



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Larissa Müller

Interactive Design - Studien der interdisziplinären
Zusammenarbeit von Design und Informatik

Larissa Müller

Interactive Design - Studien der interdisziplinären
Zusammenarbeit von Design und Informatik

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 6. August 2010

Larissa

Thema der Bachelorarbeit

Interactive Design - Studien der interdisziplinären Zusammenarbeit von Design und Informatik

Stichworte

Arduino, LilyPad, Interdisziplinarität, Interactive Design, Wearable Computing, Seamless Interaction, Physical Computing, Sense-Act-Modell

Kurzzusammenfassung

Die Arbeit beschäftigt sich mit der interdisziplinären Zusammenarbeit von Design und Informatik. Es werden Vorstudien in Form von Kursen und studentischen Projekten vorgestellt, bewertet und evaluiert. Bewährte Methoden der Informatik und Anforderungen an ein Kunstprojekt werden unter zur Hilfenahme der Vorstudien, auf Anwendbarkeit innerhalb der interdisziplinären Zusammenarbeit überprüft. Den Abschluss bildet ein Ablaufplan eines erfolgreichen studentischen Projektes.

Larissa Müller

Title of the paper

Interactive Design - Case studies of an interdisciplinary cooperation between design and computer science

Keywords

arduino, lilyPad, interdisciplinarity, interactive design, wearable computing, seamless interaction, physical computing, sense-act-modell

Abstract

This thesis describes the interdisciplinary cooperation of the departments design and computer science. The case studies which have been developed for this work are pinpointed at the beginning. Therefore approved methods from the fields of software engineering and the requirements of an art project are analysed. The main object is to review these two different approaches for a combined practical use and hereby showing the deficit of each. Showing those flaws will help to optimize the cooperation in further project.

Danksagung

Bedanken möchte ich mich bei allen, die diese Arbeit korrekturgelesen haben. Des Weiteren geht ein besonderer Dank an meine Kommilitonen, die dafür gesorgt haben, dass ich viel Freude am Studium hatte. Namentlich erwähnen möchte ich Heiner Perrey und Markus Schüring. Außerdem bedanke ich mich bei allen, die an der erfolgreichen Umsetzung der Vorstudien mitgewirkt haben. Des Weiteren möchte ich mich bei meinen Professoren bedanken, die mich durch Vorlesungen und Projekte zu dieser Arbeit motiviert haben. Besonders Prof. Dr. Gunter Klemke, Prof. Dr. Kai von Luck und Prof. Dr. Birgit Wendholt hatten immer ein offenes Ohr für Fragen jeglicher Art. Die Projektarbeit neben dem Vorlesungsplan hat mir sehr viel Spaß gemacht und dafür gilt mein besonderer Dank André Jeworutzki und Svenja Keune.

Inhaltsverzeichnis

Abbildungsverzeichnis	9
1. Einleitung	11
1.1. Aufbau	12
1.2. Aufbau Designstudium	13
2. Vorstudien	14
2.1. Vorwort zu den Vorstudien	14
2.1.1. Begrifflichkeiten	14
2.1.2. Beteiligte Personen	15
2.1.3. Forschungsgebiete	15
2.2. Erste Kurse an der HAW	17
2.2.1. Pentiment 2008	17
2.2.2. Ambient Awareness	19
2.3. Interdisziplinäre Kurse	19
2.3.1. Experiment Sonia Maas	20
2.3.2. Pentiment Toaster Edwin - Communicate with Creatures	21
2.3.3. Smart Objects	24
2.4. Schülerkurse von kurzer Dauer	26
2.4.1. Konzept: „Snakes on a Plane“	26
2.4.2. Fokus	26
2.4.3. Herbsthochschule für Mädchen Teil 1	26
2.4.4. Herbsthochschule Teil 2	27
2.4.5. Faszination Games 1	29
2.4.6. Faszination Games 2	30
2.5. Einsemestrige Schülerkurse	31
2.5.1. Schülerkurs Teil 1	32
2.5.2. Schülerkurs Teil 2	33
2.5.3. Fazit aus den längeren Schülerkursen	34
2.6. Studentische Projekte	36
2.6.1. Ekaterina Ibraimova	36
2.6.2. Helene All	40
2.6.3. Svenja Keune	41

2.6.4. Bewertung der studentischen Projekte	43
3. Evaluation der Vorstudien	44
3.1. Grundlagen	44
3.1.1. Beschreibung von Interactive Design anhand des Sense-Act-Modells	44
3.1.2. Laborausstattung	50
3.2. Bezug zu den Forschungsgebieten	53
3.2.1. Wearable Computing	53
3.2.2. Gründe für Interactive Design	55
3.3. Kurse	56
3.4. Probleme im Umgang miteinander	56
3.4.1. Kommunikation	56
3.4.2. Arbeitsweise	58
3.4.3. Team-Management	61
3.5. Technische Probleme	61
3.5.1. Debugging	62
3.5.2. Motoren	62
3.5.3. Lichtquellen	62
3.5.4. Energieversorgung	64
3.5.5. Haltbarkeit der Hardware	64
3.6. Fazit	64
4. Methoden für eine interdisziplinäre Zusammenarbeit von Informatik und Design	65
4.1. Sequenzielle Vorgehensmodelle	65
4.2. Agile Softwareentwicklung	66
4.2.1. Extreme Programming	66
4.2.2. Testgetriebene Entwicklung	69
4.2.3. Evolutionäre Arbeitsweise	70
4.2.4. Prototyping	71
4.3. Prüfung der Anwendbarkeit innerhalb der Zusammenarbeit	71
4.4. Anforderungen an ein Kunstprojekt	72
4.4.1. Untersuchung der Anforderungen an Hand der Vorstudien	73
4.4.2. Anforderungen an ein interdisziplinäres Projekt im Bereich Interactive Design	74
4.5. Best Practises	78
4.5.1. Ablaufplan Bachelorarbeit Svenja Keune	78
4.5.2. Prüfung der Anforderungen an ein interdisziplinäres Projekt im Bereich Interactive Design	80
4.5.3. Bewertung	83
5. Zusammenfassung und Ausblick	84

A. Anhang

86

Literaturverzeichnis

94

Abbildungsverzeichnis

2.1. Kein gutes Beispiel für Wearable Computing	18
2.2. Kiste mit intelligenter Alarmanlage	20
2.3. Poster vom Toaster-Edwin-Workshop	21
2.4. Arbeitsraum während des Toaster-Edwin-Workshops	22
2.5. Das Kamerateam bei der Arbeit	23
2.6. Die Wolke und der Schwede	24
2.7. Ausstellung Romantische Maschinen	25
2.8. Vorhang aus LEDs	25
2.9. Herbsthochschule	27
2.10. Snakes-on-a-Plane-Fotoshooting	28
2.11. Abschluss-Fotoshooting Herbsthochschule	28
2.12. Vorgelötete Hardware Faszination Games	29
2.13. Faszination Games 1	30
2.14. Faszination Games 2	31
2.15. Schülerkurs Teil 1	33
2.16. Präsentation am Margaretha-Rothe-Gymnasium	35
2.17. Präsentation am Margaretha-Rothe-Gymnasium	35
2.18. Kollektion Zellteilung einer Stadt	37
2.19. Kollektion Zellteilung einer Stadt	38
2.20. Kollektion Zellteilung einer Stadt	39
2.21. Kollektion Zellteilung einer Stadt	39
2.22. Kollektion Helene All	40
2.23. Herzschlag wird über XBee an LEDs übertragen	41
2.25. Vorstudien Svenja Keune	42
2.24. Thermochromatischer Stoff und sich selbst bewegendes Nitinolkorallenobjekt	42
2.26. Porzellanobjekt	43
3.1. Edwina ein interaktives Musikinstrument	46
3.2. Motor Shield für den Arduino Duemilanove [44]	47
3.3. Erste XBee-Tests	48
3.4. Wave Shield innerhalb der Bachelorarbeit von Svenja Keune	48
3.5. GPS Shield und Data Logger [43]	49

3.6. Arduino Duemilanove [6]	49
3.7. Screenshot Leah Buechley Tutorial [15]	50
3.8. Classroomkits	51
3.9. FTDI-Platine für die Kommunikation mit dem Rechner	51
3.10. Versuchsaufbau mit Krokodilsklemmen	52
3.11. Hilfestellung für eine erfolgreiche Verkabelung	53
3.12. Poster Herbsthochschule [29]	55
3.13. Skizze Svenja Keune Bachelorprojekt	57
3.14. Skizzenbuch Svenja Keune	59
3.15. Beweglicher Wecker	62
3.16. Kuppel	63
3.17. Streumaterial	63
4.1. Wasserfallmodell mit Rücksprungmöglichkeiten [24]	66
4.2. Spiralmodell Boehm [41]	71
4.3. Objekt 1	78
4.4. Bachelorarbeit Svenja Keune	79
4.5. Aufbau der Präsentation	80
4.6. Gemeinsame Entwicklung	81
4.8. Die Drei	82
4.7. 3D-Grafik erstellt in Maya	82
4.9. Die Drei	82
4.10. Aufbau „Der Drei“ im Labor	83
5.1. Arduino Flexible [52]	85

1. Einleitung

Die Möglichkeiten einer Zusammenkunft von Informatik und Design sind äußerst vielfältig. Ein großer Vorteil ist, dass frei von Vorgaben kreativ entwickelt werden kann. Die Informatik kann die Kreativität der Designer nutzen, um Produkte spannender zu gestalten und auch mehr Freude an der Entwicklung zu haben. Designer haben meist Schwierigkeiten die oft abstrakten, technischen Prozesse zu verstehen aber durch eine Kooperation bietet sich ihnen die Möglichkeit, ihre Kreativität um Technik zu erweitern. Es entsteht ein riesiger unglaublich spannender Bereich, der in völlig unterschiedliche Richtungen gehen kann.

Es ist die „künstlerische Freiheit“ die dafür sorgt, dass Designer nur an sehr wenige Regeln gebunden sind. Aus Sicht der Informatik bietet das Design gute Möglichkeiten neue Ideen und Konzepte auszuprobieren und sich dabei von alten Gewohnheiten und Einschränkungen zu lösen. Ein Beispiel aus der Wirtschaft für die erfolgreiche Nutzung von Design für die Informatik ist Apple. Es war Steve Jobs der dafür sorgte, dass sich die Firmenstrategie verändert hat. Die Konzentration auf die eigentlichen Bedürfnisse der Kunden und ein erstklassiges Design waren der Grund für das rasante Wachstum der Firma. In den letzten Jahren hat Apple durch die nahezu perfekte Zusammenarbeit von Design und Informatik einige sehr erfolgreiche Produkte auf den Markt gebracht. [20]

Das Thema dieser Bachelorarbeit ist Interactive Design. Interaktion ist für die Informatik ein essentieller Bereich, es beschreibt das wechselseitige Aufeinandereinfließen von Akteuren und Systemen. Über das Design kann an dieser Stelle die Interaktion interessanter und anregender gestaltet werden.

Interactive Design ist nicht nur für die Informatik interessant sondern auch das Design kann von einer interdisziplinären Zusammenarbeit profitieren. Dies zeigt auch folgendes Zitat: *„Interdisziplinarität ist immer eine Quelle neuer Ansätze und Ideen, auf die man nicht gekommen wäre, hätte man nicht in einen anderen Arbeitsbereich geblickt und andere Meinungen gehört.“* [40]

Ein weiteres Beispiel der Zusammenarbeit von Informatik und Design ist Modedesign. „Kleider machen Leute“, das wusste schon Gottfried Keller. [38] Durch seine Kleidung sagt ein Mensch viel über sich aus und kann sich so selbst darstellen. Die Vielfalt in der Modeindustrie bietet hierbei jedem die Gelegenheit dem Trend zu folgen um sich anzupassen oder

gegen den Strom zu schwimmen. Beides sind wichtige Verhaltensmuster in sozialen Netzwerken. Anpassung kann dem Menschen als Selbstschutz dienen oder der freie Geist seine Persönlichkeit untermauern. Aber auch das Zusammenleben in der Gruppe wird hierdurch vereinfacht. An dieser Stelle kann die Technik genutzt werden, um das Design hierbei zu unterstützen. Aber auch das Design kann der Informatik dienen. Durch das Auslösen von Emotionen beim Betrachter kann die Interaktion mit der Technik bewusst gesteuert werden. Ein Beispiel wäre hier ein Kinderspiel in dem durch auffällige Farben der Spielverlauf erklärt wird. Aber auch in anderen Kontexten kann die Informatik durch das Design unterstützt werden. Ein Beispiel ist hier das betreute Wohnen älterer Menschen. In der Kleidung eingebettete Sensoren können zur Überwachung von Vitalfunktionen genutzt werden. Erreichen diese kritische Werte, kann ein Krankenwagen verständigt werden. Die Sensoren sollen den Menschen natürlich nicht beeinträchtigen, deshalb ist es ein wichtiges Ziel die Technik mit dem Design zu verschmelzen, so dass ggf. niemand ihr Vorhandensein bemerkt.

Eine interdisziplinäre Zusammenarbeit zwischen Design und Informatik ist also sehr spannend und erfolgsversprechend aber leider auch schwierig: *"Kunst und Informatik passen gut zusammen - nur Künstler und Informatiker brauchen etwas Zeit bis sie das merken."* [18] Viele technische und menschliche Probleme müssen ausgeräumt werden, um die Kooperation zu verbessern. Hierzu werden sowohl bewährte Methoden aus der Informatik als auch Anforderungen an ein Kunstprojekt auf ihre Anwendbarkeit geprüft.

1.1. Aufbau

Das Kapitel „Vorstudien“ beinhaltet und erläutert alle Studien des Bereiches Interactive Design, die seit dem Pentiment-Kurs im Sommer 2008 an der HAW durchgeführt wurden. Dieses Kapitel beschreibt sowohl Kurse und deren Konzepte zur Vermittlung von Kompetenzen im Kontext Interactive Design als auch interdisziplinäre studentische Projekte.

Es folgt eine Evaluation der Vorstudien. Diese beinhaltet eine Aufstellung über die verwendete Technik. Daraufhin werden die Probleme beschrieben, die eine solche interdisziplinäre Zusammenarbeit mit sich bringt.

Im letzten Kapitel werden einige Methoden zur Softwareentwicklung vorgestellt und deren Anwendbarkeit innerhalb der interdisziplinären Zusammenarbeit von Informatik und Design geprüft. Daraufhin werden Anforderungen an ein Kunstprojekt auf ihre Tauglichkeit untersucht. Aus den gesammelten Erfahrungen der Vorstudien werden wichtige Aspekte vorgestellt, die in der zukünftigen Zusammenarbeit zu beachten sind. Diese werden an Hand eines erfolgreichen Beispiels, der Bachelorarbeit von Svenja Keune, betrachtet.

1.2. Aufbau Designstudium

Im Department „Design“ der HAW gibt es verschiedene Bachelorstudiengänge, unter anderem Textil-, Mode-, Kostüm- und Kommunikationsdesign. Die Studierenden der unterschiedlichen Studiengänge wählen zu Beginn des Semesters ihre Kurse aus einem Pool von Angeboten. Ein solches Vorlesungsverzeichnis besteht aus vier verschiedenen Arten von Kursen: künstlerische Grundlagen, Labor, Design und Theorie. Die Prüfungsordnungen der verschiedenen Studiengänge legen fest, wieviele Kurse aus jedem dieser Bereiche belegt werden müssen. Laborscheine können in Computerkursen (CAD, Photoshop), Siebdruck, Weben oder der Fertigung von Kleidungsstücken erworben werden. Die Designkurse sind auf die einzelnen Studiengänge abgestimmt, so gibt es hier unter anderem Textil-, Mode- und Kommunikationsdesign. An dieser Stelle dürfen die Studierenden auch die nicht ihrem Studiengang entsprechenden Kurse wählen, wenn die Kapazitäten es zulassen. Die Theoriekurse beinhalten unter anderem Mode- und Kostümgeschichte, sowie Kunst- und Designgeschichte. Beispiele für künstlerische Grundlagen sind Zeichnen und Malerei. Zu diesem Kurspool gehört auch der Smart-Objects-Kurs von Prof. Franziska Hübler (s. Kapitel [2.3.3](#)). Dieser gilt als computergestütztes Experiment.

2. Vorstudien

2.1. Vorwort zu den Vorstudien

Im Folgenden werden einige Begrifflichkeiten erläutert und beteiligte Personen vorgestellt. Im Mittelpunkt stehen die Erläuterungen der betroffenen Forschungsgebiete. Die Zeitspanne der Vorstudien erstreckt sich von Sommer 2008 bis Sommer 2010. Eine Zusammenarbeit ist zukünftig für mindestens zwei weitere Jahre geplant.

2.1.1. Begrifflichkeiten

In diesem Kapitel werden Begriffe erläutert, die immer wieder vorkommen und sehr wichtig für das Verständnis der Arbeit sind.

Arduino

In den Vorstudien werden Arduino Duemilino und Arduino LilyPad verwendet. In beiden Fällen handelt es sich um Mikrokontroller (s. Kapitel [3.1.1](#) und [3.1.1](#)). Arduino [7] selbst ist eine Open Source Physical Computing Platform. Im Kapitel „Evaluation der Vorstudien“ wird auf diese und weitere Hardwarebauteile näher eingegangen (s. Kapitel [3](#)).

Interdisziplinarität

Bei einer interdisziplinären Zusammenarbeit sind verschiedene Einzelwissenschaften betroffen. Jede Einzelwissenschaft hat ihre eigenen Methoden und Arbeitsweisen. Von Interdisziplinarität wird erst dann gesprochen, wenn gemeinsame Methoden entwickelt werden. Ein reines Zusammenführen der einzelnen Ergebnisse reicht nicht aus, sondern eine Zusammenarbeit beider Disziplinen ist erforderlich. Wichtig ist hierbei, dass ein Verständigungsprozess stattfindet. Um eine erfolgreiche Zusammenarbeit sicherstellen zu können, muss sowohl eine gemeinsame Sprache als auch Kriterien zur Bewertung der Qualität von Ergebnissen gefunden werden. [71]

2.1.2. Beteiligte Personen

An den Vorstudien waren viele Studenten unterschiedlicher Fachbereiche beteiligt. Das Kernteam bestand aus André Jeworutzki (Master Informatik), Svenja Keune (Produktdesign Textil) und Larissa Müller (Angewandte Informatik).

2.1.3. Forschungsgebiete

Die Zusammenarbeit von Informatik und Design kann in vielen Forschungsgebieten vorkommen. Eines der großen Forschungsgebiete ist hierbei Wearable Computing. Auch Seamless Interaction, Physical Computing und Human Computer Interaction sind verwandte Bereiche von Interactive Design und werden deshalb erläutert.

Auf weitere Bereiche wie Ubiquitous Computing, Pervasive Computing, Context Awareness soll jedoch in dieser Arbeit nicht näher eingegangen werden, da dies den Rahmen sprengen würde.

Wearable Computing

Wearable Computing ist eines der neuen großen Forschungsgebiete im Bereich der Informatik. Die Einsatzgebiete sind sehr vielfältig.

Definition: *„A wearable computer is a computer, worn and controlled by a user, that is always on and always accessible. That is, the user can always enter and execute commands, even while walking around or doing other activities.*

Unlike other wearable devices (wristwatches, regular eyeglasses, wearable radios, etc.), a WearComp is as reconfigurable as the familiar desktop or mainframe computer. Unlike other computers (including laptops and PDAs), a WearComp is inextricably intertwined with its wearer - WearComp's „always ready“ characteristic leads to a new form of synergy between human and computer.“ [47]

Diese Definition von Steve Mann beschreibt Wearable Computing in Abgrenzung zu anderen mobilen Geräten. Ein Wearable Computer ist während der Anwendung am Nutzer befestigt und mit seinem Träger untrennbar verflochten. Der Träger kann den Computer benutzen, während er anderen Aktivitäten nachgeht.

Die Forschungsbereiche wie Ambient Assisted Living und WearIt@Work sind verwandte Bereiche von Wearable Computing.

Ambient Assisted Living Unter Ambient Assisted Living (AAL) werden Konzepte, Produkte und Dienstleistungen verstanden, die neue Technologien und das soziale Umfeld miteinander verbinden und verbessern mit dem Ziel, die Lebensqualität für Menschen in allen Lebensabschnitten zu erhöhen. Es geht zum Beispiel darum, altersgerechte Assistenzsysteme für ein gesundes und unabhängiges Leben zu entwickeln und ein durch die Umgebung betreutes Wohnen sicherzustellen.

Ambient Assisted Living war unter Anderem ein Innovationsfeld im Rahmenprogramm „Mikrosysteme“ (2004-2009) des Bundesministeriums für Bildung und Forschung. Das Thema war eingebunden in eine übergeordnete Begleitforschung. [26]

Der Vollständigkeit halber wird der Forschungsbereich WearIt@Work erwähnt, der unter anderem Dienstkleidung für Feuerwehrleute entwickelt. Eine Forschung dieser Art ist jedoch schwierig an der HAW umzusetzen, da sowohl bei Ambient Assisted Living als auch bei WearIt@Work eine genaue medizinische Überwachung nötig ist. Die Anforderungen an Sicherheit und Zuverlässigkeit erschweren die Experimente in diesen Bereichen.

Human Computer Interaction

Eine „Mensch-Maschinen-Interaktion“(HCI) ist in vielen Bereichen der Informatik interessant:

„So bezeichnen viele HCI als eine Disziplin, welche sich mit dem Design, der Evaluation und Implementierung von interaktiven Computersystemen für den menschlichen Gebrauch und der Erforschung verwandter Themen beschäftigt. Aus Sicht der Informatik liegt der Fokus auf der Interaktion und speziell auf der Interaktion zwischen einem oder mehreren Menschen mit einem oder mehreren Computern.“ [49]

Das Ziel in der Ambient-Awareness-Ausstellung (s. Kapitel 2.2.2) war laut Kai Rosseburg, dass die Besucher mit den einzelnen Installationen der Ausstellung nahtlos und ohne Handbuch interagieren sollten. [58]

Seamless Interaction

Seamless Interaction ist eine Interaktionstechnik, in der eine intuitive Interaktion zwischen dem Nutzer und der Technik im Vordergrund steht. Als Teil des User-Centered-Interaktionsmodells ist bestenfalls keine Lernphase nötig und eine intuitive Steuerung möglich. [55] Eine einfache Bedienung kann erreicht werden, wenn Fähigkeiten und Techniken genutzt werden die dem Nutzer vertraut sind. Die reale Welt sollte auf die virtuellen Arbeitsabläufe übertragen werden. Das Ziel ist nicht unbedingt, dies „naturgetreu“ zu gestalten,

sondern das beim Benutzer vorhandene mentale Modell mit den Möglichkeiten des Computers zu kombinieren. Dadurch können die Vorteile der natürlichen Interaktion mit der Geschwindigkeit von Computersystemen kombiniert werden. Hierfür ist die dreidimensionale Gestensteuerung, auf Grund ihrer intuitiven Benutzbarkeit, sehr geeignet. [59]

Physical Computing

Physical Computing ist eine Erweiterung des Sense-Act-Modells (s. Kapitel 3.1.1) um physische Objekte. Beim Physical Computing verlässt der Computer die virtuelle Welt und dringt nach außen, wo Computer zum Beispiel Geräte, Maschinen oder Installationen ansteuern, die sich in der realen Welt befinden und dort wirken. Die Verknüpfung des Sense-Act-Modells mit physischen Objekten und die dadurch entstehenden Interaktionsmöglichkeiten sorgen dafür, dass der physische Raum zu einem Ein- und Ausgabemedium für die Informationsverarbeitung wird. So entsteht eine Verbindung zwischen dem medialen- und dem physischen Raum. [31]

2.2. Erste Kurse an der HAW

Der Pentiment Kurs 2008 und die Ambient-Awareness-Ausstellung waren entscheidend für die Entstehung dieser Arbeit. Auf die Beschreibung des Ablaufes dieser Kurse folgt eine kurze Bewertung, um die Entwicklung aufzuzeigen. Die ausführliche Auswertung befindet sich im Kapitel „Evaluation der Vorstudien“ (s. Kapitel 3).

