

MagicColors

Denis Zwornarz

COLLABORATORS

	TITLE : MagicColors		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Denis Zwornarz	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MagicColors	1
1.1	MagicColors	1
1.2	einleitung	1
1.3	scripts erstellen	2
1.4	befehle	3
1.5	starten	5
1.6	speicherverbrauch	6
1.7	anforderung	6
1.8	installation	6
1.9	hintergrundbilder	7
1.10	fehlerberichte	7
1.11	autor	8
1.12	history	8
1.13	zukunft	8
1.14	registrieren	8
1.15	rechtliches	9

Chapter 1

MagicColors

1.1 MagicColors

MagicColors v1.0 (27.8.1996)

Copyright ©1996, Denis Zwornarz
All rights reserved

Einleitung
System Anforderungen
Installation
MagicColors starten

Eigene Scripts erstellen
Hintergrundbilder

Der Autor
Registrierung
Rechtliches
Fehlerberichte
Was bisher geschah
Was noch kommt

1.2 einleitung

Einleitung
=====

Sinn und Zweck von MagicColors ist es, die Arbeit auf der Workbench etwas sehenswerter zu machen. MagicColors macht dabei nichts anderes, als die Farben der Workbench zu verändern. Wie dabei die Farben manipuliert werden, wird in einem Script (Programmablauf) vom Benutzer festgelegt. Ob man nun eine 'glühende' Workbench, einen Wasserfall, ein Morphing oder was auch immer haben will, hängt nur von der eigenen Phantasie ab.

Im Gegensatz zu echten Pixelanimationen werden hier nur die Farben eines Bildes

verändert. Die Rechenauslastung hängt zwar sehr stark von der eigenen Hard- und Software Konfiguration ab, ist aber in der Regel bei weitem nicht so hoch, wie bei einer Pixelanimation.

1.3 scripts erstellen

Scripts erstellen

=====

Das erstellen eigener Scripts kann, jenachdem man welchen Effekt erzielen will, einfach aber auch ziemlich komplex (nicht kompliziert) werden. Wer also seine eigenen Scripts schreiben will, sollte auf jeden Fall einen Blick auf die beiliegenden Beispiel Scripts werfen.

Zuerst möchte ich hier einmal die Arbeitsweise von MagicColors erklären. Als erstes wird das angegebene Script geladen und analysiert. Anschliessend erstellt MagicColors anhand des Scriptes einen Programmablauf, wo festgelegt wird wo, wann welche Farbe wie manipuliert wird.

Um nun MagicColors sagen zu können wie er die Farben zu verändern hat, erstellt man ein Script, in dem dies mit einigen wenigen Befehlen beschrieben wird. So ein Script sieht dann etwa so aus:

```
LAENGE=100
LOOPMODE=FB
WBPATTERN=HD0:Bilder/WB-Hintergrund.iff
ABLAUFPAUSE=1
"FADE (0, 100, 183, #ORI, #ORI, #ORI, 50, 0, 0)"
"FADE (0, 100, 185, #ORI, #ORI, #ORI, 50, 0, 0)"
"FADE (0, 100, 186, #ORI, #ORI, #ORI, 50, 0, 0)"
"FADE (0, 100, 187, #ORI, #ORI, #ORI, 50, 0, 0)"
"FADE (0, 100, 188, #ORI, #ORI, #ORI, 50, 0, 0)"
"FADE (0, 100, 189, #ORI, #ORI, #ORI, 50, 0, 0)"
.
.
.
```

Fangen wir mit dem Kopf an. Dieser muß stets am Anfang des Scripts und in der richtigen Reihenfolge angegeben werden:

1. LAENGE > hier wird die Länge des Programmablaufs in Schritten definiert.
2. LOOPMODE > gibt an, wie der Programmablauf abgespielt wird.
3. WBPATTERN > hier wird der Pfad des Workbench Hintergrundbildes festgelegt. Dieser wird benötigt, wenn man etwas einfacher auf die Farben zugreifen will (Interne Farbennummern).
4. ABLAUFPAUSE > Gibt die Pause an, die zwischen den einzelnen Schritten eingehalten werden soll.
5. Befehle > diese können in einer beliebigen Reihenfolge angegeben werden. Man sollte dabei aber beachten, das der Programablauf nach dieser Reihenfolge erstellt wird. Die Befehle müssen dabei in Anführungszeichen ("..") stehen.

