



Sharp X68000: 16-Bit-Wunder mit Commodore-Schicksal

Der japanische Amiga

Seite 12



steirischer herbst 2004: Kunst am C64 und Atari VCS

Barbarianware in Graz

Seite 8



Das Spiel, auf das alle gewartet haben

Gianas glorreiche Rückkehr

Seite 27



Programmierkurs von Paul Allen Panks

Adventures in BASIC

Seite 14



Interview mit Andre Hammer:

Tetris- und Verkaufsprofi



Liebe Loteks!

Zuerst möchte ich mich für den Fehler entschuldigen, der beim Druck der letzten Ausgabe für einige Lese-probleme sorgte. Bei einem Teil der Auflage ist ein bereits einseitig bedruckter Bogen verkehrt in die Maschine geraten, wodurch zwei Seiten auf dem Kopf standen. Ich hoffe, dass es nicht allzu schwierig war, die Artikel trotzdem zu lesen.

Nun zu etwas Erfreulicherem: Nach dem Media 2000 Computer Club in Berlin bietet auch Tobias Topyla, Herausgeber der (übrigens sehr guten) Online-Computerzeitschrift TtecX.com, Lotek64 auf seiner Seite zum Download an:

<http://www.ttecX.de/download/overview.php>

Wir bedanken uns ganz herzlich für diese Unterstützung! Mittlerweile kann Lotek64 also von vier Seiten bezogen werden: www.ttecX.de, www.media2000.info, www.c64-mags.de und natürlich – gALAKTUS sei Dank – von www.lotek64.com.

Inhaltlich ist diese Ausgabe, die zum zweiten Mal in der Geschichte von Lotek64 28 Seiten umfasst, zu einem großen Teil dem Commodore 64

gewidmet. Paul Allen Panks hat uns einen BASIC-Programmierskurs zur Verfügung gestellt, mit dem er unter Beweis stellt, dass auch mit dieser aus der Mode gekommenen Programmiersprache gute Spiele geschrieben werden können. Antitrack gibt Bastelnachhilfe und Stephan Lesch erklärt, wie fehlerhafte Demos zum Laufen gebracht werden können.

Dem „japanischen Amiga“ X68000 widmet Japan-Experte Doraemon einen Beitrag, dafür fällt diesmal die CD-Konsolen-Serie aus. Der Artikel über die Playstation wird in der nächsten Ausgabe nachgeholt.

Ich wünsche allen Leserinnen und Lesern viel Vergnügen bei der Lektüre und hoffe auf Feedback!

Lord Lotek

lotek64@aon.at

* * *

...und *last but not least*: Danke an alle neuen und alten Abonnent/inn/en, sowie an alle Mitarbeiter dieser Ausgabe: Daniel M. (Doraemon), Mathes Alberto, Andre Hammer, Stephan Lesch, Antitrack/Legend, Hiram Kümper, Simon Quernhorst. *Lektorat*: Lisbeth Zeiler.

Impressum: Herausgeber, Medieninhaber: Georg Fuchs, Waltendorfer Hauptstr. 98, A-8042 Graz/Austria

Das Lotek64-Abo

Lotek64 ist grundsätzlich kostenlos. Da die Portokosten aber sehr teuer sind, muss jeder Leser / jede Leserin selber dafür aufkommen.

Überweisung von 4 Euro, überschüssige Cents werden natürlich gutgeschrieben.

Die Mitte 2002 eingeführten neuen Posttarife haben auch Konsequenzen für die „Schnorrer-Abos“. Es werden nur noch so viele Hefte gratis verschickt, bis ein Kilogramm voll ist. Das können zwischen null und vierzehn Hefte sein.

1. Portokosten

[a.] **Abos in Österreich:** Die Portokosten für eine Ausgabe betragen 1 Euro (Versand als Brief, 50 bis 100 g). Seltsamerweise kostet es geringfügig weniger, ein Heft ins Ausland zu versenden. Ein Jahresabo kostet also 4 Euro.

[b.] **Abos außerhalb Österreichs:** Der genaue Preis ändert sich jedesmal geringfügig, da er im Tarifmodell der österreichischen Post aus einer Kombination aus Stückzahl und Gewicht berechnet wird (angefangene ganze 1000 Gramm sind immer komplett zu bezahlen). Realistisch sind erfahrungsgemäß zwischen 90 Cent und 1 Euro pro Heft. Für ein Jahresabo empfehle ich also die

2. Überweisungen aus dem EU-Ausland

Da es innerhalb der EU (egal, ob Eurozone oder nicht) keine Überweisungsgebühren mehr gibt (Stichwort „Binnenüberweisung“), ist es nicht mehr nötig, das Geld in einem Kuvert zu versenden oder auf ein deutsches Konto zu überweisen. Den Versand von Bargeld in einem Brief schlage ich nur Abonnent/inn/en aus Nicht-EU-Ländern vor.

Die Bankverbindung für Lotek64:

LOTEK64-Bankverbindung

Internationale Bankverbindung (IBAN): AT58 1200 0766 2110 8400
BIC (SWIFT): BKAUATWW, Kontoinhaber: Georg Fuchs

Innerhalb Österreichs: Konto 76621108400, BLZ 12000

Als Verwendungszweck bitte „Lotek64-Abo Vorname Nachname“ oder Ähnliches angeben, max. 35 Zeichen!

Wer ein Abo bestellt und eine Portospende überweist, muss uns natürlich trotzdem per E-Mail oder auf dem Postweg verständigen und die Adresse bekannt geben!

Hier die Regeln für das Lotek64-Abo in Kürze:

A. **PORTOSPENDE** – Wer sich an den Portokosten beteiligt – 1 Euro pro Ausgabe –, bekommt Lotek64 garantiert zugeschickt.

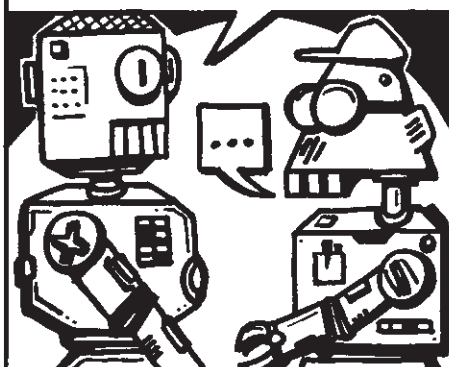
B. **KEINE SPENDE** („Schnorrer-Abo“) – Wer nichts bezahlt, bekommt Lotek64 nur, „so lange das Geld reicht“.



**GESTERN IM
JAHRE 2103**

DM

**Ist es sehr schlimm,
Herr Doktor?**



**Nun ja! Eigentlich
haben Roboter
gar keine Gefühle**



Vielseitiger NES-Fan mit radikaler Preispolitik

Interview mit Andre Hammer

Andre Hammer, Jahrgang 1979, ist Konsolenfreak und hat die Spieleabteilung der ersten Filiale des jungen österreichisch-slowenischen Elektrofachhändlers bof aufgebaut. Wir haben mit ihm über Hobbys und den Arbeitsalltag gesprochen.

Lotek64: *Durch deinen Beruf hast Du auf dem Spielmarkt der Region Graz ein Wörtchen mitzureden. Wer steckt hinter dem Spiele- und Konsolenfreak?*

Andre: Von zwei absolut gegensätzlichen Charakteren erzogen – mein Vater rational denkend, kritisch und hart; meine Mutter liebevoll, kreativ und impulsiv – wurde auch meine Denkweise entsprechend ungewöhnlich geprägt.

Die Mischung aus analytischem Denken und träumerischer Kreativität hat es mir selten leicht gemacht. Ich war kein übertrieben intelligentes, dafür aber ein vernünftiges Kind mit starkem Gerechtigkeitssinn, dem leider häufig der Mut gefehlt hat, sich gegen Ungerechtigkeiten aller Art zur Wehr zu setzen. Oft fiel es mir schwer, eine Entscheidung zu treffen, weil ich meinen Mitmenschen durch unüberlegte Handlungen kein Unrecht zufügen wollte. Am ehesten hätte man mich damals mit dem kleinen Bastian aus *Die Unendliche Geschichte* vergleichen können, in späteren Jahren mit Peter Parker, ohne die Fähigkeiten von Spiderman zu besitzen. Im Nachhinein kann ich sagen, dass ich trotz meiner Bemühungen, gut zu sein, einige Fehler gemacht habe. Heute bin ich langsam aber sicher im Begriff, mich zu einem netten und fairen, aber auch sturen und manchmal arroganten Rebellen zu entwickeln, der zwischen berechtigter und falscher Rücksichtnahme unterscheiden kann. Mittlerweile wage ich es auch, mich gegen Unfairness jeglicher Art aufzulehnen. Trotzdem bevorzuge ich den friedlichen Weg. Meine zukünftigen Ziele sind die Gründung einer Familie

und die Ausübung meiner kreativen Fertigkeiten auf beruflicher Basis. Ein wichtiger Punkt in meinem Leben sind Spiele, mit denen ich mich schon seit langem intensiv beschäftigt habe! Vordergründig interessieren mich der kreative Anteil und das Konzept eines Spieles.

Lotek64: *Welchen Bezug hast du zu „Retrocomputing“? Warst du auch Besitzer eines Commodore 64 oder anderer Systeme, die es nicht mehr gibt?*

Andre: Leider war ich nie im Besitz eines C64. Ich bin direkt mit einem PC (486er) eingestiegen und habe als erste Videospielkonsole ein NES besessen. Bei meinem Onkel bin ich früher häufig vor dem C64 gesessen und durfte auf diese Weise eine Spielesammlung von erstaunlicher Größe testen. Über ältere Heimcomputer und Konsolen habe ich mich erst nachträglich informiert. Bisher war ich im Besitz folgender Geräte: NES, Master System, Sega Mega Drive, SNES, Sony Playstation, N64, Sega Dreamcast, PS2, Game Boy/Color/SP, Xbox, Gamecube und mehrere PCs. Andere Geräte wie Amiga 500, Atari Lynx oder Sega Game Gear habe ich mir des Öfteren ausgeliehen. Die übrigen Systeme hätte ich selbstverständlich gerne besessen, aber mein Taschengeld war begrenzt und ich konnte meine Eltern nicht überzeugen, mir noch mehr Spielgeräte zu kaufen. Im Augenblick bin ich dabei, meine Sammlung im Retrobereich zu erweitern. Ich favorisiere in erster Linie einen Atari Jaguar, ein SNK Neo Geo mit dem Modul Samurai Showdown und ein Philipps CD-i, wobei ich Letzteres nur dann kau-



Andre Hammer hat mit einem NES angefangen.

fen würde, wenn ich einen der schwer erhältlichen Zelda-Teile bekommen könnte. Ich weiß, wie schlecht diese Spiele objektiv betrachtet sind, aber ich MUSS sie trotzdem haben.

Lotek64: *Du hast im bof die Spieleabteilung aufgebaut. Was kannst du allgemein über den heutigen Spielmarkt sagen? Wie kann man sich deinen Arbeitsalltag vorstellen und wie sind deine bisherigen Erfahrungen?*

Andre: Für alle Leser, die Building of Fun, kurz BOF, nicht kennen, möchte ich anmerken, dass es sich um ein junges Elektronikunternehmen in Österreich handelt, welches seine Ursprünge in Slowenien hat. Laut Angaben unserer Geschäftsführung ist es dort die unangefochtene Nr. 1.

Im November 2002, einen Monat nach der Eröffnung habe ich die Leitung der Spieleabteilung in der ersten Filiale (5.000 m²) in Graz übernommen. Ich habe mich sehr engagiert und mir das Ziel gesetzt, eine der größten und schönsten Abteilungen in Graz aufzubauen, was im ersten Jahr durchaus machbar erschien. Kleine Umsatzsteigerungen, die nach und nach wuchsen, gaben mir die Kraft und Motivation, noch mehr Arbeit in mein Vor-

haben zu stecken. An manchen Tagen war ich auch außerhalb der eigentlichen Dienstzeit aktiv. Die ersten Schritte forderten viel Einsatz, waren aber im Grunde genommen einfach umzusetzen. Mit gelungenen Events wie dem „5000 Euro-Gewinnspiel“ (erster Preis war das „BOF-Extreme-Package“, drei Konsolen inklusive Zubehör und extrafettem Spielepaket) haben wir der Konkurrenz mit Sicherheit so manchen Kunden weggenommen. Durch den Aufbau eines vielseitigen Sortiments und der radikalen Preispolitik, die durch hervorragende Konditionsblätter bedingt kein größeres Problem darstellt, ist uns schon in der Anfangszeit ein gutes Image im Spielbereich vorausgegangen. Beispielsweise haben wir im Dezember 2003 an einem starken Weihnachtssamstag den Original Sony Controller für Playstation 2 um 4,90 Euro (Listenpreis: 31,90) beworben. Die Folge war ein riesiger Tumult – Kunden stürmten in rauer Zahl das Geschäft, das Telefon klingelte ununterbrochen und der Ärger der Konkurrenz war eher groß. Einige Tage nach dem Vorfall musste ich die Situation mit dem erzürnten Sony-Chef klären, he he. Um ehrlich zu sein, bin ich kein Freund von derartigen Aktionen, weil diese weit-

läufig betrachtet viel Negatives für die Industrie bringen, allerdings sah ich mich durch damalige Umstände zu einer solchen Handlung gezwungen. Was heute noch niemand so recht glauben will: Durch einen geschickten rechnerischen Trick haben wir pro Controller sogar noch etwas verdient! Selbstverständlich war die Menge nicht übertrieben groß, aber welche Firma kann schon von sich behaupten, eine Aktion dieser Art durchgeführt zu haben? Ein weiteres Highlight war das GBA-Spiel um 1 Euro (begrenzte Stückzahl)! Es handelte sich noch dazu um *Castlevania: Harmony of Dissonance*, ein Spiel, das von Magazinen ziemlich gut bewertet wurde. Zu guter Letzt habe ich noch meinen Chef überreden können, einen zweiten Spielemitarbeiter (Christopher Tafner) mit Vollzeitbeschäftigung ins Unternehmen zu holen.

Lotek64: Klingt fast ein bisschen zu perfekt. Gab es nie Enttäuschungen oder Rückschläge?

Andre: Nicht alles ist so reibungslos verlaufen. Einige Dinge, die ich für gelungen hielt, scheiterten kläglich. Insgesamt ist es mir trotz meiner Bemühungen nicht gelungen,

mein ursprüngliches Ziel zu erreichen, was hauptsächlich daran liegt, dass ich dieses mittlerweile anders definiert habe. Der ständige Wettkampf mit der Konkurrenz und das Feilschen mit Außendienstmitarbeitern, deren Wort nicht immer Gültigkeit hat, entzieht sich dem, was mir Spaß bereitet. Ich will nur so viel sagen: Es sind die größten Firmen, die für wenig Unterstützung sorgen und dabei ein arrogantes Verhalten an den Tag legen. Glücklicherweise gibt es Ausnahmen: Viel gelernt habe ich von Martin Köstner (Play Art, Beispiel: Resident Evil), der mich in meiner Zeit als Einkäufer stets mit guten Ratschlägen unterstützt hat. Vereinbarungen mit Robert Stauchner (Koch Media, Beispiel: Spellforce) hatten stets Handschlagqualität. Auch wenige andere waren durchaus in Ordnung.

Nach fast zwei Jahren war es schließlich meine freie Entscheidung den Einkauf abzugeben. Interne Widerstände und Missverständnisse waren mitunter ein Grund dafür, allerdings muss ich erwähnen, dass ich ebenfalls einen solchen Widerstand gegen die Chefetage gebildet habe (sonst hätte es Aktionen wie den 4,90 Euro-Cont-

roller nie gegeben!). Außerdem handelt es sich um übliche Schwächen, die man bei der Konkurrenz in einer fast noch stärkeren Form vorfindet. Das kann ich beurteilen, weil ich dort meine Ausbildung gemacht habe. Alles ist auf übertriebene Härte aufgebaut, und es gibt ein hierarchisches System, welches genau befolgt werden muss. Bei BOF ist alles ein wenig freier und besser aufgebaut. An dieser Stelle ein Lob an die tolle Chefin Sandra Drumm, die im Bereich Tonträger demonstriert, wie man ein organisatorisch wertvolles Arbeitsmodell einführen und trotzdem jeden Mitarbeiter mit einbeziehen kann! Doch zurück zum eigentlichen Thema: Heute macht Christopher Tafner den Einkauf und geht dabei weniger extrem vor als ich es tat. Er achtet darauf, ein breit gefächertes Sortiment anzubieten und vernachlässigt Produkte für Freaks ebenso wenig (Beispiel: Katana Schwert für PS2, Steel Battalion Controller für Xbox). Ich selbst bin zur Hälfte in den Tonträgerbereich übergewechselt, betreue aber Spielekunden nach wie vor. Vorübergehend bleibe ich, weil die Zusammenarbeit mit meinen Kollegen einfach phantastisch ist und wir

ein ziemlich gutes Team abgeben! Im Allgemeinen kann ich nur sagen, dass die spärliche wirtschaftliche Situation auch im Gamesektor ihre Spuren hinterlässt. Mir ist aufgefallen, dass die Leute viel lieber zu Billigware greifen, vorrangig im DVD- und CD-Sektor. Spiele haben ein wenig an Bedeutung verloren, was meiner Meinung nach an den höheren Preisen und noch immer an den Raubkopien liegt.

Lotek64: Abgesehen vom allgemein spürbaren Sinken der Kaufkraft: Hast du das Gefühl, dass Titel mit Retro-Bezug ziehen? Viele Firmen bieten ja ihre alten Titel wieder an, z.B. Nintendo. Infogrames hat sich in Atari umbenannt – kann man daraus schließen, dass die alten Markennamen ziehen?

Andre: Titel mit Retro-Bezug verkaufen sich speziell bei Nintendo sehr gut. Auch hier liegen, was die Verkaufszahlen anbelangt, Spiele mit bekannten Videospielhelden wie Super Mario oder Link weit vorne. Titel wie Ice Climber oder Excite Bike verkaufen sich eher mäßig. Auch die Compilation von Namco hat sich im BOF recht brav verkauft.

Ob alte Firmenmarken ziehen, ist schwer zu sagen. Viele der Spieler einer jüngeren Generation kennen Atari, Commodore & Co nicht. Ich habe beobachtet, dass der Atari-Stick speziell von älteren Semestern gekauft wurde. Jugendliche und Kinder haben hier nicht zugegriffen.

Lotek64: A propos „Atari-Stick“: Auch Commodore, im Besitz von Tulip Computers (Niederlande), wird bald Joysticks mit 30 eingebauten C64-Spielen anbieten, wie es außer Atari auch Intellivision und Activision gemacht haben. Werdet ihr das Gerät anbieten? Glaubst du, dass es sich gut verkaufen wird?

Andre: Wenn ich daran denke, dass die Nachfrage nach dem Atari-Stick beachtlich hoch war – die Erstauslieferung war allein im BOF bereits einige Tage nach der Veröffentlichung vergriffen – kann ich mir nicht vorstellen, dass es dem Commodore-Joystick anders erge-



Einblick in Andres Spiele(be)reich in der Grazer bof-Filiale.

hen soll. Ich wäre stark dafür, pünktlich zum Release eine etwas größere Menge der C64-Version anzubieten. Aber das ist auch vom tatsächlichen Preis abhängig.

Lotek64: Du hast auf der Seite eines Grazer Radiosenders auch eine Sektion für Spieletests kreiert. Bitte erzähl etwas darüber.

Andre: Wo Konkurrenzseiten sich kommentarlos kaufen lassen und die (künstlich gehypten) Verpackungstexte abdrucken, bemühen wir uns, authentische und vor allem ehrliche Testberichte abzuliefern. Und das, obwohl in unserem Fall ebenfalls Geld für diese Dienstleistung im Spiel ist. Man sieht also, dass durch Kommunikation vieles möglich ist. Wir freuen uns jedenfalls, wenn der eine oder andere Lotek64-Leser auf unsere Seite schaut und Anregungen oder Kritik äußert. Worauf die Sache hinausläuft, ist nämlich von den Wünschen unserer Leser abhängig.

Lotek64: Wieder zurück zu dir: Was sind deine persönlichen Lieblingsspiele?

Andre: Ich werde einfach spontan ein paar meiner Favoriten nennen: Resident Evil (PSX, GC), Golden Eye (N64), Virtua Tennis (Dreamcast), Maniac Mansion (C64, Deluxe am PC), Die Gilde (PC), House Of The Dead (Automat), Civilization (PC), Siedler von Catan (Brettspiel), Zelda 3 (SNES), Yoshis Island (SNES), Knights of the old Republic (XBox) u.v.m.

Aktuell spiele ich neben den Pflichtspielen für den Soundportaltest Doom 3 (PC) und bastle gelegentlich mit dem Editor von Die Sims 2 herum.

Lotek64: Was machst du, wenn du dich nicht mit Computerspielen beschäftigst? Und was sind deine viel zitierten drei Spiele für die einsame Insel?

Andre: Ich betreibe gerne Sport und bin ein ziemlich naturverbundener Mensch. Gleichmaßen liebe ich es aber auch, gut zu essen

und liege gerne faul vor dem Fernseher. Ich lese eine Menge, sofern ich Zeit dafür finde, weil ich der Meinung bin, dass es viel über das Leben herauszufinden gibt. Allerdings konzentriere ich mich eher intensiver auf wenige Themenbereiche (Philosophie, übersinnliche Themen, Comics, Romane – häufig Fantasy oder Historisches) und habe daher auch keine umfangreiche Allgemeinbildung. Mein Lieblingslokal in der Grazer Innenstadt ist der Nachtexpress (Guns N' Roses, Red Hot Chili Peppers usw.); neuerdings halte ich mich auch gerne im WON (= World of Nightlife, bunt gemischte Lokaltitäten) im Center West auf. Meine drei Spiele für die einsame Insel (ich hasse diese Frage, wenn ich sie selbst gestellt bekomme!) sind:

- 1.) Tetris (GBA) – weil ich mit Sicherheit einen noch nie da gewesenen Rekord aufstellen kann. Sollte ich jemals von der Insel zurückkehren, würde man mich als Genie ansehen!
- 2.) Morrowind + Add Ons (PC) – weil es ein unheimlich komplexes Rollenspiel ist, welches man wohl nur auf einer einsamen Insel durchspielen kann.
- 3.) Singles (PC) – weil ich davon ausgehe, dass es auf der Insel keine Frauen gibt. Außerdem verspreche ich mir durch dieses Spiel, halbwegs für soziale Kontakte fähig zu bleiben!

Lotek64: Und sonst?

Andre: Es ist für mich eine große Ehre, in Lotek64 interviewt zu werden, zumal ich die inhaltliche Qualität dieser Zeitschrift sehr schätze. Ebenfalls möchte ich an dieser Stelle jenen Leuten danken, die aufopferungsvoll an einem Projekt arbeiten, ohne dabei den finanziellen Aspekt als wichtigsten Motivationsfaktor in den Vordergrund zu stellen.

Lotek64: Danke für das Interview und die freundlichen Worte über das Magazin!

Kein Markt für Retrospele?

Lotek64: Wie schätzt du die aktuelle Situation im Spielesektor ein (allgemein und verkaufsspezifisch)?

Joe Nettelbeck: Spiele kosten inzwischen fast so viel wie Hollywoodfilme, die Leute kaufen fast nur noch Titel aus den Top Ten, der Markt für Onlinespiele wächst viel langsamer, als alle immer glauben, und überhaupt war früher alles besser: Als überwiegend harm- und argloser Mitarbeiter eines kleineren deutschen Publishers konnte man manchmal meinen, die Lage sei hoffnungslos (wenn auch nicht ernst).

Doch gemacht, gemacht! Denn es gibt nach wie vor Chancen und Möglichkeiten, auch wenn man nicht gleich für eine Preproduction siebenstelligen Summen zahlen kann und will. Zum einen ist es ja glücklicherweise so, dass das ständige Schreien der Magazine nach Innovation im Spielervolk nur begrenzt seinen Widerhall findet. In Wahrheit wollen nämlich die meisten (mich selbst eingeschlossen) am liebsten das spielen, was sie eh schon kennen. Von mir aus könnte es noch 20 Fortsetzungen von Baldur's Gate geben, und zwar ohne große Änderungen, wenn es geht. Will sagen: Ein solides, gut gemachtes Spiel hat auf lange Sicht und mit Hilfe der Mundpropaganda allemal seine Chancen, wenn es den Geschmack der Massen trifft. Und dabei muss es nicht unbezahlbar teuer in der Entwicklung sein.

Zum anderen ist es keineswegs so, dass die denkbaren Zielgruppen ausgeschöpft wären. Klar, die Hardcore-Zocker sind schon längst übersättigt und daher ein eher undankbares Publikum. Aber hat sich beispielsweise schon mal jemand Gedanken darüber gemacht, dass unter den erfolgreichen Spielen nicht wenige zu finden sind, die sich an beide Geschlechter richten? Natürlich haben Frauen eine andere Herangehensweise an ein Spiel als



Joachim „Joe“ Nettelbeck war Chefredakteur von PC Joker, hat bei Power Play gearbeitet und ist derzeit Produktmanager bei Koch Media (Singles).

Männer, aber es ist durchaus möglich, beides unter einen Hut zu bringen. Es erscheint mir sogar sinnvoll, an gute, erwachsene Spiele nur für Frauen zu denken. Also, dann mal los: Die Ersten werden die Ersten sein...

Joe Nettelbeck, Produktmanager bei Koch Media

Lotek64: Wie schätzen Sie die Verkaufschancen von Retro-Produkten (Beispiel: C64-Joystickkonsole C64-DTV) in der heutigen Zeit ein?

Markus König: Für die Generation der 80er Jahre haben die Retro-Spiele auch heute noch ihren Reiz. Wer nun einmal mit Pacman, Galaxian, Donkey Kong und vielen anderen Klassikern aufgewachsen ist, der spielt diese Titel gerne wieder.

Jedoch schätze ich den Absatz dieser Retro-Titel als vorsichtig ein, die heutigen Spieler sind durch die neuen Konsolen optisch und akkustisch sehr verwöhnt. Auch die bekannteren Titel von damals haben höhere Chancen auf großes Interesse als die weniger bekannten. Für meinen Teil tauche ich aber immer gerne wieder in die Retro-Zeit ein und erfreue mich an den Klassikern meiner Jugend.