2.2.1. Pentiment 2008

Im Sommer 2008 fand an der HAW ein Wearable-Computing-Kurs während des Pentiment [33] statt. Dieser Kurs wurde von Eyal Sheffer, einem Professor am Shenkar College in Israel, geleitet. [23] Teilnehmer des Kurses waren nicht nur Designer und Techniker von der HAW, sondern auch Designstudenten aus Israel. Die Zielsetzung des Kurses lag darin, in kleinen Gruppen, bestehend aus Designern und mindestens einem Techniker, Objekte zum Thema Wearable Computing (s. Kapitel 2.1.3) zu erstellen.

In den Präsentationen von Eyal Sheffer erhielten die Teilnehmer einen sehr guten Überblick, was momentan diesbezüglich auf dem Markt und in der Forschung vorhanden ist. An den Anfang stellte er seine Definition von Wearable. Dazu zeigte er unter Anderem folgendes Bild:



Abbildung 2.1.: Kein gutes Beispiel für Wearable Computing

Bewertung: Es wurde schnell deutlich, dass innerhalb des Kurses das Design im Vordergrund stehen musste, weil von technischer Seite kaum benutzbare Materialien zur Verfügung standen. Erschwerend kam mangelnde Fachkenntnis hinzu.

Das Konzept war nicht ausgereift genug, um mit der Bandbreite der möglichen Probleme, die ein solcher Kurs mit sich bringt, umzugehen. Teilnehmer waren Studierende aus sehr unterschiedlichen Disziplinen und außerdem aus verschiedenen Ländern. Die Kommunikation war auf zweierlei Arten schwierig: Zum einen die unterschiedlichen Sprachkenntnisse im Englischen und zum anderen die spezifischen Fachtermini. So traten in der Zusammenarbeit immer wieder Verständnisprobleme auf.

Das Konzept beinhaltete viel Design und wenig Technik, sodass die Techniker nicht ausreichend am Konzept beteiligt waren. Die vorhandene Technik war auch dem Kursleiter neu, und er wusste diese nicht zu nutzen. So war es den Teilnehmern auf Grund der Kürze der Zeit nicht möglich, die technischen Möglichkeiten voll auszunutzen. Um ein Beispiel zu nennen: Es waren zwar Mikrokontroller vorhanden, jedoch mussten einige Schwierigkeiten überwunden werden, um diese zu nutzen. Es fehlten Bauteile zur Programmierung der Hardware. Es mussten also zunächst Platinen gelötet werden, um die I/Os des Mikrokontrollers anzusprechen. Auch die Software stand nicht von Anfang an zur Verfügung. In diesem Kurs kam es daher nicht zum erfolgreichen Einsatz von Mikrocontrollern. So waren die Ergebnisse des Kurses in Bezug auf die Technik stark eingeschränkt.

Trotz all dieser Probleme machte der Kurs deutlich wie interessant eine interdisziplinäre Zusammenarbeit sein kann. Daraus hat sich der Schwerpunkt dieser Arbeit entwickelt: Herauszufinden, wie sich verbesserte Bedingungen auf die Ergebnisse auswirken.

2.2.2. Ambient Awareness

Im Wintersemester 08/09 gab es eine Kooperation zwischen dem Department Informatik und dem Department Design. Unter der Leitung von Prof. Dr. Kai von Luck, Prof. Dr. Gunter Klemke, Prof. Franziska Hübler und Jeremy Abbett wurde ein Semester lang eine Ausstellung vorbereitet.

Fragestellung: *„Können wir einen Organismus gestalten, der mit verschiedenen Sinnen des Publikums kommuniziert - so wird der Betrachter zum Gestalter und Teil des Objektes?“* [32]

Dies war der erste Testlauf einer interdisziplinären Zusammenarbeit zwischen den Departments Design und Informatik. Ein Bericht, welcher die entstandene Kunstinstallation vorstellt und den Einsatz und die Probleme der Verwendung der Arduino-Plattform innerhalb des Projektes analysiert, wurde von Sebastian Gregor verfasst. [30] In diesem Projekt wurden erste Tests des Mikrocontrollers Arduino Duemilanove durchgeführt (s. Kapitel 3.1.1). Auch die Master-Projektberichte von Kai Rosseburg [58], Sven Tennstedt, Julia Pressburger [65] und Florian Burka [18] beschäftigen sich mit Teilen der Ausstellung.

Bewertung: Die Ausstellung bot eine große Vielfalt an Möglichkeiten in Hinblick auf Design. Die technische Umsetzung funktionierte nicht bei allen ausgestellten Projekten. Aber durch die Nutzung der Arduino-Plattform konnten bessere Ergebnisse erzielt werden, als innerhalb des Pentiment Kurses (s. Kapitel 2.2.1).

Dazu schreibt Florian Burka in seinem Masterprojektbericht: *„Dass viele Projekte nicht so funktionierten wie geplant, störte keinen der Besucher. Die Besucher waren zumeist von dem was funktionierte begeistert. Dass etwas nicht funktionierte störte höchstens die Schöpfer.“* [18]

2.3. Interdisziplinäre Kurse

Im Vorfeld der zwei interdisziplinären Kurse Toaster-Edwin und Smart-Objects wurde ein Testlauf mit der Schülerpraktikantin Sonia Maas durchgeführt. Auch an dieser Stelle folgt auf die Beschreibung der Kurse und des Experiments jeweils eine kurze Bewertung, um die Entwicklung darzustellen.

2.3.1. Experiment Sonia Maas

Vor Beginn des Toaster-Edwin-Workshops (s. Kapitel 2.3.2) kam eine Schülerpraktikantin ins Team. Mit ihrer Hilfe sollte getestet werden, inwieweit die von Leah Buechley entwickelten LilyPad Arduinos (s. Kapitel 3.1.1) dem Ziel nahe kommen, unerfahrene Anwender durch eine einfache Handhabung an Technik heranzuführen. [16] [42]

Die ursprüngliche Frage, wie lange es dauert, die LilyPads zu vernähen und zu verwenden, konnte nicht beantwortet werden, da sich die Praktikantin mit der Programmierung einer Kiste mit intelligenter Alarmanlage beschäftigte. Diese funktioniert folgendermaßen: Wird eine rote Karte vor den Farbsensor gehalten, macht der Smiley aus LEDs vorne auf der Kiste ein trauriges Gesicht, hält man hingegen eine grüne Karte davor, lächelt der Smiley. Ist der lächelnde Smiley (vgl. Abbildung 2.2) zu sehen, lässt sich die Kiste ohne Probleme öffnen. Ansonsten ertönt ein Alarmsignal.

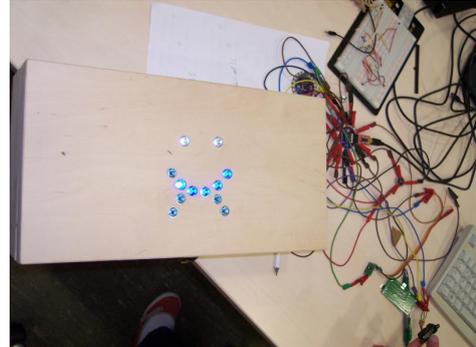


Abbildung 2.2.: Kiste mit intelligenter Alarmanlage

Bewertung: Es konnte bewiesen werden, dass das LilyPad für einen schnellen Einstieg in die Technik geeignet ist.

2.3.2. Pentiment Toaster Edwin - Communicate with Creatures

Rahmenbedingungen

Zeitspanne: Der Kurs fand in den Semesterferien während des Pentiment für eine Dauer von zwei Wochen statt. [33]

Teilnehmer: Die Struktur der Teilnehmer war eine große Herausforderung. Es nahmen Designer unterschiedlicher Departments, Informatiker, Elektrotechniker, eine Lehrerin und eine Professorin am Kurs teil.

Konzept: Unter dem Motto „Communicate with Creatures“ sollte in interdisziplinären Gruppen gearbeitet werden. Die Gruppen suchten sich selbst ein Motto aus. Von uns vorbereitete Kreativaufgaben (s. Anhang A) sollten ihnen dabei helfen, ihre Idee zu entwickeln. Durch die Aufgaben wurden sie an die technischen Möglichkeiten herangeführt. Wichtig für die Zusammensetzung der Gruppen war, dass immer ein Techniker Mitglied sein musste, um die Umsetzung zu ermöglichen. Das Konzept sah aber auch eine spielerische Heranführung an die anderen beteiligten Disziplinen vor. Jeder sollte aus dem ihm fremden Bereich möglichst viel lernen. Die Abbildung 2.3 zeigt das Poster des Kurses. Im Anhang befindet sich der Presstext zum Workshop (s. Anhang A).

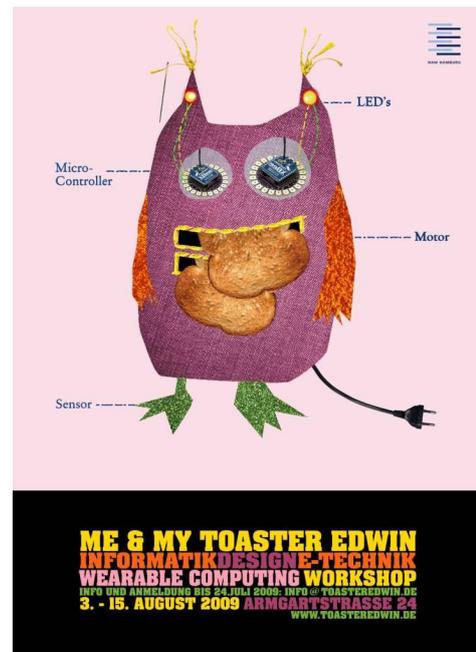


Abbildung 2.3.: Poster vom Toaster-Edwin-Workshop

Fokus Im Fokus standen hier sowohl die Technik als auch das Design.

Räumliche Ausstattung: Der Kurs fand in der Armgartstraße statt. Dort wurde uns ein Computerraum mit acht Rechnerarbeitsplätzen und ein Computerraum mit großem Arbeitstisch in der Mitte zur Verfügung gestellt (s. Abbildung 2.4).



Abbildung 2.4.: Arbeitsraum während des Toaster-Edwin-Workshops

Ablauf

Planung Die Struktur des Kurses war vollständig auf die interdisziplinäre Zusammenarbeit ausgelegt. Die Teilnehmer sollten in kleinen Gruppen zusammenarbeiten und möglichst viel voneinander profitieren. Es gab Übungen zu jedem beteiligten Bereich, so dass Informatiker nähen und Designer programmieren mussten.

Die Idee der Programmierübungen war, dass die Designer am Rechner sitzen und programmieren mussten, während die Informatiker daneben saßen und in einer Art Pair Programming (s. Kapitel 4.4.2) Hilfestellung gaben. Die Designer erhielten so in relativ kurzer Zeit einen guten Einblick in die Programmierung. Durch die Zusammenarbeit bekamen auch die Techniker einen Eindruck davon, wie Designer vorgehen. Der Ablaufplan sowie der Presstext befinden sich im Anhang A.

Die Präsentationen zu Beginn des Tages deckten die unterschiedlichen beteiligten Bereiche ab. Im Anschluss gab es in der ersten Woche Kreativ- und Programmieraufgaben (s. Anhang A). Die Aufgaben beschränkten sich nicht nur auf LilyPad Arduinos (s. Kapitel 3.1.1), sondern es wurden auch Arduino Duemilino (s. Kapitel 3.1.1) verwendet. Die Technik stand den Teilnehmern zur Verfügung, während die Materialien selbst mitgebracht werden mussten. Dieses Vorgehen ist im Department Design üblich.

Zusätzlich begleiteten Medientechniker den Kurs filmisch, wie auf Abbildung 2.5 zu sehen.



Abbildung 2.5.: Das Kamerateam bei der Arbeit

Verlauf: Die Kreativaufgaben verliefen weit besser als geplant. Zur Verfügung standen den Teilnehmern vorprogrammierte LilyPads, welche sie für ihre Objekte verwenden konnten. Sie nutzten diese aber nicht in der vorgesehenen programmierten Weise, sondern erzeugten auch unerwartete Effekte. Die Arbeitsweise der Designer zu beobachten, war sehr aufschlussreich. Die Teilnehmer experimentierten mit den vorgegebenen Teilen in unterschiedlichster Weise und machten diese Kreativaufgabe zu einem echten Erlebnis (s. Kapitel [3.4.2](#)).

Die Gruppen hatten nach den Präsentationen am Morgen die Möglichkeit, ihre Zeit frei einzuteilen. Trotzdem musste an den letzten Abenden bis spät in die Nacht gearbeitet werden, um die Produkte fertigzustellen.

Es entstand auch eine Webseite. [[36](#)]

Bewertung

Bereits in der ersten Kreativaufgabe kamen sehr ausgefallene Ergebnisse zustande, wie das Bild [2.6b](#) zeigt. Der Kurs erwies sich am Ende als Erfolg. Abschließend wurden die Produkte in einer Ausstellung präsentiert und die Besucher der Ausstellung hatten ein reges Interesse daran, dass der Kurs auch zukünftig angeboten wird. Die Abbildung [2.6a](#) zeigt ein im Kurs entstandenes Objekt.

Des Weiteren konnten wir beweisen, dass eine mobile Variante des Kurses durchführbar ist. Alle Materialien wurden in Form von Classroomkits (s. Kapitel [3.1.2](#)) in die Armgartstraße transportiert.



(a) Die Wolke



(b) Der Schwede

Abbildung 2.6.: Die Wolke und der Schwede

Auf die Frage innerhalb der Evaluationsbögen, ob die Teilnehmer sich vorstellen können, weiter in dem Bereich zu arbeiten, antworteten alle positiv.

2.3.3. Smart Objects

Der Smart-Objects-Kurs wurde geleitet von André Jeworutzki und Prof. Franziska Hübler. Teilnehmer waren acht Studierende aus dem Department Design. Diese Konstellation bedeutete einen immensen Betreuungsaufwand. Die Teilnehmer mussten innerhalb kürzester Zeit lernen sowohl zu Löten als auch zu Programmieren, um ihre Ziele zu erreichen. Trotz der erfolgreichen Vermittlung dieser Kenntnisse musste jedes Projekt zusätzlich von André Jeworutzki unterstützt werden, welcher die technischen Probleme erfolgreich löste. [34] Die Abbildung 2.7b zeigt den Flyer des Kurses.

Bewertung

Ein sehr erfolgreicher Kurs mit äußerst kreativen Ergebnissen. Die Abbildungen 2.7a und 2.8 zeigen zwei Objekte. 2.6b zeigt.



(a) Ergebnis Smart Objects



(b) Flyer Romantische Maschinen

Abbildung 2.7.: Ausstellung Romantische Maschinen



Abbildung 2.8.: Vorhang aus LEDs

2.4. Schülerkurse von kurzer Dauer

An der HAW wird regelmäßig der Versuch unternommen, Schüler zu werben. Der Schulcampus gibt Schülerinnen und Schülern die Gelegenheit, Informationen über die unterschiedlichsten Studiengänge zu sammeln und zu erfahren, wie ein praxisorientiertes Studium abläuft. [29] Im Rahmen solcher Ferienprogramme konnten die eigens dafür entwickelten Konzepte an Schülergruppen getestet werden. Die räumliche Ausstattung dieser Kurse wird in 3.1.2 beschrieben.

2.4.1. Konzept: „Snakes on a Plane“

Das Konzept basiert auf dem Film „Snakes on a Plane“. [37] Die Aufgabe bestand darin, vorgenähte Schlangen mit Technik zu füllen und hinterher mit diesen für ein Fotoshooting zu posieren. Die Kreativität beschränkte sich hierbei darauf, die Technik auszusuchen und in der Schlange zu platzieren, sodass diese möglichst lebendig aussieht. Dieses Konzept ist gut geeignet für kurze Kurse, in denen das Programmieren im Vordergrund steht. Der Zeitaufwand für das Ausstatten der Schlangen ist nicht sehr hoch. Durch Filmausschnitte ist den Teilnehmern die Aufgabe leicht zugänglich. Das Fotoshooting sorgt dafür, dass ein Andenken bleibt, wenn die Schlangen hinterher wieder auseinandergebaut werden müssen.

2.4.2. Fokus

Der Fokus innerhalb der Kurse lag auf der Programmierung. Es wurde ein Einblick in die Möglichkeiten der Informatik geboten, und das Design war der Schlüssel, der die Informatik interessanter gestaltete. Interactive Design ermöglichte so eine spielerische Umsetzung.

2.4.3. Herbsthochschule für Mädchen Teil 1

Rahmenbedingungen

Zeitspanne: Dieser Kurs erstreckte sich über einen halben Tag.

Teilnehmer: An der Herbsthochschule nahmen 12 Schüler teil, die in Zweiergruppen aufgeteilt wurden.

Technik: Es wurden die Classroomkits (s. 3.1.2) verwendet.

Ablauf

Planung: In der zehnminütigen Anfangspräsentation ging es im Wesentlichen darum, das Team, das Konzept und die vorhandene Technik vorzustellen. Daraufhin wurden vorprogrammierte Lily-Pads ausgeteilt. Aufgrund der Kürze der Zeit sollten die Schüler diese lediglich mit Sensoren und Aktoren verbinden und ihre Kreativität nutzen, um die Schlangen zu füllen. Geplant war dieser Kurs unter dem Konzept „Herbsthochschule für Mädchen“. [29] Die Mädchen sollten die Fotos und einen Eindruck von Informatik mitnehmen.



Abbildung 2.9.: Herbsthochschule

Verlauf: Trotz der sehr kurzen Einführung in die Software haben die Teilnehmer unaufgefordert begonnen, die Software zu benutzen. Sie haben eigenständig den Code verändert, und es war schnell klar, dass sie selbstständig programmieren wollten. Da drei erfahrene Kursleiter zur Verfügung standen, konnte diesem Wunsch entsprochen werden, und es wurde im Einzelnen erklärt, wie man die LilyPads programmiert.

Bewertung

Wie in der Planung bereits vermutet wurde, reichte die Zeit für die vollständige Umsetzung des Kreativteils und die Programmierung nicht aus. So konnten leider nicht alle Gruppen an dem Fotoshooting teilnehmen. Wir konnten aber ihrem Wunsch gerecht werden, etwas über das Programmieren im Allgemeinen zu lernen.

2.4.4. Herbsthochschule Teil 2

Rahmenbedingungen

Zeitspanne: Dieser Kurs erstreckte sich über 1,5 Tage.

Teilnehmer: An der Herbsthochschule nahmen 14 Schüler teil, die in Zweiergruppen aufgeteilt wurden.

Technik: Es wurden die Classroomkits (s. Kapitel 3.1.2) verwendet.

Ablauf

Planung: Für beide Herbsthochschulkurse war der geplante Ablauf ähnlich. Die Anfangspräsentation war auf 20 Minuten angelegt. Die Einführung konnte in diesem Kurs ausführlicher stattfinden, da ausreichend Zeit zum Programmieren zur Verfügung stand.

Verlauf: Der Ablauf verlief ähnlich wie in der Planung. Es standen ebenfalls drei Betreuer für die Programmierung und zwei für die kreative Umsetzung zur Verfügung. Deshalb konnten die Schlangen fertiggestellt werden.

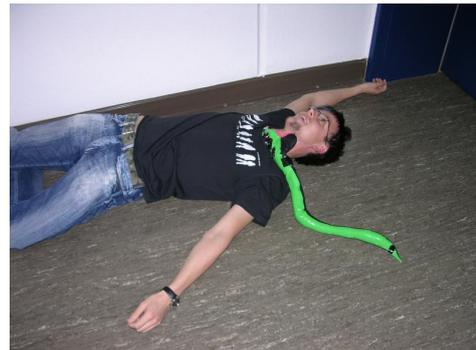


Abbildung 2.10.: Snakes-on-a-Plane-Fotoshooting

Bewertung

Es waren lediglich pädagogische Probleme, die den Ablauf erschwerten. Eine neue Herausforderung bestand zum Beispiel darin, die unterschiedlichen Altersstufen zu kombinieren. Besonders die 14-jährigen Teilnehmer zu motivieren, stellte sich als großes Hindernis dar. Zum Schluss konnten alle Gruppen am Fotoshooting teilnehmen, und bei der Abschlusspräsentation waren alle mit Freude dabei.



Abbildung 2.11.: Abschluss-Fotoshooting Herbsthochschule

2.4.5. Faszination Games 1

Rahmenbedingungen

Zeitspanne: Der Kurs erstreckte sich über eine Dauer von zwei Stunden.

Teilnehmer: An diesem Kurs nahmen zehn Schüler teil, die in Zweiergruppen aufgeteilt wurden.

Technik: Vorgelötete Hardware bestehend aus: 1 LilyPad, 1 Charger, 1 Vibrationsmotor, 1 LED, 1 Akku, 1 RGB LED, 1 Lichtsensor und 1 Grußkarte.

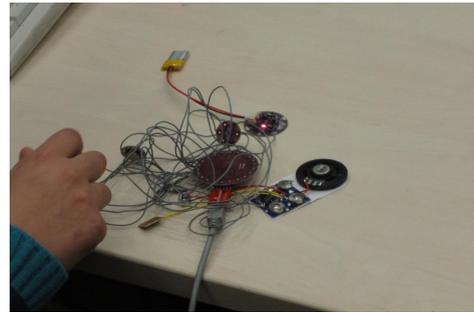


Abbildung 2.12.: Vorgelötete Hardware Faszination Games

Ablauf

Planung: In der 20igminütigen Anfangspräsentation ging es nicht nur darum, das Team, das Konzept und die vorhandene Technik vorzustellen. Der Fokus lag diesmal darin, dass die Schüler einen Einblick in Physical Computing bekommen (s. Kapitel 2.1.3).

Deshalb wurden ihnen Auszüge aus eigenen Projekten gezeigt, um zu demonstrieren, wie Informatik auch gestaltet werden kann. In diesem Kurs wurde die Elektrotechnik bewusst ausgespart und vorgelötete Hardware verteilt. Die Schüler bekamen die Aufgabe über Leah Buchleys Tutorials [15] zu versuchen, ihre Ideen umzusetzen. Sie sollten den Code von Leah Buchley (s. Kapitel 3.1.1) kopieren und mit kleineren Erklärungen modifizieren. Für Gruppen, die nicht rechtzeitig fertig wurden, wurde ein Mustercode vorbereitet, der im Notfall ausgeteilt werden sollte. Geplant war es, nach 90 Minuten mit dem Ausstopfen der Schlangen zu beginnen.

Verlauf: Die Tutorials reichten nicht aus, um die LilyPads zu programmieren. Nach 80 Minuten wurde der Mustercode an alle Gruppen ausgegeben, um diesen zu modifizieren. Die Teilnehmer sollten zum Beispiel die Farben der RGB LED verändern. Die Schüler wurden dadurch alle rechtzeitig fertig.



