
C Realisierung

Die Projektdurchführung gliedert sich in mehrere Phasen. Während der *Analyse* werden die Ziele definiert und es wird festgelegt, auf welche Weise sie am besten erreicht werden können. Diese Überlegungen wurden in früheren Teilen schon getroffen und werden deswegen hier nicht noch einmal explizit dargestellt.

An diese Phase schließt sich die *Konzeption* an. Hier wird die Software geplant, ihre Bereiche festgelegt und die Designrichtlinien aufgestellt. Außerdem werden die Inhalte ausgewählt, recherchiert und in eine geeignete Reihenfolge gebracht. Das Ergebnis dieser Phase ist ein Drehbuch, das alle Inhalte zusammenfasst.

Während der eigentlichen *Produktionsphase* müssen die gewählten Inhalte medial umgesetzt und zu einem Produkt integriert werden. Anschließend werden die interaktiven Funktionen programmiert und die fertige Software getestet und distribuiert.

9 Planung der Lernumgebung

Zur Planung der Lernumgebung gehören allgemeine Überlegungen zu den Bereichen der Software und zu ihrer Gestaltung. Das Ergebnis ist ein Überblick, aus welchen Teilen die Software besteht und wie sie zusammenhängen. Ein Styleguide stellt die wichtigsten Richtlinien zur Gestaltung zusammen.

9.1 Bereiche der Software

Die Software besteht aus mehreren Bereichen. Der größte ist der eigentliche Inhaltsbereich. Zusätzlich sind aber noch weitere Funktionen vorhanden, die als Lernhilfe dienen. Dazu gehören die Kommunikationsfunktionen („Gespräche“), ein Wörterbuch, die Kapitelauswahl und die Hilfe. Diese Funktionen sind über die Hauptnavigation jederzeit abrufbar. Schließlich sind all diese Bereiche in einen Rahmen eingebunden, der den Startscreen, das Anmelden, die Begrüßung und den Abspann umfasst.

Inhaltsbereich

Der Inhaltsbereich gliedert sich in Hauptauswahl, Inhaltspräsentation und Tests. Die Hauptauswahl bietet einen Zugriff auf die drei Hauptpunkte Licht, Licht und Gegenstände und Farben. Wenn der Benutzer über einen der Punkte fährt, erklärt ein kurzer Sprechertext, was sich dahinter verbirgt.

Den größten Teil macht die Inhaltspräsentation aus. Sie beginnt mit einer Einführung (Advanced Organizer), deren Funktion es ist, den Benutzer auf die kommenden Inhalte vorzubereiten, ihm die Ziele aufzuzeigen und relevante Ideen in seiner kognitiven Struktur anzuregen.

Anschließend folgen die eigentlichen Inhalte. Sie sind in kleinere Kapitel aufgeteilt, auf die später direkt zugegriffen werden kann. Um von einem Kapitel zum nächsten zu gelangen, muss der Anwender auf einen „Weiter“-Button klicken. So wird ihm ermöglicht, die vorhergehenden Inhalte nochmal in Ruhe durchzulesen und erst dann weiter zu machen, wenn er es möchte. Die Möglichkeit, weiter zu schalten wird erst dann gegeben, wenn der komplette Inhalt gezeigt wurde. Ein schnelles „Durchblättern“ der Inhalte ist nicht möglich, damit der Benutzer nicht einfach wichtige Teile überspringt.

Die Inhalte werden durch einen Cartoon-Professor vorgetragen, der oben in einem kleinen Fenster zu sehen ist. Die wichtigsten Punkte erscheinen zusätzlich zum Sprechertext auch als Schrift auf dem Bildschirm. Dadurch kann der Benutzer im Text zurückspringen und für ihn schwierige Stellen wiederholen. Außerdem kann das Programm auch noch angehalten werden, wenn der Anwender eine Pause machen möchte. Zusätzlich zu den Texten werden die Inhalte durch Grafiken, Fotos, Animationen und kleine Experimente repräsentiert.

Im Inhaltsbereich gibt es eine feste Sequenz aus Standardinhalten, die jeder durchlaufen muss. Daneben können von einigen Seiten Erklärungen und Zusatzinformationen aufgerufen werden. Zum Beispiel kann man von der Solarzellen-Seite zur Zusatzinformation „Wie funktionieren Solarzellen?“ gelangen. Die Erklärungen stellen dabei zusätzliche Informationen bereit, die schwierige Sachverhalte erklären sollen, die Zusatzinformationen bieten weiterführende und manchmal schwierigere Inhalte an.

In die Inhaltspräsentation sind kleine Tests eingebaut. Sie haben mehrere Funktionen: Erstens können sie testen, ob ein vorheriges Kapitel verstanden wurde. Dies ist dann sinnvoll, wenn dieses eine wichtige Voraussetzung für das nächste Kapitel darstellt. Zweitens können, ähnlich wie beim fragend-entwickelnden Unterrichtsgespräch, Fragen gestellt werden, auf die der Benutzer die Antwort nicht kennen kann. Sie dienen dazu, dass der Lernende sich Gedanken macht, zum Beispiel über die Beobachtungen in einem Experiment. Schließlich können die Fragen auch noch nach einem Kapitel gestellt werden um zu prüfen, ob der Lernende wichtige Ideen verstanden hat.

Näheres zu der Auswertung und Funktionsweise von Tests findet sich unten im Abschnitt „Testauswertung und Adaptivität“ (4.3).

Hilfe

Die Hilfe ist ein wichtiger Bereich einer Software, besonders wenn - wie bei mir - die Zielgruppe eher ungeübt im Umgang mit dem Computer ist. Wenn Fragen oder Probleme bei der Bedienung der Software auftauchen und nicht geklärt werden, kann dies schnell zu Frustration führen, die Software wird nicht mehr benutzt. Die Hilfe gliedert sich in zwei Bereiche: eine Anleitung und eine Liste mit Problemen, die auftreten können.

Die Anleitung wird jedem Benutzer gezeigt, der sich das erste Mal neu anmeldet. Hier wird erklärt, welche Bereiche das Programm hat, welche Steuerungsmöglichkeiten es gibt, wie ein Button aussieht und Ähnliches. Sie muss einmal komplett durchlaufen werden.

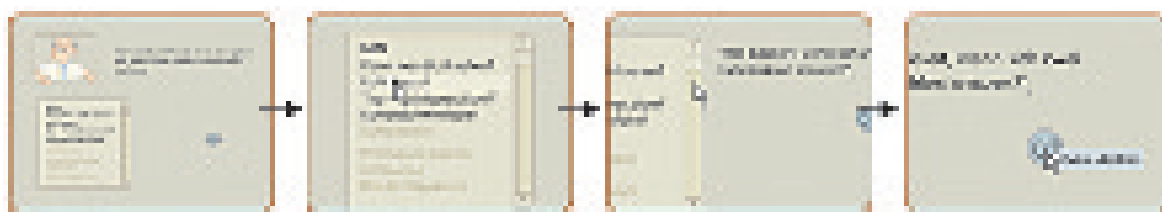


Abb. 12: Sequenz „Wie funktioniert die Kapitelauswahl?“

Unter dem Menüpunkt „Hilfe“ findet man die Problem- und Fragenliste. Dazu gehören zum Beispiel „Wieso kann ich nicht alle Kapitel auswählen?“ oder „Was bedeutet ‚Einstellungen für Macromedia Flash Player?‘“. In den Fällen, wo es für den Benutzer hilfreich ist, werden

diese Fragen mit einer Animation beantwortet, die die genauen Mausbewegungen und -klicks zeigt, die der Anwender durchführen muss, um das Problem zu beheben bzw. die gewünschte Funktion auszuführen. Die obige Abbildung zeigt so eine Bildfolge.

Kommunikation

Die verschiedenen Möglichkeiten zur Kommunikation findet man unter dem Menüpunkt „Gespräche“. Hier kann man dann zwischen Forum, Chat und Nachrichten wählen. Forum und Chat öffnen beide ein Browserfenster und verlinken auf eine externe Seite, die unter www.greenzone.de liegt.

Das *Forum* bietet die Möglichkeit, mit anderen Lernenden oder einem Tutor Fragen und Probleme zu diskutieren. Die Kommunikation ist asynchron, das heißt man schreibt einen Beitrag und muss dann erst warten, bis irgendjemand antwortet. Das kann je nachdem, wie viele Menschen das Forum benutzen, wenige Minuten bis einige Tage dauern. Der Vorteil eines Forums ist, dass alle Beiträge in einer relativ übersichtlichen Form gespeichert werden. Dadurch hat man neben einer Kommunikationsplattform auch gleich ein Archiv mit Fragen und Antworten. Das Forum zu meiner Lernsoftware soll dazu dienen, sachliche Fragen zu Programm und Inhalt zu klären.

Der *Chat* erlaubt synchrone Kommunikation. Wie bei einem normalen Gespräch können die Teilnehmer zu jeder Zeit ihre Beiträge schreiben, die auch sofort im Chat erscheinen. Anders als das Forum sind die Beiträge nicht nach Themengebieten geordnet, sondern sie erscheinen in der Reihenfolge, in der sie eingegangen sind. Der Chat ist dafür gedacht, eine freie, ungezwungene Kommunikation zwischen den Lernenden zu ermöglichen.

Die letzte Kommunikationsart, der *Nachrichtenservice*, erlaubt es dem Anwender, direkte Fragen an den Tutor zu stellen. Hier öffnet sich, anders als bei Forum und Chat, kein neues Fenster, sondern der Service ist direkt in die Lernsoftware eingebunden. Man kann einfach seine Nachricht in ein Textfeld eingeben und abschicken. Die Antwort erscheint ebenfalls direkt im System. Zusätzlich wird sie zur Dokumentation an die Email-Adresse des Fragenden geschickt, da die Software Fragen und Antworten nicht speichert.

Wörterbuch

Das Wörterbuch enthält eine Reihe von Begriffen, die in den Inhalten vorkommen. So kann der Anwender immer schnell nachschauen, was ein Begriff bedeutet. Umgesetzt wurde dieses Glossar als eine scrollbare (d.h. man kann sie nach oben und unten bewegen), alphabetisch geordnete Liste. Klickt man auf einen Begriff, erscheint in einem Textfeld daneben die Erklärung.

gramm geworfen werden. Entscheidet man sich hier erneut fürs Beenden, folgt ein kurzer Abspann mit den Namen aller an der Produktion Beteiligten, anschließend schließt sich das Programm.

9.2 Layout & Design

Bei der Planung von Lernmedien ist die Gestaltung der Oberfläche besonders wichtig, da sie die Schnittstelle zwischen Medium und Benutzer ist. Sie muss an die Zielgruppe sowie an die Lehrinhalte und Lernziele angepasst sein. In diesem Kapitel werden die Überlegungen dargestellt, die zum Design meines Produkts geführt haben. Als Ergebnis dieser Phase entsteht ein Styleguide, der die Gestaltungsregeln für die Produktion zusammenfasst.

Layout

Das Screen-Layout stellt das Grundlage für das Design der Benutzeroberfläche dar. Diese sollte leicht verständlich und bedienbar sein, so dass der Benutzer sich gut zurechtfindet. Gut ist es hier, wenn Inhalts- und Navigationsbereich klar voneinander getrennt sind. Darüberhinaus sollte das Design konsistent sein, ähnliche Elemente sollten auch in ähnlicher Form dargestellt werden, zum Beispiel alle Buttons als rote Kreise. Aber auch die Position der Elemente sollte auf allen Screens beibehalten werden, es ist verwirrend wenn die Navigation erst am oberen Rand und beim nächsten Screen unten in der Ecke liegt. [vgl. Bruns, 2000; Riempp, 2000]

Um eine einheitliche und konsistente Anordnung zu gewährleisten bietet es sich an, ein Raster (Grid) zu erstellen. Dazu muss in einem ersten Schritt die insgesamt verwendete Fläche bestimmt werden.