Markus König, Marketing Manager bei Electronic Arts

Die CFÖ (Computerförderung)

Die Absicht der CFÖ ist es, eine starke Gemeinschaft aufzubauen. Es geht darum, ein gemeinsames Medium für Computerclubs, Bastler, Hersteller, Händler usw. einzurichten. Dabei sollen vor allem die Ausnahmereisnerungen der Computerszene präsentiert und eine Kommunikationsmöglichkeit angeboten werden. Ziel ist es, möglichst viele kleine und extreme Gruppen und Individualisten zu unterstützen und diesen ein gemeinsames Sprachrohr zur Außenwelt zu geben. Hierzu soll ein (weltweites) Anzeigenblatt aufgebaut werden. Möglichst in den Sprachen Deutsch, Englisch, Französisch, Spanisch. Preise für Anzeigen liegen grundsätzlich so niedrig wie irgendwie möglich. Werbung für Clubs und deren Magazine sollen grundsätzlich umsonst sein, ebenso Kontaktanzeigen, Informationen, Fragen und Antworten (die letzten vier Punkte gelten für alle). Möglichst viele Informationen sollen in möglichst wenig Seiten gepresst werden und dabei eine möglichst hohe Leserschaft erreicht werden. Es ist nicht (!) die Absicht, bereits bestehenden Magazinen Konkurrenz zu machen oder gar zu übernehmen, sondern diese zu fördern.

Das Projekt soll auf Deutsch starten und vorerst nur als PDF-Dateien im Internet veröffentlicht werden, bis eine gewisse Leserschaft erreicht wird. Außerdem werden RTF-Dateien den Magazinen zur Verfügung gestellt werden. Die Anzeigen werden nach einem noch aufzubauenden Nummernsystem sortiert, so dass jeder Club und jedes Magazin die für sie interessanten Anzeigen selbst übernehmen können.

Erscheinungstermine der Dateien 2004:

- * 29.08.2004
- * 31.10.2004
- * 26.12.2004

Erscheint als Dateiformat: Rich Text Format (RTF).

Es ist jetzt endlich sicher gestellt, dass die Anzeigen bearbeitet werden können, fast schon egal, wo ich auf dieser Welt bin. Die Postanschrift ist gültig bis Anfang Oktober 2004. Danach muss ich wechseln.

Anzeigen bitte in deutscher oder englischer Sprache per Post oder E-Mail (derzeit nur als ASCII-.txt-Datei, nicht gepackt, Datei als Anhang) an unten stehende Adresse. Post aus Europa dauert circa eine Woche. E-Mails derzeit nicht größer als 8 kb pro Person. Angenommen wird alles, was irgendwie mit Computern zu tun hat, insbesondere Kontaktanzeigen. Eine Rubrikliste existiert bisher nicht, es wird eine neue auf Basis der eingehenden Anzeigen aufgebaut und so dem Markt neu angepasst. Sprachen bisher: Deutsch und Englisch.

Die Listen werden so gehalten, dass die Anzeigen für jedes System zusammengefasst stehen, so dass sie ohne Probleme übernommen werden können. Es werden nur ernstgemeinte und sachliche Anzeigen erfasst, um einen hohen Standard zu haben. Aus den eingehenden Anzeigen wird ein RTF-Dokument erstellt, welches den beteiligten Magazinen und Clubs wieder zum eigenen Abdruck kostenlos zur Verfügung gestellt wird. Außerdem werden die Daten auch allen Beteiligten zur Verfügung gestellt. So soll ein systemunabhängiges und vereinigendes Medium aufgebaut werden, das allein zum Ziel hat, den Pioniergeist speziell der Computertechnologie zu bewahren, zu verfechten und der Öffentlichkeit zugänglich zu machen sowie natürlich zu fördern. Mit der Zeit soll dann jedoch auch ein komplett eigenes Heft gemacht werden, wobei immer die Zusammenarbeit mit Clubs und anderen Magazinen in der ganzen Welt angestrebt wird.

Telefonanrufe derzeit am besten ab 16 Uhr Ortszeit (ab 22 Uhr MEZ in Europa bzw. Montag ab 5.00 AM ganztags) oder nach Absprache per E-Mail. Gib diese Information bitte auch weiter.

Euer Alexander Ausserstorfer

Alexander Ausserstorfer
193, avenue Morel, Kamouraska
GOL 1L0, Kanada
Telefon: +1-418-492-1240
(Zeit von New York, -6h MEZ)
E-Mail:
greentears@chiemgau-net.de

Retrowel t

Ein Traum vom Team17

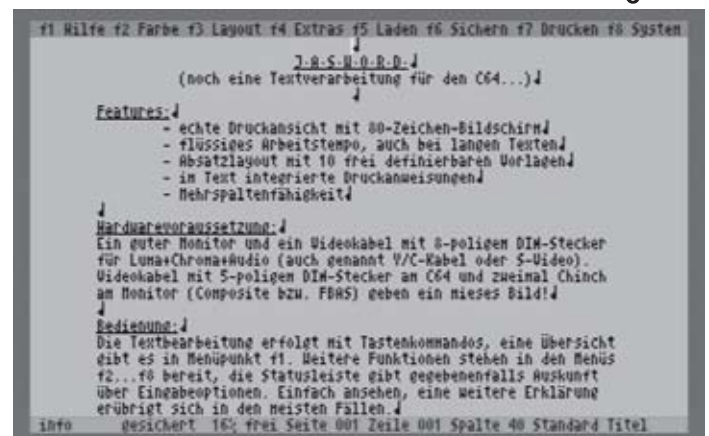
Dream17 ist eine Seite für Fans der legendären Spieleschmiede Team17, die der Brite Ben Paddon vor einem Jahr ins Leben gerufen hat. Ben, der seit mittlerweile zwölf Jahren süchtig nach den Spielen von Team17 ist, beschränkt sich im Gegensatz zu den meisten anderen Seiten dieser Art nicht auf die Worms-Serie, sondern möchte zum ersten Mal die vielen anderen Spiele in ihrer Gesamtheit würdigen. Die Seite wird diesem Anspruch auch gerecht, wovon man sich

jederzeit überzeugen kann. Neben einer Menge Downloads gibt es viele andere Bereiche, die allerdings noch nicht alle mit Leben und Inhalten gefüllt sind. Besonders Amiga-Besitzern sei die Seite ans Herz gelegt, schließlich stammen Hits wie Alien Breed, Apidya, Miami Chase, Project X, Superfrog und die bekannte Worms-Reihe von Team17.

>> <http://www.dream17.co.uk>



JASWORD: Neue C64-Textverarbeitung



Franz Kottira, Organisator des einzigen österreichischen Szene-Ereignisses, des Commodore Meeting Wien, hat eine Textverarbeitung für den C64 veröffentlicht. Da das

Programm mit einem 80-Zeichen-Bildschirm arbeitet, ist ein Monitor mit Chroma/Luma-Kabel Voraussetzung. Das kostenlose Programm gibt es hier:

>> <http://members.chello.at/wiener.freiheit/jasword.htm>

kurz und buendig

Neues und Altes von C-1 und Amiga OS4

Das erwartete Veröffentlichungsdatum des C-1 wurde mittlerweile um mehr als ein Jahr überschritten. Wer daraus den Schluss zieht, dass der C-1 nun gar nicht mehr erscheinen wird, dürfte falsch liegen. Jens Schönfeld teilte am 22. August mit, dass die Arbeiten am Wunderboard zügig voranschreiten. Der C64-Kern läuft nun schon sehr zufriedenstellend, der IEC-Bus wurde inzwischen implementiert. Damit ist es nun möglich, Programme von einer 1541 zu laden. Öffentliche Demonstrationen mit Boulder Dash, Paradroid und Giana Sisters zeigten die Leistungsfähigkeit des C-1, da zwei der drei

erwähnten Spiele Decruncher verwenden, die mit illegalen Opcodes arbeiten. Jeri Ellsworths „compatibility core“ scheint keine Probleme damit zu haben. Auf ein Veröffentlichungsdatum wollte sich Schönfeld aber nicht festlegen.

Weniger gut sieht es beim Amiga OS4 aus: Zwar wurde endlich eine Betaversion ausgeliefert, allerdings haben inzwischen selbst viele der Geduldigsten die Hoffnung aufgegeben, jemals mit einem Amiga One unter OS4 zu arbeiten. Die Firma Amiga, Inc. hat außerdem die Rechte am Amiga OS verkauft, die Zukunft des Systems ist also genauso ungewiss wie in den vergangenen zehn Jahren. Die Amiga One-Hardware, schon bei der Veröffentlichung vor etwa zwei Jahren etwas angestaubt, ist mittlerweile angesichts der rasanten Entwicklung am PPC-Markt (siehe Apple) total veraltet.

Amiga-Fans werden sich wohl endgültig vom Traum einer zeitgemäßen Amiga-Plattform verabschieden müssen. Zum Glück ist die Emulation weit fortgeschritten, sodass man sich wenigstens um die Zukunft der klassischen Software keine Sorgen machen muss.



Jeri Ellsworth mit dem C-1-Board.

The Infinite Loop, Juni 2004

Der US-Zeitschrift The Infinite Loop, der – wie berichtet – die Einstellung drohte, hat nun genügend neue Abonnements geworben, um ihr Erscheinen fortsetzen zu können. Die bunte Sommerausgabe lockt mit einer Menge Fotos von der Spring Expo, die in Louisville, Kentucky abgehalten wurde. „Star-gäste“ waren der in letzter Zeit in Misskredit geratene Maurice Randall, Jim Butterfield und C-1-Designerin Jeri Ellsworth.

Abo: The Infinite Loop, P.O. Box 746, Grand Junction, CO 81502, USA. Europa-Abo: 37 US-D.



Defender of the Crown++++++

Eines der eindrucksvollsten Spiele für den Commodore 64 wurde von Nostalgia endlich in einer kompletten und verbesserten Fassung veröffentlicht. Nostalgia, mittlerweile bei 167 Veröffentlichungen angelangt, haben dem 1987 erschienenen Action/Strategiespiel Defender of the Crown einen Screen zurückgegeben, der bei fast allen verfügbaren Versionen – das Original ausgenommen – gefehlt hat. Schuld daran ist der weit verbreitete Eaglesoft-Crack, bei dem der Bildschirm, der im Original nach einem gewonnenen Turnier erscheint, gefehlt hat. Das ist aber nicht alles: Die Nostalgia-Fassung enthält ein Doc-File und einen Trainer mit neun Cheats. Unglaublich: Das Programm wurde auf eine einzige Diskettenseite gepackt.

>> <http://nostalgia.c64.org>



Digital Talk 68

Ende August ist die 68. Ausgabe des deutschen Diskettenmagazins Digital Talk erschienen. Unter anderem erzählt GALAKTUS, der neben vielen anderen Aktivitäten auch die Lotek64-Homepage betreut, wie er zum C64-Freak wurde. Lesenswert wie immer.



>> <http://www.digitaltalk.de>

Archon, ein immer noch sehr beliebtes Schach-Ballerspiel-Crossover aus dem Jahre 1983, wird unter Einbeziehung von Elementen des Original-Codes für Windows neu programmiert.

>> www.archonevolution.com



Mit Low-Tech geht es auch. Eine sichere und bequeme Wohnung erfordert nicht unbedingt einen erhöhten technischen Aufwand – wichtig ist die Abstimmung auf Ihre Bedürfnisse. otto-konsum

Third Places / steirischer herbst 2004

Am 16. Oktober werden C64 und Atari VCS gefeiert!

Im Rahmen einer Veranstaltungsreihe des Festivals steirischer herbst gibt es in Graz zwischen dem 9. Oktober und dem 7. November Filme, Vorträge und Gespräche, Workshops, Live-Acts zu den Themen Fußball, Games und Musik-Clips. Am 16. Oktober können Besucherinnen und Besucher einen Nachmittag lang in die Geschichte der Computer-Demos eintauchen, die in der Heimcomputerszene bis heute eines der wichtigsten Ausdrucksmittel darstellen. Hiram Kümper wird einen Überblick über die Geschichte der Demos am Commodore 64 bieten, während Simon Quernhorst demonstriert, was man der antiken Atari 2600-Konsole, der ersten Spielkonsole, die die Kinderzimmer der ganzen Welt eroberte, heute noch alles entlocken kann.

„Third Places“ beschreiben in Abgrenzung zum „First Place“, dem Zuhause, und dem „Second Place“, dem Arbeitsplatz, informelle halböffentliche Räume, die über ihr Grundangebot

hinaus wichtige soziale Funktionen für die Gesellschaft übernehmen, ähnlich dem Marktplatz im Mittelalter, der Gemeinschaftsbildung ebenso dienend wie der Unterhaltung.

Geprägt wurde der Begriff von Ray Oldenburg in seinem 1989 erschienenen Buch „The Great Good Place“, in dem er den zunehmenden Verlust der Third Places beklagte, ohne die

eine Gesellschaft jedoch nicht funktionieren könne.

Als Beitrag zu dieser Debatte spürt die Programmschiene *Third Places* durchaus kritisch anhand der für 2004 gewählten Themen *Fußball*, *Games* und *Musik-Clips* räumlich und inhaltlich Stätten und Bereichen einer mit den Third Places verbundenen visuellen und medialen Alltagskultur nach, die an der Schnittstelle von elitärer Hoch- und populärer Eventkultur, von Massen- und Insiderphänomenen, von Unterhaltungswert und kreativem kulturellem Mehrwert liegen.

Simon Quernhorst, Atari VCS 2600

Simon: In früher Kindheit hatte ich bereits Kontakt mit einer Pong-Konsole, anschließend mit Atari VCS und danach Commodore 64. Jahrelang bin ich in der C64-Szene aktiv gewesen, bevor ich beschlossen habe, neue Programme für Atari VCS zu entwickeln. Besonders der Minimalismus der alten Systeme hat es mir angetan – mit geringen Mitteln trotzdem ansprechende Effekte darzustellen. So benötigt mein Spiel *Mental Kombat* inklusive aller Grafiken und Funktionen lediglich 4 KB. Die Einfachheit der Pixel und die Erkennbarkeit der Bildpunkte als solche finde ich als abstrakte Darstellungsform sehr reizvoll.

LoteK64: Was wirst du in Graz präsentieren?

Simon: Die Demoerstellung für Systeme mit Diskettenlaufwerken hat ja bereits lange Tradition, da die Demos kostengünstig weitergegeben werden konnten und die Entwicklung direkt auf den Homecomputern erfolgen konnte. Bei den frühen Konsolen wurden Demos eigentlich nur des Ausprobierens wegen auf den Entwicklungssystemen entwickelt und nicht, um ein Demo zu schaffen und zu verbreiten, welches ein bestimmtes Thema oder speziell darauf abgestimmte Effekte darstellt.

Um diese Lücke zu schließen, erstelle ich derzeit ein Multipart-Demo für das Atari VCS, welches als Thema auch den Steirischen Herbst in Graz behandelt. In Zeiten der Emulatoren auf PCs können Programme für alte Konsolen nun leichter entwickelt und auch als Dateien ausgetauscht werden. Das Demo werde ich auch als Modul in geringer Auflage produzieren, damit auch andere echte Retro-Sammler das Demo auf der Originalhardware ansehen können.

LoteK64: Auf welchem System programmierst du?

Simon: Ich bin generell Sammler von alten Video- und Computerspielen und entwickle derzeit neue Spiele für das Atari VCS (Video Computer System, später 2600 genannt). Das System wurde 1977 auf den Markt gebracht und besitzt lediglich 128 Byte RAM, den man sich auch noch mit dem Prozessor (Stackverwaltung) teilen muss. Neben zwei Audiostreamen, kann das Gerät zwei einfarbige Sprites mit jeweils nur acht Pixeln Breite darstellen – erst durch entsprechende Programmierung kann man diese Objekte vervielfachen. Im Gegensatz zu anderen alten Systemen liegt der einzige Vorteil in der (für damalige Verhältnisse) großen Farbpalette von bis zu 128 Farben (bei NTSC-Geräten). Zum Vergleich: der C64 besitzt

lediglich 16 Farben. Die Programmierung erfolgt im 65xx-Assembler, die anschließende Kompilierung erzeugt dann Maschinensprache (so wird z.B. der Assemblerbefehl „NOP“ übersetzt in den Instruktionscode \$EA). Da viele alte Geräte (z.B. Atari VCS, C64, NES) ähnliche Prozessoren besitzen, kann man das Wissen um diese Sprache auf verschiedenen Plattformen nutzen, lediglich die Systemarchitektur ist individuell.

LoteK64: Seit wann beschäftigst du dich mit diesem System?

Simon: Das Atari VCS besitze ich seit Anfang der 80er Jahre, den C64 seit 1987. Im Laufe der Jahre kamen alle (für mich) relevanten Systeme hinzu. Neben C64 und VCS sind PC-Engine, SNK NeoGeo AES und Sega MegaDrive meine Lieblingssysteme, da ich kurzweilige Arcade-Spiele bevorzuge.

In meiner C64-Zeit bin ich in diversen bekannten Gruppen gewesen und habe auf diesem System zuletzt auf



Simon Quernhorst, Atari VCS

THE PARTY in Aars (DK) eine C64-Grafik (Platz 2) und ein C64-Demo (Platz 5 in der Wild-Compo) herausgebracht. Neben der Entwicklung von eigenen Demos und Spielen habe ich auch die Special Editions der neuen C64-Spiele „Metal Warrior Trilogy“ und „Metal Warrior 4“ produziert. Um etwas ganz Besonderes herzustellen, erscheinen die produzierten Spiele nur in geringer Auflage und mit besonderen Gegenständen wie bisher z.B. Stirnbändern, Ansteckbuttons, Soundtrack-CDs und Dog-Tags. Gerne hätte ich auch eine Special Edition des neuen Spiels *Turrican 3* aufgelegt, wegen rechtlicher Unklarheiten haben der Programmierer und ich aber leider Abstand davon nehmen müssen.

Simon Quernhorst, 28, lebt in Wesel (NRW) und ist Consultant für Personalwirtschaftssysteme.



Simon Quernhorsts Demo für die klassische Atari VCS-Konsole wurde exklusiv für den steirischen Herbst programmiert.

Der Titel Third Places steht dabei als Metapher für Positionierungsfragen im Kunst- und Kulturbereich.

Games (Computer- und Videospiele)

Ziel der geplanten Veranstaltungen ist es, das Bild und die Identität der Spiele und Spieler genauer zu hinterfragen und die Bedeutung der Spiele-Kultur als wesentlicher Bestandteil der Gesellschaft des 21. Jahrhunderts in ihrer Positionierung zwischen Wissenschaft und Unterhaltungskultur einer Neubewertung zu unterziehen, um die Verständnisbarrieren zwischen den Generationen abzubauen. Am 16. Oktober klinkt sich auch Lotek64 in die Veranstaltungsreihe ein. Im Rahmen des Games-Projektes wird Hiram Kümpfer ein C64-Demo sowie einen Überblick über die Demo-Geschichte dieses Systems präsentieren, während Simon Quernhorst seinem Atari 2600 Erstaunliches entlockt. Das Publikum hat die Möglichkeit, den beiden Codern beim Programmieren zuzuschauen, anschließend werden die Ergebnisse auf Großleinwand zu sehen sein. Die Programme werden für Third Places erstellt und demonstrieren die Fähigkeiten der betagten Computersysteme.

Lotek-Exkurs: barbarian ware

Zu Beginn der Eisenzeit taucht an griechischen, kretischen und zyprischen Fundorten rückständige, hand-



Ausschnitt aus Simon Quernhorsts Atari-Demo.

Hiram Kümpfer, Commodore 64

Hiram: Zum einen möchte ich einen Parforce-Ritt durch die Demo-Geschichte des Commodore 64 machen. Das ist recht interessant und man kann ganz wunderbar Entwicklungslinien aufzeigen, sowohl technischer als auch inhaltlicher Art.

Zum anderen bringe ich ein Demo-Projekt aus der Civitas-Schmiede mit, das „EUDAIMONIA“, also „Glückseligkeit“ heißt. Es soll einige klassische Antworten der Philosophiegeschichte auf die Frage nach dem Glück verketteten – das ist der Rahmen. In praxe heißt das, eine Anreihung von Hires-Portraits berühmter Philosophen mit Zitaten und einigen kleineren Effekten.

Lotek64: Wie bist du zum Programmieren gekommen? In welcher Sprache programmierst du?

Hiram: Ich programmiere praktisch nur auf dem C64, kleinere Versuche mit dem Amiga, die ich schnell wieder aufgegeben habe, ausgenommen. Die Anfänge waren dabei wohl die üblichen: Über BASIC hin zu Assembler. Ich habe an einigen Demo-Projekten programmiertechnisch eher an der Seitenlinie mitgearbeitet, ansonsten aber hauptsächlich Tools, sowie ein Hypertextsystem namens „CAMEL“ (Commodore Activist's Markup Extention Language) ähnlich HTML zur Verwendung für digitale Fanzines entwickelt.

Insgesamt sehe ich mich aber auf dem C64 eher als Graphiker.

Meine Programmierfähigkeiten sind bescheiden und dank der Arbeitsteilung der verbliebenen C64-Freunde in Programmiergruppen können den Programmierpart auch weit fähigere Leute übernehmen. Auf dem PC habe ich einige kleinere wissenschaftliche Hilfsprogramme in Python geschrieben, was ich in Zukunft, wenn es die Zeit zuließe, gerne noch etwas intensiver machen würde, weil diese Programmiersprache in meinen Augen eine ganze Menge Potential hat.

Lotek64: Seit wann beschäftigst du dich mit diesem System? Was gibt es über deine Gruppe Civitas zu wissen?

Hiram: Ich weiß offen gestanden nicht mehr ganz genau, wann ich meinen ersten C64 gekauft habe, es war in jedem Fall irgendwann Ende der 80er. Gespielt habe ich nach der wohl üblichen anfänglichen Begeisterung seltsamerweise recht bald nicht mehr viel, sondern ich begeisterte mich dann für BASIC. Das auch ziemlich lange, denn ich war gehörig widerborsig gegen die Einsicht, dass man ohne Assembler leider im Programmieren nicht weit kommt. 1995 haben ein Freund und ich zusammen ein digitales Fanzine, die PUBLICATION, aufgezogen, die es auch tatsächlich noch immer gibt. Ende 1996 ist daraus die Gruppe CIVITAS entstanden, die mit präntösen, lateinischen Titeln und aufgeblasenen Sprüchen wie „Living proof against darwinism“ noch immer existiert und tatsächlich, ja



Hiram Kümpfer, Commodore 64

man wundert sich, noch immer Spaß an der Sache hat. Wir sind auch ein ganz munterer Haufen, einige sind schon sechs oder sieben Jahre dabei. CIVITAS ist auch der eigentliche Grund, wie ich – nämlich aus Ermangelung eines eben solchen – zum Graphiker geworden bin. Nachdem ich lange hauptsächlich Logos gemacht habe, gilt meine Liebe nun seit vielleicht zwei Jahren dem einfarbigen Hires-Modus, mit dem man ganz wunderbare Dinge, so wie Zeichnungen auf einer Serviette oder einem Bierdeckel, machen kann. Das ist auch, zu meinem Bedauern, ein aussterbender Ast, will sagen: Graphikmodus, in der C64-Szene.

Hiram Kümpfer ist Historiker und Philosoph, verdient seinen Lebensunterhalt am Historischen Institut der Ruhr-Universität in Bochum und promoviert über mittelalterliche Rechtsbücher.

gemachte Keramik auf, obwohl bereits jahrhundertlang mit der Töpferscheibe gearbeitet worden war. Diese Keramik, manchmal als „barbarian ware“ bezeichnet, verfügt über ein Pendant in der Welt der Computerspiele, die in den 80er Jahren des



20. Jahrhunderts noch klar als solche identifizierbar waren. Heute verschwimmen zusehends die Grenzen zwischen einzelnen am Computer generierten Medien: Durch die rasant technische Entwicklung und die Verfügbarkeit der benötigten Hardware (3D-Grafik, Surround-Sound) simulieren Computerspiele realistisch wirkende Welten. Nicht alle, die noch mit pixeliger Klötzchengrafik und monoton piepsenden Computern aufgewachsen sind, wollen die Errungenschaften der Pionierzeit widerstandslos aufgeben und produzieren abseits kommerzieller Interessen Programme

für ihre Computer, die oft durch hohe künstlerische Qualität bestechen.

Am 16. Oktober werden zwei Vertreter dieser Szene, Hiram Kümpfer und Simon Quernhorst (siehe die Interviews auf diesen Seiten) „Barbarianware“ präsentieren – sowohl im historischen Überblick als auch als zeitgenössisches Ausdrucksmittel.

Third Places Schwerpunktprogramm

9.10 Eröffnungsveranstaltungen, Askö-Stadion

Vortrag, Match und Konzert

17h Eröffnungsvortrag Büro für kognitiven Urbanismus (Wien), *Asynchrone Räume: Architekturen des Temporalen*

18 h Eröffnungsmatch *Beruf: Homeless*

20.30 h Übertragung auf Großleinwand WM Qualifikation Österreich – Polen

22.30 h Konzert Binder & Kriegelstein und Jan Pieter Martens

10.10. Schwerpunkt Fußball, Askö-Stadion

Film, Vorträge, Gespräche und Fußballworkshops

11h Film *Rudi Rastlos – der Ball rollt für Ruanda*

13-14.30h *When Saturday comes* (zu Fußball/Film)

16-17h *Wir kommen von weit her und wir werden noch weit gehen* (zu Fußball/Arbeit/Migration)

19-20.30h *Die zwölften Männer* (zu Fußball/Welt)

Vortragende: Muhsin Ertugral (SV Mattersburg), Roman Horak (Wien), Hermann Kern (Kapfenberg), Olaf Möller (Köln), John Williams (Leicester)

GesprächsteilnehmerInnen: Max Annas (Köln), Martin Behr (Graz), Ulrike Davis-Sulikowski (Wien), Stefan Erhardt (München), Werner Fenz (Graz), Christine Frisinghelli (Graz), Piet Fuchs (Köln), Michael Glawogger (Wien), Rudi Gutendorf (Koblenz), Horst Heldt (VfB Stuttgart), Josef Hickersberger (Rapid Wien), Reinhard Krennhuber (Wien), Gerhard Koch (Graz), Brigitte Kowanz (Wien), Philipp Köster (Berlin), Helmut Kronjäger (StfV Graz), Martin Madlmayr (Graz), Jan Pieter Martens (SC Unteriebenbrunn), Eric Orie (FC Lustenau), Christian Petzold (Berlin), Gilbert Prilasnig (Graz), Hubert Rieß (Graz), Klaus Salmutter (Sturm Graz), Frankie Schinkels (SKN St. Pölten), Peter Schöttel (Rapid Wien), Gerhard Stocker (FC Wacker Tirol), Armin Thurnher (Wien), Vitus Weh (Wien), Wolfgang Weisgram (Marz)

14-20.00h „just flick to kick!“, Subbuteo Tischfußballturnier initiiert von Stefan Sandner (Wien)

14.30-16h und 17.30-19h Trainingsworkshops mit Jan Pieter Martens, Gilbert Prilasnig u.a.

(beschränkte Teilnehmerzahl, Voranmeldung unter 0316 823007-27)

ca. 20.30h DJ Lines - Chris Duller, Michael Hatz u.a.

