

Phosphor – Weiterentwicklung animierter Übergänge

Jochen Lohrmann

Proseminar “Mensch-Computer-Interaktion”

Lange-Reute-Strasse 16
89160 Temmenhausen
jochen.lohrmann@uni-ulm.de

Einleitung

Wem ist es nicht schon mal so ergangen: Man arbeitet am PC und verändert versehentlich eine Einstellung oder verschiebt ein Icon oder einen Ordner. Doch nun stellt sich die Frage: Wo ist der Ordner hin, welcher Regler wurde verschoben oder habe ich in einer Checkbox einen Haken gesetzt oder entfernt? Helfen könnten hierbei animierte Übergänge welche die Veränderungen anzeigen. Der *Windows Media Player 10* hat zum Beispiel solch einen animierten Übergang. Im Fullscreen-Modus werden die Bedienelemente langsam aus dem Bildschirm geschoben. Damit weiß man wohin diese Bedienleiste verschwunden ist. Doch in Situationen, in denen man diese Übergänge nicht benötigt (wenn man möglicherweise schon weiß wohin die Bedienelemente verschwinden, den Ordner, das Icon verschieben wollte, usw.) zwingen sie den Benutzer zum Warten bis die Animation vorbei ist. Das kann sehr nervenaufreibend und zeitaufwendig sein. Deswegen wurde Phosphor entwickelt. Es zeigt die Veränderung mit einem grünen Schimmern an, das langsam verschwindet. Dadurch kann man weiterarbeiten und muss nicht darauf warten bis die Animation vorbei ist. Trotzdem sieht man sofort was verändert wurde und kann dies, falls gewünscht, wieder rückgängig machen. Dadurch erhält man den Vorteil von animierten Übergängen, das verzögerte Erkennen von Veränderungen, aber der Nachteil, die Wartezeiten entfallen. In dieser Ausarbeitung wird Phosphor vorgestellt, sowie mögliche unterschiedliche Animationsvarianten und –möglichkeiten, welche man einsetzen könnte. Zudem werden noch zwei Studien gezeigt, die sich mit dem Thema befassen haben.

Die Phosphor-Oberfläche

Die Phosphor-Oberfläche (siehe Fig.1.) ist dafür konzipiert Veränderungen über die eigentliche Aktion hinaus anzuzeigen. Also auch noch kurze Zeit nach der Ausführung der Änderung (verschieben eines Reglers,...) diese sichtbar mit einem grünen Schimmern anzuzeigen, das nach und nach verschwindet. Somit weiß der Benutzer, was gerade verändert wurde und wie die ursprüngliche Situation war. Natürlich verschwindet der Afterglow-Effekt nach einer Weile, da er sonst eher zu Verwirrung führen könnte. Außerdem zeigt die Stärke des Glühens die zeitliche Abfolge an, in der die Aktionen ausgeführt wurden. Damit kann der Benutzer seine letzten Schritte

2 Proseminar “Mensch-Computer-Interaktion”

nachvollziehen. Weil man nicht warten muss, bis das Glühen verschwunden ist kann man ohne Verzögerungen weiterarbeiten. Somit ist der Arbeitsfluss nicht gestört, wie das bei Animationen der Fall ist. Weil der Benutzer nicht warten muss, bis der Effekt vorbei ist, wie es bei Animationen der Fall wäre, ist es nicht so schlimm wenn er länger dauert. Dadurch entsteht der Vorteil, dass man keine optimierten Effektzeiten benutzen muss. Es reicht, wenn eine obere Grenze eingeführt wird, die verhindert, dass der Afterglow-Effekt nicht störend wirkt.