Abbildung 2.13.: Faszination Games 1

Bewertung:

Die Problematik dieses Kurses lag ebenfalls in der geringen Motivation der Schüler. Sie wurden leicht durch technische Randerscheinungen abgelenkt, was dadurch noch verstärkt wurde, dass einige mit den gestellten Aufgaben über- und andere unterfordert waren.

2.4.6. Faszination Games 2

Rahmenbedingungen

Zeitspanne: Der Kurs erstreckte sich über eine Dauer von zwei Stunden.

Teilnehmer: An diesem Kurs nahmen neun Schüler teil, die in Zweiergruppen aufgeteilt wurden.

Technik: Vorgelötete Hardware bestehend aus: 1 LilyPad, 1 Charger, 1 Vibrationsmotor, 1 LED, 1 Akku, 1 RGB LED, 1 Lichtsensor und 1 Grußkarte.

Ablauf

Planung: Die Einführung und Anfangspräsentation fand wie in Faszination Games 1 (s. Kapitel 2.4.5) beschrieben statt. Die Erfahrungen, dass die Leah Buchley's Tutorials [15] zeitlich nicht ausreichend sind, führten dazu, dass die Kursleiter den Code mitprogrammierten. Die Schüler bekamen so über den Beamer eine Kurzanleitung und mussten lediglich Modifikationen vornehmen. Gezeigt wurde, wie LEDs angeschaltet und Sensordaten eingelesen werden. Dies sollten die Teilnehmer kombinieren.

Verlauf: Die Schüler konnten mit Anleitung ihre Schlangen programmieren.

Bewertung

Diesmal waren die Schüler aufmerksamer und arbeiteten ohne Ablenkung.



Abbildung 2.14.: Faszination Games 2

2.5. Einsemestrige Schülerkurse

Durch die Teilnahme von Xenia Rendtel, Informatiklehrerin am Margarethe-Rothe Gymnasium [56], am Toaster-Edwin-Workshop (s. Kapitel 2.3.2), bot sich die Gelegenheit die neu konzipierten Kurskonzepte in längeren Kursen zu testen. Lehrerin und Schulleitung erhofften sich durch die Vermischung von Informatik und Design, das Interesse für Informatik bei den Schülern zu stärken. [34] Aus dem Erfolg des ersten Kurses in einer achten Klasse ergab sich im darauffolgenden Semester die Chance, dieses Konzept auch in einer 13. Klasse zu testen.

„Die komplexen Problemstellungen der Informatik ermöglichen arbeitsteiliges Vorgehen. Die Schülerinnen und Schüler erlernen soziale Fähigkeiten, wie Kommunikation und Kooperationsfähigkeit. Sie lernen, dass diese Kompetenzen wesentlich sind, um Aufgaben und Projekte erfolgreich zu bewältigen. Außerdem werden wichtige Aspekte und Denkansätze der Projektarbeit, wie Planung, Entscheidungen treffen, Präzision, Aufteilung von Aufgaben und Einhaltung von Absprachen, vermittelt.“ [57]

Dieses Zitat von Kai Rosseburg beschreibt sehr treffend die Vorteile der Informatik innerhalb des Schulunterrichts. In seiner Bachelorarbeit geht es zwar nicht um Interactive Design, sondern um die Entwicklung einer Programmierumgebung für roboterbasierten Informatikunterricht an Schulen, die verfolgten Ziele sind aber sehr ähnlich.

2.5.1. Schülerkurs Teil 1

Rahmenbedingungen

Teilnehmer: An diesem Kurs nahm eine achte Klasse mit 12 Schülern teil, die in Zweiergruppen aufgeteilt wurden.

Zeitspanne: Der Kurs erstreckte sich über sechs Termine mit einer Dauer von je 90 Minuten.

Konzept: Es wurden drei verschiedene, zu entwickelnde Objekte vorgegeben. In diesem Fall waren es Mikrofon, Sonnenbrille und Kopfhörer.

Zusätzlich wurden per Losverfahren zwei dazugehörige Adjektive ermittelt. Beispiele sind „das lustig tanzende Mikrofon“ oder „der ängstliche, pickelige Kopfhörer“. Diese zufällige Kombination aus 2 Adjektiven und einem Nomen sollte als kreativer Anreiz genommen werden, eine Geschichte zu dem zu entwickelnden Objekt zu erfinden. Der Lehrplan der achten Klasse sieht unter anderem vor, die Benutzung von Word zu vermitteln. Die Schüler sollten sich deshalb zu ihrem Objekt eine Geschichte ausdenken, diese aufschreiben und am Ende vortragen. Die komplette Dokumentation beinhaltete auch Bilder und wurde am Ende in der Notengebung berücksichtigt.

Für das basteln der Objekte standen Muster sowie vorgefertigte Schnitte zur Verfügung. Die Muster sollten zeigen, was am Ende ein mögliches Ergebnis wäre. Die Abbildungen [2.15a](#) und [2.15b](#) zeigen zwei Objekte. Die Schnitte dienten der Vereinfachung des Kreativteils, denn sie mussten lediglich ausgeschnitten und mit entsprechender Technik gefüllt werden.

Ablauf

Planung: Die Schüler sollten in dem Kurs auf spielerische Art und Weise Programmierkonzepte lernen. Für die Benutzung der diversen Sensoren und Aktoren waren Kenntnisse über Schleifen, Variablendeklaration und „if/else“-Strukturen nötig.



(a) Mikrofon



(b) Brille

Abbildung 2.15.: Schülerkurs Teil 1

Es gab zu Beginn jeder Stunde eine kurze Einführung in das jeweilige Thema, und anschließend mussten die Aufgaben selbstständig gelöst werden. Um die Anwesenheit der Schüler vollständig für die Programmierung nutzen zu können, musste das Design als Hausaufgabe umgesetzt werden. Am letzten Termin war eine Abschlusspräsentation geplant, bei der eine Jury das „beste Objekt“ wählen sollte.

Verlauf: Die Kursleitung sah sich mit einer schwierigen Altersgruppe konfrontiert, die die Arbeit nicht immer leicht machte. Die Arbeitsmoral bei der Erstellung der Hausaufgaben war sehr schlecht, deshalb musste, ein Extratermin angeboten werden, um die Ergebnisse rechtzeitig zur Abschlusspräsentation fertigstellen zu können.

Bewertung

Der Extratermin sorgte dafür, dass am Ende doch alle fertig waren und tolle Ergebnisse präsentieren konnten. Um zukünftig solch einen Kurs erfolgreich anbieten zu können, ist es sinnvoll, einen zusätzlichen Termin als Puffer einzuplanen. Die Schüler konnten am Ende sogar ihrer Schulleiterin ihren selbst geschriebenen Quellcode ausführlich erklären.

2.5.2. Schülerkurs Teil 2

Rahmenbedingungen

Zeitspanne: Der Kurs erstreckte sich über sieben Termine mit einer Dauer von je 90 Minuten. Der zusätzliche Termin wurde an dieser Stelle berücksichtigt.

Teilnehmer: An diesem Kurs nahm eine 13. Klasse mit 12 Schülern teil, die in Zweiergruppen aufgeteilt wurden.

Konzept: Die Aufgabe bestand darin, Gegenstände aus dem Alltag durch Interaktivität interessanter zu gestalten. Die zu entwickelnden Objekte sollten miteinander kommunizieren können und so interessante Ergebnisketten auslösen. *„Bei Schwärmen ergibt sich ihr komplexes Verhalten aus der Emergenz Einzelner. Bei agentenorientierten Objekten sind bereits Einzelne zu komplexen Verhalten fähig. Durch die Funktechnologie wären beispielsweise Schwarmalgorithmen oder intelligenter Staub denkbar: Durch eine zielbewusste Verhaltensweise interagieren viele einzelne Objekte mit den Anwendern und umgekehrt.“* [34]

Ablauf

Planung: Es konnte davon ausgegangen werden, dass bereits Grundlagen der Programmierung bei den Schülern vorhanden waren. Deshalb sollte direkt mit der Lösung der Aufgaben begonnen werden. Die Ideenfindung und die Umsetzung des Kreativteils wurden als Hausaufgabe gestellt.

Verlauf: Aufgrund der schlechten Arbeitsmoral in Bezug auf die Erstellung der Hausaufgaben und der noch nicht ausgereiften Programmierkenntnisse musste erneut ein Extratermin angeboten werden.

Bewertung

Eine komplexe Interaktion der verschiedenen Objekte war nicht möglich, weil insgesamt zu viel Zeit für die Vermittlung von Grundlagenwissen verwendet werden musste. Trotz anderer Erwartungen waren die Programmierkenntnisse nicht deutlich von denen der achten Klasse zu unterscheiden. Am Ende entstanden trotzdem kreative Objekte, wie die Abbildungen [2.16a](#), [2.16b](#) und [2.17a](#) zeigen.

2.5.3. Fazit aus den längeren Schülerkursen

Eine Umsetzung in dieser Form ist weder in einer achten Klasse noch in einer 13. Klasse angebracht. Durch die Anreise, den daraus resultierenden Auf- und Abbau der Hardware sowie einer schlechten Hausaufgabenmoral geht sehr viel Zeit verloren. Eine mögliche Umsetzung solcher Kurse ist nur in Form von Projekttagen bzw. Workshops sinnvoll.



(a)



(b)

Abbildung 2.16.: Präsentation am Margaretha-Rothe-Gymnasium



(a)



(b)

Abbildung 2.17.: Präsentation am Margaretha-Rothe-Gymnasium

2.6. Studentische Projekte

Erfolgreicher als die Schülerkurse war die interdisziplinäre Zusammenarbeit in studentischen Projekten. Zum Beispiel ging es in den Projekten mit Helene All und Ekaterina Iffraimova darum, Kleidung anzufertigen, die interaktiv auf ihre Umwelt reagiert. Die vielen Anforderungen an tragbare Kleidung (s. Kapitel 3.2.1) sorgten dafür, dass oftmals Zugeständnisse an Technik und Design gemacht werden mussten. Häufig kam es dazu, dass Kabel neu gelötet oder Schnitte passend gemacht werden mussten. Während Ekaterina sich auf Licht und Schatten konzentrierte, versuchte Helene über Motoren Bewegung umzusetzen. Beide nutzten die Technik, um ein ausgefallenes Design zu erreichen. Weitere Projekte, bei denen es allerdings nicht um Mode ging, entstanden in Zusammenarbeit mit Svenja Keune, welche versucht, über Interaktion und ausgefallene Materialien zu überzeugen. Der Projektbericht von André Jeworutzki beschäftigt sich mit Teilen dieser Arbeiten. [34] Im weiteren Verlauf werden die einzelnen Projekte näher beschrieben.

2.6.1. Ekaterina Iffraimova

Beschreibung

Die Zusammenarbeit mit Ekaterina Iffraimova begann innerhalb des ersten Pentiment-Kurses (s. Kapitel 2.2.1). Sie studiert Modedesign an der HAW. Schon während des Kurses entstand die Neugierde daran auszuprobieren, was in einer derartigen Kooperation erreicht werden kann, wenn unter verbesserten Bedingungen gearbeitet wird. Der Plan war, eine Kollektion zu entwickeln. Diese sollte den Standort Hamburg beinhalten. Geplant war den Standort der HAW optisch in das Design einzufügen. Es sollten die Hochhäuser am Berliner Tor dargestellt werden. Die Fenster sollten LEDs ent-

halten und je nach Tageszeit leuchten. Die Abbildung 2.18 zeigt ein fertiges Objekt. Erste Prototypen wurden auf der Basis von LilyPads entwickelt (s. Kapitel 3.1.1). Diese erwiesen sich jedoch nicht als praktikabel, denn die Einfachheit der LilyPads bringt zu viele Einschränkungen mit sich (s. Kapitel 3.5.3). So konnte zum Beispiel der Wunsch, 1000 LEDs anzusteuern, von technischer Seite nicht umgesetzt werden. Der erste Prototyp, welcher auf der Ausstellung von Silpion [54] gezeigt werden sollte, beinhaltete sowohl LEDs als auch Kompressoren. Leider ist es bei den LilyPads nur möglich, eine geringe Anzahl von LEDs anzusteuern. Hierfür wurden die Möglichkeiten des LilyPads zur Kommunikation genutzt. Die Kompressoren wurden aus Blutdruckmessgeräten ausgebaut. Durch die Nutzung von Transistoren war es so möglich Plastiktüten mit Luft zu füllen, die in dem Overall eingearbeitet waren.



Abbildung 2.18.: Kollektion Zellteilung einer Stadt

Bewertung



Abbildung 2.19.: Kollektion Zellteilung einer Stadt

Die Integration der Hardware führte oftmals zu Diskussionen, und am Ende konnte keine Einigung gefunden werden. An dieser Stelle wurde das erste Mal deutlich, wie schwierig eine interdisziplinäre Zusammenarbeit zwischen Informatikern und Designern sein kann.

Die später beschriebenen Kommunikationsschwierigkeiten traten hier erstmalig auf (s. Kapitel 3.4.1). Während der Techniker die Idee nicht für umsetzbar hielt, fühlte der Designer sich mit seiner Idee nicht verstanden. Es wurde verlangt, dass die fertige Hardware sich exakt in das Design einarbeiten lässt. Zum Beispiel wurden die Stromkabel zunehmend als störend empfunden. Alternativen waren allerdings sehr kostenaufwändig und erschienen häufig auch aus der Perspektive der Wirtschaftlichkeit nicht sinnvoll, denn ein immens hoher Aufwand für die Herstellung steht für den Informatiker in keinem Verhältnis zu dem Produkt oder dem Lerneffekt. Hier wäre etwas mehr Anpassungsfähigkeit an die Technik von Seiten der Designer wünschenswert gewesen.

Trotz aller Schwierigkeiten entstand am Ende eine innovative Kollektion wie die Abbildungen 2.19, 2.20a, 2.20b, 2.21a und 2.21b zeigen.

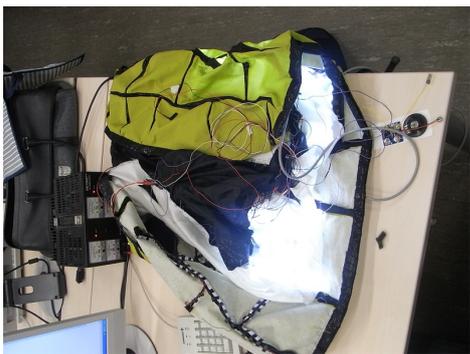


(a)



(b)

Abbildung 2.20.: Kollektion Zellteilung einer Stadt



(a)



(b)

Abbildung 2.21.: Kollektion Zellteilung einer Stadt



(a) Hochfahrbare Kuppel, dahinter drehbare Kugeln



(b) Drehbare Servomotoren mit daran befestigten Kugeln

Abbildung 2.22.: Kollektion Helene All

2.6.2. Helene All

Beschreibung

Auch Helene All studiert Modedesign an der HAW und war Teilnehmerin am ersten Pen-timent Kurs (s. Kapitel 2.2.1). Helene hat für ihre Diplomarbeit ein Konzept entwickelt, das Technik beinhaltet. Die von ihr entwickelten Teile beinhalten viel Bewegung. Motoren werden dazu genutzt, „das Innere sichtbar zu machen“. Der Verschaulichung dienen die Abbildungen 2.22a und 2.22b. Ein weiteres Objekt entstand durch die Idee, den Herzschlag eines Menschen für andere sichtbar zu machen. Hierfür wurde ein handelsübliches Messgerät verwendet. Die Daten von dem Messgerät wurden per Funk an einen LilyPad übertragen, so dass die LEDs dem Herzschlag nachempfunden pulsierten (s. Abbildung 2.23).

Bewertung

Die Vorstudie kann zu diesem Zeitpunkt der Arbeit nicht bewertet werden, da die Zusammenarbeit noch nicht beendet ist.



Abbildung 2.23.: Herzschlag wird über XBee an LEDs übertragen

2.6.3. Svenja Keune

Die Zusammenarbeit mit Svenja Keune erstreckt sich bis zum Ende dieser Arbeit und soll im Master fortgesetzt werden. Diese Zusammenarbeit hat bislang die meisten Erfolge gebracht. Im Folgenden wird ihre Bachelorarbeit beschrieben, die in enger Zusammenarbeit mit dem Department Informatik entstanden ist. Zu Beginn werden ihre Vorstudien vorgestellt, weil diese Interactive Design in einer sehr kreativen Form umsetzen. Die Herausforderung einer Bachelorarbeit im Department Design liegt darin, dass nicht nur ein schriftlicher Teil ausgearbeitet, sondern zusätzlich ein praktisches Modell angefertigt werden muss.



(a)



(b)

Abbildung 2.25.: Vorstudien Svenja Keune

Vorstudien

Als Vorstudien wurden unter Anderem Objekte Korallen nachempfunden aus Porzellankanülen gebrannt. Diese wurden mit Textilien, wie zum Beispiel thermochromem Stoff, kombiniert. Stoff dieser Art verfärbt sich unter Wärmeeinfluss. Ein anderes Objekt wurde mit Nitinoldraht (auch geläufig als Gedächtnisdraht) ausgestattet. Durch Zuführung von Hitze kehrt dieser selbstständig in eine gespeicherte Form zurück. Die Speicherung einer Form wurde mit Hilfe des Departments Maschinenbau erreicht. In diesen Laboren konnte der Nitinoldraht 15 Minuten bei 600 Grad erhitzt werden. Das nächste Objekt wurde mit LEDs ausgestattet. So entstanden drei Objekte mit völlig verschiedenen Interaktionsmöglichkeiten. Allein an dieser Vorstudie wird noch einmal deutlich, wie breit sich eine Zusammenarbeit von Informatik und Design erstrecken kann. Die Abbildungen [2.24](#), [2.26](#), [2.25a](#) und [2.25b](#) zeigen die entstandenen Objekte.



Abbildung 2.24.: Thermochromatischer Stoff und sich selbst bewegendes Nitinolkorallenobjekt

Bachelorarbeit Svenja Keune

„Diese Bachelorarbeit behandelt ein Projekt, in dem persönliche Eindrücke aus Design und Technik vereint werden, um eine weitere Bewusstseinssebene zu schaffen. Das Zusammenspiel und die Möglichkeiten der verschiedenen Disziplinen dienen der Wirkung der Arbeiten vor allem im Bezug auf den Betrachter, der durch die verwendeten Mittel über vielerlei Wege erreichbar ist.“ [40]



Abbildung 2.26.: Porzellanobjekt

Es handelt sich bei dieser Bachelorarbeit um eine Installation textiler Objekte. Diese stellen einen Verbund von Organismen dar, die miteinander in Verbindung stehen, aufeinander reagieren und miteinander kommunizieren. [40]

Die Teile der Installation bestehen aus verschiedenen Materialien, die durch Mechanik, Elektronik und Projektion interaktiv werden. Zu Beginn der Arbeit standen Skizzen und eine umfassende Recherche über Oberflächen und deren Beschaffenheit und Eigenschaften in der Natur. Es wurden Prototypen aus Salzteig angefertigt, um verschiedene Größen, Formen und Anordnungen unter realistischen Bedingungen zu visualisieren.

Für die Bachelorarbeit war die Auseinandersetzung mit textilen Oberflächen, aber auch eine konzeptionelle Überarbeitung der Interaktionsmöglichkeiten in Bezug auf Erkennbarkeit und Bedienbarkeit geplant. [40] An dieser Stelle sollten also die Aspekte von Seamless Interaction berücksichtigt werden.

Da es zu weit gehen würde, alle 15 Objekte an dieser Stelle genauer zu erläutern konzentriert sich die vorliegende Bachelorarbeit auf die wesentlichsten Teile. Zu diesen gehört zum Beispiel ein Lüfter, der verwendet wird, um ein Objekt mit Luft zu füllen oder ein Pustesensor, der bei Auslösung für eine wellenförmige Lichtbewegung sorgt. Des Weiteren ist es möglich, dass einige Objekte sich über Distanssensoren gesteuert im Morsealphabet unterhalten können. Darauf wird im späteren Verlauf der Arbeit näher eingegangen (s. Kapitel 4.5) Die Beispiele beweisen, dass die Interaktion von Objekten, die durch kreative Kombination von Informatik und Design erzielt wird, im Fokus dieser Bachelorarbeit steht.

2.6.4. Bewertung der studentischen Projekte

Das studentische Projekt wurde in mehrere kleine Teilprojekte aufgeteilt, so dass eine Umsetzung möglich wurde. Trotz zahlreicher Schwierigkeiten (s. Kapitel 3) wurde die Zusammenarbeit zwischen Informatik und Design nicht abgebrochen und soll im Master weiter ausgebaut werden.

3. Evaluation der Vorstudien

In diesem Kapitel werden die Vorstudien evaluiert. Hierzu sind Grundlagen in Bezug auf Technik und Forschungsgebiete nötig. Im Anschluss an diese werden die technischen und menschlichen Probleme dargestellt und erläutert, die sich in der Zusammenarbeit ergeben haben.

3.1. Grundlagen

3.1.1. Beschreibung von Interactive Design anhand des Sense-Act-Models

Interactive Design besteht in der Regel aus drei Teilen: Sense, Plan und Act. Ein solches Sense-Act-Modell ist aus der Robotik bekannt. Alle entstandenen Objekte sind mit diesem Modell zu beschreiben. Im Folgenden wird dieses Modell und eine mögliche Laborausstattung genauer erläutert. Außerdem wird in diesem Kapitel näher darauf eingegangen, welche Materialien in den Vorstudien in Bezug auf dieses Konzept verwendet wurden. Hierfür werden einige Sensoren und Aktoren vorgestellt und die verwendeten Microkontroller erläutert.

Die Metapher des Sense-Act-Modells besteht im Wesentlichen darin, dass Sensordaten eingelesen und verarbeitet werden und daraufhin eine Reaktion in Form eines Aktors ausgelöst wird. Ein Beispiel ist ein Microkontroller, der eine LED anschaltet, sobald ein Lichtsensor Werte liefert, die auf Dunkelheit hinweisen.

Die Aufgabe für das Design bestand innerhalb der interdisziplinären Vorstudien darin, die zwei Teile Sense und Act kreativ zu kombinieren, während die Informatik meist den Plan-Teil koordinierte. Des Weiteren sorgte der Informatiker häufig dafür, dass die Verkabelung, also der elektrotechnische Anteil, korrekt umgesetzt wurde. Ausnahmen waren die Kurse, in welchen wie in Abschnitt [2.3.2](#) beschrieben, die Technik von Nicht-Technikern umgesetzt wurde.

Für die Informatik bedeutete dies: Es werden Sensordaten eingelesen und ausgewertet. Daraufhin wird eine Reaktion ausgelöst, also ein Aktor betätigt. Gesteuert wird das Ganze von einem Microkontroller.

Sensoren

Es gibt sehr viele verschiedene Sensoren. In der Regel wurden in den Vorstudien Abstands- und Lichtsensoren verwendet, da deren Arbeitsweise leicht zugänglich ist. In den Kursen hatte dies den Vorteil, dass ein erfolgreicher Umgang mit diesen Sensoren leicht zu erlernen war. Selbst in den kurzen Kursen (s. Kapitel 2.4.3) wurde mindestens einer der beiden verwendet. Als Abstandssensor stand ein Infrarotsensor von Sharp [25] zur Verfügung, der bis zu einer maximalen Reichweite von 30 cm über Infrarot den Abstand misst. Diese Distanz variiert ja nach Sensortyp.