Fußballfilme im Non-Stop-Kino am 24.10. 11 - 23 h siehe Folder oder www.steirischerbst.at

15.10. – 17.10. Schwerpunkt Games, Pädagogische Akademie

Vorträge, Podiumsdiskussionen, Lehrerworkshop, Live-Acts

15.10. *born Klicking: die Netzwerkkinder und ihre Spielekultur*

14-18h Vorträge von Konrad Becker (Wien); Simon Egenfeldt-Nielsen (Copenhagen); Mathias Petzold (Köln); Tom Savigar (London); workshop mit Sylvia Eckermann (Wien)

18h Podiumsdiskussion

16.10. *Leben im Netz: communities & identities*

13-19h *barbarian ware*, eine C64/Atari2600-Performance, organisiert von Lotek64 (Graz)

14-18h Vorträge von Mary Flanagan (New York); Beate Geissler und Oliver Sann (Köln); plan-B (Graz); Jan Schob (Celle);

18h Podiumsdiskussion

17.10. *Spiele als Crossdisziplin in Wissenschaft & Kunst*

14-18h Vorträge von Conrad Bodman (London); Jesper Juul (Copenhagen); Katharina Schlieben

(Zürich); Mathias Fuchs (Manchester); Orhan Kipcak (Graz)

18.30h Live-Act toise (Graz)

22.10. – 23.10. Schwerpunkt Clips, Non-Stop-Kino

Filmprogramm, Symposium, Konzert

22.10. 16-18h *Musik-Clips: Einführung, Ökonomien, Strategien und Öffentlichkeiten*

18h Filme und Clips

Breaking Free - Frühe Pop-Filme (1960-75)

Filme und Clips von Bruce Conner/Ray Charles (1961), Jud Yalkut/ The Beatles (1966), D. A. Pennebaker/Bob Dylan (1967), M. Jon Rubin (1967-69), Bruce Conner/Ed Cobb/Toni Basil (1966), Gunvor Nelson/Pat Gleeson (1972), Karpo Godina (1971), Pat O'Neill/T. Rex (1972),

19.30h *Breaking Through* - Proto-MTV-Clips und frühe Musikvideos (1975-85)

Filme und Clips von Gerald Casale & Chuck Statler/Devo (1976), Bruce Conner/Devo (1978), Graeme Whifler/The Residents (1978), Dara Birnbaum/Rhys Chatham (1980), Paul Dougherty & Walter Robinson/Suicide (1978), Bruce Conner/Brian Eno–David Byrne (1982), Godley & Crème/Herbie Hancock (1983), Zbigniew Rybczynski (1984), Jean-Baptiste Mondino/Madonna (1986)

Barbetrieb open end

23.10. 14-18h *Musik-Clips: Ökonomien, Strategien und Öffentlichkeiten*

18 h Filme und Clips

Going Nowhere - Ortsuche und Körperformatierung im Pop-Film nach 1988

Filme von James Herbert/R.E.M. (1988), Sadie Benning/Come (1995), Jem Cohen/Sparklehorse (2001), Mike Mills/Air (1998), W.I.Z./Flowered Up (1992), Steve Hanft/Beck (1994)

19.30h *Over the Edge* - Ortsuche und Körperformatierung in aktuellen Clips (1998 bis heute)

Roman Coppola/Phoenix (2002), Deborah Schamoni & Ulli Lindemann/Die Goldenen Zitronen (1998), Jonas Odell/Franz Ferdinand (2003), Shynola/Radiohead (2001), Graw Böckler/Ulf Lohmann (2002), Sandeep Mehta/Thomas Brinkmann (2003), Traktor/Fatboy Slim (2001), Michel Gondry/Steriogram (2004), Anna Berger & Michel Klöforn/Sensorama (2001), Garth Jennings/Beck (2002)

21h Konzert mit Michael Posch, Robert Lepenik/Helmut Kaplan, REAS von tonto (Graz) und Visuals des Studiengangs für Informationsdesign der FH Joanneum (Graz) von Stefan Fauland, Tina Fleck, Philipp Hoffer, Jonathan Hoier, Markus Hornof, Marko Hödl, Katrin Höfling, Michael Kara, Wolfgang Kolar, Margareta Liebmann, Iris Moustakidis, Karin Pichler, Armin Reautschnig, Evelyn Schlagbauer, Christian Schrei, Elvira Stein, Sigrid Thallinger, Michael Wagner

Der Grazer Künstler Constantin Luser begleitet das Gesamtprojekt mit einer Illustration, die auf verschiedenen Ebenen eingreift und auf die inhaltlichen Schwerpunkte reagieren wird, u.a. im Katalog sowie im öffentlichen Raum im Rahmen einer Kooperation mit dem Plakatprojekt *Kunst auf Zeit 2004 ALLOVER*. Seine Plakatflächen, subtile grafische Markierungen urbaner Linien, überziehen die Stadt, um dann im Zuge der Neuvermietung der Standorte in einem fading-Effekt langsam zu verschwinden.

Die *crew8020* setzt seine seit 2001 bestehende Veranstaltungsserie *quodlibet* fort. Unter dem Aspekt der Grenzaufhebung zwischen third und first place werden unter anderem die Schwerpunktthemen Fußball, Games und Musik-Clips behandelt. Homeless World-Cup Spieler und Organisatoren, Verehrer der Extremmusik, Computerfreaks mit Vorliebe zum Motorsport, Ärzte mit Fußballstadien, Künstler mit Tulpengärten, Kuratorinnen mit Hotelzimmer und weitere Bewohner dieser Stadt präsentieren sich und ihre Intimsphäre im privaten Raum.



Erlebniswelten

Vom Computing-Baukasten der Firma kosmos bis zum C64

1969: Aufbruchstimmung, Apollo-Mission, erste bemannte Mondlandung erfolgreich abgeschlossen. 68er-Bewegung, Studentenrevolten, the beat goes on und so auch die Technik. Erfindergeist war zur Genüge vorhanden, siehe Band- und Plattenanlagen, Lochkarten im Großformat. Zeit also für private Computer, die sich jeder leisten kann.

— von Mathes Alberto —

Anfangs waren nach heutiger Währung etwa 10.000 Euro nötig, wenn man als Privatperson in den Genuss eines Computers kommen wollte. Es waren vor allem der spezialisierte Fachhandel und Computershops, die die Geräte im Angebot hatten. An Hobbygeräte dachte dabei noch niemand. Auch die geringe Auswahl an Programmen für Geschäftszwecke bzw. aus dem Reich der Text- und Datenbanksoftware. War der Computer „mittels BASIC und/oder Maschinensprache“ zu programmieren, natürlich am Monochrom-Monitor und begleitet von Piepstönen, später sogar in Farbe und in Stereoqualität, wusste man, dass man es mit einem besonders leistungsfähigen Gerät zu tun hatte. Vielleicht war er sogar mit zwei Diskettenlaufwerken ausgestattet...

Computer im Eigenbau

Die Entwicklung führte bekanntlich zu Systemen wie XT/AT, 286, 386, 486, Pentium 1, 2, 3, 4 etc. Bemerkenswert ist aber vor allem die Epoche der Vielfalt. In den 1970er und 1980er Jahren hatte noch kein System totale Marktdominanz errungen. Größere Bevölkerungskreise wurden von der Industrie motiviert, an der Entwicklung teilzunehmen. Genährt wurde die Technikbegeisterung durch Ereignisse wie die Mondlandung, die ohne moderne Datenverarbeitung nicht möglich gewesen wäre. Daran anknüpfend warf die Firma kosmos, bekannt durch Experimentier-

kästen, einen Baukasten für „computing“ auf den Markt, was heute nur noch wenigen bekannt sein dürfte. Stichwort: „Bau dir selber einen Computer“ mit einfachsten Mitteln, großer Baukasten mit wenig Inhalt nach dem Auspacken, zudem nicht ganz billig für Kabel, Schiebereglerkontakte, Steckbuchsen und einige Miniglühlampen, die die Funktionen sichtbar machten. Dazu ein Plastikgehäuse zum Einbau, insgesamt konnten um die 200 Verbindungen erstellt werden.

Je nach Schalt- ähm: Logikplan des Programmes mussten zuerst Kabel zugeschnitten und auf der Platte eingesteckt werden. Dabei ging es um Logik von „ja“ und „nein“. Die Schieberegler steuerten dann den Funktionsablauf, der die Reaktion mittels Glühlampen ausgab. Selbst für einfachste Operationen waren meist mehr als 20 Kabel zu verbinden. Bei so vielen Kontakten ging die Überschaubarkeit schnell verloren und die Fehlersuche erwies sich als schwierig. Oft lag es an Kontaktproblemen, wenn ständig unerwünschte Ergebnisse erzielt wurden. Immerhin: ein eigener Computer und somit etwas zum Erlernen der neuen Technik. Bunte Bastelstunde eben.

Die Firma kosmos brachte später noch einen weiteren Compibaukasten auf den Markt, der im Prinzip wie ein ZX-Computer aufgebaut war: so klein und diesmal voll elektronisch. Nun ging es nicht um das Basteln, sondern erstmals um echte Programmierung. Zur Ein- und Ausgabe gab es daher auch eine kleine Tastatur.

Der Heimcomputer aus dem Kaufhaus

Richtiges Homecomputing fing für die Mehrheit der User aber erst mit Geräten wie dem Commodore VC-20 (als Volkscomputer tituliert) an. Weil dieser einfach per Antennenkabel an TV-Geräte angeschlossen werden konnte und zudem aufgrund seiner speziellen Chips für Grafik und Musik sehr leistungstark erschien, kam auch bald eine Menge Software und Zubehör auf den Markt. Erstmals setzte sich auch in Europa die Datensette als einfacher Massenspeicher durch, in den USA hatte dieses Medium bereits durch die Verwendung in frühen Apple-Systemen viele Anhänger gefunden. Datenverbreitung war nun kommerziell wie auch privat möglich, das mühsame Abtippen von langen Listings aus Fachbüchern und -zeitschriften gehörte nicht mehr zum Alltag des Users.

Der große Vorteil von Rechnern wie dem VC-20 und dann des Commodore 64 waren die vielen Schnittstellen, an die allerlei Zusatzgeräte angeschlossen werden konnten: Joysticks, Maus, Lightpen, Lenkrad, Paddles, Tastaturen, vor allem aber Drucker, Plotter, Diskettenlaufwerke. User- und Expansionsport boten den Usern fast grenzenlose Möglichkeiten, sich diverse Erweiterungen nutzbar zu machen oder selbst zu bauen. Erweiterungen wie das Action Replay-Modul oder Final Cartridge 3 etc. waren bald aus dem C64-Alltag nicht mehr wegzudenken. Die Leistungsstandards wurden dadurch in die Höhe getrieben. Das Internet ist dem C64 nicht verschlossen geblieben, mittlerweile können CD-ROM-Laufwerke und Festplatte angeschlossen werden, wenn man über das nötige Kleingeld verfügt. Für viele blieb der Heimcomputer vor allem eine Spielkonsole, die nebenbei auch mit Textverarbeitungsprogrammen gefüttert werden konnte. Gerade am C64-Sektor haben aber die User dazu beigetragen, dass sich die Begeisterung für diesen Computer bis heute gehalten hat.



Die Baukästen der Firma kosmos machten sich schon lange vor den Heimcomputern in den Kinderzimmern breit. Der junge Vulkanier im Bild brütet gerade über einem logischen Problem.

Der japanische Amiga: X68000

1987 bricht bei uns in Europa das 16-Bit-Zeitalter an. Die Rede ist von Atari ST und Commodore Amiga. Auch in Japan wird gleichzeitig ein neuer 16-Bit-Rechner auf den Markt gebracht, der technisch durchaus mit Amiga und ST mithalten kann. Leider kennt hierzulande auch heute noch kaum jemand diesen Computer der Firma Sharp. Die Rede ist vom X68000.



Äußerlich unterscheidet sich dieser Rechner deutlich vom Amiga und ST, da er eine Art Doppel-Slim-Tower darstellt. Ausgestattet war der Computer mit zwei 5,25" Laufwerken (ob die Japaner schon damals wussten, dass 5,25"-Disketten länger halten?) und später auch mit einer Festplatte. An Ports fehlte es dem X68000 nicht: So hatte er zwei MSX-kompatible Joystickports sowie einen parallelen und einen seriellen Port. Insgesamt konnten bis zu vier Floppys angeschlossen werden. Außerdem gab es neben dem üblichen TV-Ein- und -Ausgang auch noch einen Anschluss für Headset und Mikrophon. Sechs Jahre später wurde die Rechnerserie mit folgender Grundausstattung abgeschlossen: 68030-CPU (25MHz), 4 MB RAM und 80 MB-Festplatte. Sharp brachte im Laufe der Jahre immerhin ca. 20 verschiedene Versionen des Computers auf den Markt.



Die Software

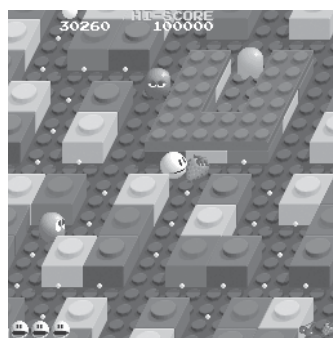
Der X68000 wurde – genau wie der Amiga – hauptsächlich zum Spielen benutzt. Das Angebot reicht von umfangreichen RPGs (die auch schon mal 13 Disketten füllen können) über Strategie- und Actionspiele bis hin zu sehr gelungenen Arca-



deumsetzungen. Das kommt daher, dass die Firmen meist selbst die Konvertierung übernommen haben und nicht wie auf dem Amiga Drittfirmen wie z.B. Elite, die das ganze Geld für Lizenzen ausgeben mussten und sich dann keine anständigen Programmierer mehr leisten konnten. So gut wie jede



namhafte Arcadefirma ist auf dem X68000 vertreten: Sega, Konami, Capcom, Taito, Namco usw. Die Titel kommen dem Original schon ziemlich nahe. Auch Hentai-Games (Sex-Manga) sind natürlich auf dem X68000 vertreten. Nur wenige westliche Firmen wagten es, für den exotischen Computer ihre Titel zu veröffentlichen: Als Beispiel sei Dungeon Master von FTL erwähnt. Wieviele X68000-Spiele insgesamt erschienen sind, ist leider nicht be-



kannt, doch X68000-Experte Keigo Oishi versichert, dass es sehr viele sind. Das letzte offizielle Spiel erschien am 26. Juli 1997. Es handelt sich um das „grow up game“ Princess Maker, welches auch für diverse andere Systeme erschienen ist. Danach war Schluss mit offizieller Software, es erschien nur noch

von Fans programmierte Software, so wie wir das bei uns kennen.

Tipps für Sammler

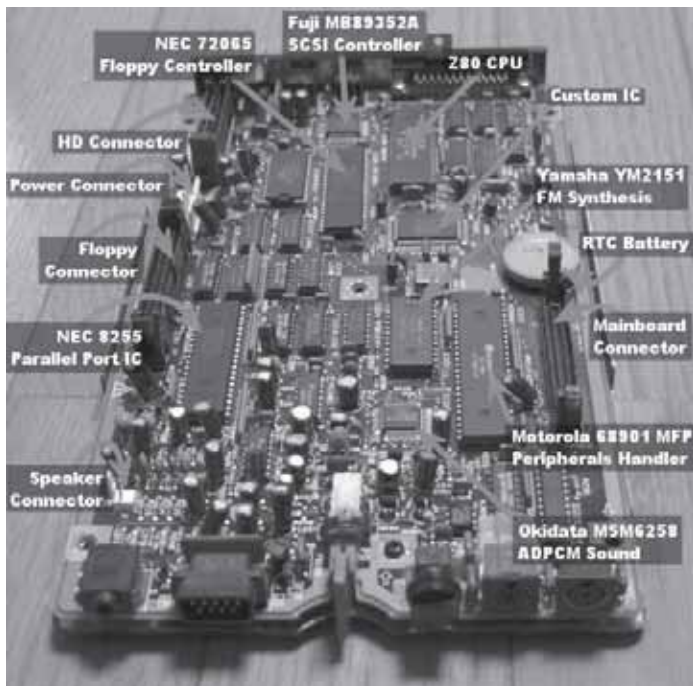
Wer sich einen X68000 zulegen möchte, dem sei gleich gesagt, dass die Preise bei 30.000-50.000 Yen liegen. Die Spiele kosten 1.000-

Technische Daten Sharp X68000

Veröffentlicht	Japan, 1987
CPU	Motorola MC68000, 10 MHz
RAM	1 MB RAM (aufrüstbar bis zu 12 MB) + 512 kB VRAM + 512 kB Text-RAM
ROM	1 MB
Video	256x240 / 256x256 / 512x240 / 512x256 / 512x512 Bildpunkte bei 65.000 Farben, 640x480 / 768x512 / 1024x1024 bei 16 Farben. Unterstützung für Sprites (16x16 Pixel)
Audio	2-Kanal-FM-Sound (Yamaha 2151, 8 Oktaven) + PCM (OKI MSM6258V, 4-Bit Mono)
Speichermedien	2 x 1MB-5,25"-Diskettenlaufwerk intern (Festplatte bei späteren Modellen)
Betriebssystem	Human 68K 1,0
Schnittstellen	2 x Joystick, Audio I/O, Stereoscope, NTSC Video I/O, 2 Erweiterungsports, Anschluss für externes Diskettenlaufwerk, SCSI, RS232

Technische Daten Sharp X68000 Expert II

Veröffentlicht	Japan, 1990
CPU	Hitachi HD68HC000, 10/15 MHz; Koprozessor 80C51
RAM	2 MB RAM (aufrüstbar bis zu 12 MB) + VRAM: 512 kB (Grafik) + 512 kB (Text) + 32 kB (Sprites)
ROM	BIOS (128 kB) + Zeichengenerator (768 kB)
Video	256x240 / 256x256 / 512x240 / 512x256 / 512x512 Bildpunkte bei 65.000 Farben, 640x480 / 768x512 / 1024x1024 bei 16 Farben. Unterstützung für Sprites (16x16 Pixel)
Audio	2-Kanal-FM-Sound (Yamaha 2151, 8 Oktaven) + PCM (OKI MSM6258V, 4-Bit Mono)
Speichermedien	2 x 1MB-5,25"-Diskettenlaufwerk intern (Festplatte bei späteren Modellen)
Betriebssystem	Human 68K 3,02 + SX-Window 3,01
Schnittstellen	2 x Joystick, Audio I/O, NTSC Video I/O, 2 Erweiterungsports, Anschluss für externes Diskettenlaufwerk, SCSI, RS232



5.000 Yen. Das Gerät selbst wiegt zwischen 8 und 10 kg, was die Versandkosten erheblich in die Höhe treibt. Außerdem wird ein Spannungswandler benötigt (Preis im Fachhandel ca. 10 Euro), da das Gerät nicht mit 220 Volt läuft und es mit einem lauten Knall das Zeitliche segnen würde. Man sollte

beim Kauf unbedingt darauf achten, dass das Gerät mindestens 2 MB RAM hat, da darunter die meisten Spiele nicht laufen. Eine Alternative sind die Emulatoren, die in der folgenden Liste aufgeführt sind.

Doraemon

X68000-Links

>> <http://www.ksky.ne.jp/~yamama/emul/index.htm>
(X68000-Emulator für Windows)

>> http://homepage2.nifty.com/m_kamada/java/x68000/
(X68000-Emulator für Java)

>> <http://www.zophar.net/x68000.html>
X68000-Informationen (englisch)

>> <http://www.jcec.co.uk/x68guide.htm>
Information über „EX68“-Software (englisch)

>> <http://www.zoom-inc.co.jp>
Eine Firma, die Spiele für das X68000-System entwickelt hat und diese jetzt als Freeware verschenkt.

X68000 Spiele Top 10 von Keigo Oishi

- | | |
|--------------|-----------------------------------|
| 1. Phalanx | 6. Gradius |
| 2. Granada | 7. Akumajou Dracula (Castlevania) |
| 3. Sol-Feace | 8. Geograph Seal |
| 4. Genocide2 | 9. Star Wars |
| 5. Bosconian | 10. Etoile Princess |



Ich möchte mich bei Keigo Oishi aus Japan bedanken. Ohne ihn und seine umfangreiche Hilfe wäre dieser Artikel nie zustande gekommen. Wer mehr über Keigo Oishi und seine Software erfahren möchte, kann seine Internetseite besuchen: <http://hp.vector.co.jp/authors/VA003645/>



POST

Hallo Lotek64!
Kürzlich bin ich auf das Interview mit Pooka im Rahmen der Superfrog-Story (Ausgabe 10) gestoßen, und obwohl ich nicht deutsch spreche, scheint mir der Artikel ziemlich umfassend zu sein. Ich sage das natürlich nur, um euch zu schmeicheln, damit ihr meine Webseite <http://www.dream17.co.uk> erwähnt. Es handelt sich dabei um eine Team17-Fanseite, bei der zur Zeit auch Pooka mitarbeitet (hier bitte ein fettes Grinsen einfügen.)
Ben Paddon

(Anm. der Red.: Wir haben die Seite angeschaut und geben die Empfehlung gerne weiter. Nähere Informationen gibt es auf Seite 4 bei den Neuigkeiten.)

In letzter Sekunde...

Turrican III ist da!

Mit T3 ist eines der besten C64-Spiele der jüngeren Vergangenheit entstanden. Den Codern von Smash Designs, die nun über eine (kommerzielle) Fortsetzung von Katakis nachdenken, haben hervorragende Arbeit geleistet und ein drei Diskettenseiten langes Ballerspektakel abgeliefert.



<http://www.smash-designs.de>

...und wer das Spiel gleich auf seinem PC oder Mac ausprobieren möchte, kann sich über die neue Version 1.15 des Emulators VICE freuen:

<http://www.viceteam.org>

Adventures programmieren

Ein BASIC-Tutorial von Paul Allen Panks

In diesem Programmierkurs zeigt uns Paul Allen Panks, wie man auf unkomplizierte Weise in BASIC Adventures programmieren kann. Das hier besprochene Spiel Orathan läuft so, wie es hier vorliegt, auf einem Commodore 64.

Als vor vielen Jahren die ersten Textadventures entwickelt wurden, verfügten die Computer über einen sehr begrenzten Arbeitsspeicher. Aus diesem Grund wurde anstatt Grafik meistens Text verwendet, um verzauberte Welten und mystische Landschaften zu beschreiben. Auch mythologische Monster wie Trolle oder Drachen wurden eher mit Worten als mit Bildern dargestellt.

Was Textadventures damals so populär machte, ist dasselbe, was auch Bücher so interessant macht: Vorstellungskraft. Wir lieben es, beim Lesen eine bildliche Vorstellung der Charaktere und Orte zu entwickeln, besonders in rein fiktionalen Erzählungen.

Das erklärt auch den Erfolg von Dungeons & Dragons der Firma TSR Hobbies in den 1970er Jahren, ebenso wie die Beliebtheit von Tolkiens Herr der Ringe. Die Phantasie steht im Zentrum jeder großen Geschichte, und Textadventures bilden keine Ausnahme. Fast jeder Mensch hat irgendwann einmal den Wunsch, selbst eine Geschichte zu schreiben: über etwas, das er oder sie getan, gehört, gesehen oder geträumt hat. Die riesige Zahl an Büchern über die Kunst des Schreibens beweist das.

Ich wurde schon oft (per E-Mail) gefragt, wie Adventures eigentlich geschrieben werden. Wie habe ich beispielsweise „Westfront PC: The Trials of Guilder“ programmiert? Welche Sprache habe ich verwendet? Und ob ich nicht deonstrieren könne, wie man das macht? – Die meisten meiner Adventures sind in BASIC geschrieben. Richtig, in BASIC! Diese alte Programmiersprache, die von den meisten nach den 80er Jahren aufgegeben wurde. Abgesehen von PowerBASIC und einer Handvoll anderer kommerzieller Produkte ist BASIC ziemlich in den Hintergrund getreten. Warum sollte also ein Adventure in BASIC geschrieben werden?

Der Grund heißt Einfachheit. BASIC verwendet Befehle, die der Alltagssprache ähnlich sind und verwendet eine simple Struktur. BASIC verwaltet außerdem Strings und Arrays. Außerdem verwende ich diese Programmiersprache seit 1983, als ich am Commodore 64 zum ersten Mal damit konfrontiert war. Es ist nur zu natürlich, dass ich BASIC verwendete, um Adventures zu schreiben.

Im folgenden Artikel werde ich also erklären, wie Adventures geschrieben werden. In BASIC. Dabei soll Zeile für Zeile erklärt werden, was im BASIC-Programm geschieht. Grundlage ist mein Spiel „Orathan“ für den Commodore 64.

Initialisierung

```
0 POKE 53280,0: POKE 53281,0:HP = 25:LV = 2:EX = 0:G = 100:HM = 25:MM = 15:MP = 15:IC = 5:CT = 102:RM = 1:AS = „:L1$ = „: PRINT CHR$(14)"" CHR$(8): GOTO 50
1 PRINT „NOW SAVING „NAS""...: OPEN 15,8,15: „SO:“ + NAS + „,SW“: OPEN 2,8,2,0:“ + NAS + „,SW“
2 PRINT# 2,HP: PRINT# 2,HM: PRINT# 2,MP: PRINT# 2,MM: PRINT# 2,G: PRINT# 2,RM: PRINT# 2,CT: PRINT# 2,IC: PRINT# 2,DM: PRINT# 2,AC
3 PRINT# 2,EX: PRINT# 2,N: PRINT# 2,V: FOR X = 7 TO 25: PRINT# 2,LO(X): NEXT : CLOSE 2: PRINT „DONE.“: GOTO 107
4 PRINT „NOW LOADING „NAS""...: OPEN 2,8,2,0:“ + NAS + „,SR“
5 INPUT# 2,HP: INPUT# 2,HM: INPUT# 2,MP: INPUT# 2,MM: INPUT# 2,G: INPUT# 2,RM: INPUT# 2,CT: INPUT# 2,IC: INPUT# 2,DM: INPUT# 2,AC
6 INPUT# 2,EX: INPUT# 2,N: INPUT# 2,V: FOR X = 7 TO 25: INPUT# 2,LO(X): NEXT : CLOSE 2: PRINT „DONE.“: FOR X = 1 TO 1900: NEXT : PRINT „“: GOTO 100
7 PRINT „CONTINUE“: PRINT : PRINT „ENTER YOUR CHARACTER NAME“: INPUT NAS: IF LEN (NAS) > 16 THEN PRINT : PRINT „TOO LONG!“: FOR X = 1 TO 1900: NEXT : GOTO 7
8 GOTO 4
```

Zeile 0 setzt den Bildschirm auf Schwarz und weist dem Spieler die drei Werte Hitpoints (HP=25), Level (LV=2) und Experience (EX=2) zu. Der Goldbesitz wird in G (G=100) festgehalten. Das Maximum an Hitpoints wird in HM (HM=25) festgelegt. Magiepunkte werden in den Werten MM (MM=15) und MP (MP=15) gespeichert. Außerdem wird festgelegt, dass der Spieler höchstens fünf Gegenstände tragen kann (IC=5) und dass das Spiel nach 102 „Zügen“ zurückgesetzt wird (CT=102).

