektgruppen gearbeitet, sodass nicht alle Schüler alle Aufgaben bearbeiten müssen. Eine weitere Neuerung

¹ Einrückungen in Jython entsprechen Klammern in Java.

ist, dass manipulierte Audiodateien und Bilder von den Projektgruppen zu Videos zusammengefügt werden.

5.3. Nachbereitung

Die Präsentation in der Nachbereitung orientiert sich ebenfalls an der Unterrichtsreihe aus dem Sommersemester 2013. Sie wurde ausgehend von der vorhandenen Präsentation neu erstellt, wobei das Thematisieren von Videobearbeitung hinzukam.

Ausgangsmaterialien zur Nachbereitung

Teil II

DIE LERNEINHEIT

Kapitel 6 Die Software zur Medienmanipulation

6.1. Beschreibung der Software zur Medienmanipulation



Abbildung 6.1.: Die Benutzeroberfläche

Wie bereits erwähnt, wird den Schülern in dieser Lerneinheit eine Software zur Verfügung gestellt, in die die von den Schülern geschrieben Methoden integriert werden. Diese stellt verschiedene Funktionalitäten zur Verfügung. Bilder sowie Audiodateien werden in geeignete Datenstrukturen überführt, in diesem Fall Pixel- und Double-Arrays. Bilder können geöffnet, gespeichert und angezeigt werden. Genauso können Wav-Dateien geladen, gespeichert und abgespielt werden. Die Benutzeroberfläche ist in zwei Hälften aufgeteilt. Die obere Hälfte steht für die Bildmanipulation zur Verfügung. Hier gibt es zwei Flächen zur Anzeige von Bildern. Auf der linken Seite gibt es ein Auswahlmenü, in dem der Benutzer wählen kann, ob die über das Programmmenü gestarteten Funktionen auf das linke oder auf das rechte Bild angewandt

Beschreibung der Software zur Medienmanipulation werden sollen. Genau so ein Auswahlmenü existiert auch für das untere Fenster. Das untere Fenster ist für die Soundmanipulation konzipiert. Diese bietet die Möglichkeit, zwei Sounds zu laden, wobei bei jeder geladenen Audiodatei Informationen zur Datei angezeigt werden. Darüber hinaus existieren zwei Buttons, mit denen die Audiodateien abgespielt werden können. Auf der linken Seite befindet sich zusätzlich zum Auswahlmenü für die rechte und die linke Audiodatei noch ein weiteres Auswahlmenü und ein Eingabefeld. Diese sind für die Methode zum Kürzen von Audiodateien gedacht. Hier kann festgelegt werden, wie viele Sekunden einer Audiodatei abgeschnitten werden sollen sowie ob vorne oder hinten abgeschnitten werden soll. Über das Dateimenü ist das Offnen und Speichern von Bildern und Audiodateien möglich. Das Menü "Bildmanipulation" ist aufgeteilt in zwei Unterkategorien. Unter "Vorgegeben"können die während der Durchführung im Schülerlabor auf jeden Fall zu implementieren Funktionen ausgeführt werden. Unter "Eigene Funktionen" können zusätzliche Methoden implementiert werden. Die Namen der Einträge in diesem Menü können durch Verändern eines Strings innerhalb der Klasse, die die Schüler bearbeiten, verändert werden. Im Menü "Sound Manipulation"können die auf den Arbeitsblättern beschriebenen Funktionen zur Manipulation von Audiodateien ausgeführt werden.

6.2. Didaktische Überlegungen für den Entwurf der Medienbearbeitungssoftware

Für die Unterrichtsreihe wurde eine Medienbearbeitungssoftware entwickelt. Da diese in erster Linie dazu dienen sollte, ein geeigneter Unterrichtsgegenstand für Kurse zu sein, die erst wenig Erfahrung mit der Programmierung in Java haben, musste dies bei der Entwicklung der Software berücksichtigt werden. Die Software sollte hauptsächlich dazu dienen, dass die Schüler...

Zielsetzung bei dem Entwurf der Software

- mit Arrays arbeiten sollten
- auf die Datenstrukturen mit Hilfe von Get- und Set-Methoden zugreifen sollten.

Diese beiden Punkte wurden als sehr wichtig angesehen. Der erste aus dem Grund, dass Arrays einer der zentralen informatischen Inhalte der Lerneinheit sein sollten. Der zweite Punkt wurde aus dem Grund als sehr wichtig angesehen, dass Schüler hier das Prinzip der Datenkapselung berücksichtigen sollten. Thematisiert werden kann dies später

im Unterricht, allerdings sollte innerhalb der Lerneinheit nicht der Eindruck bei den Schülern entstehen, dass beim Programmieren auf Attribute von Objekten anders als auf diese Weise zugegriffen werden sollte.

Diese Entscheidungen hatten Konsequenzen, die grundsätzlich bei dem Entwurf von Unterrichtsmaterial zum Bearbeiten von Medien mit Java berücksichtigt werden müssen. Die einzige Möglichkeit, beide oben genannten didaktischen Überlegungen umzusetzen war, jeden Pixel eines Bildes als einzelnes Objekt zu erzeugen. Dies führt allerdings dazu, das bei dem Laden von hochauflösenden Bildern mehrere Millionen von Objekten erzeugt werden. Dies beeinträchtigt die Geschwindigkeit des Programms. Außerdem reicht bei normaler Konfiguration von Java der Speicherplatz, der Java zur Verfügung steht, hierzu nicht aus. Dies führt zu Programmabstürzen. Es war nicht erwünscht, dass sich diese Probleme negativ auf die Motivation der Schüler auswirken könnten. Deswegen mussten alternative Lösungen in Betracht gezogen werden. Eine Möglichkeit wäre, Bilder zwar als Objekte zu erzeugen, die Farbwerte der Pixel des Bildes allerdings nur in einem Integer-Array innerhalb des Bildobjekts zu speichern. Hierbei wäre es möglich gewesen, ebenfalls mit Get- und Set-Methoden zu arbeiten. Allerdings hätte dies zur Folge gehabt, dass die Schüler selbst nicht mehr direkt mit Array gearbeitet hätten. Get- und Set-Methoden hätten hierbei zusammen mit den Koordinaten des jeweiligen Pixels aufgerufen werden müssen. Die andere Möglichkeit wäre gewesen, in diesem Fall das Integer-Array des Bildobjekts direkt zu bearbeiten, allerdings wäre hierdurch der Aspekt der Datenkapselung weggefallen.

Beide Möglichkeiten hätten sowohl das Speicher- als auch das Performanceproblem gelöst. Allerdings wurde hier entschieden, den didaktischen Überlegungen Vorrang zu geben. Die Software wurde so entworfen, dass jeder Pixel als einzelnes Objekt erzeugt wird. Die Bilder werden einfach vor der Bearbeitung, sobald sie geladen werden, auf etwas weniger als zwei Millionen Pixel¹ verkleinert. Dies klingt im ersten Moment nach einer geringen Auflösung. Allerdings ist dies eine Auflösung, die immer noch über der liegt, mit der Bilder beispielsweise in sozialen Netzwerken hochgeladen werden. Außerdem ist hier auch die geplante Anwendung, die Erstellung von Videos, zu beachten. Full HD Videos haben in etwa eine Auflösung von zwei Millionen Pixeln. Auch für das Erstellen von Videos ist diese Auflösung also durchaus geeignet.

Eine weitere didaktische Überlegung bei dem Entwurf der Software war das Auslagern der von den Schülern zu implementierenden Methoden in eigene Klassen. Schüler sollten nicht, besonders in Anbetracht ihrer anzunehmenden geringen Programmierkenntnisse, durch einen Quelltext überfordert werden, der für die Aufgaben selbst keine

Umsetzung der Ziele

Vermeidung von Performanceproblemen

Berücksichtigung der Aufgabenstellungen

¹ Das entspricht 2 Megapixeln

Rolle spielt. Die ursprünglichen Methoden wurden also aufgespalten, sodass nur der Teil, den die Schüler auch bearbeiten sollen, in eine eigene Klasse ausgelagert wurde. Es handelt sich dabei um die Klassen BildOperationen.java und SoundOperationen.java.

Kapitel 7 Aufbau der Lerneinheit

7.1. Überblick über die Lerneinheit

7.1.1. Zielsetzung

In dieser Lerneinheit erstellen die Schüler eine Software zur Medienmanipulation in Projektarbeit (Sechsergruppen). Ein Grundgerüst ist hierfür vorhanden, wobei die Schüler nur noch Methoden schreiben müssen, die ein Array (entweder ein Bild oder eine Audiodatei) übergeben bekommen, diese dann umsortieren oder pixelweise verändern. Das Erstellen dieser Methoden findet hauptsächlich im Schülerlabor statt. Die fünfstündige Vorbereitungseinheit dient dazu, die Grundlagen (Bildformate, Farbwerte, Arrays, for-Schleifen) zu erarbeiten. In der Nachbereitung (eine Stunde) wird auf die gesellschaftliche Bedeutung der Medienmanipulation eingegangen. Als Anwendung und für die Motivation der Schüler wird im Schülerlabor mit Kameras, der erstellten Software und dem Windows Movie Maker von jeder Gruppe ein Video erstellt.



Abbildung 7.1.: Ablauf der Lerneinheit

7.1.2. Vorbereitung im Rahmen des Unterrichts

1./2. Stunde: Einstieg; Modulablauf, Zielsetzung wird vorgestellt.

Einstieg: Den Schülern wird ein fertiges Video gezeigt, das ebenfalls mit der Software erstellt wurde, die im Modul erstellt wird. Dies dient als Vorschau darauf, was die Schüler im Verlauf des Moduls erarbeiten werden. Die fertige Software wird auch kurz demonstriert. (15 min) Erarbeitung: Mit Rastergrafiken und Bildern ohne Raster auf Papier sollen SuS die Unterschiede zwischen Vektor- und Rastergrafiken erarbeiten. Dies geschieht in Zweiergruppen, wobei jeweils einer der Schüler seine Grafik dem anderen beschreibt. Bei den Rastergrafiken erfolgt die Beschreibung durch Koordinaten, bei Vektorgrafiken durch geometrische Formen. (20 min) Sicherung: In einer Präsentation werden von der Lehrkraft verschiedene Anwendungsbereiche für Grafiken vorgestellt. Die Schüler sollen sich dazu äußern, welches Grafikformat für welchen Anwendungsbereich geeigneter ist. (10 min)

2./3. Stunde: Grundkenntnisse in Java wiederholen/erarbeiten.

Erarbeitung: Hierzu wird ein Arbeitsblatt zu Arrays und for-Schleifen bearbeitet. Darauf werden die Grundlagen erklärt und von den Schülern in Aufgaben kleine Programme geschrieben, um diese Grundlagen direkt anzuwenden. (65 min) Sicherung: Die Schüler bearbeiten einen kurzen Test und korrigieren sich gegenseitig. (25 min)

Ablauf der einzelnen Unterrichtsstunden

4. Stunde: RGB-Farbdarstellung.

Zu Beginn wird noch einmal gemeinsam der Test der letzten Stunde besprochen. (10 min) Ein Arbeitsblatt zur Darstellung von Farben im RGB-Farbraum wird bearbeitet. Neben erklärenden Abschnitten gibt es hier eine Aufgabe zum grundsätzlichen Mischverhalten von Farben, sowie eine Aufgabe zu diskreten Farbwerten. (20 min)

4./5. Stunde: Zweidimensionale Arrays, verschachtelte for-Schleifen

Dies wird anhand eines Arbeitsblattes erarbeitet. Die Schüler lernen, wie ein zweidimensionales Array durchlaufen wird und wenden dies an. In einer Aufgabe wird ein Integer-Array, das ein monochromes Bild darstellen soll, bearbeitet. (60 min)

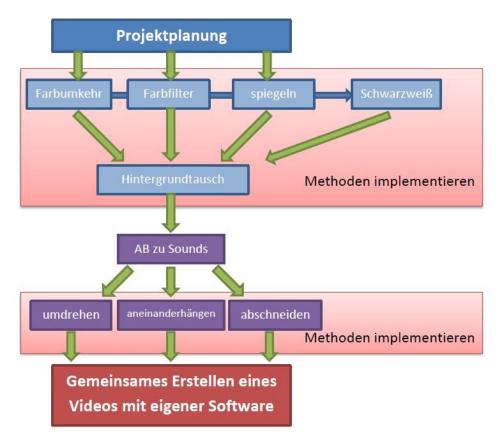


Abbildung 7.2.: Aufbau des Moduls für das Schülerlabor

7.1.3. Modul

Der Ablauf des Moduls wird in einer Präsentation vorgestellt, außerdem werden noch einmal die Software zur Medienbearbeitung und das Beispielvideo gezeigt. (10 min)

Projektplanung. Die Gruppen verteilen die Aufgaben unter ihren Mitgliedern.¹ Außerdem sollen die Schüler erste Überlegungen anstellen, welche eigenen Methoden zur Bildbearbeitung sie zusätzlich implementieren könnten. Ein erstes Konzept zur Gestaltung des Videos soll ebenfalls erdacht werden. (10 min)

Ablauf der einzelnen Modulphasen

Die Gruppen bearbeiten die *Methoden zur Bildbearbeitung*, die sie verteilt haben. Diese sind Methoden zum:

- Spiegeln
- Farbfilter anwenden

¹ Im Modul wird als 6er-Projektgruppe gearbeitet.

- Farben umkehren
- Schwarzweißbild erstellen (Bonusaufgabe, für schnellere Zweierteams)

Wenn diese Methoden erfolgreich implementiert wurden, soll jedes Zweierteam innerhalb der Projektgruppe eine Methode zum Hintergrundaustausch implementieren. Zu allen Methoden stehen Arbeitsblätter zur Verfügung, insgesamt sind für diesen Block 60 Minuten vorgesehen.

Hiernach erhalten die Schüler Zeit, um weitere Bilder zu erstellen, zu bearbeiten und zusätzliche Methoden zu implementieren. Diese Methoden sind nicht vorgegeben, in die Software können jedoch bis zu fünf weitere Methoden integriert werden. Es kann ebenfalls deren Bezeichnung festgelegt werden, ohne dass die Schüler dafür mit GUI-Programmierung in Kontakt kommen. (60 min)

Im nächsten Abschnitt sollen die Schüler, verteilt auf die drei Zweierteams, jeweils eine Methode zur *Bearbeitung von Audiodateien* implementieren. Diese drei Methoden werden ebenfalls in die Software integriert. (45 min)

Mit der nun fertigen Software sollen Audiodateien, die mit den Mikrofonen in den Laptops aufgenommen oder beispielsweise auf www.freesound.org heruntergeladen werden können, sowie Bilder bearbeitet und mit dem sehr benutzerfreundlichen Windows Movie Maker zu einem Video zusammengefügt werden. (60 min)

Die fertigen Videos werden vorgestellt. Selbst erdachte Methoden sollen dabei erläutert werden. (30 min)

7.1.4. Nachbereitung

Eine Schulstunde: Informatik/Gesellschaft, das Thema ist die Glaubwürdigkeit von Medien.

Einstieg: In einer Präsentation werden Bilder gezeigt, deren Aussagen bewusst verändert wurden, auch Beispiele, die in Zeitungen abgedruckt wurden. Außerdem wird ein Video gezeigt, indem Material aus Fernsehserien sehr stark bearbeitet wurde. Zu diesem Video wird im Anschluss noch gezeigt, wie und aus welchem Ausgangsmaterial das Video erstellt wurde. Dies soll demonstrieren, dass auch Videos

Ablauf der Nachbereitung mit einfachen Mitteln so verfälscht werden können, dass völlig andere Szenen dargestellt werden. (20 min) Erarbeitung: Die SuS sollen in Vierergruppen (homogen Pro- oder Contra Medienmanipulation) Argumente für ihren Standpunkt finden. (5 min) Erarbeitung/Sicherung: Die Argumente sollen im Kurs diskutiert werden. Die Hauptargumente werden hierbei an der Tafel festgehalten. (20 min)

7.2. Didaktische Überlegungen zum Aufbau der Lerneinheit

Bevor auf die genaue Planung der einzelnen Unterrichtsstunden und der Abschnitte des Moduls eingegangen wird, ist es sinnvoll zu erläutern, welche grundsätzlichen Überlegungen zu dem Entwurf der Einheit beigetragen haben.

Den Kontext Medienmanipulation in einer so frühen Phase des Programmierens mit Java einzusetzen, bietet einige Vorteile. Anstatt Arrays lediglich mit Zahleneinträgen oder als Namenslisten kennenzulernen, kann hier für die Schüler ein interessanter Kontext geboten werden. Die gestalterischen Möglichkeiten, diese Arrays zu manipulieren, bieten Möglichkeiten zur Motivation und Differenzierung. Bei den Schülern sind noch sehr geringe Programmierkenntnisse zu erwarten, aufgrund des Kontextes können jedoch Aufgabenstellungen entwickelt werden, bei denen trotzdem Erfolgserlebnisse möglich sind. Hierzu wurde eine Software entwickelt, die die Schüler im Rahmen der Aufgaben ergänzen sollen. Das Ziel hierbei war, dass die Schüler die im Rahmen der Aufgaben geschriebenen Methoden nicht nur als kleine Programmteile ansehen, sondern erleben können, wie diese in einer Software zum Einsatz kommen. Hierbei ist davon ausgegangen worden, dass Schüler es als wesentlich zufriedenstellender und motivierender ansehen, am Ende der Lerneinheit durch ihre Arbeit eine Medienbearbeitungssoftware fertig gestellt zu haben, als zurückblickend nur einzelne, unzusammenhängende Programme entwickelt zu haben. Des Weiteren war es möglich, in diesem Kontext direkt auch die Anwendung der erstellten Software in den Unterricht einzubetten. Die Lerneinheit wurde so geplant, dass die Schüler am Ende Ergebnisse haben, die Software sowie das Video, die sie mitnehmen und vorzeigen können. Da die Unterrichtsreihe für eine frühe Phase der Informatik innerhalb der Oberstufe geplant wurde, ist gerade dies nicht zu vernachlässigen. Es besteht durchaus die Möglichkeit, dass Kurse diese Reihe durchführen, die vorher keinen oder nur sehr wenig Informatikunterricht in Anspruch nehmen konnten. Somit kann bereits sehr früh ein positives Bild vom Informatikunterricht bei den Schülern gefördert werden sowie ein Verständnis der Problem- und Produktorientierung des Informatikunterrichtes. Außerdem bietet dieser Kontext

Nutzen der Vorteile des Kontextes Medienmanipulation die Möglichkeit, dass Schüler ihr Wissen und ihre Kompetenzen an einem motivierenden Beispiel verbessern, und das möglichst frei und spielerisch. Da dies auch ein Motivationsfaktor ist, ist anzunehmen, dass Schüler eher bereit dazu sind, sich intensiv mit den Lernmaterialien auseinanderzusetzen.

Bei der Planung des Schülerlabormoduls war es beabsichtigt, eine möglichst kreative Arbeitsweise der Schüler zu fördern. Damit dies umgesetzt werden kann, ist es erforderlich, dass die Schüler während der Durchführung geistig in der Lage sind, kreativ zu arbeiten. Deswegen sollten die Schüler nicht mit grundlegend neuen Inhalten übermäßig kognitiv belastet werden. Dadurch, dass die Grundlagen bereits bekannt sind, kann das Kurzzeitgedächtnis entlastet werden. Aus diesem Grund wurde das Erarbeiten von Grundlagen vollständig in die vorbereitende Unterrichtsreihe ausgelagert. Diese Entscheidung wurde auch noch aus einem weiteren Grund getätigt. Hierdurch ist für die Moduldurchführung mehr Zeit vorhanden. Dadurch kann eine kreative und freie Arbeitsweise überhaupt erst stattfinden. Da für das Bearbeiten von Aufgabenstellungen mehr Zeit zur Verfügung steht, müssen die Schüler weniger kleinschrittig angeleitet werden. Dies ist von Vorteil, da die Schüler selbst mehr über das Bearbeiten der Aufgabenstellungen nachdenken müssen. Dies fördert nach konstruktivistischer Vorstellung die Konstruktion von Wissen.[5, S. 10] Eine weitere Überlegung war, dass sich die Schüler nicht durch zu kleinschrittige Anleitungen bevormundet fühlen. Diese Entscheidung kann dazu führen, dass die Bereitschaft, sich mit dem Thema auseinanderzusetzen, eher bis zum Ende des Moduls aufrecht erhalten bleibt.

Ein weiterer Grund, weshalb gerade dieser Abschnitt der Reihe im Schülerlabor durchgeführt werden soll, ist, dass auf diese Weise das Modul so aufgebaut werden kann, dass über den gesamten Zeitraum bei den Schülern eine klare Zielvorstellung vorhanden ist. Nachdem die Zielsetzung des Moduls am Anfang vorgestellt wird, können die Aufgabenstellungen so gewählt werden, dass die Aufgaben durchgängig zu dem eigentlichen Modulziel hinführen und dies den Schülern auch bewusst ist. Außerdem handelt es sich bei dem für das Modul vorgesehenen Teil der Unterrichtsreihe um den Anteil, für den es erforderlich ist, dass durchgehend Fotos erstellt werden können. Im Schülerlabor ist es möglich, hierfür das Equipment, die Kameras, die Stative, die Beleuchtung und den Bluescreen aufzustellen, sodass diese während der gesamten Durchführung zur Verfügung stehen. In einem Informatikraum an einer Schule sind hierfür nicht unbedingt die Materialien und der Platz gegeben.

Für die Unterrichtsstunde nach der Durchführung des Moduls wurde der gesellschaftliche Aspekt des Themas der Lerneinheit ausgewählt. Aufteilung der Lerninhalte auf Modul und Unterrichtsreihe Dies wurde am Ende der Lerneinheit platziert, da Schüler zu diesem Zeitpunkt bereits ein sehr klares Bild von dem fachlichen Aspekt des Themas haben. Somit ist eine Grundlage für die Diskussion der gesellschaftlichen Aspekte vorhanden. Ein weiterer Grund, dieses Thema für die letzte Unterrichtsstunde zu wählen, hängt auch wieder mit dem frühen Zeitpunkt innerhalb der Oberstufe zusammen. Auch hier war die Absicht, das eventuell noch im Aufbau befindliche Bild des Informatikunterrichts bei Schülern zu formen. Die Verbindung zwischen den informatischen Inhalten und der Lebenswelt der Schüler kann den allgemeinen Bezug des Informatikunterrichts zu ihrem persönlichen Leben deutlicher machen.

Kapitel 8 Detaillierte didaktische Begründung

8.1. Grundlagen für didaktische Überlegungen und allgemeine Schlussfolgerungen

Nach konstruktivistischer Vorstellung des Lernens ist Lernen nur durch aktives Auseinandersetzen des Lernenden mit den Lerninhalten möglich. Verstehensprozesse und Erkenntnisse sind von daher nicht vermittelbar, sondern Lernende müssen diese selbst durchlaufen.[5, S. 10]

Berücksichtigung konstruktivistischer Vorstellungen

Bei der Planung der Lerneinheit und dem Entwurf der Materialien standen deswegen diese Verstehensprozesse im Vordergrund. Das Ziel war außerdem, das Lernen möglichst aktiv und selbst gesteuert zu gestalten. Nach jedem Lernschritt sollten Aufgaben dazu anregen, noch einmal über das Gelernte nachzudenken und es anzuwenden, so dass das Stattfinden der Verstehensprozesse gesichert ist.

Bezieht man die Funktion des Gedächtnisses mit ein, so ist dieses Vorgehen sogar noch wichtiger. Das Kurzzeitgedächtnis ist nur in der Lage, sehr wenig Information zu behalten, daher ist es sinnvoll, Lerninhalte möglichst sofort zu wiederholen und anzuwenden. [5, S. 12]

Eine weitere Grundlage für didaktische Überlegungen sind die Prinzipien didaktischen Handelns nach Peter Hubwieser. [5, S. 14-23] Diese sind:

Prinzipien didaktischen Handelns

- Motivierung
- Kreativitätsförderung

- Strukturierung
- Übung
- Veranschaulichung
- Bewertung und Erfolgssicherung
- Variabilität und Flexibilität
- Differenzierung

Es wurde ebenfalls versucht, diese Prinzipien in der Lerneinheit anzuwenden. Hubwieser nennt hier ebenfalls einige Praxistipps, die an dieser Stelle nicht aufgezählt werden, jedoch im Folgenden bei der Begründung, warum Unterrichtsmaterialien die oben genannten Prinzipien erfüllen, zur Anwendung kommen.

8.2. Didaktische Begründung für Unterrichtsmaterialien der vorbereitenden Unterrichtsreihe.

8.2.1. Einführung und Grafikformate

Lerninhalte

Die Schüler erhalten einen Überblick über den Verlauf der gesamten Lerneinheit. Außerdem lernen die Schüler die Grafikformate Rasterund Vektorgrafik kennen. Zu diesen beiden Grafikformaten sind die Schüler in der Lage, Anwendungsbereiche zuzuordnen, in denen das jeweilige Grafikformat zu bevorzugen ist.

Vorwissen

Für diese Unterrichtsstunde benötigen die Schüler kein Vorwissen.

Inhaltliches Vorgehen

Die vorbereitende Unterrichtsreihe beginnt mit einer Präsentation, in der den Schülern das Thema der gesamten Lerneinheit vorgestellt wird. Hierbei wurde zunächst der Begriff Medienmanipulation erläutert. Im Anschluss daran wurden einige lebensnahe Beispiele gewählt, um zu zeigen, wo den Schülern im Alltag Medienmanipulation begegnet. Die Absicht hierbei war, die Schüler zu motivieren, indem das Thema Medienmanipulation mit dem persönlichen Erfahrungsbereich der Schüler in Verbindung gebracht wird. Während der Präsentation wird ebenfalls das Ziel der Reihe vorgestellt. Diese Zielvorstellung trägt dazu bei, das Prinzip der Strukturierung anzuwenden. Die Schüler sollen ein zweckgerichtetes Beziehungsgefüge aufbauen, in dem die Inhalte der Lerneinheit möglichst direkt mit dem Zweck, zu dem diese innerhalb der Reihe aus Sicht der Schüler vermittelt werden, in Verbindung gebracht werden. So können diese zu einem logischen Beziehungsgefüge zusammengesetzt werden. Um diesen Vorgang zu unterstützen und die Schüler aktiv mit einzubinden, sollen sich die Schüler zuerst selbst Gedanken machen, welche Kenntnisse sie benötigen, um das Ziel der Reihe erreichen zu können. Dies wird im Unterrichtsgespräch zusammengetragen.

Aufbau der Einführungspräsentation

Die Planung des restlichen Verlaufs der ersten Stunde wurde, wie bereits erwähnt, aus einer Unterrichtsreihe entnommen, die in einem Seminar zur Fachdidaktik Informatik entwickelt wurde. Es gibt einige didaktische Überlegungen, die dazu geführt haben, dass die Unterrichtsstunde übernommen wurde. In dieser Stunde sollen die Schüler den Unterschied zwischen Raster- und Vektorgrafiken erarbeiten. Dies geschieht ohne Computer, nur mit Hilfe von Buntstiften und Papier. Man kann insgesamt von einer sehr spielerischen Herangehensweise sprechen. Dies ist besonders geeignet für die erste Stunde innerhalb der Reihe. Dies ist der Fall, da diese Stunde sich von dem allgemein üblichen Informatikunterricht, in dem eher paarweise am Computer gearbeitet wird, stark unterscheidet. Diese Vorgehensweise erfüllt das Prinzip didaktischen Handelns Variabilität und Flexibilität. Um dieses Prinzip umzusetzen ist es sinnvoll, unterschiedliche Lehrmethoden sowie Medien und Sozialformen einzusetzen.[5, S. 22] Darüber hinaus ist es besonders an dieser Stelle sinnvoll, eine Unterrichtsstunde einzusetzen, die sich von (wahrscheinlichem) alltäglichen Unterricht unterscheidet. Schüler erhalten so direkt den Eindruck, dass die gesamte Lerneinheit sich vom Konzept her eher vom alltäglichen Informatikunterricht unterscheidet. Dies kann dazu beitragen, das Interesse an der gesamten Lerneinheit zu wecken. Wichtig ist hier auch, dass die Unterschiede zwischen Raster- und Vektorgrafiken nicht einfach nur

Aufbau der Aufgabe zum Kennenlernen der Grafikformate

in einem erklärenden Text gelesen werden, sondern Schüler sich diese in Zweiergruppen selbst erarbeiten sollen. Dies tun sie, indem sie in der zweiten Aufgabe die vorgegebenen Fragen in Zweiergruppen diskutieren und die Ergebnisse aufschreiben. Auch dies ist, wie bereits erwähnt, nach konstruktivistischem Lernverständnis sinnvoll, da sich die Schüler auf diese Weise eigenständiger mit den Lerninhalten auseinandersetzen.

Im Anschluss daran werden als Kurs verschiedene Anwendungsfälle für Grafikformate durchgesprochen, dies dient zur Übung und Sicherung der erarbeiteten Lerninhalte.

8.2.2. Arbeitsblatt zu Arrays und for-Schleifen

Lerninhalte

Die folgende Grafik stellt die Gliederung der Lerninhalte dieser Stunde zu Arrays dar:

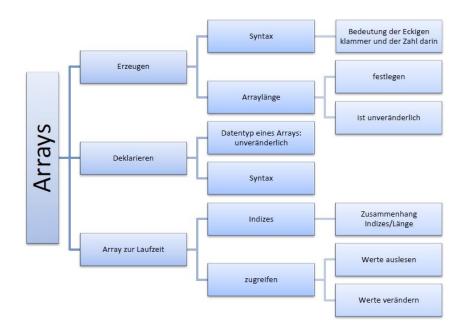


Abbildung 8.1.: Lerninhalte zu Arrays

Hierbei handelt es sich um das Grundwissen, das benötigt wird, um Arrays in Java zu verwenden.

Zu den for-Schleifen müssen die Schüler ebenfalls ein recht komplexes, wenn auch im Vergleich zu den Arrays weniger komplexes, Beziehungsnetz aufbauen.

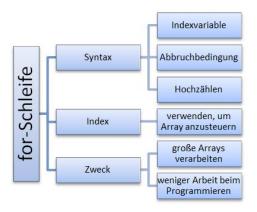


Abbildung 8.2.: Lerninhalte zu for-Schleifen

Auch hierbei handelt es sich wieder um das Grundwissen, ohne das for-Schleifen nicht richtig verwendet werden können.

Vorwissen

Für die Bearbeitung dieses Arbeitsblattes müssen die Schüler die primitiven Datentypen String und Integer kennen. Einfache Wertzuweisungen sind ebenfalls erforderlich.

Konzeptionelle Überlegungen

Das Arbeitsblatt zu Arrays und for-Schleifen wurde so konzipiert, dass Schüler nach dem Lesen von informierenden Texten die gerade gelesene Information in einer Aufgabe anwenden müssen. Hierbei wurden die Prinzipien Übung und Erfolgssicherung berücksichtigt. Das gerade Gelesene soll so vor dem Vergessen bewahrt werden, indem eine aktive Auseinandersetzung stattfindet. Bevor die for-Schleifen eingeführt werden, wird eine Aufgabenstellung präsentiert, die ohne Schleifen nicht effizient lösbar ist. Dies soll Schüler motivieren sich mit dem Thema auseinanderzusetzen, da der unmittelbare Nutzen von Schleifen deutlich wird. In der letzten Aufgabe dieses Arbeitsblattes sollen die Schüler eine einfache for-Schleife anhand von gegebenen Fragestellungen analysieren. Auch dies soll wieder eine möglichst aktive Auseinandersetzung mit den Grundlagen von for-Schleifen fördern.

Inhaltliche Vorgehensweise

Arrays:

Anhand der Grafik zu den Lernzielen wird deutlich, dass innerhalb der zweiten und dritten Stunde zu der vorbereitenden Unterrichtsreihe bei Schülern ein komplexes Beziehungsnetz aufgebaut werden muss. Als erster Schritt sollten die Schüler eine Vorstellung eines Arrays zur Laufzeit erhalten. Hierzu wurde auf dem Arbeitsblatt den Schülern gegenüber ein Vergleich mit Listen erwähnt, da mit dem Begriff Liste jeder Schüler etwas anfangen können sollte. Außerdem wurde ein Array mit nur drei Einträgen abgebildet, was der Veranschaulichung dienen soll.

Vorstellung eines Arrays zur Laufzeit

Anschließend wird auf das Deklarieren und Erzeugen von Arrays eingegangen. Die Syntax der Anweisungen wird auf dem Arbeitsblatt durch Beispielprogrammzeilen eingeführt. Dies hat den Zweck, das Abstraktionsniveau möglichst gering zu halten. Die Programmzeilen werden durch Beschriftungen der einzelnen Abschnitte erläutert. Dies hat den Zweck, dass die einzelnen Abschnitte der Programmzeilen und ihre Bedeutung erkannt werden, dabei aber direkt als Ganzes erfasst werden können. In einem anschließenden, kurzen Text werden die einzelnen Teile der Programmzeilen noch einmal erläutert, was das richtige Verständnis der Abbildungen unterstützen soll.

Deklarieren und Erzeugen von Arrays

Nach demselben Schema wurde auch das Zugreifen auf die Einträge der Arrays erklärt.

Zugreifen auf Arrayeinträge

In der anschließenden Aufgabe sollen die Schüler die gerade gelesenen Inhalte anwenden und deren Zusammenhänge verstehen. Die Syntax der Programmbefehle soll durch das Schreiben eines eigenen kleinen Programms geübt werden. Außerdem soll ein Zusammenhang zwischen dem gerade geschriebenen Programmcode und der Vorstellung eines Arrays hergestellt werden. Hierzu sollen die Schüler das Array, das in ihrem Programm erzeugt wird, skizzieren. Auch hierbei soll wieder unterstützt werden, dass die Schüler sich Datenstrukturen, wenn möglich, visuell vorstellen. Zum Schluss soll erläutert werden, warum auf einen Eintrag außerhalb der Grenzen des Arrays nicht zugegriffen werden kann. Eine "Index out of bounds exception" wurde auf dem Arbeitsblatt bereits erläutert. Das Ziel hierbei ist, das Bewusstsein bei den Schülern zu fördern, dass wirklich nur innerhalb der Grenzen des Arrays auf Einträge zugegriffen werden darf.