Als Lichtsensor diente der Helligkeitsmesse [61] von LilyPad (s. Kapitel 3.1.1). Ein Vorteil für die Nutzung dieses Sensors in den Kursen besteht darin, dass er Teil des Online-Tutorials [15] von Leah Buechley ist.

Auch eine Kamera kann als Sensor eingesetzt werden. An der HAW sind diesbezüglich viele Projekte in Arbeit. Erwähnenswert ist die 3D Kamera, die derzeit von Arne Bernin in Bezug auf Gestenerkennung getestet wird. Aber auch herkömmliche Kameras lassen sich bei Verwendung entsprechender Algorithmen als Sensor nutzen. Andere Möglichkeiten bieten Mikrofone (s. Kapitel 2.6.3), Herzschlagsensoren (s. Kapitel 2.6.2), Temperatursensor, Gyroskope etc. Eine nähere Betrachtung der einzelnen Teile ist an dieser Stelle nicht relevant.

Aktoren

Zu den Aktoren zählen unter Anderem die bereits beschriebenen Komponenten Licht und Motoren (s. Kapitel 3.5.3 und 3.5.2). Ein Exponat, das während des Toaster-Edwin-Kurses (s. Kapitel 2.3.2) entstand, nutzte Ton als Aktor. Über Abstandssensoren wurden je nach Entfernung Töne erzeugt und über Lautsprecher ausgegeben.

„Edwina“ ist eine Kreatur, die als interaktives Musikinstrument genutzt werden kann (s. Abbildung 3.1). Eine Grußkarte, die durch Betätigung eines Auslösers zehn Sekunden Ton aufzeichnen und durch Ansteuerung wieder abspielen kann, ist ein weiteres Beispiel für die Nutzung von Ton als Aktor (s. Abbildung 2.12).

Der Kurs „Das Unsichtbare“ unter Leitung von Prof. Dr. Gunter Klemke, Prof. Franziska Hübler und Prof. Dr. Birgit Wendholt beschäftigt sich derzeit mit der Projektion als Aktor. Durch die Nutzung von Processing werden Kameradaten verarbeitet und in Form von Projektionen eine Aktion getätigt. Processing ist eine Open Source Programmierumgebung und Programmiersprache auf der Basis von Java. Die Beliebtheit bei Künstlern und Studenten ist darauf zurückzuführen, dass ein schneller Einstieg in die graphische Programmierung möglich ist. [28]



Abbildung 3.1.: Edwina ein interaktives Musikinstrument

Mikrocontroller

Näheres über die Gründe für die Verwendung von Arduinos ist in der Ausarbeitung von Sebastian Gregor für den Arduino Duemilanove zu finden. [31] Er vergleicht die Tools Phidgets, d.tools und Arduino. André Jeworutzki stellt Boda Blocks, Quilt Snaps, Teeboard, Lego Mindstorm und LilyPads gegenüber. [35]

Arduino

Arduino ist eine Open Source Physical Computing Plattform. Es besteht aus einer einfachen Mikrocontrollerplatine und der dazugehörigen Software, um diese zu programmieren. Es gibt

unterschiedliche Boards. Für Schülerkurse und im Bereich Wearable Computing haben sich LilyPad Arduinos als geeignet erwiesen, während für Installationen die Arduino Duemilanove gängig sind. Für viele Objekte wurde gegen Ende der Vorstudien auch der Arduino Pro Mini häufiger verwendet. [8]

Eine Einführung in die Programmierung unter Arduino befindet sich im Anhang A.

Der Hauptvorteil bei Arduino ist die Erweiterbarkeit durch Steckplatinen. Im Folgenden werden ausgewählte vorgestellt.

Motor Shield Die Ansteuerung von Motoren, Steppern und Servomotoren wird durch das Motorshield für den Arduino vereinfacht. [44] Die Bedienung ist, wie bei den meisten Produkten von Arduino, sehr einfach. Es gibt für die Ansteuerung der Motoren Codebeispiele. Es ist also mit einem fertigen Motorshield schnell möglich, Tests durchzuführen.

Ein Nachteil ist, dass das Motor Shield vor der Nutzung gelötet werden muss (s. Abbildung 3.2).



Abbildung 3.2.: Motor Shield für den Arduino Duemilanove [44]

Weiterhin ist das Motor Shield recht groß und nicht für das LilyPad erhältlich. Für die meisten Motoren wird eine externe Stromversorgung benötigt, so dass der USB-Anschluss nicht ausreicht. Diese Nachteile führen dazu, dass nur selten ein Motor Shield in den Projekten verwendet wurde.

XBee Shield Ein XBee Shield existiert sowohl für das LilyPad Arduino als auch für Arduino Platinen. Durch diese Steckplatine ist es möglich, kabellos über ZigBee [1] zu kommunizieren (s. Abbildung 3.3). Ein Vorteil ist hierbei, dass nicht gelötet werden muss. Die Benutzung ist jedoch etwas komplexer als gewohnt.

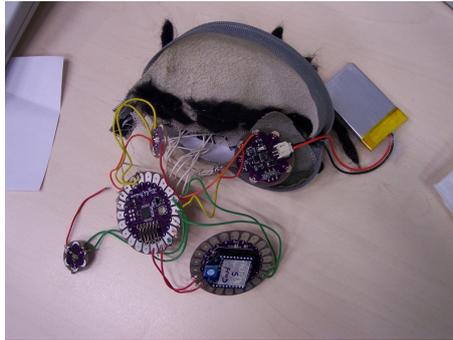


Abbildung 3.3.: Erste XBee-Tests

Weitere Shields Das Wave Shield stellt eine Steckplatine für den Arduino Duemilanove dar [45] und kann 12 bit unkomprimierte Audiodateien bis 22KHz abspielen (s. Abbildung 3.4). Die Audiodateien werden von einer SD/MMC Karte gelesen. Auch diese Steckplatine muss gelötet werden. Verwendung hat das Wave Shield bei der Bachelorarbeit von Svenja Keune gefunden. Mit seiner Hilfe konnten Morsezeichen wiedergegeben werden (s. Kapitel 2.6.3).

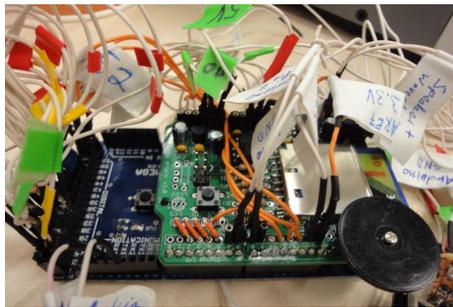


Abbildung 3.4.: Wave Shield innerhalb der Bachelorarbeit von Svenja Keune

Ein weiteres Shield ist das GPS Shield und Data Logger (s. Abbildung 3.5). [43] Dieses Shield kann die genaue Position und Zeit speichern.

Vorteile bei der Nutzung der Arduino Plattform

Da viele Nutzer diese Plattform verwenden, existiert eine große Community, mit deren Hilfe man entstehende Probleme schnell via Internet lösen kann. Außerdem handelt es sich um eine Open Source Software und auch die Hardware ist kostengünstig. Eine einfache Handhabung sowohl in der Soft- als auch in der Hardware ist darauf zurückzuführen, dass es speziell für Designer und Künstler entwickelt wurde. Des Weiteren ist Arduino eine Cross-Plattform, d.h. die Software läuft unter Windows, Macintosh OSX und Linux. [7]



Abbildung 3.5.: GPS Shield und Data Logger [43]

Arduino Duemilanove

Der Arduino Duemilanove wurde in den Vorstudien häufig eingesetzt (s. Abbildung 3.6). Ein großer Vorteil ist, dass viele Erweiterungen in Form von Steckplatinen existieren. Auch der Netzteilanschluss, der sich direkt auf dem Board befindetet, war häufig hilfreich, denn der Arduino Duemilanove kann bis zu 12 Volt am Ausgang liefern. Nachteile des Duemilanove sind, dass er recht groß ist und bei einer Verwendung von Shields viele I/Os blockiert sind. [4]

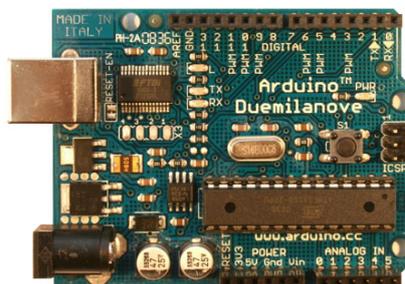


Abbildung 3.6.: Arduino Duemilanove [6]

LilyPad Arduino

Das LilyPad wurde von Leah Buechley [14], Professorin am Massachusetts Institute of Technology [48] in Boston, entwickelt. Eine einfache Benutzung ermöglicht schnelles Prototyping. Das Ziel der LilyPads liegt darin, ein analoges Baukastensystem zu den Lego Mindstorms darzustellen (s. Abbildung 3.7). Leah Buechleys Forschungsschwerpunkt liegt im Bereich Pädagogik. In den Kursen konnten gute Erfahrungen mit dem LilyPad gesammelt werden. Sie sind leicht zugänglich für Technik-Unerfahrene. Leah Buechley stellt des Weiteren gute Online-Tutorials zur Verfügung (s. Abbildung 3.7). [15] [15] In den Tutorials sind sowohl Einlei-



Abbildung 3.7.: Screenshot Leah Buechley Tutorial [15]

tungen für die Verkabelung als auch Codebeispiele vorhanden. Der Test im Kurs Faszination Games 1 (s. Kapitel 2.4.5) zeigte jedoch, dass eine schnelle Nutzung in einer niedrigen Altersstufe nicht möglich ist. Ein wichtiger Grund hierfür ist die Sprachbarriere, da die Tutorials in englischer Sprache verfasst sind.

Der Mikrocontroller auf dem LilyPad ist ein Atmel ATmega168V [2] oder ATmega328V [3] mit 8 MHz Taktgeschwindigkeit. Er arbeitet mit einer Betriebsspannung zwischen 2,7 und 5,5 Volt. [7] [5]

Das LilyPad Arduino hat vor allem den Vorteil, dass es sowohl vernähbar als auch waschbar ist. Shields wie bei dem Arduino Duemilino Board existieren allerdings bisher nicht.

3.1.2. Laborausstattung

Zu einem Interactive-Design-Labor gehört mehr als nur eine Rechnerausstattung. Die benötigte Hardware an Sensoren, Aktoren und Microcontroller wurden bereits beschrieben (s. Kapitel 3.1.1). An dieser Stelle wird die Laborausstattung dargestellt, die sich innerhalb der Vorstudien bewährt hat.

Ein Interactive-Design-Labor sollte, wenn möglich, transportabel sein, damit Kurse an verschiedenen Orten stattfinden können. Dabei haben sich Classroomkits bewährt.

Classroomkits

Die erfolgreiche Bearbeitung von Aufgaben im Bereich Interactive Design in kleinen Teams wird durch Classroomkits unterstützt. Ein solches Kit enthält Hardware-Teile, die man auch zur Durchführung von schnellem Prototyping verwenden kann (s. Kapitel 4.2.4). Zusätzlich zu dem Kit werden Kabel benötigt. Diese befinden sich in einer Gemeinschaftskiste für alle Gruppen.

Der Inhalt der sogenannten Classroomkits variierte im Laufe der Vorstudien. Insbesondere die Anzahl der einzelnen Teile musste an das entsprechende Konzept angepasst werden. Im Folgenden werden nur die Grundbestandteile aufgezählt (s. Abbildung 3.8).

Exemplarisch befindet sich eine genaue Aufstellung über den Inhalt, der während des Toaster Edwin Kurses (s. Kapitel 2.3.2) verwendet wurde, im Anhang A.



Abbildung 3.8.: Classroomkits

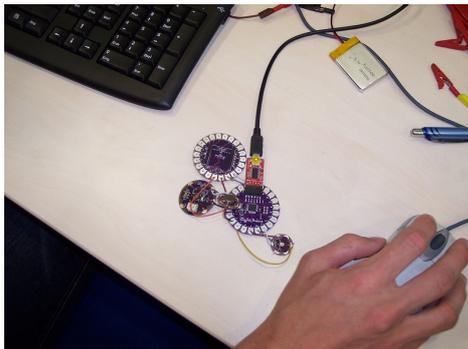


Abbildung 3.9.: FTDI-Platine für die Kommunikation mit dem Rechner

LilyPad Arduino und Zubehör Zusätzlich zu den LilyPad Arduinos werden sowohl ein Mini-USB-Kabel als auch eine FTDI-Platine benötigt, um die Kommunikation mit dem Rechner zu gewährleisten (s. Abbildung 3.9).

Ist der LilyPad Arduino an den Rechner angeschlossen, wird die Stromversorgung über das USB-Kabel sichergestellt. Weitere Bestandteile des Kits sind ein LilyPad LiPower und ein Polymer Lithium Ionen Akku, sowie ein LiPo Charger. Der Akku kann an den LiPower angeschlossen und für die Aufladung über den LiPo Charger genutzt werden. Zum Aufbau der Stromkreise sind Krokodilsklemmen hilfreich. Farblich wurden schwarze und rote verwendet, um plus und minus leicht unterscheiden zu können.

Die Krokodilsklemmen sind für den Anschluss an die I/Os der LilyPads gut geeignet (s. Abbildung 3.10).

Weitere von Leah Buechley entwickelten LilyPad-Bestandteile des Kits sind RGB LEDs, Buzzer, helle weiße LEDs, Lichtsensor, Temperatursensor, Buttonplatine, Vibrationsplatine und Beschleunigungssensor.

Auch Funkmodule, die LilyPad XBee Platine, der XBee Explorer USB, sowie zugehörige XBee Chips sind in dem Kit vorhanden (s. Abbildung 3.17a).

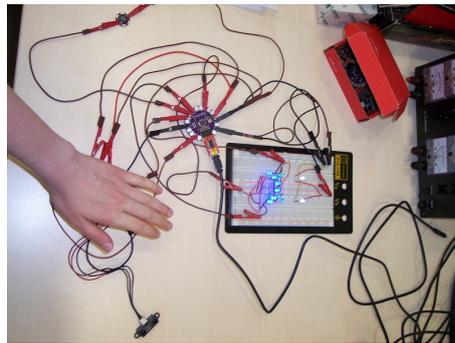


Abbildung 3.10.: Versuchsaufbau mit Krokodilsklemmen

Duemilanove Das Kit beinhaltet einen Arduino Duemilanove. Zusätzlich wird ein normales USB-Kabel benötigt, um die Programme auf den Mikrokontroller zu laden und eine Stromversorgung zu gewährleisten.

Restliche Teile Ein kleiner Motor und ein Servomotor sind für Bewegung zuständig. Auf größere Motoren, wie zum Beispiel Schrittmotoren wird verzichtet, da sie eine größere Stromversorgung als die anderen Teile des Kits benötigen. Die Kits müssten sonst um Netzteile und Motor Shields erweitert werden.

Außerdem sind ein weiterer Sensor, der Flex Sensor, und ein USB Hub vorhanden.

Raumausstattung

Der Raum ist ausgestattet mit acht Rechnerarbeitsplätzen. Auf den Rechnern muss die Arduino-Software installiert sein. [7] Außerdem liegen auf dem Desktop sämtliche Folien als Hilfestellung. Des Weiteren wurden an den Wänden Abbildungen von LilyPads mit diversen Sensoren und Aktoren angebracht. Die Abbildungen, auf denen auch die nötige Verkabelung der Hardware zu sehen ist, dienen den Teilnehmern zusätzlich als Unterstützung (s. Abbildung 3.11).

Designanteil

Für eine erfolgreiche Zusammenarbeit muss der Raum zum Beispiel Nähmaschinen, Bügel-eisen und -brett sowie Puppen beinhalten.

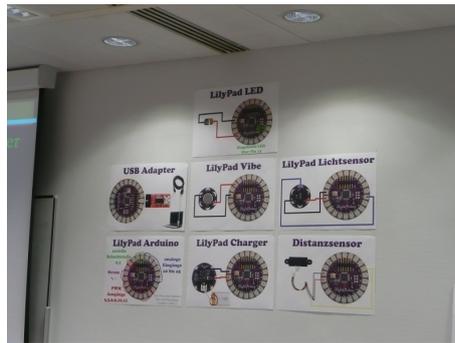


Abbildung 3.11.: Hilfestellung für eine erfolgreiche Verkabelung

3.2. Bezug zu den Forschungsgebieten

3.2.1. Wearable Computing

Die Arbeiten begannen unter dem Thema Wearable Computing. An dieser Stelle wird dargestellt, welche Schwierigkeiten sich in diesem Bereich ergeben haben. Abschließend werden die Vorteile von Interactive Design erläutert.

Herausforderungen beim Wearable Computing

Die Herausforderungen im Bereich Wearable Computing sind sehr vielfältig. André Jeworutzki hat in seinem Masterprojektbericht folgende Herausforderungen dargestellt:

„- Gewährleistung der Mobilität

- Eingeschränkte Stromstärke und Stromkapazität
- Optimale Platzierung und Verbindung der Technik finden
- Technik auf Stoff fixieren, der dünn und flexibel ist
- Technik unsichtbar in die Kleidung integrieren“ [34]

Diese Ausarbeitung bezieht sich auf die Studentischen Projekte von Helene All (s. Kapitel 2.6.2) und Ekaterina Ibraimova (s. Kapitel 2.6.1).

Diese Herausforderungen sollen zum Teil aufgegriffen und ergänzt werden.

Die intelligente Kleidung muss:

- tragbar sein
- darf den Benutzer nicht behindern
- gut gestaltet sein
- muss einen Nutzen haben, wobei an dieser Stelle auch ein gutes Design einen Nutzen darstellt

Daraus ergeben sich folgende Herausforderungen an Hard- und Software:

Herausforderungen an die Software Daten müssen erfasst, gesammelt, synchronisiert und übertragen werden.

Herausforderungen an die Hardware Eine mobile Energieversorgung und die Platzierung der Hardware stellen besondere Anforderungen dar. Da gewisse Design-Vorgaben häufig eine kabellose Datenübertragung voraussetzten, stellte sich die Verlegung der Kabel immer wieder als Hindernis heraus. Beispiel hierfür ist das Projekt Ekaterina: Hier musste im fertigen Overall gelötet werden, um die Kommunikation der einzelnen LilyPads zu gewährleisten (s. Kapitel 2.6.1). Des Weiteren darf die Hardware nicht heiß werden und keine gefährliche Spannung erzeugen.

Bewertung Das Wearable Computing gestaltet sich für die Informatik in der interdisziplinären Zusammenarbeit häufig als schwierig. Die Vorstudien beweisen, dass der fachliche Schwerpunkt bisher im Design lag.

Der Vollständigkeit halber ist zu erwähnen, dass Wearable Computing auch in unseren Kurs-Projekten eine große Rolle gespielt hat. In diesem Bereich ist nicht nur Leah Buechley zu erwähnen, deren Forschungen bereits genauer dargestellt wurden (s. Kapitel 3.1.1). Auch Arbeiten wie „Deploying a Wearable Computing Platform for Computing Education“ [51] zeigen, dass Wearable Computing für Kursprojekte, in denen Anfänger an Technik herangeführt werden, erfolgreich genutzt werden können. Dies konnte in den Vorstudien belegt werden (s. Kapitel 2.3.1).

Jedoch wurde in den Vorstudien deutlich, dass eine Beschränkung auf Wearable Computing nicht nötig ist. Die Teilnehmer sollten in ihrer Kreativität nicht eingengt werden. Kreative Objekte müssen nicht unbedingt „wearable“ sein, um die in den Kursen geplanten Inhalte zu vermitteln.

3.2.2. Gründe für Interactive Design

Nicht „wearable“ sein zu müssen bedeutet, dass der Kreativität keine Grenzen gesetzt werden. Es können große Installationen wie ebenso viele kleine Objekte entwickelt werden. Mit der Kombination komplexer Interaktionen, agentenbasiertem Lernen aus der künstlichen Intelligenz und einer Funktechnologie wie Zigbee [1] sind sogar Schwarmalgorithmen oder intelligenter Staub umsetzbar. [34]

„Playfulness is a innate sentiment factors of human. Spontaneously, people have this general whenever they are working, studying and doing other activities. [!]The change of concept in modern product design has gone to the people-centered, which makes the interactive emotional experience become a key point in the interactive design. This optional experience includes easy-to-use, effective operation, security, and a sense of accomplishment, visual pleasure and emotional meet.“ [46]

Daraus lässt sich folgern, dass Spielen der Schlüssel ist, der Interactive Design so erfolgreich macht. Menschen haben einen Spieltrieb, den sie in allen Lebenslagen nutzen. Es hat einen Wandel im modernen Produktdesign gegeben: Der Mensch wird in den Vordergrund gestellt. Die interaktive, emotionale Erfahrung im Interactive Design sollte deshalb eine einfache Benutzbarkeit und eine effektive Arbeitsweise, Sicherheit, visuelle Freude und einen emotionalen Wettkampf beinhalten. [46]



Abbildung 3.12.: Poster Herbsthochschule [29]

Interactive Design bietet außerdem viele Möglichkeiten, um Schüler und Lehrer spielerisch an technische Bereiche heranzuführen und wäre somit ideal, den zu wenig geförderten technischen Unterricht interessanter zu gestalten. [50]

In der Informatik ist besonders eine Frauenförderung wichtig. In diesem Bereich wird bereits viel unternommen, und auch hier bietet Interactive Design eine hervorragende Möglichkeit, junge Mädchen für die Informatik zu begeistern. In den Vorstudien ist in der „Herbsthochschule für Mädchen“ ein solcher Versuch unternommen worden (s. Kapitel 2.4.3) (s. Abbildung 3.12).

„I was going to be a ballet dancer, but I think I will be an electric scientist like you now“ [12]

Aussagen wie diese von einer Teilnehmerin, die nach einem Kurs ihren Berufswunsch ändert, zeigen, wie erfolgreich solch eine Art der Frauenförderung sein kann.

„Although not all students will (or should) pursue careers in engineering or technological fields, all can benefit from a basic understanding of the nature of technology and how it transforms social, economic, and cultural systems. Engineering outreach programs introduce students to the joys and frustrations of engineering.“ [64]

Auch in unseren Kursen war es nicht ausschließlich das Ziel, alle Teilnehmer für ein Studium in der Informatik zu begeistern. Strukturiertes Vorgehen, Teamarbeit und die Fähigkeit Probleme selbstständig zu lösen, sind auch in anderen Berufsfeldern wichtig.

3.3. Kurse

In den Kursen konnte bewiesen werden, dass die Konzepte es ermöglichen ganz unterschiedliche Teilnehmer an diesen Bereich heranzuführen. Es konnten Erfahrungen mit Schülern, Lehrern, Studenten und Professoren gesammelt werden. Durch gute Vorbereitung, Konzepte und Hardware ist es heute möglich, viele verschiedene Arten von Kursen anzubieten. In keinem der Kurse wiederholten sich die Probleme des Pentiment Kurses von 2008 (s. Kapitel [2.2.1](#)).

Es erwies sich in den Kursen als hilfreich, die Gruppe zuerst darum zu bitten, ihre Verkabelung erneut zu prüfen. Gerade bei der Verwendung von Krokodilsklemmen können leicht Fehler entstehen (s. Kapitel [3.10](#)). [15] Auch wurden häufig die Objekte an falsche I/Os angeschlossen. Die Unterscheidung von plus und minus ist nicht immer trivial.