Wichtige Hinweise: Der Text wurde aus dem Englischen übersetzt wurde, die Programmpassagen selbst aber im Original belassen. Das hier kommentierte Programm ist 53 Blocks groß, bedarf also einiger Tipparbeit. Wer keine Lust hat, das Programm selbst einzugeben, kann das Spiel – neben vielen anderen Adventures von Paul Panks – hier als d64-Image downloaden: <http://www.geocities.com/dunric/c16.html>

Die bei der Konvertierung des Listings entstandenen Probleme mit einigen Sonderzeichen sowie der Groß-/Kleinschreibung bitten wir zu entschuldigen.

Der Spieler beginnt in Raum #1 (RM=1). Die Zeichen werden auf lowercase geschaltet – PRINT CHR\$(14) –, die Commodore/Shift-Tastenkombination wird außer Kraft gesetzt (CHR\$(8)). Danach springt das Programm in Zeile 50.

Die Zeilen 1 bis 3 speichern die Werte des Spielers auf Diskette, damit sie später wieder aufgerufen werden können. Das Spiel fragt dann: „Are you playing a saved game?“ Die Werte aller wichtigen Variablen, die wir im ersten Absatz besprochen haben, werden damit auf Diskette gesichert. Das schließt auch den Wert von

LO(X) ein, der die Räume angibt, in dem sich die 25 für unser Spiel wichtigen Gegenstände befinden – darüber später mehr.

In den Zeilen 4 bis 6 werden die Werte von Diskette geladen. Damit kann ein vorher gespeicherter Spielstand wieder hergestellt werden und ermöglicht es uns, dort weiter zu spielen, wo wir zuvor aufgehört haben.

Die Zeilen 7 und 8 sind eine Subroutine, die aufgerufen wird, wenn der Spieler auf die Frage „Are you playing a saved game?“ mit Y (Ja) antwortet. In diesem Fall wird der Spieler gebeten, einen (höchstens 16 Buchstaben langen) Namen eines Charakter-Files einzugeben. Hat der Spieler z.B. einen Charakter namens „Charlie“ auf der Diskette gespeichert, würde er hier „Charlie“ eingeben, und der Spielstand würde geladen werden.

Initialisierung der Datenstrukturen

Gehen wir nun ein Stück weiter im Listing:

```
50 PRINT „ORATHAN“: PRINT „(C)COPYRIGHT 2003 BY PAUL PANKS“: PRINT „ALL RIGHTS RESERVED.“: PRINT : PRINT „PLEASE WAIT...“: DIM NOS(25),LO(25),M(45,6),DES(45),VBS(18),EX(25)
51 FOR X = 1 TO 25: READ NOS(X): NEXT
52 FOR X = 1 TO 25: READ LO(X): NEXT
53 FOR X = 1 TO 43: FOR Y = 1 TO 6: READ M(X,Y): NEXT : NEXT
54 FOR X = 1 TO 43: READ DES(X): NEXT
55 FOR X = 1 TO 15: READ VBS(X): NEXT : FOR X = 1 TO 25: READ EX(X): NEXT : GOSUB 800
56 PRINT „ARE YOU PLAYING A SAVED GAME (Y/N)“:
57 GET KS: IF KS = „“ THEN 57
58 IF KS = „Y“ OR KS = „y“ THEN GOTO 7
59 IF KS = „N“ OR KS = „n“ THEN GOTO 75
60 GOTO 57
```

Zeile 50 gibt einen kleinen Begrüßungsbildschirm aus – nicht viel, aber ausreichend. Außerdem werden wichtige Arrays dimensioniert. Diese sind: NOS(25), LO(25), M(45,6), DES(45), VBS(18), und EX(25). Unsere Wörter und Gegenstände sind in NOS gespeichert, ihre Lageplätze finden wir in LO(25).

M(45,6) enthält die Karte unseres Spiels. Es handelt sich dabei um einen zweidimensionalen Array, der pro Raum sechs Richtungen enthält, in die wir uns bewegen können. Um beispielsweise festzulegen, dass Raum 45 nach Norden in Raum 23 führt, würden wir das so ausdrücken: M(45,1) = 23. Das bedeutet, dass man mit der Richtung NORTH in Raum 45 Raum 23 erreicht. Dazu haben wir festgelegt, dass „1“ „Norden“ bedeutet, „2“ „Süden“, „3“ „Osten“, „4“ „Westen“, „5“ „hinauf“ und „6“ „hinunter“. Die erste Zahl „M“ ist IMMER die Nummer des Raums, gefolgt von einer der sechs Richtungen.

DES(45) speichert die Beschreibungen der Räume. Damit wird es ermöglicht, bis zu 45 Raumbeschreibungen zu verwalten, wobei die Zahl der Räume auch erweitert werden kann und nur durch den verfügbaren Speicher begrenzt wird.

VBS(18) enthält unsere Verben, darunter gebräuchliche Wörter wie GO, GET, DROP, WIELD, UNWIELD, WEAR, REMOVE, BUY, SELL, USE, CLIMB, INVENTORY, LOOK, EXAMINE, READ, KILL, SAVE, QUIT, EAT und DRINK. Je mehr Verben verwendet werden, desto komplexer – und länger – wird unser Programm.

Zu guter Letzt dimensioniert Zeile 50 auch noch String Array EX(25). EX enthält die Beschreibungen der Gegenstände und der Monster. In Orathan sind das zum

Beispiel die Gegenstände SWORD, TORCH, LANTERN, ROPE, GOBLIN und andere Gegenstände und Monster (bis zu 25). Auch diese Zahl kann vergrößert werden, wodurch der Speicherplatzbedarf allerdings auch ansteigen würde.

Die Zeilen 51 bis 55 lesen die DATA-Zeilen am Ende des Listings. Die DATA-Zeilen enthalten alle bisher erwähnten Informationen, die in Zeile 50 dimensionierten Felder eingeschlossen. Ich habe versucht, jedes READ-Statement (in einer FOR/NEXT-Schleife) als einzelne Programmzeile zu schreiben, damit das Debuggen aufgrund der besseren Lesbarkeit einfacher wird.

Die Zeilen 56 bis 60 fragen beim Spieler (der Spielerin) nach, ob er oder sie ein gespeichertes Spiel von Diskette laden möchte. Zeile 57 hält das Programm an, bis eine Taste gedrückt wird, bevor die Zeilen 58 und 59 dafür sorgen, dass entweder „Y“ oder „N“ eingegeben wird. Sollte kein akzeptabler Wert eingegeben werden, springt das Programm durch das Kommando in Zeile 60 zurück in Zeile 57, die Eingabe muss wiederholt werden.

Gehen wir nun weiter:

```
75 PRINT : PRINT „ENTER A CHARACTER NAME“: INPUT NAS: IF LEN (NAS) > 16 THEN PRINT : PRINT „TOO LONG!“. FOR X = 1 TO 1900: NEXT : GOTO 56
```

Zeile 75 wird aufgerufen, wenn der Spieler ein neues Spiel starten möchte. Der Name des zu erstellenden Charakters wird in NAS gespeichert. Wenn die Länge von NAS 16 Zeichen (die höchste zulässige Länge) übersteigt, springt das Programm in Zeile 56 (die Yes/No-Abfrage) zurück.

Die Hauptschleife

Dieser Abschnitt ist ziemlich wichtig, er enthält die Hauptschleife des Programms.

```
100 PRINT [CLEARSCREEN] „ORATHAN“ : PRINT DE$(RM)
101 FOR X = 7 TO 25: IF LO(X) = RM THEN PRINT NOS(X) : „
102 NEXT
```

In Zeile 100 wird zunächst der Bildschirm gelöscht, dann wird der Titel „Orathan“ ausgegeben. [Im Original in Groß-/Kleinschreibung und revers!] Danach wird die Beschreibung des aktuellen Raums, die in DE\$(RM) gespeichert ist, ausgegeben.

Zeile 101 eröffnet eine Schleife und überprüft, ob irgendeiner der 25 Gegenstände des Spiels im Raum ist (IF LO(X)=RM THEN PRINT NOS(X) : „). Ist dies der Fall, wird das über das PRINT-Kommando am Bildschirm ausgegeben (PRINT NOS(X) : „).

Zeile 102 erhöht den Wert von X, bis der Wert 25 erreicht wird, danach wird die Schleife abgebrochen.

```
103 IF LO(25) = 998 THEN PRINT : PRINT „CONGRATULATIONS!“: PRINT „YOU HAVE DEFEATED MORDIMAR!“: FOR
X = 1 TO 1900: NEXT : PRINT „GREAT JOB!“: PRINT : PRINT „THE END“: CLR : END
```

Zeile 103 ist sehr wichtig: Hier wird im aktuellen Spielstand ein „Gewinnfaktor“ abgefragt. Da es das Ziel von Orathan ist, ein bestimmtes Monster zu besiegen, ist dieser Wert besonders wichtig. Wird das Monster in einer Schlacht besiegt, wird es zum Ort 998 (LO(25)=998) transferiert. Hat nun der Ort des Monsters tatsächlich den Wert 998, wird ein Text ausgegeben, der dem Spieler kurz zum Sieg gratuliert. Nicht sehr feierlich, aber für unsere Zwecke ausreichend.

```
105 PRINT „OBVIOUS EXITS“: FOR X = 1 TO 6: IF M(RM,X) > 0 THEN PRINT NOS(X) : „
106 NEXT : PRINT
```

Zeile 105 gibt aus, welche Ausgänge sichtbar sind („Obvious Exits:“), bevor eine kurze Schleife gestartet wird. Ist der Wert von M (RM,X) größer als 0, weiß das Programm, dass in dieser Richtung ein Ausgang zu finden ist – näheres dazu haben wir schon weiter oben besprochen.

Trifft dies zu, wird mittels PRINT NOS(X) : „ ausgegeben, in welche Richtung(en) sich unser Charakter bewegen kann.

Der Parser

```
107 CLOSE 2: PRINT : PRINT „>“: OPEN 2,0: INPUT#2,AS: CLOSE 2: PRINT
```

Zeile 107 wird öfter aufgerufen als jede andere Programmzeile. Sie markiert den Beginn der Parserroutine. Ein Parser ist im Prinzip eine Routine, die Sätze in einzelne Wörter zerlegt, bevor diese an andere Routinen zur Verarbeitung von Verben und Substantiven weiter gereicht werden. Der Parser ist eine Art Polizist, der die Verkehrsströme kontrolliert, abgesehen davon, dass er dafür verantwortlich ist, dass das Programm den Anschein erweckt, (in diesem Fall englische) Sätze zu verstehen.

Alles, was der Spieler eingibt, wird zunächst in der Variable AS gespeichert. In einem einfachen Adventure wie Orathan reichen Zwei-Wort-Kommandos (Verb/Substantiv) wie GET SWORD oder GO NORTH völlig aus.

OPEN 2,0 öffnet einen Kanal zum C64, INPUT#2,AS funktioniert wie ein INPUT-Befehl. Der Unterschied: Das lästige Fragezeichen am Beginn der Abfrage wird so eliminiert. CLOSE 2 schließt den Prozess sauber ab, danach wird die Ausgabe am Bildschirm eine Zeile weiter gerückt.

```
108 IF AS = „N“ OR AS = „NORTH“ THEN N = 1: GOTO 130
109 IF AS = „S“ OR AS = „SOUTH“ THEN N = 2: GOTO 130
110 IF AS = „E“ OR AS = „EAST“ THEN N = 3: GOTO 130
111 IF AS = „W“ OR AS = „WEST“ THEN N = 4: GOTO 130
112 IF AS = „U“ OR AS = „UP“ THEN N = 5: GOTO 130
113 IF AS = „D“ OR AS = „DOWN“ THEN N = 6: GOTO 130
```

Die Zeilen 108 bis 113 machen genau das, wonach es aussieht: Wenn die Eingabe AS z.B. „N“ (übliche Abkürzung für „Gehe in Richtung Norden“) lautet, wird dem Substantiv-Flag N die entsprechende Kompass-Richtung 1 zugewiesen (Norden = 1, Süden = 2, Osten = 3 etc.). Das Programm wird dann zur Zeile 130 geschickt.

```
114 IF AS = „L“ OR AS = „LOOK“ THEN GOTO 100
```

Zeile 114 ermöglicht die Eingabe ein „L“ anstelle von „LOOK“. Gibt der Spieler „L“ oder „LOOK“ ein, springt das Spiel zurück in Zeile 100, um die Beschreibung des Raums auszugeben.

```
115 IF AS = „I“ OR AS = „INV“ OR AS = „INVE“ OR AS = „INVEN“ OR AS = „INVENTORY“ THEN GOTO 150
```

Zeile 115 prüft, ob der Befehl „INVENTORY“ (oder auch eine der gängigen Abkürzungen „I“, „INV“, „INVE“, „INVEN“) eingegeben wurde. Ist dies der Fall, wird die Inventory-Routine in Zeile 150 aufgerufen. Diese wird zum gegebenen Zeitpunkt erläutert.

```
116 IF AS = „SAVE“ OR AS = „SAVE GAME“ THEN GOTO 1
```

Hier wird überprüft, ob „SAVE“ oder „SAVE GAME“ eingegeben wurde. Ist das der Fall, wird auf Zeile 1 verwiesen, die die entsprechende Subroutine enthält.

```
117 IF AS = „LIST“ THEN GOTO 850
```

Zeile 177 überprüft, ob das eingegebene Kommando (AS) „LIST“ lautet. Dieses Kommando funktioniert nur, wenn man sich im Village Store befindet. Ist dem so, wird zu Zeile 850 verzweigt.

```
118 IF AS = „SCORE“ OR AS = „?“ OR AS = „SC“ THEN GOTO 860
```

Zeile 188 kontrolliert, ob „SCORE“ (oder eine der gängigen Abkürzungen „?“ und „SC“) eingegeben wurde. Ist das der Fall, wird zur Zeile 860 verzweigt, von der die Werte des Spielers ausgegeben werden.

```
120 FOR X = 1 TO LEN (AS): IF MID$(AS,X,1) = „ „ THEN VS = LEFT$(AS,X - 1):NS = MID$(AS,X + 1,X + 30)
121 NEXT
```

Nun „parsen“ die beiden Zeilen 120 und 121 den eingegebenen Satz in AS in zwei oder mehrere Ausdrücke. Die Routine ist einfach und kann daher leicht in die Irre geführt werden. Wenn die Reihenfolge Verb/Substantiv eingehalten wird, funktioniert sie aber einwandfrei.

Zeile 120 untersucht die Länge von AS (LEN(AS)), danach wird geprüft, ob sich ein Leerzeichen im eingegebenen Satz befindet (IF MID\$(AS,X,1) = „ “). Ist das der Fall, wird VS (das VERB) den Zeichen VOR dem Leerzeichen zugewiesen (VS=LEFT\$(AS,X-1)), der Bereich NACH dem Leerzeichen wird NS (NOUN, engl. für SUBSTANTIV) zugewiesen (NS = MID\$(AS,X + 1,X + 30)). Obwohl das nicht die effizienteste Weise ist, dieses Problem zu lösen, ist sie bei einem Spiel wie diesem gut genug. Der fortgeschrittene Parser, den ich für andere Spiele geschrieben habe, würde in diesem Artikel zu weit führen.

```
125 V = 0: FOR X = 1 TO 15: IF VS = VB$(X) THEN V = X
```

```
126 NEXT : IF V = 0 THEN PRINT „HUH? CHECK YOUR VERB.“: GOTO 107
```

Diese beiden Zeilen prüfen, ob ein eingegebenes Verb in den DATA-Zeilen am Ende des Programms vorkommt – in diesem Fall im String Array (VB\$(15)). Passt eines der 15 Verben mit dem Wort in VS zusammen, wird das Flag der Variable auf X (V=X) gesetzt, die Schleife wird unterbrochen. Andernfalls fährt das Programm mit Zeile 126 fort und fährt mit der Prüfung fort, bis alle 15 Verben an der Reihe waren. Wurde dabei nichts gefunden, wird V auf 0 gesetzt, das Programm gibt „HUH? CHECK YOUR VERB.“ auf dem Bildschirm aus. Danach springt das Programm zur Hauptschleife (Zeile 107) zurück.

Es ist sehr wichtig, dass am Beginn der Zeile 125 das Flag der Variable auf 0 gesetzt wird (V=0). Damit wird sichergestellt, dass alte Eingaben des Spielers nicht mitverarbeitet werden.

```
127 N = 0: FOR X = 1 TO 25: IF NS = NOS(X) THEN N = X
```

```
128 NEXT : IF N = 0 AND V <> 10 THEN PRINT „HUH? CHECK YOUR NOUN.“: GOTO 107
```

Die Zeilen 127 und 128 ähneln den vorhergehenden Programmzeilen. Zeile 127 setzt das Substantiv-Flag auf 0 (N=0), danach wird eine ähnliche Schleife wie im oberen Abschnitt durchgeführt. Es gibt allerdings einen Unterschied: Wenn N=0 und das Verb-Flag NICHT den Wert 10 hat, gibt das Spiel die Meldung „HUH? CHECK YOUR NOUN.“ aus. Warum das wichtig ist? Nun, wenn der Spieler z.B. EXAMINE TAVERN eintippt, ist das Wort TAVERN zwar in den Raumbeschreibungen vorhanden, aber NICHT in der Tabelle der Gegenstände und Monster. Deshalb wird das Programm bei V=10 zu Zeile 129 weitergehen, um das EXAMINE-Kommando auch dann abzuarbeiten, wenn kein gültiges Verb vorliegt (N=0).

Die Post-Parser Routine

```
129 ON V GOTO 130,140,160,170,180,190,200,207,210,220,230,239, 250,270,280
```

Zeile 129 ist wie der in einem früheren Absatz erwähnte Verkehrspolizist. Sie lenkt den Programmfluss zur notwendigen VERB-Subroutine, abhängig davon, was der Spieler eingegeben hat.

Es gibt natürlich mehrere Möglichkeiten, dieses Problem zu bewältigen, aber „ON V GOTO“ ist die effizienteste Methode, da sie nur eine einzige Programmzeile benötigt.

Das Verb GO

Die Zeilen 130-135 erklären das Verb GO (gehen) im Detail. „GO“ bewegt den Spieler zwischen den „Räumen“ hin und her. Die GO-Subroutine ist die einfachste

Verben-Routine, aber bestimmte Aspekte müssen trotzdem erklärt werden:

```
130 IF RM = 11 AND N = 2 THEN IF LO(8) < > 0 AND LO(9) < > 0 THEN PRINT „IT'S MUCH TOO DARK!“. PRINT „YOU NEED A SOURCE OF LIGHT.“. GOTO 107
```

Zeile 130 prüft, ob der Spieler in Raum 11 (RM=11) ist, und ob er versucht, nach Süden zu gehen (N=2). Ist das der Fall, wird auch überprüft, ob dem Spieler die beiden Gegenstände TORCH (Fackel) und LANTERN (Laterne) fehlen (LO(8)<>0 AND LO(9)<>0). In meinen Programmen ist die Stelle 0 immer dazu da, um anzuzeigen, dass der Spieler über einen Gegenstand verfügt. Wenn LO(8) also gleich 0 wäre, würde der Spieler die Fackel tragen. Wäre LO(9) gleich 0, hätte der Spieler die Laterne bei sich.

Da sowohl die Laterne als auch Brennstoff (FUEL) benötigt werden, um von Raum 11 aus nach Süden zu gehen, gibt das Spiel die Meldung aus: „IT'S MUCH TOO DARK! YOU NEED A SOURCE OF LIGHT.“ Danach springt das Programm zu Zeile 107 (Hauptschleife) zurück.

```
131 IF LO(19) = 41 AND RM = 41 AND N = 1 THEN PRINT „YOU ARE STOPPED BY „NOS(19)“.“. PRINT „IT HISSES, YOU MAY NOT PASS!“. GOTO 107
```

```
132 IF LO(24) = 39 AND RM = 39 AND N = 1 THEN PRINT „YOU ARE STOPPED BY „NOS(24)“.“. PRINT „HE GROWLS, NO PASSAGE HERE!“. GOTO 107
```

Die Zeilen 131 und 132 sind interessant, da sie verhindern, dass der Spieler in bestimmten Räumen nach Norden geht (in diesem Fall 39 und 41), wenn bestimmte Zustände eintreten. Hier ist es der Troll, der, falls er sich in Raum 41 (LO(19)=41) befindet, zur Warnung führt: „YOU ARE STOPPED BY TROLL. IT HISSES, YOU MAY NOT PASS!“.

Zeile 132 macht das gleiche, nur dass sie die Präsenz des BARBARIAN überprüft. Ist dieser anwesend, wird der Versuch, nach Norden zu gehen, mit der Meldung „YOU ARE STOPPED BY BARBARIAN. IT GROWLS, NO PASSAGE HERE!“ quitiert.

```
133 IF RM = 8 AND N = 3 THEN PRINT „THE DOOR (TO THE EAST) IS LOCKED!“. GOTO 107
```

Zeile 133 prüft, ob sich der Spieler in Raum 8 befindet (IF RM=8) und nach Osten gehen möchte (N=3). Ist das der Fall, wird angezeigt, dass die Türe versperrt ist. Das Programm springt zurück in Zeile 107.

```
134 IF M(RM,N) > 0 THEN RM = M(RM,N): GOTO 100
```

Hier die Zeile, auf die wir gewartet haben: Wenn die oben genannten Voraussetzungen nicht zutreffen, überprüft die Programmzeile 134, ob unser Held in eine der sechs Richtungen gehen kann. Wenn ja, wird der aktuelle Raum zum Zielraum erklärt (RM=M(RM,N)). Das mag verwirrend klingen, aber hier wird nichts anderes getan als BASIC mitzuteilen, dass der in M(RM,N) gespeicherte Wert auch in der Variable RM gespeichert werden soll. R kann keinen Wert größer als 6 annehmen. Die Datenstrukturen in den Zeilen 1301 bis 1343 sind bewusst so angelegt, dass sich der Spieler an keine Stelle bewegen kann, die das Spiel nicht „kennt“.

```
135 PRINT „YOU CAN'T GO THAT WAY!“. GOTO 107
```

Diese Zeile wird aufgerufen, wenn sich der Spieler in keine der sechs Richtungen bewegen kann oder wenn die gewählte Richtung (N) den Wert 0 hat. Eine einfache Fehlermeldung wird ausgegeben, das Programm springt zurück zur Hauptschleife.

Das Verb GET

```
140 IF LO(N) = RM AND N < 18 AND N > 6 THEN LO(N) = 0: IC = IC + 1: PRINT „TAKEN.“. GOTO 107
```

```
141 PRINT „YOU CAN'T DO THAT HERE.“. GOTO 107
```

GET ist eine sehr einfache Routine, die mit lediglich zwei Programmzeilen auskommt. Zeile 140 prüft, ob der Platz des Gegenstandes mit dem aktuellen Raum übereinstimmt (IF LO(N)=RM) und ob der Wert der Variable des Gegenstandes (N) höher als 6 und nicht höher als 17 ist (N<18 AND N>6).

Wenn N nicht höher als 17 ist, handelt es sich um einen Gegenstand wie LANTERN, FUEL, SWORD usw., nicht um ein Monster – diese sind in den Werten über 17 gespeichert. Wenn N größer als 6 ist, handelt es sich nicht um die Richtungen wie NORTH, SOUTH, EAST usw.

Wenn beide Bedingungen erfüllt sind, wird der Wert, der in N gespeichert ist, auf 0 (Inventar) gesetzt. Der Gegenstandszähler wird einfach um 1 erhöht, (IC=IC+1), dazu wird eine einfache Botschaft ausgegeben, dass der Gegenstand aufgenommen wurde. Danach springt das Programm zurück in die Hauptschleife (Zeile 107).

Wird die Bedingung von Zeile 140 jedoch nicht erfüllt, gibt das Spiel die Meldung „YOU CAN'T DO THAT HERE.“ aus. Danach kehrt das Programm zur Zeile 107 zurück.

Das Inventar

Das Inventar, aufgerufen durch das INVENTORY-Kommando (siehe Zeile 115), erlaubt es dem Spieler, sein Gepäck zu überprüfen. Dieses „Verb“ ist in einem Textadventure sehr wichtig, ohne es könnte man kaum den Überblick bewahren.

```
150 PRINT „YOU ARE CARRYING.“: AC = 0: DM = 1: SI = 0: IC = 0: FOR X = 7 TO 25: IF LO(X) = 0 THEN PRINT NOS(X): IC = IC + 1: SI = 1
```

Zeile 150 gibt „YOU ARE CARRYING.“ aus, bevor bestimmte Variablen zurückgesetzt werden. „AC=0“ bedeutet, dass die Rüstungsklasse vorher zurückgesetzt wird – ebenso DM=1 („Player Potential Damage Inflicted Flag“) und die getragenen Gegenstände (IC=0) und das spezielle Flag SI (SI=0). Die Zeile durchläuft eine kurze Schleife (FOR X=7 TO 25), die mit dem Wert 7 beginnt, um die Richtungen (North, South...) zu überspringen. Wenn der Spieler einen Gegenstand trägt, (IF LO(X)=0)),

wird dieser am Bildschirm angezeigt (PRINT NOS(X)), und das „Items carried“-Flag wird um 1 erhöht (IC=IC+1). Das spezielle Flag SI wird dann auf 1 gesetzt. Warum das? Ich komme gleich darauf zu sprechen.

```
151 IF LO(X) = 105 THEN PRINT NOS(X) (WIELDED): IC = IC + 1: SI = 1: DM = N: WD = 1
```

Zeile 151 prüft, ob getragene Waffen vorhanden sind. Ich speichere diesen Wert in 105, also weit über der Grenze für Räume (45) in diesem Adventure. Du kannst jede Zahl verwenden, die leicht zu merken ist, z.B. 1000 oder 999. Gib aber Acht, dass es keine Raumnummer oder Inventar-Nummer ist, sonst wird die Waffe an zwei Plätzen gleichzeitig auftauchen.