Nachdem die Gesamtfläche feststeht, kann sie in Bereiche für Inhalte und Navigation unterteilt werden. Es gibt zwei Bereiche für die Navigation: der rechte und der untere Rand. In der linken oberen Ecke ist ein fester Bereich für den virtuellen Professor reserviert. Abbildung 14 zeigt das Raster, das meinem Design zu Grunde liegt und zwei Screens, die an diesem Grid ausgerichtet sind.



Abb. 14: Das Grid (oben) und zwei Screens (Test & Inhaltsseite)

Metapher & Stil

Metaphern erleichtern den Umgang mit der Software, indem sie etwas aus dem Alltag Bekanntes auf die Darstellung der Objekte auf dem Bildschirm übertragen. Ein bekanntes Beispiel ist die Desktop-Metapher des Betriebssystems Windows: Dateien werden in Ordnern abgelegt, wenn man etwas löschen möchte, verschiebt man es in den Papierkorb. [vgl. Thissen, 2000]

Man sollte darauf achten, dass die Metapher nicht zu sehr in den Vordergrund rückt und störend wirkt. Ihre optische Darstellung darf nicht über die Inhalte dominieren, die Software sollte noch schnell bedienbar sein [vgl. Bruns, 2000]. Außerdem sollte die Metapher zu den Inhalten passen, zum Beispiel eine Konzertsaal-Umgebung für eine Lernsoftware zur Musik.

Ich habe eine Universität mit Labor als Metapher gewählt, sie allerdings eher unauffällig umgesetzt. Die Hauptauswahl sieht aus wie eine Tafel, die Experimente werden im „Lichtlabor“ gemacht und ein Cartoon-Professor leitet durch die Software und stellt die Inhalte vor.



Abb. 15: Stil & Metapher: Das Lichtlabor, Buttons und der Professor

Das gesamte Design sollte einem einheitlichen Stil folgen, zum Beispiel klassisch-schlicht oder verspielt-bunt. Mischt man diese verschiedenen Stile, wirkt das ganze Produkt unausgegoren und beliebig. Wie überhaupt beim Design sollte man sich bei der Wahl eines Stils an der Zielgruppe orientieren. Ich habe für mein Lernprogramm eine bunte, runde, leicht cartoonhafte, aber trotzdem realitätsnahe Darstellung gewählt, die auch nicht zu verspielt ist. Die Figur, die durch das Programm leitet ist eine Car-

toon-Figur, die Gegenstände im „Lichtlabor“ und die Grafiken und Buttons wurden in einem ähnlichen Stil gehalten. Auf dem Bild oben kann man einige der Elemente sehen, die den Stil bestimmen.

Farbe

Farbiges Lernmaterial generell positiver eingeschätzt, als welches das lediglich Graustufen verwendet. In Blickbewegungsstudien konnte festgestellt werden, dass farbige Flächen bevorzugt betrachtet werden, dabei besonders helle und gesättigte [vgl. Bruns, 2000]. Deswegen sollten Hintergründe weniger gesättigt oder dunkler sein als die Flächen, in denen Inhalte stehen. Der Hintergrund sollte außerdem ruhig sein, Grafiken stören hier oft, außer wenn sie unauffällig sind. Bei mir ist die Hintergrundfarbe ein gedecktes, fast graues hellblau, der Inhaltsbereich ist farbiger und sticht mehr hervor.

Es sollten nicht zu viele Farben auf einmal verwendet werden, besonders dann, wenn sie sehr bunt und grell sind. Eine solche Farbkombination „wird nur von sehr wenigen Benutzern akzeptiert, sie ergibt ein unruhiges Gesamtbild und verwirrt (...)“ [Thissen, 2000, S. 128].

Farben können auch zur Strukturierung verwendet werden, zum Beispiel können alle Seiten eines Oberpunkts die gleiche Farbe haben. Dadurch kann der Benutzer schnell erkennen, in welchem Bereich er ist. Ich verwende diese Farbcodierung, um zwischen Inhaltsbereich und Zusatzfunktionen zu trennen. Jeder der Hauptpunkte hat eine eigene Farbe, alle Inhalte sind hellblau hinterlegt. Jede Farbe hat ihre eigene Bedeutung, zum Beispiel wirkt gelb hell und freundlich oder weiß rein und vollkommen. Bei der Wahl der Farben für die unterschiedlichen Bereiche wurde dies berücksichtigt. Der Kommunikationsbereich ist rot, er fordert den Benutzer zu Aktivität auf. Orange als Farbe der Weisheit findet sich beim Wörterbuch wieder. Die Hilfe ist in grün gehalten, das beruhigend und entspannend wirkt; Eigenschaften, die ein Hilfe Suchender gut gebrauchen kann [vgl. Thissen, 2000]. Für die Kapitelauswahl habe ich ein Blau gewählt, was allerdings keine weitere Bedeutung hat, sondern schlicht die letzte verbleibende Farbe war. Alles, was zum Rahmen der Software gehört ist in einem Beigeton gehalten. Die folgende Abbildung zeigt die verschiedenen Farben für vier der Hauptbereiche. Alle Hintergrundfarben sind hell, damit sich die schwarze Schrift gut davon abhebt.



Abb. 16: Farbcodierung der Hauptbereiche: Kapitelauswahl, Wörterbuch, Hilfe, Kommunikation

Schrift

In der Typografie unterscheidet man zwischen serifenloser Schrift und Serifenschrift. Serifen nennt man die kleinen Häkchen an den Buchstaben. Serifenschriften sind im Allgemeinen besser lesbar, die Serifen führen beim Lesen das Auge und machen manche Buchstaben leichter erkennbar. Deswegen werden sie oft bei Druckerzeugnissen eingesetzt. Schrift am Bildschirm gehorcht jedoch ihren eigenen Regeln. Allgemein sind Texte am Bildschirm recht schwierig zu lesen. Im Vergleich zum Druck ist die Textqualität am Monitor schlecht, so dass die klassischen serifenbehafteten Druckschriftarten nur schlecht lesbar sind, vor allem in kleinen Größen. Daher sind serifenlose Schriften zu bevorzugen. Die Größe der Schrift sollte nicht zu klein gewählt werden, je nach Schriftart liegt die Mindestgröße bei 10 bis 12 Punkt. Kursive und fette Schrift ist schwer lesbar und sollte deshalb nur zur Hervorhebung verwendet werden. Blinkende Texte oder der ausschließliche Einsatz von Großbuchstaben sollten vermieden werden, da sie die Aufmerksamkeit zu sehr auf sich ziehen und schlecht lesbar sind.

Neben der Schriftart muss man sich auch Gedanken über Zeilenabstand, Ausrichtung und Zeilenlänge machen. Der Zeilenabstand sollte so groß sein, dass das Lesen möglichst angenehm ist. Ist er zu klein, wird die Unterscheidung zwischen den einzelnen Zeilen schwierig, ist er zu groß, entstehen optische Löcher im Text. Ein eineinhalb- bis zweifacher Zeilenabstand hat sich beim Screendesign bewährt. Texte sollten linksbündig im sogenannten Flattersatz ausgerichtet werden. Das bedeutet, dass die Zeilen, im Gegensatz zum Blocksatz, an verschiedenen Stellen enden. Der Vorteil ist, dass der Abstand zwischen den Wörtern und Buchstaben immer gleich groß ist. Dies erleichtert das Lesen. [vgl. Thissen, 2000]

Für die Zeilenlänge werden ca. 60 bis 80 Buchstaben empfohlen. Ist die Zeile zu kurz, werden viele Worttrennungen notwendig, was den Lesefluss stört. Bei zu langen Zeilen fällt die Unterscheidung zwischen den einzelnen Zeilen schwer, das Auge verirrt sich zwischen den Zeilen. Ein Textblock sollte nicht länger als 15 Zeilen sein [vgl. Bruns, 2000].

Navigation & Orientierung

Es gibt grundsätzlich zwei Arten, wie eine Navigation in das Angebot integriert werden kann. Man kann entweder eine klare Trennung zwischen Informations- und Navigationselementen vornehmen oder Navigation und Inhalte miteinander verweben [vgl. Thissen, 2000]. Im Extremfall sind Navigations- und Informationselemente nicht mehr zu unterscheiden (explorative Navigation). Die folgende Abbildung zeigt Beispiele dafür.

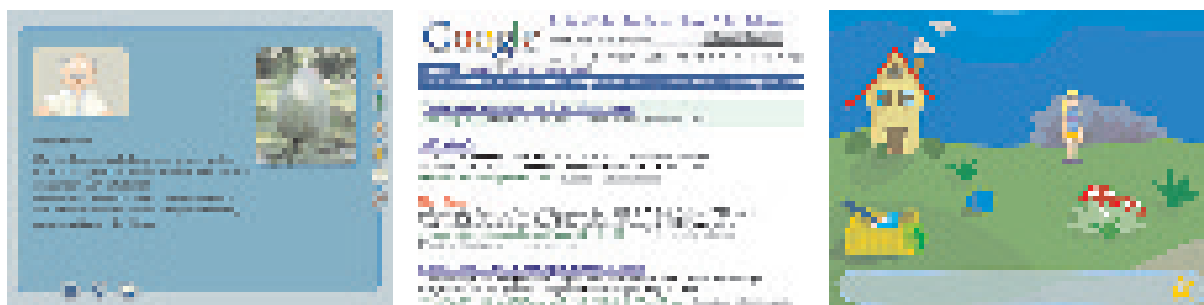


Abb. 17: getrennte, verwobene und explorative Navigation: Welt der Optik, Google, Astro Susi [www.google.de, www.greenzone.de/astro-susi]

Eine explorative Navigation findet man hauptsächlich bei spielerischen Anwendungen, wo man sich durch virtuelle Welten bewegen muss. Für ernsthafte Anwendungen ist das nicht sinnvoll, da man sonst mehr Zeit damit verbringt, die Links zu suchen als Inhalte zu lesen. Bei klassischen Websites sind oft Verknüpfungen in den Text der Seite integriert, aber besonders hervorgehoben (typischerweise unterstrichen).

Eine gute Navigation ist intuitiv bedienbar und muss nicht erst erlernt werden. Als wesentliche Anforderung an Navigationselemente ergibt sich daher, dass sie schnell erkennbar sein sollten [vgl. Thissen, 2000]. Deswegen sehen alle Buttons bei meinem Produkt gleich aus: ein hellblauer Kreis mit einer dunkleren Umrandung und einem Symbol darin. Buttons, die der

Benutzer anklicken muss, haben zusätzlich noch eine blinkende Markierung, damit der Anwender sie schnell findet. Bei außergewöhnlichen Aktionen, zum Beispiel wenn der Benutzer ein Experiment durchführen muss, erscheint eine Aufforderung, was er machen muss. Auch diese Aufforderungen sind durch einen orangenen Kreis mit Ausrufezeichen klar gekennzeichnet.

Außerdem ist es gut, wenn man den Elementen ansieht, was ihr Zweck ist. Dies habe ich so gelöst, dass erstens die Symbole selbst schon aussagekräftig sind, zum Beispiel ein sprechender Kopf für „Gespräche“. Abbildung 18 zeigt die Buttons und ihre Bedeutung. Zweitens erscheint ein Tooltip (Beschreibungstext), wenn man mit der Maus über den Button fährt.

Die Navigation bei meiner Lernsoftware unterteilt sich in zwei klar getrennte Bereiche: das Hauptmenü und der Steuerungsbereich bzw. die Subnavigation. Das



Abb. 19: Die Buttons und ihre Bedeutung

Hauptmenü ist ständig am rechten Rand zu sehen, hier findet man alle Lernhilfen sowie die Hauptauswahl und kann das Programm beenden. Dieses Menü bietet zusätzlich noch eine Orientierungsfunktion: Der Button des Bereichs in dem man sich gerade befindet ist farbig markiert.