Gerade beim Erstellen eines neuen Scripts kann es vorkommen, das der Syntax eines Befehles nicht eingehalten wird, oder falsche Werte übergeben werden. In

der jetzigen Version verarbeitet MagicColors solche Falscheingaben noch nicht 100%ig. Im schlimmsten Fall jedoch, bricht MagicColors lediglich mit seiner Arbeit ab. Einen Systemabsturz hatte ich deswegen noch nicht und sollte (!!!) auch nicht passieren.

1.4 befehle

Befehle

=====

LAENGE

~~~~~

Syntax: LAENGE=n

Wirkungsweise: Defeniert die Länge des Programmablaufs in Schritten. Bei jedem Schritt wird dann eine oder mehrere Farben verändert.

Details: n > Anzahl der Schritte [2 bis 1000]

Beispiel: LAENGE=100 > Programmablauf mit 100 Schritten defenieren.

-----

LOOPMODE

~~~~~

Syntax: LOOPMODE=mode

Wirkungsweise: Gibt an, wie der Programmablauf abgespielt werden soll.

Details: mode > 'F' nur vorwärts
 'B' nur rückwärts
 'FB' vorwärts und rückwärts

Beispiel: LOOPMODE=FB > spielt den Programmablauf erst vorwärts, dann rückwärts.

WBPATTERN

~~~~~

Syntax: WBPATTERN=Bild

Wirkungsweise: Normalerweise spricht man die einzelnen Farben direkt an. Durch die Art und Weise, wie die Workbench die Farben verwaltet, ist es aber möglich, das die Farbe, die man ändern will, beim nächsten Neustart ein anderes Farbregister zugeordnet bekommt. Hiermit ist es nun möglich die Farben des Hintergrundbildes immer mit der Nummer anzusprechen, wie sie im eigentlichen Bild abgespeichert wird. Dazu wird die Originalfarbe in der Farbpalette der Workbench gesucht.

Zur Zeit werden nur IFF Bilder unterstützt.

Detail: Bild > Pfad und Name des Bildes.

Beispiel: WBPATTERN= > um kein Bild anzugeben.

WBPATTERN=RAM:pic.iff > sucht die Farben aus dem angegebenen Bild in der Farbpalette der Workbench.

-----  
ABLAUFPAUSE

~~~~~

Syntax: ABLAUFPAUSE=n

Wirkungsweise: Definiert die Pause zwischen den einzelnen Schritten, die eingehalten werden soll.

Details: n > Zeitangabe in 1/50 Sekunden [1 bis 1000]

Beispiel: ABLAUFPAUSE=2 > wartet nach jedem Schritt 2/50 Sekunden, bevor es mit dem nächsten Schritt weitergeht.