Nun wird geprüft, ob der Spieler etwas trägt (im Sinne von „angelegt hat“). In diesem Fall wird der Gegenstand angezeigt, gefolgt von einem „ (WIELDED)“. Der Wert erhöht sich um 1, (IC=IC+1), und das „Player Potential Damage Inflicted Flag“ in DM nimmt den Wert des aktuellen Gegenstandes, der in N gespeichert ist, an (DM=N). Unser Spezialflag in SI wird auch auf 1 gesetzt (SI=1). Außerdem wird ein einfaches Flag in WD gesetzt, um dem Programm mitzuteilen, dass ein Gegenstand angelegt wurde (WD=1).

```
152 IF LO(X) = 205 THEN PRINT NOS(X) (WORN): IC = IC + 1: SI = 1: AC = AC + N
```

Zeile 152 überprüft, ob eine Rüstung getragen wird. Diesen Wert speichere ich in 205, also wieder weit über den Raumnummern. Da die Zahl um 100 höher ist als jene in Zeile 151, merke ich sie mir leicht. Du kannst aber auch z.B. 2000 oder 1999 nehmen, falls dir das leichter fällt. Wichtig ist nur, keine Raum- oder Inventarnummer zu verwenden.

```
153 NEXT X: IF SI = 0 THEN PRINT „ALAS, YOU ARE EMPTY-HANDED.“. GOTO 107
```

Diese Programmzeile erhöht die Durchläufe der Schleife, die in Zeile 150 gestartet wurde, um 1. Wenn X den Wert 25 erreicht hat, wird die Schleife beendet, und die Zeile wird weiter interpretiert. In diesem Fall wird unser Spezialflag in SI ausgelesen, wobei überprüft wird, ob der Wert 0 beträgt. Erinnerst du dich? Dieses Flag wird auf 1 gesetzt, falls der Spieler irgendetwas trägt. Dabei kann es sich um Gegenstände, Waffen oder Rüstung handeln. Wenn das nicht der Fall ist (IF SI=0), hat der Spieler offensichtlich gar nichts bei sich, weshalb das Spiel ausgibt: „ALAS, YOU ARE EMPTY-HANDED.“. Danach springt das Programm in Zeile 107 zurück.

```
154 GOTO 107
```

Hat der Spieler einen Gegenstand (eine Waffe, eine Rüstung) bei sich, also bei SI=1, springt das Programm direkt zu Zeile 154, die das Programm zur Hauptschleife zurück springen lässt.

Das Verb DROP

DROP (ablegen) ist einfach eine Umkehrung von GET oder TAKE.

```
160 IF LO(N) = 0 THEN LO(N) = RM: IC = IC - 1: PRINT „DROPPED.“. GOTO 107
```

```
161 PRINT „YOU MUST BE CARRYING THAT TO DROP.“. GOTO 107
```

Zeile 160 prüft, ob der Raum von N gleich 0 ist (IF LO(N)=0). Ist unser aktueller Gegenstand im Inventar des Spielers, wird er in den Raum verschoben, in dem sich der Spieler gegenwärtig befindet (LO(N)=RM). Anders gesagt: Der Spieler „wirft“ das Objekt zu Boden. Das „Getragener Gegenstand“-Flag wird um 1 erhöht (IC=IC-1), und das Spiel gibt eine Botschaft aus, z.B. „DROPPED“. Danach geht es in Zeile 107 weiter.

Das Verb WIELD

Das Verb WIELD führt immer wieder zu Verwirrung, und das mit vollem Recht. Was macht „wield“ (anlegen) genau, und wäre „equip“ (ausrüsten) eine bessere Alternative? In englischsprachigen Dungeon-Spielen ist wield einfach das gebräuchlichere Wort, deshalb bin ich dabei geblieben.

```
170 IF LO(N) = 0 THEN IF N = 7 THEN LO(7) = 105: DM = N: PRINT „SWORD IS NOW WIELDED.“. GOTO 107
```

```
171 IF LO(N) = 0 THEN IF N = 15 THEN LO(15) = 105: DM = N: PRINT „BROADSWORD IS NOW WIELDED.“. GOTO 107
```

```
172 PRINT „YOU CAN'T WIELD THAT!“. GOTO 107
```

Die Programmzeilen 170 und 171 überprüfen, ob der Spieler das anzulegende Objekt bei sich hat (IF LO(N)=0)). Ist das der Fall, wird überprüft, ob es entweder das SWORD (N=7) oder das BROADSWORD (N=15) ist. Trifft eine der beiden Bedingungen zu, setzen die beiden Zeilen den Ort des Objekts auf 105, jenen Ort, den ich für eine „getragene Waffe“ bestimmt habe.

Das „Player Damage Potential Flag“, gespeichert in DM, wird danach so gesetzt, dass es mit dem aktuellen Gegenstand in N übereinstimmt. In den beiden Fällen entspricht N also entweder dem SWORD (N=7) oder dem BROADSWORD (N=15). Sind beide Bedingungen unzutreffend, springt das Programm in Zeile 172, wo einfach ein Fehler ausgegeben wird, bevor es zur Zeile 107 zurückspringt.

Das WIELD-Kommando könnte auch mit einer FOR/NEXT-Schleife gelöst werden, das würde in diesem Artikel, der für Einsteiger gedacht ist, jedoch zu weit führen.

Das Verb UNWIELD

Das Verb UNWIELD ist das Gegenteil von WIELD. Hat der Spieler eine Waffe angelegt, und diese Subroutine wird aufgerufen, überprüft das Programm die Zeilen 180 und 181, ob sie überhaupt abgelegt werden kann und was damit gemacht werden kann.

```
180 IF LO(N) = 105 THEN LO(N) = 0: PRINT „OK.“: DM = 1: GOTO 107
```

Neues bei Protovision

JTR, MacGyver und Thunderblade von Protovision (für mehr Informationen über unsere Members siehe „About“-Rubrik unserer Homepage) waren auf der Classic Gaming Expo (CGE UK) in Croydon/UK. Thunderblade hat jede Menge Fotos gemacht, die ihr euch hier anschauen könnt: http://www.protovision-online.de/london_cge_fotos/. Für mehr Fotos von diesem Event folgt diesem Link: <http://www.cgeuk.com/gallery/2004>

Protovision im Maniac Magazin

Protovision wurde im größten deutschen Konsolen-Magazin MANIAC (<http://www.maniac.de>) erwähnt, Ausgabe 10/2004.

Protovision T-Shirts

Von nun an sind hochwertige Protovision T-Shirts in verschiedenen Größen in unserem Online-Shop erhältlich: <http://www.protovision-online.de/catalog>

Retro Gamer Interview

Wir hatten kürzlich eine Plauderei mit der Retro Gamer, dem schnell wachsenden Retro-Magazin aus Großbritannien. Das ausführliche Interview wird in Ausgabe 5 veröffentlicht, die in wenigen Tagen erscheint. Besorg' sie dir! <http://www.retrogamer.net>

>> www.protovision-online.de

181 PRINT „YOU CAN'T UNWIELD THAT!“. GOTO 107

Nun prüft Zeile 180, ob der Spieler eine Waffe angelegt hat (IF LO(N)=105). Ist dies der Fall und ist gleichzeitig der Ort des aktuellen Gegenstands, der in „N“ gespeichert ist, gleich 105 (der Ort der getragenen Gegenstände), wird dem Ort der Wert 0 zugewiesen (das Inventar des Spielers). Das „Player Potential Damage Flag“ in DM wird auf 1 gesetzt, was bedeutet, dass der Spieler mit nichts als seinen bloßen Händen „bewaffnet“ ist, wodurch die Kampfkraft nur minimal ist (DM=1). Zeile 181 verweist dann zur Zeile 107.

Das Verb WEAR

Dieses Kommando ist einfach. Der Spieler kann nur bestimmte Gegenstände aufnehmen, und das nur, wenn er sie nicht bereits trägt.

190 IF LO(N) = 0 THEN IF N > 10 AND N < 18 AND N < > 12 AND N < > 15 THEN LO(N) = 205: PRINT „YOU WEAR „NOS(N)“.“: AC = AC + N: GOTO 107

Zeile 190 prüft, ob der Spieler den aufzunehmenden Gegenstand trägt (IF LO(N)=0), und wenn das so ist, wird weiters geprüft, ob die Nummer des aktuellen Gegenstandes in N zwischen 10 und 18 liegt, außerdem, ob sie nicht gleich 12 oder 15 ist. Treffen alle diese Bedingungen zu, wird der Gegenstand als anlegbare Rüstung betrachtet und LO(N)=205 bewegt den Gegenstand in die „WEAR“-Sektion – in meinen Spielen gewöhnlich 205. Wie bei WIELD sollte auch bei WEAR darauf geachtet werden, dass sie über den möglichen Orten („Räumen“) im Spiel liegt.

Zeile 190 erhöht schließlich die Rüstungsklasse des Spielers (in AC gespeichert) um die Zahl des Gegenstandes in N (AC=AC+N). Danach wird am Bildschirm ausgegeben, dass der Spieler nun den Gegenstand trägt. Danach springt das Programm zurück zur Hauptschleife.

191 PRINT „YOU CAN'T WEAR THAT!“. GOTO 107

Treffen die Bedingungen in Zeile 190 nicht zu, wird einfach eine Fehlermeldung ausgegeben, die den Spieler informiert, dass er den angegebenen Gegenstand nicht aufnehmen kann. Danach springt das Programm zurück zur Zeile 107.

Das Verb REMOVE

REMOVE ist das Gegenteil von WEAR, und damit ein sehr wichtiges Verb. Sobald der Spieler eine neue und wertvollere Rüstung erhält, ist es notwendig, die schwächere Rüstung abzulegen und die neue anzulegen.

200 IF LO(N) = 205 THEN LO(N) = 0: AC = AC - N: PRINT „YOU REMOVE „NOS(N)“.“: GOTO 107

201 PRINT „YOU CAN'T REMOVE THAT!“. GOTO 107

Zeile 200 prüft, ob der aktuelle Wert des Gegenstandes (gespeichert in N) gleich 205 ist (IF LO(N)=205), wobei 205 der Ort von WEAR im Programm ist. Ist dies der Fall, handelt es sich um eine Rüstung, sie wird als getragen betrachtet und ins Inventar zurück gelegt (LO(N)=0). Das Rüstungsklassen-Flag (in AC gespeichert) wird nun um jenen Wert verringert, der in N (die aktuelle Rüstung) gespeichert ist (AC=AC-

N). Es wird eine Mitteilung ausgegeben, die den Spieler informiert, dass die Rüstung gerade abgelegt wurde. Nun springt das Programm wieder in Zeile 107.

Wird die Bedingung in Zeile 200 nicht erfüllt, wird Zeile 201 aufgerufen, die den Spieler informiert, dass er den gewünschten Gegenstand nicht ablegen kann. Danach geht es zurück zur Zeile 107.

Das Verb LIGHT

202 IF LO(N) = 0 THEN IF N = 7 OR N = 8 THEN IF LT = 0 THEN LT = 1: PRINT „IT IS NOW ABLAZE.“: GOTO 107

203 PRINT „YOU CAN'T DO THAT.“: GOTO 107

207 IF LO(N) = 0 THEN IF N = 8 OR N = 9 THEN IF LT = 0 THEN LT = 1: PRINT „IT IS NOW ABLAZE.“: GOTO 107

208 PRINT „LIGHT WHAT?“. GOTO 107

Das Verb EXTINGUISH

210 IF LT = 0 THEN PRINT „EXTINGUISH WHAT?“. GOTO 107

211 IF LO(N) = 0 THEN IF N = 8 OR N = 9 THEN LT = 0: PRINT „UK.“: GOTO 107

212 PRINT „YOU CAN'T EXTINGUISH THAT!“. GOTO 107

Das Verb CLIMB

220 IF RM = 26 THEN IF LO(10) = 0 THEN RM = 27: PRINT „YOU CLIMB UP...“: FOR X = 1 TO 1900: NEXT : GOTO 100

221 PRINT „YOU CAN'T CLIMB THAT, SILLY!“. GOTO 107

Das Verb USE

230 IF LO(N) = 0 THEN IF N = 8 OR N = 9 THEN IF LT = 0 THEN LT = 1: PRINT „IT IS NOW ABLAZE.“: GOTO 107

231 IF LO(N) = 0 THEN IF N = 10 THEN IF RM = 26 THEN RM = 27: PRINT „YOU CLIMB UP...“: FOR X = 1 TO 1900: NEXT : GOTO 100

232 IF LO(N) = 0 THEN IF N = 12 THEN IF RM = 8 THEN RM = 9: PRINT „CLICK!“. FOR X = 1 TO 1900: NEXT : GOTO 100

233 PRINT „USE WHAT?“. GOTO 107

Das Verb KILL und die FIGHT-Routine, Teil 1

Das Verb KILL ist ziemlich kompliziert und bedarf einer genauen Erklärung. Es handelt sich um keine simple Subroutine, eine Menge Betatesters waren nötig, um Bugs zu beseitigen. Das Verb wird in zwei Teilen behandelt. Die Zeilen 239 bis 246 enthalten den ersten Teil, 400 bis 499 den zweiten.

Nun werden wir uns das Wort Zeile für Zeile vornehmen:

239 MH = INT (RND (1) * 250) + 1: IF N = 25 THEN MH = 1200

Programmzeile 239 setzt die Variable MH (die Hit Points des Monsters) auf einen Zufallswert zwischen 1 und 250. Danach wird geprüft, ob die aktuelle Gegenstandszahl in N gleich 25 ist (IF N=25). Ist dies der Fall, wird der Wert MH auf 1200 gesetzt (MH=1200).

240 I = INT (RND (1) * 25) + 1: IF N < 18 OR LO(N) < > RM THEN PRINT „YOU MUST BE JOKING!“. GOTO 107

Zeile 240 setzt die Variable I auf einen Zufallswert zwischen 1 und 25. Danach wird geprüft, ob die Zahl des Gegenstandes in N entweder kleiner als 18 (N<18) oder nicht im aktuellen Raum vorhanden ist (LO(N)<>RM). Treffen beide Bedingungen zu, hat der Spieler versucht, irgendetwas anderes als ein Monster anzugreifen – zum Beispiel hätte er im Zorn gegenüber seiner Laterne handgreiflich werden können. Das Programm springt zurück in Zeile 107.

241 IF I < = 5 THEN PRINT „YOU MISSED.“

Zeile 241 prüft, ob der Wert von I, der oben definiert wurde, kleiner als 5 ist (I<=5). Ist das so, hat der Spieler sein Ziel, ein Monster wie DRAGON, GOBLIN etc. verfehlt.

242 DM = 1: FOR X = 7 TO 25: IF LO(X) = 105 THEN DM = X

Zeile 242 setzt das „Player Potential Damage Flag“ in DM auf 1 (DM=1). Außerdem wird eine Schleife gestartet, die mit X=7 beginnt und bis 25 läuft (FOR X=7 TO 25). Hier wird geprüft, ob der Gegenstand (LO(X)) eine getragene Waffe ist (IF LO(X)=105). Ist dies der Fall, wird der Wert DM auf den Wert der getragenen Waffe gesetzt (DM=X). Die Schleife springt dann in Zeile 243 und vergleicht weitere Gegenstände, aber da der Spieler immer nur eine Waffe tragen kann, ändert sich dadurch der Wert von DM nicht.

243 NEXT : IF I > = 6 AND I < = 10 THEN PRINT „YOU HIT „NOS(N)“ HARD!“. MH = MH - DM

Zuerst wird der Zähler der Schleife aus Zeile 242 um 1 erhöht. Sobald X das Maximum (25) erreicht hat, fährt Zeile 243 fort, indem der Wert in I verarbeitet wird. Ist I eine Zahl von 6 bis 10, signalisiert das Programm dem Spieler mit einer Meldung, dass er das Monster schwer getroffen hat. Die Hit Points des Monsters werden entsprechend verringert (MH=MH-DM).

244 IF I > = 11 AND I < = 18 THEN PRINT „YOU SMASHED „NOS(N)“: PRINT „WITH A BONE-CRUSHING SOUND!“. MH = MH - DM * 2

Zeile 244 funktioniert so wie Zeile 243, prüft aber, ob I eine Zahl von 11 bis 18 ist. Ist das der Fall, gibt das Programm die Meldung „YOU SMASHED (dieses oder jenes Monster) WITH A BONE-CRUSHING SOUND!“ aus und die Hit Points des Monsters werden noch weiter verringert (MH=MH-DM*2).

Hinweis: Wenn du deine eigene Kampfroutine schreibst, denk daran, dass du den Schaden, den ein Spieler (oder Monster) anrichten kann, Zeile für Zeile berechnest, sonst kann es richtig verwirrend werden. Auf diese Weise folgen „zärtlichen Berührungen“ und „harten Schlägen“ „zerschmetternde Treffer“ usw.

```
245 IF I >= 19 THEN PRINT „THE „NOS(N)“ DODGED“: PRINT „YOUR ATTACK.“
```

Zeile 245 prüft, ob I größer oder gleich 19 ist (IF I>=19). Ist das der Fall, hat das Monster die Attacke des Spielers erfolgreich abgewehrt. Der Spieler richtet also „0 Schaden“ an.

```
246 GOTO 400
```

Nun springt das Programm in Zeile 400, wo der zweite Teil der Kampfroutine beginnt. Dazu später!

Das Verb BUY

BUY ist nicht in jedem Spiel notwendig. Verwende es nur, wenn du ein Dorf oder eine Stadt einbauen willst, wo Spieler Gegenstände (Waffen, Rüstungen) einkaufen können. Die Kauf-Routine steckt in den Zeilen 250 bis 265:

```
250 IF RM <> 3 THEN PRINT „NO SHOP HERE!“: GOTO 107
```

```
251 IF LO(18) <> 3 THEN PRINT „NO CLERK HERE.“: GOTO 107
```

```
252 P = 0: GOSUB 500
```

```
263 IF P = 0 OR LO(N) <> 99 THEN PRINT „THE CLERK FROWNS.“: PRINT „HE SAYS, ‘OE DON’T SELL THAT HERE.’“: GOTO 107
```

```
264 IF G >= P THEN LO(N) = 0: G = G - P: IC = IC + 1: PRINT „YOU HAND THE CLERK “P“GOLD COINS.“: PRINT „THE CLERK HANDS YOU“: PRINT „THE „NOS(N)“ “: PRINT „HE SAYS, ‘THANK YOU.’“: GOTO 107
```

```
265 PRINT „THE CLERK SAYS, ‘YOU CAN’T AFFORD THAT.’“: GOTO 107
```

Zeile 250 prüft, ob der Spieler tatsächlich im Shop ist (IF RM<>3). Wenn nicht, erklärt das Programm: „NO SHOP HERE!“ und springt zurück in Zeile 107.

Zeile 251 prüft, ob ein Verkäufer anwesend ist (IF LO(18)<>3). Ist kein Verkäufer hier, teilt uns das Programm das mit und springt in Zeile 107.

Zeile 252 setzt das Preis-Flag in P auf 0 (P=0) und übergibt die Kontrolle an Zeile 500, in der sich die Subroutine für den Preis eines Gegenstandes befindet.

Zeile 263 prüft, ob P gleich 0 ist (IF P=0), oder ob der Ort von N (dem aktuellen Gegenstand) das Geschäftslokal ist (LO(N)<>99). Befindet sich der Gegenstand nicht im Shop, und der Verkäufer teilt mit, dass dieser Gegenstand nicht in seinem Angebot sei.

Zeile 264 prüft, ob der Preis des zu kaufenden Gegenstandes kleiner oder gleich dem Wert des vom Spieler getragenen Goldes ist. Trifft das zu, kann der Spieler den Gegenstand kaufen. Das wird auch am Bildschirm ausgegeben, das Programm springt zurück in Zeile 107.

Zeile 265 wird aufgerufen, wenn der Spieler nicht genug Gold bei sich hat.

Das Verb SELL

SELL ist das genaue Gegenteil von BUY. Es prüft, ob der Spieler sich im Shop befindet und ob er den zu verkaufenden Gegenstand bei sich hat. Die Goldmenge wird entsprechend erhöht und der Gegenstand im Inventar des Geschäfts der angebotenen Gegenstände abgelegt.

```
270 IF RM <> 3 THEN PRINT „YOU ARE NOT IN THE SHOP!“: GOTO 107
```

```
271 IF LO(18) <> 3 THEN PRINT „THE CLERK ISN’T HERE.“: GOTO 107
```

Zeile 271 prüft die Anwesenheit des Verkäufers. Ist er nicht da, wird dies vom Programm gemeldet, das Spiel springt zur Hauptschleife.

```
272 P = 0: GOSUB 500: IF P = 0 OR LO(N) <> 0 THEN PRINT „THE CLERK SAYS, ‘YOU AREN’T’: PRINT „CARRYING THAT“: GOTO 107
```

Zeile 272 setzt das Preis-Flag in P auf 0 (P=0). Danach wird dieselbe Routine (in Zeile 500) aufgerufen wie beim Kauf eines Gegenstandes. Wenn P gleich 0 ist (IF P=0) oder wenn der Ort des Gegenstandes (N) nicht 0 ist (LO(N)<>0), wird eine Fehlermeldung ausgegeben, der Verkauf wird nicht getätigt. Mit anderen Worten: Wenn der Spieler den Gegenstand nicht besitzt oder der Gegenstand keinen Wert (0) hat, kann der Verkauf nicht fortgesetzt werden.

```
273 G = G + P / 2: LO(N) = 99: PRINT „YOU HAND THE CLERK“: PRINT „THE „NOS(N)“ “: PRINT „THE CLERK HANDS YOU“ P / 2: PRINT „GOLD COINS.“: PRINT „HE SAYS, ‘THANK YOU.’“: GOTO 107
```

Zeile 273 erhöht den Goldvorrat des Spielers um die Hälfte des Wertes, um den er ursprünglich gekauft wurde. Das ist in Online-Spielen dieser Art üblich, um den Handel realistischer zu gestalten.

Der Ort des verkauften Gegenstandes in N wird auf 99 gesetzt (LO(N)=99), das ist der Ort des Inventars des Shops.

Das Verb EXAMINE

Das Verb EXMINE kann sehr unterschiedlich gehandhabt werden, je nachdem, wie du dein Adventure schreiben möchtest. Bei mir kann der Spieler mit diesem Wort sowohl Orte als auch Gegenstände untersuchen, egal, ob diese getragen werden oder auf dem Boden liegen.

```
280 IF LO(N) = RM OR LO(N) = 0 OR LO(N) = 105 OR LO(N) = 205 THEN PRINT EX$(N): GOTO 107
```

```
281 IF N = 0 THEN PRINT „YOU NOTICE NOTHING UNUSUAL“: PRINT „ABOUT THE „N$“ “: GOTO 107
```

```
282 PRINT „THAT ISN’T HERE.“: GOTO 107
```

```
399 END
```

Zeile 280 prüft, ob der Gegenstand im aktuellen Raum vorhanden ist (IF LO(N)=RM). Außerdem wird überprüft, ob der Gegenstand mitgeführt wird (LO(N)=0), angelegt ist (LO(N)=105), oder getragen wird (LO(N)=205). Trifft eine dieser Bedingungen zu, gibt das Programm eine Beschreibung des zu prüfenden Gegenstandes aus (PRINT EX\$(N)). Danach springt das Spiel in Zeile 107.

Zeile 281 überprüft, ob N gleich 0 ist. Erinnerst du dich an Zeile 128, in der das Programm N=0 AND V<>10 vergleicht? V=10 bedeutet, dass das EXAMINE-Kommando eingegeben wurde, und wenn N=0, bedeutet das, dass der Spieler zum Beispiel die Orte TAVERN oder WELL untersuchen will. Diese werden bei den Orten, nicht bei den Gegenständen wie LANTERN oder SWORD, beschrieben. Also wird eine einfache Meldung ausgegeben, das Programm springt in Zeile 107 zurück.

Falls nötig, können Zeilen wie 281 erweitert werden:

```
281 IF N=0 THEN IF A$="EXAMINE WELL" AND RM=3 THEN PRINT "THE WELL EXTENDS DOWN SOME 300 FEET."
```

```
THERE IS MOSS GROWING ON THE INSIDE WALLS.“:GOTO 107
```

Auf diese Weise kannst du ein größeres Vokabular vortäuschen und dem Spiel gleichzeitig mehr Tiefe geben. (Die Zeile 399 beendet das Programm, wird aber nie aufgerufen. Der Befehl ist eigentlich unnötig.)

Das Verb KILL und die FIGHT-Routine, Teil 2

Das Wort KILL wurde teilweise schon in den Zeilen 239 bis 246 behandelt, hier wird die Routine fortgesetzt. Ich werde wieder Zeile für Zeile erklären, was hier vorgeht. Die Routine kann komplexer gemacht werden, indem spezielle Zaubersprüche, Bewegungen, Treffer etc. hinzugefügt werden.

```
400 PRINT „>“: FOR X = 1 TO 1900: NEXT : PRINT „YOUR HP:“HP/“HM: PRINT „THEIRS “:MH
```

Zeile 400 gibt einen Prompt (>) aus, wartet ungefähr zwei Sekunden (FOR X=1 TO 1900) und meldet dann den Status (PRINT“YOUR HP :“HP“/“HM:PRINT“THEIRS :“MH).

```
401 PRINT „>“: I = INT ( RND (1) * 25) + 1: IF I <= 5 THEN PRINT „THEY MISSED YOU.“
```

Zeile 401 gibt wieder einen Prompt aus, setzt die Variable I auf eine Zahl von 1 bis 25, und wenn I kleiner 5 ist, meldet das Programm, dass der Spieler nicht getroffen wurde.

```
402 IF I >= 6 AND I <= 10 THEN PRINT „THE „NOS(N)“ HIT YOU HARD!“: HP = HP + AC / 2: HP = HP - AC: HP = HP - I
```

Wenn I eine Zahl von 6 bis 10 ist, wird gemeldet, dass der Spieler schwer getroffen wurde. Dementsprechend werden die Hit Points verringert (HP=HP+AC/2:HP=HP-AC:HP=HP-I).

(HP=HP+AC/2) erhöht die HP um die Hälfte des in AC gespeicherten Wertes, bevor sie um die Hälfte des in AC, des Rüstungsklassen-Flags, gespeicherten Wertes, verringert werden (HP=HP-AC). Schließlich werden die HP um I erhöht (in diesem Fall zwischen 6 und 10), um das Spiel ausgeglichener zu machen.

```
403 IF I >= 11 AND I <= 18 THEN PRINT „THE „NOS(N)“ SMASHED“: PRINT „YOU WITH A BONE-CRUSHING SOUND!“: HP = HP - 150: HP = HP + AC
```

Zeile 403 ist weniger komplex. Hier wird überprüft, ob I eine Zahl von 11 und 18 ist. Ist das der Fall, meldet das Programm, dass das Monster den Spieler mit einem „bone-crushing sound“ getroffen hat.