Die Subnavigation am unteren Rand ändert sich, je nachdem, wo man ist. Im Inhaltsbereich sind ständig drei Steuerungsbuttons vorhanden. Ein *Start-Stopp-Button* hält das Programm an und spielt es wieder ab. Ein *Wiederholen-Button* ermöglicht es, einen Abschnitt von vorne abspielen zu lassen und über einen *Geschwindigkeitsregler* lässt sich einstellen, ob das Programm schnell oder langsamer abläuft. Am Ende jedes Kapitels erscheint außerdem noch ein *Weiter-Button*, der zum nächsten Kapitel springt. Zu manchen Inhalten gibt es Zusatzinformationen oder Erklärungen. Hier erscheint dann neben der Steuerung ein *Mehr dazu*- oder ein *Erklärungsbutton*. Wenn man gerade in einem solchen Zusatzmodul ist, kann man über einen *Zurück-Button* zur Ursprungsseite zurückkehren.

Springt man aus dem Inhaltsbereich in einen der Hauptpunkte, erscheint dort ein *Zurück zu den Inhalten-Button*, der ermöglicht, direkt zu der Stelle zu springen, an der man den Inhaltsbereich verlassen hat.

Man sollte dem Anwender immer auch eine Orientierungsmöglichkeit geben. Dadurch kann man folgende Fragen beantworten: Wo bin ich gerade? Wo war ich schon? Was kommt noch alles? Wie viel habe ich schon gesehen? [vgl. Thissen, 2000] Besonders wichtig ist dies bei Hyper-Strukturen, da sich der Benutzer dort besonders schnell desorientiert fühlen kann.

In der „Welt der Optik“, die sequentiell strukturiert ist, ist es einfacher, die Orientierung zu behalten. Das Hauptmenü zeigt an, in welchem Oberpunkt man sich gerade befindet, im Inhaltsbereich kann man an der Überschrift erkennen, wo man ist. Hier kann man auch abschätzen, wie lange das Kapitel noch dauert: Wenn der Bildschirm voll ist, ist auch das Kapitel zu Ende, wenn er noch fast leer ist, dauert es noch eine Weile. In der Kapitelauswahl kann man erkennen, was und wieviel man schon gesehen hat und was noch kommt.

Styleguide

Der Styleguide stellt alle grafischen Richtlinien für die Produktion zusammen. Dazu gehört das zu Grunde gelegte Raster mit genauen Pixelangaben zu den verschiedenen Linien, die verwendeten Farben und Schriften, die Größen von Elementen wie zum Beispiel Buttons und Ähnliches. Dadurch wird gewährleistet, dass das Produkt einheitlich gestaltet wird und keine Brüche im Design auftreten. Je nachdem, ob man die Produktion selbst durchführt, oder ob ein Mitarbeiter oder sogar ein anderes Unternehmen dafür zuständig ist, entscheidet sich, wie detailliert die Angaben im Styleguide sind. Im Anhang ist der Styleguide für „Professor Heinrichs Welt der Optik“ angefügt.

10 Redaktion

Die Redaktion umfasst die Planung und Recherche der Inhalte. Bei der Konzeption von medialen Lernangeboten muss hier wesentlich genauer vorgegangen werden als bei der Planung eines normalen Unterrichts, da Fehler in der Konzeption während des Lehrens nicht mehr geändert werden können. Neben der Planung von *Curriculum* und *Unterrichtsprogramm* muss auch die *Zielgruppe* genauer untersucht werden. Anschließend können die Inhalte und Assets *recherchiert* werden. Das Ergebnis dieser Phase ist das *Drehbuch*.

Bei der Planung von Curriculum und Unterricht habe ich mich an einem Modell von Johnson (vgl. Ausubel, 1980) orientiert.

Das Curriculum beschreibt die Lerninhalte, welche am Schluss vom Lernenden gewusst werden sollen, als eine Reihe von ILOs (Intended Learning Outcomes - erwünschte Lernresultate). Diese dienen als Grundlage für die Entwicklung des Unterrichtsprogramms. Das befasst sich mit der Methode, wie dieser Stoff vermittelt werden kann und beschreibt geeignete Lehrmaterialien, die Lernziele und die Struktur der Inhalte.

Nachdem die Schüler dieses Unterrichtsprogramm durchlaufen haben, werden die tatsächlichen Lernresultate (ALOs, Actual Learning Outcomes) evaluiert. Durch einen Vergleich mit den vorher festgelegten Lernresultaten können Stärken und Schwächen des Curriculums und des Unterrichts festgestellt werden. Diese werden dann bei einer Verbesserung der Planungen berücksichtigt.

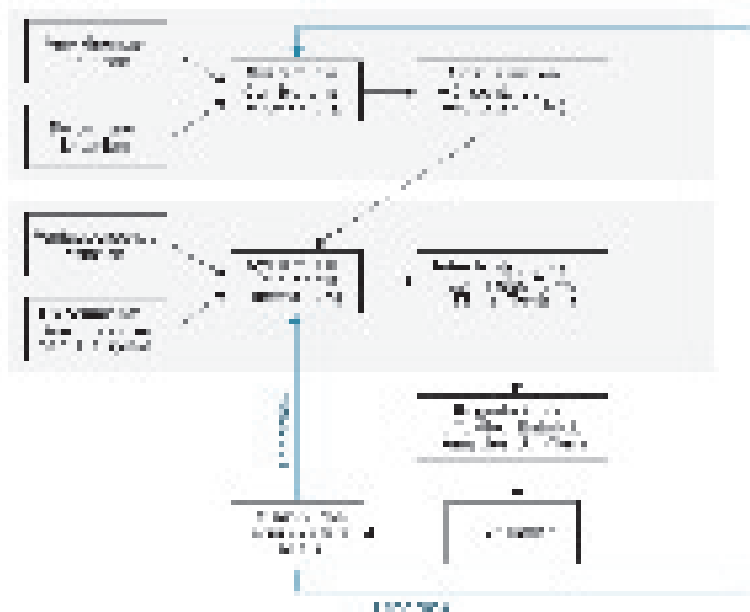


Abb. 19: Unterrichtsentwicklung nach Johnson (vereinfacht)
[vgl. Ausubel, 1980, Abb. 10.1, S. 415; Abb. 10.5, S. 417; Tab. 10.1, S. 420f.]

10.1 Curriculum

Das Curriculum beschreibt die Inhalte, die der Anwender lernen soll, also die ILOs aus obigem Modell. Dabei müssen als erstes die Inhalte zu einem Themengebiet ausgewählt werden. Anschließend werden Begriffe zu diesem Bereich gesammelt und in einer Matrix angeordnet. Dabei kann man die Assimilationstheorie zugrunde legen und eine Hierarchie aufbauen, bei der die allgemeinsten Begriffe oben stehen und die andern Ideen subsumieren. Daraus und aus anderen Überlegungen ergibt sich schließlich die Reihenfolge der Inhalte. [vgl. Ausubel, 1980]

Auswahl der Inhalte

Zu den Aufgaben bei der Entwicklung einer Lernsoftware gehört es, unter vielen möglichen Inhalten zu einem Thema die geeigneten auszuwählen. Einige Gesichtspunkte, die man dabei bedenken sollte sind [vgl. Friehs, 2001]:

- Können allgemeine Regeln und Gesetze aus den Inhalten erschlossen werden?
- Ist das Thema momentan bedeutsam für die Schüler? d.h. Sind die Inhalte für gegenwärtige Probleme hilfreich?
- Wird das Thema in Zukunft eine Bedeutung für die Schüler haben? d.h. Haben die Schüler später einen Nutzen davon?
- Lassen sich die Inhalte gut mit den Methoden der Lernsoftware darstellen?
- Haben die Schüler ein Interesse an dem Thema und wenn nicht, wie kann man es wecken?

Für das gewählte Gebiet, das Licht, ergibt sich insgesamt eine gute Eignung. Die Kinder können erkennen, dass sich viele Phänomene des Alltags durch die Physik beschreiben lassen. Sie haben einen momentanen Nutzen aus dem Gelernten, nämlich das Verstehen, wie einige Dinge in ihrer Umgebung funktionieren. Auch für die Zukunft sind Kenntnisse über Licht hilfreich, schon deswegen, weil das Thema später im Unterricht wieder vorkommt. Die meisten Kinder haben ein großes Interesse daran, Alltagsbeobachtungen zu erklären.

Begriffsmatrix

Im zweiten Schritt müssen die genauen Inhalte definiert werden. Hier muss man darauf achten, dass Umfang und Tiefe der Information angemessen sind und dass die Inhalte sachlich richtig sind.

Um die Inhalte zu bestimmen, müssen zunächst alle relevanten Begriffe gesammelt werden. Dazu bieten sich Methoden wie das Brainstorming an, wobei alle Ideen ungeordnet zusammengetragen werden. Anschließend werden diese Ideen gegliedert. Hier empfiehlt [Ausubel, 1980] eine hierarchische Ordnung, um übergeordnete Ideen zu identifizieren, die zuerst gelehrt werden sollten. Davon ausgehend werden die Inhalte auf die wichtigsten Aspekte reduziert. Die so entstehende Begriffs- (und auch Lehrsatz-) Matrix stellt dann die ILOs dar, also die erwünschten Lernresultate. An Hand von dieser Matrix kann anschließend die Unterrichtsplanung durchgeführt werden.

10.2 Unterrichtsplanung

Bei der Unterrichtsplanung werden die ausgewählten Inhalte in eine geeignete Reihenfolge gebracht und die Lehrmittel bzw. die mediale Umsetzung geplant. Dabei steht die Frage „Welche Aktivitäten (mediale Elemente) und Themen können die Begriffe gut vermitteln?“ im Vordergrund. Organisationshilfen, die das Einordnen in die kognitive Struktur vereinfachen, sollten hierbei ebenso bedacht werden wie Unterrichtshilfen (Texte, Experimente etc.). In die Phase der Unterrichtsplanung fällt auch das Festlegen der Lernziele. [vgl. Ausubel, 1980]

Doch zunächst werden hier einige der relevanten pädagogischen und lernpsychologischen Grundlagen wiederholt, die bei der Wahl der Lehrmittel helfen können.

Pädagogische Grundlagen

Bei der Planung des Unterrichts sollte unbedingt daran gedacht werden, sinnvolles Lernen zu ermöglichen. Viele Beobachtungen im Bereich der Schule zeigen die Vorteile dieses Lernens. So werden zum Beispiel Verallgemeinerungen, Prinzipien und deren Anwendungen besser behalten als konkrete Formeln, Symbole oder Fachbegriffe. Um sinnvolles Lernen zu erreichen, müssen die Inhalte klar und dem kognitiven Entwicklungsstand entsprechend dargebo-

ten werden. Außerdem sollte dafür gesorgt werden, dass die neuen Inhalte unter vorhandene Ideen subsumiert werden. Dabei kann ein Advanced Organizer helfen, der am Anfang eines Kapitels die wichtigsten Ideen vorstellt und so vorhandenes Wissen aktiviert und für den Lernprozess bereitstellt. Vorurteile bzw. volkstümliche Vorstellungen müssen aus der Welt geschafft werden, um das Lernen der richtigen Sachverhalte zu erleichtern.

Auch bei Tests und Fragen sollte man darauf achten, dass man die Lernenden nicht zu mechanischen Lernen verleitet. Deswegen dürfen keine wörtlichen Lehrsätze o.ä. abgefragt werden. Statt dessen sind Aufgaben, die einen Transfer verlangen zu bevorzugen.

Für Lernende in der konkret-operationalen Phase sollten viele Beispiele, Experimente oder Demonstrationen als empirische Stütze angeboten werden, an Hand derer die Inhalte vermittelt werden. Die Inhalte sollten auf alltägliche Erfahrungen der Lernenden Bezug nehmen, da diese Dinge sie interessieren und sie auch dazu motiviert sind, sie zu lernen.

In der abstrakt-logischen Phase ist rezeptives Lernen dem entdeckenden überlegen, da das so erworbene Wissen besser transferierbar und allgemeingültiger ist.