Berücksichtigung der Arraygrenzen

For-Schleifen:

Dass die nicht zu lösende Aufgabe zu Arrays der Motivation die-

Zweck von for-Schleifen

nen soll, wurde bereits erwähnt, hierbei soll allerdings auch noch der Zweck von Schleifen deutlich werden. Die Schüler haben gerade ein Programm geschrieben, in dem fünf Array-Einträge erstellt und ausgegeben werden sollten, von daher sollte den Schülern sofort auffallen, dass das Anlegen von 100 Array-Einträgen ohne Schleifen sehr aufwändig ist. Hierbei soll also die Arbeitsersparnis durch Schleifen deutlich werden.

Als nächstes wird die Syntax von for-Schleifen erläutert. Auch hier wird das wieder anhand von Beispielquelltext getan, um das Abstraktionsniveau möglichst niedrig zu halten. Auch dazu werden die Programmzeilen schrittweise erläutert.

Die restliche Auseinandersetzung mit der Syntax sowie mit der Funktion dieser einfachen for-Schleife findet in der nächsten Aufgabe statt. Die Schüler sollen anhand der Ausgabe eines kleinen Programms das Verhalten der Zählvariable innerhalb der for-Schleife nachvollziehen. Dies umfasst nicht nur das Hochzählen, sondern auch den Anfangsund Endwert in Abhängigkeit von der Deklaration und der Abbruchbedingung im Schleifenkopf. Anhand dieser Aufgabe sollen Schüler ebenfalls nachvollziehen, wie die Zählvariable verwendet wird, um die Einträge des Arrays anzusteuern. Dies geschieht in der letzten Teilaufgabe, um so auch die Verbindung zu Arrays herzustellen.

Syntax

Verwendung Zählvariablen

8.2.3. Test zu Arrays und for-Schleifen

Der anschließende Test soll zur Erfolgssicherung und Übung dienen. Hier werden noch einmal die Grundlagen zu for-Schleifen und Arrays angewandt. Der Test soll von den Schülern gegenseitig kontrolliert und verbessert werden. Dies soll ebenfalls eine aktivere und tiefere Verarbeitung der Lerninhalte fördern. Das erneute Anwenden der Grundlagen auf Papier ist auch insofern von Vorteil, dass typisches Trial-and-Error-Verfahren der Schüler vermieden wird. Es soll bewirkt werden, dass Schüler sich zuerst Gedanken machen, bevor sie die Anweisungen zu Papier bringen, wodurch ebenfalls eine höhere Verarbeitungstiefe erzielt werden soll.

Argumente für einen schriftlichen Test

Inhaltlich wurde bewusst eine Aufgabe gewählt, in der ein String Array bearbeitet werden soll. Auf diese Weise kann eine, zumindest geringe, Transferleistung vorausgesetzt werden, da String Arrays bisher zwar erwähnt, jedoch nicht verwendet wurden. Der Test dient ansonsten hauptsächlich zur Wiederholung. Es soll ein Array deklariert und erzeugt werden, was Inhalt des Arbeitsblattes war und bereits in den Übungsaufgaben durchgeführt wurde. Außerdem sollen im Test dem Array Einträge zugeordnet werden. Diese sollen auch ausgegeben werden. Die beiden Punkte wurden ebenfalls bereits in den

Auswahl der abgefragten Lerninhalte

Übungsaufgaben des Arbeitsblattes behandelt und von den Schülern angewandt. Ein Vorteil der Wiederholung an dieser Stelle ist, dass sich hier jeder einzelne Schüler mit dem Test auseinandersetzen muss. Bei Zweiergruppen, in denen ein Schüler die ganze Arbeit übernimmt, der andere sich jedoch nicht wirklich beteiligt, erhält dieser hier wenigstens ein Feedback über seine bisherige Lernleistung.

8.2.4. Arbeitsblatt zur RGB Farbdarstellung

Lernziele

In diesem Abschnitt der Unterrichtsreihe sollen sich die Schüler mit der RGB-Farbdarstellung beschäftigen. Der Umgang mit RGB-Farbwerten ist im weiteren Verlauf der Lerneinheit von großer Bedeutung, da während der Durchführung im Schülerlabor Farbwerte bearbeitet werden. Hierfür ist allerdings nicht nur die Kenntnis der Grundlagen erforderlich, sondern auch ein gewisses Gefühl für Farbwerte und das Mischen von Farben. Die Schüler sollen also eine Vorstellung davon haben, was in etwa passiert, wenn der Wert eines Farbkanals um 10 erhöht wird und wie sich verschiedene Farben durch die Grundfarben zusammensetzen lassen. Dementsprechend wurden die Lernziele formuliert.



Abbildung 8.3.: Lerninhalte zur RGB-Farbdarstellung

In diesem Abschnitt der Unterrichtsreihe sollen sich die Schüler also im Wesentlichen mit zwei Unterthemen auseinandersetzen. Einerseits geht es um das Mischen von Farben aus den Grundfarben Rot, Grün und Blau und wieso es überhaupt möglich ist, alle Farben aus diesen Grundfarben zusammenzusetzen. Das andere Unterthema ist die Digitalisierung. Hierbei geht es darum, wie die Farbwerte durch Zahlen dargestellt werden können.

Vorwissen

Für diese Unterrichtsstunde ist kein Vorwissen nötig.

Konzeptionelle Überlegungen

Im ersten informierenden Text zur Farbmischung aus den Grundfarben Rot, Grün und Blau wird der Bezug des informatischen Inhalts zu dem persönlichen Leben der Schüler hergestellt. Dies geschieht, indem die Parallele zur Farbwahrnehmung aufgezeigt wird. Auch dies soll zur Motivation beitragen. Die Zusammensetzung von farbigem Licht ist ein Thema, das ebenfalls in der Physik eine Rolle spielt. In der Physikdidaktik ist bekannt, dass Schüler oft falsche Vorstellung von der Zusammensetzung von Farben haben, da sie Farbaddition und Farbsubtraktion nicht getrennt voneinander betrachten. Aus ihrer persönlichen Erfahrung, beispielsweise aus dem Kunstunterricht, wissen Schüler, dass sich die Farben Gelb und Blau zu Grün mischen lassen. Im RGB-Farbraum stimmen diese Vorstellungen allerdings nicht.[11, S. 42] Um Schülern zu helfen, die richtige Vorstellung zu entwickeln, wurde im Anschluss an den ersten informierenden Text eine Aufgabe gestellt.

Mögliche Denkfehler zur Farbdarstellung

Inhaltliches Vorgehen

Zunächst wurde ein Text verfasst, in dem den Schülern erklärt wird, dass Menschen in Wirklichkeit nur drei verschiedene Farben wahrnehmen können und diese dann erst vom Gehirn zu den unterschiedlichen Farben, die wir sehen, zusammengesetzt werden. Dies wird später als Grundlage dafür verwendet, um zu erklären, warum aus Rot, Grün und Blau das Farbspektrum gemischt werden kann. Im Anschluss daran wurde eine Abbildung verwendet, in der das Mischen der Grundfarben dargestellt wird. Hierbei wurde eine Abbildung gewählt, aus der hervorgeht, zu welchen Farben sich die Grundfarben in gleicher Konzentration mischen. Zu dieser Abbildung wurde erwähnt, dass diese Abbildung auch erzeugt werden kann, indem die Lichtkegel von drei Scheinwerfern in den Grundfarben so ausgerichtet werden, dass sie sich überschneiden. Dies soll später beim Verständnis helfen, dass das Nichtvorhandensein von allen Grundfarben Schwarz ergibt. Alle Grundfarben in maximaler Intensität ergeben jedoch Weiß. Da normalerweise mit weißen Oberflächen gearbeitet wird, die leere Dokumente, Hintergründe oder Ähnliches darstellen, ist es gut möglich, dass die Schüler davon ausgehen, dass die Farbe (0,0,0) für Weiß steht. Genau diese Vorstellung soll an dieser Stelle korrigiert werden. Damit sich die

Wahrnehmung von Farben

Schüler mit dieser Grafik auseinandersetzen, wurde eine Aufgabe gestellt. Schüler sollen hier eintragen, wie verschiedene Farben, aber auch Schwarz und Weiß, aus den Grundfarben eingemischt werden können. Gelb und Grün wurden hier als Farben gewählt, damit den Schülern, falls erforderlich, der Gegensatz zur Farbsubtraktion auffällt. Aus diesem Grund wurde zuerst nach der Zusammensetzung der grünen Farbe gefragt. Gehen Schüler hier davon aus, dass Grün aus Gelb und Blau zusammengesetzt wird, so sollte ihnen spätestens wenn direkt danach nach Gelb gefragt wird, der Fehler auffallen. An der Grafik ist eindeutig zu erkennen, dass Gelb aus Grün und Rot gemischt wird. Die Zusammensetzung von Weiß, Schwarz und Grau sollen die Schüler ebenfalls in dieser Aufgabe erläutern, damit sichergestellt wird, dass sie diese Aussage der Grafik auch durchdacht haben.

Vermeiden von Verständnisschwierigkeiten

Anschließend werden die informatischen Aspekte erläutert. Zunächst wird erklärt, dass durch die bereits erwähnte menschliche Farbwahrnehmung das Darstellen von Farben durch drei Farbkanäle überhaupt erst möglich wird. In einem als wichtig gekennzeichneten Kasten wird erwähnt, welche Notation für Farbwerte gewählt wird und welche Farbtiefe (24-bit) verwendet wird. Die wichtigsten Gedanken wurden hier fett markiert, um den Schülern das Textverständnis zu erleichtern.

Einführen des RGB **Farbraums**

In der anschließenden Aufgabe sollen Schüler diese Informationen anwenden. Damit den Schülern eventuelle eigene Verständnislücken auffallen, sollen sie nun erst Farbwerte schätzen, wobei die angegebene Notation verwendet werden muss. Dann sollen sie sich mit Hilfe eines Tools selbst korrigieren. Sollten hierbei Fehler auffallen, können die Schüler direkt über die Ursachen dafür nachdenken, was ebenfalls noch einmal dazu beiträgt sich gedanklich aktiv mit den Inhalten des Arbeitsblattes zu beschäftigen. Die Selbstkontrolle findet hier mit einem in MS Paint integrierten Tool zur Farbauswahl statt. Auch dies kann dazu beitragen das Verständnis von Farbwerten zu verbessern, da zur Kontrolle die Schüler erst einmal die Farbwerte durch Verschieben der Regler ansteuern müssen. Hierbei wird angezeigt, wie sich die Farbwerte verändern.

Übung/Sicherung zum RGB **Farbraum**

8.2.5. Arbeitsblatt zu zweidimensionalen Arrays und Bildern

Lerninhalte

Auf diesem Arbeitsblatt für die vierte und fünfte Stunde sollen zweidimensionale Arrays und verschachtelte for-Schleifen behandelt werden. Diese sind von zentraler Bedeutung für die Methoden, die im Schülerlabor implementiert werden sollen, also auch für das gesamte

Softwareprojekt. Deswegen ist es notwendig, diese Themen noch vor der Durchführung des Moduls zu behandeln. Anhand der Voraussetzungen für diesen Abschnitt wird auch klar, warum diese Themen als Letztes vor der Durchführung im Schülerlabor behandelt werden.

Zweidimensionale Arrays



Abbildung 8.4.: Lerninhalte zu den zweidimensionalen Arrays

Für das Softwareprojekt im Schülerlabor ist es erforderlich, dass Schüler eine Vorstellung davon haben, wie zweidimensionale Arrays gebraucht werden können um Bilder darzustellen. Das Erzeugen von zweidimensionalen Arrays ist nicht zwingend für das Projekt im Schülerlabor erforderlich, trotzdem ist es im Umgang mit Arrays durchaus wichtig und soll deswegen auch auf dem Arbeitsblatt behandelt werden. Hierzu gehört, dass die Schüler die Syntax der Anweisung dafür verstehen und kennen. Außerdem ist es insbesondere in diesem Kontext wichtig, die Größe des Arrays, also die Höhe und Breite, bezogen auf Bilder, festzulegen. Des Weiteren sollten Zugriffe auf zweidimensionale Arrays erlernt werden. Da dies später über Methoden geschehen soll, die unabhängig von der Größe des Arrays funktionieren sollen, müssen die Schüler in der Lage sein, auf die Längen von zweidimensionalen Arrays zugreifen zu können. Außerdem muss der Zugriff über die zwei Koordinaten erlernt werden.

Lerninhalte zu zweidimensionalen Arrays

Verschachtelte for-Schleifen

Ein Unterthema, das den Schülern hier vermittelt werden soll, ist das Implementieren von verschachtelten for-Schleifen. Hierbei muss den Schülern bewusst sein, wie der Schleifenkopf aufgebaut wird und dass eine for-Schleife in einer anderen ausgeführt werden muss. Darüber hinaus soll den Schülern auch zur Veranschaulichung eine Vorstellung des Ablaufs der verschachtelten for-Schleifen vermittelt werden. Sie

Lerninhalte zu verschachtelten for-Schleifen



Abbildung 8.5.: Lerninhalte zu den verschachtelten for-Schleifen

sollen in der Lage sein, sich den Ablauf anhand eines Beispiels vorzustellen. Hierzu muss den Zählvariablen die entsprechende Raumrichtung zugeordnet werden. Außerdem sollte die Reihenfolge, in der das Bild durchlaufen wird, klar werden. Anhand dessen ist es nachvollziehbarer, was bei der Ausführung von verschachtelten for-Schleifen geschieht. Den Schülern sollte hierbei auch Bewusst sein, bis zu welchen Werten der Zählvariablen die Schleifen durchlaufen werden und wie viele Schritte die innere Schleife bei jedem Schritt der äußeren macht.

Vorwissen

Um diese Lernziele zu erreichen, müssen die Schüler auf jeden Fall den Umgang mit einfachen for-Schleifen beherrschen. Damit die Darstellung von Bildern durch Arrays verstanden werden kann, muss das Konzept der Speicherung von Bildinformationen durch Zahlen bekannt sein. Diese Vorkenntnisse haben die Schüler innerhalb der Reihe bereits auf dem Arbeitsblatt zu for-Schleifen und zur RGB-Farbdarstellung erarbeitet.

Konzeptionelle Überlegungen

Da die Verwendung von zweidimensionalen Arrays für Schüler wahrscheinlich eine Neuerung ist, wurde hier darauf geachtet, das Prinzip der Veranschaulichung anzuwenden. Zweidimensionale Arrays werden hierbei direkt als Bilder eingeführt. So wird den Schülern eine bildhafte Vorstellung der Arrays ermöglicht. Anhand dieser Bilder wird ebenfalls dargestellt, wie ein Bild durch zwei for-Schleifen durchlaufen Bildhafte Vorstellung von Arrays

wird. Diese bildhafte Vorstellung soll das Verstehen des neuen Konzepts begünstigen und erleichtern. Eine Aufgabenstellung soll an dieser Stelle dazu führen, diese bildhafte Vorstellung zu durchdenken und anhand dieser das Konzept von verschachtelten for Schleifen zu begreifen. Dies soll auch hier wieder durch eine aktive gedankliche Auseinandersetzung geschehen.

Zuletzt soll das Konzept noch einmal in einer Methode zur Bildbearbeitung angewandt werden. Dies dient zum Sichern und Vernetzen des Gelernten, sowie zur Selbstkontrolle der Schüler. Darüber hinaus wird dadurch die Anwendung des Gelernten deutlich, was das Gelernte strukturieren und die Motivation aufrechterhalten soll. Die Motivation soll dadurch gestärkt werden, dass die Schüler an dieser Stelle direkt ein erstes Bild bearbeiten.

Entscheidung

Auswahl des

Anwendungsbei-

spiels

gegen **Eclipse**

In dieser Stunde wurde darauf verzichtet, die Farbwerte durch drei Farbkanäle darzustellen. Dies hätte vorausgesetzt, dass im Unterricht Eclipse benutzt wird, was, wenn Eclipse nicht ohnehin im Unterricht verwendet wird, zu unnötigen Schwierigkeiten führen kann. Da viele Schulen durch externe IT-Firmen betreut werden, kann die Installation von Eclipse und den benötigten zusätzlichen Plugins eventuell zu Schwierigkeiten führen, was letztendlich bedeuten würde, dass die Vorbereitung nicht rechtzeitig vor dem Termin der Moduldurchführung im Schülerlabor beendet werden könnte. An den Lernzielen, die zu diesem Arbeitsblatt formuliert wurden, wird deutlich, warum dies von Nachteil wäre. Die Lernziele sind eine wichtige Voraussetzung für das erfolgreiche Durchführen des Moduls. Auch der Umgang mit Eclipse könnte zu Schwierigkeiten führen. Um mit den vorgegebenen Klassen zu arbeiten und eine eigene Methode direkt in die zur Verfügung gestellte Software zu integrieren, müsste das bestehende Softwareprojekt bei jedem Benutzer einzeln importiert und lauffähig gemacht werden. Dies wäre zwar von den Schülern mit Hilfe einer ausführlichen Anleitung durchaus in ein paar Minuten möglich, jedoch würde es vom eigentlichen Unterrichtsgegenstand ablenken. Das Importieren des bestehenden Projekts im Schülerlabor dagegen kann problemlos vor der Durchführung von den betreuenden Lehrkräften durchgeführt werden, wodurch dieses Problem vermieden wird, ebenso kann dort die benötigte Software rechtzeitig installiert werden.

Entscheidung gegen die

Klasse Pixel

Ein weiteres Problem für die Schüler bestünde darin, dass sie neben der Verwendung von verschachtelten for-Schleifen und zweidimensionalen Arrays auch noch gleichzeitig die Verwendung der Klasse Pixel erlernen müssten. Dies könnte dazu führen, dass Schüler überfordert werden. Aus diesem Grund ist es sinnvoll, sich in dieser Stunde auf die Verwendung der for-Schleifen und des Arrays zu konzentrieren. Die Speicherung der Bildinformationen wird durch einen bekannten Datentyp ermöglicht. Es werden einfach Integer-Werte verwendet. Hier muss dann lediglich für die Schüler die Verbindung zu Bildinformationen hergestellt werden.

Inhaltlicher Aufbau

Um an das anzuknüpfen, was die Schüler bereits wissen, wurde gleich zu Anfang die Verbindung zwischen Integer-Werten und Bildern hergestellt. Hierbei wurde zunächst in einem Text erläutert, dass es sich um Schwarzweiß-Werte handelt. Es wäre an dieser Stelle auch möglich gewesen, die Darstellung anstatt auf Schwarzweiß auf einen Farbkanal zu reduzieren. Allerdings hat die bildliche Darstellung von nur einem Farbkanal wenig Praxisbezug, was den Aspekt der Motivation durch die praxisnahe Anwendung nicht berücksichtigen würde. Schwarzweißbilder hingegen kommen in der Praxis häufig vor. Deswegen wurde dieses Beispiel gewählt. Zunächst wird auf dem Arbeitsblatt dargestellt, wie Bildinformationen in einem eindimensionalen Array gespeichert werden können. Diese Analogie wird anschließend auf zweidimensionale Arrays erweitert. Hierzu werden zwei Tabellen gegenübergestellt. In der einen befinden sich Zahlenwerte, in der anderen Tabelle die Bildinformation, die diesen Zahlenwerten entspricht. Hierbei wurde ein möglichst kleines Bild ausgewählt. Bereits mit einer Größe von 6*5 Pixeln ist es möglich, ein einfaches Gesicht darzustellen. Auch diese Überlegung fand in Verbindung mit dem Praxisbezug statt, da es für den Praxisbezug erforderlich ist, keine bedeutungslosen Bildinformationen zu verwenden. Anschließend wird eine Programmzeile angegeben, mit der ein solches Array erzeugt werden kann. Hierbei wurde darauf eingegangen, wie die Größe des Arrays festgelegt wird. Um den Eintrag, der der Höhe des Bildes und dem Eintrag, der der Breite des Bildes entspricht, auseinander halten zu können, wurde hier ein nicht quadratisches Bild verwendet. An dem Bild kann so direkt nachvollzogen werden, welcher Eintrag für die Höhe und die Breite steht, außerdem wird diese Programmzeile noch einmal in einem kurzen Text erläutert.

Konstruktion und Verwendung des Anwendungsbeispiels

Um auf die Koordinaten der einzelnen Pixel sowie die Zuordnung der Zählvariablen zu den Koordinatenachsen einzugehen, wurde das Bild erneut abgebildet. Nun wurden die Zeilen und Spalten beschriftet, außerdem wurden Pfeile angebracht, die die Koordinatenachsen darstellen sollen. Diese wurden mit i und j beschriftet. Außerdem wurde beispielhaft ein Pixel herausgegriffen und dessen Koordinaten angegeben. Die beschriftete Version des Bildes wurde deswegen separat angegeben, da das erste Bild dazu dienen sollte, die Zahlenwerte mit den Bildinformationen in Verbindung zu bringen und so den Schülern eine

Einsatz der Grafiken auf dem Arbeitsblatt

gedankliche Vorstellung zu ermöglichen. Die beschriftete Version sollte jedoch wie bereits erwähnt eher auf die Koordinaten eingeben. Um also die Bilder nicht zu unübersichtlich zu machen und somit die Aussagen unklarer zu machen, wurde das Bild für jede Aussage einzeln verwendet.

An dieser Stelle wurde auch direkt in einem als wichtig gekennzeichneten Kasten die Information gegeben, wie die Schüler auf die Höhe und Breite des Bildes zugreifen können. Dies wird zwar erst für die letzte Aufgabe auf dem Arbeitsblatt benötigt, an dieser Stelle passt es aber inhaltlich am besten auf das Blatt. Die .length Anweisungen können so direkt mit dem beschrifteten Bild verglichen werden. Dies soll auch dazu beitragen, dass Schüler, wenn sie später über den Einsatz der .length Ausdrücke nachdenken, dies mit der bildhaften Vorstellung eines Arrays verknüpfen. Die Anweisungen können so bewusst den richtigen Raumrichtungen zugeordnet werden.

von Höhe und Breite der Bilder

Verwendung

Als vorletzter Teil des erklären Abschnitts wird noch erwähnt, dass zweidimensionale Arrays mit verschachtelten for-Schleifen durchlaufen werden, wobei es sich um eine for-Schleife in einer anderen for-Schleife handelt. Außerdem wurden hier die Zählvariablen der inneren und äußeren for-Schleife zugeordnet, was es Schülern später möglich machen sollte, den Schleifenkopf richtig aufzubauen. Um zu veranschaulichen, wie zwei verschachtelte for-Schleifen ein Array durchlaufen, wurde dies noch in einer Grafik dargestellt. Der Korrektheit halber wurde erwähnt, dass diese Reihenfolge nur dann zutrifft, wenn jeweils die Koordinaten [i][j] in der Schleife angesteuert werden. Das ursprüngliche 5 mal 6 Bild wurde hier erneut, allerdings noch etwas vergrößert, verwendet. Es wurde so vergrößert, dass genug Platz war, um durch Pfeile den Verlauf der Abfragen in der Schleife übersichtlich darzustellen. Damit die Bildaussage richtig verstanden werden kann, wurde hier noch erwähnt, dass die Schleife mit der Zählvariable j jedes Mal vollständig durchläuft bis i erhöht wird. Diese exemplarische bildhafte Darstellung soll auch wieder dazu beitragen, dass Schüler sich den Ablauf der for-Schleifen vorstellen und so besser verstehen können.

Einsatz der verschachtelten Schleifen

Bis zu dieser Stelle enthält das Arbeitsblatt einen großen informierenden Abschnitt. Es ist jedoch keinesfalls davon auszugehen, dass Schüler diesen aufmerksam gelesen und verarbeitet haben. Deswegen wurden noch Aufgaben entworfen, an denen die Kernaussagen des Arbeitsblattes wiederholt und angewandt werden sollen. Zunächst soll das letzte Bild beschriftet werden, sodass die Zählvariable den Raumrichtungen zugeordnet werden. Dies ist bereits zwei Seiten früher in einer Abbildung dargestellt worden, die Schüler sollten sich also entweder daran erinnern oder zumindest zurückblättern können. Außerdem soll zugeordnet werden, welche Raumrichtung durch die innere und welche Raumrichtung durch die äußere Schleife durchlaufen Aufgaben Anwendung der Lerninhalte

wird. Auch dies soll an der Grafik eingetragen werden. Hierdurch werden die Zählvariablen noch einmal den Schleifen zugeordnet, da die Lösungen beider Aufgaben in dieselben Felder nebeneinander eingetragen werden. Die Schüler haben bis zu diesem Zeitpunkt noch nie einen Schleifenkopf einer verschachtelten for-Schleife gesehen. Es wäre also möglich, dass sie an dieser Stelle des Arbeitsblattes noch damit überfordert wären, eine Methode mit einer verschachtelten for-Schleife zu implementieren. Deswegen wurde erst einmal ein Schleifenkopf als Lückentext angegeben. In diesem müssen erneut die Zählvariablen eingetragen werden, was auch wieder eine Anwendung der letzten Aufgabe darstellt. Außerdem müssen die Abbruchbedingungen formuliert werden, was bedeutet, dass die Verwendung der Arraygrenzen noch einmal wiederholt wird. An dieser Stelle könnten bereits die .length Anweisungen verwendet werden, jedoch ist die Aufgabe nicht so gestellt, dass dies gefordert wird. Da hier davon gesprochen wird, dass das Array, wie es bisher auf dem Arbeitsblatt angegeben wurde, verwendet wird, ist davon auszugehen, dass Schüler hier diskrete Zahlenwerte eintragen. Die .length Anweisungen werden daher gezielt am Ende der nächsten Aufgabe abgefragt.

Lückentext zu verschachtelten Schleifen

Bei der zweiten Aufgabe des Arbeitsblattes handelt es sich um eine Programmieraufgabe. Aus den bereits genannten Gründen wurde hier komplett auf die Klasse Pixel und die Software für das Schülerlabor verzichtet. Damit der Stil der Aufgabenstellung im Vergleich zu den Aufgabenstellungen im Schülerlabor konsistent ist, wurde diese Aufgabe möglichst ähnlich gestaltet. Hierzu wurde eine Klasse mit einer Main Methode geschrieben, die die Schüler bearbeiten sollen. In dieser Klasse wird ein Integer Array erzeugt, das dem zweidimensionalen Array auf dem Arbeitsblatt entspricht. Dieses Array wird dann mit einer Methode, die die Schüler erstellen sollen, bearbeitet. Zur Kontrolle wird das Array im Anschluss daran in der Konsole ausgegeben. Die Ausgabe ist so gestaltet, dass das ursprüngliche, unbearbeitete Array neben dem bearbeiteten ausgegeben wird. So können Schüler die Einund Ausgabe vergleichen.

Implementieren der ersten verschachtelten for-Schleife

Die Aufgabe der Schüler besteht nun darin, eine Methode zu implementieren, die das Bild dunkler macht. Dies geschieht durch das Halbieren aller Zahleneinträge in Array. Die Methode soll, genau wie später die Methoden, die im Schülerlabor implementiert werden, in einem vorgegebenen Methodenrumpf implementiert werden. Da nicht sicher davon ausgegangen werden kann, dass die Schüler bereits mit Methoden gearbeitet haben, wird dieser Methodenrumpf in der Aufgabenstellung angegeben und erläutert. Hierbei wurden nur die wesentlichen Aspekte erwähnt, die zur Bearbeitung der Aufgabenstellung erforderlich sind. Diese sind das übergebene Bild sowie die Anweisung zur Rückgabe. Der Begriff "Parameter" wurde weggelassen. Außerdem wurde erwähnt, dass die übergebene Variable in der Methode einfach bearbeitet werden kann. Zum Lösen dieser Aufgabe müssen die Schüler eine verschachtelte for-Schleife verwenden sowie auf die Einträge des zweidimensionalen Arrays zugreifen. Die Kerninhalte des Arbeitsblattes werden hier also noch einmal angewandt.

Die letzte Teilaufgabe dieser Programmieraufgabe besteht darin, die Methode so anzupassen, dass beliebig große Arrays bearbeitet werden können. Hierzu müssen die .length Anweisungen verwendet werden. Wie bereits erwähnt, kann an dieser Stelle durchaus davon ausgegangen werden, dass die for-Schleife zunächst mit diskreten Grenzen implementiert wird. Daher ist diese Aufgabenstellung sinnvoll.

8.3. Didaktische Begründung für Materialien des Moduls

8.3.1. Einführungspräsentation

Lerninhalte

Diese Präsentation soll den Schülern in erster Linie einen Überblick über den Ablauf des Moduls geben. Die Schüler sollen im Voraus wissen, welche Phasen während des Moduls durchlaufen werden und was ihre Aufgaben dabei sind. Außerdem sollen sie einen Ausblick darauf erhalten, was das fertige Softwareprodukt, wenn alle Aufgaben bearbeitet wurden, sein wird. Darüber hinaus sollen sie einen Eindruck davon erhalten, welche Videos erstellt werden können und wie die zu implementierenden Methoden dabei zum Einsatz kommen könnten. All das soll die Arbeit der Schüler während des Moduls unterstützen. Eine klare Zielvorstellung erhöht die Motivation und macht es den Schülern leichter, ihre Arbeit zu planen.

Vorwissen

Für die Einführung in das Modul ist kein Vorwissen notwendig.

Konzeptionelle Überlegungen

Durch den Ausblick auf die fertigen Produkte soll eine Motivierung stattfinden. Darüber hinaus kann durch eine Einführungspräsentation aber auch eine Strukturierung der Lerninhalte erfolgen. Dadurch, dass Schüler eine klare Zielvorstellung haben, können die einzelnen Arbeitsschritte sowie die Lerninhalte einem Zweck zugeordnet werden. Diese können dann in den Augen der Schüler eine innere Ordnung erhalten. Ebenso soll hier bereits eine Kreativitätsförderung stattfinden. Die Schüler erhalten durch die Präsentation eines Beispielvideos die Möglichkeit, erste Vorstellungen und Ideen für ihr eigenes Video zu entwickeln. Da die Schüler an dieser Stelle bereits wissen, dass sie später ein eigenes Video herstellen sollen, ist anzunehmen, dass sie das Beispielvideo im Hinblick darauf betrachten, was sie mit ähnlichen Möglichkeiten in einem eigenen Video darstellen können.

Inhaltliche Vorgehensweise

Gleich zu Beginn wird die fertige Software demonstriert. Dies findet am Smartboard statt, da so die Demonstration von der Lehrkraft gesteuert werden kann. Die Lehrkraft kennt sich mit der Software aus und kann so die wesentlichen Funktionen hervorheben. Aus ähnlichen Gründen wurde auch festgelegt, dass das Video von der Lehrkraft präsentiert werden soll und nicht etwa von den Schülern an den eigenen Arbeitsplätzen angesehen wird. So wird verhindert, dass unnötig viel Zeit in Anspruch genommen wird, da die Lehrkraft die Vorführung zielgerichtet durchführen und danach sofort zum nächsten Punkt übergehen kann. Die restlichen Folien wurden so gestaltet, dass der Ablauf des Moduls vorgestellt wird. Zunächst wird darauf eingegangen, wie im Modul gearbeitet wird, sodass es keine Missverständnisse gibt und die Projektgruppen im Anschluss gebildet werden können. Der restliche Teil der Folien geht auf den Modulablauf selbst ein. Hier wurde mit Overlays gearbeitet, sodass die Schüler die Informationen auf der Folie stückweise erfassen können. Die einzelnen Punkte werden also nach und nach eingeblendet. Außerdem wurde hier zu den meisten Punkten ein Bild hinzugefügt, was die Folie optisch ansprechender wirken lässt, aber auch helfen soll, einen Überblick über die wichtigsten Punkte zu erhalten, selbst wenn der Text nicht gelesen wird.

8.3.2. Arbeitsblatt zur Projektplanung

Lerninhalte

Die Schüler sollen, vor allem wenn sie das noch nie getan haben, mit diesem Arbeitsblatt angeleitet werden, als Gruppe die Aufgaben eines Projektes zu verteilen sowie bei einer Projektarbeit als Gruppe einen eigenen Entwurf des Produktes zu entwickeln. Dieser Entwurf soll hier bereits entwickelt werden, damit er von den Schülern als Ziel für die Reihe angesehen werden kann.

Voraussetzungen

Auf diesem Arbeitsblatt wird von den Schülern erwartet, die ersten Projektaufgaben zu verteilen. Um die Aufgaben einschätzen zu können, ist es daher von Vorteil, eine Vorstellung davon zu haben, wie die Methoden zur Bildbearbeitung in etwa funktionieren. Da die Schüler bereits in der letzten Stunde eine Bildbearbeitung durchgeführt haben, wenn auch nur auf einem Integer Array, ist diese Voraussetzung also erfüllt. In einer weiteren Aufgabe sollen sich die Schüler Gedanken machen, ob noch zusätzliche Methoden zur Bildbearbeitung entworfen und implementiert werden können. Auch hierfür ist es erforderlich, eine Vorstellung davon zu haben, wie diese Methoden funktionieren könnten. Abgesehen von zusätzlichen Methoden, sollen sich die Schüler Gedanken darüber machen, wie das fertige Video aussehen könnte. Um hier keine unrealistischen Erwartungen zu wecken, sollten die Schüler bereits einen Eindruck davon haben, was im Rahmen des Moduls möglich ist. Diesen Eindruck haben die Schüler durch das Vorführen des Beispielvideos erhalten.