Dass die Gestaltung eines Afterglow-Effekts nicht immer einfach ist und oft ein gewisses Maß an Kreativität fordert, sieht man an Comboboxen und Checkboxes. Damit der erzeugte Effekt leicht verständlich und nicht eher verwirrend ist, muss man dem Effekt das richtige Aussehen verleihen. Wenn man zum Beispiel einer Combobox einen Afterglow-Effekt hinzufügen möchte, reicht es nicht, die alte Auswahl grün schimmernd in die Combobox zu setzen. Dies würde zu Verwirrung führen, weil die neue Auswahl ja auch drin steht und der Effekt diese überdecken würde. Genauso schwer ist es wenn eine Combobox selektiert werden soll. Man muss ja anzeigen, dass vorher kein Haken gesetzt war. Diese Probleme mussten gelöst werden (siehe Fig. 1.). Dagegen ist die Gestaltung bei Schiebereglern eher einfach: vom alten Standpunkt des Reglers bis zum neuen wird das Schimmern erzeugt.

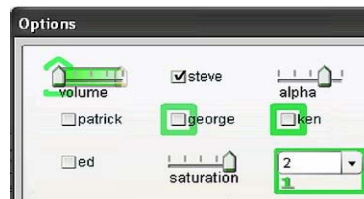


Fig. 1. Der Phosphor-Effekt angewendet auf Checkboxes, Comboboxen und Schieberegler

Andere Einsatzgebiete

Die Einsatzmöglichkeiten von Phosphor sind allerdings nicht nur auf solche simple grafische Objekte wie Schieberegler, Checkboxes und Comboboxen beschränkt. Auch Icons, Fenster, usw. können mit Phosphor einen Afterglow-Effekt erhalten. Hierbei wird jedoch nicht mehr das grüne Schimmern verwendet wie zuvor. Stattdessen wird ein animierter Übergang genommen und dieser in mehrere Einzelbilder aufgeteilt. Aus diesen wird dann ein einzelnes Bild generiert, das die Veränderung in einem Effekt darstellt (dargestellt in Fig. 2.).

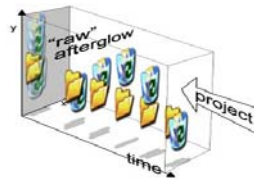


Fig. 2. Erstellung des Phosphor-Effekts

Effekte, welche man einsetzen könnte, sind etwa der „strobe“-Effekt (Fig. 3.a) oder „motion blur“ (Fig. 3.b). Da diese Übergänge allerdings nicht immer ganz eindeutig sein können, wird bei Phosphor jedoch meist ein dritter Effekt benutzt, die „speed lines“ (Fig. 3.c). Wie die anderen auch wurde dieser Effekt Comic-Darstellungen entnommen. Er zeigt die Bewegung mit mehreren Linien von dem aktuellen Standort zum alten Standort hin an.

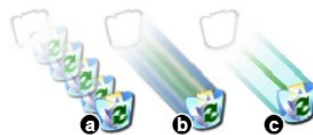


Fig. 3. Die unterschiedlichen Effekte: a) strobe, b) motion blur und c) speed line Effekt

Um die zeitlichen Abfolgen darzustellen, verschwindet der Pfad langsam von seinem Beginn an. Natürlich liegen auch neuere Bilder (zum Beispiel beim „strobe“-Effekt) über den alten.

Um die unterschiedlichen Gründe für Übergänge (kopieren, verschieben, extrahieren,...) darzustellen wurde das Aussehen der Übergänge verändert. Wenn das Objekt an der ursprünglichen Stelle noch erhalten bleibt, so wie beim Kopieren, Extrahieren, usw., sieht man am Beginn der Spur das ganze Icon (Fig. 4.b). Dafür bleibt nur ein Umriss übrig, wenn das Objekt am Ende nicht mehr am alten Platz vorhanden ist (z.B. beim Verschieben), wie in Fig. 4.a.



Fig. 4. Mögliche Darstellung eines “move”- (a) oder “copy”-Übergangs (b)

Falls das Objekt nur temporär verschoben wurde, gehen die „speed lines“ in der Mitte etwas zusammen. Dies soll darstellen, dass das Icon an der alten Stelle kleben bleibt

4 Proseminar “Mensch-Computer-Interaktion”

(Fig. 5.). Der Benutzer soll dadurch erkennen, dass das Objekt diese Aktion nur für eine Weile ausführt und dann der alte Zustand wieder entsteht.