Alle Konzepte konnten den Spaßfaktor erfolgreich integrieren.

3.4. Probleme im Umgang miteinander

Im Folgenden wird auf die Probleme eingegangen, die während der Vorstudien aufgetreten sind. Einige davon wurden durch die Unterschiedlichkeit der beiden involvierten Fachbereiche ausgelöst. Zu diesen Problemen gehörten die Kommunikation, die verschiedenen Arbeitsweisen und ein schwieriges Team-Management.

3.4.1. Kommunikation

Bei einer interdisziplinären Zusammenarbeit gibt es wie schon innerhalb der Vorstudien erwähnt, einige Schwierigkeiten. Viele entstandene Probleme und Missverständnisse sind auf fehlerhafte Kommunikation zurückzuführen.

Probleme in der Kommunikation zwischen Menschen, beschrieben von Paul Watzlawik, [67] sind schwierig. In einer Kooperation zwischen zwei verschiedenen Fachbereichen wie der Informatik und dem Design kommt erschwerend hinzu, dass beide Fachsprachen sehr unterschiedlich sind. Die Vorstudien haben gezeigt, dass es immer wieder nötig ist, dem Gegenüber Begriffe zu erklären, die einem anderen Fachmann geläufig wären.

Das Prinzip „Übung macht den Meister“ gilt hier nur begrenzt. Die Kommunikation wurde im Laufe der Zeit zwar besser, allerdings konnten die Probleme bis zum Ende der Vorstudien nie gänzlich ausgeräumt werden. So fiel es besonders schwer, dem Kooperationspartner alle Fachtermini angemessen zu erklären. Dies war auf beiden Seiten ein Problem. Es bedarf noch vieler Projekte, um dieses Problem vollständig auszuräumen. Bis dahin ist Rücksichtnahme gefordert.

Kommunikationsprobleme ziehen auch andere Probleme nach sich: So verursachte zum Beispiel eine mangelhafte bzw. fehlerhafte Kommunikation, dass die gegenseitigen Erwartungen nicht erfüllt werden konnten.

Es ist nicht nur nötig, die unterschiedlichen Fachbegriffe aus dem Weg zu räumen. Es ist auch wichtig, zu verstehen was der andere sagt. Ein Designer spricht häufig in Bildern. Die Vorstellungen, die ein Informatiker hat, wenn ein Designer über Formen und Farben spricht, stimmen häufig nicht mit denen des Designers überein (s. Abbildung 3.13). Die Designer sind es gewohnt, Skizzen ihrer Ideen anzufertigen. Für einen Informatiker reichen diese aber nicht aus. Bei der Beschreibung der technischen Umsetzung besteht eine große Gefahr, dass beide aneinander vorbeireden, weil der Informatiker wahrscheinlich unwissentlich ein falsches Bild von der Idee hat.

Das nächste Problem ist nun, dass der Designer die Fachsprache des Informatikers nicht versteht und deshalb nicht nachvollziehen kann, was von seinen Vorstellungen eigentlich umsetzbar ist. So kam es in den Vorstudien an mehreren Stellen zu Enttäuschungen (s. Kapitel 2.6.1). Diese Probleme sind nur durch einen hohen Kommunikationsaufwand und enge Teamarbeit zu beseitigen.



Abbildung 3.13.: Skizze Svenja Keune Bachelorprojekt

3.4.2. Arbeitsweise

Die Arbeitsweise von Designern und Informatikern schien zunächst sehr unterschiedlich und die Herangehensweise an ein Projekt nahezu gegensätzlich. Während viele Vorgehensmodelle der Informatiker eine akribische Planung und Dokumentation voraussetzen, erscheint die Arbeitsweise von Designern eher spontan.

Ein weiteres Problem der verschiedenen Arbeitsweisen beschreiben Julia Pressburger und Sven Tennstedt in ihrem Masterprojektbericht über Ambient Awareness:

„Im Laufe des Projektes fiel auf, dass Informatiker und Designer aus verschiedener Art Holz geschnitzt sind und auf unterschiedliche Weise an ihre Arbeit heran gehen. Während die Informatiker es durch den Aufbau ihres Studiums gewohnt sind im Team zu arbeiten, steht im Studium der Designer scheinbar der eigene Stil und die eigene und damit alleinige Arbeit im Vordergrund.“ [65]

Informatik

Die Informatik neigt dazu zu abstrahieren. Es werden Projektmodelle erstellt, welche dem Verständnis vor der Situation(Theorie) und zur Anleitung in der Situation(Methode) dienen. [22]

Innerhalb der Informatik sind Baumstrukturen, Graphen oder Diagramme gebräuchliche Darstellungsformen. Dies hat sich in der Zusammenarbeit als wenig hilfreich erwiesen, da Designer in ihrem Studium keinen Kontakt zu solchen Datenstrukturen haben.

Informatiker halten sich an Richtlinien und gehen nach Mustern vor. Wenn man sich an gewisse Vorgaben hält, erhöht das die Wahrscheinlichkeit, dass ein Produkt am Ende gelingt. Logisches und methodisches Vorgehen stehen hier im Vordergrund. Einige aus dem Software Engineering bekannten Modelle werden später genauer erläutert (s. Kapitel 4). An dieser Stelle ist vorwegzunehmen, dass eine Messbarkeit an Qualitätsmerkmalen gegeben ist. Als Beispiele für die Merkmale guter Softwarequalität sind zum Beispiel Wartbarkeit, Zuverlässigkeit, Effizienz, Benutzerfreundlichkeit und Portabilität zu nennen. [19]

Im Studium werden Informatiker dazu ausgebildet, den Kunden zu verstehen und genau das umzusetzen, was er will. Er hat weniger Freiheiten in der Umsetzung als der Designer.

Design

Designer entwerfen Skizzen, bevor sie mit der Entwicklung beginnen. Sie können diese Skizzen auch näher erläutern, allerdings ist es in den Vorstudien an dieser Stelle immer wieder zu Verständnisproblemen gekommen (s. Abbildung 3.14).

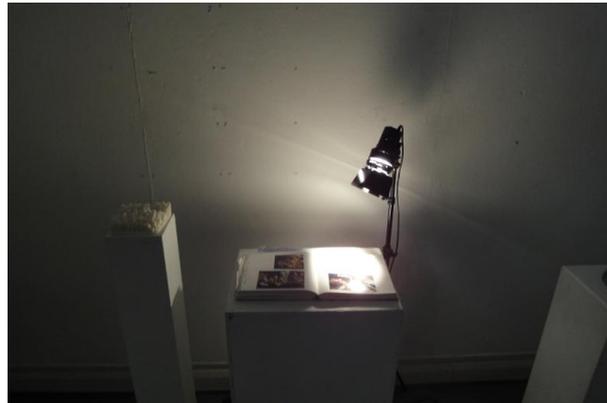


Abbildung 3.14.: Skizzenbuch Svenja Keune

Die Methoden der Designer sind sehr unterschiedlich. Es gibt zum Beispiel Zeichenkurse, bei denen für eine Aufgabe sehr schnell und in Umrissen gezeichnet, in der nächsten Aufgabe dagegen sehr detailgetreu gearbeitet werden muss.

Häufig werden Design Workshops zu bestimmten Themen oder Fragestellungen veranstaltet. Ziel solcher Workshops ist, entweder ein Produkt oder ein Konzept zu entwerfen. Durch solche Workshops können Ideen generiert werden.

Einen Einblick in die Art des Studiums im Department Design ermöglicht folgendes Beispiel:

Am Anfang des Semesters wird ein Thema vorgegeben, zum Beispiel Fehler zu kultivieren. Hiermit können unter Anderem Löcher gemeint sein oder aber zufällig entstandene Fehler. Es ist immer wichtig, dass möglichst kreative Lösungen gefunden werden.

Für die Erstellung eines Produktes gibt es verschiedene Herangehensweisen:

1. materialorientiert - mit vorgegebenen Materialien experimentieren
2. zielorientiert - es muss ein Ziel mit Hilfe eines Konzeptes erreicht werden

In der Bearbeitungsphase eines Objektes ist es für einen Designer immer möglich, von dem eigentlichen Ziel abzuweichen. Es ergeben sich im Entwicklungsprozess teilweise neue, bessere Ziele. Es gibt Designer, die Entscheidungen aus dem Bauch heraus treffen. Andere planen und entwerfen Konzepte.

Schwierigkeiten bekommen Designer dann, wenn dem Betrachter nicht gefällt, was er sieht. In der Zusammenarbeit gab es oft völlig enttäuschte Designer, die verzweifelt waren, weil ihrem betreuenden Professor die Ideen nicht gefielen. Diese fehlende Planbarkeit ist nicht nur enttäuschend für den Studierenden, sondern erschwert die Zusammenarbeit immens.

Geschmack ist schließlich relativ, und Schönheit ist schwer messbar. Die Qualität ist an dieser Stelle also nicht messbar. Der Designer ist auch ein Betrachter, und ihm muss an jeder Stelle des Entwicklungsprozesses gefallen, was er sieht. Das heißt, wenn ein Zwischenergebnis nicht seinen Vorstellungen entspricht, ist es durchaus möglich, dass die komplette Idee verworfen wird. Das Vorgehen ähnelt dem Wasserfallmodell (s. Kapitel 4.1): Entweder es funktioniert oder es geht komplett schief. Es kann jedoch auch sein, dass die Idee so lange überarbeitet wird, bis das Ergebnis gefällt. Dann handelt es sich eher eine iterative, evolutionäre Herangehensweise.

Ein Beispiel gibt: *„Auf Anstoß meiner Professoren diesmal mit mehr Farbe. Allerdings sehe ich dem etwas kritisch entgegen. Farben sind zwar schön, sind aber auch Geschmacksache. Jeder hat andere Geschmäcker, sie verändern sich schnell und man hat sich schnell an ihnen satt gesehen. Farben sind nicht neutral und können daher nicht wertfrei angesehen werden. Ist es also sinnvoll sie zu benutzen, obwohl man dadurch Betrachter abstößt und Vorurteile schürt? Andererseits können Farben ja auch sehr positive Stimmungen und Reize hervorrufen.“* [39]

Design ist nicht in allen Bereichen frei von Vorgaben. Als Beispiele sind hier Kommunikations- oder Industriedesigner zu nennen. Soll zum Beispiel ein Auto gestaltet werden, ist das natürlich nicht ohne genaue Vorgaben möglich. Kommunikationsdesigner müssen unter anderem Kampagnen planen. Auch hierfür bekommen sie eine Menge Kontext, wie zum Beispiel die Zielgruppe des zu bewerbenden Produktes. Innerhalb der Vorstudien beschränkte sich der Kontakt zu beiden Studiengängen auf die Teilnahme am Pentiment Kurs (s. Kapitel 2.3.2).

Vorteile dieser Arbeitsweise Im Design können Fehler als Lernereignis betrachtet werden, so dass sich auf dem Weg etwas entwickeln kann. Es ist sogar möglich, einen ganz anderen Weg zu gehen, als am Anfang geplant. Ein Fehler kann sogar das Endergebnis sein.

Dies beschreibt Svenja Keune (s. Kapitel 2.6.3): *„Vielfach tauchten durch die Suche nach einer Lösung auch interessantere Ansätze und Ideen auf. Die Schwierigkeiten waren einfach nur ein Teil des ständigen Weiterentwicklungsprozesses. Dieser Prozess ist noch nicht abgeschlossen. Es gibt Ansätze oder Verbesserungsmöglichkeiten die bearbeitet werden müssten. Dies wird wohl ein Teil der Bestrebungen im Master werden.“* [40]

3.4.3. Team-Management

Das Management eines Projektes ist schon schwierig, wenn nur Informatiker beteiligt sind. Hierzu gibt es bereits viele Überlegungen, wie man die Zusammenarbeit in einem solchen Projekt ermöglicht. Ein Beispiel hierfür ist das Buch „The Mythical Man-Month: Essays on Software Engineering“. [13]

Auch Kai Rosseburg beschreibt in seinem Masterprojektbericht über Ambient Awareness (s. Kapitel 2.2.2) diese Problematik. [58]

„Eine Sache, die für nächste Projekte wünschenswert wäre, ist eine dedizierte Projektleitung. Dieses Projekt mit seinen acht Teilprojekten und fast 30 Teilnehmern, war eigentlich zu umfangreich, um es ohne Projektleitung, die den Blick für das Ganze bewahrte, zu realisieren. Ohne das Engagement einzelner Studenten hätte es an dieser Stelle große Probleme gegeben.“ [58]

Man braucht einen Projektmanager, der dafür sorgt, dass die aus dem Software Engineering bekannten Probleme das Projekt nicht gefährden. Eine solche Form der Kontrolle ist in einem interdisziplinären Projekt schwer umzusetzen, da jeder Fachbereich gleichberechtigt sein soll. Eine Entscheidung, ob ein Designer oder ein Informatiker die Leitung übernimmt, ist also kaum möglich.

Die Umsetzung erfordert durch das Fehlen einer Projektleitung ein hohes Maß an Selbstdisziplin der einzelnen Teilnehmer. Außerdem sind ausgeprägte Schlüsselqualifikationen nötig. Der Umgang ist wegen der vielen Probleme nicht immer leicht. An Schulen wird zum Erwerb sozialer Handlungskompetenzen das Buddy-Projekt umgesetzt. Dieses arbeitet nach dem Motto 'Aufeinander achten. Füreinander da sein. Miteinander lernen'. Es soll dazu führen, dass Schüler mehr Verantwortung für sich und andere übernehmen und Konflikte konstruktiv lösen. [69]

Alle diese Fähigkeiten, die Schüler in diesem Projekt lernen sollen, sind vonnöten, um ein interdisziplinäres Projekt durchzuführen.

3.5. Technische Probleme

Im Folgenden werden die Erfahrungen geschildert, die einzelne Projekte in Bezug auf die Nutzung der Technik sammeln konnten.

3.5.1. Debugging

Das Debuggen mit Hard- und Software ist äußerst schwierig, da der Fehler auf beiden Seite liegen kann.

3.5.2. Motoren

Motoren erwiesen sich sowohl in den Einzelprojekten als auch in den Kursen als Problem. Sie verbrauchen in der Regel mehr Strom als LEDs. Fehlende Kenntnisse im Bereich der Mechanik erschwerten zusätzlich die Benutzung. Ein der Aufgabe angemessener Einbau erwies sich häufig als schwierig. Ein Beispiel hierfür ist ein Gruppenprojekt aus dem Pentiment Kurs 2009 (s. Kapitel 2.3.2). Die Idee der Teilnehmer bestand darin, einen Wecker zu bauen, der weglaufen kann. Dieses Weglaufen zu simulieren, gestaltete sich als äußerst schwierig. Der Bewegungsablauf des Laufens konnte schließlich mit einer Metallkonstruktion simuliert werden (s. Abbildung 3.15).



Abbildung 3.15.: Beweglicher Wecker

Kuppel

Bei einem weiteren Projekt von Helene All (s. Kapitel 2.6.2) erwies sich die genaue Ansteuerung der Servomotoren als schwierig. Ein Metallgerüst sollte seitlich an einem Modell hoch- und runterfahren. Gewährleistet wurde dies mit einer Metallkonstruktion. Das Stahlgerüst war zu schwer für die im Labor befindlichen Servomotoren. Es waren sehr große Servos nötig, um die entsprechende Leistung erzielen zu können. Das Stahlgerüst war äußerst schwer zu transportieren, da nach jedem Auf- bzw. Abbau die Servos neu eingestellt werden mussten. Wurde dies vergessen, mussten die Servos ersetzt werden (s. Abbildung 3.16). [34]

3.5.3. Lichtquellen

Bei den LEDs gab es Probleme mit der Ansteuerung über LilyPads, da die Möglichkeiten sehr begrenzt sind. Laut Spezifikation lassen sich nur 10 LEDs gleichzeitig betreiben. Außerdem war es schwierig, die LEDs über leitfähiges Garn mit Strom zu versorgen, denn der Widerstand des Garns war zu hoch, so dass die Helligkeit der LEDs stark reduziert wurde.



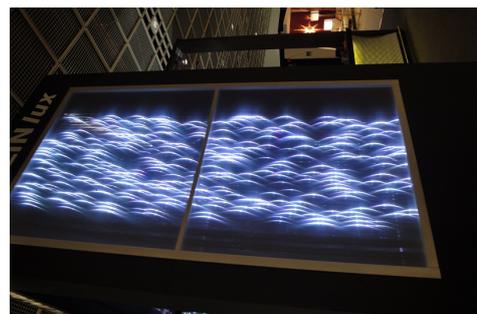
Abbildung 3.16.: Kuppel

In den Projekten wurde der Wunsch laut, das Licht von LEDs zu streuen. Nach vielen Test stellte sich heraus, dass Tischtennisbälle einen angenehmen Effekt haben (s. [Abbildung 3.17a](#)).

Vielversprechende Materialien wurden auf der Light-and-Building-Messe in Frankfurt entdeckt (s. [Abbildung 3.17b](#)).



(a)



(b)

Abbildung 3.17.: Streumaterial

3.5.4. Energieversorgung

Kurze Akkulaufzeiten machten es erforderlich, dass ein Austausch der Batterie jederzeit möglich sein muss. Denkt man an eine Modenschau werden die Grenzen offensichtlich. Auch die Größe und Form der Akkus ist problematisch.

Die Energieversorgung ist selbst bei statischen Objekten nicht selten ein Problem, da zum Teil lebensgefährliche Spannungen eingesetzt werden müssen.

3.5.5. Haltbarkeit der Hardware

Sämtliche Hardware sollte robust sein. Die Erfahrung hat gezeigt, dass Designer häufig zu Hause an den Objekten weiterarbeiten (s. Kapitel 4.4.2). Also muss ein Objekt transportabel sein. Intelligente Kleidungsstücke sollen notwendigerweise waschbar sein, deshalb wurden LilyPad Arduinos verwendet (s. Kapitel 3.1.1).

Ein Problem, das ebenfalls sehr häufig auftrat, waren Kurzschlüsse. Durch unsachgemäße Befestigung funktionierten Objekte während der Präsentation nicht.

3.6. Fazit

In der Zusammenarbeit gibt es viele Schwierigkeiten. Sowohl technisch als auch menschlich gesehen gestaltet sich eine reibungslose Zusammenarbeit als große Herausforderung. Es müssen Wege gefunden werden die interdisziplinäre Zusammenarbeit von Informatik und Design zu optimieren. Es ist zu prüfen, ob aus der Informatik bekannte Methoden anwendbar sind.

Erste Erfolge konnten bereits erzielt werden, wie man der Bachelorarbeit von Svenja Keune (s. Kapitel 2.6.3) entnehmen kann:

„Probleme oder Schwierigkeiten gab es immer und überall. In dem Projekt traten sich gehäuft auf, weil nicht nur gestalterische Fragen und Umsetzungsprobleme auftraten, sondern auch Integrationsprobleme der Technik, Software- und Hardwarefehler und Verständnisschwierigkeiten. Auf Grund der langen Kooperation, vieler Erfahrungen, fachübergreifendes Wissen auf beiden Seiten und außerordentliches Engagement durch den Masterstudenten André Jeworutzki konnten so ziemlich alle Schwierigkeiten gelöst oder umgangen werden.“ [40]

4. Methoden für eine interdisziplinäre Zusammenarbeit von Informatik und Design

Im Folgenden werden Modelle der Informatik vorgestellt und geprüft, inwieweit diese in der interdisziplinären Zusammenarbeit von Informatik und Design anwendbar sind. Solche Vorgehensmodelle werden unter Anderem dazu genutzt, Vorhersagen und Standards von Qualität oder Kosten einzuhalten.

Des Weiteren werden die Anforderungen an ein Kunstprojekt die Martin Sukale formuliert hat [63] [62] auf ihre Anwendbarkeit geprüft und anhand der Erfahrungen aus den Vorstudien erweitert.

Im Anschluss wird der Ablaufplan einer Vorstudie vorgestellt.

4.1. Sequenzielle Vorgehensmodelle

Sequenzielle Modelle ordnen die Aktivitäten, die bei der Entwicklung durchlaufen werden, Phasen zu. Jede dieser Phasen liefert eine Dokumentation, welche als Meilenstein für den Projektfortschritt genutzt werden kann. Bei dem Übergang von einer Phase in die nächste wird vorausgesetzt, dass die Letzte abgeschlossen ist. Typische Vertreter sequenzieller Modelle sind das Phasen- und das Wasserfallmodell (s. Abbildung 4.1). Der Unterschied der beiden liegt darin, dass beim Wasserfallmodell kontrollierte Iterationen möglich sind. Solche Rückgriffe auf bereits abgeschlossene Phasen sind aber nur in Ausnahmefällen erlaubt, um Fehler zu beseitigen. Ein Problem bei sequenziellen Modellen ist die „Perfect Understanding Assumption“, also die Annahme, man könne zu Beginn des Entwicklungsprozesses, also in der Analysephase, alle Anforderungen des Kunden aufnehmen. In der Praxis wird das Wasserfallmodell trotzdem häufig verwendet, da durch die Einteilung in Phasen direkt auf einen Auslieferungstermin hingearbeitet werden kann. Ein weiteres Problem ist, dass bei strikter Einhaltung der Phasen keine fertige Teillösung entsteht. Es kann also im Fall einer Verzögerung keine Teilimplementierung ausgeliefert werden. [53] [17]

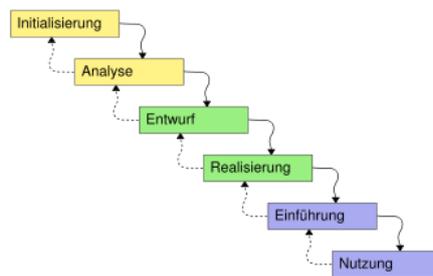


Abbildung 4.1.: Wasserfallmodell mit Rücksprungmöglichkeiten [24]

4.2. Agile Softwareentwicklung

Bei der Agilen Softwareentwicklung wird Agilität verwendet. Das Ziel liegt darin, den Entwicklungsprozess schlanker und flexibler zu gestalten. Diese Art der Softwareentwicklung stellt einen Gegentrend zu den immer größer und komplexer werdenden Vorgehensmodellen dar. Auch durch Veröffentlichung des Agilen Manifests erlangten sie schnell großer Beliebtheit. [66] Dort werden neue Akzente gesetzt:

„Individuals and interactions over processes and tools“

„Working software over comprehensive documentation“

„Customer collaboration over contract negotiation“

„Responding to change over following a plan“ [11]

In diesem Manifest geht es um die Hervorhebung von Kommunikation, Kooperation und Erfahrung. Laut Kent Beck u. a. sind Mut und Offenheit für Änderungen wichtiger als das Befolgen eines festgelegten Plans.

Die Agile Softwareentwicklung beinhaltet außerdem Methoden, die den Entwickler kreativ arbeiten lassen. [68]

4.2.1. Extreme Programming

Extreme Programming (XP) stellt Teamarbeit, Offenheit und die stete Kommunikation zwischen allen Beteiligten in den Vordergrund. Es fördert den Teamgeist so weitreichend, dass jedem Teammitglied klar sein sollte, dass das Ziel nur als Einheit zu erreichen ist. [70]

Innerhalb des XP gibt es vier prägende Werte: Kommunikation, Einfachheit, Feedback und Courage. Kommunikation gilt als wesentlicher Teil des Projekterfolges. Einfachheit meint vor allem keine unnötige Komplexität. Das Feedback informiert die Entwickler über den Grad der

Zufriedenheit des Kunden. Courage soll zur verantwortlichen Eigeninitiative der Projektbeteiligten ermutigen. Das bedeutet, sie sollen ohne hundertprozentige Absicherung die nach ihrer Meinung für den Projekterfolg nötigen Schritte in Angriff nehmen.