Die Hit Points des Spielers werden verringert (HP=HP-150), bevor das Programm den Wert der Rüstungsklasse einrechnet (HP=HP+AC). Je höher die Rüstungsklasse, desto weniger HP wird der Spieler nach einem Volltreffer verlieren.

```
404 IF I >= 19 THEN PRINT „YOU DODGED THEIR ATTACK.“
```

Wenn I höher oder gleich 19 ist, hat der Spieler den Angriff des Monsters erfolgreich pariert.

```
405 PRINT „>“: FOR X = 1 TO 1900: NEXT : IF MH < 1 THEN PRINT „THE „NOS(N)“ DIED.“: PRINT „YOU KILLED „NOS(N)“ “: GOTO 410
```

Zeile 405 gibt wieder einen Prompt aus, wartet zwei Sekunden und prüft, ob der Wert von MH (die Hit Points des Monsters) kleiner als 1 ist. Ist das der Fall, erklärt das Spiel, dass das Monster im Kampf gestorben ist und springt in Zeile 410.

```
406 IF HP < 1 THEN PRINT „ALAS, YOU HAVE FALLEN...“: FOR X = 1 TO 1900: NEXT : PRINT „BETTER LUCK NEXT TIME!“: CLR : END
```

Zeile 406 prüft, ob der Spieler in der Schlacht gefallen ist (IF HP<1). Ist dies der Fall, wird das am Bildschirm angezeigt und das Programm legt eine kleine Pause ein. Danach wird das Programm beendet.

In den meisten Adventures würdest du allerdings ins Hauptmenü zurückspringen. Als Alternative wäre ein „Fegefeuer“ denkbar, aus dem der Spieler nach einer Verringerung der Hit Points oder des Goldvorrates zurückkehren kann. In meinen anderen Spielen, z.B. in „Westfront PC: The Trials of Guilder“, verwende ich diese Methode – das Spiel endet nicht einfach, wenn der Spieler in einer Schlacht gefallen ist.

```
407 FOR X = 1 TO 200: NEXT : GOTO 240
```

Zeile 407 lässt das Spiel eine sehr kurze Pause einlegen, bevor das Programm in Zeile 240 zurückspringt.

```
410 FOR X = 7 TO 25: IF LO(X) = 100 + N THEN PRINT „YOU FOUND „NOS(X)“ ON IT!“: LO(X) = RM
```

```
411 NEXT : IF I > 20 THEN LV = LV + 1: HM = HM + 25: HP = HM: MM = MM + 8: MP = MM: G = G + I: PRINT „YOU GAINED A LEVEL!“
```

Die Zeilen 410 und 411 enthalten eine wichtige Funktion: Die Gegenstände eines besieigten Monsters können aufgesammelt werden.

Zeile 410 startet eine Schleife, die von 7 bis 25 läuft (FOR X=7 TO 25). Dadurch werden einerseits die Kompassrichtungen ausgelassen, andererseits durchsucht die Schleife jeden Gegenstand, den ein Monster mit sich führen kann (bis zu X=25). Wenn der Ort von X gleich 100 plus der Wert von N (IF LO(X)=100+N) ist, liegt der Gegenstand auf dem Monster, und eine Nachricht wird ausgegeben. Der beim Untersuchen des toten Monsters gefundene Gegenstand wird dann in den Raum gelegt (LO(X)=RM).

Anmerkung: Normalerweise platziere ich Gegenstände, die Monster mitführen, in Orten über 100 PLUS der Zahl (in N) des Monsters. Wenn also zum Beispiel der Verkäufer (N=18) die Laterne (N=9) hat, ist der Ort der Laterne (LO(9)=118 oder LO(9)=18 + 100).

Zeile 411 erhöht den Wert von X durch das NEXT-Kommando, und wenn das Maximum an Durchläufen (X=25) erreicht ist, wird geprüft, ob I größer als 20 ist (IF I>20). Ist dies der Fall, ist der Spieler um eine Klasse aufgestiegen (LV=LV+1). Zusätzlich wird der Wert in HM (maximale Hit Points) um 25 erhöht (HM=HM+25), die normalen Hit Points und die maximalen Magic Points werden ebenfalls erhöht (HP=HM), (MM=MM+8), (MP=MM). Schließlich wird der Goldvorrat (in G gespeichert) um den Wert der Variable I angehoben.

```
412 G = G + I:EX = EX + N:LO(N) = 998: GOTO 107
```

```
499 END
```

Zeile 412 räumt die Kampfrououtine auf, setzt den Goldwert auf I (G=G+I), erhöht den Erfahrungswert des Spielers um N (EX=EX+N), während das besiegte Monster in den „Spirit“-Raum gelegt wird (LO(N)=998). Dabei handelt es sich normalerweise um den Raum mit der höchsten erlaubten Nummer. Aus Gründen der Übersichtlichkeit verwende ich die Zahl 998. (Zeile 499 beendet das Programm, die Zeile ist also überflüssig und kann weggelassen werden.)

Die Subroutine für die Preise im Shop

Die Zeilen 500 bis 512 legen die Preise im Shop fest. Zuerst wird P auf 0 gesetzt (P=0), dann werden bestimmte Bedingungen geprüft:

```
500 P = 0: IF N = 7 THEN P = 40
```

```
501 IF N = 8 THEN P = 10
```

```
502 IF N = 9 THEN P = 22
```

```
503 IF N = 10 THEN P = 6
```

```
504 IF N = 11 THEN P = 180
```

```
505 IF N = 12 THEN P = 4
```

```
506 IF N = 13 THEN P = 126
```

```
507 IF N = 14 THEN P = 88
```

```
508 IF N = 15 THEN P = 500
```

```
509 IF N = 16 THEN P = 45
```

```
510 IF N = 17 THEN P = 50
```

```
512 RETURN
```

Dieser Abschnitt erklärt sich von selbst: Wenn die Schleife (N) einen bestimmten Wert erreicht, wird das Preis-Flag auf einen bestimmten, in P gespeicherten Wert gesetzt. Wird also zum Beispiel geprüft, ob der Gegenstand LANTERN (N=9) vorrätig ist, wird diese Subroutine aufgerufen, um den Preis zu bestimmen. In diesem Fall wird das Programm in Zeile 502 fündig und legt den Preis mit 22 Goldstücken (P=22) fest.

Die Subroutine zur Bestimmung des Ortes von Gegenständen

Die Zeilen 700 bis 715 sind ungewöhnlich und in einem Adventure normalerweise nicht nötig. Sie wurden zum Debuggen im Programm gelassen. Ich weiß nicht sicher, ob Orathan überhaupt noch auf diese Zeilen zugreift. Dieser Programmteil prüft im Wert, der in einem String gespeichert ist, ob die Buchstabenkombinationen AL, BL, CL, DL, EL, FL, GL, HL, IL, JL, oder KL vorkommen. Wenn das der Fall ist, wird HP\$ in die Mitte des Strings L1\$ geschoben, wo es weiter verarbeitet und in Orte verschoben wird, die in LO(25) (LO(X)=VAL(HP\$)) definiert sind.

```
700 FOR X = 1 TO LEN (L1$)
```

```
701 IF MID$(L1$,X,2) = „AL“ THEN HP$ = MID$(L1$,X + 2,4):LO(7) = VAL (HP$)
```

```
702 IF MID$(L1$,X,2) = „BL“ THEN HP$ = MID$(L1$,X + 2,4):LO(8) = VAL (HP$)
```

```
703 IF MID$(L1$,X,2) = „CL“ THEN HP$ = MID$(L1$,X + 2,4):LO(9) = VAL (HP$)
```

```
704 IF MID$(L1$,X,2) = „DL“ THEN HP$ = MID$(L1$,X + 2,4):LO(10) = VAL (HP$)
```

```
705 IF MID$(L1$,X,2) = „EL“ THEN HP$ = MID$(L1$,X + 2,4):LO(11) = VAL (HP$)
```

```
706 IF MID$(L1$,X,2) = „FL“ THEN HP$ = MID$(L1$,X + 2,4):LO(12) = VAL (HP$)
```

```
707 IF MID$(L1$,X,2) = „GL“ THEN HP$ = MID$(L1$,X + 2,4):LO(13) = VAL (HP$)
```

```
708 IF MID$(L1$,X,2) = „HL“ THEN HP$ = MID$(L1$,X + 2,4):LO(14) = VAL (HP$)
```

```
709 IF MID$(L1$,X,2) = „IL“ THEN HP$ = MID$(L1$,X + 2,4):LO(15) = VAL (HP$)
```

```
710 IF MID$(L1$,X,2) = „JL“ THEN HP$ = MID$(L1$,X + 2,4):LO(16) = VAL (HP$)
```

```
711 IF MID$(L1$,X,2) = „KL“ THEN HP$ = MID$(L1$,X + 2,4):LO(17) = VAL (HP$)
```

```
712 NEXT : RETURN
```

```
714 FOR X = 1 TO 20: PRINT MID$(A$,X,1): NEXT : PRINT : FOR X = 21 TO 40: PRINT MID$(A$,X,1): NEXT :
```

```
PRINT : FOR X = 41 TO 60: PRINT MID$(A$,X,1): NEXT : PRINT : FOR X = 61 TO 80: PRINT MID$(A$,X,1):
```

```
715 NEXT : PRINT : FOR X = 81 TO 100: PRINT MID$(A$,X,1): NEXT : PRINT : RETURN
```

Ausgabe der Anleitung

Die Zeilen 800 bis 804 verwalten die Spielanleitung von Orathan. Das ist eine unumgängliche Notwendigkeit, wenn der Spieler wissen möchte, worum es in dem Spiel eigentlich geht. Alternativ kann ein HELP-Befehl definiert werden, der alle Verben auflistet, die das Spiel „versteht“. Egal, für welche Variante man sich entscheidet, eine Kurzanleitung sollte in jedem Adventure enthalten sein, ob es nun 45 Räume umfasst oder 4.500.

```
800 PRINT „ORATHAN“ : PRINT : PRINT „YOUR GOAL IN ORATHAN IS TO DEFEAT“ : PRINT
```

```
„AN EVIL SORCERER. HE IS SAID TO“
```

```
801 PRINT „BE HOLED UP IN A HULKING CASTLE (JUST“ : PRINT „BEYOND THE FOREST)“ : PRINT : PRINT „HE HAS
```

```
MANY HENCHMAN TO CALL TO HIS“ : PRINT „AIDE, SO BE CAREFUL“ : PRINT
```

```
802 PRINT „PRESS ANY KEY TO BEGIN!“
```

```
803 GET K$: IF K$ = „“ THEN GOTO 803
```

```
804 PRINT „[Clearscreen]“ : RETURN
```

Nach Ausgabe der kurzen Einführung in die Geschichte wird der Spieler in Zeile 803 aufgefordert, eine Taste zu drücken. Das Programm wartet nun auf einen Tastendruck und springt danach in Zeile 804, die den Bildschirm löscht, bevor es zum Ursprung der Subroutine zurückspringt.

Das Verb LIST

Das LIST-Kommando in den Zeilen 850 bis 853 ist nur im Shop von Bedeutung. Wird es dort eingegeben, listet das Programm alle zum Verkauf stehenden Gegenstände auf.

```
850 IF RM <> 3 THEN PRINT „WHAT?“ : GOTO 107
```

Zeile 850 prüft, ob sich der Spieler im Shop (Raum 3) befindet. Ist das nicht der Fall, wird ein simpler Hinweis ausgegeben und das Programm springt zurück zur Hauptschleife.

```
851 IF LO(18) <> 3 THEN PRINT „THE CLERK ISN'T HERE.“ : GOTO 107
```

Zeile 851 prüft, ob sich der Verkäufer im selben Raum wie der Spieler befindet (IF LO(18)<>3). – in diesem Fall Raum 3. Trifft dies nicht zu, wird diese Information am Bildschirm ausgegeben und das Programm springt in Zeile 107.

```
852 SI = 0: FOR X = 7 TO 18: IF LO(X) = 99 THEN PRINT NO$(X):“““ : N = X: GOSUB 500: PRINT P
```

```
853 NEXT : GOTO 107
```

Die Zeilen 852 und 853 führen nun die LIST-Anweisung aus. Zuerst durchläuft das Programm eine kurze Schleife (FOR X=7 TO 18), mit der sichergestellt wird, dass nur Gegenstände von 7 bis 18 beachtet werden – also alle Gegenstände, die der Spieler aufnehmen, tragen, anlegen, ablegen, wegwerfen usw. kann. Ist der Wert von LO(X) 99 (IF LO(X)=99), wird der Name des Gegenstandes angezeigt (PRINT NO\$(X):“““), gefolgt davon, dass N die selbe Nummer wie X zugeordnet wird (N=X), bevor die Subroutine in Zeile 500 (die Preise) aufgerufen wird. Zeile 852 druckt anschließend den Namen des Gegenstandes in P aus, bevor die Schleife in Zeile 853 beendet und der Programmfluss zu Zeile 107 umgeleitet wird.

Das Wort SCORE

Das Kommando SCORE ist in einer einzigen Zeile abgehandelt, obwohl es fast genau so wichtig ist wie INVENTORY. Ohne den Befehl SCORE würde der Spieler nicht wissen, wie viele Hit Points verfügbar sind und wie viele Erfahrungspunkte gesammelt wurden.

```
860 PRINT „YOU ARE “N$““ : PRINT „YOU HAVE “HP“OF“HM“HIT POINTS.“ : PRINT „YOU ALSO
```

```
HAVE“EX“EXPERIENCE“ : PRINT „POINTS AND“G“GOLD COINS.“ : PRINT „YOU ARE CARRYING“IC“ITEMS.“ : GOTO
```

```
107
```

Die DATA-Strukturen: Gegenstände in NO\$(25)

Zuletzt kommen wir zu den DATA-Strukturen. Dabei handelt es sich um den „Motor“ des Programms. Im Prinzip ist ein Textadventure nichts anderes als eine große Datenbank. Die Zeilen 1100 bis 1124 enthalten die Namen der Objekte von NO\$(25). Beachte, dass sie nach Typ geordnet sind: Zuerst die Richtungswörter, dann die gewöhnlichen Gegenstände, dann die Namen der Monster. Das vereinfacht das Verarbeiten des Kommandos und erlaubt den verschiedenen Verb-Routinen gleichzeitig, nach einem bestimmten Muster vorzugehen.

```
1100 DATA „NORTH“, „SOUTH“, „EAST“, „WEST“, „UP“, „DOWN“
```

```
1107 DATA „SWORD“, „TORCH“, „LANTERN“, „ROPE“, „ARMOR“
```

```
1112 DATA „KEY“, „SHIELD“, „ROBE“, „BROADSWORD“, „HELMET“, „RING“
```

```
1118 DATA „CLERK“, „TROLL“, „GOBLIN“, „ELF“, „DRAGON“, „WIZARD“
```

```
1124 DATA „BARBARIAN“, „MORDIMAR“
```

Die DATA-Strukturen: Orte von Gegenständen in LO(25)

Die Programmzeilen 1200 bis 1218 enthalten die Werte von LO(25). Hat ein Gegenstand einen Wert über 17, wird er als Monster betrachtet und kann nicht aufgehoben werden. Weiters setzt Zeile 1200 die ersten sechs Orte von Gegenständen auf den Wert 89, wodurch sie nicht aufgehoben werden können. Wer würde schon Himmelsrichtungen mitnehmen wollen?

Die folgende DATA-Liste sollte sich selbst erklären. Die Zahlen entsprechen den Orten, in denen sich Gegenstände befinden.

1200 DATA 89,89,89,89,89,89
1207 DATA 9,5,99,21,124,5,38,31,99,118,121
1218 DATA 3,41,18,13,15,36,39,43

Die DATA-Strukturen: Die Karte in M(45,6)

Die Zeilen 1301 bis 1343 enthalten die Karte des Spiels, die Raum für Raum in M(45,6) gespeichert ist. Jede Zeile entspricht einem Raum im Spiel und enthält die sechs möglichen Richtungen (North, South, East, West, Up und Down).

Bitte beachte, dass der Spieler, wenn $M(RM,N)>0$, sich in die Richtung bewegen kann, die in N (1 = Norden, 2 = Süden etc.) gespeichert ist. Wenn $(M(RM,N)=0)$, kann sich der Spieler nicht in diese Richtung bewegen, eine Fehlermeldung wird ausgegeben. In der folgenden Tabelle legt jeder Wert, der nicht gleich 0 ist, fest, dass es sich im jeweiligen Raum um keine gültige Richtung handelt. Sehen wir uns als Beispiel Raum 1 an:

1301 DATA 0,2,0,4,0: REM ROOM 1

Der Spieler kann hier NICHT nach Norden gehen, aber er kann sich in südlicher Richtung in Raum 2 bewegen, ebenso nach oben (in Raum 4). Eine Bewegung nach Osten und Westen ist nicht möglich, ebenso wenig nach unten. Nach diesem Schema sind die auch die restlichen Räume zu verstehen:

1302 DATA 1,6,3,0,0,0: REM ROOM 2	1323 DATA 0,0,0,22,0,0: REM ROOM 23
1303 DATA 0,0,0,2,0,0: REM ROOM 3	1324 DATA 18,0,0,0,0,0: REM ROOM 24
1304 DATA 0,0,5,0,0,1: REM ROOM 4	1325 DATA 11,0,28,0,0,0: REM ROOM 25
1305 DATA 0,0,0,4,0,0: REM ROOM 5	1326 DATA 19,0,0,0,0,0: REM ROOM 26
1306 DATA 2,10,7,0,0,0: REM ROOM 6	1327 DATA 0,0,0,0,26: REM ROOM 27
1307 DATA 8,0,6,0,0,0: REM ROOM 7	1328 DATA 0,36,0,25,0,0: REM ROOM 28
1308 DATA 0,7,9,0,0,0: REM ROOM 8	1329 DATA 0,37,0,0,0,0: REM ROOM 29
1309 DATA 0,0,0,8,0,0: REM ROOM 9	1330 DATA 22,33,0,31,0,0: REM ROOM 30
1310 DATA 6,11,12,0,0,0: REM ROOM 10	1331 DATA 0,32,30,0,0,0: REM ROOM 31
1311 DATA 10,25,0,18,0,0: REM ROOM 11	1332 DATA 31,0,0,0,0,0: REM ROOM 32
1312 DATA 0,19,13,10,0,0: REM ROOM 12	1333 DATA 30,0,33,0,0,0: REM ROOM 33
1313 DATA 14,0,15,12,0,0: REM ROOM 13	1334 DATA 0,0,35,33,0,0: REM ROOM 34
1314 DATA 0,13,0,0,0,0: REM ROOM 14	1335 DATA 43,39,0,34,0,0: REM ROOM 35
1315 DATA 0,20,0,13,0,0: REM ROOM 15	1336 DATA 28,42,37,0,0,0: REM ROOM 36
1316 DATA 0,18,0,0,0,0: REM ROOM 16	1337 DATA 29,0,0,36,0,0: REM ROOM 37
1317 DATA 0,0,18,0,0,0: REM ROOM 17	1338 DATA 0,0,0,39,0,0: REM ROOM 38
1318 DATA 16,24,11,17,0,0: REM ROOM 18	1339 DATA 35,41,38,40,0,0: REM ROOM 39
1319 DATA 12,26,0,0,0,0: REM ROOM 19	1340 DATA 0,0,39,0,0,0: REM ROOM 40
1320 DATA 15,0,21,0,0,0: REM ROOM 20	1341 DATA 39,0,42,0,0,0: REM ROOM 41
1321 DATA 0,0,0,20,0,0: REM ROOM 21	1342 DATA 36,0,0,41,0,0: REM ROOM 42
1322 DATA 0,30,23,0,0,0: REM ROOM 22	1343 DATA 0,35,0,0,0,0: REM ROOM 43

Die DATA-Strukturen: Raumbeschreibungen in DE\$(45)

Die Zeilen 1401 bis 1443 enthalten die Beschreibungen der Räume, die im String DE\$(45) in Programmzeile 50 gespeichert werden. Darin sind alle Räume des Spiels eingeschlossen. Wird die Variable RM auf den aktuellen Raum gesetzt, wird eine der Beschreibungen ausgegeben:

1401 DATA „YOU ARE INSIDE A SMALL TAVERN. AN UPSTAIRS HALLWAY HEADS TO THE SECOND FLOOR.“
1402 DATA „YOU ARE STANDING BY A WELL. TO THE SOUTHLIES A PATH OUT OF TOWN.“
1403 DATA „YOU ARE IN A SMALL STORE. COMMANDS HERE INCLUDE „BUY <ITEM>“; „SELL <ITEM>“ AND „LIST“.“
1404 DATA „YOU ARE STANDING ON THE SECOND FLOOR OF THIS TAVERN. A ROOM STANDS TO THE EAST.“
1405 DATA „YOU ARE INSIDE A SMALL ROOM. A BED IS HERE WHERE YOU CAN REST COMFORTABLY.“
1406 DATA „YOU ARE ON A PATH HEADING OUT OF TOWN. A VILLAGE WELL STANDS TO THE NORTH.“
1407 DATA „YOU ARE IN A SMALL CATHEDRAL. AN ALTAR STANDS TO THE NORTH.“
1408 DATA „YOU ARE STANDING IN FRONT OF AN ALTAR. IT HAS A FEW ITEMS ON IT OF INTEREST. A STORAGE ROOM IS TO THE EAST.“
1409 DATA „YOU ARE IN A STORAGE ROOM. NOT MUCH HERE SAVE FOR A FEW WOODEN BARRELS.“
1410 DATA „YOU ARE ON A WELL-LIT PATH HEADING OUT OF TOWN. A FOREST LOOMS TO THE EAST, DARK AND FORBODING.“
1411 DATA „YOU ARE ON A PATH TRAVELING SOUTH. A LUSH, GREEN MEADOW IS DIRECTLY SOUTH.“
1412 DATA „YOU ARE IN A DARK FOREST. TREES SURROUNDYOU IN ALL DIRECTIONS HERE.“
1413 DATA „YOU ARE IN A DARK FOREST. THE TREES ARE QUITE THICK HERE. A LAKE IS DIRECTLY EAST.“
1414 DATA „YOU ARE FACING A DEAD END IN THE FOREST. STORM TOSSED TREES BLOCK YOUR PATH NORTH AND EAST.“
1415 DATA „YOU ARE STANDING BY A LAKE. THE WATER GLISTENS IN THE FADING SUN. A CLEARING IS SOUTH.“
1416 DATA „YOU ARE STANDING IN A BURNT FOREST. THE TREES ARE CHARRED BY A RECENT FIRE. THE ONLY EXIT IS BACK SOUTH.“
1417 DATA „YOU ARE STANDING IN A DYING, BURNT FOREST. IT HAS BEEN CHARRED BLACK TO THE EARTH BY A RECENT BLAZE.“
1418 DATA „YOU HAVE REACHED A BURNT FOREST. THE TREES ARE CHARRED BLACK. TO THE EAST LIES A NARROW PATH.“
1419 DATA „YOU ARE WALKING IN A FOREST. LITTLE CREATURES SCURRY ABOUT THE GROUND, WHILE LIGHT

FILTERS DOWN FROM THE CANOPY ABOVE.“

1420 DATA „YOU ARE STANDING IN A SMALL CLEARING. THERE IS NOTHING AROUND BUT BLUE SKY.“
1421 DATA „YOU HAVE REACHED A SMALL COTTAGE ACROSS FROM A CLEARING. IT IS MOSTLY BARE, BUT HAS SOME USEFUL ITEMS NONETHELESS.“
1422 DATA „YOU ARE INSIDE A DARKENED CELL. IT IS MOSTLY BARE.“
1423 DATA „YOU ARE FACING ANOTHER CELL. THIS ONE HAS A PORTION OF THE WALL KNOCKED OUT.“
1424 DATA „YOU ARE STANDING IN A BURNT FOREST. THE PATH IS BLOCKED SOUTH BY FALLEN TREES AND DEBRIS.“
1425 DATA „YOU ARE WALKING IN A LUSH, GREEN MEADOW. THE GRASS TICKLES YOUR FEET AS YOU WALK ON BY. TO THE EAST, THE HILLSIDE SLOPES DOWN SOME SOUTH.“
1426 DATA „YOU ARE FACING A TREE WITH LOW-LYING BRANCHES. IT LOOKS CLIMBABLE. YOU MAY NEED A ROPE, HOWEVER.“
1427 DATA „YOU ARE STANDING IN A HUT ATOP THE TREE. THERE IS A BOOKSHELF HERE NEXT TO A TABLE. SOMEONE OBVIOUSLY LIVES HERE.“
1428 DATA „YOU ARE WALKING IN A LUSH MEADOW SOUTH OF THE FOREST. TO THE WEST, THE MEADOW OPENS UP SOME INTO A MOUNTAIN RANGE NEARBY.“
1429 DATA „YOU HAVE REACHED A SMALL HILLSIDE IN THE MEADOW. IT SLOPES DOWN HERE NEAR A LARGE PATH (HEADING WEST).“
1430 DATA „YOU ARE INSIDE A DARK DUNGEON BELOW THE CASTLE. IT TURNS HERE AND HEADS EAST PAST A ROW OF CELLS.“
1431 DATA „YOU ARE INSIDE A SMALL CELL. IT IS MOST-LY BARE, SAVE FOR A SMALL TABLE.“
1432 DATA „YOU ARE INSIDE A DARK CELL. THERE IS NOTHING HERE BUT COBWEBS.“
1433 DATA „YOU ARE INSIDE A DARK DUNGEON BELOW THE CASTLE LIBRARY. IT IS VERY DARK HERE! UP AHEAD, YOU CAN MAKE OUT THE FAINT OUT- LINE OF SMALL CELLS.“
1434 DATA „YOU ARE INSIDE THE CASTLE LIBRARY. STRA-NGLY, MOST OF THE BOOKS HAVE BEEN TAKEN.“
1435 DATA „YOU ARE IN A PARLOR INSIDE THE CASTLE. A THRONE ROOM IS NORTH, AND A LIBRARY WEST.“
1436 DATA „YOU ARE ON A PATH BY A LARGE, ROLLING MEADOW. THE PATH BREAKS OFF HERE AND HEADS WEST.“
1437 DATA „YOU ARE IN A LUSH, GREEN MEADOW. IT TURNS HERE BY A LARGE STONE CASTLE.“
1438 DATA „YOU ARE IN A ROOM INSIDE THE CASTLE. IT IS MOSTLY EMPTY.“
1439 DATA „YOU ARE IN THE MAIN HALL OF THE CASTLE. IT HEADS NORTH HERE INTO THE MAIN PAR- LOR.“
1440 DATA „YOU ARE IN A SMALL ROOM. THERE ISN'T MUCH HERE OF INTEREST, OBVIOUSLY.“
1441 DATA „YOU ARE STANDING BEFORE A LARGE, HULKINGCASTLE. THE DRAWBRIDGE IS DOWN. WOULD SOMEBODY BE EXPECTING YOU?“
1442 DATA „YOU ARE ON A PATH NEAR A LARGE MEADOW. A CASTLE LOOMS TO THE WEST, BECKONING YOU TO EXPLORE IT FURTHER.“
1443 DATA „YOU ARE STANDING BEFORE A GOLDEN THRONE.THE JEWELS ENCRUSTED ON EITHER SIDE SHOWA BRILLIANCE RARELY SEEN.“

Die DATA-Strukturen: Verben in VB\$(25)

Die Zeile 1500 ist einfach: Sie enthält eine Liste aller bekannten Verben, die das Programm „versteht“. Je mehr Verben wir hinzufügen, desto komplexer wird das Adventure. Mehr Verben bedeuten aber auch mehr Arbeit. Wenn ein Verb nicht benötigt wird, sollte es also weggelassen werden.