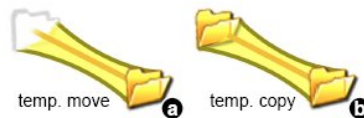


Fig. 5. Im Vergleich zu Fig. 4. dagegen die zeitlich Effekte bei temporärer Ausführung

Indem die „speed lines“ in eine Richtung enger werden, soll verdeutlicht werden, dass ein Objekt in einem anderen verschwindet (z.B. beim verschieben in einen Ordner) oder aus ihm herauskommt (beim extrahieren). Es wirkt dann so wie wenn das Objekt schrumpft oder wächst und somit in das andere Objekt hinein passt bzw. passte (Fig. 6.).



Fig. 6. Das “Schrumpfen” und “Wachsen” des Objekts beim einordnen oder extrahieren

Es gibt jedoch auch Übergänge ohne Bewegungen. Manchmal liegt ein Icon hinter dem anderen und wird in den Vordergrund geholt. Hierbei bleibt das Objekt stehen und deswegen können keine Animationen und „speed lines,, eingebaut werden (Fig. 7.a). Dafür wurde eine vertikale Bewegung eingeführt, so dass es aussieht, als ob das hintere Icon über das vordere hinweg springt (Fig. 7.b+c). Auf diese Bewegung kann man eine Animation anwenden, die verdeutlicht von wo das Objekt herkommt und dass es nicht einfach plötzlich da war. Jedoch kann dies auch weglassen werden, wenn man annehmen kann, dass der User weiß woher das Icon kommt.

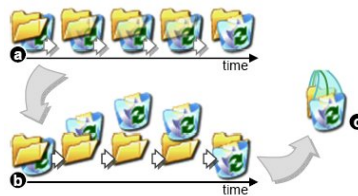


Fig. 7. Effekt wie er normalerweise auftreten würde (a) und mit “überspringen” (b + c)

Wenn Objekte die Plätze tauschen, könnte dies zu Verwirrung führen, da die „speed lines“ sich überlagern aber jeweils in die entgegen gesetzte Richtung wirken. Deswegen sollte man die Effekte in diesem Fall so führen, dass sie sich nicht gegenseitig überlagern, sondern zum Beispiel jeder einen Bogen in eine andere Richtung beschreibt. Genauso sollte man die Effekte verändern, wenn man viele Objekte bewegt. Wenn diese alle in die selbe Richtung und auf einmal bewegt werden könnte man sie als Block sehen und einen großen Effekt auf sie wirken oder, bei mehreren unterschiedlichen Bewegungen, einfach den Pfad kürzen.

Studien

Es wurden zwei Studien mit Phosphor durchgeführt. In der ersten Studie sollte überprüft werden, ob Benutzer Veränderungen mit Phosphor besser erkennen. Dafür wurde die Oberfläche von Fig. 1. benutzt, wobei die Anordnung variiert wurde. 500 Millisekunden nachdem jeder der 12 Uni-Studenten einen „Go“-Button geklickt hatte wurden eine Reihe von Veränderungen automatisch ausgeführt, dabei gab es unterschiedliche Geschwindigkeiten (slow, medium, fast). Danach waren 4000 ms Pause damit die Afterglow-Effekte verschwinden konnten. Der Benutzer sollte so viele Veränderungen wie möglich rückgängig machen. Wenn er dies gemacht hatte, dann musste er „Ok“ klicken. Die Zeit und die Anzahl der Fehler wurden gemessen und ausgewertet. Mit Phosphor waren die Studenten signifikant schneller als ohne diese Übergangseffekte. Die Fehleranzahl war bei beiden Situationen ungefähr gleich. Bei einer Befragung gaben alle Probanden an, dass sie Phosphor bevorzugen würden.