Überlegungen im Vorfeld

Das Arbeitsblatt zur Projektplanung wurde entworfen, um den Schülern den Projektgedanken bewusst zu machen. Die Schüler sollten nicht lediglich als Gruppe Aufgaben bearbeiten, sondern sich auch, wenn auch nur in geringerem Ausmaß als in einem richtigen Schulprojekt, absprechen und eigene Ziele festlegen.

Es kann nicht von vorneherein davon ausgegangen werden, dass die Schüler projektartiges Arbeiten gewöhnt sind, deswegen sollte dieses Arbeitsblatt Schüler bei der Projektplanung Schritt für Schritt anleiten. Die grundlegenden Methoden zur Bildbearbeitung müssen auf

Notwendigkeit für ein Arbeitsblatt zur Projektplanung jeden Fall vorgegeben werden, wenn man die geringe Programmiererfahrung der Schüler berücksichtigt. Um trotz dieser Festlegung den Eindruck zu vermitteln, dass bei diesem Projekt Freiheiten zur Gestaltung vorhanden sind, soll bei der Projektplanung bereits über den Entwurf des Videos nachgedacht werden. So können entsprechende Motivation und Kreativitätsförderung sichergestellt werden.

Außerdem war beabsichtigt, bei der Projektarbeit Möglichkeiten zur Differenzierung zu geben. Aus diesem Grund wurden die vorgegebenen Methoden so gewählt, dass sie, was das Implementieren angeht, unterschiedliche Schwierigkeitsgrade besitzen. Außerdem wurde für besonders schnelle Zweiergruppen eine zusätzliche Methode angegeben, die implementiert werden kann, falls die anderen Gruppen noch Zeit benötigen.

Auswahl der Aufgaben

Inhaltliche Strukturierung

In der ersten Aufgabe findet die Aufteilung in der ersten Phase des Projekts statt. In einem kurzen motivierenden Abschnitt wird zunächst erklärt, warum Planung wichtig ist. Hochmotivierte Schüler, die direkt mit dem Programmieren anfangen wollen, oder auch Schüler, die Gruppenarbeiten nicht gewohnt sind oder einfach nicht mögen, könnten bei nicht ausreichender Motivation zur Planung diese als störend oder überflüssig empfinden und einfach überspringen. Unter diesen Umständen würde jedoch nicht das projektartige Arbeiten gefördert werden.

Motivation zur Projektplanung

Anschließend wurden die jetzt zu verteilenden Methoden aufgelistet. Zusätzlich zu dieser Auflistung wurden direkt Beispielbilder angegeben, damit sich die Schüler etwas unter den Beschreibungen vorstellen können. Außerdem wurde zu jeder Methode der Schwierigkeitsgrad angegeben. Die Schüler können also die Aufgaben an dieser Stelle so verteilen, dass die Zweiergruppen, die sich bei der Programmierung sicherer fühlen, die schwierigeren Aufgaben übernehmen. So werden schwächere Gruppen nicht später überrascht, wenn sie merken, dass sie die schwierigste Aufgabe übernommen haben. Hier wurde auch direkt auf die Zusatzaufgabe zur Differenzierung eingegangen, sodass die Schüler dies bei ihrer Planung berücksichtigen können. Bevor die Planung des Videos stattfindet, wurde noch ein Kasten eingefügt, der kompakt darstellt, wie aus Einzelbildern Bewegungsabläufe für das Video zusammengesetzt werden können. Das Prinzip kann zwar bereits bei dem Beispielvideo beobachtet werden, es ist aber nicht sichergestellt, dass die Schüler diese Beobachtung auch durchführen und zu den richtigen Schlüssen kommen. Da die Möglichkeit des Einsatzes von Bewegungsabläufen wesentlich mehr Ideen für das Video ermöglicht,

Verteilen der Aufgaben

Festlegen der Projektziele wurde diese Beschreibung direkt vor der Planung des Videos platziert. Zur Planung des Videos wurden konkrete Punkte genannt, über die die Schüler nachdenken sollten. Es wurde bewusst zuerst dazu aufgefordert, dass über die Zielsetzung des Videos nachgedacht wird, und erst danach über die Frage, welche Fotos dafür gemacht werden sollten. Dies soll verhindern, dass Schüler bei dem Erstellen ihres Videos ziellos Fotos machen und diese nachher ohne ein Konzept zusammen geworfen werden. Dies würde sicherlich zu weniger zufriedenstellenden Ergebnissen führen. Zum Schluss des Arbeitsblattes wurde noch erwähnt, das erst einmal nur zwei oder drei Fotos gemacht werden sollen, um die Methoden zu testen. Dies soll verhindern, dass zu Beginn des Moduls zu viel Zeit damit verbracht wird, etliche Fotos zu machen oder aus dem Internet zu suchen und die Programmierung in den Hintergrund rückt. Im schlimmsten Fall könnte es sonst sogar passieren, dass nicht mehr genug Zeit für das Programmieren vorhanden ist.

8.3.3. Umgang mit Eclipse

Lerninhalte

Es soll im Schülerlabor keinesfalls die Verwendung von Eclipse im Vordergrund stehen. Da Eclipse jedoch genutzt wird, müssen die Schüler einige grundlegende Dinge können. Da anzunehmen ist, dass Eclipse nicht im Unterricht verwendet wurde, muss in jedem Fall die Benutzeroberfläche erklärt werden. Die für das Modul relevanten Elemente der Benutzeroberfläche sind die Aufgaben der einzelnen Fenster und der Button zum Ausführen von Quelltext. Da mit einem vorgegebenen Softwareprojekt gearbeitet wird, müssen die Schüler auch in jedem Fall in der Lage sein, das Projekt zu importieren und danach das Projekt anzupassen, sodass es lauffähig wird. Diese Anpassung muss aufgrund der Verwendung von SWT durchgeführt werden.

Voraussetzungen

Um zu verstehen, wozu die Fenster der Oberfläche benutzt werden, brauchen Schüler zumindest eine geringe Programmiererfahrung. Außerdem sollten die Schüler dem Umgang mit unbekannter Software gegenüber aufgeschlossen sein und ein ausreichendes Maß an Kompetenz besitzen, um sich mithilfe einer Anleitung in einer unbekannten Umgebung zurechtzufinden.

Überlegungen im Vorfeld

Da Eclipse für die Schüler mit großer Wahrscheinlichkeit eine völlig neue Entwicklungsumgebung ist, soll verhindert werden, dass die Verwendung der Entwicklungsumgebung von den eigentlichen Inhalten des Moduls ablenkt. Aus diesem Grund sollte ein Arbeitsblatt entworfen werden, das auf die im Modul relevanten Funktionen von Eclipse eingeht und die Verwendung dieser erklärt. Selbst grundlegende Funktionen, wie etwa das Erstellen einer neuen Klasse, wurden nicht thematisiert, wenn diese nicht für die Durchführung im Schülerlabor notwendig sind. Außerdem sollte diese Anleitung möglichst einfach gestaltet werden, was einschließt, die einzelnen Schritte detailliert durch Bilder darzustellen, sodass diese theoretisch sogar nur anhand der Bilder verfolgt werden könnten.

Inhaltliche Begründungen

Zunächst wurde die Oberfläche erläutert. Hierzu wurde ein Bild der gesamten Oberfläche verwendet, die einzelnen Fenster der Oberfläche mit Zahlen beschriftet und am Rand des Bildes dann die einzelnen Fenster erläutert. Die Erläuterungen können über die Zahlen zugeordnet werden. Die Bezeichnungen der Fenster wurden in den Beschreibungen besonders hervorgehoben, da diese Bezeichnungen im späteren Verlauf des Moduls teilweise noch eine Rolle spielen. Im Anschluss daran wurde kurz und knapp beschrieben, wie Programme getestet werden können. Hierzu wurde ein Bild eines kleinen Teils der Benutzeroberfläche verwendet und der entsprechende Button markiert. Das restliche Arbeitsblatt beinhaltet eine Anleitung zum Import von existierenden Projekten. Hierbei wurden jeweils wichtige Begriffe im Text hervorgehoben und zu jedem einzelnen Schritt Ausschnitte von Screenshots eingefügt. Es wurde darauf geachtet, dass die jeweils relevanten Punkte auf den Screenshots markiert sind, sodass es leichter ist, die Anleitung durchzuführen. Außerdem wurden die Zahlen zum Durchnummerieren der Bilder direkt neben diesen markierten Stellen angebracht, wodurch erzielt werden soll, dass die Aufmerksamkeit der Schüler auf die Markierungen gelenkt wird.

8.3.4. Arbeitsblatt Anweisungen für das Modul

Lerninhalte

Abgesehen von der Verwendung der for-Schleifen, die bereits in der vorbereitenden Unterrichtsreihe behandelt wurden, ist der Umgang mit der Klasse Pixel von zentraler Bedeutung, um die Aufgaben im Schülerlabor erfolgreich bearbeiten zu können.



Abbildung 8.6.: Lerninhalte zu den Anweisungen für das Modul

Für die Bearbeitung der Aufgaben müssen die Schüler die Getund Set-Methoden verwenden. Auf diese muss über die einzelnen Pixel-Objekte zugegriffen werden. Damit die Schüler sich nicht noch zusätzlich Gedanken machen müssen, Werte die kleiner als 0 oder größer als 255 sind abzufangen, wurde die Klasse Pixel so gestaltet, dass diese unerlaubten Werte abgefangen werden. Damit die Schüler die Werte nicht trotzdem selbst abfangen, sollte ihnen diese Tatsache bekannt sein.

Eine weitere Neuerung im Vergleich zu der vorbereitenden Unterrichtsreihe ist der Umgang mit Pixel-Arrays, deren Einträge Objekte vom Datentyp Pixel sind. Bisher wurden lediglich Arrays aus primitiven Datentypen verwendet, deren Werte direkt verändert werden konnten. In diesem Fall sollen jedoch über die Arrayeinträge Methoden aufgerufen werden. Solche Aufrufe sollen die Schüler ebenfalls verwenden können.

Abfangen unerlaubter Eingaben

Besonderheiten bei Objekt-Arrays

Voraussetzungen

Die Schüler sollten zu diesem Zeitpunkt den Umgang mit Arrays beherrschen. Außerdem sollten die Schüler die Farbdarstellung durch die Grundfarben Rot, Grün und Blau kennen, um überhaupt den Sinn der

Get- und Set-Methoden zu verstehen.

Erfahrung mit der Objektorientierung wäre von Vorteil, um mit der Klasse Pixel umzugehen. Da dies zu einem früheren Zeitpunkt in der Einführungsphase jedoch nicht vorausgesetzt werden kann, wurde das Arbeitsblatt so gestaltet, dass Erfahrung in der Objektorientierung nicht zwingend vorausgesetzt wird.

Konzeptionelle Überlegungen

Da nicht davon ausgegangen werden kann, dass Objektorientierung bereits im Unterricht behandelt wurde, sollte dieses Blatt nicht einfach als Dokumentation der Klasse gestaltet werden. Ein UML-Diagramm sollte deswegen beispielsweise nicht verwendet werden. Ebenso sollten die Begriffe "Objekt" und "Klasse" auf dem Arbeitsblatt vermieden werden. Zu allen Methoden, die auf dem Blatt erläutert werden, mussten die Anweisungen daher so angegeben werden, dass sichergestellt wird, dass die Schüler in ihrem eigenen Programmcode trotzdem die Notation "Objekt.Methode" anwenden. Außerdem musste ersichtlich werden, wie die Anweisungen im Programmcode angewandt werden können.

Inhaltliche Vorgehensweise

Zunächst wurden die im Modul benötigten Methodenaufrufe tabellarisch dargestellt, wobei deren Zweck erläutert wurde. Hierbei wurde, um Missverständnisse zu vermeiden, durch einen Pfeil noch mal darauf hingewiesen, dass der Teil der Anweisung, der vor dem Punkt steht, der Name der Variable ist. Auf den Begriff "Objekt" wurde hier, wie bereits erwähnt, bewusst verzichtet.

Anschließend wurde in einem Kasten darauf hingewiesen, was programmintern passiert, wenn unerlaubte Farbwerte zugewiesen werden.

Damit die Schüler die hier beschriebenen Anweisungen richtig anwenden können, wurden hier noch Beispiele angegeben. Im ersten Beispiel wurde der Rotwert eines Pixels ausgelesen und einer Integervariable zugewiesen. Da die Schüler bereits mit Integervariablen gearbeitet haben, lässt sich so ein Bezug zu dem bereits Bekannten herstellen, wodurch die Beispielprogrammzeile besser verstanden werden kann. Anschließend wurden diese Programmzeilen noch so abgeändert, dass der Zugriff über ein Array stattfindet. So sollte schrittweise zu dem korrekten Verständnis dieser Programmzeile hingeführt

Darstellung der Anwendung anhand Beispielprogrammzeilen werden, indem von bekanntem ausgegangen wird, und dies schrittweise so verändert wird, dass letztendlich eine Programmzeile verstanden werden kann, wie sie auch bei der Bearbeitung der Aufgaben zur Anwendung kommt.

Zuletzt wurde noch erklärt, wie ein neues Pixelarray erzeugt werden kann. Zur Bearbeitung der vorgegebenen Aufgaben ist dies nicht erforderlich. Ein neues Array zu erzeugen kann jedoch bei der Implementierung eigener Ideen der Schüler hilfreich sein. Hier war es erforderlich, den Begriff "Objekt" zu verwenden. Es handelt sich jedoch ohnehin um einen optionalen Teil, der wahrscheinlich nur von fachlich sehr guten Schülern angewandt wird. Diese könnten noch Zeit haben und motiviert sein, eigene Methoden zu entwickeln. Diese werden wahrscheinlich nicht von dem Begriff "Objekt" verunsichert.

Für die restlichen Schüler besteht der Vorteil in diesem Kasten darin, dass noch einmal eine verschachtelte for-Schleife abgebildet wird. Dies kann bei der Erinnerung an die vorherige Stunde hilfreich sein. Zur Erinnerung an die vorherige Stunde wurde hier auch noch ein weiterer kleiner Kasten angebracht, der darauf hinweist, welchen Koordinatenachsen die Zählvariablen entsprechen. Die Erklärung der benötigten Anweisungen zusammen mit diesen Rückblicken zur letzten Stunde sollten die Schüler eigentlich vollständig ausrüsten, um die ersten Methoden zur Bildbearbeitung zu implementieren.

Erzeugen von Objekt-Arrays

Hilfestellung zu verschachtelten Schleifen

8.3.5. Die Aufgabenblätter zu den Methoden zur Bildbearbeitung

Lerninhalte

Da die Projektgruppen die Aufgaben unter sich aufteilen sollen und die Aufgaben daher parallel bearbeitet werden, sind sich die Aufgabenstellung in dieser Phase sehr ähnlich.

In allen drei Aufgaben soll das Durchlaufen des Arrays noch einmal wiederholt werden. Außerdem muss in allen Methoden die Klasse Pixel verwendet werden. Diese Methoden müssen in einem dafür vorgegebenen Methodenrumpf implementiert werden, wobei Schüler die Parameter und Rückgabedatentypen beachten müssen.

Bei der Bearbeitung der Aufgabe zur Farbumkehr lernen die Schüler darüber hinaus, wie das Inverse eines Farbwerts gebildet wird. Während der Aufgaben zum Farbfilter lernen die Schüler gezielt einzelne Farbwerte zu manipulieren. Bei der Aufgabe zum Spiegeln des Bildes, was die schwerste der drei ist, lernen die Schüler Koordinaten

Lerninhalte zu allen vorgegebenen Bildbearbeitungsmethoden

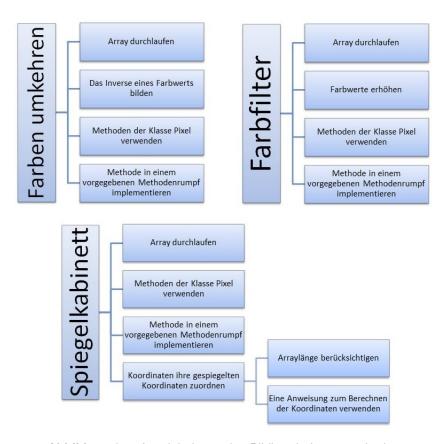


Abbildung 8.7.: Lerninhalte zu den Bildbearbeitungsmethoden

zu transformieren. Die Koordinaten eines Pixels müssen den jeweils gespiegelten Koordinaten zugeordnet werden. Hierzu ist es erforderlich, die Arraylänge zu berücksichtigen und eine Anweisung zu formulieren, mit der die Koordinaten auf die gespiegelten Koordinaten abgebildet werden. Während der Zusatzaufgabe, in der eine Methode zum Erstellen eines Schwarzweißbildes implementiert werden soll, lernen die Schüler einem beliebigen Farbwert die entsprechende Graustufe zuzuordnen.

Voraussetzungen

Um die Arbeitsblätter erfolgreich bearbeiten zu können, müssen die Schüler in jedem Fall mit zweidimensionalen Arrays und verschachtelten for-Schleifen umgehen können. Was das Programmieren angeht, müssen darüber hinaus auch die Methoden der Klasse Pixel angewandt werden können. Außerdem muss das Prinzip von RGB-Farbwerten verstanden worden sein. Diese Voraussetzungen sind jedoch durch die

Unterrichtsstunde zu Farbwerten in der vorbereitenden Unterrichtsreihe, die Schulstunde in der Unterrichtsreihe zu zweidimensionalen Arrays und die Hinweisen zu den Anweisungen für das Modul gegeben.

Konzeptionelle Überlegungen

Was das Aufteilen der Aufgabenstellungen und die Aufgabenstellungen selbst betrifft, so ist hierzu bereits einiges im Abschnitt zu dem Arbeitsblatt für die Projektplanung erwähnt worden. Darüber hinaus ist allerdings zu erwähnen, dass bei diesen Aufgaben die Schüler möglichst selbstständig arbeiten sollten. Aus diesem Grund sollten zu den Aufgaben nur die Hilfestellungen gegeben werden, die zum Lösen unbedingt erforderlich sind. Zum Veranschaulichen und der Funktionsweise der einzelnen Methoden sollten auf allen Arbeitsblättern Beispielbilder eingefügt werden. Auf diese Weise kann sichergestellt werden, dass die Schüler die Aufgabenstellung verstehen und vor allem ihre Lösungen selbst vergleichen und kontrollieren können. Innerhalb der Aufgabenstellungen wurden auch die Methodenrümpfe abgebildet. Dies soll verhindern, dass es zu Verwechslungen kommt.

Grad des selbstständigen Arbeitens

Die Lösungen der Gruppen sollten jeweils zusammengetragen werden. Bei der Planung der Unterrichtsreihe wurde hierbei für die Verwendung von Dropbox entschieden. Die Dropbox ist leicht zu verwenden und der Austausch von Daten kann unkompliziert und schnell stattfinden. Dies ist insbesondere hilfreich, wenn die Projektgruppen später ein Video erstellen und währenddessen gemeinsam auf viele Dateien zugreifen müssen. Anstelle von Dropbox eine Lernplattform zu verwenden, würde hier wahrscheinlich dazu führen, dass die Arbeit erheblich verlangsamt würde. Da es später sinnvoll ist, dass jedes der Zweierteams innerhalb der Gruppe alle Funktionen der Bildbearbeitungssoftware verwenden kann, sollen also nach diesem Abschnitt die Quelltexte soweit ausgetauscht worden sein, dass dies auch der Fall ist.

Entscheidung für Dropbox

Inhaltliche Vorgehensweise

Arbeitsblatt zur Farbumkehrung

Zunächst wurde kurz auf den Zweck einer solchen Methode eingegangen und dann die Aufgabenstellung formuliert. Zusätzlich zur Aufgabenstellung wurde der Hinweis gegeben, wie das Umgekehrte eines Farbwertes gebildet werden kann, da dies weder in der Unterrichtsreihe zur Vorbereitung noch im Modul thematisiert wurde. Es könnte

den Schülern zwar auch möglich sein, selbst darauf zu kommen, allerdings ist es unwahrscheinlich, dass alle Schüler selbst darauf kommen können. Die zweite Aufgabenstellung fordert die Schüler lediglich dazu auf, ihren Quelltext hochzuladen, wobei der Name der Datei und das Dateiformat vorgegeben werden, um hier ein Chaos oder eventuell einen Datenverlust der Projektgruppen zu verhindern, was sich sicherlich negativ auf die Motivation auswirken würde.

Arbeitsblatt zum Farbfilter

Die Formulierung der Aufgabe entspricht vom Prinzip her der auf dem Arbeitsblatt zum Umkehren der Farben. Allerdings wird hier nicht vorgegeben, wie die Farbwerte verändert werden sollen. Die Schüler sollen also selbst herausfinden, inwiefern die Farbwerte verändert werden sollten. Dieser offenere Arbeitsauftrag soll dazu beitragen, etwas Planungsfreiheit zu erzeugen, wie es in Projekten gewünscht ist. Auch auf diesem Arbeitsblatt wurde dazu aufgefordert, die Methode nachher in die Dropbox hochzuladen.

Arbeitsblatt zur Spiegelung

Auch hier wurde die Aufgabe analog zu den anderen Arbeitsblättern gestellt. Es ist jedoch nicht unbedingt anzunehmen, dass die Schüler mit dem Spiegeln der Koordinaten problemlos zurechtkommen. Aus diesem Grund wurde hier ein Hilfskasten erstellt, in dem ein Array dargestellt wird. Die Anträge wurden so beschriftet, dass ihre Position im Array abzulesen ist. Hier wurde noch einmal erwähnt, was die .length Anweisung für dieses Array ergeben würde. Zu einem Eintrag des Arrays wurde erwähnt, welche Koordinate die hierzu gespiegelte Koordinate wäre. Dieses Beispiel soll den Schülern helfen, den Ausdruck zum Spiegeln der Koordinaten zu formulieren. Auf die Länge des Arrays wurde eingegangen, da Schüler diese Länge für den Ausdruck benötigen. Außerdem wurde hier konkret die Länge dieses Arrays genannt, sodass beim Durchrechnen des Beispiels Schlüsse gezogen werden können, die dabei helfen, einen Ausdruck zu formulieren, der nicht zum Überschreiten der Grenzen des Arrays führt. Direkt unter dem Kasten wurde etwas Platz gelassen, damit die Schüler zum Spiegeln der Koordinaten Zeichnungen machen können. Dieser Platz wurde auch als solcher gekennzeichnet. Als letzte Aufgabe des Arbeitsblattes steht auch hier wieder die Aufforderung, die funktionierende Methode hochzuladen.

Arbeitsblatt zum Schwarzweißbild

Dieses Arbeitsblatt entspricht vom Aufbau her auch den restlichen Arbeitsblättern zu den Methoden für die Bildbearbeitung. Zu der Umwandlung von Farbwerten zu Graustufen wurde hier ein Hinweis gegeben, wie diese Umwandlung stattfinden könnte. Dieser Hinweis wurde gegeben, da es sich um eine Bonusaufgabe handelt und die Gruppe, die diese bearbeitet nicht so viel Zeit damit verbringen soll, das sie sonst eventuell zu weit hinter ihre Projektgruppenmitglieder zurückfällt.

8.3.6. Arbeitsblatt zum Hintergrundtausch

Lerninhalte

Bereits an den Lernzielen wird deutlich, dass sich das Implementieren dieser Methode wesentlich von den bisher implementierten Methoden unterscheidet.

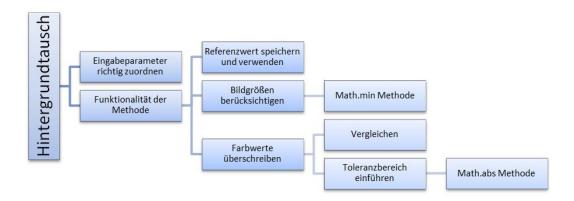


Abbildung 8.8.: Lerninhalte des Arbeitsblattes zum Hintergrundtausch

Die Schüler lernen hier mit Methoden zu arbeiten, die mehrere Eingabeparameter haben. Weitere Lernziele werden deutlich, wenn betrachtet wird, was die Schüler zum Implementieren der Methode alles anwenden müssen. Beispielsweise müssen die Schüler zu Anfang ihrer Methode einen Referenzwert speichern und diesen später immer wieder verwenden. Dies ist eine Vorgehensweise, die für viele Problemstellungen geeignet ist. Außerdem lernen Schüler den Umgang mit der "Math"Klasse, die bei der Programmierung mit Java eine wichtige Rolle spielt. Sie lernen außerdem, bei dem Vergleichen von Werten einen Toleranzbereich zu verwenden. Da die Größen beider Bilder berücksichtigt werden müssen, wird auch das Einhalten von Arraygrenzen noch einmal vertieft.

Voraussetzungen

Voraussetzungen sind hier der sichere Umgang mit zweidimensionalen Pixelarrays und der Klasse Pixel allgemein. Da die Methode eine wesentlich höhere Komplexität besitzt als die bereits implementierten Methoden, sollten die Schüler diese Grundlagen beherrschen, da das Implementieren dieser Methode schon bei Beherrschen der Grundlagen eine Herausforderung darstellt. Werden die Grundlagen nicht beherrscht, ist es wahrscheinlich nicht möglich, die Methode ohne die Hilfe anderer erfolgreich zu implementieren. Die Schüler haben an dieser Stelle bereits Methoden zur Bildbearbeitung implementiert, in denen sie zweidimensionale Pixelarrays verwendet haben. Die Voraussetzungen sollten also zu diesem Zeitpunkt des Moduls erfüllt sein.

Konzeptionelle Überlegungen

Da das Implementieren dieser Methode die Möglichkeit bietet vieles zu lernen, wurde festgelegt, dass diese Aufgabe von jedem Zweierteam bearbeitet werden soll.

Inhaltliche Vorgehensweise

Zunächst wurde kurz begründet, warum jede Zweiergruppe diese Methode einzeln implementieren soll. Da die Schüler bis zu diesem Punkt das Implementieren der Methoden aufgeteilt haben, könnten die Schüler es sonst eventuell als störend empfinden, diese Methode aus Sicht der Projektgruppe dreimal implementieren zu müssen. Erst nach diesem Hinweis fängt das eigentliche Arbeitsblatt an.

Das Arbeitsblatt beginnt auch hier wieder mit einem Bild, an dem die Schüler sehen können, welche Ergebnisse die Methode liefern soll. Auch hier wurde wieder, wie bei den anderen Arbeitsblättern zu den Methoden, zunächst ein kurzer motivierender Text verfasst, wozu diese Form der Bildbearbeitung benutzt werden kann. Anschließend wurde der Methodenrumpf abgebildet, in dem die Methode implementiert werden soll.

Aufgrund der Komplexität der zu implementierenden Methode werden die Schüler durch Teilaufgaben angeleitet. Zunächst werden die Schüler aufgefordert, ein Foto vor dem Bluescreen aufzunehmen, damit die Methode getestet werden kann. Anschließend wird erklärt, welcher der Parameter für das Bild vor dem Bluescreen und welcher der

Schrittweises Anleiten zur Methode Hintergrundtausch Parameter für den neuen Hintergrund steht. Die nächste Teilaufgabe fordert dazu auf, den Referenzwert zu speichern. Dies geschieht bereits an dieser Stelle, da verhindert werden soll, dass sich die Schüler mit dem Speichern des Referenzwertes beschäftigen, wenn bereits die for-Schleife implementiert wird. Der Referenzwert muss gespeichert werden, bevor irgendetwas an dem Bild verändert wird und es kann nicht davon ausgegangen werden, dass die Schüler an dieser Stelle von alleine zu diesem Schluss kommen. Außerdem wurde erwähnt, dass die Farbwerte in Integervariablen gespeichert werden sollten. Dies sollte verhindern, dass die Schüler hier ein Pixel-Objekt als Referenzwert speichern, dessen Einträge dann bei dem Austauschen des Hintergrundes überschrieben werden. Dies würde verhindern, dass die Methode funktioniert, da der Referenzwert verändert wird. Da mit großer Wahrscheinlichkeit die Schüler noch nicht objektorientiertes Programmieren erlernt haben, ist nicht anzunehmen, dass ihnen dieser Fehler auffallen würde.

Als nächstes wird auf die Größe der Arrays eingegangen, wobei die Verwendung der Math.min Methode erklärt wird. Es ist nicht anzunehmen, dass die Schüler diese Methode bereits kennen. Ohne diese Methode kann der von den Schülern erstellte Quelltext jedoch schnell unübersichtlich werden. Dies wäre der Fall, wenn die for-Schleifen nicht einfach bis zum Minimum in der Längen laufen würden, sondern je nachdem, welches Array größer wäre, unterschiedliche for-Schleifen ausgeführt würden. Dies würde passieren, wenn Schüler versuchen, das Problem mithilfe von if-Verzweigungen zu lösen. Da diese Verzweigung für die Höhe und die Breite durchgeführt werden müsste, müssten die Schüler also vier verschiedene Fälle bearbeiten. Dies wäre eine mögliche Fehlerquelle und könnte den Ablauf der Aufgabe behindern. Bei der Berechnung des Toleranzwertes wurden ähnliche Hinweise gegeben, da dies zu genau denselben Problemen führen könnte. Außerdem wurde noch ein Hinweis gegeben, wann genau die Farbwerte der Pixel ausgetauscht werden sollten und wann nicht.

Auf diesem Arbeitsblatt wurde ebenfalls wieder Platz für Skizzen gelassen, da hier auch wieder mit mehreren Arrays unterschiedlicher Größe gearbeitet wird, und Zeichnungen eventuell den Schülern helfen könnten die Methode zu implementieren.

8.3.7. Arbeitsblatt Soundbearbeitung

Lerninhalte

Als weitere Anwendung für Arrays und for-Schleifen wird im Modul die Soundbearbeitung verwendet. Die Schüler können hierbei lernen, wie Audiodateien in Double-Arrays gespeichert werden können und

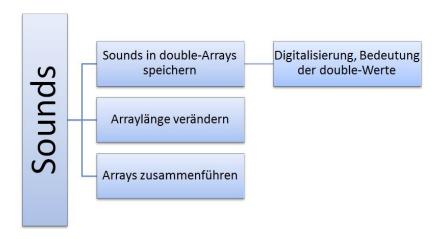


Abbildung 8.9.: Lerninhalte bei der Bearbeitung von Audiodateien

welche Bedeutung die Einträge der Arrays haben, also die Digitalisierung von Tönen. Außerdem können die Schüler, je nach Aufgabenstellung lernen, wie die Länge eines Arrays verändert wird oder wie zwei Arrays zusammengeführt werden.

Voraussetzungen

Voraussetzungen sind lediglich der Umgang mit eindimensionalen Arrays, der während der vorbereitenden Unterrichtsreihe und des Moduls definitiv in ausreichendem Maß behandelt wurde, sowie die Verwendung eines vorgegebenen Methodenrumpfes. Die Voraussetzungen sollten also in Anbetracht in der bisherigen Arbeitsmaterialien durchaus erfüllt sein.

Konzeptionelle Überlegungen

Die Verwendung von Sounds ist aus zwei Gründen sinnvoll. Erstens ist es einfach noch einmal eine andere Anwendung für Arrays im Bereich Medien. Zweitens können die auf diese Weise bearbeiteten Audiodateien benutzt werden, um das Video der Gruppe zu erstellen. Da die im Modul verwendete Software nur .wav-Dateien mit einem Kanal bearbeiten kann, müssen Audiodateien, die die Schüler verwenden möchten, in dieses Format umgewandelt werden. Audacity wurde zu diesem Zweck ausgewählt, da Audacity frei verfügbar ist und außerdem von den Schülern auch verwendet werden kann, um eigene Sounds aufzunehmen.

Entscheidung für Audacity

Alle drei Methoden, die mithilfe dieses Arbeitsblattes implementiert werden sollen, bieten interessante Lerngelegenheiten. Trotzdem wurde entschieden, die drei Methoden wieder innerhalb der Gruppe aufzuteilen. Dies bedeutet zwar einen Nachteil, was das Vermitteln der Lerninhalte angeht, allerdings passt diese Vorgehensweise besser in das Gesamtkonzept des Moduls. Die Zielsetzung soll aus Schülersicht das Erstellen der Software und anschließend eines eigenen Videos sein. Der Hintergrundtausch wurde bereits von jeder Zweiergruppe einzeln implementiert, um nicht zu viele Aufgaben zu stellen, die nicht als Projektgruppe bearbeitet werden, sollte daher an dieser Stelle wieder das projektartige Arbeiten im Vordergrund stehen. Die Methode zum Umdrehen von Sounds wurde gewählt, da sich rückwärts abgespielte Sounds gut bei dem Erstellen des Videos verwenden lassen. Auf die Methoden zum Aneinanderhängen von Audiodateien und zum Abschneiden von Teilen einer Audiodatei trifft dies ebenfalls zu, allerdings sind diese auch unter dem Aspekt der Veränderung von Arraylängen interessant.

Auswahl der Aufgabenstellungen

Inhaltliche Vorgehensweise

Zunächst wurde eine kurze Information darüber gegeben, wie Audiodateien mit Audacity umgeformt werden können. Auch hier wurde wieder ein Screenshot verwendet, um die Anleitung möglichst leicht verständlich zu gestalten. Anschließend wurde kurz die Speicherung von Audiodateien in Double-Arrays erläutert. Dabei wurde direkt darauf eingegangen, dass die Methoden so gestaltet werden sollen, dass ein Ausgabe-Array erstellt wird, in das Einträge aus dem Eingabe-Array übertragen werden. Dies erleichtert später die Aufgaben in denen die Länge des Arrays verändert wird, da hierzu in jedem Fall ein neues Array mit angepasster Länge erzeugt werden muss. Um dieses Prinzip leichter verständlich zu machen, wurde auch hier wieder ein Bild erstellt, auf dem dargestellt wird, wie die Einträge aus dem Eingabearray in das Ausgabearray übertragen werden. Dies soll zur Veranschaulichung dienen.