Der Entwicklungsprozess besteht aus drei Teilen: Release Planung, Iterative Release sowie Akzeptanztests und Release Veröffentlichung. Zuvor werden zusammen mit dem Kunden so genannte User Stories entwickelt. Diesen wird eine entsprechende Priorität zugewiesen. Das erste Release beinhaltet den Systemkern. Nach erfolgreichen Akzeptanztests werden in kontinuierlicher Abfolge neue Releases (Systemerweiterungen) erstellt. Die Entwicklungszeit der Releases soll möglichst kurz gehalten werden. Durch den Akzeptanztest nach jedem Release bekommt das Entwicklerteam ein Feedback. Dieses Vorgehen entspricht exakt der Vorgehensweise bei evolutionärer, prototyping orientierter Softwareentwicklung (s. Kapitel 4.2.3) (s. Kapitel 4.2.4).

Der Prozess beginnt mit Komponententests, so genannten Unit Tests. Danach erst folgen Entwurf und Implementierung der jeweiligen Systemkomponente. Entwurf und Implementierung werden solange wiederholt, bis alle Tests erfolgreich durchlaufen sind. Wenn alle in Form von User Stories spezifizierten Akzeptanztests erfolgreich beendet sind, wird das jeweilige Release zur Benutzung freigegeben. Ist dies nicht der Fall, wird erneut der Teilprozess Release Planung durchlaufen. Außerdem sind im Extreme Programming auch teambildende Maßnahmen eingeplant. [17]

Zusammenfassend sind 14 charakterisierende Punkte zu nennen [9] [53]:

planning game: Es findet ein Planungsspiel statt, in dem Ziele, Leistungsmerkmale und Funktionsumfang gemeinsam von Auftraggebern und Entwicklern diskutiert, beschlossen und dokumentiert werden. Es entstehen so genannte User Stories oder Story Cards. Außerdem werden Prioritäten festgelegt.

small releases: Es werden nur kleine aber semantisch vollständige Releases mit überschaubarem Funktionszuwachs entwickelt und ausgeliefert.

methapor: Es gibt eine einfache Systemvision. So wird unnütze Komplexität vermieden.

simple design: Der Entwurf und die Implementation der Releases wird von unnötiger und nicht zweifelsfrei erforderlicher Komplexität sofort befreit. Es wird nur das aktuelle Problem gelöst, keine Konzeption auf Vorrat gemacht.

testing: Eine Entwicklung beginnt mit dem Schreiben von Tests, und es wird kontinuierlich getestet.

refactoring: Entwurf und Implementierung werden ständig überarbeitet und verbessert.

pair programming: Unter dem Aspekt, dass zwei mehr sehen, wird in Paaren gearbeitet. Somit wird jede Codezeile von zwei Programmierern entwickelt und verantwortet. Die Tastatur ist an den anderen abzugeben, sobald dieser einen Fehler sieht oder eine bessere Idee hat. Wichtig ist an dieser Stelle außerdem, dass immer nur eine Sache zur Zeit bearbeitet wird. Dies kann durch den Einsatz einer ToDo-Liste erreicht werden. Der Nutzen hinsichtlich der Wissensicherung und Qualitätssteigerung wird den Mehraufwand übertreffen.

collective code ownership: Jeder im Team übernimmt Verantwortung für das gesamte System. Dies gilt auch dann, wenn der einzelne es nicht gänzlich kennt. Wer eine Verbesserungsmöglichkeit am System sieht, darf und soll diese jederzeit umsetzen.

continuous integration: Der Integrationsprozess ist ein kontinuierlicher, kein punktueller Prozess. Der Code wird sofort getestet und integriert.

40-hour-week: Es wird nicht mehr als 40 Stunden in der Woche gearbeitet. Wenn diese Grenze doch einmal überschritten wird, darf dies in der darauffolgenden Woche nicht wieder geschehen, denn persönliche Leistungsfähigkeit setzt innere Ausgeglichenheit voraus.

on-site customer: Dem Entwicklerteam gehören ein oder mehrere Mitarbeiter des Auftraggebers an. Diese stehen den Entwicklern jederzeit für Fragen zur Verfügung.

coding-standards: Zur Förderung der Kommunikation und Steigerung der Effizienz werden die Programme nach einheitlichen Richtlinien und Programmierstandards entwickelt, die das Team selbst festlegt. Durch pair programming, refactoring und collective ownership sind in einem XP-Projekt mehrere Code-Stile nicht akzeptabel.

short documentation: Die Dokumentation wird klein gehalten und es wird nicht nachträglich dokumentiert. Ein wichtiger Aspekt ist, dass die Dokumentation unempfindlich gegenüber Systemänderungen ist.

Extreme Programming ist für kleinere Projekte gut geeignet. Das Grundprinzip ist die so genannte Leitgewichtigkeit und die rasche Erstellung eines ausführbaren Systems. Kurze Entwicklungszeiten sind gut für die Vorhersage der Entwicklungszeit. [17]

Trotz der Einfachheit und nahezu generellen Anwendbarkeit ist XP kein „silver bullet“. Die Entwicklung großer Systeme mit verteilter Entwicklung erfordern Maßnahmen zur Beherrschung der Komplexität. Vor allem der Aspekt der kollektiven Codeverantwortung inklusive der Änderungsmöglichkeiten ist hier problematisch. [53]

Scrum

Scrum versucht schwergewichtige sequentielle Vorgehensmodelle und die agile Vorgehensweise zu verbinden. Es ähnelt dem Spiralmodell (s. Kapitel 4.2.3). Scrum bietet einen Werkzeugkasten von Prozessen und Methoden im Rahmen eines agilen Vorgehensmodells. Die Selbstorganisation des Entwicklerteams und deren eigenverantwortliche Auswahl der eingesetzten Mittel stehen im Vordergrund.

Dieses Vorgehensmodell gibt nur den Entwicklungsrahmen vor, es macht also wenige und abstrakte Vorgaben. Ziel ist es, ein Produkt zu entwickeln, das die Wünsche des Kunden zur Auslieferungzeit reflektiert und auf Ballast verzichtet. [17]

Drei Phasen werden wiederholt angewendet: Pre-Sprint (Planung Vision), Sprint (Software-Entwicklung und deren Überwachung) und Post-Sprint (Review mit Feedback)

4.2.2. Testgetriebene Entwicklung

Eine testgetriebene Entwicklung ist Teil vieler agiler Methoden. Sie ist auch ein wesentlicher Bestandteil des Extreme Programming (XP). Häufig tritt sie in Verbindung mit Pair Programming auf.

Bei einer testgetriebenen Entwicklung geht es darum, vor der Konzeption der einzelnen Komponenten Tests zu erstellen. Diese Tests können dann automatisiert ablaufen, um jederzeit zu testen, ob das System die gewünschte Funktionalität bereitstellt. Regressionstests nach einer Fehlerbehebung sind wichtig, um zu prüfen, ob der Fehler behoben wurde oder Neue entstanden sind. Neben den Regressionstests sind auch Usability-, Performanz- und Komponententests wichtige Bestandteile einer testgetriebenen Entwicklung. Für deren Umsetzung sind häufig Mock-Objekte nötig. Diese dienen als Stellvertreter für bestimmte Teile des

Systems während der Tests. Die Vorteile eines testgetriebenen Vorgehens sind, dass der Programmcode in der Regel stärker modularisiert ist, da aus Entwicklersicht das System aus kleinen Arbeitseinheiten besteht, die unabhängig voneinander geschrieben und getestet werden können. So werden, weil kleinere Softwareeinheiten entstehen, eine lose Kopplung und schlanke Schnittstellen erreicht. [10] [66] [19] [72]

4.2.3. Evolutionäre Arbeitsweise

In der evolutionären Entwicklung wird eine zyklische Anordnung der Entwicklungsschritte vorgenommen, und Rückgriffe werden eingeplant. Anforderungen werden während des Entwicklungsprozesses erschlossen, da davon ausgegangen wird, dass diese sich ändern können. Fehler gelten als Lernereignisse. [22]

Dieser Ansatz ähnelt dem im Design beschriebenen Vorgehen, da auch dort Fehler als Teil des Lernprozesses gelten und Anforderungen sich jederzeit ändern können (s. Kapitel 3.4.2).

Ein Beispiel für die evolutionäre Arbeitsweise ist das Spiralmodell von Boehm. Es besteht aus vier Quadranten, die spiralförmig wiederholt durchlaufen werden. Im ersten Quadranten werden Ziele identifiziert, Alternativen festgelegt und Einschränkungen bestimmt. Im zweiten werden Alternativen bewertet und Risikoquellen aufgedeckt sowie Maßnahmen zu deren Beherrschung definiert. Innerhalb des dritten Quadranten wird das System auf der jeweiligen Stufe entwickelt und abgenommen. Im vierten wird die nächste Stufe geplant. Erkenntnisse innerhalb des Prozesses können in diesem Modell berücksichtigt werden. Es betrachtet die Softwareentwicklung als risikogetriebenen Prozess und sieht deshalb eine Risikoanalyse vor. Jeder Zyklus endet mit einer Überprüfung des Systems, bei dem auch der Kunde beteiligt ist. Erst nach erfolgreicher Abnahme wird der nächste Schritt geplant. Das Hauptziel ist hier, dass sich alle Beteiligten über das Geleistete und das im nächsten Schritt Geplante einig sind. Das Projekt kann außerdem in Teilprojekten bearbeitet werden, die in parallel durchzuführende Spiralzyklen münden (s. Abbildung 4.2).

Das Spiralmodell kann sowohl für die Entwicklung als auch für die Wartung von Software eingesetzt werden. Am Anfang steht die Annahme, dass ein bestimmtes Aufgabenfeld durch den Einsatz oder die Anpassung von Software sinnvoll bewältigt werden kann. Durch den spiralförmigen Entwicklungsprozess wird diese Hypothese geprüft und am Ende widerlegt, oder es ist ein fertiges Produkt entstanden. [53] [17]

Dieses Modell eignet sich für den Einsatz von Qualitätssicherungsmaßnahmen. Außerdem erkennt es Fehler so frühzeitig, dass schnell Lösungsalternativen entwickelt werden können. Ein Nachteil liegt darin, dass Zeit- und Kostenpläne am Anfang nur schwer zu erstellen sind. [53]

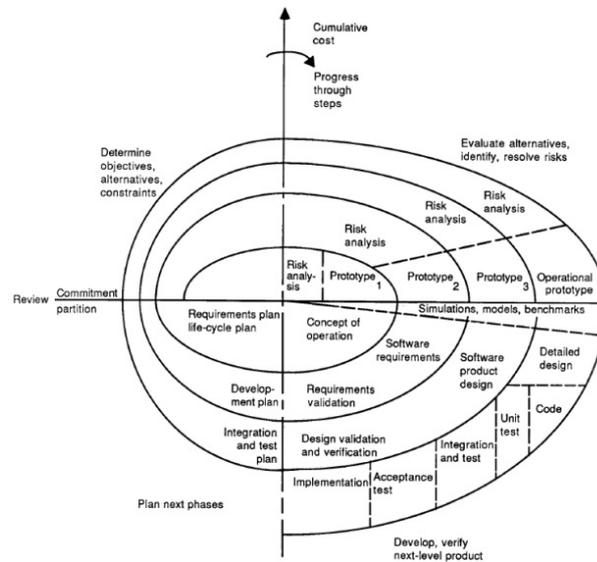


Abbildung 4.2.: Spiralmodell Boehm [41]

4.2.4. Prototyping

Prototyping wird unter anderem zum Testen verwendet. In einer phasenorientierten Entwicklung, wie zum Beispiel beim Wasserfallmodell (s. Kapitel 4.1), wird linear ein Schritt nach dem anderen abgearbeitet. Beim Prototyping wird ein Feedback und eine Kommunikation, also ein Autor-Kritiker-Zyklus, hinzugefügt. [27]

Besonders das evolutionäre Prototyping hat sich in der Praxis bewährt. Es ist ein kontinuierliches Verfahren, das sich an schnell verändernde Rahmenbedingungen anpasst. [21]

Prototypen dienen allen Beteiligten zum Experimentieren und zum Sammeln von Erfahrungen. Außerdem können sie als eine Diskussions- und Entscheidungsgrundlage verwendet werden. Sie schaffen somit eine Kommunikationsgrundlage für alle am Entwicklungsprozess beteiligten Gruppen. [21]

4.3. Prüfung der Anwendbarkeit innerhalb der Zusammenarbeit

Die in der Informatik bewährten Methoden können in einer interdisziplinären Zusammenarbeit nicht unmittelbar umgesetzt werden, da diese Modelle auf die Zufriedenheit des Kunden ausgelegt sind, die es hier jedoch nicht gibt. Die Qualität innerhalb solcher Modelle ist aber

nur dann messbar, wenn auch die gleichen Ziele verfolgt werden. Ziele wie Reproduzierbarkeit, Vorhersagbarkeit und Transparenz sind möglicherweise jedoch gar nicht erwünscht. Ein Beispiel wäre, ein Objekt zu entwickeln, dessen Interaktion aus bewusster Intransparenz der Bedienbarkeit besteht. Es könnte dadurch eine Art Spiel entstehen, das den Benutzer durcheinanderbringen soll.

4.4. Anforderungen an ein Kunstprojekt

Die von Martin Sukale beschriebenen Anforderungen an ein Kunstprojekt [63], welche bereits Sebastian Gregor am Beispiel Ambient Awareness [30] untersucht hat, werden an dieser Stelle aufgegriffen. [63] [62]

*„**Flexibilität** der künstlerische Entwurf soll so frei wie möglich, bestenfalls ohne technische Vorgaben entstehen können*

***Wiederverwendbarkeit** bei temporären Ausstellungen soll die Hardware im Folgeprojekt weiterhin einsetzbar sein*

***Skalierbarkeit** um gestalterische Konzepte fortspinnen, verbessern zu können sollen spontan beliebig Komponenten hinzugefügt werden können*

***Bedienbarkeit** die Projektteilnehmer haben evtl. wenig technisches Vorwissen, es muss möglich sein, eine Installation intuitiv und einfach „zusammenzustecken“, Plug- And-Play*

***Integrierbarkeit** die Komponenten müssen evtl. im Ausstellungskontext unsichtbar werden, oder z.B. tragbar sei*

***Autonomie** die Komponenten müssen ohne Anwesenheit eines Technikers „Stand- Alone“ lauffähig sein, ohne User-Interface eingaben starten - z.B. bei Installationen im öffentlichen Raum und über längere Zeit*

***Zuverlässigkeit** Ausfälle der Technik beim Aufbau, Eröffnung und Betrieb von Kunstausstellungen sind teuer und schlecht für die Reputation*

***Low-Cost** kreative Ideen sollen unabhängig von einem großem finanziellen Budget z.B. im Zusammenhang von studentischen Projekten realisierbar sein, vorgestellte Lösungen werden sich nur durchsetzen, wenn diese auch kostengünstig einsetzbar sind“ [63]*

4.4.1. Untersuchung der Anforderungen an Hand der Vorstudien

Flexibilität: Gewünscht ist Flexibilität vor allem auf Seiten der Designer. Bestenfalls sollen alle ihrer Ideen technisch umsetzbar sein. Für die erfolgreiche Zusammenarbeit in einem interdisziplinären Projekt zwischen Informatik und Design muss an dieser Stelle ein evolutionärer Entwicklungsprozess beginnen. Wenn ganz frei von technischen Vorgaben Ideen entwickelt wurden, hatte dies in den Projekten mit Studierenden häufig Enttäuschungen zur Folge. Ein gewisses Grundverständnis sollte vorhanden sein oder gemeinsam erarbeitet werden.

Wiederverwendbarkeit: Für Kursprojekte ist es sehr wünschenswert, die Hardware komplett wiederverwenden zu können. Die Kosten könnten so gering gehalten werden. Die Kurse in den Vorstudien haben allerdings bewiesen, dass nicht alle Hardware wiederverwendbar ist. Kleinteile gehen leicht verloren. Der Verschleiß der Einzelteile ist durch den Ein- und Ausbau in die Objekte sehr groß. Sowohl Stoffe als auch Kabel sind nur begrenzt wiederverwendbar.

Skalierbarkeit: Für einen evolutionären Entwicklungsprozess ist es unerlässlich, stets einzelne Komponenten erweitern zu können. Die Ideen sind sehr schnelllebig und verändern sich häufig. Es ist wichtig, immer wieder durch Hinzufügen neuer Komponenten frische Ideen anzustoßen und das Produkt zu verbessern.

Bedienbarkeit: In den Vorstudien gab es an dieser Stelle häufig Probleme. Der Transport von Teilen ist für die Entwicklung in der Vergangenheit unerlässlich gewesen. Die Designer mussten ihre Kleidungsstücke mitnehmen, um auf Puppen daran arbeiten zu können. Solche Puppen stehen in Informatiklaboren nicht zur Verfügung. In dem Projekt Helene All (s. Kapitel [3.5.2](#)) wurden Servomotoren angesteuert, um die Kuppel hoch- und runterfahren zu können. Ein Problem dabei war, dass die Servomotoren überdrehten und zerstört wurden, wenn sie sich nicht in der richtigen Stellung im Verhältnis zu dem Gestell befanden. Die Bedienbarkeit war also nicht von Anfang an in allen Projekten Grundvoraussetzung.

Integrierbarkeit: Im Bereich des Wearable Computing ist die Integrierbarkeit essentiell. Eine wichtige Forderung der Designer bestand darin, dass die Hardware nicht sichtbar sein soll. Vor allem Kabel störten sehr. Der Widerstand des elektronisch leitfähigen Garns war zu hoch, um dieses zu verwenden (s. Kapitel [3.5](#)). In der Zwischenzeit sind aber schon neue Arten von leitfähigem Garn auf dem Markt und müssen getestet werden.

Autonomie: Ausstellungen dauern zumeist mehrere Tage. Ein Beispiel ist die Ausstellung Romantische Maschinen (s. Kapitel 2.3.3) im Gämeviertel. Hier musste die Technik längere Zeit und auch ohne die Anwesenheit eines Technikers lauffähig sein.

Zuverlässigkeit: In den Kursen kam es zu großen Enttäuschungen, wenn in der Abschlusspräsentation ein Wackelkontakt oder ein Kurzschluss dafür sorgte, dass das Objekt nicht funktioniert. Des Weiteren wurden einige Ergebnisse in den Schulen präsentiert, ohne dass Kursleiter anwesend waren, die im Notfall hätten helfen können.

Low-Cost: Die Erfahrungen mit den Kursen haben gezeigt, dass sehr darauf geachtet werden muss, welche Hardware verwendet wird und ob diese zumindest teilweise wiederverwendbar ist. Die Schule wollte keine Materialkosten auf sich nehmen. In den Projekten mit Studierenden spielten die Kosten ebenfalls eine große Rolle, da Abschlussarbeiten im Designdepartment auf eigene Kosten finanziert werden müssen.

4.4.2. Anforderungen an ein interdisziplinäres Projekt im Bereich Interactive Design

Die beschriebenen Anforderungen an ein Kunstprojekt werden im Folgenden anhand der Vorstudien erweitert, da diese Arbeit nicht von einem reinen Kunstprojekt handelt, sondern von der interdisziplinären Zusammenarbeit von Informatik und Design. Hierfür werden Indikatoren vorgestellt, die auf ein gelungenes Projekt hindeuten. Auch der Aspekt der gelungenen Kooperation muss dabei berücksichtigt werden:

„Ein Indiz für die gelungene Kooperation ist das gegenseitige Interesse an den fachfremden Themengebieten.“ [40]

Indikatoren für ein gelungenes Projekt in einer solchen Zusammenarbeit sind:

1. Am Ende muss ein Objekt abgeschlossen sein.
2. Mit dem Produkt müssen alle Teilnehmer zufrieden sein.
3. Kein Vertreter einer Disziplin darf das Gefühl haben, vernachlässigt worden zu sein.

Folgende Aspekte sind bei der erfolgreichen Durchführung einer interdisziplinären Zusammenarbeit von Informatik und Design zusätzlich zu den eben beschriebenen Anforderungen nötig:

Soft Skills: Es erfordert ein hohes Maß an Soft Skills, um erfolgreich ein interdisziplinäres Projekt zwischen Design und Informatik umzusetzen. Viele der bereits erwähnten Probleme (s. Kapitel 3) lassen sich nur so aus dem Weg schaffen.

Es ist wichtig, dass die Bereitschaft eines jeden Teilnehmers vorhanden ist, sich auf eine aus dem Studium wenig bekannte Art der Projektarbeit einzulassen. Hierfür ist es nötig, Verständnis für die Arbeitsweise des anderen Fachbereichs zu haben.

Informatiker müssen lernen, sich mehr einzubringen und Kreativität aufzubringen. Sie müssen mehr Mitbestimmung fordern, um ihre Motivation zu stärken. Designer müssen teamfähiger werden und den Informatiker mehr einbeziehen. Beide müssen lernen, sich für den Gegenüber verständlich auszudrücken. Außerdem muss ihnen klar sein, dass das Ziel nur gemeinsam erreichbar ist. Jeder muss das Gefühl haben ein Teil des Ganzen und ersatzlich zu sein.

Gemeinsame Ideenfindung: Es ist wichtig, dass die Idee von allen gemeinsam entwickelt wird. Informatiker legen bei den Ideen viel Wert auf Funktionalität, während für das Design eher das Äußere und die Idee selbst eine entscheidende Rolle spielt. Eine Lösung muss kooperativ erarbeitet werden. Nur so sind beide Parteien motiviert, sich auf eine solche zum Teil schwierige Zusammenarbeit einzulassen.

Gemeinsame Sprache finden: Für die Etablierung einer Zusammenarbeit muss eine gemeinsame Sprache entwickelt werden. Wie bereits erläutert haben Informatiker Schwierigkeiten Skizzenbücher und Designer UML zu verstehen. Es müssen also andere Wege der Verständigung gefunden werden (s. Kapitel 3.4.1). Eine gute Möglichkeit ist sich die Zeit zu nehmen dem Anderen die Sprache zu erklären.

Bei der Diskussion über Entwürfe und für eine Dokumentation müssen für alle Beteiligten verständliche Darstellungsformen gefunden werden. Eine Möglichkeit für eine solche Dokumentation ist ein Blog. Dieser sollte für alle Teilnehmer zugänglich sein. Des Weiteren ist es wichtig, dass er jederzeit vollständig ist. Dazu zählen die Idee, der Code, Skizzen, UML-Diagramme, eine textuelle Beschreibung und Fotos.

Rollen klären: Es ist wichtig, zu Beginn eines solchen Projektes darüber zu sprechen, wer welche Rolle im Team einnimmt. So können Missverständnisse, falsche Hoffnungen und Erwartungen minimiert werden. Vor allem ist ein Teamleiter zu bestimmen. Außerdem muss sich jemand für die Bestellung der Hardware verantwortlich fühlen.