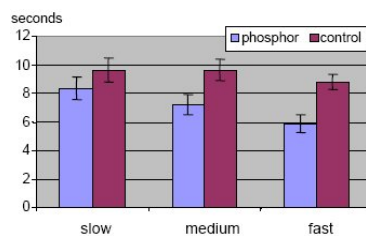


Fig. 8. Darstellung der Zeiten beim Bearbeiten der ersten Studie

Bei der zweiten Studie sollte überprüft werden ob Phosphor im Vergleich zu Animationen (in vier verschiedenen Geschwindigkeiten), zu einer schlechteren Performance führt. Dafür wurde simuliert, dass Icons in ein Fenster verschoben werden und ein Benutzer die Aktion auf ihre Richtigkeit prüft. Bei der Simulation wurden 11 kleine Fenster und 25 gleiche Icons angezeigt. Dann wurde für 1200 ms ein Fenster und drei Icons hervorgehoben und danach wieder entfernt. Drei Icons wurden dann bewegt. Der Benutzer musste dann entscheiden, ob die drei hervorgehobenen Icons in das hervorgehobene Fenster geschoben wurden. Dachte er, dass es richtig war, sollte er „Y“ drücken ansonsten „N“. Bewertet wurden mehrere Situationen. Entweder es stimmte alles, ein falsches Icon wurde bewegt oder eines der drei Icons ging nicht ins Fenster. Wenn ein falsches Objekt bewegt wurde, wurde zudem noch unterschieden ob es von

6 Proseminar “Mensch-Computer-Interaktion”

dem hervorgehobenen Icon weit weg oder in dessen unmittelbarer Nähe war. Dasselbe wurde noch statt mit drei Icons mit einem gemacht oder es wurde eines hervorgehoben und drei Objekte bewegten sich. Zwölf Probanden nahmen an der Studie teil. Am Ende hatte Phosphor fast jedesmal eine bessere Performance und nur bei der schnellsten Animation konnte kein signifikanter Unterschied festgestellt werden. Aber es wurden auch Probleme bei langen Pfaden festgestellt, da die Benutzer diese nicht ganz erfassen können.

Zusammenfassung

Phosphor ist ein interessanter Ansatz, um grafische Übergänge so darzustellen, dass sie auch kurze Zeit später noch nachvollziehbar sind. Auch das Problem, dass man auf den Übergang warten muss, wie bei Animationen, wurde gut gelöst. Wie die vorgestellten Studien auch gezeigt haben, hilft Phosphor Veränderungen an der grafischen Oberfläche zu erkennen und zu merken. Dabei ist die Performanz nicht signifikant schlechter als würde man mit Animationen arbeiten, sondern meist sogar besser.

Dieser Ansatz ist ein starker Konkurrent zu Animationen, wie ich finde. Die Entwickler pickten die Vorteile von Animationen heraus und entfernten einen großen Nachteil. Man sieht Veränderungen schnell und muss trotzdem nicht warten bis die jeweilige Animation vorbei ist. Jedoch könnte es schnell unübersichtlich werden, wenn man eine große Arbeitsfläche hat und viele Bewegungen darauf ausführt. Dies kann jedoch möglicherweise durch Optimierung der Pfadlänge behoben werden. Meiner Meinung nach sind manche dieser Vorschläge überflüssig. Ich glaube zum Beispiel, dass man nicht unbedingt wissen muss, ob ein Objekt vorher hinter einem anderen stand oder nicht. Bei normalen grafischen Oberflächen wäre wohl das verschieben oder kopieren in einen Ordner ein sinnvolles Einsatzgebiet. Bei den restlichen Möglichkeiten (einfaches verschieben, temporäres kopieren,...) ist es eher Spielerei und dafür da, dass es gut aussieht. Alle Möglichkeiten dieser grafischen Übergänge, die vorgeschlagen wurden, könnten bei Desktops und Programmen, an denen mehrere Benutzer mit Remote Control arbeiten, sehr sinnvoll sein. So kann jeder Benutzer erkennen, was gerade verändert wurde. Auch wäre der Einsatz von unterschiedlichen Farben für jeden Benutzer denkbar, so dass man auch noch erkennen könnte, wer was verändert hat. Jedoch ist das ganze Projekt bis jetzt noch Zukunftsmusik und wird noch nicht eingesetzt. Es wird aber weiter daran geforscht.

Literatur:

1. Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. UIST '06, October 15-18, 2006, Montreux, Switzerland, Seiten 169-178