Veränderung der Arraylänge

Im Anschluss wurde ein Kasten erstellt, in dem die Bedeutung der Einträge des Double-Arrays bei der Tonspeicherung erklärt wird. Auch hier wurde wieder ein Bild eingefügt, um das Prinzip leichter verständlich zu machen. Dieser Kasten wurde als Zusatzinformation gekennzeichnet, da diese Information nicht zwingend erforderlich ist, um die Aufgaben zu bearbeiten und sonst eventuell Schüler verwirrt werden, da sie denken, sie müssten diese Information irgendwo in der Aufgabe anwenden.

Arrays zum Speichern von Audiodateien

In dem Kasten, in dem die eigentliche Aufgabe gestellt wird, wird auch

Aufgabenstellungen zu Audiodateien wieder kurz begründet warum, neben der Bildbearbeitung, auch die Tonbearbeitung für den weiteren Projektverlauf wichtig ist. Zu den Methoden selbst wird nur kurz erläutert, was die zu implementierenden Methoden können sollen, und welche Eingabeparameter dafür zur Verfügung stehen. Auf ausführlichere Erläuterungen wurde verzichtet, da das Bearbeiten von Audiodateien unkomplizierter ist als das Bearbeiten von Bildern. Die Schüler sollten also nach der Bildbearbeitung auch ohne größere Hilfestellungen in der Lage sein, diese Aufgaben zu bearbeiten. Auch hier wurde wieder ein Kasten zum Anfertigen von Zeichnungen eingefügt, in dem die Schüler zum besseren Verständnis Skizzen anfertigen können.

8.3.8. Hilfsblatt zur Videobearbeitung

Lerninhalte

Die Schüler sollen in der Lage sein, aus vorhandenen Bildern und Tonspuren ein Video zusammenzusetzen.

Voraussetzungen

Für die Benutzung des Windows Movie Makers sind keine Vorkenntnisse nötig.

Konzeptionelle Überlegungen

Für die Bearbeitung der Videos wurde als Software der Windows Movie Maker ausgewählt. Dieser bietet die benötigten Funktionen und hat dabei eine sehr unkomplizierte und intuitive Benutzeroberfläche. Aus diesem Grund ist eine ausführliche Anleitung nicht erforderlich. Bilder und Tonspuren können einfach über die entsprechende Schaltfläche importiert werden, es reicht sogar die Dateien einfach auf der Oberfläche des Movie Makers abzulegen. Lediglich das Anpassen der Zeit, die die einzelnen Bilder angezeigt werden, sollte erklärt werden, da die entsprechende Schaltfläche eventuell sonst übersehen wird.

Inhaltliche Vorgehensweise

Um sicherzustellen, dass wirklich die richtigen Schaltflächen zum Importieren von Bildern und Tonspuren gefunden werden, wurde ein Screenshot abgebildet, auf dem diese Schaltflächen markiert wurden. Außerdem wurde ein Screenshot der gesamten Benutzeroberfläche abgebildet. Auf diesem Screenshot wurde markiert, in welchem Feld die Zeit, die ein Bild angezeigt wird, eingestellt werden kann. Außerdem wurde die Fläche markiert, in der die importierten Bilder aufgelistet sind und zu dieser erwähnt, dass Bilder, die bereits eingefügt wurden, beliebig oft kopiert und wieder eingefügt werden können. So müssen Schüler mehrfach verwendete Bilder nicht jedes Mal erneut importieren.

8.4. Didaktische Begründung für die Materialien zur Nachbereitung

8.4.1. Unterrichtsstunde zur Glaubwürdigkeit von Medien in der Schule

Lernziele

Schüler sollen sich eine fundierte Meinung über die Auswirkungen der Medienmanipulation bilden. Sie sollen außerdem einen Eindruck davon erhalten, inwieweit auch Nachrichtenmedien manipuliert sein können. Abgesehen von Bildern soll auch die Manipulierbarkeit von Videos betrachtet werden.

Vorwissen

Um sich eine fundierte Meinung über Medienmanipulation bilden zu können, sollten die Schüler wissen, welche Möglichkeiten die Informatik zur Medienmanipulation bietet. Da die Schüler im Schülerlabor gerade eigene Methoden zur Manipulation von Medien implementiert haben, sollte dieses Vorwissen vorhanden sein.

Inhaltliche Vorgehensweise

Zunächst werden nacheinander zwei Bildausschnitte desselben Bildes besprochen. Das eine Bild vermittelt den Eindruck, dass ein Soldat vor dem Ertrinken gerettet wird. Das andere vermittelt den Eindruck, dass derselbe Soldat mit einer Waffe bedroht wird. Dies soll den Schülern zeigen, dass selbst die Wahl eines Bildausschnittes die Aussage eines Bildes verändern kann. Auf zwei weiteren Folien werden zwei Beispiele angeführt, in denen Bilder bewusst von einer Zeitung manipuliert wurden, um sensationellere Nachrichten zu verbreiten. Diese Beispiele wurden gewählt, um den Schülern zu zeigen, dass Nachrichten teilweise wirklich gezielt manipuliert werden, die Möglichkeiten zur Medienmanipulation also nicht nur existieren, sondern auch genutzt werden, um Falschnachrichten zu verbreiten.

Präsentation zu den gesellschaftlichen Auswirkungen

Um zu zeigen, wie stark auch Videos manipuliert werden können, wird im Anschluss daran ein manipuliertes Video gezeigt, das auf YouTube gefunden wurde. Das Besondere an diesem Video ist, dass es auf YouTube dazu ein weiteres Video gibt, in dem gezeigt wird, wie das Video manipuliert wurde. So sollen die Schüler einen Eindruck davon erhalten, wie stark Videos manipuliert werden können, indem sie sehen, wie stark sich das Endergebnis von dem Ausgangsmaterial unterscheidet.

Im Anschluss daran soll eine Gruppendiskussion stattfinden. Die Schüler sollen in Vierergruppen Argumente dafür finden, warum die Medienmanipulation entweder eine Bereicherung oder eine Gefahr ist, dafür stehen ihnen fünf Minuten zur Verfügung. Ziel dieser fünf Minuten soll sein, Argumente zu finden, sodass das Thema anschließend im Kurs diskutiert werden kann. In dieser anschließenden Diskussion werden die Hauptargumente gesammelt, so dass die Schüler einen Überblick über die Hauptargumente erhalten. Aufgrund der zuvor gezeigten Beispiele sollten die Schüler motiviert sein, sich an dieser Diskussion zu beteiligen. Nach dem Sammeln von Argumenten in den ersten fünf Minuten der Diskussion sind bereits alle Schüler eingebunden, was ebenfalls die Bereitschaft der Schüler, sich an der Diskussion mit dem gesamten Kurs zu beteiligen, verbessern soll.

Gruppendiskussion zu den gesellschaftlichen Auswirkungen

Kapitel 9 Fazit

Es war möglich, anhand eines motivierenden Kontextes, der Medienmanipulation, ein breites Spektrum an Lernzielen abzudecken. Die Lernmaterialien konnten dabei so gestaltet werden, dass sie für Informatikkurse geeignet sind, die erst ein paar Wochen mit Java programmieren. Die Zielgruppe sind also Schüler des ersten Jahres in der Oberstufe. Durch die im Rahmen der Hausarbeit entworfene Lerneinheit können die Schüler Arrays, Schleifen, die Digitalisierung von Bildund Toninformationen, Arbeitsweisen bei einem Softwareprojekt und die gesellschaftlichen Auswirkungen der Medienmanipulation mit Informatiksystemen kennenlernen.

Da nicht nur ein Modul für das Schülerlabor entworfen wurde, sondern auch eine Unterrichtsreihe, in die dieses eingebettet ist, war überhaupt erst genügend Raum vorhanden, um diese Lerninhalte umfangreich in der Lerneinheit zu behandeln. In der vorbereitenden Unterrichtsreihe werden bereits in der Schule die Grundlagen erlernt, die dann während des Softwareprojektes im Schülerlabor angewandt und vertieft werden. Durch die Verwendung einer vorbereitenden Unterrichtsreihe konnte sichergestellt werden, dass die Schüler, die das Modul im Schülerlabor durchführen, alle in etwa denselben Wissensstand haben und das Modul erfolgreich absolvieren können. Außerdem konnte das Vermitteln der Grundlagen aus dem Modul ausgelagert werden, was freieres, projektartiges Arbeiten im Schülerlabor möglich macht.

Vorteile der Einbettung in eine Unterrichtsreihe

Bei Betrachtung der Zeitplanung in der Lerneinheit und der Arbeitsmaterialien wird klar, dass durch das Verwenden dieses Kontextes kein großer Zeitverlust innerhalb der Lerneinheit, beispielsweise durch das Einführen des Kontextes, entsteht. Der Kontext wird in diesem Fall allmählich, parallel zu den Lerninhalten, eingeführt. Das für das Schülerlabor vorgesehene Projekt wurde so geplant, dass die Projektarbeit durch die Schüler zeitlich sehr effizient durchgeführt werden kann.

Trotzdem konnte den Schülern durch das Projekt gestalterischer Freiraum geboten werden. Die Schüler implementieren vorgegebene Methoden zur Bild- und Tonbearbeitung. Sie erstellen im Rahmen des Gestaltungsfreiräume trotz geringer Vorkenntnisse Projektes ein Video, wozu sie die implementierten Methoden verwenden sowie bei Bedarf weitere implementieren können. Da die Schüler noch über wenig Erfahrung in der Programmierung verfügen, konnte nicht viel Freiraum bei dem eigentlichen Softwareprojekt geboten werden. Um die Schüler trotzdem zu motivieren, wurde daher einfach mehr Freiraum bei einem anderen Bereich des Projekts, dem Erstellen des Videos, gegeben.

Durch die Verwendung des Kontextes Medienmanipulation entsteht jedoch im Vergleich zu in der Ausarbeitung eines Unterrichtes, in dem dieselben informatischen Inhalte ohne Kontext unterrichtet werden, ein größerer Aufwand für die Lehrkraft. In das Thema Bild- und Tonbearbeitung muss eine entsprechende Einarbeitung stattfinden. Dadurch, dass im Rahmen dieser Hausarbeit eine umfangreiche Lerneinheit zu diesem Thema erstellt wurde, können Lehrkräfte dieses Problem umgehen. Die Arbeitsmaterialien sind so gestaltet, dass eine Einarbeitung der Lehrkraft in das Thema nicht erforderlich ist.

Es war auch möglich, abwechslungsreiche Lernmaterialien zu verwenden, was durch die Verwendung eines Kontextes erleichtert wurde. Die vorbereitende Unterrichtsreihe beginnt mit einer Präsentation, in der das Thema Medienmanipulation vorgestellt wird. Darauf folgt erst einmal eine Unterrichtsstunde, in der die Schüler Raster- und Vektorgrafiken spielerisch kennenlernen. Hierbei werden Bilder entweder durch Beschreiben ihrer Formen oder Angabe von Farben und Koordinaten an andere Schüler vermittelt. Darauf folgt ein Abschnitt, in dem die Schüler Arrays und Schleifen kennenlernen, wobei sie Programmieraufgaben lösen sollen. Um erst einmal wieder einen Kontrast zu den Programmieraufgaben zu schaffen, lernen die Schüler danach erst einmal RGB Farbwerte kennen, wobei nicht programmiert, sondern die Zusammensetzung und Einschätzung von Farbwerten im Vordergrund steht. Im letzten Abschnitt der vorbereitenden Unterrichtsreihe werden zweidimensionale Arrays und verschachtelte for-Schleifen behandelt, wobei auch darauf geachtet wurde, dass nicht nur Programmieraufgaben, sondern auch andere Aufgabenformen zum Einsatz kommen.

Das Modul selbst bildet auch wieder einen Kontrast zu der vorbereitenden Unterrichtsreihe, in der die Schüler in Partnerarbeit Aufgabenblätter bearbeiten. Im Modul wird dagegen durchgehend als Projektgruppe gearbeitet. Hierbei wird nicht nur auf Arbeitsblättern auf die einzelnen zu implementierenden Methoden eingegangen, sondern auch auf die Projektplanung selbst. Außerdem steht neben dem Programmieren das Erstellen eines Videos im Vordergrund, was auch wieder für Abwechslung sorgt.

Kontexte begünstgen Abwechslungsreiche Lernmaterialien Die Unterrichtsstunde zur Nachbereitung unterscheidet sich auch wieder von den bis dahin durchgeführten Lernabschnitten. In dieser Stunde wird als Kurs über die gesellschaftlichen Auswirkungen der heutigen Möglichkeiten zur Medienmanipulation diskutiert.

Auffällig ist, dass die Abwechslungen bei den Phasen der Lerneinheit jeweils dadurch möglich wurden, dass ein Kontext verwendet wurde. Zwischen den Phasen, in denen Programmiert wurde, konnten jeweils Phasen eingebaut werden, in denen sich die Schüler mit kontextbezogenen Inhalten beschäftigen.

Die Arbeitsmaterialien wurden insgesamt so gestaltet, dass die Schüler nicht überfordert werden. Dies setzt voraus, dass die Schüler nicht mit unnötigen Problemen konfrontiert werden, wie beispielsweise der Umwandlung von Dateien in Arrays und umgekehrt. Hierzu bot sich an, die Methoden in vorgefertigten Methodenrümpfen implementieren zu lassen. Auf diese Weise können bei der Programmierung die Lerninhalte der Stunden im Vordergrund stehen, da nicht unnötig Zeit verbraucht wird, in der die Schüler selbst das Programmgerüst erstellen.

Auf das Wesentliche reduzierte Aufgaben

Um Programmieraufgaben auf diese Weise zu stellen, muss also ein Programmgerüst vorgegeben werden. In diesem Fall wurde ein umfangreiches Programmgerüst in Form der Medienmanipulationssoftware vorgegeben. Dies hat auch wieder einige Vorteile mit sich gebracht. Dadurch, dass die Schüler ihre Methoden mit einer vorgegebenen grafischen Benutzeroberfläche testen und verwenden können, wird das Testen erleichtert sowie die Motivation gesteigert.

Vorteile der Verwendung einer vorgefertigten Software

Dieses Vorgehen, nur die für die Lerninhalte relevanten Programmteile implementieren zu lassen, lässt sich grundsätzlich im Informatikunterricht einbauen, da die oben genannten Vorteile unabhängig vom verwendeten Kontext bestehen. Es ist allerdings auch noch aus einem weiteren Grund von Vorteil, Schüler auf diese Weise programmieren zu lassen. In der hier beschriebenen Lerneinheit wurde nicht auf objektorientierte Programmierung eingegangen. Da der richtige Umgang mit Methoden bei objektorientierter Programmierung zwingend erforderlich ist, wird also durch frühzeitiges Verwenden von Methoden ein nahtloserer Übergang zur Objektorientierung möglich. Auch bei weniger komplexer Software ist diese Art der Aufgabenstellung möglich. Die Aufgabe zu zweidimensionalen Integer-Arrays in der letzten Stunde der vorbereitenden Lerneinheit ist ein Beispiel dafür. Programmieraufgaben zu stellen, in denen Programme lediglich vervollständigt werden, sollte also in jedem Fall bei der Unterrichtsplanung in Betracht gezogen werden.

Vorbereitung der Objektorientierung

Insgesamt kann daher gesagt werden, dass die in dieser Hausarbeit beschriebene Lerneinheit auf die Schüler sehr motivierend wirken dürfte und bei der Durchführung effizient die Programmierkenntnisse der Schüler verbessern sollte.

Kapitel 10 Ausblick

Es soll keinesfalls der Eindruck vermittelt werden, dass die Lerneinheit als vollständig optimiert angesehen werden sollte. Die behandelten Inhalte bieten noch vieles an Erweiterungsmöglichkeiten. Die Bearbeitung von Audiodateien ist beispielsweise nur sehr kurz thematisiert worden. Dieses Thema könnte ausführlicher behandelt werden, wobei es auch möglich ist, Audiodateien in ähnlicher Weise zu verändern, wie es in der Lerneinheit mit den Bildern getan wurde. Die Digitalisierung von Tönen bietet außerdem die Möglichkeit, das Thema Digitalisierung allgemein anzusprechen. Hierbei könnten Digitalisierungsfehler, Speicherbedarf und die technische Umsetzung thematisiert werden. Es wäre möglich, mit Hilfe von Mikrofonen und Microcontrollern Töne aufzunehmen und anschließend zu speichern. Dies würde alles in die Richtung der Technischen Informatik laufen und könnte leicht so umfangreich werden, dass daraus ein eigenes Modul für das Schülerlabor erstellt werden könnte.

Erweiterungsmöglichkeiten zur Bearbeitung von Audiodateien

Es wäre ebenfalls möglich gewesen, die GUI-Programmierung mehr zu thematisieren. Die grafische Oberfläche der verwendeten Software wurde mit Hilfe von Jigloo entwickelt. Hierbei handelt es sich um ein Plugin für Eclipse, mit dem sehr einfach Benutzeroberflächen gestaltet werden können. Die Anwendung dieses Plugins ist ähnlich der des Android App Inventors. Da sehr viel mehr Funktionen zur Verfügung stehen, ist die Anwendung von Jigloo etwas anspruchsvoller. Trotzdem kann so das Erstellen von grafischen Benutzeroberflächen auf eine schülergerechte Weise stattfinden, wenn ausreichend angeleitet wird.

Erweiterungsmöglichkeiten zur GUI-Programmierung

Eine weitere Möglichkeit, die Lerneinheit zu erweitern, wäre den Aspekt des Softwareprojekts auszubauen. Im Buch "Didaktik der Informatik" wurde darauf eingegangen, wie Softwareprojekte im Informatikunterricht gestaltet werden können. Vergleicht man diese Ausführung mit dem Softwareprojekt innerhalb dieser Unterrichtsreihe, so fallen die Erweiterungsmöglichkeiten auf.[9, S. 306-308] Beispielsweise wäre es möglich, bei der Implementierung der Software

Erweiterungsmöglichkeiten zur Projektorientierung weniger Vorgaben zu machen und die Schüler dafür mehr in Problemanalyse und Entwurf einzubinden. Dies könnte dadurch erreicht werden, dass statt der Methoden das Thema eines Videos vorgegeben wird (Beispiel: Werbespot) und die Schüler dann selbst bestimmen, welche Methoden implementiert werden. Außerdem wäre es möglich, die Rollenverteilung während eines Softwareprojekts in das Modul einzuarbeiten. Die Aufgaben, die innerhalb der Gruppe, eventuell auch mehrfach, verteilt werden könnten, sind: der Rechnerbeauftragte, der Projektüberwacher, der Schnittstellenbeauftragte, der Tester, der Dokumenteur und der Kümmerer.[9, S. 306-308]

Eine Möglichkeit, die Lerneinheit etwas umzustrukturieren, ohne den Umfang zu erhöhen, wäre den Schwerpunkt von der Bildbearbeitung etwas in Richtung der Bearbeitung von Audiodateien zu verlagern. Da die Bearbeitung von Audiodateien auf eindimensionale Arrays begrenzt ist, wäre es möglich, diese zunächst als Kontext zu verwenden. Hierbei müsste Zeit in der vorbereitenden Unterrichtsreihe für die Einführung in den Kontext verwendet werden, wobei dafür ein Teil der Einführung in die Bildbearbeitung in das Modul verschoben werden könnte.

Erweiterungsmöglichkeit: früheres Einbinden von Audiodateien

Die didaktischen Begründungen für diesen derzeitigen Aufbau der Lerneinheit wurden ausführlich thematisiert. Erweiterungsmöglichkeiten sind also so zu verstehen, dass sie vor allem bei anderer Schwerpunktsetzung oder dem Ausarbeiten einer Lerneinheit von größerem Umfang zum Einsatz kommen könnten.

Anhang A Literaturverzeichnis

Printed References

- [5] Peter Hubwieser. *Didaktik der Informatik*. 3. Aufl. Heidelberg: Springer Verlag, 2007.
- [9] Andreas Schubert Sigrid; Schwill. *Didaktik der Informatik*. 1. Aufl. München: Spektrum Verlag, 2004.
- [11] Martin Hopf (Hrsg.) Hartmut Wiesner Horst Schecker. *Physikdidaktik kompakt*. 1. Aufl. Hallbergmoos: Aulis Verlag, 2011.

Online References

- [1] Bildungsstandards Informatik für die Sekundarstufe 1. URL: https://www.gi.de/fileadmin/redaktion/empfehlungen/Bildungsstandards_2008.pdf (besucht am 28.07.2014).
- [2] Contextualized Support for Learning Homepage. URL: http://home.cc.gatech.edu/csl/1 (besucht am 28.07.2014).
- [3] Einführung in die Programmierung in JAVA Ein Leitprogramm in Informatik. URL: http://schuelerlabor.informatik.rwth-aachen.de/sites/default/files/materialien/Java-Programmierung%20-%20Leitprogramm.pdf (besucht am 28.07.2014).
- [4] Gesellschaft für Informatik Meldungen. URL: https://www.gi.de/aktuelles/meldungen/detailansicht/article/rekordhoch-bei-informatik-erstsemestern-4-mehrstudienanfaenger-in-der-informatik.html (besucht am 28.07.2014).
- [6] Java Wav File IO The Lab Book Pages. URL: http://www.labbookpages.co.uk/audio/javaWavFiles.html (besucht am 28.07.2014).

- [7] Kernlehrplan Informatik Gymnasiale Oberstufe. URL: http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/upload/klp_SII/if/GOSt_Informatik_Endfassung.pdf (besucht am 28.07.2014).
- [8] Media Computation Teachers Website. URL: www.mediacomputation. org (besucht am 28.07.2014).
- [10] Vorgaben Abitur 2016 Informatik. URL: https://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15 (besucht am 28.07.2014).

Anhang B Unterrichtsmaterialien



InfoSphere Schülerlabor Informatik der RWTH Aachen

Vorstellung des Moduls

Medienmanipulation



Was ist Medienmanipulation?



- ▶ Medienmanipulation ist das verändern von Medien. Alltägliche Medien sind:
 - Bilder



Sounds



Videos(=mehrere Bilder und Sounds)



Wo begegnet uns MM?



▶ Überall!



(c) Sarah Morris



(c) copyright Blender Foundation | www.bigbuckbunny.org



Quelle(beide) Bildblog.de



3 InfoSphere – Schülerlabor Informatik

RWTHAAC

Was Manipulieren wir?



- ▶ Bilder
- ▶ Sounds



Und zwar mit selbst geschriebener Software.

▶ Diese fügen wir dann zu einem Video zusammen.
 (Deswegen eine kleine Vorschau...)



Was ist das Ziel der Reihe?



▶ Das schreiben eigener Programme zur Medienmanipulation. Diese werden in eine GUI integriert, sodass diese sogar Praxistauglich sind.

(Demonstration...)



InfoSphere – Schülerlabor Informatik



Voraussetzungen MM



▶ Selbst Programme zur Medienmanipulation schreiben?

Was denkt ihr? Was müsst ihr dafür können?



Diskussion. Überlegt euch als 2er-Gruppe <u>in 2 Minuten mindestens</u> <u>5 Fähigkeiten</u>, von denen ihr denkt, dass ihr diese brauchen werdet.



Überblick über das Modul



- In der Schule:
 - ▶ Wie werden Farben und Bilder gespeichert?
 - ▶ Wie durchläuft ein Programm ein Bild?
 - Ihr erstellt eine erste Methode zur Bildbearbeitung.
 - Ihr integriert diese Methode in die vorhandene GUI

7 InfoSphere – Schülerlabor Informatik



Überblick über das Modul



- ▶ Im Schülerlabor:
 - Verschiedene Methoden zur Bildbearbeitung Programmieren
 - Verschiedene Methoden zur Soundbearbeitung Programmieren
 - Anwendung: Aus euren Bildern/Sounds ein Video erstellen



Überblick über das Modul



- ▶ In der Schule, nach dem Schülerlabor:
 - Nachbesprechung: Wie glaubwürdig sind Medien?

RWTHAACHE

InfoSphere – Schülerlabor Informatik

Medienmanipulation Grafikformate



Vektor- und Rastergrafiken



In dieser Stunde werdet ihr Vektor- und Rastergrafiken kennenlernen. Dieses Arbeitsblatt enthält Arbeitsaufträge, die euch helfen sollen, die Konzepte der Formate kennenzulernen.

Damit heute bei dem Kennenlernen der Grafikformate keine Langeweile aufkommt, veranstalten wir eine Zeichenstunde.



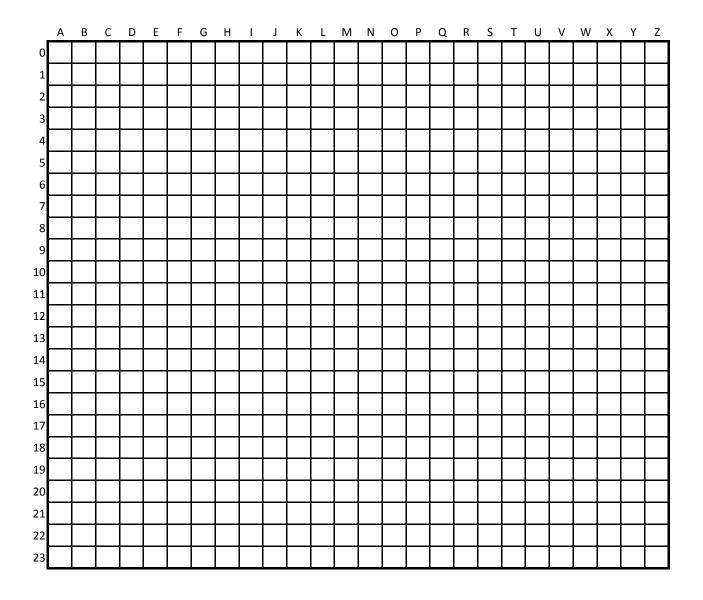


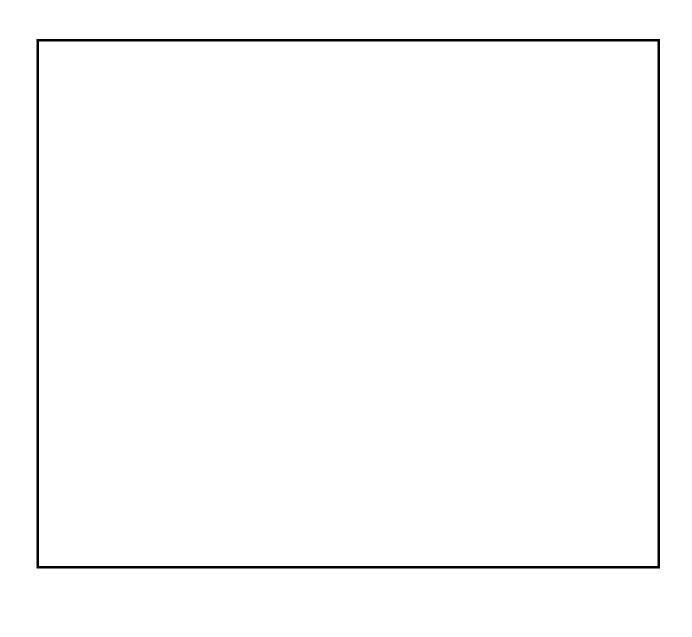
- 1. Kontrolliert, ob ihr alles habt, was ihr braucht! Im Detail sind das:
 - a) Eine Packung Stifte
 - b) Einen Umschlag für jeden. Öffnet diese noch nicht!

Auf dem Umschlag stehen jeweils ein Buchstabe und eine Zahl. Pro Gruppe müssen diese beide unterschiedlich sein. Wenn das nicht der Fall ist, meldet euch!

- 2. Sobald das Startzeichen kommt, öffnet derjenige von euch, der den Umschlag A hat, diesen. Zeigt eurem Partner nicht das Bild. Gebt ihm die leere Vorlage.
 - a) Ihr habt jetzt **genau** 3 Minuten Zeit, eurem Partner das Bild so akkurat zu beschreiben, dass dieser es auf seiner Vorlage möglichst genau farbig zeichnen kann.
 - b) Wartet dann bis zum nächsten Startsignal und tauscht die Rollen. Jetzt ist Umschlag B dran.
- 3. Einige andere Schüler in der Klasse haben die gleiche Nummer auf ihrem Umschlag und damit das gleiche Bild. Nehmt das gezeichnete Ergebnis eures Partners mit, findet eure Nummer und unterhaltet euch über eure jeweilige Vorgehensweise.
 - a) Wie seid ihr vorgegangen?
 - b) Was war unterschiedlich?
 - c) Welche Methode hat das bessere Ergebnis erzielt?
 - d) Wieso war diese Methode geeigneter?
 - e) Welche Vorteile haben die einzelnen Methoden?

Schreibt eure Ergebnisse auf, wir werden sie gleich im Anschluss in der Klasse diskutieren!





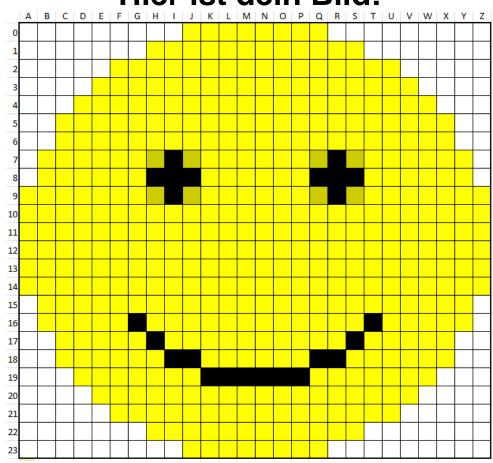
InfoSphere World of Informatics

Medienmanipulation Grafikformate

Neben diesem Bild findest du im Umschlag die Zeichenvorlage für dein Gegenüber.

Orientiere dich bei der Beschreibung am Raster!

Hier ist dein Bild:

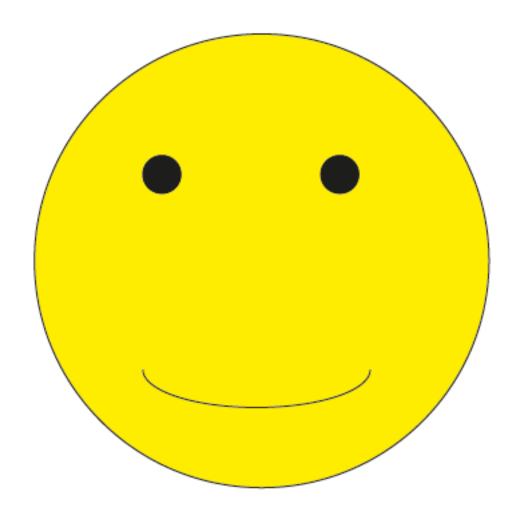


Nenne das Motiv nicht!

Medienmanipulation Grafikformate



Hier ist dein Bild:



Erwähne nicht um welches Objekt es geht!

Beachte folgende Regeln:
Beschränke dich bei deiner Beschreibung auf
Farben und geometrische Formen (Kreis,
Ellipse, Linie,...).

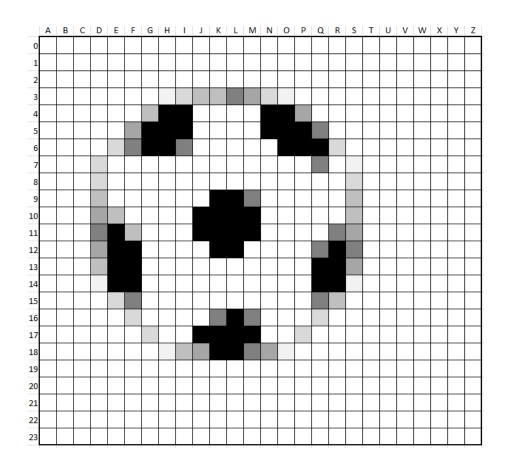
InfoSphere World of Informatics

Medienmanipulation Grafikformate

Neben diesem Bild findest du im Umschlag die Zeichenvorlage für dein Gegenüber.

Orientiere dich bei der Beschreibung am Raster!

Hier ist dein Bild:



Nenne das Motiv nicht!

Medienmanipulation Grafikformate



Hier ist dein Bild:



Erwähne nicht um welches Objekt es geht!

Beachte folgende Regeln:
Beschränke dich bei deiner Beschreibung auf
Farben und geometrische Formen (Kreis,
Ellipse, Linie,...).

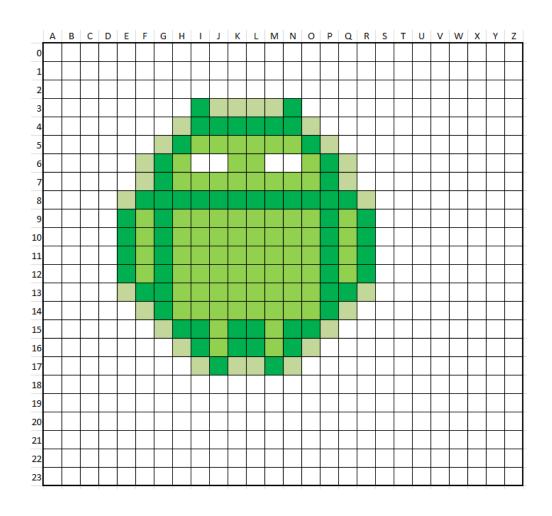
InfoSphere World of Informatics

Medienmanipulation Grafikformate

Neben diesem Bild findest du im Umschlag die Zeichenvorlage für dein Gegenüber.

Orientiere dich bei der Beschreibung am Raster!

Hier ist dein Bild:



Nenne das Motiv nicht!

Medienmanipulation Grafikformate



Hier ist dein Bild:



Erwähne nicht um welches Objekt es geht!

Beachte folgende Regeln:
Beschränke dich bei deiner Beschreibung auf
Farben und geometrische Formen (Kreis,
Ellipse, Linie,...).



InfoSphere Schülerlabor Informatik der RWTH Aachen

Quiz zu Grafikformaten

Infosphere



Das Quiz



Auf den nächsten Seiten seht ihr verschiedenste Grafiken.

Welches Format ist jeweils das geeignetere, Raster oder Vektor?



Fotos





InfoSphere – Schülerlabor Informatik

Fotos



▶ Rastergrafiken.