Für meine Zielgruppe gilt also, dass neue Sachverhalte mit Beispielen, kleinen Experimenten und Demonstrationen eingeführt werden sollten. Es können aber auch schon einige Dinge auf einer abstrakteren Ebene behandelt werden, falls es nicht möglich ist, ein passendes Beispiel zu finden. Ein rein verbaler Unterricht ist jedoch auch dann nicht besonders geeignet.

Die Lerninhalte sollten vielfältig präsentiert werden, rezeptive (Text, Bilder) und entdeckende Methoden (Experimente, Zwischenfragen) ergänzen sich hier gut. Lernmedien sollten außerdem so geplant werden, dass sie den Lernenden eine intensive und auch aktive Auseinandersetzung mit den Inhalten ermöglichen.

Lernziele

Das Lernziel kann in zwei Komponenten aufgeteilt werden. Die inhaltliche Komponente beschreibt die Fakten und Konzepte, die gelernt werden sollen, die Ergebniskomponente welche Kompetenzen der Lernende als Ergebnis des Lernens erwerben soll [vgl. Kerres, 2001]. Die inhaltlichen Lernziele wurden in der Curriculumsplanung schon festgelegt, hier gilt es jetzt, mit dieser Grundlage die Ergebniskomponente zu definieren.

Wie in 2.5 beschrieben, kann man zwischen kognitiven, affektiven und psychomotorischen Lernzielen unterscheiden. Die letzte Art von Zielen ist bei meiner Lernsoftware nicht vorhanden, da sie Kenntnisse und keine handwerklichen Fertigkeiten vermitteln soll. Der Anteil der affektiven Ziele ist eher gering. Die Kinder sollen merken, dass Physik nichts Schlimmes ist und nicht unbedingt mit Mathematik und Formeln zusammenhängen muss. Außerdem sollen

sie erkennen, dass sich viele Alltagsbeobachtungen naturwissenschaftlich erklären lassen und nichts „magisches“ an sich haben. Den größten Teil machen die kognitiven Lernziele aus, deswegen werden sie hier entsprechend der Bloomschen Taxonomie unterteilen. Die Liste ist keineswegs vollständig, sondern stellt jeweils nur ein Beispiel dar.

- Kenntnisse: Der Mond ist keine Lichtquelle, sondern reflektiert das Licht der Sonne.
- Verstehen: Licht und Wärme als Teile des elektromagnetischen Spektrum verstehen.
- Anwenden: Abschätzen der Dauer, die das Licht für eine bestimmte Strecke braucht.
- Analyse: Bedingungen überlegen, wann man Lichtstrahlen in der Luft sehen kann.
- Synthese: Herleiten, dass auch farbiges Licht, dass sich kreuzt, sich nicht beeinflusst.
- Bewerten: Bewerten, ob der Begriff „sichtbares Licht“ für einen Teil des elektromagnetischen Spektrums gut gewählt wurde.

Die Analyse der Lernziele an Hand dieses Schemas ist relativ schwierig, da einem die meisten Lernziele höherer Art nicht richtig bewusst sind. Besonders die Analyse und das Bewerten vollziehen sich beim Lernen im Inneren des Lernenden und werden nicht konkret abgefragt.

Verstehen und Anwenden sind die Hauptziele der Software, an einigen Stellen wird auch ein Transfer des Wissens in andere Kontexte verlangt, also eine Synthese. Allerdings werden die Lernenden sich auch Kenntnisse aneignen, selbst wenn darauf in der Software kein Wert gelegt wird, oder Sachverhalte analysieren und bewerten.

Struktur & Sequenz

Bei der Festlegung der Struktur eines Lernangebots muss man sich zuerst zwischen einem Hyper-System und einer festen Sequenz entscheiden. Kerres bietet dazu eine Hilfe an:

	Sequenz	Hyper-Struktur	
Lehrstoff	sequentiell abhängig	flach gegliedert	
Zielgruppe	homogen	inhomogen	
Lernstil	unselbstständig	selbstständig	Tab. 2: Sequenz oder Hyperstruktur?
Vorwissen	niedrig	hoch	[vgl. Kerres 2001, S. 314]

Da das Vorwissen gering, der übliche Lernstil oft unselbstständig und die Zielgruppe homogen ist und außerdem (sequentielle) Abhängigkeiten im Stoff bestehen, ist für mein Projekt eine sequentielle Organisation besser geeignet. Hinzu kommt noch, dass die Zielgruppe im Umgang mit Computern eher ungeübt ist und auch wenig Erfahrung mit Hypertextstrukturen hat.

Wenn man sich für das Sequenzmodell entscheidet, muss man die Inhalte in eine geeignete Reihenfolge bringen. Die Arbeitsgruppe um Gagné entwickelte dazu ein universelles Modell, das erfolgreiches Lernen als ein Folge von neun Aktivitäten beschreibt [vgl. Kerres, 2001, S. 189]:

- Aufmerksamkeit erzielen
- Lehrziele mitteilen
- an Vorwissen anknüpfen
- Lernmaterial präsentieren
- Lernhilfen anbieten
- Gelerntes anwenden
- Rückmeldung geben
- Leistung testen
- Behaltensleistung und Transfer fördern

Die ersten drei Schritte können in einem Advanced Organizer zusammengefasst sein. Das Präsentieren der Inhalte und das Anbieten von Lernhilfen findet bei meiner Software parallel statt, da der Lernende jederzeit Zugriff auf die Zusatzfunktionen (Kommunikation, Lexikon etc.) hat. Die letzten vier Punkte werden von mir als Multiple-Choice-Tests mit Feedback realisiert.

Man kann leicht erkennen, dass Gagné hier nur ein Rahmenmodell bereitstellt, die tatsächliche Umsetzung muss eigenständig überlegt werden. Diese Ungenauigkeit des Modells war Gagné durchaus bewusst, er schreibt dazu: „To imagine that the design process can be reduced to a set of cook-book prescriptions is wishful thinking“. (Sich vorzustellen, dass der Gestaltungsprozess auf das Umsetzen von Kochbuch-Anweisungen reduziert werden kann, ist Wunschdenken.) [Kerres, 2001, S. 190]

Außerdem fällt auf, dass das eigentliche Gliedern der Inhalte nicht genauer beschrieben wird. Posner & Strike (1976 [vgl. Kerres, 2001]) untersuchten verschiedene Vorschläge zur Sequenzierung von Inhalten und ermittelten dabei mehrere Prinzipien, von denen hier einige dargestellt werden.

Zunächst können Inhalte nach der *Ordnung der Dinge* sequenziert werden, z.B. der Geschichtsunterricht fortschreitend in der Zeit, ein Reiseführer nach der räumlichen Anordnung der Städte. Diese Ordnung ist recht einfach, aber nicht immer sinnvoll.

Zweitens kann die *logische Struktur* des Inhalts als Ordnungskriterium dienen. Hier kann nach inhaltlichen Abhängigkeiten (Addition vor Multiplikation), der Komplexität (von leicht zu schwer) oder einer Objekthierarchie (Übergeordnetes zuerst) geordnet werden. Diese Sequenzierung entspricht der Assimilationstheorie, die vorschlägt, von Allgemeinen zum Speziellen, vom Vertrauten zum Fremden und vom Leichten zum Schweren vorzugehen.

Man kann auch dem *Erkenntnisprozess* folgen, also die Inhalte zuerst darstellen, die als erstes entdeckt wurden (zum Beispiel beim Atommodell erst das Atom als feste Kugel, später kreisende Elektronen um den Kern). Dieser Ansatz wird besonders dann angewendet, wenn das eigenständige Entdecken gefördert werden soll.

Schließlich kann der *Lernende* und seine Fortschritte zur Ordnung herangezogen werden. Dazu gehören das Berücksichtigen des Vorwissens, des Interesses und des erlebten Schwierigkeitsgrads. Dieses Modell kann natürlich auch mit den anderen Arten der Sequenzierung verknüpft werden.

10.3 Zielgruppe & Vorwissen

Bei der Planung einer Lernsoftware muss natürlich immer auch die Zielgruppe berücksichtigt werden. Die Gestaltung der Oberfläche, der Schwierigkeitsgrad der Sprache und die Darbietung der Inhalte müssen an sie angepasst sein.

Der erste Punkt ist einleuchtend: Eine Software für Erwachsene muss sicherlich anders aussehen als eine für Kinder. Denn während Kinder bunte Farben, viele Bilder und eine verspielte Gestaltung lieben, bevorzugen Erwachsene eher dezente und übersichtliche Designs. Aber nicht nur das *Alter* spielt hier eine Rolle. Auch das *kulturelle Milieu* oder die Gruppenzugehörigkeit der Benutzer muss berücksichtigt werden. Ein Beispiel für diese Differenzierung sieht man auf dem Zeitungsmarkt. Boulevardblätter sind eher für bildungsferne Milieus mit geringem Anspruch ausgelegt, das Design ist bunt und auffällig. Der Bildungsbürger bevorzugt eine klare und strenge Gestaltung wie bei seriösen Tageszeitungen.

An diesen beiden Bedingungen (Alter und kulturelle Zugehörigkeit) muss sich auch die Sprache ausrichten. Texte für Kinder oder Laien dürfen nur wenige Fachwörter enthalten, die außerdem erklärt werden müssen. Bei Kindern muss zusätzlich berücksichtigt werden, dass auch ihr allgemeiner Wortschatz kleiner ist und sie viele Fremdwörter aus dem alltäglichen Gebrauch, zum Beispiel „Kommunikation“, nicht kennen. Bei der Entwicklung der Lernsoftware stieß ich hier auf die Schwierigkeit, für diese Wörter verständliche Synonyme zu finden, also, um im Beispiel zu bleiben, „Gespräche“ als Ersatz für „Kommunikation“.

Aber nicht nur die einzelnen Wörter, sondern auch der Satzbau und der Aufbau der Texte muss an die Zielgruppe angepasst werden. Bei einem Lernangebot für Kinder sollten die Sätze nicht zu kompliziert oder zu lang sein. Außerdem sollte die Sprache anschaulich sein und viele Beispiele oder Analogien aus dem alltäglichen Leben verwenden.

Schließlich sollte auch noch die Darbietung der Inhalte zielgruppengerecht sein. Wo für Erwachsene oder Experten eines Gebiets eine schlichte Grafik mit einem kurzen Text aus-

reichen kann, um einen Sachverhalt zu erklären, kann für Kinder oder Laien eine anschauliche Animation mit Sprechertext sinnvoller sein. Insgesamt sind lange theoretische Texte für Kinder weniger geeignet.

Ein wichtiger Faktor bei der Planung der Inhalte ist das *Vorwissen* der Lernenden. Deswegen sollten dazu genaue Recherchen erfolgen. Dies geht natürlich umso besser, je homogener die Zielgruppe ist. In meinem Fall sind die Voraussetzungen relativ einfach zu bestimmen, da ja alle Kinder zuvor die Grundschule besucht haben und daher in etwa das selbe Grundwissen haben. Natürlich haben manche noch zusätzliches Wissen, das sie sich andersweitig angeeignet haben, aber zumindest von einem Grundwissen zum Thema kann man ausgehen.

Um die Vorkenntnisse zu bestimmen, habe ich einige Grundschulbücher aus Bayern durchgelesen. Dabei konnte ich feststellen, dass Grundlagen zu Licht und Farben in der dritten und vierten Klasse schon behandelt wurden. Dazu gehören auch schon fortgeschrittene Themen wie die Lichtaufspaltung am Prisma oder das Bestimmen des Brennpunktes einer Linse.

Bei Lernsoftware muss man außerdem die *Medienkompetenz* berücksichtigen, also der Grad, in dem der Lernende sich mit dem Trägermedium (dem Computer) und dessen Bedienung auskennt. Geübte Benutzer finden sich auch in neuen Anwendungen schnell zurecht, ungeübte brauchen länger dazu oder lernen bei komplizierten Systemen nie alle Möglichkeiten kennen. Deswegen muss für meine Zielgruppe die Bedienbarkeit des Programms recht einfach gestaltet werden. Allerdings haben, laut Aussage einer Grundschullehrerin, schon viele der Kinder Erfahrung im Umgang mit dem Computer. Daher muss nicht erst erklärt werden, wie die Maus funktioniert oder ähnliches.