FADE

~~~~

Syntax: FADE(position,länge, farb-nr, vonR, vonG, vonB, nachR, nachG, nachB)

Wirkungsweise: Erstellt ab der angegebenen Position einen Farbverlauf von der Quellfarbe zur Zielfarbe. Der Befehl muß in Anführungszeichen stehen.

Details: position > Anfang des Farbverlaufes [1 bis 998]

länge > die Länge des Farbverlaufes [3 bis 1000]

farb-nr > gibt an, welche Farbe verändert werden soll.

'3' entspricht der Farbregisternummer [0 bis 255]

'#3' entspricht der Farbnummer des Originalbildes (WBPATTERN muß dazu ein Bild übergeben worden sein). [#0 bis #255]

vonR gibt den jeweiligen Farbton der Quellfarbe an.

vonG > '27' entspricht dem 28/256 Teil eines Farbtone [0 bis 255]

vonB '#ORI' entspricht der Originalfarbe [#ORI]

nachR gibt den jeweiligen Farbton der Quellfarbe an.

nachG > '27' entspricht dem 28/256 Teil eines Farbtone [0 bis 255]

nachB '#ORI' entspricht der Originalfarbe [#ORI]

'+10%' 10% hellere [+0% bis +100%]

'-10%' 10% dunklere [+0% bis +100%]

Beispiel: "FADE(1,50,#3,#ORI,#ORI,#ORI,0,0,0)" > erzeugt einen Farbverlauf mit 50 Abstufungen ab Position 1.

-----  
SET

~~~

Syntax: SET(position, farb-nr, rot, grün, blau)

Wirkungsweise: Setzt eine Farbe auf den angegebenen Farbwert. Der Befehl muß in Anführungszeichen stehen.

Details: position > Anfang des Farbverlaufes [1 bis 999]
 farb-nr > gibt an, welche Farbe verändert werden soll.
 '3' entspricht der Farbregisternummer [0 bis 255]
 '#3' entspricht der Farbnummer des Originalbildes (WBPATTERN
 muß dazu ein Bild übergeben worden sein) [#0 bis #255]
 rot gibt den jeweiligen Farbton an
 grün > '27' entspricht dem 28/256 Teil eines Farbtone [0 bis 255]
 blau '#ORI' entspricht der Originalfarbe [#ORI]

Beispiel: "SET(1,20,255,0,0)" > setzt das Farbregerister Nr.20 auf die Farbe Rot.

 PAUSE

~~~~~

Syntax: PAUSE(position,dauer)

Wirkungsweise: Wartet die angegebene Zeit auf einer bestimmten Position. Der Befehl muß in Anführungszeichen stehen.

Details: position > wo die Pause eingehalten werden soll [1 bis 1000]  
 dauer > Zeitangabe in 1/50 Sekunden [1 bis 32767]

Beispiel: "PAUSE(23,200)" > auf der Position 23 wird 4 Sekunden gewartet, bevor der nächste Schritt abgearbeitet wird.

-----  
 CYCLE

~~~~~

Syntax: CYCLE(position, farbnr, smooth, 1. farbnr, 2. farbnr, ..., 256. farbnr)

Wirkungsweise: Wechselt die Farbwerte der angegebenen Farben ähnlich wie es DPaint macht.

Details: position > Anfang des Farbenwechsels
 farb-nr > gibt an, welche Farbe verändert werden soll.
 '3' entspricht der Farbregisternummer [0 bis 255]
 '#3' entspricht der Farbnummer des Originalbildes (WBPATTERN
 muß dazu ein Bild übergeben worden sein) [#0 bis #255]
 smooth > (wird noch nicht unterstützt)
 n.farbnr > hier werden die Farben angegeben, von denen die Farbwerte
 übernommen werden sollen. Es können max. 256 Farben
 angegeben werden.
 '3' entspricht der Farbregisternummer [0 bis 255]
 '#3' entspricht der Farbnummer des Originalbildes [#0 bis ↵
 #255]

1.5 starten

Starten
=====

Um MagicColors zu starten, genügt die Eingabe von 'MagicColors <Script File>' in einem Shell/CLI Fenster. Es erscheint eine kurze Meldung, das MagicColors installiert ist. Sollte ein Fehler auftauchen (zu wenig Speicher, Fehler im Script, etc.), so erfolgt eine entsprechende Fehlermeldung.

Um MagicColors abzubrechen, genügt es den Mauszeiger in die obere linke Ecke der Workbench zu bewecken. Es erfolgt ein kurzes Aufblitzen der WB, der mitteilt das MagicColors erfolgreich aus dem System entfernt worden ist.