Fotos bestehen nicht aus fest definierten, geometrischen Formen. Und davon, dass Computer in Fotos zuverlässig beliebige Objekte erkennen und in ihre geometrischen Formen zerlegen sind wir noch weit entfernt.

Stichwort: Künstliche Intelligenz



Logos und Formen





InfoSphere – Schülerlabor Informatik



Logos und Formen



Richtig, das Vektorformat.

Warum? Es lässt sich beliebig vergrößern. Ein Bild sagt mehr als 1000 Worte:

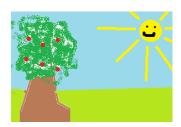


InfoSphere – Schülerlabor Informatik



"Kunstwerke"





7 InfoSphere – Schülerlabor Informatik



"Kunstwerke"



Eine Rastergrafik!

Hier wäre eine Vektorgrafik an der Grenze ihrer Möglichkeiten, denn Freihandformen lassen sich nur sehr schwer vektoriell ausdrücken!

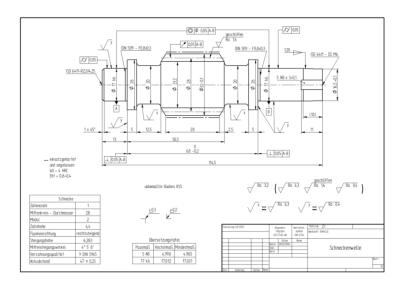


InfoSphere – Schülerlabor Informatik



Technische Zeichnungen





InfoSphere – Schülerlabor Informatik

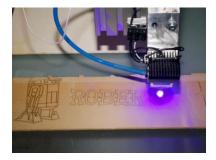


Technische Zeichnungen



Bitte nur als Vektorformat!

Da Produktionsmaschinen Wege abfahren, müssen sie diese als Vektoren auslesen können!





Drucksachen





11 InfoSphere – Schülerlabor Informatik



Drucksachen



Die typische Rastergrafik!

Gängige Drucker fahren das Papier Zeile für Zeile ab. Vektorgrafiken müssen hierfür zunächst in Rastergrafiken umgewandelt werden!





Medienmanipulation Arrays und for-Schleifen





Ihr habt das Konzept einer Rastergrafik kennengelernt. Jetzt wollt ihr Bilder aber nicht nur darstellen, sondern auch bearbeiten. Hierfür benötigt ihr eine geeignete Möglichkeit, Grafiken zu speichern, um die Rastergrafiken mit dem Computer zu bearbeiten. Außerdem braucht ihr eine Schleife um diese durchlaufen zu können. Ihr werdet auf diesem Arbeitsblatt also...

- ...Arrays kennenlernen. In diesen können die Farbwerte der Rastergrafik gespeichert werden.
- ...lernen, diese mit einer for-Schleife zu durchlaufen.

Arrays - Grundlagen

Arrays sind Listen mit mehreren Einträgen. Diese Einträge müssen alle denselben Datentyp haben. (z.B. int, double, String, etc.) Der Datentyp der Daten ist festgelegt, sobald das Array deklariert wird. Die Anzahl der Einträge des Arrays wird beim Erzeugen festgelegt.

Hier ein Beispiel für ein Integer Array:

Das Array "zahlenarray" hat die Länge 3 und den Datentyp int[].

Die Einträge sind also von 0 bis "Arraylänge-1" durchnummeriert.

	zahlenarray				
0	7				
1	8				
2	255				

Arrays – Deklarieren und Erzeugen

Deklarieren:

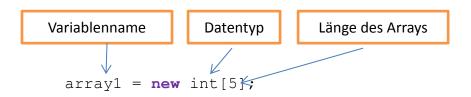
Wie alle Variablen müssen Arrays zuerst im Programmcode deklariert werden. (Hierdurch wird erst mal nur festgelegt, dass es das Array gibt.) Das geschieht mit einer Anweisung der Form:



Datentyp (z.B. int, double, String) legt den Datentyp der Einträge fest. "name" ist der Name des Arrays. Die Eckigen klammern bedeuten, dass Variable "name" ein Array ist.

Erzeugen:

Arrays sind in Java Objekte und müssen daher auch erzeugt werden. Das ist über eine Anweisung der folgenden Form möglich, wenn das Array bereits deklariert ist.



Medienmanipulation Arrays und for-Schleifen



"name" ist der Name des Arrays, wie er in der Deklaration gewählt wurde. Auch der **Datentyp** muss derselbe wie bei der Deklaration sein. "new" bedeutet, dass hier ein neues Objekt erzeugt wird. Die **Länge** des Arrays muss eine ganze Zahl sein und kann nachher nicht mehr verändert werden.

Arrays – Auf die Einträge zugreifen

Prinzipiell greift man auf die Einträge eines Arrays aus zwei verschiedenen Gründen zu: um Einträge zu verändern oder sie auszulesen. Einen einzelnen Eintrag steuert man auf folgende Weise an:

Variablenname

Stelle des Arrays, auf die zugegriffen wird

Array1[2]

Dies kann jetzt wie eine Variable des entsprechenden Datentyps verwendet werden.



Genau diese Zugriffe führen häufig zu Fehlern und Programmabstürzen. Es muss gelten: $0 \le index \le l \ddot{a}nge - 1$. Wenn im Programm versucht wird, einen anderen Index zu verwenden, stürzt das Programm mit einer "Array index out of bounds exception" ab, da es auf eine Stelle des Array zugreift, die es nicht gibt.

Mögliche Operationen auf einem int-Array mit Namen array1 wären:

Wert ändern:

array1[1] = 2; //Setzt den zweiten Eintrag auf 2.

Wert verwenden:



- 1. Schreibt jetzt ein Programm, in dem:
 - a) Ein Integer-Array der Länge 5 erzeugt wird.
 - b) Den einzelnen Einträgen des Arrays unterschiedliche Werte zugewiesen werden.
 - c) Die Werte der Reihe nach ausgegeben werden.

Außerdem:

- d) Skizziert euer Array auf einem Blatt Papier. Nummeriert dabei die Einträge und gebt die Datentypen der Einträge an.
- e) Begründet: Warum kann auf den Eintrag [5] nicht zugegriffen werden? Was passiert, wenn man es dennoch versucht?

Medienmanipulation Arrays und for-Schleifen



For-Schleifen

Mit Arrays der Länge 5 lässt sich so eine Aufgabe noch bewältigen, aber würdet ihr auf dieselbe Art folgende Aufgabenstellung lösen wollen?



- ? Stellt euch vor, ihr müsstet ein Programm schreiben in dem:
 - a) Ein Integer-Array der Länge 100 erzeugt wird.
 - b) Den einzelnen Einträgen des Arrays die Werte 1 bis 100 zugewiesen werden.
 - c) Die Werte der Reihe nach ausgegeben werden.

Um das Problem zu lösen, muss eine Kontrollstruktur verwendet werden, eine for-Schleife.

Eine for-Schleife führt eine Operation so oft wie erwünscht durch und zählt dabei mit, wie oft die Operation schon durchgeführt wurde. In einem Programm könnte eine for-Schleife so aussehen:

```
for(int i=0; i<5; i++) {
        System.out.println(i);
}</pre>
```

int i=0; \rightarrow Zählvariable i<5; \rightarrow Abbruchbedingung



- 2. Schreibt ein Programm, das diesen Code-Schnipsel ausführt. Findet heraus...
 - a) ...welche Zahlenwerte ausgegeben werden.
 - b) ...bis zu welchem i der Teil in der Schleife ausgeführt wird. Wie nennt man die Bedingung dafür, dass die Anweisung in der Schleife ausgeführt wird?
 - c) ...wie man i nennt (siehe Arbeitsblatt).
 - d) ...was i++ bedeutet.
 - e) ...wo in der for-Schleife der Anfangs- und der Endwert für i steht.
 - f) Außerdem: Verwendet eine for-Schleife, um die Einträge eures Arrays aus Aufgabe 1 auszugeben.

1)	 		
)	 	 	
)			
)		 	
.)	 	 	







Dieser Test soll eine Lernkontrolle vor allem für euch selbst sein. Löst die Aufgaben deswegen ohne Hilfsmittel und jeder für sich alleine.



- 1. Ergänze den Code auf diesem Blatt so, dass...
 - a) ...ein Array mit dem Inhalt "Ford", "Arthur", "Marvin" erzeugt wird.
 - b) ...der Inhalt spaltenweise ausgegeben wird.
- 2. Korrigiert euch als 2er-Gruppe gegenseitig.

	//Array deklarieren
L	
Г	//Array erzeugen
L	
Г	//Werte in das Array eintragen
L	//Das Array mit einer for-Schleife ausgeben
Γ	

Medienmanipulation Farbdarstellung





Um Bilder darzustellen, müssen erst einmal Farben dargestellt werden. Auf diesem Arbeitsblatt wird gezeigt, wie Farben durch Zahlenwerte repräsentiert werden. Dabei bietet es sich leider an, euch einige Illusionen bezüglich eures Sehvermögens und der Mischung von Lichtfarben zu nehmen. Da muss aber jeder einmal durch.

Wie wir Farben sehen

Aus eigener Erfahrung wissen wir, dass man als Mensch die unterschiedlichsten Farben sehen kann. Leider ist das nur die halbe Wahrheit. In Wirklichkeit kann jeder von uns nur 3 verschiedene Farben sehen. Diese sind Rot, Grün und Blau. Für andere Farben hat unser Auge einfach keine Rezeptoren. Aber wieso sehen wir dann so viele Farben? Das Gehirn (und teilweise auch schon das Auge) berechnet die Farben aus den Rot-, Grün- und Blau-werten.

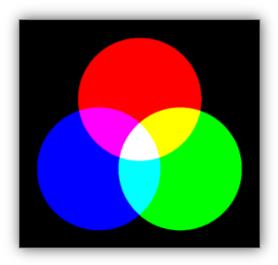


Abbildung 1: Farbmischung. Quelle: metacolor.de

Auf Abbildung 1 seht ihr, wie sich die Grundfarben mischen. Dasselbe Bild lässt sich auch durch einen roten, grünen und blauen Scheinwerfer erzeugen. Die Lichtfarben addieren sich, deswegen spricht man von Farbaddition.



- 1. Wieviel wisst ihr? Testet euer Wissen über Farben und notiert euer Ergebnis.
 - a) Wie erhält man diese Farben durch Farbaddition aus den Grundfarben?
 Grün:

Gelb:

Weiß:

Schwarz:

Grau:

Medienmanipulation Farbdarstellung



Was das mit Informatik zu Tun hat

Unser Auge kann also nicht unterscheiden, ob es gelbes Licht, oder eine Mischung aus grünem und rotem Licht sieht. (Das Licht ist nicht dasselbe, sieht aber identisch aus.) Daher reicht es auch, Farben einfach nur durch ihren **Rot-, Grün- und Blauwert** zu speichern. Das spart erheblich Speicherplatz, da 3 **Farbkanäle** (Farbanteil einer Grundfarbe als Zahlenwert) natürlich weniger Speicherplatz benötigen, als etwa 10, oder vielleicht 100 Farbkanäle.



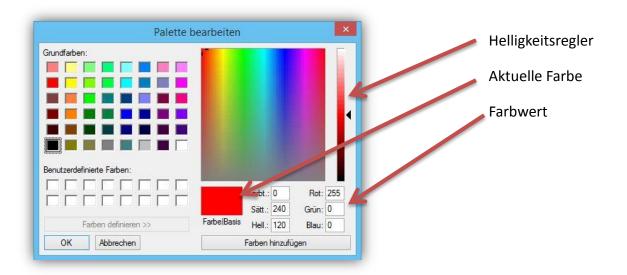
Farbwerte werden durch ihre Anteile an Rot, Grün und Blau definiert. Schwarz wird beispielsweise durch (0,0,0) dargestellt, was etwa bedeutet, dass alle Kanäle "dunkel" sind.

Wir Arbeiten mit der Üblichen **Farbtiefe** von **24Bit, also 8Bit je Farbe**. $(2^8 = 256)$ Jeder Farbkanal kann hierbei also Werte von 0(Minimum) bis 255(Maximum) annehmen.



- 2. Testet eure Fähigkeit, Farben einzuschätzen. Vervollständigt die Tabelle auf der nächsten Seite, indem ihr...
 - a) ...erst alle Farbwerte schätzt.
 - b) ...danach mit Paint (siehe unten) die Farbwerte herausfindet.

In Microsoft Paint findet ihr oben rechts den Button "Palette bearbeiten", damit öffnet ihr ein Fenster, in dem ihr RGB-Werte zu Farben herausfinden könnt.







Farbe	RGB-Wert (geschätzt)	RGB-Wert (verglichen)







Ihr seid jetzt fast soweit, dass ihr Bilder bearbeiten könnt. Ein letzer Schritt fehlt allerdings noch: Zweidimensionale Arrays für das Abspeichern von Bildinformationen verwenden.

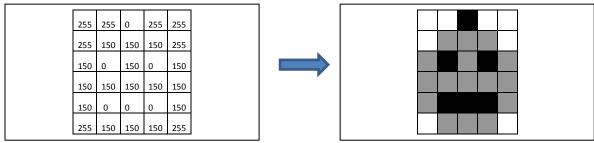
Schon mit einer einzigen Zahl pro Array-Eintrag lassen sich sehr leicht Bilder Darstellen. Der Einfachheit halber werdet ihr auf diesem Arbeitsblatt nur einen Farbkanal verwenden, also einen Integer-Wert pro Array-Eintrag, was etwa einem Schwarzweiss-Bild entsprechen würde.

Schwarzweiß-Arrays

Arrays, wie ihr sie bisher kennengelernt habt, könnte man also bereits zum Speichern von **Grauwerten** nutzen. Nehmen wir ein int-Array "zeile" mit der Länge 5.



Offensichtlich reicht ein einfaches Array nicht aus, um Bilder darin zu speichern. Benötigt wird ja eher so etwas:



Ein **zweidimensionales** Bild kann also nur in einem **zweidimensionalen Array** gespeichert werden. Um so ein Array der Größe 5*6 zu erzeugen, genügt die Anweisung:

Der erste Wert in der eckigen Klammer steht für die Breite ("x-Achse"), der zweite für die Höhe ("y-Achse"). Die Einträge (bild[i][j]) sind also wie folgt verteilt:

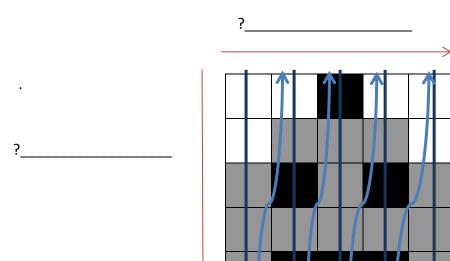


Medienmanipulation Bilder Bearbeiten



Um zweidimensionale Arrays zu durchlaufen, benötigt ihr zwei for-Schleifen. Eine for-Schleife mit **Zählvariable i** und <u>in dieser</u> dann die zweite mit **Zählvariable j**.

Diese durchlaufen dann, wenn in der Schleife jeweils der Eintrag mit den Koordinaten [i][j] angesteuert wird, das Array in folgender Reihenfolge:



Immer, wenn die Schleife für j vollständig durchgelaufen Ist, wird i um 1 erhöht.



- 1. Es handelt sich hierbei um eine for-Schleife, in der eine andere for-Schleife ausgeführt wird. Es gibt also eine innere und eine äußere for-Schleife.
 - a) Seht euch das obige Bild an und ordnet die Laufindizes i und j ihren Achsen zu.
 - b) Für das Durchlaufen welcher Richtung (rote Pfeile) ist die **innere** und für welche Richtung die **äußere** for-Schleife zuständig? Tragt es in die ?-Felder an dem Bild zusätzlich zu i und j noch "innen" und "aussen" ein.
 - c) Vervollständigt den folgenden Schleifenkopf, mit dem das Array "bild", wie es auf diesem Blatt vorkommt, durchlaufen wird.

```
for (int __ = 0; __ < ___; __++) {
  for (int __ = 0; __ < ___; __++) {
  }
}</pre>
```

Medienmanipulation Bilder Bearbeiten





- 2. Endlich ist es soweit, ihr könnt eure erste eigene Methode schreiben, mit der ihr ein Bild bearbeiten könnt.
 - a) Benutzt die Datei Graubild.java, die euch zur Verfügung gestellt wird.
 - b) Implementiert im Methodenrumpf "machedunkler" eine Methode, die mit zwei for-Schleifen alle Pixel (= int-Werte) des Bildes abdunkelt, indem der Zahlenwert halbiert wird. Der Methodenrumpf befindet sich ganz oben in der Datei und sieht so aus:

```
public static int[][] machedunkler(int[][] bild){
    //Hier eigenen Code
    return bild;
}
```

Die Variable "bild" wird **übergeben**, es ist also das Array, mit dem ihr in eurer Methode arbeiten müsst. Darin sind bereits die Bildinformationen gespeichert. Ihr müsst nur noch die Werte, an die ihr mit **bild[i][j]** kommt, verändern. Diese werden anschließend **zurückgegeben**.

- c) Testet, ob eure Methode funktioniert, indem ihr das Programm ausführt.
- d) Es können beliebig große Bilder übergeben werden. Passt euren Code so an, dass die Höhe und Breite des Bildes festgestellt wird und die Schleifen bis zu diesen Seitenlängen laufen.



Im Schülerlabor werdet ihr eigene Programme zur Bildbearbeitung schreiben. Diese könnt ihr auch später zuhause verwenden. Ihr müsst lediglich einen USB-Stick mitbringen um die Software mitzunehmen.

Listing B.1: Klasse Graubild.java

```
1 class Graubild{
    //Dies hier ist euer Methodenrumpf
   public static int[][] machedunkler(int[][] bild) {
      //Hier eigenen Code
      return bild;
10
     Den Code ab hier braucht ihr euch nicht mehr
         anzusehen.
20
    */
    public static void main(String args[]){
      //Array erzeugen und deklarieren.
      int[][] bild = new int[5][6];
      //Grauwerte eintragen.
      for (int i = 0; i < 5; i++) {</pre>
        for (int j = 0; j < 6; j++ ) {</pre>
          bild[i][j] = 150;
30
        }
      }
      //Pixel wei machen
      bild [0][0] = bild [1][0] = bild [3][0] = bild
35
          [4][0] = bild[0][1] = bild[0][5] = bild[4][1] =
         bild[4][5] = 255;
      //Pixel schwarz machen
      bild [2][0] = bild[1][2] = bild[3][2] = bild[1][4] =
          bild[2][4] = bild[3][4] = 0;
      //Array zum Kopieren
40
      int[][] vorherbild = new int[5][6];
      //Kopieren des Arrays
      for (int i = 0; i < 5; i++) {</pre>
        for (int j = 0; j < 6; j++) {
45
          vorherbild[i][j] = bild[i][j];
        }
```

```
}
      //Ausfhren der Methode
50
      bild = machedunkler(bild);
      System.out.println("
                                      Vorher
                                 Nachher");
      System.out.println();
      //Arrays ausgeben
      for (int j = 0; j < 6; j++) {
55
        for (int i = 0; i < 5; i++ ) {</pre>
          //Zeile des ersten Bildes
          if (vorherbild[i][j]<10) {</pre>
            System.out.print("
                                 " + vorherbild[i][j]);
          }else{
60
            if (vorherbild[i][j]<100) {</pre>
              System.out.print(" " + vorherbild[i][j]);
            }else{// end of if
              System.out.print(" " + vorherbild[i][j]);
65
          }
        }
        //Zeile des zweiten Bildes
        for (int i = 0; i<5; i++) {</pre>
          if (bild[i][j]<10) {</pre>
            System.out.print("
                                   " + bild[i][j]);
          }else{
            if (bild[i][j]<100) {</pre>
75
                                    " + bild[i][j]);
              System.out.print("
            }else{// end of if
              System.out.print( " " + bild[i][j]);
80
          }
        System.out.println();
        System.out.println();
85
      }
    }
90
95 }
```



InfoSphere Schülerlabor Informatik der RWTH Aachen

Modulablauf

Medienmanipulation



Zielsetzung



 Programmieren einer eigenen Bild- und Tonbearbeitungssoftware.

(Demonstration...)



Anwendung: Erstellen eures eigenen Videos.

(Hierzu ein Beispiel...)

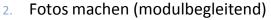


RWTHAACHEN

Ablauf



- ▶ Wie gehen wir dazu vor?
 - 1. Es handelt sich um ein Softwareprojekt
 - Möglichst 6er-Projektgruppen (evtl. kleiner)
 - ▶ Erst Projektplanung, dann loslegen





- 3. Fotos mit eigenen Methoden bearbeiten
- 4. Eine Tonspur mit eigenen Methoden erstellen









Medienmanipulation AB1:Projektplanung





Da ihr in diesem Modul als Gruppe an eurem Projekt arbeiten werdet, solltet ihr euch zunächst über einiges absprechen. Dieses Arbeitsblatt soll euch dabei helfen...

- ...die Aufgabenverteilung abzusprechen.
- ...euch bereits jetzt Gedanken zu machen, wie euer Endprodukt aussehen könnte.











Gruppe auch gemeinsam Bildbearbeitungssoftware programmieren und damit dann ein Video, das euch gefällt, erstellen könnt, muss jeder im Team wissen, was seine Aufgaben sind. Deswegen ist es sinnvoll zuerst die Aufgaben zu verteilen. Das solltet ihr jetzt hier tun.

Es gibt insgesamt 3 Aufgaben zu verteilen. Eine für jede 2er Gruppe eures Teams. Solltet ihr ein kleineres Team sein, müsst ihr die Aufgaben anders aufteilen. Die Aufgaben sind:

- a.) Eine Methode implementieren, die die Farben umkehrt. Schwierigkeitsstufe 1.
- b.) Farbfilter implementieren. Es sollen Farbfilter für die 3 Farbkanäle implementiert werden. Schwierigkeitsstufe 2.
- c.) Eine Methode soll ein Bild an der mittleren horizontalen Achse so **spiegeln**, dass ein symmetrisches Bild Schwierigkeitsstufe 3.

Sollte eine Gruppe schneller fertig werden als die anderen, ist noch eine Funktion zu implementieren, die ein Bild in ein Schwarzweiß-Bild umwandelt.

Sprecht euch jetzt ab, welche 2er Gruppe welche Funktion implementieren soll.

Tipp: Bewegungsabläufe:











Da wir mit Einzelbildern arbeiten, müssen Bewegungsabläufe aus Einzelbildern zusammengesetzt werden.

Medienmanipulation AB1:Projektplanung





- 2. Sprecht euch jetzt über folgende Punkte ab:
 - a) **Welche Methoden** könntet ihr noch zusätzlich zum Bearbeiten implementieren? Keine Sorge, euch fällt im Verlauf des Moduls bestimmt noch mehr ein.
 - b) In welche **Richtung** soll euer Video gehen, wie könnte es am Ende aussehen?
 - c) Was für **Fotos** solltet ihr dafür machen?

Hier könnt ihr Notizen über eure Planung und erste Ideen festhalten.	į
	! !
]
	1
	İ
	1
	į
	!
	7
	1/

Macht jetzt zunächst als Team ein paar (2-3) Fotos, um eure Methoden zu testen, danach geht es mit den Arbeitsblättern zu den Methoden weiter. Während des Moduls könnt ihr jederzeit weitere Fotos machen.

Medienmanipulation Eclipse Benutzen

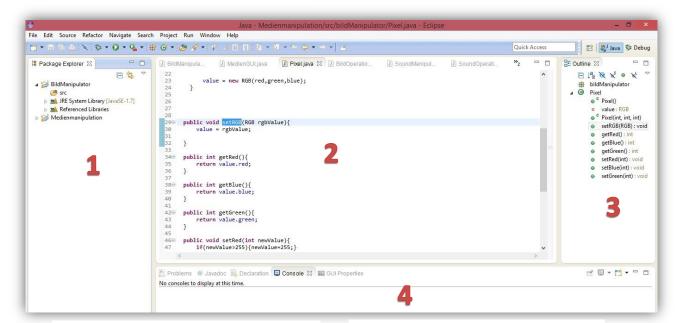




Dieses Infoblatt soll die grundlegende Benutzung von Eclipse erläutern. Unter anderem kann mit Eclipse Software in der Programmiersprache Java entwickelt werden.

Die Oberfläche

- Der "Package Explorer". Hier könnt ihr durch die Klassen eures Projektes navigieren.
- 2) Der **Editor**. Hier wird der Code bearbeitet



- Outline. Auflistung der Methoden und Variablen in der Klasse. Durch Klicken springt der Editor an die entsprechende Stelle.
- 4) **Konsole**: Hier werden Ausgaben und Fehler angezeigt, wenn euer Programm gestartet wird.

Programme testen

Um Programme zu testen, klickt auf den grünen "Play" Button

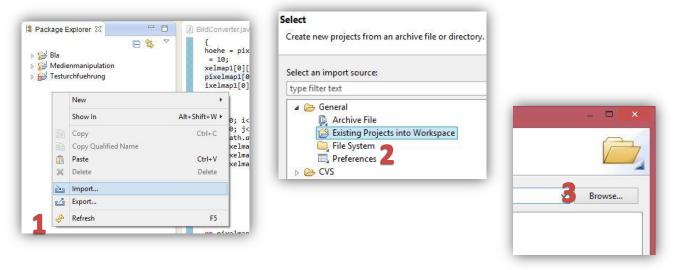


Medienmanipulation Eclipse Benutzen

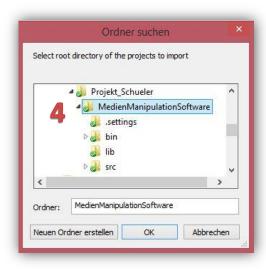


Projekte importieren

Sobald ihr mit der GUI arbeitet, müsst ihr den Quelltext der GUI importieren. Hierzu klickt ihr mit der rechten Maustaste auf den "Package Explorer" und dann auf "Import…".(1)



- 2. Anschließend wählt ihr im Dialogfenster, dass ein existierendes Projekt importiert werden soll.
- 3. Klickt auf "Next". Klickt im nächsten Fenster dann rechts Oben auf "Browse".
- Navigiert jetzt in den Ordner, in dem das zu importierende Projekt liegt und wählt dort den "MedienManipulationSoftware" Ordner. Klickt dann auf "OK".



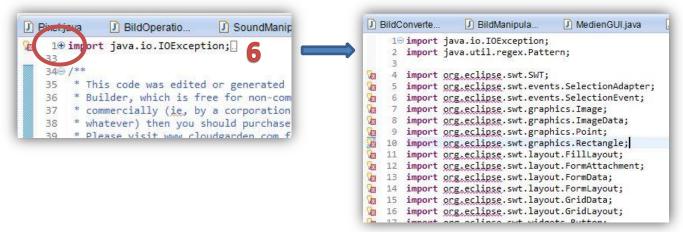
5. Klickt im nächsten Fenster auf "Finish".



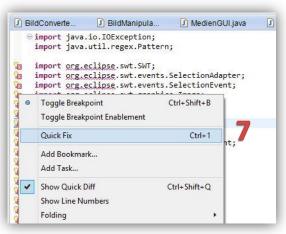


Im Editor werden jetzt Fehler angezeigt. Einige Pakete müssen noch geladen werden.

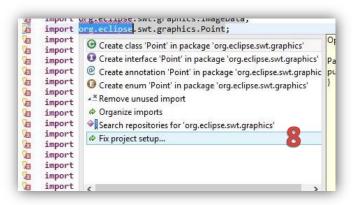
6. Wählt im "Package Explorer" die Klasse "MedienGUI"(in einem der Unterordner des Projekts), öffnet sie und lasst durch Klicken auf "+" alle importierten Pakete anzeigen.



7. Klickt dann mit der rechten Maustaste auf einen der Importe, die Probleme verursachen und wählt "Quick Fix".



8. Klickt dann auf "Fix Project Setup".



Medienmanipulation Anweisungen für das Modul





Dieses Merkblatt gibt euch einen Überblick über die Anweisungen, die ihr im Zusammenhang mit den im Modul verwendeten Datenstrukturen braucht.

Anweisungen

Anweisung	
pixel Pixel = new	Erzeugt eine Variable vom Typ Pixel mit den Farbwerten
Pixel(int red, int	red,green,blue.
green, int blue)	
<pre>pixel.setRed(int wert)</pre>	Setzt den Rot-Kanal auf "wert".
<pre>pixel.setGreen(int wert)</pre>	Setzt den Grün-Kanal auf "wert".
pixel.setBlue(int wert)	Setzt den Blau-Kanal auf "wert".
<pre>pixel.getRed()</pre>	Gibt den Rot-Wert eines Pixels zurück.
<pre>pixel.getGreen()</pre>	Gibt den Grün-Wert eines Pixels zurück.
pixel.getBlue()	Gibt den Blau-Wert eines Pixels zurück.



Variablenname



Die set-Methoden fangen unerlaubte Werte ab. pixel.setRed(1000) setzt also den Rotwert auf 255. Bei negativen Werten wird auf 0 gesetzt.

Um also den Rot-Wert der Pixel-Variable "pixel" in der Variable rotWert zu speichern, lautet die Anweisung:

```
int rotWert = pixel.getRed();
```

Da ihr nicht direkt mit den Bezeichnern der einzelnen Pixeln, sondern mit einem Array arbeitet, sehen diese Anweisungen für gewöhnlich wie folgt aus:

```
int rotWert = array1[i][j].getRed();
```

Medienmanipulation Anweisungen für das Modul





Wann immer ihr ein <u>neues</u> Pixel-Array erzeugt, müsst ihr beachten, dass die einzelnen Einträge des Arrays Pixel, also einzelne Objekte sind, die auch noch alle einzeln erzeugt werden müssen.

```
for(int i=0; i<array1.length; i++) {
  for(int j=0; j<array1[0].length; j++) {
    array1[i][j] = new Pixel(rot, gruen, blau);
}</pre>
```

Wobei rot, gruen und blau Integer-Werte sind, die der aktuelle Pixel dann erhält.

Erinnerung:

Der i-Wert entspricht der x-Koordinate und der j-Wert der y-Koordinate.

Medienmanipulation AB 2a: Farben Umkehren



Farben Umkehren



Abbildung 1: Beispiel für die Farbumkehr



1. Irgendwie gruselig, das Gesicht. Erst recht, wenn man die Farben umkehrt. Ihr wollt euren Fotos einen etwas surrealen Anstrich geben? Kein Problem. Implementiert einfach eine Methode, die die Farben umkehrt.

Hierzu müssen die Farbwerte jedes einzelnen Pixels umgekehrt werden. Der Methodenrumpf ist in der Klasse **BildOperationen.java** bereits vorhanden:

```
public static Pixel[][] farbenUmkehren(Pixel[][] pixelmap) {
     :
    return pixelmap;
}
```

Tipp: Durchlauft das "pixelarray" mit Schleifen (eine von links nach rechts und eine von oben nach unten) und setzt die Farbwerte auf den umgekehrten Wert.

Vergesst nicht das Testen. Wie bei der Softwareentwicklung üblich, ist das Testen ein wichtiger Bestandteil. Erst wenn die Methode bei geeigneten Testfällen brauchbare Ergebnisse liefert, könnt ihr sicher sein, dass sie funktioniert.

Medienmanipulation AB 2a: Farben Umkehren





Tipp: Das Umgekehrte eines Farbwertes ist: 255-Farbwert.



2. Wenn ihr eure Methode erfolgreich getestet habt, diese also Funktioniert, soll sie eurem gesamten Team zur Verfügung stehen. Kopiert die gesamte Methode in eine Textdatei (.txt), die ihr farbenUmkehren.txt nennt und legt diese in eurem Dropbox-Ordner ab.

Wenn alle Methoden implementiert sind, geht es mit dem Arbeitsblatt Hintergrundtausch weiter. Solltet ihr schon fertig sein und eure Teammitglieder brauchen noch etwas Zeit für ihre Aufgabe, dann bearbeitet vorher noch das Arbeitsblatt zum Schwarzweiß-Bild.

Medienmanipulation AB 2b: Farbfilter



Farbfilter



Abbildung 1: Beispiel für den Farbfilter



1. Was wären Instagram und co. nur ohne Farbfilter? Da Farbfilter Bilder interessanter aussehen lassen können, kommen sie reichlich zum Einsatz. Manchmal um eine gewisse Stimmung zu erzeugen oder einfach nur eine Portion Retro-Look. Damit eure Fotos auch in diesen Genuss kommen, programmiert doch Farbfilter-Methoden für die drei Grundfarben.

Die Methodenrümpfe sind in der Klasse BildOperationen bereits vorhanden:

Tipp: Durchlauft das "pixelarray" mit Schleifen (eine von links nach rechts und eine von oben nach unten) und setzt den Wert einer Farbe auf einen höheren Wert.

Vergesst nicht das Testen. Wie bei der Softwareentwicklung üblich, ist das Testen ein wichtiger Bestandteil. Erst wenn die Methode bei geeigneten Testfällen brauchbare Ergebnisse liefert, könnt ihr sicher sein, dass sie funktioniert.

Medienmanipulation AB 2b: Farbfilter





2. Wenn ihr eure Methode erfolgreich getestet habt, diese also Funktioniert, soll sie eurem gesamten Team zur Verfügung stehen. Kopiert also die gesamte Methode in eine Textdatei (.txt), die ihr **Farbfilter.txt** nennt. Legt diese in eurem Dropbox-Ordner ab.

Wenn alle Methoden implementiert sind, geht es mit dem Arbeitsblatt Hintergrundtausch weiter. Solltet ihr schon fertig sein und eure Teammitglieder brauchen noch etwas Zeit für ihre Aufgabe, dann bearbeitet vorher noch das Arbeitsblatt zum Schwarzweiß-Bild.