10.4 Recherche

Nach den Überlegungen zu Curriculum, Unterrichtsplanung und Vorwissen der Zielgruppe beginnt die Recherche. Sie unterteilt sich in zwei Bereiche: die Suche nach Informationen zu den gewählten Inhalten und die Suche nach Visualisierungen zu den fertigen Texten.

Als Quellen für die Texte dienten mir Physikbücher und Lexika, diverse Experimentierbücher für Kinder und das Internet. Eine genaue Auflistung findet sich im Anhang. Besonders hilfreich war hier der Karlsruher Physikkurs, der an der Universität Karlsruhe, Abteilung Didaktik der Physik entwickelt wurde. Ausgehend von diesen Unterlagen und der Gliederung konnten jetzt zu jedem Kapitel kurze Texte geschrieben werden. Jeder Text existiert in einer Sprecher- und einer Schrift-Variante. Das liegt daran, dass der Sprechertext zusätzlich zu den Informationen einige Füllwörter und Beispiele etc. enthält, die im geschriebenen Text nicht vorkommen sollen, da dieser sonst zu lang wird.

Für jeden der Texte wurde anschließend eine Liste von Assets erstellt, die die Inhalte passend begleiten oder veranschaulichen sollten. Die folgende Abbildung zeigt einen kurzen Textausschnitt mit der dazugehörigen Asset-Liste.

Es gibt eine ganze Menge von verschiedenen Lichtquellen: Die Sonne, Glühlampen, Leuchtschrift usw. Bestimmt kennst Du noch viele andere! Überleg Dir mal ein paar, während ich hier meine Sachen sortiere...ah ja...da ist das...	
Aber nicht alles, was leuchtet, ist eine Lichtquelle. Zum Beispiel ein Spiegel: wenn man ihn mit einer Lampe anstrahlt, sieht es so aus, als würde er selbst Licht erzeugen. Dabei sieht man nur die gespiegelte Lampe!	
fade	Lichtquellen (Sonne, Glühlampe, Leuchtschrift) [fade_licht]
scr	Professor kramt in Sachen
exp	Licht auf Spiegel -> der leuchtet nur, wenn Licht an ist [spiegel]

Abb. 20: Textausschnitt und zugehörige Assetliste

Zu jedem Asset wird angegeben, von welchem Typ es ist, zum Beispiel „fade“ für eine Überblendung von Bildern, „scr“ für Dinge, die bei der Professor-Figur passieren oder „exp“ für ein virtuelles Experiment. Der Text zusammen mit der Liste stellt schon eine Art Vor-Drehbuch dar. Tatsächlich existiert diese Darstellung der Inhalte zusätzlich zu dem eigentlichen Drehbuch, da dieses ziemlich unübersichtlich geworden wäre, wenn die kompletten Texte darin eingebaut wären. Eine Zuordnung zwischen Drehbuchseite und Text erfolgt über eine fortlaufende Seitennummer im Drehbuch und die entsprechende Textbenennung.

10.5 Drehbuch

Das Drehbuch (oder auch Screenbook) stellt die einzelnen Seiten des Angebots und deren Verknüpfung dar. Eines anzufertigen kann recht aufwendig sein, besonders wenn man jeden Screen (jede Seite der Software) grafisch darstellt. Laut Götz&Häfner (1992 [vgl. Bruns, 2000]) benötigt die Erstellung 40% der Gesamtentwicklungszeit. Diese Zahl halte ich für etwas zu hoch gegriffen, meine Vermutung ist, dass in diese Zahl alle voran gegangenen Konzeptionsschritte hier mit einfließen.

Um den trotzdem hohen Aufwand zu beschränken, habe ich den grafischen Entwurf und das Drehbuch voneinander getrennt, das heißt im Drehbuch stehen nur die einzelnen Inhalte etc., ohne dass deren grafische Anordnung ersichtlich wird. Diese wird im Styleguide genau beschrieben. Das Drehbuch ist zusammen mit dem Styleguide die Grundlage für die Produktion. Die folgende Abbildung zeigt, passend zum vorherigen Beispiel, eine Seite des Drehbuchs.



Abb. 21: Drehbuchseite mit Erklärungen: (1) vorhandene Medienformen, (2) allgemeine Informationen, (3) ausgeführte Skripte, (4) Navigationsmöglichkeiten, (5) Interaktionsmöglichkeiten

Wie man sieht, enthält das Drehbuch zahlreiche Informationen: Welche Medienformen vorhanden sind und welche Assets genau eingebunden werden sollen, allgemeine Informationen zu der Seite, ausgeführte Skripte und eine Beschreibung der Navigations- und Interaktionsmöglichkeiten. Theoretisch sind sogar noch viel mehr Informationen denkbar. In erster Linie diente mir das Drehbuch zur Orientierung während der Produktion und zur Koordination zwischen den Text-Dateien und deren Position im Programm. Im Verlauf der Produktion haben sich allerdings noch einige Dinge verändert. So wurden neue Assets hinzugefügt, eine Seite auf zwei aufgeteilt oder zusätzliche Verlinkungen geschaffen. Obwohl ich mich also nicht genau an das Drehbuch halten konnte, hat es trotzdem sehr viel zu einer gut organisierten und übersichtlichen Arbeit beigetragen.

11 Produktion der Assets & Integration

In dieser Phase wird der Rahmen der Software erstellt, also die einzelnen Seiten für die Inhalte und die anderen Funktionen im Autorensystem angelegt. Dabei werden der Styleguide und das Layout berücksichtigt. Außerdem werden die in Recherche ausgesuchten Assets produziert und anschließend in die Lernsoftware eingebaut (integriert).

In diesem Kapitel wird zunächst die Produktion der einzelnen Teilmedien - Sprechertext, Grafiken, Fotos, Animationen und Experimente - beschrieben. Anschließend wird auf das Erstellen des Rahmens und die Integration der Assets eingegangen. Tatsächlich liefen diese beiden Prozesse parallel ab, es wurde kapitelweise erst die Assets erstellt, die dann in die Software eingebaut wurden.

11.1 Sprechertext

Der Sprechertext war am aufwendigsten zu produzieren. Das lag zum einen daran, dass ich mit einem Sprecher zusammenarbeiten musste und zum anderen daran, dass die Bearbeitung von Sounddateien erheblich mehr Zeit in Anspruch nimmt, als die von Grafiken.

Das große Problem war es hier, einen geeigneten Sprecher zu finden. Das Hauptkriterium für die Auswahl war eine tiefe, freundliche Stimme, wenn möglich ohne Akzent oder Dialekt. Ein professioneller Sprecher wäre mit Sicherheit am besten gewesen, allerdings konnte ich keinen finden, der gegen eine nur geringe Bezahlung bereit gewesen wäre, die Texte zu sprechen. Deswegen musste ich auf mir bekannte Personen ausweichen. Bei einem kommerziellen Projekt wäre es sehr viel einfacher gewesen, einen Sprecher zu finden, da das einzige Problem das fehlende Budget war.

Der Sprechertext musste als erstes aufgenommen werden. Dazu standen ein sehr gutes Mikrofon (AKG C2000B Kondensatormikrofon) und ein kleines Mischpult (Phonic MM 1002) zur Verfügung, aufgenommen wurde auf MiniDisc. Bei der Aufnahme wurde darauf geachtet, die Soundqualität möglichst optimal zu halten, um den Aufwand für späteres Nachbearbeiten zu minimieren.

Von MiniDisc wurde der Sound anschließend in den Computer gebracht. Mit ProTools, einem Soundbearbeitungsprogramm, konnten die Textstücke eingelesen und im Wave-Format gespeichert werden. Dieses Format wurde deswegen gewählt, weil es die Dateien unkomprimiert speichert. Es gehen also keine Informationen verloren, die Qualität wird nicht schlechter.

Mit Adobe Premiere wurden diese großen Soundfiles in einzelne Sätze zerlegt. Dabei wurde auch die beste Version von mehrmals gesprochenen Texten ausgewählt. Diese Einzelteile wurden zu sinnvollen Abschnitten zusammengefasst, die etwa fünf Sekunden lang sind.

11.2 Grafiken

Die Grafiken wurden direkt in Flash erstellt, da dieses Programm vielfältige Zeichenfunktionen zur Verfügung stellt. Für die Darstellung von Gegenständen wurde eine räumliche Sicht gewählt. Diese kann mit Hilfe von Fluchtlinien konstruiert werden, Licht und Schatten verstärken den Eindruck der Tiefe und machen das Ganze plastischer und auch weniger steril.

Gegenstände, die besonders realistisch erscheinen sollten, wurden mit einer Digitalkamera fotografiert, in Flash importiert und dort in Vektorgrafiken umgewandelt. Dazu bietet das Programm die Funktion „Bitmap nachzeichnen“, die automatisch die Kanten von Objekten erkennt und in eine Vektordarstellung umrechnet.

Die Grafiken haben entweder eine visualisierende Funktion, die Inhalte vermittelt oder eine dekorierende Funktion, die das Lernprogramm optisch ansprechender macht.

11.3 Fotos

Bei Fotografien besteht entweder die Möglichkeit, sie selbst zu erstellen, oder sie von Anderen zu beziehen. Dabei muss man darauf achten, ob die Verwendung der Bilder mit Kosten verbunden ist.

Als Quelle für Fotos diente mir hauptsächlich eine Photodisc aus der Micrografx Graphics Suite, die Bilder enthält, die ohne weitere Lizenzgebühren verwendet werden können. Meistens mussten die Bilder noch bearbeitet werden, damit sie sich gut in meine Software einfügten. Motive, die auf der CD nicht vorhanden waren, fotografierte ich mit einer Digitalkamera selbst oder ließ sie von anderen fotografieren. Dies war zum Beispiel bei den Bildern eines Siliziumkristalls der Fall, da ich nicht die Gelegenheit hatte, die Fotos selbst zu machen. Auch diese Bilder mussten noch aufbereitet werden.

In zwei Fällen war es mir auf keine der oben genannten Arten möglich, ein Foto zu bekommen, nämlich bei einem Bild der Sonnenoberfläche und einer Fotografie von Einstein. Deswegen musste ich die Bilder aus dem Internet besorgen. Bei diesen Bildern ist die Frage der Rechte ungeklärt. Allerdings handelt es sich bei meinem Lernprogramm um ein nicht-kommerzielles und auch nicht-öffentliches Produkt. Aus diesem Grund können die Bilder auch einfach so verwendet werden. Für den Fall, dass das Programm eines Tages veröffentlicht wird, habe ich mir die Internetadressen notiert, um dann dort nachfragen zu können.

Die Bilder wurden als unkomprimierte Bitmaps oder PNGs (Portable Network Graphics) gespeichert, damit keine Qualität verloren geht.

Manche Fotos waren auch nur eine Grundlage für Grafiken. Die Bilder haben meist eine dekorierende Funktion, oft zeigen sie Beispiele für die behandelten Inhalte. So dienen sie dazu, die Informationen konkreter und anschaulicher zu machen.

11.4 Animationen

Genauso wie die Grafiken wurden auch die Animationen direkt in Flash produziert, einige enthalten vorher erstellte Fotos oder Grafiken. Um eine Animation zu erzeugen, müssen die einzelnen Bilder einfach nur in der Zeitleiste angeordnet werden. Das Programm kann auch Tweenings erzeugen. Dabei wird ein Start- und ein Zielbild angegeben, Flash interpoliert die dazwischenliegenden Frames. So können zum Beispiel Bewegungen, Überblendungen oder Zooms schnell realisiert werden.