Teambildende Maßnahmen: Ausflüge wie zum Beispiel zu Messen sind sehr förderlich für die Kommunikation und für die Gruppendynamik. Außerdem wird das Verständnis für den anderen Bereich verbessert und die Motivation erhöht. In einer solchen Atmosphäre wird es einfacher Konflikte anzusprechen und zu klären. Das Verständnis des Informatikers kann an dieser Stelle sensibilisiert und Interesse geweckt werden, da auch über Gesehenes direkt diskutiert werden kann.

Pair programming: Die Partner im Pairprogramming bestehen an dieser Stelle nicht aus zwei Softwareentwicklern, sondern aus einem Designer und einem Informatiker.

Die Entwicklung des Objektes sollte in direkter Zusammenarbeit entstehen. Äußerst schwierig gestaltet sich dies in den Disziplinen, die nur von einem der beiden Partner beherrscht werden. Dem Designer wird schnell langweilig, sobald der Programmcode geschrieben werden muss. Allerdings ist seine Anwesenheit hier äußerst wichtig. Denn wenn er den Wunsch äußert, dass die LEDs blinken, hat er eine gewisse Vorstellung, wie dies auszusehen hat. Der Informatiker geht hingegen pragmatisch vor und lässt sie einfach blinken. Es ist also wichtig, dass der Designer anwesend ist, um direkt einzugreifen und Tests durchzuführen, ob das Umgesetzte seinen Vorstellungen entspricht.

Der Designer braucht ein geräumiges und gut ausgestattetes Atelier, um seiner Kreativität uneingeschränkt freien Lauf lassen zu können. Die bisherige Zusammenarbeit fand in Informatiklaboren statt, in denen den Designern nicht ausreichend Möglichkeiten zur Entwicklung zur Verfügung standen. Die Designer mussten also ihre Objekte häufig transportieren. Probleme bestehen an dieser Stelle darin, dass der Transport gefährlich ist und immer wieder zur Zerstörung von Objekten führte (s. Kapitel 3.5). Außerdem arbeitet der Designer in diesem Fall allein mit eigenen Vorstellungen, wie die Technik am Ende aussieht, und versucht dann, ausreichend Platz dafür zu lassen. Kleinste Änderungen im technischen Material führen dazu, dass beides nicht mehr zusammenpasst. Auch bei der designerischen Umsetzung ist also die Anwesenheit des Informatikers unerlässlich, um Fehlkonstruktionen zu vermeiden. Dies gilt nur dann wenn nicht gemeinsam in einem Atelier gearbeitet wird.

Klar definierte Schnittstellen würden dafür sorgen, dass eine getrennte Entwicklung möglich wird. Eine solche Definition gestaltet sich aber aufgrund der Arbeitsweise der Designer und der Einschränkungen der Technik als schwierig.

Teammeetings: Es ist wichtig, angemessen oft Teammeetings abzuhalten, damit verhindert wird, dass allzu lange aneinander vorbeigearbeitet wird. Auf diesen müssen die Ziele aller Teilnehmer verglichen werden. Nur so kann verhindert werden, dass an verschiedenen Zielen bzw. mit unterschiedlichen Vorstellungen gearbeitet wird. Das unter Pair Programming (s. Kapitel 4.4.2) beschriebene gemeinsame Arbeiten schließt solche Teammeetings jedoch

nicht aus, denn auch wenn gemeinsam entwickelt wird, ist es wichtig, sich regelmäßig zusammenzusetzen und über Ziele, Fortschritte und das weitere Vorgehen zu sprechen.

Evolutionäre Entwicklung: Einige Probleme, die in der Zusammenarbeit auftreten können, sind durch eine evolutionäre Arbeitsweise vermeidbar. Das heißt, es entsteht nur dann ein Produkt mit dem beide zufrieden sind, wenn sie den Mut und die Motivation haben, erneut von vorn anzufangen, falls sich herausstellt, dass die Vorstellungen und technischen Möglichkeiten nicht zusammenpassen. Die Idee darf also nicht einfach verworfen, sondern es muss gemeinsam an der Umsetzung gearbeitet werden. Hierfür empfiehlt sich evolutionäres Prototyping (s. Kapitel 4.2.4).

Prototyping: Prototypen erwiesen sich immer wieder als hilfreich. Die daraus entstehenden Diskussionen sind auch für die interdisziplinäre Zusammenarbeit von Design und Informatik unerlässlich. Problematisch ist dies aber aufgrund der damit verbundenen Kosten. Die Materialien für einen Prototypen sind nicht immer wiederverwendbar. Es bedarf deshalb noch weiterer Vorstudien, um hier Lösungen zu finden, indem zum Beispiel billigere Materialien anstatt der echten verwendet werden oder Tools wie CAD zur Planung genutzt werden. An dieser Stelle soll die Kostenfrage nicht unberücksichtigt bleiben, denn während der vorliegenden Vorstudien haben sich Prototypen als unerlässlich erwiesen.

Testgetriebene Entwicklung: Eine testgetriebene Entwicklung nach bekannten Methoden ist bisher nicht umsetzbar. Es geht schließlich nicht nur um Software, sondern auch um Hardware. Die Hard- und Softwaretests kombinieren zu können, stellt einen wichtigen Fortschritt dar. JTag Tool (Joint Test Action Group) IEEE Standard 1149.1, ist eine Möglichkeit zum Hardwaredebuggen und -testen. In der Softwareentwicklung hingegen ist JUnit ein sehr bekanntes Tool, um Testfälle für die Software zu definieren.

Bei dem JTag Tool ist es möglich herauszufinden, ob die I/Os den gewünschten Pegel haben, also die gewünschte Funktion erfüllen. Außerdem besteht die Möglichkeit, Breakpoints zu setzen und einzelne Register auszulesen. Unter Anderem sind so Kurzschlüsse erkennbar. Dies ist auf den Arduino Duemilanove nicht möglich. Nur bei dem Arduino Mega ist ein solcher Boundary Scan durchführbar.

Durch eine testgestriebene Entwicklung wird das Refaktorisieren einfacher, also Hard- und Software verbessert. Es könnte nach jedem Ein- und Umbau geprüft werden, ob die Funktionalität noch gewährleistet ist. Außerdem ist es so einfacher zu verstehen, welche Funktionalität gegeben sein muss.

Für die Zukunft ist es also wichtig, eine Kombination aus Soft- und Hardwaretests zu entwickeln, um eine testgetriebene Entwicklung vornehmen zu können.

4.5. Best Practises

4.5.1. Ablaufplan Bachelorarbeit Svenja Keune

Die Bachelorarbeit von Svenja Keune (s. Kapitel [2.6.3](#)) ist ein sehr gelungenes Beispiel für die interdisziplinäre Zusammenarbeit von Informatik und Design. Denn selbst wenn es sich hierbei um eine Bachelorarbeit im Designbereich handelt, konnten viele der gesammelten Erfahrungen erfolgreich in der Kooperation eingesetzt werden.

Das Projekt begann am 10. Mai 2010. Ab diesem Zeitpunkt waren es noch ca. acht Wochen bis zur Thesisabgabe. Ausgelegt war das Projekt für einen Zeitraum von sieben Wochen. Eine Bachelorarbeit im Department Design besteht aus einem Kreativteil, der am Ende in Form einer Abschlusspräsentation benotet wird, und einer schriftlichen Thesis, bestehend aus Konzept, Inspiration und Fotos der fertigen Arbeit. Die Vorstudien für diese Bachelorarbeit dauerten ein Semester. In dieser Zeit entstand auch die Grundidee. Die Planung sah wie folgt aus:

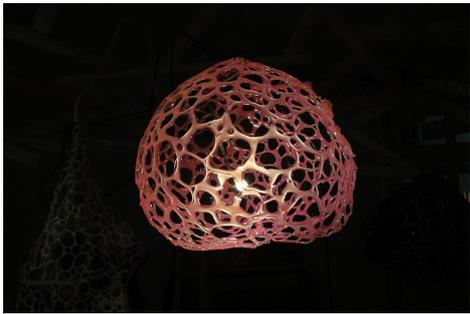
1. Woche Ein Teammeeting zur Ideenbesprechung, darauffolgend die Absprache mit den betreuenden Professoren. Die von den Professoren gewünschten Änderungen erforderte ein erneutes Teammeeting. Beginn der Anfertigung der ersten Objekte.

Meilensteine Zwei fertige Objekte (s. Abbildungen [4.4a](#) und [4.3](#)).



Abbildung 4.3.: Objekt 1

2. Woche Bearbeitung der Flächen, außerdem Abschluss der Konzeptplanung.



(a)



(b)

Abbildung 4.4.: Bachelorarbeit Svenja Keune

Meilensteine Der Plan steht bzgl. Optik, Funktion, Materialität, Elektronik und Integration.

3. Woche Ein Treffen mit einer Gruppe die ähnliche Arbeiten zur Inspiration. Erstellung 2 weiterer Objekte.

Meilensteine Fünf bis sieben fertige Objekte.

4. Woche Weitere Flächenproduktionen und die Entwicklung von Objekten mit Nitinoldraht (s. Abbildung 4.4b), Sensoren und deren Verarbeitung.

Meilensteine Technische Umsetzung der drei aufwändigsten Objekte (s. Abbildungen 4.8a, 4.8b und 4.9).

5. Woche Fertigstellung der restlichen Objekte und der Beginn der schriftlichen Thesis.

Meilensteine Fertigstellung aller Objekte.

6. Woche Präsentationsplanung bzgl. Ort und Art der Darstellung. Weitere Thesisbearbeitung und eine teambildende Maßnahme durch einen Ausflug zum Designfestival nach Berlin.



(a)



(b)

Abbildung 4.5.: Aufbau der Präsentation

Meilensteine Abschluss der Thesis.

7. Woche Aufbau der Präsentationsfläche, daraufhin ein professionelles Fotoshooting und der Druck der Thesis. Die Abbildungen [4.5a](#) und [4.5b](#) zeigen den Entstehungsprozess der Präsentationsfläche und letzte Vorbereitungen.

Meilensteine Abgabe der Thesis.

8. Woche Durchführung der Abschlusspräsentation.

4.5.2. Prüfung der Anforderungen an ein interdisziplinäres Projekt im Bereich Interactive Design

Soft Skills Durch die lange, erfolgreiche Zusammenarbeit gab es in diesem Bereich keine Probleme.

Gemeinsame Ideenfindung Aufgrund der Tatsache, dass es sich an dieser Stelle um eine Bachelorarbeit handelt, ist dieser Punkt nur begrenzt beachtet worden. Es wurde eher darüber gesprochen, welche Möglichkeiten der gemeinsamen Umsetzung bestehen.

Gemeinsame Sprache finden Es wurde während der Bearbeitungszeit ein Blog geführt, in dem sowohl Ideen als auch Fortschritte zeitnah beschrieben wurden. [39] Svenja hat sich mit Grundlagen der Programmierung und UML auseinander gesetzt und die Informatiker konnten ihr Verständnis für Skizzen schulen.

Rollen klären Die Rollenverteilung war in dieser Zusammenarbeit unproblematisch. Da es sich um ihre Bachelorarbeit handelte, übernahm Svenja selbsterklärend die Teamleitung.

Teambildende Maßnahmen Während der Vorstudien fand ein Ausflug zur Light-and-Building-Messe nach Frankfurt statt. Er diente der Inspiration, und es konnten viele neue Materialien mitgenommen werden. Das Verständnis für den anderen Bereich wurde durch viele Gespräche über das Gesehene gefördert. Der Ausflug zum Designfestival in Berlin wirkte sich ebenso positiv aus.

Pair Programming Die Zusammenarbeit wurde an dieser Stelle erfolgreich umgesetzt. Die meiste Zeit wurde im Labor oder in Svenjas Garten gemeinsam entwickelt (s. Abbildung 4.6).



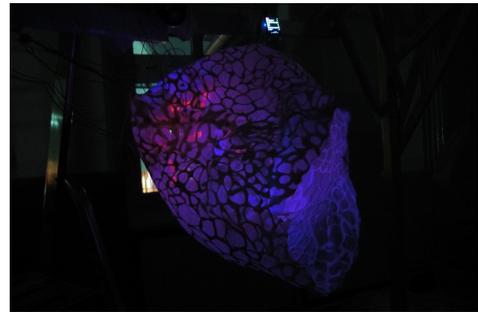
Abbildung 4.6.: Gemeinsame Entwicklung

Teammeetings Teammeetings fanden trotz der gemeinsamen Entwicklung statt und innerhalb die konnten die Ideen sich positiv weiterentwickeln. Sie waren eine große Bereicherung für das Projekt.

Evolutionäre Entwicklung Durch die lange Zusammenarbeit war die Entwicklung sehr erfolgreich, da alle von den gemeinsamen Erfahrungen profitieren konnten. Die



(a)



(b)

Abbildung 4.8.: Die Drei

technischen Möglichkeiten wurden schon während der Ideenfindung berücksichtigt.

Prototyping Das Erstellen von Prototypen gestaltete sich in der Vergangenheit als äußerst schwierig (s. Kapitel 4.2.4). Innerhalb dieses Projektes wurde deshalb das erste Mal der Versuch gemacht, zu Beginn eine 3D-Grafik für ein Objekt anzufertigen (s. Abbildung 4.7). Dieses Objekt ist in enger Teamarbeit zwischen André und Svenja entstanden. Sowohl Idee als auch Umsetzung erfolgten gemeinsam. Die Idee reifte im Laufe der Entwicklung erster Prototypen. So wurden statt nur einem gleich drei miteinander interagierende Objekte entwickelt (s. Abbildungen 4.8a, 4.8b, 4.9).

Titel: „Die Drei“:

„Die Drei“ bilden ein Gespann aus drei Objekten, die sich über die Anderen unterhalten. Die Unterhaltung wird durch Abstandssensoren ausgelöst. Wer das Morsealphabet beherrscht, könnte der Unterhaltung folgen. Entweder über Ton, oder die Lichtzeichen, die das gerade „sprechende“ Objekt gibt. Das Gespräch wird aber übersetzt und kann über ein Display verfolgt werden. Der Einsatz von Technik ist bei diesem Projekt recht komplex. Jeder der Drei hat einen Distanzsensor, mehrere LED's und einen Lautsprecher. Für die Ansteuerung der Lautsprecher war eine Schaltung mit Mosfets nötig. Für die Töne ein Waveshield, für die vielen LED's ein Arduino Mega. Auch die Programmierung war aufwändig.“ [40]

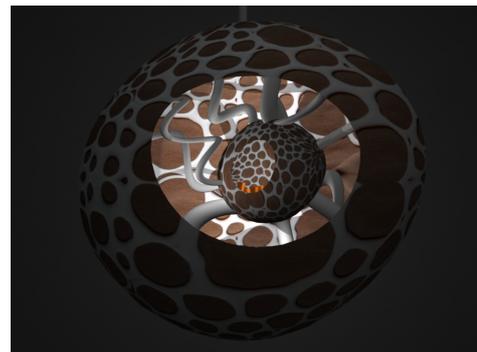


Abbildung 4.7.: 3D-Grafik erstellt in Maya



Abbildung 4.9.: Die Drei

Dass die Programmierung sehr aufwändig war, ist relativ zu betrachten. Der Zeitaufwand war höher als bei vielen bisherigen Objekten. Ausgelöst wurde dies unter Anderem durch die Tatsache, dass es sich um mehrere Objekte handelte, die kommunizieren. Die bereits beschriebenen Probleme in der Arbeit mit einer Kombination aus Soft- und Hardware, unter Anderem ein zeitintensives Debuggen, traten auch an dieser Stelle auf.



Abbildung 4.10.: Aufbau „Der Drei“ im Labor

Außerdem wird bei diesem Beispiel sichtbar, dass die Ideen sehr schnelllebig sind und durch neue, bessere Ideen abgelöst werden. Diese Art der Entwicklung muss sich also rasch veränderlichen Anforderungen anpassen. Diese Anforderungen sind nicht mit denen eines Kunden vergleichbar, sie sind keinem der Projektbeteiligten zu Beginn oder einem anderen Zeitpunkt der Entwicklung klar, eben weil sie sich stetig verändern. Die Prototypen müssen also leicht veränderlich sein, was sich äußerst schwierig gestaltet. Zum Beispiel ist es sehr zeitintensiv Kabellängen anzupassen. Die Abbildung 4.10 zeigt, dass den Testaufbau im Labor.

Testgetriebene Entwicklung Eine testgetriebene Entwicklung war aufgrund der beschriebenen Probleme nicht möglich (s. Kapitel 4.4.2).

Aufwand Die Entwicklungszeit bewegte sich in einem für eine Bachelorarbeit angemessenen Rahmen. Die komplette Bearbeitungszeit betrug acht Wochen. Jedoch beinhaltete dies neben der technischen Umsetzung noch die Konzeption, eine schriftliche Ausarbeitung und die Materialbearbeitung.

Portabilität Die Portabilität ist gegeben. Der Aufbau dauerte mit zwei Helfern ca. 3 Stunden und die Hardware ist mit dem Auto transportierbar. Bei der Aufstellung ist die Anwesenheit eines Technikers vonnöten, da der technische Aufbau sehr kompliziert ist.

4.5.3. Bewertung

Aufgrund des bereits erworbenen großen Erfahrungsschatzes konnte dieses Projekt in der Kürze der Zeit umgesetzt werden. Allerdings war die Planung etwas optimistisch terminiert. Zwar konnten viele der gesetzten Meilensteine nicht eingehalten werden, aber am Ende steht dennoch ein erfolgreich umgesetztes Projekt.

5. Zusammenfassung und Ausblick

In dieser Arbeit wurden Vorstudien zum Thema Interactive Design vorgestellt, welche durch die interdisziplinäre Zusammenarbeit der Departments Design und Informatik entstanden sind. Die Autorin dieser Arbeit war an der Durchführung unter Anderem im Bereich Projekt-Management und bei der Vorbereitung und Leitung von Kursen beteiligt. Es wurden die Vorstudien detailliert dargestellt und ausführlich bewertet. Im Anschluss wurden Anforderungen an ein Kunstprojekt sowie bewährte Methoden aus der Informatik auf Anwendbarkeit geprüft und Möglichkeiten aufgezeigt, die eine Zusammenarbeit produktiver gestalten.

Diese Arbeit zeigt welche Herausforderungen aber gleichzeitig auch Chancen eine interdisziplinäre Zusammenarbeit von Informatik und Design bieten. Im Interactive Design ist ein breites Spektrum an Fähigkeiten, von Design Aspekten über Nähen bis hin zu Elektrotechnik und Mechanik, erforderlich. Die Kooperation beider Disziplinen bietet den großen Vorteil, dass beide gleichermaßen voneinander lernen. Negative Aspekte sind unter Anderem eine schwierige Kommunikation und die damit verbundenen Missverständnisse (s. Kapitel [3.4.1](#)).

Die Kurse haben gezeigt, dass es möglich ist mit Interactive Design Anfängern spielerisch Programmierkonzepte zu vermitteln. So konnten viele Teilnehmer an diese Bereiche herangeführt werden. Als negativ erwiesen sich die geringe Motivation mancher Schüler (s. Kapitel [2.4.4](#) und [2.4.5](#)) und eine falsche Zeitplanung (s. Kapitel [2.5.1](#) und [2.5.2](#)).

Eine Herausforderung im Interactive Design ist die rasante technische Entwicklung. Um neue technische Möglichkeiten nutzen und neue Interaktionsformen ausprobieren zu können, müssen stetig kleinere und leistungsfähigere Produkte in die Kurse eingebaut werden, ein Beispiel hierfür ist der in naher Zukunft erhältliche Arduino Flexible (s. [Abbildung 5.1](#)).

Ein spannender Bereich ist die Erforschung sozialer Aspekte bei interaktiven Elementen. Scott Snibbe [\[60\]](#) zeigt dies in seinen Ausstellungen zum Beispiel durch ein interaktives Objekt, das erst ausgelöst wird, wenn mehrere Anwender involviert sind. So ergibt sich ein soziales Zusammenwirken. [\[34\]](#)

Der nächste Versuch der interdisziplinären Zusammenarbeit wird im Anschluss an diese Arbeit unternommen. In der Kooperation mit Svenja Keune sollen spannende neue Projekte umgesetzt werden. Der soziale Aspekt soll hier eine große Rolle spielen. Es sind mehrere Installationen geplant, die dem eher technisch orientierten Campus der HAW Hamburg am



Abbildung 5.1.: Arduino Flexible [52]

Berliner Tor etwas mehr Leben einhauchen und die Studierenden untereinander besser vernetzen sollen. Das Design soll hierfür Menschen anlocken und sie sowohl miteinander als auch mit der Technik interagieren lassen.

Die weitere Zusammenarbeit soll außerdem dafür sorgen, die beschriebenen Probleme weiter einzugrenzen. Die gemachten Erfahrungen innerhalb dieser Arbeit beschreiben die Schwierigkeiten bei der Schnittstellendefinition. Es gilt herauszufinden, inwieweit solche definiert werden können, um eine getrennte Entwicklung zu ermöglichen. Oder aber ob die gemeinsame Entwicklung andere Vorteile mitsichbringt (s. Kapitel 4.4.2). Die Eingrenzung der Probleme im Prototyping können möglicherweise durch den Einsatz von Tools wie zum Beispiel CAD erreicht werden (s. Kapitel 4.4.2). Desweiteren ist die Kombination von Hard- und Softwaretests ein sehr spannender Bereich, den es näher zu beleuchten gilt (s. Kapitel 4.4.2).

A. Anhang

Der Anhang hat folgenden Inhalt:

1. Ausgewählte Aufgaben
2. Ablaufplan
3. Presstext
4. Bestandsliste eines Classroomkits
5. Merkblatt

ME & MY TOASTER EDWIN



The „Funky Switch“

- **Plant mehrere Schalter**
- **Und setzt mindestens einen Schalter praktisch um**
- **denkt dabei auch an individuelle Kontaktlösungen: Metallknöpfe, Reißverschlüsse, ...**
- **Präsentiert eure Ergebnisse**
- **Zeit: 2 Std
exkl. Präsentation**

ME & MY TOASTER EDWIN



I. Aufgabe

Simple Circuit + Trash Art

- Baut den Versuchsaufbau X erneut auf
- nehmt, statt der Krokodilklemmen, isolierten Draht
- kombiniert die versch. Materialien mit dem Aufbau
- Präsentiert eure Ergebnisse
- Zeit: 1.5 Std
exkl. Präsentation

Krokodilsklemmen Schwarz:	20
Krokodilsklemmen Rot:	20
Arduino:	1
USB Hub:	1
LilyPad Akkus	3
LilyPad:	3
LilyPad Charger:	3
FTDI:	3
Xbee Shields:	3
Xbee Chips:	3
Xbee Explorer:	1
mini-USB:	2
großer-USB:	1
LilyPad RGB:	5
LilyPad LED:	30
Vibrator	2
Button	2
Lichtsensoren	2
Beschleunigungssensoren	1
Buzzer	1
Motor Drive K20WD	1
Servo Motor	1
Flexsensor	1

Toaster Edwin – Wearable Computing Workshop

Stelldichein: Technik trifft Design!

In diesem zweiwöchigen Workshop werden zunächst vernähbare Minicomputer vorgestellt, mit denen man Sensoren, LEDs und Motoren steuern kann. Anschließend werden in kleinen Gruppen eigene Projektideen zum Thema "communicate with creatures" umgesetzt.

Der Workshop ist für bis zu 20 Teilnehmer ausgelegt. Es sind keine Vorkenntnisse erforderlich. Die Teilnahme ist kostenlos.