1.6 speicherverbrauch

Speicherverbrauch
=====

Der Speicherverbrauch für den Programmablauf kann unter gewissen Umständen recht hoch werden. Wieviel nun MagicColors an Speicher benötigt, hängt von der Anzahl der Farben auf der Workbench ab und der Länge des Programmablaufes. Für einen Programmschritt werden bei einer 256 Farben Workbench 4kB benötigt.

1.7 anforderung

System Anforderung
=====

Mindestkonfiguration:

- ein Amiga
- Kickstart 3.0 oder höher
- AGA oder Grafikkarte

Getestet wurde MagicColors unter folgenden Konfigurationen:

- A4000/060, AGA/CyberVision, Kickstart 3.0, WB 3.1
- A4000/040, AGA/CyberVision/Retina BLT, Kickstart 3.0, WB3.1
- A1200/030, AGA, Kickstart 3.0, WB3.1

Die Anzahl der Farben auf der Workbench und die Anzahl der Farben die verändert werden sollen, fällt bei einer Grafikkarte nicht sonderlich ins Gewicht. Bei einem AGA Bildschirm sollte man mit den Farben sparsam umgehen (muß in späteren Versionen von MagicColors nicht mehr beachtet werden). Der Speicherverbrauch hängt vom Umfang der Farbspielerein ab.

1.8 installation

Installation
=====

Hierbei muß man nicht viel beachten. MagicColors kopiert man am besten in das C: Verzeichnis. Die Scripts und Hintergrundbilder kann man irgendwo auf der Platte unterbringen, wo man sie später auch wieder findet. Der Pfad für die Hintergrundbilder muß dabei eventuell in den Scripts geändert werden.

Um sich nun die beiliegenden Beispiele ansehen zu können, muß man erst mit WBPatten die Beispielsbilder als Workbench Hintergrund einstellen (z.B. für Beispiel_1.script > Beispiel_1.IFF6 als Hintergrundbild). Wichtig dabei ist, daß die Workbench genügend freie Farbbregister besitzt, da es sonst zu Farbverfälschungen kommt und man dann eventuell nichts oder nur Chaos sieht. Die Beispielsbilder brauchen alle eine Workbench mit mindestens 32 bzw. 64 Farben.

Jetzt genügt ein Doppelklick auf das jeweilige Icon des Beispiel-Scripts.

1.9 hintergrundbilder

Hintergrundbilder
=====

Normalerweise will man nur die Hintergrundbilder (Patterns) der Workbench und deren Fenster animieren. Nimmt man dazu ganz normale Bilder (z.B. mit MagicWB Farbpalette), sehen die Farbspielerein eher verwirrend aus und ausser einem einfachen heller/dunkler werden der Farben ist nicht viel drin.

Um nun ansprechende Hintergrundbilder zu erstellen, muß man schon ein wenig Zeit aufbringen. Es ist nicht wichtig wie das eigentliche Bild aussieht, sondern nur wie die Farben angeordnet sind. Hat man erst einmal begriffen, wie man das anstellt, kann man sogar richtige Animationen erstellen.

Die Workbench (bzw. IPrefs/WBPatten) verteilt die Farben so, daß keine Farbbregister unnötig belegt werden. Sollte ein Hintergrundbild also aus 32 Farben bestehen und alle hätten die gleiche oder beinahe die selbe Farbe, dann würde die Workbench wahrscheinlich nur ein Farbbregister dafür verwenden. Ebenso würde versucht werden die Farben des Bildes in schon bestehende Workbench Farben, die eigentlich von anderen Programmen genutzt werden, miteinzubinden.