Medienmanipulation AB 2c: Spiegelkabinett



Spiegelkabinett



Abbildung 1: Beispiel für die Methode



 So einfach könnt ihr euch aussehen lassen, als ob ihr von einem anderen Planeten kommt. Abhängig von eurem ursprünglichen Foto könnt ihr mit diesem Effekt die unterschiedlichsten Bilder erzeugen. Schreibt einfach eine Methode "spiegelKabinettVertikal", die die rechte Bildhälfte verdoppelt. Verwendet dazu die get- und set-Methoden der Pixel.

Der Methodenrumpf hierzu ist in der Klasse BildOperationen bereits vorhanden:

Tipp: Durchlauft das "pixelarray" mit Schleifen(eine von links nach rechts und eine von oben nach unten) und setzt die Farbwerte jedes Pixels auf der linken Seite auf die Farbwerte des entsprechenden Pixels auf der rechten Seite. (Siehe Kasten auf der nächsten Seite.)

Vergesst nicht das Testen. Wie bei der Softwareentwicklung üblich, ist das Testen ein wichtiger Bestandteil. Erst wenn die Methode bei geeigneten Testfällen brauchbare Ergebnisse liefertkönnt ihr sicher sein, dass sie funktioniert.

Medienmanipulation AB 2c: Spiegelkabinett





Was ihr in jeder Zeile tun müsst, wird eher klar, wenn ihr über ein Beispiel nachdenkt. Dies hier soll ein Array "pixelarray" darstellen.

							<i>/ "</i>			
i	0	1	2	3	4	5	6	7	8	9

Ihr braucht folgende Werte: i (die Indexvariable zum Durchlaufen), pixelarray.length (ergibt hier 10). Jetzt braucht ihr einen Ausdruck, der zu jedem Wert die entsprechende Zahl berechnet. Zu 1 wäre das 8.

Hier könnt ihr Skizzen davon machen, wie die Pixel umsortiert werden sollen.



2. Wenn ihr eure Methode erfolgreich getestet habt, diese also Funktioniert, soll sie eurem gesamten Team zur Verfügung stehen. Kopiert also die Methode in eine Textdatei (.txt), die ihr **spiegelKabinettVertikal.txt** nennt. Legt diese in eurem Dropbox-Ordner ab.

Wenn alle Methoden implementiert sind, geht es mit dem Arbeitsblatt Hintergrundtausch weiter. Solltet ihr schon fertig sein und eure Teammitglieder brauchen noch etwas Zeit für ihre Aufgabe, dann bearbeitet vorher noch das Arbeitsblatt zum Schwarzweiß-Bild.

Medienmanipulation AB 2 Bonus: Schwarzweiß



Schwarzweiß-Bild



Abbildung 1: Beispiel für ein Schwarzweißbild



1. Schwarzweiß-Bilder haben einfach diesen ganz besonderen Charme. Zum Glück kann jedes Bild in ein Schwarzweiß-Bild umgewandelt werden. Probiert es aus.

Implementiert eine Methode, die ein Schwarzweiß-Bild erstellt.

Der Methodenrumpf hierzu ist in der Klasse BildOperationen bereits vorhanden:

```
public static Pixel[][] schwarzweiss(Pixel[][] pixelmap) {
    :
    return pixelmap;
}
```

Tipp: Durchlauft das "pixelarray" mit Schleifen(eine von links nach rechts und eine von oben nach unten) und setzt die Farbwerte jedes Pixels auf den entsprechenden Grauwert.

Medienmanipulation AB 2 Bonus: Schwarzweiß





Tipp: Graue Pixel unterscheiden sich von bunten Pixeln dadurch, dass alle 3 Farbkanäle des grauen Pixels denselben Wert haben. Je höher der Wert, desto heller das grau.

Ihr müsst euch also überlegen, wie ihr aus 3 unterschiedlichen Farbkanälen eines Pixels, die sozusagen jeweils eine unterschiedliche Helligkeit besitzen, die durchschnittliche Helligkeit der 3 Kanäle berechnet. Natürlich sind auch andere Lösungen möglich



2. Wenn ihr eure Methode erfolgreich getestet habt, diese also Funktioniert, soll sie eurem gesamten Team zur Verfügung stehen. Kopiert also die gesamte Methode in eine Textdatei (.txt), die ihr **schwarzweiss.txt** nennt. Legt diese in eurem Dropbox-Ordner ab.

Wenn alle Methoden implementiert sind, geht es mit dem Arbeitsblatt Hintergrundtausch weiter.

Medienmanipulation AB 3: Hintergrundtausch





Die Methode "hintergrundtausch" ist etwas anspruchsvoller als die Methoden, die ihr bisher implementiert habt. Das macht sie besonders geeignet dafür, etwas dabei zu lernen. (Arraygrenzen, Sinnvolles Testen, Toleranzen bei Farbwerten usw.) Damit alle im Team die Gelegenheit haben, ihre Fähigkeiten zu verbessern, ist vorgesehen, dass jede Zweiergruppe im Team diese Methode implementiert.

Hintergrundtausch





 Habt ihr euch schon einmal gefragt, wie in Filmen Schauspieler in alle möglichen, vielleicht sogar computergenerierte Welten kommen? Das Prinzip ist einfach, was irgendwie zum Nachmachen anregt. Fotos von euch in einer anderen Welt oder in einem anderen Jahrhundert? Kein Problem. Implementiert einfach eine Methode, die den Hintergrund eines Bildes austauscht.

Der Methodenrumpf ist auch hierfür bereits vorhanden:

Beachtet folgende Hinweise:

- a) Ihr braucht ein Foto von euch vor dem Bluescreen, um diese Methode zu testen.
- b) Das pixelarray1 ist das Bild, bei dem der Hintergrund ausgetauscht werden soll. Pixelarray2 ist der neue Hintergrund.
- c) Ihr braucht einen Farbwert, den ihr als Hintergrundfarbe definiert, nehmt einfach den Pixel in der (0,0) Ecke, also links oben. Speichert die Farbwerte als erstes in 3 int-Variablen ab.

Medienmanipulation AB 3: Hintergrundtausch



d) Bilder können unterschiedlich groß sein. Damit das Programm nicht abstürzt, sollten die Schleifen nur bis zum Ende des kleineren Arrays laufen. Das Minimum der beiden Längen von 2 Arrays erhaltet ihr auf diese Weise:

```
int minBreite = Math.min(pixelarray1.length, pixelarray2.length);
```

Dasselbe müsst ihr für die Höhe der Bilder durchführen.

Bevor ihr weitermacht: Stürzt euer Programm ab, wenn ihr die Schleifen ohne Inhalt laufen lasst? Testet das erst mal.

- e) Vergleicht jetzt in der Schleife jeweils den Farbwert des aktuellen Pixels mit dem Referenzwert, den ihr gespeichert habt. Stimmen die Werte überein, dann überschreibt die Farbwerte des aktuellen Pixels.
- f) Die Farbwerte der einzelnen Pixel des Bluescreens weichen voneinander ab. Erzeugt eine Integer Variable "toleranz". Farbwerte des pixelarray1 sollen ausgetauscht werden, wenn alle 3 Farbkanäle nicht mehr als "toleranz" von dem Referenzfarbwert abweichen.

<u>Tipp:</u> Den Abstand zwischen zwei Zahlen (a, b), erhält man mit Math.abs (a, b)

g) Auch diese Methode solltet ihr ausgiebig testen.

Hier könnt ihr euch etwas notieren, Skizzen davon machen, wie die Pixel umsortiert werden sollen oder Aufzeichnen, welche Probleme bei Bildern unterschiedlicher Größe auftreten können.

Ab jetzt werdet ihr erst einmal ohne Arbeitsblätter weiter machen. Ihr könnt selbst entscheiden, wie ihr weiter macht um Material für euer Video zu erstellen. Überlegt euch ab jetzt eigene Bildbearbeitungen, implementiert diese und erzeugt Bilder für euer Video. In etwa einer Stunde geht es dann weiter mit der Bearbeitung von Sounds.

Listing B.2: Klasse BildOperationen.java - Version für Aufgabenstellung

```
1 package bildManipulator;
 public class BildOperationen {
   public static Pixel[][] rotfilter(Pixel[][] pixelmap)
      //Hier eigene Methode schreiben.
     return pixelmap;
   public static Pixel[][] gruenfilter(Pixel[][] pixelmap
      //Hier eigene Methode schreiben.
     return pixelmap;
    }
   public static Pixel[][] blaufilter(Pixel[][] pixelmap)
      //Hier eigene Methode schreiben.
     return pixelmap;
    }
20
    public static Pixel[][] farbenUmkehren(Pixel[][]
       pixelmap) {
      //Hier eigene Methode schreiben.
     return pixelmap;
    }
25
    public static Pixel[][] spiegelKabinettVertikal(Pixel
       [][] pixelmap) {
      //Hier eigene Methode schreiben.
     return pixelmap;
   }
30
    public static Pixel[][] schwarzweiss(Pixel[][]
       pixelmap) {
      //Hier eigene Methode schreiben.
35
     return pixelmap;
    }
    public static Pixel[][] hintergrundtausch(Pixel[][]
       pixelmap1,
        Pixel[][] pixelmap2) {
40
      //Hier eigene Methode schreiben.
     return pixelmap1;
    }
```

```
45
    /*
     * Hier knnen die Namen der Zustzlichen Funktionen
        festgelegt werden. Unter
    * diesen erscheinen die Funktionen dann in der GUI.
50
    */
   private static String nameVonEigener1 = "NameEigenel";
   private static String nameVonEigener2 = "NameEigene2";
   private static String nameVonEigener3 = "NameEigene3";
   private static String nameVonEigener4 = "NameEigene4";
   private static String nameVonEigener5 = "NameEigene5";
   public static String getNameVonEigener1() {
      return nameVonEigener1;
60
   public static String getNameVonEigener2() {
      return nameVonEigener2;
65
   public static String getNameVonEigener3() {
     return nameVonEigener3;
   public static String getNameVonEigener4() {
     return nameVonEigener4;
   public static String getNameVonEigener5() {
     return nameVonEigener5;
   //Methodenrmpfe fr zustzliche Methoden.
   public static Pixel[][] eigenel(Pixel[][] pixelmap) {
     //Hier eigene Methode schreiben.
     return pixelmap;
   public static Pixel[][] eigene2(Pixel[][] pixelmap) {
      //Hier eigene Methode schreiben.
     return pixelmap;
    }
   public static Pixel[][] eigene3(Pixel[][] pixelmap) {
     //Hier eigene Methode schreiben.
     return pixelmap;
    }
   public static Pixel[][] eigene4(Pixel[][] pixelmap) {
```

Medienmanipulation Sounds bearbeiten



Sounds bearbeiten



Nicht nur, dass ihr jetzt mit eurem Wissen über Arrays Bilder bearbeiten könnt, Arrays sind in sehr vielen Kontexten nützlich. Zum Beispiel zum Bearbeiten von Sounds. Noch besser: diese könnt ihr dann für euer Video verwenden. Sounds können in Audacity aufgenommen oder z.B. auf www.freesound.org

heruntergeladen werden.

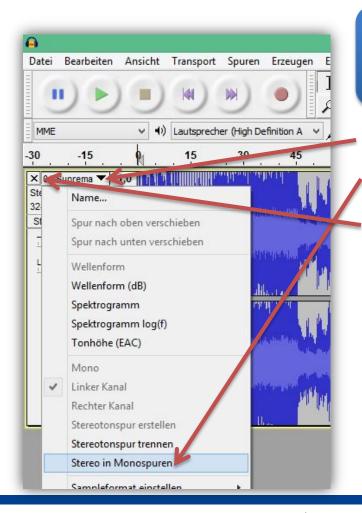


Vorsicht! Ihr könnt hier nur mit **Mono-Sounds**, also Sounds mit nur einem Kanal, und nur mit **.wav Dateien** arbeiten.

Mit Audacity lassen sich beliebige Sounds in dieses Format umwandeln.

- Mp3-Dateien: Mit Audacity laden und dann als .wav exportieren (Datei→Exportieren).
- Stereo Dateien: Siehe Bild unten

Umwandeln von Stereo in Mono-Dateien mit Audacity:



Sorgt am besten jetzt schon dafür, dass euch mindestens 2 Sounds zum Testen eurer Methoden zur Verfügung stehen.

- 1. Auf den Namen der Tonspur klicken.
- 2. "Stereo in Monospuren" auswählen.
- 3. Eine Tonspur schließen.
- 4. Datei →exportieren als .wav

Medienmanipulation Sounds bearbeiten



Rückgabe eurer Methoden:

Sounds können als double-Arrays werden. jeder gespeichert In der Methoden werden double-Arrays übergeben, deren Werte entsprechend verändert im jeweiligen double-Array "ausgabesound" zurückgegeben werden sollen. Das Array "ausgabesound" müsst ihr selbst erzeugen.

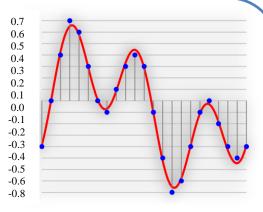
	doub	ole[]	eing	abes	ound	<u> </u>
Frame	0	1	2	3	4	5
Wert:	-0.4	0.0	0.4	0.7	0.6	0.3

frame 0 1 2

Wert: 0.4 0.7 0.6

Zusatzinfo: Bedeutung der Array-Einträge

Die Einträge der double-Arrays nennt man in diesem Kontext "Frames". Die Qualität einer Sounddatei wird durch ihre "Samplerate" begrenzt. Diese legt fest, wie viele Frames für eine Sekunde Sound verwendet werden (vgl. Auflösung). Sounds sind Digitalisierte Schallwellen. Auf dem Bild rechts seht ihr eine Schallwelle (rot) und die digitalen Integer-Zahlenwerte oder Frames, die diese repräsentieren (blau). In unserem Fall handelt es sich um Double-Frames, die im Bereich von -1.0 bis 1.0 liegen dürfen.



Quelle: Wikipedia

Dieses Arbeitsblatt enthält drei Aufgabenstellungen. Diese sollen ebenfalls im Team verteilt werden.



Um die Tonspur für euer Video zu erzeugen, braucht ihr ein paar grundlegende Methoden, mit denen ihr dann aus verschiedenen Sounds eure Tonspur zusammenbauen könnt. Diese grundlegenden Methoden sollen in dieser Aufgabe implementiert werden. Ladet diese, wenn sie funktionieren auch in den Dropbox-Ordner.

- 1. Implementiert die Methode "rueckwaerts" in der Klasse Soundoperationen. Diese soll einen Sound umdrehen. Zurückgeben müsst ihr ebenfalls ein double-Array.
- 2. Die Methode "aneinanderhaengen" soll in der Klasse Soundoperationen implementiert werden. Hier werden zwei Sounds, einer im "eingabesound1", der andere im "eingabesound2" übergeben. Der "ausgabesound" soll eine Aneinanderreihung der beiden sein. Nach dem Ausführen sollte sich die Anzahl der Frames, die in der GUI angezeigt wird, verändern.

Medienmanipulation Sounds bearbeiten



3. Implementiert die Methoden "abschneidenvorne" und "abschneidenhinten" in der Klasse Soundoperationen.java. Die Anzahl an Frames, die abgeschnitten werden soll, wird ebenfalls übergeben. Die Länge des "ausgabesound" muss entsprechend angepasst werden. Nach dem Ausführen sollte sich die Anzahl der Frames, die in der GUI angezeigt wird, verändern.

I I	
	,'

Ihr habt etwa 60 Minuten Zeit mit Sounds zu experimentieren und mit dem nächsten AB ein Video zu erstellen.

Listing B.3: Klasse SoundOperationen.java - Version für Aufgabenstellung

```
1 package soundManipulation;
 public class SoundOperationen {
    public static double[] rueckwaerts(double[]
       eingabesound) {
      //Hier eigene Methode schreiben.
     return ausgabesound;
   }
    public static double[] aneinanderhaengen(double[]
       eingabesound1, double[] eingabesound2) {
      //Hier eigene Methode schreiben.
     return ausgabesound;
15
    }
    public static double[] abschneidenVorne(double[]
       eingabesound, int framesZuSchneiden) {
      //Hier eigene Methode schreiben.
      return ausgabesound;
    }
25
    public static double[] abschneidenHinten(double[]
       eingabesound,
        int framesZuSchneiden) {
      //Hier eigene Methode schreiben.
     return ausgabesound;
    }
```

Medienmanipulation Erstellen der Videos





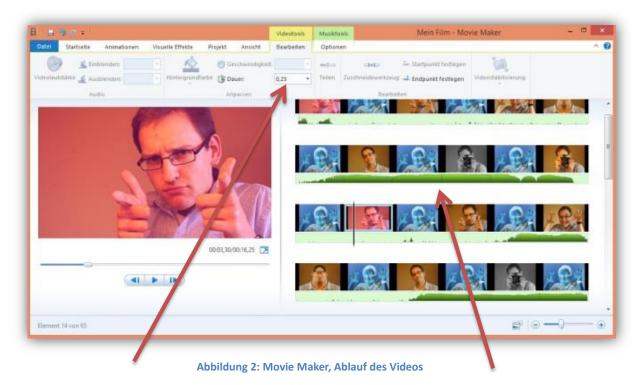
Ihr könnt ihr eure Videos mit einem einfachen Tool erstellen.

Auf diesem Arbeitsblatt werdet ihr sehen, wie ihr aus euren Bildern und eurer Sounddatei ein Video erstellen könnt. Verwendet wird der Movie Maker(Microsoft).



Gleich nach dem Start (Startseite) des Programmes "Movie Maker" können über diese Buttons Bilder (auch mehrere auf einmal) und der Sound eingefügt werden.

Abbildung 1: Movie Maker, Bilder und Sounds öffnen



"Dauer" legt fest, wie lange jedes Bild angezeigt wird. Hier: 0,25 (Sekunden). (Registrierkarte "Bearbeiten") Einmal eingefügt, können Bilder beliebig oft kopiert und wieder eingefügt werden.



InfoSphere Schülerlabor Informatik der RWTH Aachen

Glaubwürdigkeit von Medien

Medienmanipulation



Wie glaubwürdig sind Medien?





Welchen Eindruck vermittelt dieses Bild?

(Links: Soldat des Landes "A", rechts: Soldat des Landes "B")



Wie glaubwürdig sind Medien?





Welchen Eindruck vermittelt dieses Bild?

(Links: Soldat des Landes "B", rechts: Soldat des Landes "A")

InfoSphere – Schülerlabor Informatik



Wie glaubwürdig sind Medien?





So leicht lässt sich die Aussage eines Bildes verändern.

RWTHAACHEN

Wie glaubwürdig sind Medien?



▶ Noch weitere Beispiele:





5 InfoSphere – Schülerlabor Informatik



Wie glaubwürdig sind Medien?



▶ Noch weitere Beispiele:



Quelle: Bildblog.de



Quelle: Bildblog.de



Was ist alles Möglich?



► Ein kleines Beispiel soll zeigen, wie auch Videos verändert werden können

(Harmloses Fanvideo, von Einzelperson erstellt, Charaktere aus zwei Fernsehserien treffen aufeinander)

Das Video:

https://www.youtube.com/watch?v=q3bGYIjQ5Uw

Wie stark es bearbeitet wurde:

https://www.youtube.com/watch?v=1Zrn37KzxuE

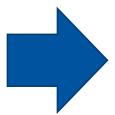
7 InfoSphere – Schülerlabor Informatik



Was ist alles Möglich?



▶ Deswegen die Frage: Computergestützte Medienmanipulation, Bereicherung oder Gefahr?



- 1. Gruppendiskussion. Teilt euch in *4er-Gruppen* auf, es sollt genauso viele "*Gefahr*"- wie "*Bereicherung*"-Gruppen geben.
- Diskutiert <u>5 Minuten</u> in der Gruppe und findet Argumente für euren Standpunkt. Haltet diese fest.
- Anschließend wird <u>als Kurs</u> diskutiert, wobei ihr eure Argumente einbringt, diese werden an der <u>Tafel</u> gesammelt.



Anhang C Musterlösungen der Programmieraufgaben

Listing C.1: Klasse MedienGUI.java

```
1 class Graubild Loesung {
    //Array 5*6 anpassen vorher, nachher ausgabe
   public static int[][] machedunkler(int[][] bild) {
      for (int i = 0; i < bild.length; i++) {</pre>
        for (int j = 0; j < bild[0].length; j++ ) {</pre>
          bild[i][j] = bild[i][j]/2;
      }
      return bild;
15
    public static void main(String args[]){
      //Array erzeugen und deklarieren.
      int[][] bild = new int[5][6];
      //Grauwerte eintragen.
      for (int i = 0; i < 5; i++) {</pre>
        for (int j = 0; j < 6; j++) {
          bild[i][j] = 150;
        }
25
      }
      //Pixel wei machen
      bild [0][0] = bild [1][0] = bild [3][0] = bild
          [4][0] = bild[0][1] = bild[0][5] = bild[4][1] =
         bild[4][5] = 255;
30
      //Pixel schwarz machen
      bild [2][0] = bild[1][2] = bild[3][2] = bild[1][4] =
          bild[2][4] = bild[3][4] = 0;
```

```
//Array zum Kopieren
      int[][] vorherbild = new int[5][6];
35
      //Kopieren des Arrays
      for (int i = 0; i < 5; i++) {</pre>
        for (int j = 0; j < 6; j++ ) {</pre>
          vorherbild[i][j] = bild[i][j];
40
        }
      }
      //Ausfhren der Methode
      bild = machedunkler(bild);
45
      System.out.println("
                                      Vorher
                                 Nachher");
      System.out.println();
      //Arrays ausgeben
      for (int j = 0; j < 6; j++) {
        for (int i = 0; i < 5; i++ ) {</pre>
50
          //Zeile des ersten Bildes
          if (vorherbild[i][j]<10) {</pre>
            System.out.print(" " + vorherbild[i][j]);
          }else{
            if (vorherbild[i][j]<100) {
              System.out.print("
                                   " + vorherbild[i][j]);
            }else{// end of if
              System.out.print(" " + vorherbild[i][j]);
          }
60
        }
        //Zeile des zweiten Bildes
        for (int i = 0; i<5; i++) {</pre>
65
          if (bild[i][j]<10) {</pre>
            System.out.print("
                                   " + bild[i][j]);
          }else{
            if (bild[i][j]<100) {</pre>
                                    " + bild[i][j]);
              System.out.print("
70
            }else{// end of if
              System.out.print( " " + bild[i][j]);
75
          }
        System.out.println();
        System.out.println();
      }
80
    }
```

```
85
  }
           Listing C.2: Klasse BildOperationen.java - Musterlösung
1 package bildManipulator;
  public class BildOperationen {
    public static Pixel[][] rotfilter(Pixel[][] pixelmap)
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
         Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
            von Oben nach
                                  // Unten
          pixelmap[i][j].setRed(pixelmap[i][j].getRed() +
              20);
      }
      return pixelmap;
    public static Pixel[][] gruenfilter(Pixel[][] pixelmap
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
         Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
            von Oben nach
                                  // Unten
          pixelmap[i][j].setGreen(pixelmap[i][j].getGreen
              () + 20);
        }
20
      }
      return pixelmap;
    public static Pixel[][] blaufilter(Pixel[][] pixelmap)
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
         Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
            von Oben nach
                                  // Unten
          pixelmap[i][j].setBlue(pixelmap[i][j].getBlue()
              + 20);
        }
30
      }
```

return pixelmap;

```
}
    /*
35
     * Diese Funktion kehrt die Farben. Das Inverse eines
        Farbwertes ist
     * 255-Farbwert.
    public static Pixel[][] farbenUmkehren(Pixel[][]
       pixelmap) {
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
40
         Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
           von Oben nach
                                 // Unten
          // Invertiere jeden der Farbkanle.
          pixelmap[i][j].setRed(255 - pixelmap[i][j].
45
             getRed());
          pixelmap[i][j].setGreen(255 - pixelmap[i][j].
             getGreen());
          pixelmap[i][j].setBlue(255 - pixelmap[i][j].
             getBlue());
     }
50
     return pixelmap;
    }
    * Diese Funktion erstellt ein Symmetrisches Bild
        durch verdoppeln der
     * rechten Bildhlfte.
     */
    public static Pixel[][] spiegelKabinettVertikal(Pixel
       [][] pixelmap) {
      for (int i = 0; i < pixelmap.length / 2; i++) { //</pre>
         von Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
60
           von Oben nach
                                 // Unten
          pixelmap[i][j].setRed(pixelmap[pixelmap.length -
               i - 1][j]
              .getRed());
          pixelmap[i][j].setGreen(pixelmap[pixelmap.length
65
              -i-1][j]
              .getGreen());
          pixelmap[i][j].setBlue(pixelmap[pixelmap.length
              - i - 1][j]
              .getBlue());
70
        }
```

```
}
      return pixelmap;
    }
     * Diese Funktion erstellt ein Schwarzweissbild.
        Hierzu wird jeder Farbkanal
     * eines Pixels auf den Durchschnitt der drei
        Farbkanle des Pixels gesetzt.
    public static Pixel[][] schwarzweiss(Pixel[][]
       pixelmap) {
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
80
          Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
           von Oben nach
                                 // Unten
          int helligkeit = (pixelmap[i][j].getRed()
              + pixelmap[i][j].getGreen() + pixelmap[i][j
                  ].getBlue()) / 3; // Errechne
85
                                                   Durchschnitt
          pixelmap[i][j].setRed(helligkeit);
          pixelmap[i][j].setGreen(helligkeit);
          pixelmap[i][j].setBlue(helligkeit);
90
      }
      return pixelmap;
    }
     * Diese Funktion tauscht den Hintergrund aus. Als
        Referenzfarbwert des
     * Hintergrundes wird pixelmap[0][0] genommen, also
        der Pixel in der linken
     * oberen Ecke. Da der Bluescreen Schatten und Falten
        aufweist, wird ein
     * Toleranzbereich festgelegt. Verndert wird
        pixelmap1. Pixel knnen nur
     * ausgetauscht werden, solange Pixel[i][j] in beiden
100
        Bildern existiert. Der
     * Algoritmus muss also so begrenzt sein, dass nicht
        versucht wird Pixel
     * auzutauschen, die ausserhalb der Minima der
        Dimensionen beider Bilder
     * liegen.
     */
    public static Pixel[][] hintergrundtausch(Pixel[][]
       pixelmap1,
        Pixel[][] pixelmap2) {
```

```
// Hier Hintergrundfarbe von Pixel links oben nehmen
      Pixel farbwertHintergrund = pixelmap1[0][0];
110
      // Toleranzbereiche berechnen
      int toleranz = 30;
      int minRed = farbwertHintergrund.getRed() - toleranz
      int maxRed = farbwertHintergrund.getRed() + toleranz
115
      int minGreen = farbwertHintergrund.getGreen() -
      int maxGreen = farbwertHintergrund.getGreen() +
          toleranz;
      int minBlue = farbwertHintergrund.getBlue() -
          toleranz;
      int maxBlue = farbwertHintergrund.getBlue() +
120
          toleranz;
      // Minimum der Bilddimensionen berechnen
      int breite = Math.min(pixelmap1.length, pixelmap2.
          length);
      int hoehe = Math.min(pixelmap1[0].length, pixelmap2
          [0].length);
125
      for (int i = 0; i < breite; i++) { // von Links nach</pre>
           Rechts
        for (int j = 0; j < hoehe; j++) { // von Oben nach
            Unten
             berprfen ob die Farbwerte von pixelmap2 in
              der Nhe der
          // Hintergrundfarben liegen
130
          if (pixelmap1[i][j].getRed() > minRed
              && pixelmap1[i][j].getRed() < maxRed) {
            if (pixelmap1[i][j].getGreen() > minGreen
                 && pixelmap1[i][j].getGreen() < maxGreen)
135
                    {
              if (pixelmap1[i][j].getBlue() > minBlue
                   && pixelmap1[i][j].getBlue() < maxBlue)
                      {
                 // Es handelt sich um den Hintergrund. Die
                     Pixel
                 // werden ausgetauscht.
                pixelmap1[i][j].setRed(pixelmap2[i][j].
                    getRed());
                pixelmap1[i][j]
```

```
.setGreen(pixelmap2[i][j].getGreen());
                pixelmap1[i][j].setBlue(pixelmap2[i][j].
145
                    getBlue());
             }
150
          }
        }
      }
      return pixelmap1;
155
160
     * Hier knnen die Namen der eigenen Funktionen
         festgelegt werden. Unter
     * diesen erscheinen die Funktionen dann in der GUI.
    private static String nameVonEigener1 = "Welle";
    private static String nameVonEigener2 = "
       SpiegelkabinettLinkeSeite";
    private static String nameVonEigener3 = "NameEigene3";
    private static String nameVonEigener4 = "NameEigene4";
    private static String nameVonEigener5 = "
       FadingHintergrundtausch";
170
    public static String getNameVonEigener1() {
      return nameVonEigener1;
    public static String getNameVonEigener2() {
175
      return nameVonEigener2;
    }
    public static String getNameVonEigener3() {
      return nameVonEigener3;
    }
180
    public static String getNameVonEigener4() {
      return nameVonEigener4;
    }
185
    public static String getNameVonEigener5() {
      return nameVonEigener5;
    /*
```

```
* Rotfilter analog zu einem Rotfilter in der
         Fotografie. Soll den Rotwert
     * jedes Pixels erhhen. Die anderen Filter
         funktionieren analog.
    public static Pixel[][] eigenel(Pixel[][] pixelmap) {
195
      // hier eigene Funktion schreiben
      Pixel[][] neues = new Pixel[pixelmap.length][
          pixelmap[0].length];
      for (int i = 0; i < pixelmap.length; i++) { // von</pre>
          Links nach Rechts
         for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
            von Oben nach
200
                                  // Unten
          neues[i][j] = new Pixel(0, 0, 0);
           int verschobenesi = i
               + (int) (Math.sin(10.0 * (double) j
                   / (double) pixelmap.length)
205
                   * pixelmap.length / 10);
           if (verschobenesi < 0)</pre>
             verschobenesi = 0;
           if (verschobenesi > pixelmap.length - 1)
            verschobenesi = pixelmap.length - 1;
210
          neues[i][j].setRed(pixelmap[verschobenesi][j].
              getRed());
          neues[i][j].setGreen(pixelmap[verschobenesi][j].
              getGreen());
          neues[i][j].setBlue(pixelmap[verschobenesi][j].
              getBlue());
         }
215
      }
      return neues;
    public static Pixel[][] eigene2(Pixel[][] pixelmap) {
220
      // hier eigene Funktion schreiben
      for (int i = 0; i < pixelmap.length / 2; i++) { //</pre>
          von Links nach Rechts
        for (int j = 0; j < pixelmap[0].length; j++) { //</pre>
            von Oben nach
                                  // Unten
225
           pixelmap[pixelmap.length - i - 1][j].setRed(
              pixelmap[i][j]
               .getRed());
          pixelmap[pixelmap.length - i - 1][j].setGreen(
              pixelmap[i][j]
               .getGreen());
```

```
pixelmap[pixelmap.length - i - 1][j].setBlue(
230
              pixelmap[i][j]
               .getBlue());
      }
      return pixelmap;
235
    }
    public static Pixel[][] eigene3(Pixel[][] pixelmap) {
      // hier eigene Funktion schreiben
      return pixelmap;
240
    public static Pixel[][] eigene4(Pixel[][] pixelmap) {
      // hier eigene Funktion schreiben
      return pixelmap;
245
    }
    public static Pixel[][] eigene5(Pixel[][] pixelmap1,
       Pixel[][] pixelmap2) {
      // hier eigene Funktion schreiben
250
      // Hier Hintergrundfarbe eintragen
      Pixel farbwertHintergrund = pixelmap1[0][0];
      // Toleranzbereiche berechnen
      int toleranz = 30;
255
      int minRed = farbwertHintergrund.getRed() - toleranz
      int maxRed = farbwertHintergrund.getRed() + toleranz
         ;
      int minGreen = farbwertHintergrund.getGreen() -
          toleranz;
      int maxGreen = farbwertHintergrund.getGreen() +
260
          toleranz;
      int minBlue = farbwertHintergrund.getBlue() -
          toleranz;
      int maxBlue = farbwertHintergrund.getBlue() +
          toleranz;
      // Minimum der Bilddimensionen berechnen
265
      int breite = Math.min(pixelmap1.length, pixelmap2.
          length);
      int hoehe = Math.min(pixelmap1[0].length, pixelmap2
          [0].length);
      for (int i = 0; i < breite; i++) { // von Links nach</pre>
           Rechts
```

```
for (int j = 0; j < hoehe; j++) { // von Oben nach
270
             Unten
              berprfen ob die Farbwerte von pixelmap2 in
              der Nhe der
           // Hintergrundfarben liegen
          if (pixelmap1[i][j].getRed() > minRed
               && pixelmap1[i][j].getRed() < maxRed
275
               && pixelmap1[i][j].getGreen() > minGreen
               && pixelmap1[i][j].getGreen() < maxGreen
               && pixelmap1[i][j].getBlue() > minBlue
               && pixelmap1[i][j].getBlue() < maxBlue) {
280
             // Es handelt sich um den Hintergrund. Die
                Pixel
             // werden ausgetauscht.
            pixelmap1[i][j].setRed(pixelmap2[i][j].getRed
                ());
            pixelmap1[i][j].setGreen(pixelmap2[i][j].
                getGreen());
            pixelmap1[i][j].setBlue(pixelmap2[i][j].
285
                getBlue());
           } else {
             // Sonst nur Vordergrund mit Hintergrund
                mischen.
            pixelmap1[i][j]
                 .setRed((pixelmap2[i][j].getRed() +
                    pixelmap1[i][j]
                     .getRed()) / 2);
290
            pixelmap1[i][j]
                 .setGreen((pixelmap2[i][j].getGreen() +
                    pixelmap1[i][j]
                     .getGreen()) / 2);
            pixelmap1[i][j]
                 .setBlue((pixelmap2[i][j].getBlue() +
                    pixelmap1[i][j]
                     .getBlue()) / 2);
           }
         }
300
      }
      return pixelmap1;
    }
305
          Listing C.3: Klasse SoundOperationen.java - Musterlösung
 package soundManipulation;
  public class SoundOperationen {
```

```
/*
     * Die Methode 1sst einen Sound rckwrts laufen,
        hierzu werden die
     * Eintrge des Buffers in verkehrter Reihenfolge in
        den Ausgabebuffer
     * geschrieben.
     */
    public static double[] rueckwaerts(double[]
       eingabesound) {
      int numFrames = eingabesound.length;
      double[] ausgabesound = new double[numFrames];
15
      for (int i = 0; i < numFrames; i++) {</pre>
        ausgabesound[i] = eingabesound[numFrames - (i + 1)
           ];
20
      return ausgabesound;
    }
     * Hier werden nacheinander die Inhalte von Bufferl
        und Buffer2 in den
     * Ausgabebuffer kopiert. Der Ausgabebuffer besitzt
        bereits exakt die Lnge
     * der beiden Buffer zusammengenommen.
     */
   public static double[] aneinanderhaengen(double[]
       eingabesound1, double[] eingabesound2) {
      int numFrames1 = eingabesound1.length;
      int numFrames2 = eingabesound2.length;
      double[] ausgabesound = new double[numFrames1 +
         numFrames2];
35
      for (int i = 0; i < numFrames1; i++) {</pre>
        ausgabesound[i] = eingabesound1[i]; // bertrage
           die Eintrge des ersten
                        // Arrays
      for (int i = 0; i < numFrames2; i++) {</pre>
40
        ausgabesound[i + numFrames1] = eingabesound2[i];
           // bertrage der
                               // Reihenfolge nach ab
                               // der Position, die
                               // zuetzt in der vorigen
                               // Schleife beschrieben
45
```

```
// wurde.
      }
     return ausgabesound;
    }
     * Hier soll der Inhalt des Buffers, ausgenommen der
     * "framesZuSchneiden" Eintrge, in den Ausgabebuffer
        geschrieben werden.
     * Der Ausgabebuffer hat hierfr exakt die richtige
        Lnge.
    public static double[] abschneidenVorne(double[]
       eingabesound, int framesZuSchneiden) {
      int framesZuSpeichern = eingabesound.length -
         framesZuSchneiden;
      double[] ausgabesound = new double[framesZuSpeichern
      for (int i = 0; i < framesZuSpeichern; i++) {</pre>
        ausgabesound[i] = eingabesound[i +
           framesZuSchneiden];
      }
     return ausgabesound;
65
    }
     * Hier soll der Inhalt des Buffers, ausgenommen der
        letzten
     * "framesZuSchneiden" Eintrge, in den Ausgabebuffer
        geschrieben werden.
     * Der Ausgabebuffer hat hierfr exakt die richtige
        Lnge.
     */
    public static double[] abschneidenHinten(double[]
       eingabesound,
        int framesZuSchneiden) {
75
      int framesZuSpeichern = eingabesound.length -
         framesZuSchneiden;
      double[] ausgabesound = new double[framesZuSpeichern
         ];
      for (int i = 0; i < framesZuSpeichern; i++) {</pre>
        ausgabesound[i] = eingabesound[i];
      return ausgabesound;
```

```
85 }
```