Weitere Information unter: <http://www.toasteredwin.de>

Termin:

Der Workshop findet im Rahmen des Pentiments vom 03. bis 15.08.2009 statt.

Veranstaltungsort

HAW Hamburg, Department Design
Armgartr. 24
22087 Hamburg

Montag 03.08.2009

9:00	Einleitung	Larissa	5	min
	Präsentation: Einführung LilyPad	André + Larissa	25	min
	Übung: LilyPads verbinden	André + Larissa	60	min
10:30	Kreativaufgabe: #1	Svenja	90	min
12:00	Mittagspause / Raumwechsel 214		60	min
13:00	Kreativaufgabe: #1 präsentieren	Teilnehmer	30	min
13:30	Projektarbeit: Gruppen	Svenja + Larissa	15	min
13:45	Ablauf erklären (Prototyp)	Svenja + Larissa	15	min
14:00	Projektarbeit: Ideenfindung	Teilnehmer	60	min
15:00	Projektarbeit: Ideen vorstellen	Teilnehmer	90	min
16:30	Abschluss	Larissa	30	min
17:00	Internes Meeting	Kursleiter		

Dienstag 04.08.2009

9:00	Einleitung	Larissa	5	min
	Präsentation: Programmieren	Larissa + André	15	min
	Live-Vorführung: Programmieren	Larissa + André	10	min
	Übung: Programmieren #1	Larissa + André	30	min
	Pause		10	min
	Übung: Programmieren #2	Larissa + André	50	min
	Pause		10	min
	Präsentation: Prototypen	Robert	30	min
	Präsentation: Projektseite Homepage	Robert	30	min
12:00	Mittagspause / Raumwechsel 214		60	min
13:00	Kreativaufgabe: #1 (Vortag) programmieren	Svenja	90	min
14:30	Pause		15	min
14:45	Projektarbeit: Prototyp beginnen	Teilnehmer	45	min
15:30	Projektarbeit: Projektidee präsentieren	Teilnehmer	60	min
16:30	Abschluss	André	30	min
17:00	Internes Meeting	Kursleiter		

Mittwoch 05.08.2009

9:00	Einleitung	Larissa	5	min
9:05	Präsentation: Mode	Ekaterina	25	min
9:30	Kreativaufgabe: #2 Funky Switch	Ekaterina + Svenja	120	min
	Übung: Nähen (nebenbei)	Ekaterina		
12:00	Mittagspause / Raumwechsel 214		60	min
13:00	Kreativaufgabe: #2 Ergebnis präsentieren	Svenja	30	min
13:30	Projektarbeit: Prototyp weiterbearbeiten	Teilnehmer	150	min
16:00	Projektarbeit: Zwischenstand präsentieren	Teilnehmer	30	min
16:30	Abschluss	Svenja	30	min
17:00	Internes Meeting	Kursleiter		

Donnerstag 06.08.2009

9:00	Einleitung	Larissa	5	min
	Präsentation: Mode + Licht	Ekaterina	25	min
	Pause		10	min
9:30	Präsentation: E-Technik	Klaus	30	min
10:00	Pause		10	min
10:10	Projektarbeit: Prototyp weiterbearbeiten	Teilnehmer	110	min
	Übung: E-Technik (nebenbei)	Klaus		
12:00	Mittagspause / Raumwechsel 214		60	min

13:00	Projektarbeit: Prototyp weiterbearbeiten	Teilnehmer	150	min
15:30	Projektarbeit: Prototyp präsentieren	Teilnehmer	60	min
16:30	Abschluss	Ekaterina	30	min
17:00	Internes Meeting	Kursleiter		
Freitag 07.08.2009				
9:00	Einleitung	Larissa	5	min
	Präsentation: LilyPad Kommunikation ZigBee	André + Larissa	10	min
	Übung: LilyPad Kommunikation ZigBee	André + Larissa	35	min
	Pause		10	min
10:00	Präsentation: Arduino Duemilanove + Shield	André + Larissa	15	min
	Übung: Motor/Wave-Shield	André + Larissa	45	min
11:00	Pause		10	min
	Präsentation: Technische Textilien	Svenja	20	min
	Pause		10	min
11:30	Präsentation: Einführung Löten	Klaus	15	min
	Übung: Löten	Klaus	15	min
12:00	Mittagspause / Raumwechsel 214		60	min
13:00	Kreativaufgabe: #3 Zahnbürste	Svenja	120	min
15:00	Kreativaufgabe: #3 Ergebnis präsentieren	Teilnehmer	30	min
15:30	Grillen	Alle		
Montag 10.08.2009				
9:00	Einleitung	Larissa	5	min
	Kreativaufgabe: #4 Kommunikation	Svenja	175	min
12:00	Mittagspause		60	min
13:00	Kreativaufgabe: #4 Ergebnis präsentieren	Teilnehmer	60	min
14:00	Projektarbeit	Teilnehmer	90	min
15:30	Projektarbeit: Zwischenpräsentation + Probleme	Teilnehmer	60	min
16:30	Abschluss	Larissa	30	min
17:00	Internes Meeting	Kursleiter		
Dienstag 11.08.2009				
9:00	Einleitung	Larissa	5	min
9:05	Präsentation: Videos interaktive Projekte #1	Svenja	20	min
9:30	Projektarbeit	Teilnehmer	175	min
12:00	Mittagspause		60	min
13:00	Projektarbeit	Teilnehmer	210	min
16:30	Abschluss	André	30	min
17:00	Internes Meeting	Kursleiter		
Mittwoch 12.08.2009				
9:00	Einleitung	Larissa	5	min
9:05	Präsentation: Videos interaktive Projekte #2	Svenja	20	min
9:25	Projektarbeit	Teilnehmer	175	min
12:00	Mittagspause		60	min
13:00	Projektarbeit	Teilnehmer	150	min
15:30	Projektarbeit: Zwischenpräsentation + Probleme	Teilnehmer	60	min
16:30	Abschluss	Ekaterina	30	min
17:00	Internes Meeting	Kursleiter		

Donnerstag 13.08.2009

9:00	Einleitung	Larissa	5	min
9:05	Präsentation: Ambient Awareness (Videos)	Svenja	20	min
9:05	Projektarbeit	Teilnehmer	175	min
12:00	Mittagspause		60	min
13:00	Generalprobe	Teilnehmer	210	min
16:30	Abschluss	Svenja	30	min
17:00	Internes Meeting	Kursleiter		

Freitag 14.08.2009

9:00	Einleitung	Larissa	5	min
	Projektarbeit	Teilnehmer	175	min
12:00	Mittagspause		60	min
13:00	Projektarbeit	Teilnehmer	210	min
16:00	Abschluss	Larissa	30	min

Samstag 15.08.2009

14:00	Ausstellung	Teilnehmer		
20:00	Party	Alle		

Nicht vergessen:

- Jeweiligen Tagesablauf an die Tafel schreiben.
- Raum ankündigen



Toaster Edwin

Merkblatt

07.12.2009



Arduino- IDE: Voreinstellungen

1. Arduino-Board mit PC verbinden
2. Board auswählen: *Tools -> Board -> LilyPad oder Duemilanove with ATmega328*
3. Serial Port auswählen: *Tools -> Serial Port -> COM mit der höchsten Nummer*



Arduino-IDE: Nutzung



- Nach jeder Änderung des Programms neu hochladen: *File -> Upload to I/O Board*



- Werte oder Text anzeigen:
 1. `Serial.begin(9600);` // steht im `setup()`
 2. `Serial.println(wert);`
 3. „Serial Monitor“- Button drücken



Programmaufbau

```
// Dies ist ein Kommentar, wird ignoriert, eingeleitet durch zwei //
1. Variablen | int ledPin = 13; // Variable ledPin, um Pin 13 (LED auf dem Arduino) zu benutzen.
              |
              | // setup() wird als erstes einmal ausgeführt.
              | void setup() {
              |     Serial.begin(9600); // Serial-Monitor aktivieren, um Werte ausgeben zu können.
2. setup()   |     pinMode(ledPin, OUTPUT); // Nötig, weil ledPin in digitalWrite auftaucht.
              |     digitalWrite(ledPin, HIGH); // Schalte LED an (Pin 13 -> LED auf dem Arduino).
              | }
              |
              | // loop() wird nach setup() unendlich wiederholt. Hier steht, was das Programm tut.
              | void loop() {
3. loop()    |     Serial.println("Hallo Toaster Edwin!"); // Text im Serial-Monitor anzeigen.
              | }
              |
```

Wichtige Befehle (Funktionen)

`digitalWrite(pin, HIGH);` ; -----

Schalte an, was am Pin hängt: LED, Vibrationsmotor, Grußkarte...

`digitalWrite(pin, LOW);` ; -----

Schalte aus, was am Pin hängt.

`analogWrite(pin, wert);` ; -----

Dimme, was am Pin hängt: LED, Motor, Buzzer...

```
// Beginne bei 0 und erhöhe jeden Durchlauf dimmWert um Eins bis 255.
for(int dimmWert = 0; dimmWert <= 255; dimmWert += 1) {
    analogWrite(ledPin, dimmWert); // Schreibe neuen dimmWert auf ledPin.
    delay(10); // Kurz warten (10ms), damit die Helligkeitsstufen sichtbar werden.
}
```

`analogRead(sensorPin);` ; -----

Werte vom Pin einlesen, an dem ein Sensor angeschlossen ist.

Sensoren sind zum Beispiel: Licht-, Distanz-, Temperatur-, Bewegungssensor...

```
int sensorWert = analogRead(sensorPin); // sensorPin lesen und Wert speichern.
if(sensorWert > 100) { // Wenn sensorWert größer 100...
    digitalWrite(ledPin, HIGH); // ...dann LED an
} else {
    digitalWrite(ledPin, LOW); // ...sonst LED aus
}
```

`Serial.println(wert);` ; -----

Anzeigen eines Wertes (besonders aus Variablen) im Serial-Monitor.

`delay(1000);` ; -----

1000 Millisekunden (= 1 Sekunde) warten.

Im `setup()` nicht vergessen:
`pinMode(pin, OUTPUT);`

Nur PWM-Pins: 3, 5, 6, 9, 10, 11
Werte von 0 bis 255 möglich.

Nur Pins: a0, a1, a2, a3, a4, a5

Im `setup()` nicht vergessen:
`Serial.begin(9600);`

In dieser Zeit ist keine Reaktion
auf neue Sensorwerte möglich.

Wichtige Links

- <http://www.toasteredwin.de>
- <http://web.media.mit.edu/~leah/LilyPad>
- <http://arduino.cc/en/Reference>

Literaturverzeichnis

- [1] ZigBee Alliance. <http://www.zigbee.org/>. Webseite. Abruf: 03.07.2010, 14:02 Uhr.
- [2] Atmel. http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf. Webseite. Abruf: 03.07.2010, 15:39 Uhr.
- [3] Atmel. http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf. Webseite. Abruf: 03.07.2010, 15:40 Uhr.
- [4] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martin, and David Mellis. <http://arduino.cc/en/Main/ArduinoBoardDuemilanove>. Webseite. Abruf: 03.07.2010, 14:20 Uhr.
- [5] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martin, and David Mellis. <http://arduino.cc/en/main/arduinoboardlilypad>. Webseite. Abruf: 17.06.2010, 11:46 Uhr.
- [6] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martin, and David Mellis. <http://arduino.cc/en/uploads/main/arduinoduemilanove.jpg>. Webseite. Abruf: 22.7.2010, 22:31 Uhr.
- [7] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martin, and David Mellis. <http://www.arduino.cc/>. Webseite. Abruf: 04.06.2010, 11:46 Uhr.
- [8] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martin, and David Mellis. <http://www.arduino.cc/en/Main/ArduinoBoardProMini>. Webseite. Abruf: 16.07.2010, 14:20 Uhr.
- [9] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison Wesley, 2000.
- [10] Kent Beck. *Test Driven Development by Example*. Addison Wesley, 2002.
- [11] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian MarickRobert, C. Martin, Steve Mellor, Ken Schwaber Jeff Sutherland, and Dave Thomas. <http://agilemanifesto.org/>. Webseite. Abruf: 24.06.2010, 11:00 Uhr.
- [12] L.J. Bottomley, S. Rajala, and R. Porter. Engineering outreach teams: K-12 outreach at north carolina state university. In *Frontiers in Education Conference, 1999. FIE '99. 29th Annual*, volume 3, pages 13A7/14 –13A7/17 vol.3, 1999.

- [13] Frederick P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison Wesley, 1995.
- [14] Leah Buechley. <http://web.media.mit.edu/leah/>. Webseite. Abruf: 25.06.2010, 14:20 Uhr.
- [15] Leah Buechley. <http://web.media.mit.edu/leah/lilypad/>. Webseite. Abruf: 09.06.2010, 20:21 Uhr.
- [16] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 423–432, New York, NY, USA, 2008. ACM.
- [17] Christian Bunse and Antje von Knethen. *Vorgehensmodelle kompakt*. Spektrum, 2008.
- [18] Florian Burka. Ambient Awareness - Federfarbener Tunnel. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2009. Abruf: 08.06.2010, 15:15 Uhr.
- [19] Bettina Buth. QA und VVT. Folien, Hochschule für Angewandte Wissenschaften Hamburg, <https://pub.informatik.haw-hamburg.de/home/pub/prof/buth/Master-Anwendungen-1/SE-QA-VVT-slides.pdf>, 2010. Abruf: 24.06.2010, 19:50 Uhr.
- [20] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
- [21] Julian Mack Christiane Floyd, Guido Gryczan. Einführung in die Softwaretechnik Lernen und Prototyping in der Softwareentwicklung. Paper, Universität Hamburg, <http://www.inf.fu-berlin.de/lehre/WS01/SWT/Prototyping.pdf>, 2010. Abruf: 11.06.2010,20:50 Uhr.
- [22] Julian Mack Christiane Floyd, Guido Gryczan. Einführung in die Softwaretechnik Software-Entwicklungsprojekte und Projektmodelle. Paper, Universität Hamburg, <http://www.inf.fu-berlin.de/lehre/WS01/SWT/Projektmodelle.pdf>, 2010. Abruf: 11.06.2010,20:300 Uhr.
- [23] Shenkar College. <http://www.shenkar.ac.il/english/home/>. Webseite. Abruf: 08.06.2010, 10:16 Uhr.
- [24] ComputerBase. <http://pics.computerbase.de/lexikon/91631/350px-Wasserfallmodell.svg.png>. Webseite. Abruf: 22.7.2010, 22:31 Uhr.
- [25] Conrad. http://www.produktinfo.conrad.com/datenblaetter/175000-199999/185336-da-01-en-Distanz_Sensor_GP2D120.pdf. Webseite. Abruf: 16.06.2010, 14:10 Uhr.

- [26] Christine Weiß Dr. Hartmut Strese. <http://www.aal-deutschland.de/>. Webseite. Abruf: 07.06.2010, 13:21 Uhr.
- [27] Christiane Floyd. A systematic look at prototyping. Paper, TU Berlin, [http://www.imamu.edu.sa/Scientific_selections/abstracts/Documents/A Systematic Look At Prototyping.pdf](http://www.imamu.edu.sa/Scientific_selections/abstracts/Documents/A_Systematic_Look_At_Prototyping.pdf), 2010. Abruf: 11.06.2010, 19:50 Uhr.
- [28] Ben Fry and Casey Reas. <http://processing.org/>. Webseite. Abruf: 16.06.2010, 15:10 Uhr.
- [29] Hochschule für Angewandte Wissenschaften Hamburg. <http://www.haw-hamburg.de/schulcampus.html>. Webseite. Abruf: 02.07.2010, 15:23 Uhr.
- [30] Sebastian Gregor. Emotional Tent. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, [http://users.informatik.haw-hamburg.de/ ubicomp/](http://users.informatik.haw-hamburg.de/ubicomp/), 2009. Abruf: 08.06.2010, 15:40 Uhr.
- [31] Sebastian Gregor. Physical Interaction Design. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, [http://users.informatik.haw-hamburg.de/ ubicomp/](http://users.informatik.haw-hamburg.de/ubicomp/), 2009. Abruf: 16.06.2010, 15:40 Uhr.
- [32] Prof. Franziska Hübler, Prof. Dr. Kai von Luck, Prof. Dr. Gunter Klemke, and Jeremy Abbett. <http://ambientawareness.org/>. Webseite. Abruf: 08.06.2010, 15:21 Uhr.
- [33] Manuel von Klipstein Jan T. Krahn. <http://www.pentiment.de/frameset/frameset.html>. Webseite. Abruf: 08.06.2010, 10:13 Uhr.
- [34] André Jeworutzki. Interaktive Objekte Stelldichein: Informatik trifft Design. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, [http://users.informatik.haw-hamburg.de/ ubicomp/](http://users.informatik.haw-hamburg.de/ubicomp/), 2009. Abruf: 09.06.2010, 19:40 Uhr.
- [35] André Jeworutzki. Pädagogik intelligent verkleidet: Lehre von Design, Elektrotechnik und Informatik im Rahmen eines Workshops von Studenten für Studenten. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, [http://users.informatik.haw-hamburg.de/ ubicomp/](http://users.informatik.haw-hamburg.de/ubicomp/), 2009. Abruf: 16.06.2010, 19:40 Uhr.
- [36] André Jeworutzki, Svenja Keune, and Larissa Müller. <http://users.informatik.haw-hamburg.de/projects/toasteredwin/>. Webseite. Abruf: 25.06.2010, 14:30 Uhr.
- [37] Silab Kamawall. <http://www.moviereporter.net/filme/671-snakes-on-a-plane>. Webseite. Abruf: 09.06.2010, 20:21 Uhr.
- [38] Gottfried Keller. *Kleider machen Leute*. Reclam, 2000.
- [39] Svenja Keune. <http://svenjabachelorproject.posterous.com/>. Webseite. Abruf: 13.06.2010, 12:30 Uhr.

- [40] Svenja Keune. DIE HELLSTE KERZE AUF DER TORTE ORGANISMEN UNTERHALTEN SICH. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, 2010.
- [41] Universität Koblenz. <http://www.uni-koblenz.de/FB4/Institutes/IST/AGEbert/Teaching-/SS07/SoftwareTechnik/processmodels003.png>. Webseite. Abruf: 16.07.2010, 14:10 Uhr.
- [42] Winnie W.Y. Lau, Grace Ngai, Stephen C.F. Chan, and Joey C.Y. Cheung. Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 504–508, New York, NY, USA, 2009. ACM.
- [43] Limor. <http://www.ladyada.net/make/gpsshield/>. Webseite. Abruf: 16.06.2010, 11:00 Uhr.
- [44] Limor. <http://www.ladyada.net/make/mshield/>. Webseite. Abruf: 16.06.2010, 11:46 Uhr.
- [45] Limor. <http://www.ladyada.net/make/waveshield/design.html>. Webseite. Abruf: 16.06.2010, 11:00 Uhr.
- [46] Zhongyi Lv and Zheng Yang. Playful factors in interactive design of automotive gps. In *Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on*, pages 133 –136, 22-25 2008.
- [47] Steve Mann. <http://www.eyetap.org/defs/glossary/wearcomp/>. Webseite. Abruf: 07.06.2010, 12:21 Uhr.
- [48] MIT. <http://web.mit.edu/>. Webseite. Abruf: 25.06.2010, 14:30 Uhr.
- [49] Matthias Vogt Mohammadali Rahimi. Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2008.
- [50] J.R. Mountain and R.L. Wells. Engaging k-12 students and teachers using competitive interactive design. In *Frontiers in Education, 2002. FIE 2002. 32nd Annual*, volume 1, pages T3C–12 – T3C–17 vol.1, 2002.
- [51] G. Ngai, S.C.F. Chan, J.C.Y. Cheung, and W.W.Y. Lau. Deploying a wearable computing platform for computing education. *Learning Technologies, IEEE Transactions on*, 3(1):45 –55, jan.-march 2010.
- [52] Inc. O'Reilly Media. http://blog.makezine.com/archive/2010/06/flexible_arduino_is_flexible.html. Webseite. Abruf: 17.07.2010, 14:20 Uhr.

- [53] Gustav Pomberger and Wolfgang Pree. *Software Engineering Architektur-Design und Prozessorientierung*. Hanser, 2004.
- [54] Dipl.-Ing.(FH) Patrick Postel, Dipl.-Ing.(FH) Jaroslaw Zdrzalek, and Dipl.-Ing.(FH) Sebastian Schünemann. <http://www.silpion.de/>. Webseite. Abruf: 08.06.2010, 20:21 Uhr.
- [55] Julia Pressburger. Interaktive Kunst. Bericht, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2009. Abruf: 21.06.2010, 8:40 Uhr.
- [56] Xenia Rendtel. <http://www.mrg-online.de/joomla/>. Webseite. Abruf: 02.07.2010, 17:00 Uhr.
- [57] Kai Rosseburg. Entwicklung einer Programmierumgebung für roboterbasierten Informatikunterricht an Schulen. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, 2007.
- [58] Kai Rosseburg. Ambient Awareness Insector. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2009. Abruf: 05.06.2010, 11:15 Uhr.
- [59] Marcus Rödiger. Interaktive Steuerung von Computersystemen mittels Erkennung von Körpergesten. Masterarbeit, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2010.
- [60] Scott Snibbe. <http://www.snibbe.com/>. Webseite. Abruf: 18.7.2010, 16:31 Uhr.
- [61] Sparkfun. <http://www.sparkfun.com/datasheets/DevTools/LilyPad/LilyPad-LightSensor.pdf>. Webseite. Abruf: 16.06.2010, 14:20 Uhr.
- [62] Martin Sukale. Computergestützte Kunstprojekte - Neuere technologische Entwicklungen. Studienarbeit, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2008. Abruf: 23.06.2010, 8:40 Uhr.
- [63] Martin Sukale. Konstruktion eines Netzwerkes eingebetteter Systeme für interaktives Design. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2008.
- [64] J.F. Sullivan, S.E. Davis, J.L. deGrazia, and D.W. Carlson. Beyond the pipeline: building a k-12 engineering outreach program. In *Frontiers in Education Conference, 1999. FIE '99. 29th Annual*, volume 1, pages 11B5/21 –11B5/26 vol.1, 1999.
- [65] Julia Pressburger Sven Tennstedt. Emotional Tent Ambient Awareness. Projektbericht, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2009. Abruf: 08.06.2010, 15:20 Uhr.

-
- [66] Sören Voskuhl. Bewegungsbasierte Computerinteraktion zur Navigation in Informationsbeständen. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, <http://users.informatik.haw-hamburg.de/ubicomp/>, 2009.
- [67] Paul Watzlawick. <http://www.paulwatzlawick.de/>. Webseite. Abruf: 10.06.2010, 12:21 Uhr.
- [68] Wikipedia. Agile softwareentwicklung — wikipedia, die freie enzyklopädie, 2010. [Online; Stand 28. Juli 2010].
- [69] Wikipedia. Buddy-projekt — wikipedia, die freie enzyklopädie, 2010. [Online; Stand 28. Juli 2010].
- [70] Wikipedia. Extreme programming — wikipedia, die freie enzyklopädie, 2010. [Online; Stand 28. Juli 2010].
- [71] Wikipedia. Interdisziplinarität — wikipedia, die freie enzyklopädie, 2010. [Online; Stand 28. Juli 2010].
- [72] Wikipedia. Testgetriebene entwicklung — wikipedia, die freie enzyklopädie, 2010. [Online; Stand 28. Juli 2010].

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) bzw. §25(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 6. August 2010

Ort, Datum

Unterschrift