Ich verwende zwei verschiedene Arten von Animationen. Die erste ist recht einfach: Bilder - meist Fotos, aber auch Grafiken oder Animationen - werden überblendet. Diese Animationen haben eher eine dekorierende Funktion, sie bebildern den Sprechertext.



Abb. 22: Bildfolge „Lichtquellen“

Die zweite Variante sind von Fall zu Fall unterschiedliche Animationen: ein Kaninchen hüpft in das Bild, Wasserstoffteilchen verschmelzen zu Helium, Licht breitet sich im Zimmer aus usw. Diese Animationen sind schwieriger herzustellen und benötigen auch mehr Zeit, denn jede von ihnen muss einzeln konzipiert werden. Diese Animationen haben meist eine visualisierende Funktion, sie stelle Sachverhalte erklärend dar. Ein gutes Beispiel dafür findet man im Kapitel „Leuchtstoffröhren“.

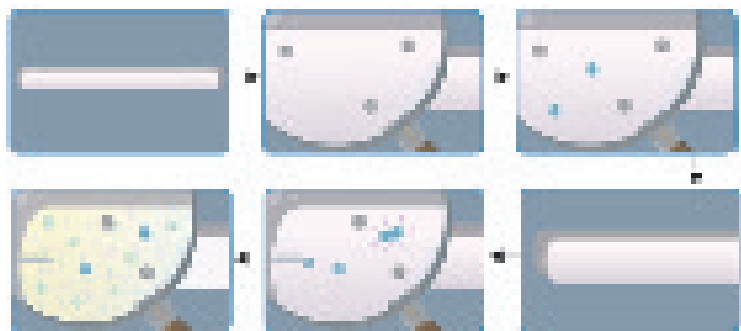


Abb. 23: Bildfolge „Leuchtstoffröhre“

Eine ganz besondere Animation stellt der Professor dar. Die Figur kann blinzeln, die Pupillen bewegen, die Arme heben und winken, mit dem Mund Sprechbewegungen machen und mit dem Schnurrbart wackeln. Um das alles möglich zu machen, wurden die beweglichen Teile als einzelne Grafiken erstellt und zu der Figur zusammengesetzt. Mund und Augen sind allerdings keine Grafiken, sondern schon kleine Animationen, die immer im Kreis ablaufen (loopen). Abbildung 24 zeigt die Bildfolge der Mund-Animation.

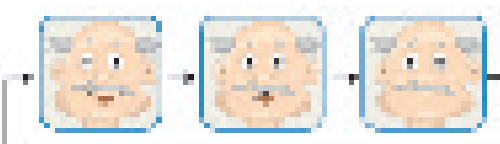


Abb. 24: Bildfolge: Mundanimation

Die Bewegungen von Armen, Schnurrbart und teilweise Mund sind nicht gezeichnet, sondern werden durch Actionscript gesteuert. Die Skriptsprache bietet die Möglichkeit, Objekte zu drehen oder zu positionieren. So wird der Schnurrbart jedes Mal, wenn der Professor anfängt, zu sprechen, um 5 Grad gedreht und kurz danach wieder zurückgesetzt. Bei Sprechpausen wird die Mundbewegung angehalten, beginnt der Sprechertext wieder, so spielt sie wieder weiter. Bei der Armbewegung wurden außerdem Zufallsvariablen verwendet, so dass die Bewegung immer zu anderen Zeitpunkten oder unterschiedlich stark erfolgt. Hätte man dies alles fest animieren wollen, wäre der Aufwand viel größer gewesen. Außerdem bietet die Zufälligkeit eine Abwechslung.

11.5 Experimente

Die Experimente wurden ähnlich produziert wie die Animationen, allerdings besteht zusätzlich die Möglichkeit zur Interaktion. Deswegen müssen in die Animationen an einigen Stellen Aktionen eingebaut werden, die auf Benutzereingaben reagieren. Dazu stehen in Flash Buttons zur Verfügung. Klickt man auf so einen Button, wird die gewünschte Aktion vom Programm ausgeführt.

Die einfachste Art einer Interaktion ist, wenn das Experiment an bestimmten Stellen anhält und durch einen Klick des Benutzers wieder abspielt. Komplizierter wird es, wenn dem Anwender mehrere Möglichkeiten zur Verfügung stehen, er also zwischen mehreren Varianten wählen kann. Hier muss das Experiment verzweigt sein. Natürlich kann man auch beide Arten kombinieren, das Experiment kann streckenweise linear und dann wieder verzweigt sein.

Neben diesen Aktionen, die nur den Ablauf des Programms ändern, gibt es auch noch andere. Ein Beispiel sieht man beim Vakuum-Experiment, wo man mit einem Thermometer über verschiedene Gegenstände fahren kann und deren Temperatur angezeigt bekommt.

Die Experimente dienen dazu, das Lernen für den Anwender interessanter zu machen. Sie beziehen den Lernenden mit ein und fordern ihn zu Aktivitäten auf. Bei einigen Experimenten muss der Anwender außerdem noch Schlüsse aus seinen Beobachtungen ziehen.

11.6 Integration

Als erstes muss ein Rahmen geschaffen werden, in den die einzelnen Assets eingebunden werden können. In Flash wurde dazu als erstes das Design umgesetzt, also die Hintergründe und andere feststehende Bereiche gezeichnet und die Navigation eingebaut. Auch das grobe Gerüst der Inhalte wurde in diesem Schritt erstellt, für jeden der Hauptpunkte (Hauptauswahl, Hilfe, Gespräche etc.) und die weiteren Screens (Abspann, Anmeldung usw.) wurde ein Marker angelegt. Ein Marker ist eine Kennzeichnung in der Zeitleiste, um direkt zu einem Bereich springen zu können. Statt „gehe zu Frame Nummer 216“ kann man einfacher über „gehe zur Hilfe“ zu diesem Teil gelangen. In den Hauptfilm wurden anschließend an die festgelegten Stellen die Schrift-Texte und Bilder für die Hauptpunkte eingefügt. Zum Schluss konnte der Sprechertext integriert werden.

Für die Inhalte wurden mehrere kleinere Flashdateien angelegt, zum einen um die Ladezeiten am Anfang geringer zu halten, zum anderen weil Flash eine Begrenzung der maximalen Framezahl besitzt, die ich sonst überschritten hätte. Außerdem wird so auch die Übersichtlichkeit erhöht. Der Hauptfilm lädt diese Kapitelfilme erst dann in den Player, wenn sie gebraucht werden. Die Abbildung zeigt, wie die Filme zusammen hängen.

Jeder dieser Filme wurde genauso vorbereitet wie der Hauptfilm. Anschließend wurden als erstes die Sprechertexte eingefügt, da alle anderen Inhalte zeitlich an diese synchronisiert werden. Unter Beachtung dieses Texts wurden zum Schluss der Schrift-Text und die anderen Assets auf dem Bildschirm platziert.

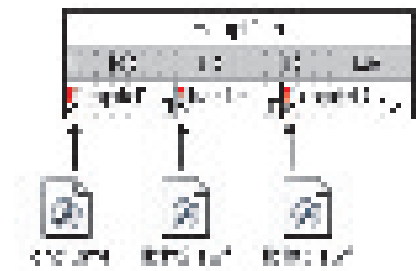


Abb. 25: Hauptfilm und nachgeladene Kapitelfilme

12 Programmierung & Fertigstellung

Anschließend an die Integration der Assets in die Flashfilme mussten den Filmen Interaktivität, Adaptivität, Kommunikations-Elemente und weitere Funktionen hinzugefügt werden. Dabei wurden verschiedene Technologien und Programmiersprachen verwendet, auf die ich im folgenden Abschnitt kurz eingehen werde. Im Anschluss daran wird die Umsetzung einiger der programmierten Funktionen erläutert, um wichtige Programmierbausteine des Projekts zu zeigen und auch, um die Komplexität einiger vermeintlich einfacher Funktionalitäten aufzuzeigen.

Der letzte Punkt der Realisierung ist das Testen und Veröffentlichen des Produkts, dass am Schluss dieses Kapitels beschrieben wird.

12.1 Verwendete Technologien

Die meisten Funktionen konnten direkt in Flash mit Actionscript umgesetzt werden. Nur für die Kommunikations-Elemente - den Chat, das Forum und den Nachrichten-Service - war eine weiter reichende Programmierung mit PHP notwendig. Nachrichten und Forum benötigen zusätzlich noch eine Datenbank zur Speicherung, hier habe ich MySQL gewählt. Diese drei Technologien werden im Folgenden kurz beschrieben.

Flash MX

Flash MX ist eine Art Autorensystem, um interaktive multimediale Anwendungen für das Internet zu erstellen. Der Hersteller des Programms meint sogar: „Macromedia Flash MX ist die schnellste Art, (...) multimediale Internet-Inhalte und -Anwendungen zu erstellen. Leistungsfähige Video-, Multimedia- und Anwendungsentwicklungsfunktionen ermöglichen das Erstellen von umfangreichen Benutzeroberflächen, Online-Werbeauftritten, interaktiven Schulkursen und Frontends für Unternehmensanwendungen.“ [Macromedia, 2002d]

Shockwave Flash ist ein vektorbasiertes Grafikformat. Vektorgrafiken speichern die Inhalte als mathematische Vektoren. Für eine Linie wird zum Beispiel nur der Anfangs- und Endpunkt sowie eventuell noch Farbe und Linienstärke angegeben. Für die Anzeige am Bildschirm

werden dann diese Vektoren in eine pixelorientierte Darstellung umgewandelt. Die Datenmengen, die bei dieser Grafikart entstehen, sind deutlich geringer als bei Pixelgrafiken. Denn bei diesen wird für jeden Bildpunkt (Pixel) ein Farbwert abgespeichert. Natürlich haben Vektorgrafiken auch einen Nachteil, den man oben schon erkennen konnte: Die mathematischen Beschreibungen müssen vom Computer in eine Pixeldarstellung umgerechnet werden. Besonders Grafiken oder auch Animationen mit vielen Elementen können so selbst einen schnellen Prozessor in die Knie zwingen, die Darstellung läuft nicht mehr flüssig ab, sondern „ruckelt“.

Durch die Darstellung im Plugin entfällt einer der Nachteile von HTML-Seiten: die Inhalte werden auf jedem System gleich dargestellt. Auch die Verwendung ausgefallener Schriftarten ist kein Problem mehr, da Flash die Schriftkonturen in den Film mit einbetten kann.

Für das Erstellen von Vektorgrafiken stellt Flash eine ganze Reihe von Zeichenwerkzeugen zur Verfügung, die dem Designer aus Programmen wie Macromedia Freehand oder Adobe Illustrator bekannt sind. Alle Funktionalitäten für das Zeichnen und Manipulieren von Objekten zu erklären, nimmt ganze Bücher in Anspruch weswegen ich an dieser Stelle auf weiterführende Literatur verweisen möchte, besonders auf Wolter (2001) und Clarkson (2001).

Neben der eingebauten Medienform „Vektorgrafik“ können außerdem Audio- und Videodateien sowie Pixelgrafiken eingebunden werden.

Für Interaktionen mit dem Benutzer stehen Texteingabefelder, Buttons und die Skriptsprache ActionScript zur Verfügung. Aktionen können vom Benutzer oder von der Filmsequenz selber ausgelöst werden. So kann der Film zum Beispiel automatisch an einer bestimmten Stelle anhalten und erst nach einem Klick des Benutzers auf einen Button wieder abspielen.

ActionScript wurde nach dem ECMAScript-Standard (ECMA-262) entwickelt, aus dem auch JavaScript entstand. Die beiden Sprachen sind sich tatsächlich sehr ähnlich, allerdings stellt Flash noch eine Vielzahl von Flash-spezifischen Funktionen zur Verfügung. So kann die Zeitleiste kontrolliert (anhalten, abspielen und springen) und auf Benutzeingaben reagiert werden. Außerdem kann man Objekte erzeugen und verändern, Daten auf dem lokalen Rechner speichern und auf verschiedene Arten mit einem Webserver kommunizieren. Da es hier weiter nicht wichtig ist, werde ich nicht weiter auf Actionscript eingehen. Mehr Informationen dazu (allerdings nur für Flash 5) findet man bei Moock (2001). Später in diesem Kapitel werden einige Funktionen der Sprache kurz vorgestellt.