1500 DATA „GO“;„GET“;„DROP“;„WIELD“;„UNWIELD“;„WEAR“;„REMOVE“;„LIGHT“;„EXTINGUISH“;„CLIMB“;„USE“; „KILL“;„BUY“;„SELL“;„EXAMINE“

Die DATA-Strukturen: Beschreibungen von Gegenständen in EX\$(25)

Die Zeilen 1600 bis 1623 enthalten die Namen der Gegenstände, die in NO\$(25) gespeichert sind. Davon müssen die Daten in EX\$(25) unterschieden werden, die die Beschreibungen der Gegenstände enthalten. Das mag verwirrend sein, verwenden also einen passende Variablenbezeichnung wie NO\$ für die NAMEN der Gegenstände und EX\$ für die BESCHREIBUNGEN. Eine Beschreibung ist zum Beispiel „A SMALL TORCH“, während der Name des Gegenstandes „TORCH“ lautet. Je genauer ein Gegenstand beschrieben wird, desto klarer wird das Bild, das sich der Spieler davon macht.

1600 DATA „N“;„S“;„E“;„W“;„U“;„D“
1607 DATA „A NORMAL SWORD.“;„A SMALL TORCH.“;„A BRASS LANTERN STAINED WITH BLOOD.“;„A SMALL COIL OF ROPE.“;„A PIECE OF CHAINLINKED ARMOR.“
1612 DATA „A SMALL KEY.“;„A LARGE CRUSADER SHIELD.“;„AN ELVEN ROBE.“;„A LARGE BROADSWORD.“;„A BRASS HELMET.“;„A MAGICAL RING OF PROTECTION.“
1618 DATA „THIS IS A STOUT CLERK.“;„A VICIOUS TROLL.“;„A GOBLIN WITH HIDEOUS TEETH.“;„AN ELF OF THE SURROUNDING FOREST.“;„A LARGE DRAGON WITH TOUGH SCALES.“
1623 DATA „A WIZARD WITH FLOWING HAIR AND A LONG BEARD. HE APPEARS WISE.“;„A DEADLY BARBARIAN. HE LOOKS MEAN.“;„THIS IS MORDIMAR, AN EVIL SORCERER.“

Zu guter letzt beenden wir „Orathan“ mit folgender Zeile – Kommentar überflüssig: 629999END

Ich hoffe, dass dir die Lektüre dieses kleinen BASIC-Tutorials Spaß gemacht hat. Es erläutert, wie man ein ausreichend komplexes, aber dennoch simples Textadventure programmieren kann.

Paul Allen Panks

Paul Panks' Webseite:

>> <http://www.geocities.com/dunric/westfront.html>



400 Seiten 64'er Longplays

Harald Horschler, der sich schon seit geraumer Zeit als Archivar von C64-Spielelösungen betätigt, hat sich auf ein besonderes verlegerisches Abenteuer eingelassen: Er hat alle im 64'er-Magazin veröffentlichten Longplays in Buchform herausgebracht.

Publikationen zum Thema Commodore 64 sind selten geworden. Die Zeiten, in denen wöchentlich Bücher und Sonderhefte zum Thema C64-Spiele erschienen, sind lange vorbei. Umso erstaunlicher, dass Harald Horschler, den wir in Ausgabe 9 (April 2004) vorgestellt haben, nun den kompletten Katalog von Spielelösungen aus dem 64'er-Magazin in Buchform herausbringt.

Die Beschränkung auf Lösungen aus dem 64'er-Magazin ist eine sehr weise Entscheidung: Einerseits ist das Buch eine geschlossene, komplette Einheit, kein Sammelsurium willkürlich gewählter Spieletipps. Außerdem hoben sich die „Longplays“ genannten Spielehilfen aus diesem Magazin durch einen relativ einheitlich hohen Qualitätsstandard von den Lösungen aus den gängigen Spielmagazinen ab, deren Qualität sehr schwankend war.

Das Buch von Harald Horschler enthält alles, was der C64-Freak braucht, um Beschäftigung für die nächsten 10 Jahre zu finden. Auf 400 Seiten findet man über 50 Longplays, die sehr übersichtlich – unter Einbeziehung wichtiger Informationen zum Spiel sowie des Erscheinungsdatums des Originalartikels – gesetzt und mit allen wichtigen Karten aus den ursprünglichen 64'er-Beiträgen versehen sind.

Der Herausgeber hat sich die Mühe gemacht, die Texte an die aktuellen orthografischen Standards anzupassen, außerdem ist die Umschlaggestaltung so gut gelungen, dass man dem Buch auch gerne einen bevorzugten Platz im Regal zugesteht, um in den Genuss des gelben C64-Fonts auf blauem Hintergrund zu kommen. Das Buch ist nicht nur das ideale Hilfsmittel für Spieler, die die berücksichtigten Spiele ohne POKes und andere

Cheats durchspielen wollen, sondern auch ohne den eigentlichen Verwendungszweck eine sehr unterhaltsame Lektüre.

Hier einige der ca. 50 Titel, die im Buch behandelt werden: The Bard's Tale I und II, Defender of the Crown,

Elite, Elvira I und II, Ghosts'n Goblins, IO, Katakis, The Last Ninja I, II und III, Lemmings, Pirates!, Pool of Radiance, R-Type, Spherical, Times of Lore, Turrigan I und II, Ultima I bis V, Wizball und eine besonders harte Nuss: Zak McKracken.

Harald Horschler (Hrsg.), 64'er Longplays. Spiele-Klassiker für den C64 komplett durchgespielt, 2004, 400 Seiten. 29,80 Euro (Deutschland), 30,70 Euro (Österreich), 45,60 Sfr. Bestellungen im Buchhandel unter Angabe der ISBN 3-83199-00-8 oder direkt beim Verlag:

>> www.scriptorium-vd.de

Out of Orderia 2004 – die einfach schlechtere Party!

Schlechter geht nicht? Doch, zu früh gefreut! Die Gruppe Out of Order setzt dieses Jahr noch einen drauf mit der Out of Orderia 2004 – der einfach schlechteren Party! Auch im neunten Anlauf sind wieder alle Commodore 64-Enthusiasten dazu eingeladen, das Jahr mit einem absoluten Brotkasten-Highlight ausklingen zu lassen.



4 Tage pure Nostalgie unter Gleichgesinnten – doch das ist noch nicht alles:

- * Exklusive Einblicke in aktuelle C64-Spieleprojekte!
- * 1. offizielles Spieleforum für die Spiele-Zukunft am C64
- * Film-Premiere: Der Mythos um die Entwicklung des Ballerklassikers KATAKIS wird gelüftet!
- * Competitions für Kreative & Spielwütige & mehr!
- * SID-Star Search: Beweist Euer Können bei der einzig wahren SID-Karaoke!
- * Auktion (er- und versteigert mitgebrachte Soft- & Hardware)
- * Auf Wunsch: geistlicher Beistand durch Pfarrer Marius!

zwischen Kaiserslautern und Trier); gut erreichbar über die A62 und mit der Bahn (Abholung vom Bahnhof auf Wunsch); Partyplace mit separatem Schlafraum, Dusche, Küche, WCs und Spielwiese.

Start: 26. Dezember, 14 Uhr
Ziel: 29. Dezember, 18 Uhr

Eintritt: 15 Euro; Anmeldung zur Reservierung dringend empfohlen, da begrenzte Platzzahl. Computerkabel und Mehrfachsteckdosen nicht vergessen!

Kontakt: Volker Rust,
Tel. 0160 / 98692490 oder Email:
info@maerchenbraut.de

Detaillierte Informationen zur Party gibt es unter:

<http://www.out-of-order.info>

Wann, Wo & Wie:

Ort: Katholisches Pfarrhaus in Hoppstädten-Weiersbach (nahe Birkenfeld)



Gute Spiele beim Commodore 64 DTV

Wie bereits in der letzten Ausgabe berichtet, hat die Firma Commodore International BV, eine Tochter des niederländischen PC-Herstellers Tulip, eine „all-in-one-Spielkonsole“ nach dem Muster der vergleichbaren Atari- und Intellivision-Produkte entwickelt. Das Gerät, das direkt an den Fernseher angeschlossen werden kann, sieht aus wie ein Competition Pro-Joystick, enthält eine C64-ähnliche Hardware und 31 eingebaute Spiele.

Die C64-Spielkonsole, die sowohl an PAL- als auch an NTSC-Fernseher angeschlossen werden kann, wird von vier AAA-Batterien versorgt. Zwei Wünsche, die von C64-Fans immer wieder geäußert wurden, sind in dieser Fassung, die ab Oktober 2004 im Handel sein soll, nicht berücksichtigt worden: Ein Multiplayer-Modus, indem zwei Geräte miteinander verbunden werden, sowie eine Erweiterbarkeit der Software, z.B. durch Compact Flash-Cartridges. Vermutlich will Commodore abwarten, wie sich die Verkaufszahlen entwickeln, bevor entschieden wird, ein erweitertes Modell zu entwickeln. Immerhin hat man kein Geheimnis daraus gemacht, dass diese Features nicht implementiert sind.

Aber auch in der geplanten Fassung könnte das Gerät durchaus ein Hit

werden – vorausgesetzt natürlich, der Preis ist nicht zu hoch und die Emulation stimmt. Die Grafik sollte keine allzu große Probleme bereiten, beim Ton hingegen könnte man eine böse Überraschung erleben. Da Commodore wohl kaum echte SID-Chips verbaut hat, muss man auf das Schlimmste gefasst sein. Auch die Spiele müssen teilweise an das Gerät angepasst werden, weil sie in der C64-Originalfassung Tastatureingabe benötigen.

Gute Auswahl

Bei der Auswahl der 31 eingebauten Spiele hat Commodore ein gutes Händchen bewiesen. Da Adventures und andere vom Vorhandensein einer Tastatur abhängige Spiele bei einer Joystickkonsole ausscheiden, wurde eine durchaus repräsentative Auswahl an Action- und Sportspielen getroffen.

Sport

Den Beginn machen die klassischen Sportspiele von Epyx. Summer Games, Summer Games II, Winter Games, California Games, World Games und The Games – Winter Edition sind allesamt Spiele der Spitzenklasse, die hoffentlich auch im Mehrspielermodus gespielt werden können. Zwar fallen dann die spannenden direkten Duelle aus, aber schon am C64 machten die Wettkämpfe auch mit nur einem Joystick besonders viel Spaß, wenn sie mit Freunden ausgetragen wurden. Da die sechs Spiele zu den bekanntesten und beliebtesten C64-Titeln schlechthin zählen, sind sie als hervorragende Wahl zu betrachten. Mit Street Sports Baseball und Street Sports Basketball sind zwei weitere hervorragende Titel vertreten, von denen zumindest einer auch in Europa sehr populär ist. Leider wurde dem guten Street Sports Soccer das eher schwache Kick Off (sehr gut: die Amiga-Fassung) vorgezogen. Der Fußball ist also nicht optimal vertreten. Abgerundet werden die Sportspiele durch das mittelmäßige Championship Wrestling. Speedball II aus dem Jahr 1991 muss ebenfalls zur Sportabteilung gerechnet werden. Dieses Spiel gehört zur C64-Spitzenklasse.

Racing

Die Rennspiele beinhalten ebenfalls einige echte Klassiker. Pitstop II und Super Cycle, beide aus dem Hause Epyx, haben zwar im Laufe der Zeit etwas an Anziehungskraft eingebüßt, sind aber noch immer würdige Vertreter ihres Genres. Bei Pitstop II müssen allerdings Abstriche gemacht

werden, da der SplitScreen-Zweispielermodus wegfällt. 4x4 Off-Road Racing ist keine Offenbarung, Buggy Boy wäre wohl die bessere Wahl gewesen.

Großartig hingegen die Renn- und Ballerspielmischung Alleykat, die aus der Feder von Uridium- und Paradroid-Macher Andrew Braybrook stammt.

Action

Auch Actionfreunde kommen beim C64-DTV auf ihre Kosten. Hier bediente man sich vor allem bei Hewson, die gleich mehrere C64-Superhits zu verantworten hatten. Einen guten Start macht Uridium, die Mutter des sauberen Scrollings. Cyberdyne Warrior ist ein brauchbarer Shooter, der allerdings nicht an die furiosen Titel Exolon, Cybernoid und Cybernoid II heranreicht. Sehr gute Wahl!

Puzzle-Elemente enthält der Shooter Netherworld, auch das originelle Nebulus, eine Mischung aus Shooter und Platformer, fehlt nicht. Brauchbar ist auch Eliminator, verzichtbar hingegen das Spiel Eagles, ein aufpolierter Defender-Klon.

Schließlich haben noch Zynaps, ein etwas zu schwieriger Shooter, und Marauder, bei dem die Musik besser ist als das Spiel selbst, ihren Weg ins ROM gefunden.

Puzzle-Spiele

Abgerundet wird die Sammlung vom etwas eigenartigen Puzzler Chip's Challenge und dem ganz netten Mission Impossible. Paradroid und Impossible Mission, zwei der besten Spiele aller Zeiten, sind weitere Höhepunkte des C64-DTV.

Was der Mini-Konsole fehlt, sind ein paar gute Jump'n'Run-Titel. Das liegt wohl daran, dass man sich auf Spiele von Epyx und Hewson konzentriert hat.

>> www.commodoreworld.com

Porto
bezahlen
nicht
vergessen

An
Lotek64
Waltendorfer Hauptstr. 98
A-8042 Graz
österreich

Bestellkarte für Internetverweigerer

Lotek64

- ☐ Ich möchte ein Probeexemplar von Lotek64 zugeschickt bekommen.
- ☐ Ich möchte das Lotek64-Schnorrer-Abo (kostenlos) und bekomme die Zeitschrift unregelmäßig zugeschickt.
- ☐ Ich möchte das Lotek64-Fair-Trade-Abo gegen Erstattung der Portokosten (6 Euro für 5 Ausgaben).

Mein Name: _____

Meine Adresse: _____

E-Mail: _____

C64 Direct To TV mini game console

- 30 games installed.
- Connected simply to the users' TV via RF cables
- Powered by 4 x AAA batteries
- PAL and NTSC compatible.
- 320 x 240 pixel resolution running in full screen mode on TV.
- Game download and connected joystick multiplayer functionality not available on this release.



Demofixing – es kann nur besser werden

In diesem Artikel möchte ich ein paar Erfahrungen mit Demos beschreiben, die nicht so wollen, wie sie sollen. In der Regel löscht man so etwas ja einfach, aber über die Jahre hat mich öfters die Neugier gepackt, so dass ich herausfinden wollte, woran es denn nun liegt. Dabei haben sich einige Fehlermuster herauskristallisiert, die im Folgenden analysiert werden.

– von Stephan Lesch –

Inkompatibilitäten

Zuallererst fiel mir als Besitzer eines C-128 auf, dass etliche Software für den 64er nicht im C64-Modus lief. In aller Regel liegt das daran, dass der Rechner versehentlich in den 2 MHz-Modus geschaltet wird. Der Grund ist meist ein Programmierfehler, der auf einem C64 ohne Wirkung bleibt (wie z.B. eine Schleife, die die Spritefarben setzt und dabei zu weit läuft und das Register \$d030 überschreibt, das die Taktfrequenz steuert – daher auch die Bezeichnung „d030-Bugs“). Ausserdem ist ein Packer im Umlauf, dessen Entpackroutine den Rechner auf höhere Geschwindigkeit schaltet – aber leider nicht mehr zurück...

Eine andere Fehlerquelle stellt das Prozessorportregister des C128 dar, mit dem im C64-Modus die Speicherkonfiguration und die Datensette gesteuert wird. Zusätzlich hat dieses Register ein Bit, in dem der Zustand der ASCII/DIN-Taste erscheint. Wenn nun die Speicherzelle 1 beschrieben wird, um die Speicherkonfiguration zu setzen, ist das kein Problem, wenn es aber gelesen wird, erscheinen andere Werte. In diesem Fall hilft es, einfach die ASCII/DIN-Taste einzurasten. Für Programmierer heißt das, man sollte vor einem Vergleich die Bits ausmaskieren, die einen nicht interessieren.

Solche Kompatibilitätsprobleme sind natürlich kaum vermeidbar, da viele Programme geschrieben werden, bevor eine neue Hardware erscheint, und selbst danach viele Programmierer diese Hardware oder die nötigen Informationen dazu einfach nicht zur Verfügung haben – ähnlich wie bei Spielen, die heute nachträglich an die CMD-Laufwerke oder die SuperCPU angepasst werden.

Datenverluste

Der häufigste Grund für Abstürze von Programmen, die man auf Diskette bekommt, sind sicherlich Kopierfehler, die nicht bemerkt und weiterverbreitet werden. Ebenso kann es

passieren, dass jemand Dateien eines Programms beim Kopieren übersieht, oder nur 35 Spuren einer 40-spürigen Diskette kopiert. In diesen Fällen lässt sich kaum etwas reparieren; man kann nur nach Ersatz suchen: Möglicherweise ist eine Version zu finden, in der der betreffende Teil erhalten ist.

Wenn man sich viel Arbeit machen will, kann man versuchen, defekte Files zu rekonstruieren: so könnte man z.B. bei einer Musikcollection versuchen, ein defektes Musikstück aus einer anderen Quelle zu übernehmen. Bei einem File, bei dem ausgerechnet der Entpacker-Vorspann defekt ist, kann man diesen vielleicht aus einem anderen Programm übernehmen und einige Werte anpassen. Ich würde mir allerdings dreimal überlegen, ob mir das Ergebnis die Arbeit wert ist...

Hoppla, ganz vergessen?

Manches Demo endet nach Drücken der Space-Taste mit einem leeren blauen Bildschirm und einer Ready-Meldung, andere halten an oder stürzen ab. Zum Teil kommt es auch von selbst nach einigen Sekunden bzw. Effekten zum Absturz.

Beim Untersuchen dieser Programme kann sich herausstellen, dass es der Programmierer nicht anders gewollt hat: Interrupts, Sound und Grafik werden zurückgesetzt und das Programm sauber beendet. In anderen

Fällen wird dagegen in einen leeren oder mit Unsinn gefüllten Speicherbereich gesprungen, und es darf geraten werden, ob da ein Programmteil schlicht vergessen wurde, nachträglich verloren gegangen ist (z.B. durch einen Fehler beim Entpacken), oder ob schlicht eine Sprungadresse fehlerhaft ist.

Da nach mehreren Vorgängerparts und diversen Verschiebe- und Entpackvorgängen oft der Speicher mit irgendwelchen Daten- und Codefragmenten gefüllt ist, ist es schwierig, herauszufinden, ob etwas davon bisher unbenutzt ist.

Ungenügendes Testen

Öfter als man meinen sollte, werden Fehler auch von den Autoren selbst übersehen – vor allem durch ungenügendes Testen.

Sicher werden die einzelnen Programmteile fast immer getestet, indem sie einzeln nach einem Reset geladen werden. Wenn diese Teile durch Packen oder Nachladen verbunden werden und hintereinander ablaufen, hinterlassen sie oft irgendwelche Veränderungen im Speicher, die einen späteren Part stören können – das System wird praktisch nie genau so hinterlassen, wie es nach einem Reset aussieht.

So kommt es vor, dass ein vorhergehender Part das Nullbyte in \$0800 überschreibt. Daraufhin können spätere Parts nicht mehr per Run gestartet werden.

Ebenso gibt es Probleme, wenn Teile der Zeropage benötigt werden, aber von einem Vorgängerpart oder einem Entpacker überschrieben wurden. So habe ich ein paar Samples gesehen, die nach einmaligem Abspielen ins Basic zurückspringen wollen – nur funktioniert dies nach dem Packen nicht mehr.

In einem anderen Fall (FHGHG/Agile) fragt ein Demopart den Tastaturpuffer ab, um die Space-Taste zu erkennen, löscht ihn aber vorher nicht. Da die Space-Taste ein Auto-Repeat hat, kann noch vom vorherigen Part ein Leerzeichen im Tastaturpuffer stehen.

Programmierfehler

Ein offenbar nicht ganz seltener Fehler bei Multipart-Demos ist es, innerhalb eines IRQ abzufragen, ob die Space-Taste gedrückt wurde, und in diesem Fall einfach in den nächsten Teil zu springen. Nun werden beim Auftreten eines Interrupts einige Bytes auf den Stack gerettet, die man vor dem Aussprung wieder zurückliest. Wird das Zurücklesen vergessen, bleiben diese Bytes auf dem Stack



Airdance 4/The Ancient Temple

zurück, wenn der nächste Teil gestartet wird. Das wiederum kann zu Kollisionen mit anderen Daten führen, die dieses Programm auf dem Stack speichern will, insbesondere, wenn der Fehler über mehrere Parts gemacht wird, und sich zig Bytes Ballast im Stack ansammeln. Die einzelnen Parts funktionieren dann für sich u.U. einwandfrei, aber nicht hintereinander. So gesehen bei „Summer Code 3“.

Einen anderen interessanten Fehler hatte die Bildersammlung „Suprise“ von Samar. Dort wird eine CIA-Echtzeituhr verwendet, um einige Sekunden zwischen den Bildern abzuwarten – die Uhr aber nicht gestartet. Offenbar läuft sie auf dem Rechner des Programmierers bereits nach dem Einschalten; auf manchen anderen eben nicht, und so wartet die Collection nach dem ersten Bild bis in alle Ewigkeit.

Bei nachladenden Programmen kann es Probleme mit falschen Dateinamen geben – in diesen Fällen steht man dann vor der Frage, ob der Name wirklich falsch ist, oder vielmehr ein File verlorengegangen ist.

Gelegentlich lädt ein Programm Dateien nach, die im Directory eingetragen sind, aber nicht über diese Einträge geladen werden, sondern über eine Spur/Sektor-Tabelle, die im Programm eingebunden ist. (D.h. statt den Umweg über das Directory zu machen, greift das Programm direkt auf die entsprechende Stelle auf der



Airdance 4/The Ancient Temple



Diskette zu.) In diesem Fall lässt sich das Ganze anstandslos per Filecopy kopieren, aber die Kopie funktioniert nicht, wenn die Dateien in anderen Positionen auf der Zieldiskette landen. Wenn man als Programmierer Trackloading verwendet, sollte man also entweder die entsprechenden Directoryeinträge entfernen, oder darauf hinweisen, dass die Diskette nur per Disk-Backup kopiert werden darf.



Summer Code 3/Flash Inc.

Die Prozedur

Wie kann man versuchen, so einen Fehler zu korrigieren?

Zunächst einmal sehen, ob sich der Fehler reproduzieren lässt, und nicht auf Zufall oder der Umgebung (Emulator, Utilitymodule...) beruht.

Dann ist die Stelle zu finden, an der etwas schief läuft – zum einen kann man eine Ahnung haben, was den Fehler verursacht, und nach entsprechenden Codestellen suchen; zum anderen kann man das Programm kurz vor dem Fehler unterbrechen und von dort weiterverfolgen; und schließlich gibt es noch die Möglichkeit, es vom Start ab zu analysieren. Wenn man einen Fehler lokalisiert hat, ist oft auch schon klar, wie er prinzipiell zu beheben oder umgehen ist; man kann dann kurz vor Auftreten des Bugs das Programm unterbrechen und eine Modifikation machen, um zu sehen, ob diese das Problem beseitigt.

Wenn es funktioniert, muss die Modifikation noch in die Datei eingearbeitet werden. Mit etwas Glück ist es sehr einfach, wenn die Korrektur direkt beim Programmstart ausgeführt werden kann (z.B. der Fall, in dem ein Zeichen im Tastaturpuffer vergessen wurde, oder der Stackpointer auf einem falschen Wert steht). Wenn das Programm geändert werden muss, kann man wiederum Glück haben – manchmal steht die entsprechende Sektion im Klartext in der Datei, und man kann einfach einige Befehle überschreiben. Oft ist die betreffende Stelle aber durch Kompression unlesbar geworden und kann nicht direkt geändert werden. In dem Fall kann man entweder die Datei auspa-

cken, ändern und wieder packen. Das bringt das Problem mit sich, dass man entweder untersuchen muss, welche Speicherbereiche eigentlich vom entpackten Programm belegt werden, oder den ganzen Speicher sichern und in Kauf nehmen muss, dass man einigen Müll mitspeichert. Eine andere Methode ist es, einen Speicherbereich zu suchen, der den Entpack/Unlink-Vorgang „überlebt“; dort bringt man dann die nötigen Modifikationen unter, und verbiegt den Entpacker-Aussprung auf diese Stelle.

Man sieht also, man kann sich mit solchen kleinen Flüchtigkeitsfehlern beliebig viel Arbeit machen; deshalb lasse ich ein Demo, bei dem ich nicht weiterkomme, auch schnell links liegen, und versuche es überhaupt nur bei Programmen, die mir gefallen. So interessieren mich z.B. Fehler in Diskmags überhaupt nicht. Ich beschäftige mich auch lieber mit den Fällen, in denen es sich um kleinere Programmfehler zu handeln scheint, denn dort lässt sich oft etwas patchen, und es gibt mir das Gefühl, etwas zu verbessern, nicht nur einen lausigen Kopierfehler zu beheben.

The Stock 64

Die Demos, die ich bisher korrigiert habe, findet ihr auf der Website von The Stock 64 (www.the-stock2000.de). Dort gibt es eine Seite mit Download-Links, zu denen auch jeweils beschrieben wird, was an den einzelnen Programmen gefixt wurde. Dort steht auch die „Stock Wares List“ von Baracuda/Carsten Broschk, eine größere Sammlung von Disketten, auf denen entweder etwas gefixt wurde, oder die einfach vielfach mit Defekten im Umlauf sind, sowie die „Suchliste“ mit fehlerhaften bzw. unvollständigen Programmen. Wenn einmal was nicht laufen will, seht doch nach, ob vielleicht einer von uns eine korrigierte Version hat. Bitte seht euch bei der Gelegenheit auch gleich die Suchliste an – vielleicht habt ihr ja eine seltene 100%-Version in der Diskbox.