Anhang D Quelltext der Medienmanipulationssoftware (ausgenommen Musterlösungen)

Listing D.1: Klasse Bildmanipulator.java

```
1 package bildManipulator;
 import org.eclipse.swt.SWT;
 import org.eclipse.swt.graphics.ImageData;
5 import org.eclipse.swt.graphics.ImageLoader;
  * Write a description of class BildManipulator here.
10 * @author (your name)
  * @version (a version number or a date)
 public class BildManipulator {
   private String file1;
  private boolean bild1Geladen = false;
   private boolean bild2Geladen = false;
   private Pixel[][] pixelmap1;
   private Pixel[][] pixelmap2;
  // private Pixel[][] undoPixelMap1;
   // private Pixel[][] undoPixelMap2;
   private boolean linkesFenster = true;
   public String getNameVonEigener1() {
     return BildOperationen.getNameVonEigener1();
   public String getNameVonEigener2() {
     return BildOperationen.getNameVonEigener2();
```

```
public String getNameVonEigener3() {
      return BildOperationen.getNameVonEigener3();
35
    public String getNameVonEigener4() {
     return BildOperationen.getNameVonEigener4();
    public String getNameVonEigener5() {
40
      return BildOperationen.getNameVonEigener5();
    public void setBild1geladen(boolean setTo) {
     bild1Geladen = setTo;
    public boolean getBild1geladen() {
     return bild1Geladen;
    public void setBild2geladen(boolean setTo) {
     bild2Geladen = setTo;
55
    public boolean getBild2geladen() {
      return bild2Geladen;
    public void setLinkesFenster(boolean setTo) {
      linkesFenster = setTo;
    public boolean getLinkesFenster() {
     return linkesFenster;
    public String getFile() {
     return file1;
70
    }
    public void setFileNull() {
      file1 = null;
75
    public void oeffne(String filename) {
      if (filename != null) {
       file1 = filename;
        int maxPixels = 1500000;
```

```
ImageData image = new ImageData(filename);
        if (image.width * image.height > maxPixels) {
          int maxheight = (int) Math.sqrt(maxPixels);
85
          int newheight = maxheight;
          int newwidth = (int) (((double) newheight / (
              double) image.height) * image.width);
          image = image.scaledTo(newwidth, newheight);
        }
90
        if (linkesFenster) {
          pixelmap1 = BildConverter.getPixelmap(image);
          bild1Geladen = true;
95
        } else {
          pixelmap2 = BildConverter.getPixelmap(image);
          bild2Geladen = true;
        }
      }
100
    }
    public void speichere(String filename) {
      ImageData[] Image = new ImageData[1];
      if (linkesFenster) {
105
        if (bild1Geladen)
           Image[0] = BildConverter.getImageData(pixelmap1)
              ;
      } else {
110
        if (bild2Geladen)
          Image[0] = BildConverter.getImageData(pixelmap2)
115
      ImageLoader imageLoader = new ImageLoader();
      imageLoader.data = Image;
      try{
      imageLoader.save(filename, SWT.IMAGE_JPEG);
      }catch(Exception e) {
          System.err.println("Caught Exception: " + e.
120
              getMessage());
      }
    }
    public ImageData getImageData() {
      if (linkesFenster) {
        return BildConverter.getImageData(pixelmap1);
      } else {
        return BildConverter.getImageData(pixelmap2);
130
      }
```

```
}
    public void SpiegelKabinettVertikal() {
      if (linkesFenster) {
135
        if (bild1Geladen)
          pixelmap1 = BildOperationen.
              spiegelKabinettVertikal(pixelmap1);
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.
140
              spiegelKabinettVertikal(pixelmap2);
      }
    }
    public void rotFilter() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.rotfilter(pixelmap1)
      } else {
150
        if (bild2Geladen)
          pixelmap2 = BildOperationen.rotfilter(pixelmap2)
      }
155
    }
    public void gruenFilter() {
      if (linkesFenster) {
        if (bild1Geladen)
160
          pixelmap1 = BildOperationen.gruenfilter(
              pixelmap1);
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.gruenfilter(
              pixelmap2);
      }
165
    }
    public void blauFilter() {
      if (linkesFenster) {
170
        if (bild1Geladen)
          pixelmap1 = BildOperationen.blaufilter(pixelmap1
              );
      } else {
        if (bild2Geladen)
175
```

```
pixelmap2 = BildOperationen.blaufilter(pixelmap2
              );
      }
180
    public void farbenUmkehren() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.farbenUmkehren(
              pixelmap1);
185
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.farbenUmkehren(
              pixelmap2);
      }
    }
190
    public void schwarzweiss() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.schwarzweiss(
195
              pixelmap1);
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.schwarzweiss(
              pixelmap2);
      }
200
    }
    public void hintergrundtausch() {
      if (linkesFenster) {
        if (bild1Geladen && bild2Geladen)
205
          pixelmap1 = BildOperationen.hintergrundtausch(
              pixelmap1, pixelmap2);
      } else {
        if (bild1Geladen && bild2Geladen)
          pixelmap2 = BildOperationen.hintergrundtausch(
              pixelmap2, pixelmap1);
      }
210
    }
    public void eigene1() {
      if (linkesFenster) {
215
        if (bild1Geladen)
          pixelmap1 = BildOperationen.eigene1(pixelmap1);
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.eigene1(pixelmap2);
220
```

```
}
    }
    public void eigene2() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.eigene2(pixelmap1);
      } else {
        if (bild2Geladen)
230
          pixelmap2 = BildOperationen.eigene2(pixelmap2);
      }
    }
235
    public void eigene3() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.eigene3(pixelmap1);
240
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.eigene3(pixelmap2);
      }
245
    }
    public void eigene4() {
      if (linkesFenster) {
        if (bild1Geladen)
          pixelmap1 = BildOperationen.eigene4(pixelmap1);
250
      } else {
        if (bild2Geladen)
          pixelmap2 = BildOperationen.eigene4(pixelmap2);
      }
255
    }
    public void eigene5() {
      if (linkesFenster) {
        if (bild1Geladen && bild2Geladen)
260
          pixelmap1 = BildOperationen.eigene5(pixelmap1,
              pixelmap2);
      } else {
        if (bild1Geladen && bild2Geladen)
          pixelmap2 = BildOperationen.eigene5(pixelmap2,
              pixelmap1);
      }
265
    }
```