PHP

PHP ist eine Skriptsprache zur Erstellung dynamischer Websites. 1994 entwickelte Rasmus Lerdorf die erste Version und stellte sie als Open Source kostenlos der Öffentlichkeit zur Verfügung. Er nannte seine Erfindung „Personal Home Page Tools“, oder abgekürzt PHP.

Später änderte sich die Bedeutung zu „PHP Hypertext Preprocessor“ [vgl. Krause, 2001]. Das Ziel von PHP ist es, Inhalte von Websites dynamisch erstellen zu können. So können zum Beispiel Kataloge für Online-Shops automatisch aus einer Datenbank generiert werden, statt mühsam von Hand erstellt zu werden. Ein weiteres Beispiel wäre eine Website, die je nach den vorhergegangenen Benutzereingaben unterschiedliche Inhalte darstellt.

Die Syntax von PHP ist an die der Programmiersprache C angelehnt, allerdings etwas vereinfacht. Da die Software von jedem weiterentwickelt werden kann, entstanden im Laufe der Zeit eine Vielzahl von Funktionen, die die Lösung von komplexen Problemen ermöglichen. So kann PHP zum Beispiel mit zahlreichen Datenbanken zusammenarbeiten, besonders häufig findet man die Kombination mit MySQL.

Aber was bringt es für eine Flashanwendung, wenn man mit dieser Skriptsprache dynamische Websites erstellen kann? Tatsächlich kann PHP noch mehr: die Ausgabe, die ein Skript erzeugt, kann zur Laufzeit in einen Flashfilm eingelesen werden. So können zum Beispiel Textfelder mit einem Text aus der Datenbank gefüllt oder Variablen auf einen bestimmten Wert gesetzt werden. Eine Anwendung dieser Möglichkeit ist im Punkt „Kommunikations-Elemente“ beschrieben.

Weiterführende Informationen zu PHP bieten Krause (2001) und www.php.net, wo man auch die aktuelle Version kostenlos herunterladen kann.

MySQL

Eine Datenbank ist, einfach ausgedrückt, eine organisierte Sammlung von Daten. Die Form der Organisation kann zum Beispiel eine Tabelle sein, man spricht dann von relationalen Datenbanken. Die folgende Abbildung zeigt eine solche Tabelle.

Name	Alter	Gewicht	Hat ein Auto
Hans	20	68	Ja
Susi	15	45	Nein
Fritz	65	68	Nein
Klara	45	57	Ja

Abb. 26: Eine beispielhafte Tabelle

Mit Hilfe von Datenbank-Management-Systemen (DBMS) kann man auf diese Daten zugreifen. Man kann Daten in Tabellen schreiben oder löschen, aber auch neue Tabellen anlegen oder alte entfernen. Weil es natürlich nicht besonders sinnvoll ist, einfach so Daten abzulegen, kann man sie auch auslesen. Dabei stehen im DBMS bestimmte Funktionen zur Verfügung, um die gewünschten Informationen zu selektieren. Man kann zum Beispiel alle Namen aus einer Tabelle suchen, die mit „A“ beginnen. Man kann auch mehrere Kriterien verknüpfen

und beispielsweise alle Personen auslesen, die ein Auto besitzen *und* jünger als 30 Jahre alt sind. Oder alle, die verheiratet *oder* leichter als 40 kg sind. Diese Auswahlkriterien müssen natürlich in der Datenbank abgelegt sein. [vgl. King, 2000]

MySQL ist so ein DBMS, das sehr gut mit PHP zusammenarbeitet. Außerdem ist es umsonst erhältlich und auf meinem Webserver installiert. Deswegen habe ich dieses System ausgewählt. Mehr zu MySQL gibt es auf www.mysql.org oder zum Beispiel in King (2000).

12.2 Kommunikations-Elemente

Im Folgenden wird die Funktionsweise der Kommunikations-Elemente Chat und Nachrichtenservice beschrieben. Das Forum wurde mir freundlicherweise von Roland Koch zur Verfügung gestellt. Deswegen kann hier nicht darauf eingegangen werden, wie es funktioniert.

Chat

Der Chat war schon aus einem früheren Projekt vorhanden. Trotzdem möchte ich kurz beschreiben, wie er funktioniert und welche Technologien dabei verwendet werden. Hier auf der Abbildung ist der Chat zu sehen. Ein Chat ermöglicht es miteinander vernetzten Benutzer an unterschiedlichen Computern, sich miteinander zu „unterhalten“, indem sie Text in ein Textfeld eingeben und abschicken. Es gibt mehrere Arten, wie ein Chat implementiert sein kann, ich verwende dazu Flash, PHP und Textdateien.

Der Chat besteht aus einem drei Bereichen. Die Benutzeroberfläche am oberen und unteren Rand besteht aus zwei Flash-filmen, diese steuern den Inhalt des mittleren Frames. Dieser wird regelmäßig jede Sekunde neu geladen, damit die neuen Texte zu sehen sind. Die Inhalte werden von einem PHP-File generiert. Dieses liegt auf dem Server und liest bzw. schreibt eine Textdatei, in der alle bisher geschriebenen Beiträge gespeichert sind. Wenn ein neuer

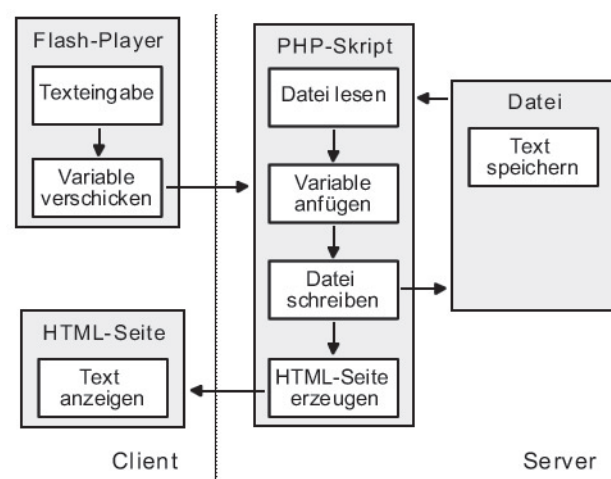


Abb. 27: Ablauf des Chats

Beitrag abgeschickt wird, wird das PHP-File aufgerufen. Es öffnet die Textdatei und fügt den neuen Text ein, der aus Flash an das Skript übergeben wurde. Anschließend wird die Textdatei in die HTML-Seite ausgegeben und die Datei wieder gespeichert. Das obige Schema (Abb. 27) zeigt den Ablauf.

Nachrichtenservice

Komplett neu entwickelt wurde der Nachrichten-Service. Er ermöglicht es den Benutzern, direkt aus dem Lernprogramm heraus Nachrichten an einen Tutor zu schicken und auch die Antwort dort zu lesen. Die Voraussetzung dafür ist eine Email-Adresse und natürlich eine bestehende Verbindung mit dem Internet. Das Versenden der Nachricht und das Abrufen der Antwort wird von einem PHP-Skript unter Verwendung von MySQL durchgeführt, das aus Flash aufgerufen wird. Das folgende Skript zeigt diesen Aufruf.

```
on (release){
    loadVariables ("http://www.greenzone.de/send.php", _root.mailMe, "POST");
    dbTimer(5);
}
```

Code 1: Abschicken einer Nachricht aus Flash

Die Benutzeraktion (User-Event), der das Skript startet ist ein Klick auf einen Button, genauer gesagt das Loslassen der Maustaste über einem Button (engl. *release*: loslassen). Der Befehl `loadVariables` lädt die Ausgabe des PHP-Skriptes als Variable in den Flashfilm. Bei diesem Aufruf werden die Variablenwerte aus Flash an das Skript übergeben.

In der nächsten Zeile startet der Befehl `dbTimer(5)` eine selbst programmierte Funktion und übergibt ihr den Wert 5. In dieser Funktion wird ein Zeitzähler gestartet, der nach der vorgegebenen Zeit (hier 5 Sekunden) eine Fehlermeldung bringt, wenn in der Zwischenzeit keine Antwort vom PHP-Skript angekommen ist. So kann der Flashfilm erkennen, ob ein Fehler vorliegt. Dies kann zum Beispiel sein, dass der Anwender nicht mit dem Internet verbunden ist oder dass der Server, auf dem das PHP-Skript liegt, ausgefallen ist. Wäre kein Timer eingebaut, so würde der Anwender überhaupt keine Rückmeldung bekommen, er wüsste nicht, was passiert. Und solche Situationen gilt es zu vermeiden. Wenn aber eine Rückmeldung vom Skript erfolgt, so wird diese in einem Textfeld angezeigt: „Die Nachricht wurde gesendet!“.

Das PHP-Skript nimmt, nachdem es vom Flashfilm aufgerufen wurde, die mitgeschickten Werte (Name, Nachricht und Email-Adresse) und speichert sie in eine Datenbank in die folgende Tabelle. Der Wert in der Spalte Antwort wird dabei auf „keine Antwort“ gesetzt.

id	name	mail	antwort
1	Susi	susi@xx.de	keine Antwort
2	Tom	t@yy.de	Ja!
3	Hans	ich@zz.de	keine Antwort
4	Susi	mail@cc.de	Nein.

Abb. 28: Tabelle für den Nachrichtenservice

Das Abrufen der Antworten erfolgt im Prinzip genauso. Der Flashfilm ruft auf die selbe Art und Weise ein PHP-Skript auf. Wieder werden Name und Email-Adresse übergeben. Das Skript schaut in der Datenbank nach, ob die mitgeschickte Email-Adresse in der Tabelle existiert.

Wenn nicht, dann ist noch keine Nachricht zu dieser Adresse vorhanden, das Skript liefert eine Fehlermeldung zurück: „Ich habe keine Nachricht von Dir erhalten!“

Ist ein Eintrag vorhanden, dann liest das Skript die passende Antwort aus und schickt sie an den Flashfilm zurück. Je nachdem, ob die Antwort ein richtiger Text ist oder nur „keine Antwort“, gibt Flash dann einen passenden Text und gegebenenfalls die Antwort aus. Das Skript schickt außerdem eine Email an den Anwender, in dem die Antwort ebenfalls zu finden ist.

12.3 Testauswertung & Adaptivität

In die Software sind mehrere Tests eingestreut, die dazu dienen, das Wissen und die Geschwindigkeit des Anwenders festzustellen und den Programmablauf entsprechend anzupassen. Man hätte auch noch andere Parameter auswerten können, zum Beispiel die Anzahl von Klicks „ins Blaue“ (also auf andere Flächen als Buttons), allerdings wäre der Aufwand sehr groß gewesen. Deswegen wird hier im folgenden beschrieben, wie der Prozess vom Lösen eines Tests, über die Kennzahlen-Ermittlung, bis hin zur Anpassung der Software abläuft.

Tests

Die Tests sind als Multiple-Choice-Fragen umgesetzt, zu denen es jeweils drei Alternativantworten gibt. Nach dem Anklicken eines Antwortbuttons kommt man zu einem Screen, auf dem die Antwort bewertet wird. Für richtige Antworten erhält man ein positives, für falsche ein negatives Feedback. Bei falschen Antworten gibt es zwei Varianten: Dem Anwender wird entweder sofort die richtige Lösung verraten oder er erhält einen kleinen Tipp und muss die Frage erneut beantworten. Dabei sind dann die vorher gewählten falschen Aussagen markiert. Welches Vorgehen man wählt, hängt mit der Art der Frage zusammen. Wenn die Lösung dem Lernenden bekannt sein müsste, so wird die zweite Variante gewählt, wenn er nur durch weiterreichendes, nicht behandeltes Hintergrundwissen auf die Lösung kommen kann, die erste.