Da ich leider nicht genau weiß, wie die Workbench die Farben verteilt, kann ich hierzu keine große Hilfe geben. Folgende Dinge sollte man beachten:

- die Workbench muß noch genügend freie Farbbregister besitzen
- keine gleichen Farben verwenden > benötigt man mehrere Farbbregister mit der selben Farbe, sollte man das Bild mit unterschiedlichen Farben abspeichern und die richtige Farbpalette erst im Script definieren.
- darauf achten, daß die Farben im Bild nicht auf der Workbench schon vorhanden sind.

Wem das alles zu kompliziert ist, der sollte einfach mal probieren.

1.10 fehlerberichte

Fehlerberichte
=====

Sollte tatsächlich einmal ein Fehler auftauchen (was ich bezweifle), würde ich mich über eine kurze E-Mail, mit einer Beschreibung des aufgetauchten Fehlers freuen.

1.11 autor

Autor von MagicColors
=====

Denis Zwornarz
Rudolf-Werner-Str.15
97490 Poppenhausen
Germany

E-Mail: dzwornarz@sw.cybersurf.de

Jede E-Mail die ich erhalte wird in jedem Fall auch beantwortet (sollte ich diese erst übersetzen müssen, dann dauert es etwas länger).

1.12 history

Was bisher Geschah
=====

27.8.1996 > erste Veröffentlichung von MagicColors v1.0

1.13 zukunft

Was noch kommt
=====

- eventuell OCS und ECS Unterstützung
- die CPU noch weniger zu belasten
- Speicherverwaltung optimieren
- Preference mit GUI

1.14 registrieren

Registrierung
=====

MagicColors ist Shareware. Jeder, der dieses Programm öfters benutzt, sollte sich bei mir registrieren lassen. Die Shareware Gebühr beträgt 2,- DM und ist auf folgendes Konto zu überweisen:

Kreissparkasse Schweinfurt
BLZ 793 501 01
Konto-Nr. 496398
Verwendungszweck: <eigene E-Mail Adresse>

Das Keyfile wird verschickt (nur als E-Mail), sobald ich die Buchungsbestätigung habe. Jeder registrierte Benutzer erhält automatisch die neueste Version von MagicColors, sowie neue Scripts und Hintergrundbilder. Der Versand erfolgt ausschliesslich als E-Mail.

Das verbreiten, modifizieren und entschlüsseln des Keyfile's ist verboten.

1.15 rechtliches

Rechtliches
=====

MagicColors v1.0 (27.8.1996)

Copyright ©1996, Denis Zwornarz
All rights reserved

Dieses Programm ist Shareware. MagicColors darf nur als komplettes Archiv weiterkopiert werden. Dieses Archiv sowie MagicColors im Einzelnen, darf ausschliesslich nur im Aminet bzw. auf der Aminet CD und in nicht kommerziellen Mailboxsystemen erscheinen. Jegliche kommerzielle Nutzung (ausser Aminet-CD) ist ohne schriftliche Genehmigung vom Autor verboten.

Weder MagicColors noch diese Anleitung darf auf irgendeine Weise modifiziert werden (ausser das Übersetzen in eine andere Landessprache).

Für eventuelle Schäden, die direkt oder indirekt durch MagicColors entstehen, übernehme ich keine Haftung.

EPILEPSIE WARNUNG ~~~~~

Bei manchen Personen kann es zu epileptischen Anfällen oder Bewußtseinstörungen kommen, wenn sie bestimmten Blitzlichtern oder Lichteffekten im täglichen Leben ausgesetzt sind. Diese Personen können einen Anfall erleiden, während sie bestimmte Fernsehbildern ausgesetzt sind. Es können auch Personen davon betroffen sein, deren Krankengeschichte bislang keine Epilepsie aufweist und die nie zuvor epileptische Anfälle gehabt haben.

Falls bei jemanden unter Einwirkungen von Blitzlichtern mit Epilepsie zusammenhängende Symptome (Anfälle oder Bewußtseinsstörungen) aufgetreten sind, sollte man sich an seinen Arzt wenden, bevor man MagicColors benutzt.