Listing D.2: Klasse Pixel.java

```
1 package bildManipulator;
  import org.eclipse.swt.graphics.RGB;
  public class Pixel {
    public Pixel() {
10
   }
     private RGB value;
    public Pixel(int red, int green, int blue) {
      if (red>255) red=255;
      if (green>255) green=255;
      if (blue>255) blue=255;
      if (red<0) red=0;
20
      if (green<0) green=0;</pre>
      if (blue<0) blue=0;</pre>
      value = new RGB(red, green, blue);
    }
25
    public void setRGB(RGB rgbValue) {
      value = rgbValue;
30
    }
    public int getRed() {
      return value.red;
    public int getBlue(){
      return value.blue;
40
    public int getGreen(){
      return value.green;
45
    public void setRed(int newValue) {
      if (newValue>255) {newValue=255;}
      if (newValue<0) {newValue=0;}</pre>
      value.red = newValue;
50
    public void setBlue(int newValue) {
```

```
if (newValue>255) {newValue=255;}
      if (newValue<0) {newValue=0;}</pre>
      value.blue = newValue;
55
    }
    public void setGreen(int newValue) {
      if (newValue>255) {newValue=255;}
      if (newValue<0) {newValue=0;}</pre>
      value.green = newValue;
65
  }
                  Listing D.3: Klasse BildConverter.java
1 package bildManipulator;
  import org.eclipse.swt.graphics.ImageData;
  import org.eclipse.swt.graphics.PaletteData;
5 import org.eclipse.swt.graphics.RGB;
  public class BildConverter {
     public static Pixel[][] getPixelmap(ImageData image) {
       Pixel[][] pixelmap = new Pixel[image.width][image.
           height];
      for (int i = 0; i < image.width; i++) {</pre>
        for (int j = 0; j < image.height; j++) {</pre>
          pixelmap[i][j]=new Pixel();
          int pixelwert=image.getPixel(i,j);
20
  //
              int red = (pixelwert) & 0xFF;
              int green = (pixelwert>>8) &0xFF;
              int blue = (pixelwert>>16)&0xFF;
          RGB rgb=new RGB (red, green, blue);
25
          Pixel pixel=pixelmap[i][j];
          pixel.setRGB(rgb);
        }
```

```
return pixelmap;
     }
35
     public static ImageData getImageData(Pixel[][]
        pixelmap) {
40
       PaletteData palette = new PaletteData(255, 65280,
          16711680);
       ImageData image = new ImageData(pixelmap.length,
          pixelmap[0].length , 24, palette);
      for (int i = 0; i < pixelmap.length; i++) {</pre>
        for (int j = 0; j < pixelmap[0].length; j++) {</pre>
45
          image.setPixel(i, j, palette.getPixel(new RGB(
             pixelmap[i][j].getRed(), pixelmap[i][j].
             getGreen(), pixelmap[i][j].getBlue())));
      }
50
       return image;
     }
               Listing D.4: Klasse SoundManipulator.java
package soundManipulation;
 import javax.sound.sampled.AudioInputStream;
 import javax.sound.sampled.AudioSystem;
5 import javax.sound.sampled.Clip;
 import java.io.*;
   * Write a description of class SoundManipulator here.
   * @author (your name)
   * @version (a version number or a date)
 public class SoundManipulator {
15
   private WavFile soundDateil;
   private WavFile soundDatei2;
   private WavFile outFile1;
   private boolean linkerSound = true;
   private boolean soundlloaded = false;
   private boolean sound2loaded = false;
    private AudioInputStream audioInputStream1;
```

```
private AudioInputStream audioInputStream2;
    private Clip clip1;
   private Clip clip2;
    public boolean getLinkerSound() {
      return linkerSound;
30
    public void setLinkerSound(boolean neuerWert) {
      linkerSound = neuerWert;
35
    public void initOut1() throws WavFileException,
       IOException {
      DisplayDetails();
      if (soundlloaded) {
        // Speichere direkt Out1
        int frames = (int) soundDatei1.getNumFrames();
40
        WavFile outFile = WavFile.newWavFile(new File(".../
           out1.wav"),
            soundDatei1.getNumChannels(), soundDatei1.
               getNumFrames(),
            soundDateil.getValidBits(), soundDateil.
               getSampleRate());
        double[] buffer = new double[frames];
45
        // Datei in Outfile betragen
        soundDateil.readFrames(buffer, frames);
        outFile.writeFrames(buffer, frames);
        outFile.close();
        ladeSound(".../out1.wav");
      }
    }
    public void initOut2() throws WavFileException,
       IOException {
      DisplayDetails();
55
      if (sound2loaded) {
        // Speichere direkt Out1
        int frames = (int) soundDatei2.getNumFrames();
        WavFile outFile = WavFile.newWavFile(new File(".../
           out2.wav"),
            soundDatei2.getNumChannels(), soundDatei2.
60
               getNumFrames(),
            soundDatei2.getValidBits(), soundDatei2.
               getSampleRate());
        double[] buffer = new double[frames];
        // Datei in Outfile betragen
        soundDatei2.readFrames(buffer, frames);
        outFile.writeFrames(buffer, frames);
```

```
outFile.close();
        ladeSound("../out2.wav");
      }
    }
70
    public void ladeSound(String dateipfad) throws
        IOException,
        WavFileException {
      if (linkerSound) {
75
        if (dateipfad != null) {
           soundDatei1 = WavFile.openWavFile(new File(
              dateipfad));
          soundlloaded = true;
        }
      } else {
        if (dateipfad != null) {
           soundDatei2 = WavFile.openWavFile(new File(
85
              dateipfad));
          sound2loaded = true;
         }
      }
    public void play1() {
      if (soundlloaded) {
        try {
          audioInputStream1 = AudioSystem
               .getAudioInputStream(new File("../out1.wav")
                   .getAbsoluteFile());
100
          clip1 = AudioSystem.getClip();
          clip1.open(audioInputStream1);
          clip1.start();
         } catch (Exception ex) {
          System.out.println("Error with playing sound.");
105
          ex.printStackTrace();
      }
110
    }
    public void play2() {
      if (sound2loaded) {
        try {
115
```

```
audioInputStream2 = AudioSystem
               .getAudioInputStream(new File("../out2.wav")
                   .getAbsoluteFile());
           clip2 = AudioSystem.getClip();
120
           clip2.open(audioInputStream2);
           clip2.start();
         } catch (Exception ex) {
          System.out.println("Error with playing sound.");
           ex.printStackTrace();
125
         }
      }
    }
130
    public void stop1(){
      if (soundlloaded) {
        try {
135
          clip1.stop();
         } catch (Exception ex) {
          System.out.println("Error with playing sound.");
          ex.printStackTrace();
         }
140
      }
    }
    public void stop2(){
145
      if (soundlloaded) {
        try {
           clip2.stop();
150
         } catch (Exception ex) {
          System.out.println("Error with playing sound.");
          ex.printStackTrace();
         }
     }
155
    }
    // Diese Details sollen zusammen mit dem Dateinamen
        angezeigt werden
    public String DisplayDetails() {
160
      if (linkerSound) {
         // Get the number of audio channels in the wav
            file
        int numChannels = soundDatei1.getNumChannels();
165
```

```
if (numChannels != 1) {
          soundlloaded = false;
          return "Bitte Sound mit nur einem Kanal laden,
              also Mono.";
170
        long Frames = soundDatei1.getNumFrames();
        int validBits = soundDatei1.getValidBits();
        int numFrames = (int) Frames;
175
        long sampleRate = soundDatei1.getSampleRate(); //
            Samples per second
        String details = "Anzahl Kanle: " + numChannels +
             "\n"
            + "Anzahl validBits: " + validBits + "\n" + "
180
               Samplerate: "
            + sampleRate + "\n" + "Anzahl Frames: " +
                numFrames;
        return details;
      } else {
185
        // Get the number of audio channels in the wav
            file
        int numChannels = soundDatei2.getNumChannels();
        if (numChannels != 1) {
          sound2loaded = false;
190
          return "Bitte Sound mit nur einem Kanal laden,
              also Mono.";
        }
        long Frames = soundDatei2.getNumFrames();
        int validBits = soundDatei2.getValidBits();
195
        int numFrames = (int) Frames;
        long sampleRate = soundDatei2.getSampleRate(); //
            Samples per second
200
        String details = "Anzahl Kanle: " + numChannels +
             "\n"
            + "Anzahl validBits: " + validBits + "\n" + "
                Samplerate: "
            + sampleRate + "\n" + "Anzahl Frames: " +
                numFrames;
        return details;
205
      }
```

```
}
210
    // public void
    public void speichere(String dateipfad) throws
        IOException,
        WavFileException {
      if (linkerSound) {
215
        if (soundlloaded) {
          int frames = (int) soundDatei1.getNumFrames();
          WavFile outFile = WavFile.newWavFile(new File(
              dateipfad),
               soundDatei1.getNumChannels(),
               soundDatei1.getNumFrames(), soundDatei1.
                  getValidBits(),
               soundDatei1.getSampleRate());
220
          double[] buffer = new double[frames];
           // Datei in Outfile betragen
          soundDateil.readFrames(buffer, frames);
          outFile.writeFrames(buffer, frames);
225
          outFile.close();
        }
      } else {
        if (sound2loaded) {
230
          int frames = (int) soundDatei2.getNumFrames();
          WavFile outFile = WavFile.newWavFile(new File(
              dateipfad),
               soundDatei2.getNumChannels(),
               soundDatei2.getNumFrames(), soundDatei2.
                  getValidBits(),
               soundDatei2.getSampleRate());
235
          double[] buffer = new double[frames];
           // Datei in Outfile betragen
          soundDatei2.readFrames(buffer, frames);
240
          outFile.writeFrames(buffer, frames);
          outFile.close();
        }
245
    }
    public void rueckwaerts() throws IOException,
250
       WavFileException {
      WavFile soundDatei;
      String outfile;
```

```
if (linkerSound) {
        if (!soundlloaded)
          return;
255
        soundDatei = soundDatei1;
        outfile = "../out1.wav";
      } else {
        if (!sound2loaded)
260
          return;
        soundDatei = soundDatei2;
        outfile = "../out2.wav";
      }
      // Get the number of audio channels in the wav file
265
      int numChannels = soundDatei.getNumChannels();
      long Frames = soundDatei.getNumFrames();
      int validBits = soundDatei.getValidBits();
      int numFrames = (int) Frames;
      long sampleRate = soundDatei.getSampleRate(); //
270
          Samples per second
      // Erzeuge Buffer um die Eingabedatei darin zu
          lagern
      double[] buffer = new double[numFrames];
      double[] ausgabebuffer;
      int framesRead;
275
      // Read frames into buffer
      framesRead = soundDatei.readFrames(buffer, numFrames
          );
280
      ausgabebuffer = SoundOperationen.rueckwaerts(buffer)
      WavFile outFile1 = WavFile.newWavFile(new File(
          outfile), numChannels,
          numFrames, validBits, sampleRate);
      outFile1.writeFrames(ausgabebuffer, framesRead);
285
      outFile1.close();
      ladeSound(outfile);
290
    }
    public void aneinanderhaengen() throws IOException,
       WavFileException {
      if (!soundlloaded)
295
        return;
      if (!sound2loaded)
        return;
```

```
boolean speicher = linkerSound;
      linkerSound = false;
300
      ladeSound(".../out2.wav");
      linkerSound = speicher;
      // Get the number of audio channels in the wav file
      int numChannels = soundDatei1.getNumChannels();
305
      int validBits = soundDatei1.getValidBits();
      int numFrames1 = (int) soundDatei1.getNumFrames();
      int numFrames2 = (int) soundDatei2.getNumFrames();
      int numFramesZusammen = numFrames1 + numFrames2;
310
      long sampleRate = soundDatei1.getSampleRate(); //
          Samples per second
      // Erzeuge Buffer um die Eingabedateien darin zu
          lagern
      double[] buffer1 = new double[numFrames1];
      double[] buffer2 = new double[numFrames2];
315
      double[] ausgabebuffer = new double[
          numFramesZusammen];
      // Erste Datei in ausgabebuffer bertragen
      soundDateil.readFrames(buffer1, numFrames1);
320
      // Zweite Datei in ausgabebuffer bertragen
      soundDatei2.readFrames(buffer2, numFrames2);
325
      ausgabebuffer = SoundOperationen.aneinanderhaengen(
          buffer1, buffer2);
      // Erzeuge ausgabedatei mit selber lnge und
          samplerate wie die
      // Eingabedateien
      outFile1 = WavFile.newWavFile(new File("../out1.wav"
330
          ), numChannels,
          numFramesZusammen, validBits, sampleRate);
      outFile1.writeFrames(ausgabebuffer,
          numFramesZusammen);
      // Close the wavFiles
      outFile1.close();
335
      speicher = linkerSound;
      linkerSound = true;
      ladeSound(".../out1.wav");
      linkerSound = speicher;
340
    public void abschneidenvorne (int sekunden) throws
       IOException,
```

```
WavFileException {
      WavFile soundDatei;
      String outfile;
345
      if (linkerSound) {
        if (!soundlloaded)
          return;
        soundDatei = soundDateil;
        outfile = "../out1.wav";
350
      } else {
        if (!sound2loaded)
          return;
        soundDatei = soundDatei2;
        outfile = "../out2.wav";
355
      long sampleRate = soundDatei.getSampleRate();
      int framesZuSchneiden = (int) (Math.abs(sekunden) *
          sampleRate);
      int frames = (int) soundDatei.getNumFrames();
360
      if (framesZuSchneiden > frames) {
        framesZuSchneiden = frames;
      int framesZuSpeichern = frames - framesZuSchneiden;
365
      double[] buffer = new double[frames];
      double[] ausgabebuffer = new double[
          framesZuSpeichern];
      // Datei in Outfile betragen
      soundDatei.readFrames(buffer, frames);
370
      ausgabebuffer = SoundOperationen.abschneidenVorne(
          buffer, framesZuSchneiden);
      // Speichern
      outFile1 = WavFile.newWavFile(new File(outfile), 1,
          framesZuSpeichern,
          soundDatei.getValidBits(), sampleRate);
375
      outFile1.writeFrames(ausgabebuffer,
          framesZuSpeichern);
      outFile1.close();
      ladeSound(outfile);
380
    }
    public void abschneidenhinten (int sekunden) throws
        IOException,
        WavFileException {
      WavFile soundDatei;
      String outfile;
385
      if (linkerSound) {
        if (!soundlloaded)
          return;
```

```
soundDatei = soundDatei1;
        outfile = "../out1.wav";
390
      } else {
        if (!sound2loaded)
          return;
        soundDatei = soundDatei2;
        outfile = "../out2.wav";
395
      long sampleRate = soundDatei.getSampleRate();
      int framesZuSchneiden = (int) (Math.abs(sekunden) *
          sampleRate);
      int frames = (int) soundDatei.getNumFrames();
      if (framesZuSchneiden > frames) {
400
        framesZuSchneiden = frames;
      }
      int framesZuSpeichern = frames - framesZuSchneiden;
      double[] buffer = new double[frames];
      double[] ausgabebuffer = new double[
405
          framesZuSpeichern];
      // Datei in buffer betragen
      soundDatei.readFrames(buffer, frames);
      ausgabebuffer = SoundOperationen.abschneidenHinten(
          buffer, framesZuSchneiden);
      // Speichern
      outFile1 = WavFile.newWavFile(new File(outfile), 1,
          framesZuSpeichern,
          soundDatei.getValidBits(), sampleRate);
415
      outFile1.writeFrames(ausgabebuffer,
          framesZuSpeichern);
      outFile1.close();
      if (linkerSound) {
        ladeSound("../out1.wav");
420
      } else {
        ladeSound("../out2.wav");
    }
425
  }
                  Listing D.5: Klasse MedienGUI.java
 1 import java.io.IOException;
  import java.util.regex.Pattern;
  import org.eclipse.swt.SWT;
 5 import org.eclipse.swt.events.SelectionAdapter;
  import org.eclipse.swt.events.SelectionEvent;
  import org.eclipse.swt.graphics.Image;
```

```
import org.eclipse.swt.graphics.ImageData;
 import org.eclipse.swt.graphics.Point;
import org.eclipse.swt.graphics.Rectangle;
 import org.eclipse.swt.layout.FillLayout;
 import org.eclipse.swt.layout.FormAttachment;
 import org.eclipse.swt.layout.FormData;
 import org.eclipse.swt.layout.FormLayout;
import org.eclipse.swt.layout.GridData;
 import org.eclipse.swt.layout.GridLayout;
 import org.eclipse.swt.widgets.Button;
 import org.eclipse.swt.widgets.Canvas;
 import org.eclipse.swt.widgets.Composite;
20 import org.eclipse.swt.widgets.Display;
  import org.eclipse.swt.widgets.FileDialog;
 import org.eclipse.swt.widgets.Label;
 import org.eclipse.swt.widgets.Menu;
 import org.eclipse.swt.widgets.MenuItem;
25 import org.eclipse.swt.widgets.Shell;
 import org.eclipse.swt.widgets.Text;
 import soundManipulation.SoundManipulator;
 import soundManipulation.WavFileException;
30 import bildManipulator.BildManipulator;
 import com.cloudgarden.resource.SWTResourceManager;
  * This code was edited or generated using CloudGarden's
       Jigloo SWT/Swing GUI
   * Builder, which is free for non-commercial use. If
      Jigloo is being used
   * commercially (ie, by a corporation, company or
      business for any purpose
   * whatever) then you should purchase a license for each
       developer using Jigloo.
   * Please visit www.cloudgarden.com for details. Use of
      Jigloo implies
  * acceptance of these licensing terms. A COMMERCIAL
      LICENSE HAS NOT BEEN
   * PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE
      CANNOT BE USED LEGALLY FOR
   * ANY CORPORATE OR COMMERCIAL PURPOSE.
   */
45 public class MedienGUI extends org.eclipse.swt.widgets.
     Composite {
   private Text textDateiInfo2;
   private Text textDateiinfo;
   private Label labelBildmanipula;
   private Label labelSoundmanipu;
   private Canvas canvasRechts;
```

```
private Canvas canvasLinks;
   private Composite composite1;
   private Menu hauptleisteMenu;
   private MenuItem bildOffnenFileMenuItem;
   private Menu menu2;
   private MenuItem offnenFileMenuItem;
   private Menu dateifileMenu;
   private MenuItem dateiMenuItem;
60
      // Register as a resource user - SWTResourceManager
      // handle the obtaining and disposing of resources
      SWTResourceManager.registerResourceUser(this);
65
   }
   private MenuItem sound1OffnenFileMenuItem;
   private MenuItem speichernFileMenuItem;
   private Menu menu3;
   private MenuItem bildSpeichernFileMenuItem;
   private MenuItem sound1SpeichernFileMenuItem;
   private MenuItem bildMenuItem;
   private Menu bildfileMenu;
   private MenuItem vorgegebenMenuItem;
   private Menu vorgegebenMenu;
   private MenuItem schwarzweissMenuItem;
   private MenuItem spiegelvertikalMenuItem;
   private MenuItem hintergrundMenuItem;
   private MenuItem farbumkehrMenuItem;
  private MenuItem farbfilterMenuItem;
   private Menu farbfilterMenu;
   private MenuItem rotfilterMenuItem;
   private MenuItem grnfilterMenuItem;
   private MenuItem blaufilterMenuItem;
   private MenuItem eigeneFunktionMenuItem;
   private Menu eigeneFunktionMenu;
   private MenuItem eigenelMenuItem;
   private MenuItem eigene2MenuItem;
   private Button Stopbutton2;
   private Button stopbutton1;
   private Composite composite5;
   private Button sound2AbspielenButton;
   private Button sound1AbspielenButton;
   private Button rechterSoundButton;
   private Button linkerSoundButton;
   private Composite composite4;
   private Label label2;
   private Label label1;
   private Button hintenButton;
   private Button vorneButton;
   private Composite composite3;
   private Button rechtesFensterButton;
```

```
private Button linkesFensterButton;
    private Composite composite2;
    private Text dauerText;
105
    private Label abschneidenDauerLabel;
    private MenuItem eigene3MenuItem;
    private MenuItem eigene4MenuItem;
    private MenuItem eigene5MenuItem;
    private MenuItem soundmanipulationMenuItem;
    private Menu soundmaniMenu;
    private MenuItem ruckwartsMenuItem;
    private MenuItem abschneidenMenuItem;
    private MenuItem hangenMenuItem;
    private BildManipulator manipulator;
    private SoundManipulator soundman;
    public MedienGUI(Composite parent, int style) {
      super(parent, style);
      manipulator = new BildManipulator();
120
      soundman = new SoundManipulator();
      initGUI();
    }
    /**
125
     * Initializes the GUI.
     */
    private void initGUI() {
      try {
        this.setSize(1250, 600);
130
        this.setBackground(SWTResourceManager.getColor
            (192, 192, 192));
        FillLayout thisLayout = new FillLayout(
            org.eclipse.swt.SWT.HORIZONTAL);
        this.setLayout(thisLayout);
135
          composite1 = new Composite(this, SWT.NONE);
          FormLayout composite1Layout = new FormLayout();
          composite1.setLayout(composite1Layout);
          composite1.setBackground(SWTResourceManager.
              getColor(64, 128,
              128));
140
           {
            Stopbutton2 = new Button(composite1, SWT.PUSH
                | SWT.CENTER);
            FormData Stopbutton2LData = new FormData();
            Stopbutton2LData.left = new FormAttachment(0,
                 1000, 1089);
            Stopbutton2LData.top = new FormAttachment(0,
145
                1000, 538);
            Stopbutton2LData.width = 36;
            Stopbutton2LData.height = 25;
            Stopbutton2.setLayoutData(Stopbutton2LData);
            Stopbutton2.setText("Stop");
```

```
Stopbutton2.addSelectionListener (new
150
                SelectionAdapter() {
              public void widgetSelected(SelectionEvent
                  evt) {
                 Stopbutton2WidgetSelected(evt);
             });
           }
155
             stopbutton1 = new Button(composite1, SWT.PUSH
                | SWT.CENTER);
            FormData stopbutton1LData = new FormData();
             stopbutton1LData.left = new FormAttachment(0,
                 1000, 610);
160
             stopbutton1LData.top = new FormAttachment(0,
                1000, 538);
             stopbutton1LData.width = 36;
             stopbutton1LData.height = 25;
             stopbutton1.setLayoutData(stopbutton1LData);
             stopbutton1.setText("Stop");
             stopbutton1.addSelectionListener(new
165
                SelectionAdapter() {
               public void widgetSelected(SelectionEvent
                 stopbutton1WidgetSelected(evt);
             });
           }
170
             sound2AbspielenButton = new Button(composite1,
                 SWT.PUSH
                 | SWT.CENTER);
            FormData sound2AbspielenButtonLData = new
                FormData();
175
             sound2AbspielenButtonLData.left = new
                FormAttachment (739,
                 1000, 0);
             sound2AbspielenButtonLData.top = new
                FormAttachment (897,
                 1000, 0);
             sound2AbspielenButtonLData.width = 145;
             sound2AbspielenButtonLData.height = 25;
180
             sound2AbspielenButton
                 .setLayoutData(sound2AbspielenButtonLData)
             sound2AbspielenButton.setText("Rechten Sound
                abspielen");
             sound2AbspielenButton
                 .addSelectionListener(new SelectionAdapter
185
                   public void widgetSelected(
                      SelectionEvent evt) {
```

```
sound2AbspielenButtonWidgetSelected(
                        evt);
                   }
                 });
190
           }
             sound1AbspielenButton = new Button(composite1,
                 SWT.PUSH
                 | SWT.CENTER);
            FormData sound1AbspielenButtonLData = new
                FormData();
195
             sound1AbspielenButtonLData.left = new
                FormAttachment (369,
                 1000, 0);
             sound1AbspielenButtonLData.top = new
                FormAttachment (897,
                 1000, 0);
             sound1AbspielenButtonLData.width = 137;
             sound1AbspielenButtonLData.height = 25;
200
             sound1AbspielenButton
                 .setLayoutData(sound1AbspielenButtonLData)
             sound1AbspielenButton.setText("Linken Sound
                abspielen");
             sound1AbspielenButton
                 .addSelectionListener(new SelectionAdapter
205
                   public void widgetSelected(
                      SelectionEvent evt) {
                     sound1AbspielenButtonWidgetSelected(
                        evt);
                 });
           }
210
             composite3 = new Composite(composite1, SWT.
                NONE);
            GridLayout composite3Layout = new GridLayout()
             composite3Layout.makeColumnsEqualWidth = true;
            FormData composite3LData = new FormData();
215
             composite3LData.left = new FormAttachment(0,
                1000, 12);
             composite3LData.top = new FormAttachment(842,
                1000, 0);
             composite3LData.width = 139;
             composite3LData.height = 58;
             composite3.setLayoutData(composite3LData);
220
             composite3.setLayout(composite3Layout);
             composite3.setBackground(SWTResourceManager.
                getColor(64,
                 128, 128));
```

```
vorneButton = new Button(composite3, SWT.
225
                  RADIO
                   | SWT.LEFT);
               GridData button3LData = new GridData();
               vorneButton.setLayoutData(button3LData);
              vorneButton.setText("Abschneiden vorne");
               vorneButton.setBackground(SWTResourceManager
230
                  .getColor(
                   64, 128, 128));
               vorneButton.setSelection(true);
              hintenButton = new Button(composite3, SWT.
235
                  RADIO
                   | SWT.LEFT);
              GridData button4LData = new GridData();
              button4LData.widthHint = 130;
              button4LData.heightHint = 16;
240
              hintenButton.setLayoutData(button4LData);
              hintenButton.setText("Abschneiden hinten");
              hintenButton.setBackground(
                  SWTResourceManager.getColor(
                   64, 128, 128));
             }
           }
245
            composite2 = new Composite(composite1, SWT.
                NONE);
            GridLayout composite2Layout = new GridLayout()
            composite2Layout.makeColumnsEqualWidth = true;
            FormData composite2LData = new FormData();
250
            composite2LData.left = new FormAttachment(0,
                1000, 12);
            composite2LData.top = new FormAttachment(0,
                1000, 59);
            composite2LData.width = 110;
            composite2LData.height = 158;
            composite2.setLayoutData(composite2LData);
255
            composite2.setLayout(composite2Layout);
            composite2.setBackground(SWTResourceManager.
                getColor(64,
                 128, 128));
260
               linkesFensterButton = new Button(composite2,
                   SWT.RADIO
                   | SWT.LEFT);
              GridData button1LData = new GridData();
              button1LData.widthHint = 112;
              button1LData.heightHint = 16;
```

```
linkesFensterButton.setLayoutData(
265
                  button1LData);
               linkesFensterButton.setText("Linkes Fenster"
                  );
               linkesFensterButton.setBackground(
                  SWTResourceManager
                   .getColor(64, 128, 128));
               linkesFensterButton.setSelection(true);
               linkesFensterButton
270
                   .addSelectionListener (new
                      SelectionAdapter() {
                     public void widgetSelected(
                         SelectionEvent evt) {
                       linkesFensterButtonWidgetSelected(
                           evt);
                     }
275
                   });
             }
               rechtesFensterButton = new Button(composite2
                  , SWT.RADIO
                   | SWT.LEFT);
280
               GridData button2LData = new GridData();
               button2LData.widthHint = 132;
               button2LData.heightHint = 16;
               rechtesFensterButton.setLayoutData(
                  button2LData);
               rechtesFensterButton.setText("Rechtes
285
                  Fenster");
               rechtesFensterButton.setBackground(
                  SWTResourceManager
                   .getColor(64, 128, 128));
               rechtesFensterButton
                   .addSelectionListener (new
                      SelectionAdapter() {
                     public void widgetSelected(
290
                         SelectionEvent evt) {
                       rechtesFensterButtonWidgetSelected(
                          evt);
                     }
                   });
             }
295
            FormData dauerTextLData = new FormData();
            dauerTextLData.left = new FormAttachment(0,
                1000, 190);
            dauerTextLData.top = new FormAttachment(790,
300
                1000, 0);
            dauerTextLData.width = 16;
             dauerTextLData.height = 15;
            dauerText = new Text(composite1, SWT.NONE);
```

```
dauerText.setLayoutData(dauerTextLData);
             dauerText.setText("0");
305
             abschneidenDauerLabel = new Label (composite1,
                SWT.NONE);
            FormData abschneidenDauerLabelLData = new
                FormData();
             abschneidenDauerLabelLData.left = new
310
                FormAttachment (0,
                 1000, 12);
             abschneidenDauerLabelLData.top = new
                FormAttachment (790,
                 1000, 0);
            abschneidenDauerLabelLData.width = 172;
             abschneidenDauerLabelLData.height = 15;
315
             abschneidenDauerLabel
                 .setLayoutData(abschneidenDauerLabelLData)
             abschneidenDauerLabel
                 .setText("Abschneiden, Dauer [Sekunden]:")
             abschneidenDauerLabel.setBackground(
320
                SWTResourceManager
                 .getColor(64, 128, 128));
           }
            textDateiInfo2 = new Text(composite1, SWT.
325
                MULTI | SWT.WRAP);
            FormData textDateiInfo2LData = new FormData();
            textDateiInfo2LData.width = 365;
            textDateiInfo2LData.height = 193;
            textDateiInfo2LData.right = new FormAttachment
                (1000, 1000,
                 -63);
330
            textDateiInfo2LData.bottom = new
                FormAttachment (1000, 1000,
                 -74);
             textDateiInfo2LData.left = new FormAttachment
                (653, 1000, 0);
            textDateiInfo2LData.top = new FormAttachment
                (555, 1000, 0);
            textDateiInfo2.setLayoutData(
335
                textDateiInfo2LData);
            textDateiInfo2.setText("Bitte Sound2 ffnen");
            textDateiInfo2.setEditable(false);
             textDateiInfo2.setBackground(
                SWTResourceManager.getColor(
                 64, 128, 128));
            textDateiInfo2.setFont(SWTResourceManager.
340
                getFont (
```

```
"Showcard Gothic", 12, 2, false, false));
          }
            textDateiinfo = new Text(composite1, SWT.MULTI
                 | SWT.WRAP);
            FormData textDateiinfoLData = new FormData();
345
            textDateiinfoLData.width = 347;
            textDateiinfoLData.height = 193;
            textDateiinfoLData.right = new FormAttachment
                (562, 1000, 0);
            textDateiinfoLData.bottom = new FormAttachment
                (1000, 1000,
                 -74);
350
            textDateiinfoLData.left = new FormAttachment
                (279, 1000, 0);
            textDateiinfoLData.top = new FormAttachment
                (555, 1000, 0);
            textDateiinfo.setLayoutData(textDateiinfoLData
                );
            textDateiinfo.setText("Bitte Sound1 ffnen");
            textDateiinfo.setEditable(false);
355
            textDateiinfo.setBackground(SWTResourceManager
                .getColor(64,
                128, 128));
            textDateiinfo.setFont(SWTResourceManager.
                getFont (
                 "Showcard Gothic", 12, 2, false, false));
            textDateiinfo.setOrientation(SWT.VERTICAL);
360
          }
            labelBildmanipula = new Label(composite1, SWT.
            FormData labelBildmanipulaLData = new FormData
                ();
            labelBildmanipulaLData.left = new
                FormAttachment (0, 1000,
                12);
            labelBildmanipulaLData.top = new
                FormAttachment(0, 1000, 40);
            labelBildmanipulaLData.width = 148;
            labelBildmanipulaLData.height = 20;
            labelBildmanipula.setLayoutData(
370
                labelBildmanipulaLData);
            labelBildmanipula.setText("(nur .jpg und .jpeg
                )");
            labelBildmanipula.setBackground(
                SWTResourceManager
                 .getColor(64, 128, 128));
            labelBildmanipula.setFont(SWTResourceManager.
                getFont (
                 "Showcard Gothic", 10, 2, false, false));
           }
```

```
labelSoundmanipu = new Label(composite1, SWT.
                NONE);
            FormData labelSoundmanipuLData = new FormData
                ();
            labelSoundmanipuLData.left = new
380
                FormAttachment(0, 1000, 12);
            labelSoundmanipuLData.top = new FormAttachment
                (555, 1000, 0);
            labelSoundmanipuLData.width = 211;
            labelSoundmanipuLData.height = 21;
            labelSoundmanipu.setLayoutData(
                labelSoundmanipuLData);
            labelSoundmanipu.setText("Soundmanipulation");
385
            labelSoundmanipu.setBackground(
                SWTResourceManager.getColor(
                 64, 128, 128));
            labelSoundmanipu.setFont(SWTResourceManager.
                getFont (
                 "Showcard Gothic", 12, 2, false, false));
          }
390
            FormData canvasRechtsLData = new FormData();
            canvasRechtsLData.top = new FormAttachment(0,
                1000, 12);
            canvasRechtsLData.width = 250;
            canvasRechtsLData.height = 191;
395
            canvasRechtsLData.right = new FormAttachment
                (1000, 1000,
                 -12);
            canvasRechtsLData.left = new FormAttachment
                (626, 1000, 0);
            canvasRechtsLData.bottom = new FormAttachment
                (508, 1000, 0);
400
            canvasRechts = new Canvas(composite1, SWT.NONE
                );
            canvasRechts.setLayoutData(canvasRechtsLData);
            canvasRechts.setOrientation(SWT.HORIZONTAL);
            FormData canvasLinksLData = new FormData();
405
            canvasLinksLData.top = new FormAttachment(0,
                1000, 12);
            canvasLinksLData.width = 252;
            canvasLinksLData.height = 191;
            canvasLinksLData.right = new FormAttachment
                (593, 1000, 0);
            canvasLinksLData.left = new FormAttachment
410
                (233, 1000, 0);
            canvasLinksLData.bottom = new FormAttachment
                (508, 1000, 0);
```

```
canvasLinks = new Canvas(composite1, SWT.NONE)
            canvasLinks.setLayoutData(canvasLinksLData);
415
            label1 = new Label(composite1, SWT.NONE);
            FormData label1LData = new FormData();
            label1LData.left = new FormAttachment(0, 1000,
            label1LData.top = new FormAttachment(0, 1000,
                12);
            label1LData.width = 200;
420
            label1LData.height = 28;
            label1.setLayoutData(label1LData);
            label1.setText("Bildmanipulation ");
            label1.setBackground(SWTResourceManager.
                getColor(64, 128,
                128));
425
            label1.setFont(SWTResourceManager.getFont(
                 "Showcard Gothic", 12, 2, false, false));
            label2 = new Label(composite1, SWT.NONE);
430
            FormData label2LData = new FormData();
            label2LData.left = new FormAttachment(10,
                1000, 0);
            label2LData.top = new FormAttachment(600,
                1000, 0);
            label2LData.width = 178;
            label2LData.height = 20;
435
            label2.setLayoutData(label2LData);
            label2.setText("(nur .wav, kein stereo)");
            label2.setBackground(SWTResourceManager.
                getColor(64, 128,
                128));
            label2.setFont(SWTResourceManager.getFont(
440
                 "Showcard Gothic", 10, 2, false, false));
          }
            FormData composite4LData = new FormData();
            composite4LData.left = new FormAttachment(10,
445
                1000, 0);
            composite4LData.top = new FormAttachment(684,
                1000, 0);
            composite4LData.width = 139;
            composite4LData.height = 58;
            composite4 = new Composite(composite1, SWT.
                NONE);
            GridLayout composite4Layout = new GridLayout()
450
            composite4Layout.makeColumnsEqualWidth = true;
            composite4.setLayout(composite4Layout);
```

```
composite4.setLayoutData(composite4LData);
             composite4.setBackground(SWTResourceManager.
                getColor(64,
                 128, 128));
455
               linkerSoundButton = new Button(composite4,
                  SWT.RADIO
                   | SWT.LEFT);
               GridData button1LData2 = new GridData();
               linkerSoundButton.setLayoutData(
460
                  button1LData2);
               linkerSoundButton.setText("linker Sound");
               linkerSoundButton.setBackground(
                  SWTResourceManager
                   .getColor(64, 128, 128));
               linkerSoundButton.setSelection(true);
               linkerSoundButton
465
                   .addSelectionListener (new
                      SelectionAdapter() {
                     public void widgetSelected(
                         SelectionEvent evt) {
                       linkerSoundButtonWidgetSelected(evt)
470
                     }
                   });
             }
               rechterSoundButton = new Button(composite4,
                  SWT.RADIO
                   | SWT.LEFT);
475
               GridData button2LData1 = new GridData();
               button2LData1.widthHint = 130;
               button2LData1.heightHint = 16;
               rechterSoundButton.setLayoutData(
                  button2LData1);
               rechterSoundButton.setText("rechter Sound");
480
               rechterSoundButton.setBackground(
                  SWTResourceManager
                   .getColor(64, 128, 128));
               rechterSoundButton
                   .addSelectionListener (new
                      SelectionAdapter() {
485
                     public void widgetSelected(
                         SelectionEvent evt) {
                       rechterSoundButtonWidgetSelected(evt
                          );
                     }
                   });
             }
490
```

```
FormData composite5LData = new FormData(1250,
            composite5LData.left = new FormAttachment(0,
                1000, 0);
495
            composite5LData.top = new FormAttachment(534,
                1000, 0);
            composite5LData.width = 1250;
            composite5LData.height = 5;
            composite5LData.right = new FormAttachment
                (1000, 1000, 0);
            composite5 = new Composite(composite1, SWT.
                NONE);
            GridLayout composite5Layout = new GridLayout()
500
            composite5Layout.makeColumnsEqualWidth = true;
            composite5.setLayout(composite5Layout);
            composite5.setLayoutData(composite5LData);
            composite5.setBackground(SWTResourceManager.
                getColor(0, 0,
505
                 0));
            composite5.setBounds(0, 320, 1250, 1);
            composite5.setSize(1250, 5);
          }
        }
510
          hauptleisteMenu = new Menu(getShell(), SWT.BAR);
          getShell().setMenuBar(hauptleisteMenu);
          {
             // Leiste
            dateiMenuItem = new MenuItem(hauptleisteMenu,
515
                SWT.CASCADE);
            dateiMenuItem.setText("Datei");
               // Erster Menupunkt
              dateifileMenu = new Menu(dateiMenuItem);
520
                 offnenFileMenuItem = new MenuItem(
                    dateifileMenu,
                     SWT.CASCADE);
                 offnenFileMenuItem.setText(" ffnen ");
525
                   // zweiter Menpunkt
                   menu2 = new Menu(offnenFileMenuItem);
                   offnenFileMenuItem.setMenu(menu2);
                     bildOffnenFileMenuItem = new MenuItem(
530
                         menu2, SWT.PUSH);
                     bildOffnenFileMenuItem
                         .setText("Bild ffnen");
                     bildOffnenFileMenuItem
```

```
.addSelectionListener (new
535
                             SelectionAdapter() {
                           public void widgetSelected(
                                SelectionEvent evt) {
                             bildOffnenFileMenuItemWidgetSelected
                                 (evt);
                            }
                         });
540
                     sound10ffnenFileMenuItem = new
                        MenuItem(
                         menu2, SWT.PUSH);
                     sound10ffnenFileMenuItem
                          .setText("Sound ffnen");
545
                     sound1OffnenFileMenuItem
                         .addSelectionListener(new
                             SelectionAdapter() {
                           public void widgetSelected(
                                SelectionEvent evt) {
550
                              sound10ffnenFileMenuItemWidgetSelected
                                 (evt);
                         });
                   }
555
                 speichernFileMenuItem = new MenuItem(
                    dateifileMenu,
                     SWT.CASCADE);
                 speichernFileMenuItem.setText("Speichern")
560
                   menu3 = new Menu(speichernFileMenuItem);
                   speichernFileMenuItem.setMenu(menu3);
                     bildSpeichernFileMenuItem = new
                        MenuItem(
                         menu3, SWT.PUSH);
565
                     bildSpeichernFileMenuItem
                          .setText("Bild Speichern");
                     bildSpeichernFileMenuItem
                          .addSelectionListener(new
                             SelectionAdapter() {
                           public void widgetSelected(
570
                                SelectionEvent evt) {
                             bildSpeichernFileMenuItemWidgetSelected
                                 (evt);
                            }
                         });
575
```

```
sound1SpeichernFileMenuItem = new
                        MenuItem(
                         menu3, SWT.PUSH);
                     sound1SpeichernFileMenuItem
                         .setText("Sound Speichern");
                     sound1SpeichernFileMenuItem
580
                         .addSelectionListener(new
                             SelectionAdapter() {
                           public void widgetSelected(
                               SelectionEvent evt) {
                             sound1SpeichernFileMenuItemWidgetSelected
                                 (evt);
585
                         });
                   }
                 }
590
               }
             }
            dateiMenuItem.setMenu(dateifileMenu);
595
            bildMenuItem = new MenuItem(hauptleisteMenu,
                SWT.CASCADE);
            bildMenuItem.setText("Bildmanipulation");
            bildfileMenu = new Menu(bildMenuItem);
600
               vorgegebenMenuItem = new MenuItem(
                  bildfileMenu,
                   SWT.CASCADE);
               vorgegebenMenuItem.setText("Vorgegeben");
605
                 vorgegebenMenu = new Menu(
                    vorgegebenMenuItem);
                 vorgegebenMenuItem.setMenu(vorgegebenMenu)
                    ;
                 {
                   schwarzweissMenuItem = new MenuItem(
                       vorgegebenMenu, SWT.PUSH);
610
                   schwarzweissMenuItem
                       .setText("Schwarz-Wei-Bild");
                   schwarzweissMenuItem
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
615
                             SelectionEvent evt) {
                           schwarzweissMenuItemWidgetSelected
                               (evt);
                         }
```

```
});
620
                   spiegelvertikalMenuItem = new MenuItem(
                       vorgegebenMenu, SWT.PUSH);
                   spiegelvertikalMenuItem
                       .setText("Spiegelkabinett Vertikal")
                   spiegelvertikalMenuItem
625
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
                            spiegelvertikalMenuItemWidgetSelected
                               (evt);
630
                         }
                       });
                   farbfilterMenuItem = new MenuItem(
                       vorgegebenMenu, SWT.CASCADE);
635
                   farbfilterMenuItem.setText("Farbfilter")
                     farbfilterMenu = new Menu(
                         farbfilterMenuItem);
                     farbfilterMenuItem.setMenu(
                        farbfilterMenu);
640
                       rotfilterMenuItem = new MenuItem(
                            farbfilterMenu, SWT.PUSH);
                       rotfilterMenuItem.setText("Rot");
                       rotfilterMenuItem
                            .addSelectionListener (new
645
                               SelectionAdapter() {
                             public void widgetSelected(
                                  SelectionEvent evt) {
                                rotfilterMenuItemWidgetSelected
                                   (evt);
                              }
                            });
650
                       grnfilterMenuItem = new MenuItem(
                            farbfilterMenu, SWT.PUSH);
                       grnfilterMenuItem.setText("Grn");
                       grnfilterMenuItem
655
                            .addSelectionListener(new
                               SelectionAdapter() {
                             public void widgetSelected(
                                  SelectionEvent evt) {
                                {\tt grnfilterMenuItemWidgetSelected}
                                   (evt);
660
                            });
```

```
blaufilterMenuItem = new MenuItem(
                           farbfilterMenu, SWT.PUSH);
665
                       blaufilterMenuItem.setText("Blau");
                       blaufilterMenuItem
                            .addSelectionListener (new
                               SelectionAdapter() {
                             public void widgetSelected(
                                  SelectionEvent evt) {
                               blaufilterMenuItemWidgetSelected
670
                                    (evt);
                           });
                     }
                   }
675
                   hintergrundMenuItem = new MenuItem(
                       vorgegebenMenu, SWT.PUSH);
                   hintergrundMenuItem
                       .setText("Hintergrund austauschen");
                   hintergrundMenuItem
680
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
                           hintergrundMenuItemWidgetSelected
                               (evt);
                         }
685
                       });
                   farbumkehrMenuItem = new MenuItem(
                       vorgegebenMenu, SWT.PUSH);
                   farbumkehrMenuItem.setText("Farben
690
                      umkehren");
                   farbumkehrMenuItem
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
                           farbumkehrMenuItemWidgetSelected
695
                               (evt);
                       });
               }// Ende VorgegebenMenu
700
               eigeneFunktionMenuItem = new MenuItem(
                  bildfileMenu,
                   SWT.CASCADE);
               eigeneFunktionMenuItem.setText("Eigene
                  Funktionen");
```

```
705
                 eigeneFunktionMenu = new Menu(
                     eigeneFunktionMenuItem);
                 eigeneFunktionMenuItem.setMenu(
                    eigeneFunktionMenu);
                 {
                   eigenelMenuItem = new MenuItem(
710
                       eigeneFunktionMenu, SWT.PUSH);
                   eigene1MenuItem
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
715
                           eigenelMenuItemWidgetSelected(
                               evt);
                         }
                       });
                   eigene1MenuItem.setText(manipulator
                       .getNameVonEigener1());
720
                   eigene2MenuItem = new MenuItem(
                       eigeneFunktionMenu, SWT.PUSH);
                   eigene2MenuItem
725
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
                           eigene2MenuItemWidgetSelected(
                               evt);
                         }
730
                       });
                   eigene2MenuItem.setText(manipulator
                       .getNameVonEigener2());
                   eigene3MenuItem = new MenuItem(
                       eigeneFunktionMenu, SWT.PUSH);
                   eigene3MenuItem
735
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
                           eigene3MenuItemWidgetSelected(
                               evt);
740
                         }
                       });
                   eigene3MenuItem.setText(manipulator
                       .getNameVonEigener3());
                   eigene4MenuItem = new MenuItem(
                       eigeneFunktionMenu, SWT.PUSH);
745
                   eigene4MenuItem
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
```

```
SelectionEvent evt) {
                           eigene4MenuItemWidgetSelected(
750
                               evt);
                         }
                       });
                   eigene4MenuItem.setText(manipulator
                       .getNameVonEigener4());
                   eigene5MenuItem = new MenuItem(
755
                       eigeneFunktionMenu, SWT.PUSH);
                   eigene5MenuItem
                       .addSelectionListener (new
                           SelectionAdapter() {
                         public void widgetSelected(
                              SelectionEvent evt) {
760
                           eigene5MenuItemWidgetSelected(
                               evt);
                         }
                       });
                   eigene5MenuItem.setText(manipulator
765
                       .getNameVonEigener5());
                 }
               }// Ende eigene Funktionen MenuItem
             }// Ende Bildmanipulation MenuItem
770
            bildMenuItem.setMenu(bildfileMenu);
             soundmanipulationMenuItem = new MenuItem(
                hauptleisteMenu,
                 SWT.CASCADE);
775
             soundmanipulationMenuItem.setText("
                Soundmanipulation");
               soundmaniMenu = new Menu(
                  soundmanipulationMenuItem);
               soundmanipulationMenuItem.setMenu(
                  soundmaniMenu);
                 ruckwartsMenuItem = new MenuItem(
780
                    soundmaniMenu,
                     SWT.PUSH);
                 ruckwartsMenuItem.setText("Rckwrts");
                 ruckwartsMenuItem
                     .addSelectionListener(new
                         SelectionAdapter() {
                       public void widgetSelected(
785
                           SelectionEvent evt) {
                         ruckwartsMenuItemWidgetSelected(
                             evt);
                     });
790
```

```
abschneidenMenuItem = new MenuItem (
                    soundmaniMenu,
                     SWT.PUSH);
                 abschneidenMenuItem.setText("Abschneiden")
                 abschneidenMenuItem
                     .addSelectionListener(new
795
                        SelectionAdapter() {
                       public void widgetSelected(
                           SelectionEvent evt) {
                         abschneidenMenuItemWidgetSelected(
                             evt);
                       }
                     });
800
                 hangenMenuItem = new MenuItem(
                    soundmaniMenu,
                     SWT.PUSH);
                 hangenMenuItem.setText("Aneinanderhngen")
                 hangenMenuItem
805
                     .addSelectionListener (new
                        SelectionAdapter() {
                       public void widgetSelected(
                           SelectionEvent evt) {
                         hangenMenuItemWidgetSelected(evt);
810
                     });
               }
             }// Ende Men Soundmanipulation
815
          }
        }
        this.layout();
820
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
825
     * Auto-generated main method to display this
     * org.eclipse.swt.widgets.Composite inside a new
         Shell.
    public static void main(String[] args) {
      Display display = Display.getDefault();
      Shell shell = new Shell(display, SWT.CLOSE);
      shell.setMinimumSize(1280, 620);
      MedienGUI inst = new MedienGUI(shell, SWT.NULL);
```

```
Point size = inst.getSize();
835
      shell.setLayout(new FillLayout());
      shell.layout();
      if (size.x == 0 && size.y == 0) {
        inst.pack();
        shell.pack();
840
      } else {
        Rectangle shellBounds = shell.computeTrim(0, 0,
            size.x, size.y);
        shell.setSize(shellBounds.width, shellBounds.
            height);
      }
      shell.open();
845
      while (!shell.isDisposed()) {
        if (!display.readAndDispatch())
          display.sleep();
      }
    }
850
    private String fileDialogOpenWav() {
      FileDialog fileDialog = new FileDialog(this.getShell
          (), SWT.OPEN);
      fileDialog.setText(" ffnen ");
      fileDialog.setFilterPath("C:/");
855
      fileDialog.setFilterNames(FILTER_NAMES_WAV);
      fileDialog.setFilterExtensions(FILTER_EXTS_WAV);
      return fileDialog.open();
860
    private String fileDialogOpenJpg() {
      FileDialog fileDialog = new FileDialog(this.getShell
          (), SWT.OPEN);
      fileDialog.setText(" ffnen ");
      fileDialog.setFilterPath("C:/");
      fileDialog.setFilterNames(FILTER_NAMES_JPG);
      fileDialog.setFilterExtensions(FILTER_EXTS_JPG);
      return fileDialog.open();
    private static final String[] FILTER_NAMES_WAV = { "
870
       Wave (*.wav)", "*.*" };
    private static final String[] FILTER_NAMES_JPG = { "
        Jpeg (*.jpg)", "*.*" };
    // These filter extensions are used to filter which
        files are displayed.
    private static final String[] FILTER_EXTS_WAV = { "*.
       wav", "*.*" };
    private static final String[] FILTER_EXTS_JPG = { "*.
875
        jpg", "*.*" };
    private String fileDialogSaveWav() {
```

```
FileDialog fileDialog = new FileDialog(this.getShell
          (), SWT.SAVE);
      fileDialog.setText("Speichern");
880
      fileDialog.setFilterPath("C:/");
      fileDialog.setFilterNames(FILTER_NAMES_WAV);
      fileDialog.setFilterExtensions(FILTER_EXTS_WAV);
      return fileDialog.open();
    }
885
    private String fileDialogSaveJpg() {
      FileDialog fileDialog = new FileDialog(this.getShell
          (), SWT.SAVE);
      fileDialog.setText("Speichern");
890
      fileDialog.setFilterPath("C:/");
      fileDialog.setFilterNames(FILTER_NAMES_JPG);
      fileDialog.setFilterExtensions(FILTER_EXTS_JPG);
      return fileDialog.open();
895
    private void bildOffnenFileMenuItemWidgetSelected(
       SelectionEvent evt) {
      String dateipfad;
      dateipfad = fileDialogOpenJpg();
      if (dateipfad != null) {
900
        if (Pattern.matches(".*jpg", dateipfad)
             || Pattern.matches(".*JPG", dateipfad)
             || Pattern.matches(".*jpeg", dateipfad)
             || Pattern.matches(".*JPEG", dateipfad)) {
          manipulator.oeffne(dateipfad);
905
          bildInCanvas();
          manipulator.setFileNull();
        }
      // TODO add your code for bildOffnenFileMenuItem.
910
          widgetSelected
    }
    private void sound10ffnenFileMenuItemWidgetSelected(
       SelectionEvent evt) {
915
      try {
        String dateipfad = fileDialogOpenWav();
        if (dateipfad != null) {
          if (Pattern.matches(".*wav", dateipfad)
               || Pattern.matches(".*WAV", dateipfad)) {
            soundman.ladeSound(dateipfad);
920
            if (soundman.getLinkerSound()) {
               soundman.initOut1();
```

```
textDateiinfo.setText(soundman.
                  DisplayDetails());
             } else {
925
               textDateiInfo2.setText(soundman.
                  DisplayDetails());
               soundman.initOut2();
             }
           }
        }
930
      } catch (IOException e) {
      } catch (WavFileException e) {
      }
935
    }
    private void sound1SpeichernFileMenuItemWidgetSelected
        (SelectionEvent evt) {
940
      try {
        soundman.speichere(fileDialogSaveWav());
      } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      } catch (WavFileException e) {
945
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
950
    private void ruckwartsMenuItemWidgetSelected(
        SelectionEvent evt) {
      try {
        soundman.rueckwaerts();
      } catch (IOException e) {
955
        // TODO Auto-generated catch block
        e.printStackTrace();
      } catch (WavFileException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }
960
    }
    private void abschneidenMenuItemWidgetSelected(
        SelectionEvent evt) {
      int sekunden = Integer.parseInt(dauerText.getText())
      try {
965
        if (hintenButton.getSelection()) {
          soundman.abschneidenhinten(sekunden);
         } else {
```

```
soundman.abschneidenvorne(sekunden);
970
         }
         if (soundman.getLinkerSound()) {
           textDateiinfo.setText(soundman.DisplayDetails())
         } else {
           textDateiInfo2.setText(soundman.DisplayDetails()
975
         }
       } catch (IOException e) {
         // TODO Auto-generated catch block
         e.printStackTrace();
       } catch (WavFileException e) {
980
         // TODO Auto-generated catch block
         e.printStackTrace();
       }
985
    private void hangenMenuItemWidgetSelected(
        SelectionEvent evt) {
      try {
         soundman.aneinanderhaengen();
       } catch (IOException e) {
990
         // TODO Auto-generated catch block
         e.printStackTrace();
       } catch (WavFileException e) {
         // TODO Auto-generated catch block
         e.printStackTrace();
995
       }
      boolean wertVonLinkerSound = soundman.getLinkerSound
       soundman.setLinkerSound(true);
       textDateiinfo.setText(soundman.DisplayDetails());
1000
       soundman.setLinkerSound(wertVonLinkerSound);
    private void bildInCanvas() {
       ImageData imagedata = manipulator.getImageData();
       double neueBreite = imagedata.width
1005
           * ((double) canvasLinks.getSize().y / (double)
              imagedata.height);
       ImageData imagedataScaled = imagedata.scaledTo((int)
           neueBreite,
           canvasLinks.getSize().y);
       imagedata = imagedataScaled;
       Image bild = new Image(this.getDisplay(), imagedata)
1010
       if (linkesFensterButton.getSelection()) {
         canvasLinks.setBackgroundImage(bild);
```

```
canvasLinks.setSize(imagedata.width, imagedata.
            height);
       }
1015
         canvasRechts.setBackgroundImage(bild);
         canvasRechts.setSize(imagedata.width, imagedata.
            height);
       }
1020
     }
     private void linkesFensterButtonWidgetSelected(
        SelectionEvent evt) {
1025
       manipulator.setLinkesFenster(true);
     }
     private void rechtesFensterButtonWidgetSelected(
        SelectionEvent evt) {
       manipulator.setLinkesFenster(false);
1030
     private void farbumkehrMenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
1035
               linkesFensterButton
               .getSelection())) {
         manipulator.farbenUmkehren();
         bildInCanvas();
       }
     }
1040
     private void schwarzweissMenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
1045
              linkesFensterButton
                .getSelection())) {
         manipulator.schwarzweiss();
         bildInCanvas();
       }
1050
     }
     private void spiegelvertikalMenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
```

```
.getSelection())
           || (manipulator.getBild2geladen() && !
1055
              linkesFensterButton
               .getSelection())) {
         manipulator.SpiegelKabinettVertikal();
         bildInCanvas();
       }
     }
1060
     private void rotfilterMenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
1065
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
         manipulator.rotFilter();
         bildInCanvas();
       }
     }
1070
     private void hintergrundMenuItemWidgetSelected(
        SelectionEvent evt) {
       if (manipulator.getBild1geladen() && manipulator.
          getBild2geladen()) {
         manipulator.hintergrundtausch();
         bildInCanvas();
1075
       }
     }
     private void bildSpeichernFileMenuItemWidgetSelected(
1080
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() &&!
              linkesFensterButton
               .getSelection()))
         manipulator.speichere(fileDialogSaveJpg());
1085
     }
     private void linkerSoundButtonWidgetSelected(
        SelectionEvent evt) {
       soundman.setLinkerSound(true);
1090
     private void rechterSoundButtonWidgetSelected(
        SelectionEvent evt) {
       soundman.setLinkerSound(false);
     }
```

```
1095
     private void sound1AbspielenButtonWidgetSelected(
        SelectionEvent evt) {
       soundman.play1();
     private void sound2AbspielenButtonWidgetSelected(
1100
        SelectionEvent evt) {
       soundman.play2();
     }
     private void eigenelMenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
1105
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() &&!
              linkesFensterButton
               .getSelection())) {
         manipulator.eigene1();
         bildInCanvas();
1110
       }
     }
1115
     private void eigene2MenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
         manipulator.eigene2();
1120
         bildInCanvas();
       }
     }
     private void eigene3MenuItemWidgetSelected(
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
               linkesFensterButton
               .getSelection())) {
1130
         manipulator.eigene3();
         bildInCanvas();
       }
     }
```

```
private void eigene4MenuItemWidgetSelected(
1135
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
         manipulator.eigene4();
1140
         bildInCanvas();
       }
     }
    private void eigene5MenuItemWidgetSelected(
1145
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
         manipulator.eigene5();
1150
         bildInCanvas();
       }
     }
     private void grnfilterMenuItemWidgetSelected(
1155
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
1160
         manipulator.gruenFilter();
         bildInCanvas();
       }
     }
     private void blaufilterMenuItemWidgetSelected(
1165
        SelectionEvent evt) {
       if ((manipulator.getBild1geladen() &&
          linkesFensterButton
           .getSelection())
           || (manipulator.getBild2geladen() && !
              linkesFensterButton
               .getSelection())) {
         manipulator.blauFilter();
1170
         bildInCanvas();
       }
     }
```

Erklärung

Ich versichere, dass ich die schriftliche Hausarbeit - einschließlich beigefügter Zeichnungen, Kartenskizzen und Darstellungen - selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle deutlich als Entlehnung kenntlich gemacht.