Aber die Antwortbuttons schicken den Benutzer nicht nur in einen neuen Frame, im Hintergrund passiert folgendes: Wenn der Film zu einem Test kommt, wird die Variable `tries`, die die Anzahl der Versuche zählt, auf 0 gesetzt. Anschließend überprüft das Programm, ob vorher schon falsche Antworten gewählt wurden und markiert diese gegebenenfalls. Gleichzeitig wird ein Timer gestartet, der die Zeit bis zur Beantwortung misst. Jetzt wartet das Programm auf eine Aktion des Benutzers, nämlich darauf, dass er auf einen der Antwortbuttons klickt. Hiervon gibt es zwei verschiedene Arten: einen für richtige und einen für falsche Antworten. Sie lösen unterschiedliche Aktionen aus, wie die Codebeispiele auf der nächsten Seite für den Test Nummer 1 zeigen. Die 0 in den eckigen Klammern muss dabei bei den anderen Tests durch die jeweilige Zahl ersetzt werden, zum Beispiel 2 bei Test drei usw.

```

on (release) {
    if (!_root.tries) {
        _root.time[0] = _root.timer.timer;
        _root.timer.gotoAndStop(1);
    }
    _root.tries1++;
    gotoAndPlay("loesung _ B");
}

```

Code 2: Aktion des Buttons
einer falschen Antwort

```

on (release) {
    if (!_root.tries) {
        _root.time[0] = _root.timer.timer;
        _root.timer.gotoAndStop(1);
    }
    _root.tries1++;
    _root.tries[0] = _root.tries;
    _root.saveTest(1);
    gotoAndPlay("loesung _ A");
}

```

Code 3: Aktion des Buttons
einer richtigen Antwort

Beide Buttons speichern im ersten Abschnitt der Skripte die Dauer bis zum Klick. Dazu wird einfach der Wert des Zählers, der vorher gestartet wurde, in die Variable `time[0]` ausgelesen und der Zähler gestoppt. Dabei wird stets nur die Zeit des ersten Versuchs gespeichert, da dieser am ehesten angibt, wie lange der Lernende für das Erfassen und Lösen einer Aufgabe benötigt.

Anschließend wird die Zahl der Versuche `tries` um eins erhöht. Und hier unterscheiden sich die beiden Skripte. Im Falle einer falschen Antwort geschieht weiter nichts, bei einer richtigen wird die Gesamtzahl der Antworten in `tries[0]` gespeichert, anschließend beginnt mit dem Aufruf der Funktion `saveTest` die Auswertungsphase. Als letztes lassen beide Skripte den Film in den entsprechenden Feedback-Frame springen.

Bei Fragen, deren Lösung sofort verraten wird, werden den drei alternativen Antworten verschiedene „Falschheitsgrade“ zugewiesen, die dann genauso wie die Anzahl der Versuche behandelt wird. Eine völlig falsche und abwegige Antwort entspricht drei Versuchen, eine Antwort, die nahe an der Realität liegt oder eine häufig verbreitete Fehlvorstellung ist, wird mit zwei Punkten bewertet und die richtige ist schließlich gleichbedeutend mit einem Versuch.

Auswertung

Die im Test ermittelten Werte für Zeitdauer (`time[0]`) und Versuchsanzahl (`tries[0]`) werden in der lokalen Datei gespeichert. Aus allen Werten von `time` und `tries` werden jetzt in der Funktion `makeIndex` zwei Kennzahlen ermittelt: eine für die Geschwindigkeit, eine für

die Versuchsanzahl. Es sind deswegen zwei getrennte Kennzahlen, da die Geschwindigkeit, in der eine Aufgabe gelöst wird, keine Aussagekraft über des Wissens hat [vgl. Ausubel, 1980].

Die Kennzahlen-Ermittlung erfolgt über eine einfache Mittelwertrechnung. Die Anzahl der Versuche bzw. die Zeiten werden addiert und durch die Gesamtzahl der gelösten Tests dividiert. Der beste Wert für den Versuche-Index ist also eins, der schlechteste drei. Bei den Zeiten gibt es keinen festen obersten Wert, da man beliebig lange für eine Aufgabe brauchen kann. Der beste Wert ist auch hier eins, was bedeutet, dass der Lernende im Schnitt eine Sekunde für jede Aufgabe benötigt hat. Hier gibt es eine Besonderheit: Wenn die Durchschnittszeit unter fünf Sekunden liegt, aber im Mittel mehr als ein Versuch benötigt wurde, wird angenommen, dass der Benutzer nur geraten hat. Dadruch wird natürlich keine Aussage möglich über Wissen oder Bearbeitungsdauer des Lernenden, die ermittelten Werte werden ignoriert.

Adaptivität

Was passiert nun mit den beiden Indizes? Sie werden dazu verwendet, das Programm anzupassen. Die Wiedergabegeschwindigkeit ändert sich automatisch und es werden je nach Wissensstand alternative Sätze oder Wörter dargeboten: Beispielsweise bei einer Versuchskennzahl von 1 zusätzliche weiterführende und bei 3 zusätzliche erklärende. Der folgende ActionScript-Ausschnitt zeigt, wie die Abspielgeschwindigkeit automatisch verändert wird.

```
addTime = Math.round((indexTime-10)/3);
if (addTime>1) {
    addTime = 1;
}
warteZeit = waitingTime + addTime;
if (warteZeit<1) {
    warteZeit = 1;
}
```

Code 4: Anpassen der
Abspielgeschwindigkeit

Zunächst wird eine Dauer von 10 Sekunden als Standard für die Lösung einer Aufgabe angenommen. Durch Versuche mit der Zielgruppe könnte diese Zahl noch genauer angepasst werden. In der ersten Zeile des Skripts wird jetzt die Differenz der tatsächlichen Durchschnittszeit `indexTime` von diesem Idealwert bestimmt. Für je drei Sekunden Abweichung soll die Zeit um eine Viertelsekunde verkürzt oder verlängert werden. Dazu wird das Ergebnis der Differenz durch drei dividiert und anschließend auf eine ganze Zahl gerundet. Der Rest des Skripts dient dazu, die Gesamtzeit auf einen geeigneten Wert zu begrenzen, da sonst auch negative oder sehr hohe Wartezeiten erreicht werden können.

Das Abspielen von unterschiedlichen Textbausteinen je nach Versuchskennzahl ist durch einfache Sprungbefehle realisiert. Deswegen wird hier auch nicht näher darauf eingegangen.

12.4 Sonstiges

Neben diesen beiden Hauptteilen der Programmierung gibt es noch viele kleinere Elemente, die ich hier kurz beschreiben möchte.

Eine wichtige Funktion ist das *Speichern* des „Spielstands“ und der Benutzerdaten. Speziell handelt es sich hier um

- Namen und Email-Adresse des Anwenders (beim Anmelden eingegeben)
- schon besuchte Kapitel
- zuletzt gesehene Kapitel
- Testauswertungen und Kennzahlen für die Adaptivität
- die eingestellte Geschwindigkeit

Dazu bietet Flash MX eine einfache Methode an, die Daten auf der Festplatte des Benutzers speichert. Dazu genügen wenige Zeilen Code:

```
userData = SharedObject.getLocal("userData");
userData.data.name = "Sara";
y = userInfo.flush();
```

Code 5: Speichern des Namens
mit Flash

In der ersten Zeile wird ein sogenanntes `SharedObject` mit dem Namen `userData` erzeugt, in das die Daten gespeichert werden können. Die beiden nächsten Zeilen zeigen beispielhaft, wie diese Daten geschrieben werden. Der letzte Befehl speichert schließlich die Daten auf der Festplatte.

Ebenfalls eine wichtige Funktion ist das Einstellen der *Geschwindigkeit*. Um dies zu beschreiben, muss natürlich erst geklärt werden, wie die Anpassung der Geschwindigkeit funktioniert. Oben wurde schon beschrieben, wie der Sound bearbeitet und in das Endprodukt eingebaut wird. Er wird in ca. fünf Sekunden lange Stücke zerteilt, die dann in den Kapiteldateien hintereinander in der Zeitleiste angeordnet werden. Zwischen den einzelnen Soundteilen ist in der Zeitleiste jeweils eine Aktion eingebaut, die ausgeführt wird, sobald der Film beim Abspielen auf den entsprechenden Frame trifft.

Dabei wird der Kapitel-Teilfilm angehalten und im Hauptfile ein Sound-Timer gestartet, indem dieser in seinen zweiten Frame geschickt wird. Der Timer besteht aus einem Stop- und drei Loop-Frames, die bei einer Abspielgeschwindigkeit des Flashfilms von zwölf Frames pro Sekunde 0,25 Sekunden entsprechen.

Immer nach einer Viertelsekunde kommt der Timer also in den letzten Frame. Dort passiert - in vereinfachter Form - folgendes: Eine Zählvariable `zeit` wird um eins erhöht, anschließend wird der Wert dieser Variable mit einem vorher festgelegten Wert `warteZeit` verglichen. Wenn `zeit` gleich groß oder größer als `warteZeit` ist, springt der Timer wieder zurück

in den ersten Frame, wo er dann stoppt. Außerdem weist er den Movieclip `kapitel` an, weiter abzuspielen. Nach der in Viertelsekunden vorgegebenen Wartezeit spielt der Kapitel-Film also weiter ab. Wenn die Geschwindigkeit des Films geändert wird, wird jetzt einfach diese Wartezeit-Variable verändert. Wenn der Film langsamer werden soll, wird sie erhöht, wenn er schneller werden soll, abgesenkt.

Es wurden noch einige andere Dinge programmiert, deren Beschreibung hier aber zu viel Platz beanspruchen würde. Ich möchte mich deswegen auf diese kleine Auswahl beschränken, die einen recht guten Überblick über die wichtigsten Funktionalitäten gibt.

12.5 Testen & Distribution

Die beiden letzten Schritte der Produktion waren das Testen und Distributieren des Produkts. Der Test beschränkte sich dabei auf technische Richtigkeit und einen fehlerfreien Ablauf des Programms. Getestet wurde auf zwei Windows 98- und einem Mac OS X-Computer. Dabei wurden auch noch einige Fehler gefunden und behoben. Trotzdem kann natürlich nicht ganz ausgeschlossen werden, dass immer noch einige kleinere Fehler enthalten sind, denn eine ausführliche Testphase nimmt sehr viel Zeit in Anspruch. Ein anderes Problem ist, dass man manche Dinge nach wiederholtem Sehen nicht mehr richtig wahrnimmt und einfach übersieht. Das Testen sollte daher am Besten von mehreren, eventuell gar nicht am Projekt beteiligten Personen durchgeführt werden. Gut ist es hier, wenn eine *Buglist* angefertigt wird, auf der alle Fehler, deren genaue Position und ihr Status (noch offen, in Bearbeitung, behoben) aufgelistet sind. So haben alle am Testen Beteiligten immer einen Überblick. Für größere Projekte mit vielen Teilnehmern gibt es sogar eigene Anwendungen, die das Anlegen und Verwalten von Buglists zur Aufgabe haben.

In der abschließenden Phase, der Distribution, wird das fertig Produkt der Öffentlichkeit zugänglich gemacht. Offline-Produkte werden auf Datenträger gebracht, Online-Versionen auf einen Webserver im Intra- oder Internet übertragen. Bei kommerziellen Offline-Anwendungen werden die Datenträger (meist CDs) über Versand- oder Einzelhandel verkauft.

Bei mir beschränkte sich die Distribution auf das Brennen von einigen CD-ROMs und deren Versand an die beiden Betreuer. Die CDs wurden so gebrannt, dass sie sowohl von Windows- als auch von Macintosh-Systemen gelesen werden können. Außerdem wurde die Online-Version auf einen Server gespielt, sie kann unter der Adresse <http://www.greenzone.de/optik> aufgerufen werden. Dabei wird das Flash-Plugin in der neuesten Version benötigt.