Technodreams/Zildjian

>> www.the-stock2000.de

=Der Trilogie vierter Teil =

oder: Die Suche nach den lange verschollen geglaubten vierten Teilen lange erwarteter Spiele

Schon der Name birgt einen hohen Marktwert: *The Last Ninja* ist das meist verkaufte und wahrscheinlich bekannteste Spiel für den Commodore 64. Insgesamt wurden mehr als 2 Millionen Titel der Reihe abgesetzt. Trotzdem gibt es für die Spielgeräte der neueren Generation weder ein Remake noch eine zeitgemäße Umsetzung. Kein Wunder, dass regelmäßig Gerüchte über eine Umsetzung des Stoffs für Windows, Playstation und Co. die Runde machen. Vor zwei Jahren gab es konkrete Hinweise, dass die Firma Simon & Schuster an einer Umsetzung arbeiteten. In ihrem „First Person Shooter“-artigen Spiel sollte der Spieler in die Rolle von Armukuni, dem letzten lebenden Ninja, schlüpfen, um Japan von der Herrschaft des bösen Shoguns Kunitoki zu befreien, zu dessen Gefolgsleuten sogar Dämonen gehören. Action-, Adventure-, Strategie- und Kampfsportelemente sollten zu einem einzigartigen Spiel verschmolzen werden, das nicht nur die Fans der alten Last Ninja-Reihe begeistern sollte.

Zunächst war von der Playstation 2 die Rede, bald kam auch die Xbox dazu.

Ende 2003 wurde Simon & Schuster von Viacom, einem der größten Medienkonzerne der Welt, geschluckt. Die Zukunft von *The Last Ninja 4* schien zunächst ungewiss, im Februar 2004 wurde dann verlautbart, an LN4 würde weiter gearbeitet. Seitdem sind keine Informationen mehr zu finden. Obwohl das Spiel nicht viel mit dem C64-Klassiker gemeinsam hat, wäre es doch schade, würde das Programm ein ähnliches Schicksal erleiden wie andere Neuauflagen wie *The Bard's Tale IV*, das vor vielen Jahren – kurz vor der Fertigstellung – von Interplay fallen gelassen wurde. Das Programm sei zu komplex und programmiertechnisch nicht mehr zu bewältigen, so damals die offizielle Stellungnahme. Während Letzteres nun doch noch – wenn auch in einer völlig anderen, modernen Aufmachung – erscheinen dürfte, ist es um *The Last Ninja 4* still geworden. Schade, die Screenshots sehen jedenfalls vielversprechend aus.



The Last Ninja IV



The Bard's Tale IV



The Bard's Tale IV >> <http://www.inxile-entertainment.com>

Zwei Kernals, zwei Module

Von Antitrack kommt diese Anleitung zum Bau einer Zwei-Kernal- bzw. Zwei-Modul-Schaltung.

Der alte ROM-Chip mit dem Kernal muss aus dem C64 ausgebaut und die folgende Hardware, wie in den beiden Illustrationen auf dieser Seite beschrieben, eingebaut werden.

Hardware:

- 1 x Eprom 27256
- 1 x TTL 74LS00
- 2 x Widerstände 3300 Ohm (10k sollte auch funktionieren)
- 2 x Schalter

Selbst wenig Erfahrene sollten in der Lage sein, das Teil in weniger als 60 Minuten zusammenzubasteln.

Und so funktioniert es

Die Standard ROM-Cardrives füttern Pin 11 („RomL“) vom Modul-Port hin zu CS (Chipselect) und OE (Output Enable) des jeweiligen ROM-Chips. Wenn ein EPROM (das Kernal ROM/Eprom) für verschiedene Speicherplätze benutzt werden soll, müssen wir mehrere chipselect-Leitungen haben. Je nachdem, welche gerade aktiviert ist, muss die höchste address-Leitung des EPROM gesetzt werden. In unserem Beispiel wird A14 des 27256'er Kernal EPROM auf 0 gesetzt, sobald ein Modul-Chip ausgewählt wird. Das passiert einfach, indem RomL von Pin11 des Modulports mit A14 verbunden wird. In allen anderen Fällen wird A14 mittels eines pull-down-Widerstandes (3k3 ohm) auf 1 gesetzt.

Die Memory-Map des EPROM (27256) sieht wie folgt aus:

0000-1fff: Modul 1
2000-3fff: Modul 2
4000-5fff: Kernal 1
6000-7fff: Kernal 2

Wir können einen ganz normalen an/aus-Schalter nutzen, um 0 bzw. 5 Volt auf die A13 des EPROMs zu legen, damit können wir zwischen zwei Paaren von Kernals bzw. Modulen wählen (man kann es eleganter machen – aber ich war zu faul).

Die richtiger Auswahl zwischen den beiden chipselect-Leitungen ergibt sich wie folgt:

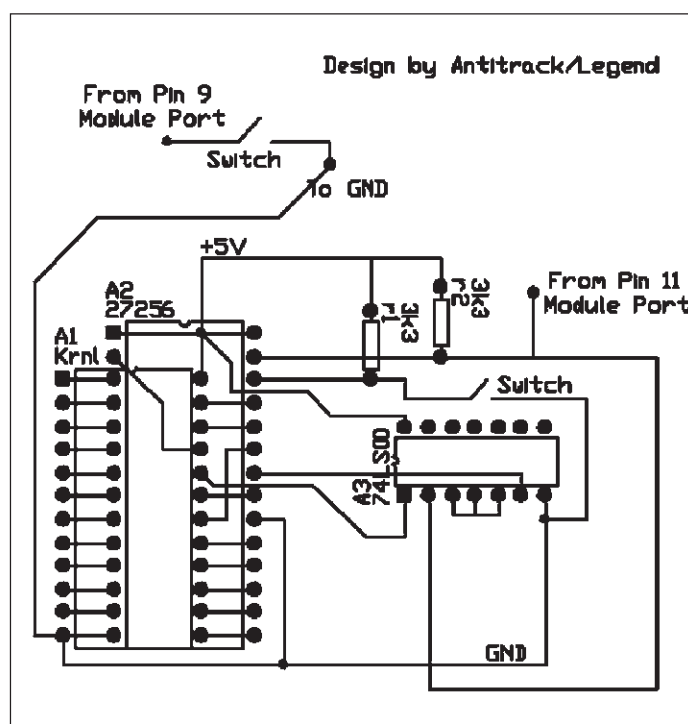
Wenn Pin11 des Modul-Ports 0 ist, muss CS für das 27256'er ebenfalls 0 sein.

Wenn Pin11 des Modul-Ports 1 ist, muss CS für das 27256'er denselben Wert haben wie das Kernalsocket selbst.

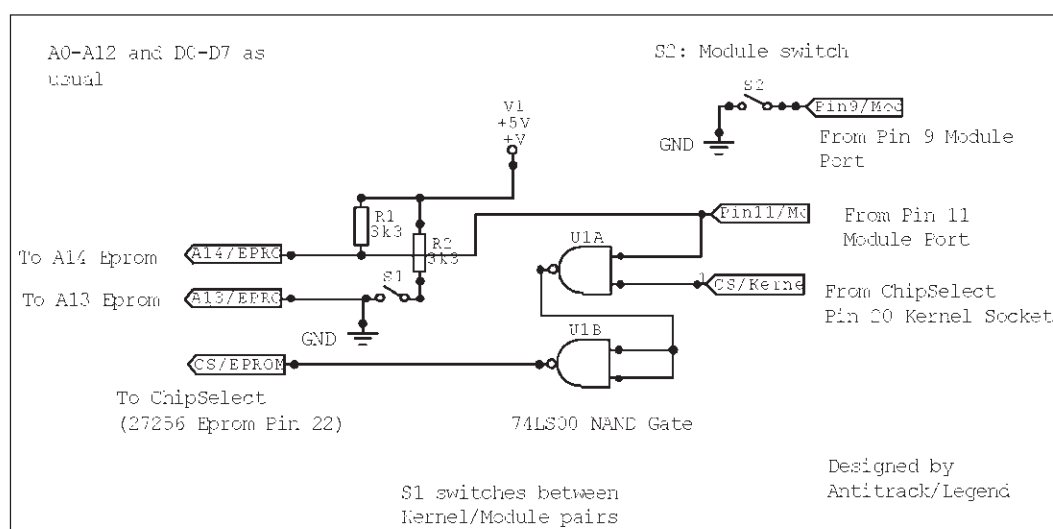
Pin11/ („RomL“)	CS (Kernal)	Ausgabe
0	0 (sollte nicht passieren)	0
0	1	0
1	0	0
1	1	1

Wir brauchen also, alles in allem, nur eine einfache AND-Verbindung, wie sie ein 74LS08 zur Verfügung stellt. In diesem Fall hier aber haben wir ein 74LS00 benutzt, weil das der einzige TTL-chip war, der hier gerade herumflog :-). Man braucht also nur sehr wenig

Euer
Antitrack/Legend



Danke an Hiram „Two Kernals“
Kümpfer für die Übersetzung!



Tabellarisch – siehe rechts oben:

Innovationen von gestern: NES-Klassiker fuer GBA

Magnavox Odyssey, Apple II, Atari VCS, Interton VC4000, Atari 800, Mattel Intellivision, Sinclair ZX81, Commodore VC 20, CBS Collecovision, Vectrex, Sinclair Spectrum, Commodore 64, Atari 5200, Hanimex, Emerson Arcadia, MSX, Sega SG 1000, Master System, Nintendo Famicom – Stopp! An dieser Stelle wollen wir beginnen und über die Neuauflage erfolgreicher NES-Klassiker berichten. Eine Ergänzung zum Artikel „Aus Alt mach Neu“ in Lotek64 #10.

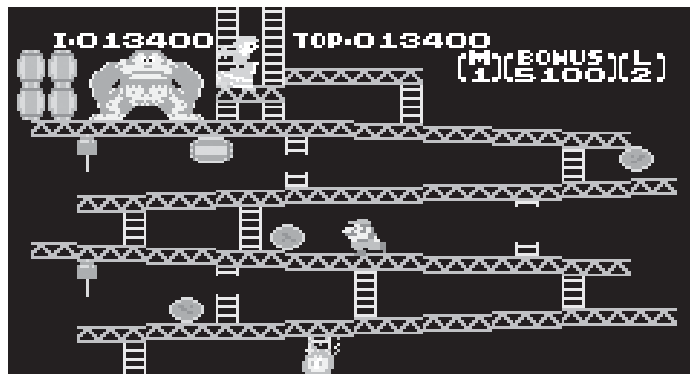
– von Andre Hammer
und Dr. Igor BOFnik –

Woher stammen die Spiele der Classic-Serie ursprünglich?

Von der Urzeit früher Videospiele bis zur modernen Epoche der Next-Generation-Konsolen erstreckt sich ein weiter und imposanter Weg, der vom japanischen Unternehmen NINTENDO maßgeblich geprägt wurde bzw. auch heute noch geprägt wird. Mitte der 80er Jahre, kurz nach dem Zusammenbruch des westlichen Konsolenmarktes, erscheint das Videospiel-system „Nintendo Entertainment System“, kurz NES, im grauen, eckigen Gehäuse. Es besticht durch wohl überlegte Elemente, die das Gerät in jeder erdenklichen Hinsicht besser machen als die Produkte der Konkurrenz. Ein Beispiel hierfür ist die Wahl des Eingabegerätes, welche sich in Form eines Joypads mit nur zwei Feuerknöpfen und dem neuartigen digitalen Steuerkreuz präsentiert, ganz dem Aufbau der frühen „Game & Watch“-Handheldspiele entsprechend. Das Konzept der Schlichtheit geht auf: Bis zur Einstellung der Produktion im Jahr 1994 erscheinen 1.200 Spielertitel. Insgesamt werden laut Herstellerangabe rund 60 Millionen Geräte verkauft. Anlässlich des aktuellen 80er-Trends veröffentlicht Nintendo seinen Game Boy Advance SP im Stil des damaligen NES-Controllers (Rote Feuerbuttons und eine Kombination der Farben Schwarz, Grau und Weiß auf die Hardware übertragen). Außerdem werden acht der erfolgreichsten und besten NES-Klassiker für den Game Boy Advance (SP) neu aufgelegt.

The Legend of Zelda

Der putzige Elfenjunge Link und sein gemeiner Widersacher Canon tragen ihren ersten Kampf um das heilige Triforce aus. Dieses wurde in mehrere Teile gespalten und in verschiedenen Landesteilen von Hyrule verteilt. Zahlreiche Verliese inklusive Endgegner und mystischen Rätseln müssen von



unserem grüngewandeten Helden erkundet werden. Hilfreich ist dabei sein Schwert und diverse andere Utensilien, die Link im Laufe seines Abenteuers erhält. Der kluge Dungeon-Bau und das Erkunden der Karte machen auch heute noch enormen Spaß. Die Melodie entpuppt sich trotz NES-Sound als Ohrwurm. Zu Recht ein Klassiker, dessen Langzeitmotivation nicht nur durch den hohen Schwierigkeitsgrad vorhanden ist.

Donkey Kong

Hierbei handelt es sich um eine exakte Umsetzung des Automaten aus dem Jahr 1981. Der kleinwüchsige Bauarbeiter „Jumpman“, bekannter unter seinem späteren Namen Mario, muss seine Freundin aus den Klauen des bösen Riesenaffen Donkey Kong befreien. Dabei läuft er an Rampen entlang, klettert Leitern hinauf und springt außerdem auch noch über gefährliche Fässer. Insgesamt bietet DK drei Level, die sich mit Ausnahme des letzten ähnlich gestalten.

Super Mario Bros.

Alles, was SMB bietet, war damals die Innovation pur. Sidescrollende Level, durch Rohre erreichbare Bonusräume, Unterwasserareale und unterschiedlich agierende Gegnertypen. In Fragezeichenblöcken sind einige Extras versteckt. Der Klempner Mario wächst bei Berührung mit einem Pilz zu einem Riesen heran und plättet Feinde mit einem gekonnten Sprung auf deren Kopf. Mittels der Feuer-Blume gibt er todbringende Schüsse ab.

Ein Titel, der auch heute nichts von seinem Charme eingebüßt hat.

Pacman

Die kleine gelbe Kugel namens Pacman frisst sich durch die ebenso gelben Pixel innerhalb eines blauen Labyrinths und achtet darauf, nicht von den herumschwirrenden Geistern erwischt zu werden. Das ist alles. Dennoch macht gerade dieses simple Prinzip einen Heidenspaß.

Bomberman, Xevious, Ice Climber & Excite Bike

Die größte Stärke von Bomberman liegt im Mehrspielermodus. Da dieser in der frühen Version fehlt, ist das

Spiel maximal mäßig interessant. Xevious ist ein netter Shooter, der mangels aktueller Vertreter dieses Genres einen genaueren Blick wert ist. Ice Climber dürfte so manchem NES-Spieler Nostalgiegefühle entlocken, da er als eines der ersten NES-Spiele dem Grundgerät beigelegt war. Excite Bike ist ein Motorradsportspiel des Stardesigners Shigeru Miyamoto. Grafik und Sound sind minimal, dafür stimmt der Spielspaß, zumal das Spiel durch den Editor treffend erweitert wurde.

Fazit

Mit einem Verkaufspreis von 19,95 Euro pro Exemplar ist die Classics-Serie überteuert. Donkey Kong wird in einer der ersten Varianten ausgeliefert und hat infolgedessen nur drei Level. Nintendo hätte die in etwa 50 KByte großen Spiele ebenso gut auf ein Modul packen können. Außerdem ist die Serie unserer Meinung nach nicht vollständig. Wo bleibt beispielsweise R.C. Pro Am? Trotzdem wird der Kauf des einen oder anderen Spieles aus dieser Serie kaum ausbleiben.



Der Abdruck erfolgt mit freundlicher Genehmigung des BOF-Teams. Weitere Testberichte des BOF-Teams unter www.soundportal.at unter IT & Multimedia / Games. Allgemeine Produktinformationen unter www.bofshop.at

Die Meinung des BOF-Teams

Andre Hammer: „Neben Boulder Dash war das allseits beliebte Jump n' Run Super Mario Bros mein erstes Spiel für die 8-BIT Konsole NES. Kaum ein anderes Game hat mich im Laufe der Zeit dermaßen süchtig gemacht. Spielern, die bereits zu sehr mit Neuheiten verwöhnt wurden und unbelastet von jeglichen Nostalgiegefühlen sind, rate ich eher von einem Kauf ab. Für Freaks könnte die zehn Euro teurere Game Boy Color-Variante interessant sein. Diese bietet das unveränderte Grundspiel mit einem Haufen Schnickschnack, den man nicht braucht, der dafür jedoch den Spielspaß um einiges aufwertet. Tipp Nr. 2 ist Zelda. Hier bedarf es

keiner großen Worte. Ein Kultspiel, das in keiner guten Game Boy-Sammlung fehlen darf!“

Kaufempfehlung:
Super Mario Bros., Zelda

Christopher Tafner: „Ich bin mir nicht ganz sicher ob man hier überhaupt eine Kaufempfehlung aussprechen kann. Denn wer will schon 19,95 für Pacman ausgeben, das er auch umsonst am Handy haben kann? Die NES Classic Serie ist meiner Meinung nach nur etwas für Nostalgie-Fans die ihre alten Klassiker noch einmal auf dem GBA spielen wollen.“



Phantastisches Remake für Dreamcast-Besitzer:

Giana ist wieder da!

Schwester Giana kehrt dank der großartigen Arbeit von Christian Nowak, Shahzad Sahaib (Kojote), Alexander Oldemeier (Op3rator), Harald Prinzhorn (Arthus) und Rainer Sinsch (Myth) zurück. Für Besitzer einer Dreamcast-Konsole oder eines GP32 gibt es ein Wiedersehen, auf das viele Fans seit langem gewartet haben. Die Rede ist von Giana, der Heldin des C64- und Amiga-Klassikers „The Great Giana Sisters“ von Armin Gessert, Manfred Trenz und Chris Hülsbeck.

Das Original wurde in den 80ern zwar schnell wieder vom Markt genommen – Nintendo bemerkte schnell, dass hier stark von Super Mario abgekupfert worden war –, dennoch ist The Great Giana Sisters bis heute eines der erfolgreichsten C64- und Amiga-Spiele aller Zeiten.



Giana's Return ist kein 1:1-Remake des Originals. Trotzdem werden Giana-Fans auf Anhieb begeistert sein. Die Levels wurden komplett neu designt und sind nun etwas schwieriger als beim Oldie. Grundsätzlich hat sich nicht viel geändert: Super Mario-Pendant Giana läuft von links nach rechts durch die Levels und muss innerhalb eines (teilweise recht knappen) Zeitlimits den Ausgang erreichen. Unterwegs müssen Hindernisse und Gegner durch Springen und/oder Schießen überwunden werden. Durch das Einsammeln von 100 Diamanten wird ein Extraleben erspielt. Anstatt auf Memorycard oder dergleichen zurückzugreifen, machen es die Programmierer so wie in den 1980er Jahren und spendieren erfolgreichen Spielern Levelcodes, die – wie in der guten alten Zeit – auf Papier notiert werden müssen. Die Hintergrundgrafiken sind viel abwechslungsreicher, es gibt nun



mehr Landschaftstypen als nur „blauer Himmel“ und „dunkle Höhle“. Insgesamt gibt es nun sechs Welten à sechs Levels, Bonuslevels nicht eingerechnet. In jeder Welt gibt es zwei Musikstücke, dazu kommt ein Remix der Original-Titelmelodie von Chris Hülsbeck. Der einzige echte Wermutstropfen ist der fehlende Zwei-Spieler-Modus. Mit Maria, Gianas Schwester, gibt es also definitiv kein Wieder-

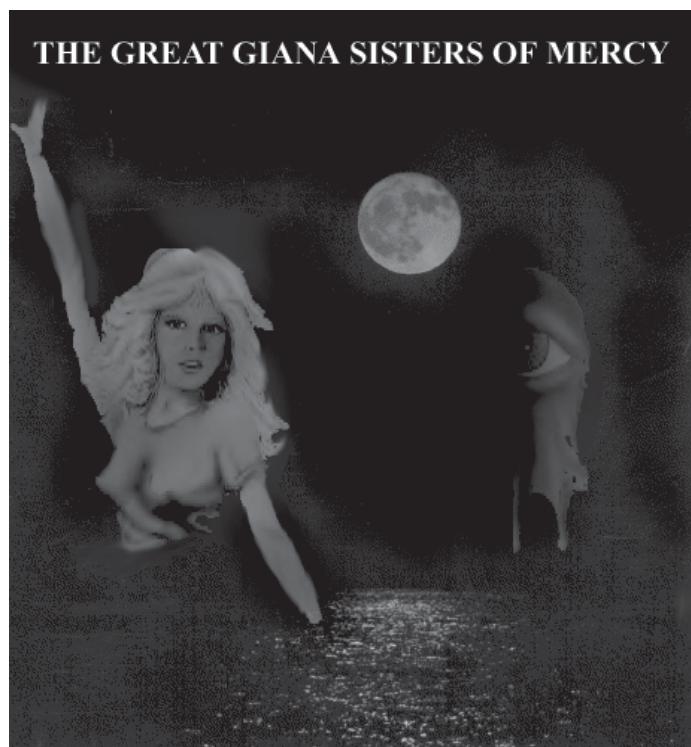
sehen. Das Spiel ist aber so gut, dass darüber getrost hinweggesehen werden kann.

Das Spiel liegt zur Zeit für GP32 und Dreamcast in der Version 0.90 vor. Eine Windows-Version ist ebenfalls in Arbeit, wird aber erst später veröffentlicht.

Giana's Return ist Freeware.

Download unter:

>> <http://www.gianas-return.de>



Erste Hilfe

Simon Quernhorst hat bereits alle Levels durchgespielt und präsentiert hier alle Levelcodes für Giana's Return. Simon: „Manche Stellen sind wirklich schwierig, aber alle Levels sind schaffbar.“

Level 2 QNISON	Level 17 VITTH
Level 3 JPVMM	Level 18 YGZBO
Level 4 FZMXE	Level 19 BHENM
Level 5 LIRKM	Level 20 XIUQO
Level 6 ZISPY	Level 21 UDKVP
Level 7 XITFI	Level 22 TFDRS
Level 8 GMOVG	Level 23 JTVZU
Level 9 GLQNC	Level 24 KOPOQ
Level 10 KTDNK	Level 25 CFJYF
Level 11 FVZGG	Level 26 EMOQL
Level 12 TBXBQ	Level 27 YFCUL
Level 13 WVEGH	Level 28 MWKNU
Level 14 DQXUM	Level 29 SJQDD
Level 15 DEOCF	Level 30 QVBPZ
Level 16 LRBIB	Bonus-Levels nach Endsequenz: ZEHFL



Bonus: Giana-Tipps für den C64

POKE 4595,X – X ist eine Zahl zwischen 0 und 255, die das „Flugverhalten“ Gianas beeinflusst.

POKE 4123,5 – Giana springt extrem hoch und läuft mit hoher Geschwindigkeit.

POKE 2064,96: POKE 5093,16: POKE 5094,8 – schaltet die Musik aus.

Überraschung: FOR T = 49152 TO 53248: POKE T,0: NEXT

Das Spiel wird mit SYS 2083 (wieder) gestartet.

ANY TIME NOW IT WILL BE WORTH BUYING A HOME COMPUTER.

You haven't seen a real home computer until you've seen the VIC 20 by Commodore. But by August you'll get your chance. Because that's when the first VIC 20's will be arriving at your Commodore dealer. Then you can take a good, long look at what makes VIC the best. Like the real typewriter keyboard with full graphics. And eight background

and 16 foreground colours. And music in three voices and three octaves, as well as language and sound effects. So don't think of buying a home computer until then. Because it simply isn't worth it.



COMMODORE 64 GAME CHARTS

1. ZAK McKRACKEN (Lucasfilm 1988)
2. MANIAC MANSION (Lucasfilm 1987)
3. IK+ (System 3 1987)
4. BARBARIAN (Palace Software 1987)
5. URIDIUM (Hewson/Graftgold 1986)
6. PROJECT FIRESTART (Electronic Arts 1989)
7. THE SENTINEL (Firebird 1986)
8. BRUCE LEE (Datasoft 1984)
9. STUNT CAR RACER (MicroStyle 1989)
10. TURRICAN II (Rainbow Arts 1991)
11. SUMMER GAMES (Epyx 1984)
12. ARMALYTE (Thalamus 1988)
13. TURRICAN (Rainbow Arts 1990)
14. WORLD CLASS LEADERBOARD (Access 1987)
15. NEUROMANCER (Electronic Arts 1988)
16. HEAD OVER HEELS (Ocean 1987)
17. MICROPROSE SOCCER (Microprose 1988)
18. WIZBALL (Ocean 1987)
19. KARATEKA (Broderbund 1985)
20. RAMPART (Domark 1992)
21. MUTANTS (Ocean 1987)
22. PARADROID (Hewson/Graftgold 1986)
23. BUBBLE BOBBLE (Firebird 1987)
24. IMPOSSIBLE MISSION 2 (Epyx 1988)
25. LEADERBOARD GOLF (Access 1986)

(Quelle: <http://www.lemon64.com>, Stand 27. August 2004)

Lotek64 #12 PREVIEW

Das legendäre Softwarehaus Magnetic Scrolls hat in den 80er Jahren mit Titeln wie The Pawn, The Guild of Thieves und Jinxter für Furore am Adventuresektor gesorgt. Lotek64 spricht mit dem Gründer des leider untergegangenen Softwarehauses.

Der Playstation ist der nächste Teil der (in dieser Ausgabe entfallenen) CD-Konsolen-Serie gewidmet.

The Ultimate History of Video Games hätte bereits in dieser Ausgabe besprochen werden sollen. Keine Zeit, das alles zu lesen. Vielleicht bis Dezember? Wir werden sehen.

Hier könnte dein Artikel angekündigt sein: Du musst ihn nur schreiben und an die in der nächsten Zeile angeführte Adresse schicken!

Änderungen sind sicher, Texte sind willkommen! -> lotek64@aon.at



Lotek64 #12 erscheint im Dezember 2004.

Lord Lotek LP-Charts September 1989

1. Wreckery – Laying Down Law
2. Fugazi – Margin Walker
3. Melvins – Ozma
4. Faith No More – The Real Thing
5. Cows – Daddy Has a Tail
6. Current 93 – Earth Covers Earth
7. Slint – Tweez
8. Pixies – Doolittle
9. Cosmic Psychos – Go the Hack
10. Beastie Boys – Paul's Boutique



Spex-Reviews September 1989

1. Fugazi – Margin Walker
2. Cosmic Psychos – Go the Hack
3. Scholly D – Am I Black Enough For You?
4. Phranc – I Enjoy Being a Girl
5. Tar Babies – Honey Bubble
6. God Bullies – Mama Womb Womb
7. Two Nice Girls – 2 Nice Girls
8. NWA – Straight Outta Compton
9. Can – Rite Time
10. Chris Isaak – Heart Shaped World



 **Post.at**

Bar freigemacht/Postage paid
8025 Graz
Österreich/Austria