

**BIING_UP! - der schnelle Arzt, wenn die Kasse nicht
Stimmt**

COLLABORATORS

	<i>TITLE :</i> BIING_UP! - der schnelle Arzt, wenn die Kasse nicht Stimmt		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 26, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	BIING_UP! - der schnelle Arzt, wenn die Kasse nicht Stimmt	1
1.1	Biing_Up , der jeden (Geld)Cheat ausschlägt!	1
1.2	bue	2
1.3	wbbu	2
1.4	wbibu	3
1.5	tzbbu	4
1.6	sonst	5
1.7	history	6
1.8	offset	8

Chapter 1

BIING_UP! - der schnelle Arzt, wenn die Kasse nicht Stimmt

1.1 Biing_Up , der jeden (Geld)Cheat ausschlägt!

WICHTIG: SOLLTE DIE MODIFIKATION NICHT WIRKSAM WERDEN
SOLLTEN SIE BIPRO STARTEN. BIPRO IST EINE
.EXE DATEI!
(Sie Sparen sich auch das Intro)

BIING_UP! Einleitung
Was ist das ?

Was braucht Biing_UP! ?
Was wills ?

Wie Bediehe ich BiingUp ?
Wie Wo Was ?

Tips zu Biing & Biing_Up!
Tips und Tricks

H I S T O R Y
Wie Was Wann ?

SONST
Bla Bla

Wie kann ich den Offset finden ?
Falls Sie wirre Angaben haben

1.2 bue

BIING_UP! Die GUI sieht 90 Gadgets vor, davon werden 76 doppelt genutzt.

Du kennst das Proplem bestimmt. Der erste Tag ist der einzige. Man kommt einfach nicht aus dem Negativen Beträgen raus!

Mit viel Experimentieren gelang es mir, die stelle zu finden, in der die Miete Festgehalten wurde. Ich schnappte mir einen HexEditor (leider eine 0.5ß Version) und experimentierte mit BiPro.

Gut, die Gebäude konnte ich jetzt Mieten; je nachdem, wieviel ich eingetragen hatte, bezahlte ich auch.

Die Parzellen waren eigentlich weniger das Proplem, so daß diese in wenigen Minuten ausfindig gemacht wurden.

Die Kosten suchte ich und fand sie. Direkt hinter der Miete....

Gut, ich Experimentierte da ein wenig herum, und mit der Zeit hing es mir zum Hals raus, jedesmal den Editor zu benutzen...

Die Gui mit den 90 Gadgets (die übrigens von der Größe und Position BERECHNET werden) mußte erst einmal stehen.

Das nächste Proplem war es, die 76 Gadgets unter Kontrolle zu bringen. Als erstes mußte ich eine Routine schreiben, die es ermöglicht, Zahlen (Binär) in ein Integer Gadget einzutragen. Ich habe es zwar gelöst, aber leider nicht so, wie ich es gewünscht hätte. Um die Zahl im ASCII - Format zu bringen, habe ich VFPrintf verwendet. Gut, es geht zwar, aber ist etwas Langsam. Vielleicht hilft mir ja jemand. (Oder ich schaffe es alleine.)

Gut, das nächste Problem, was zu lösen war: Wer in dem Integer Gadget den Wert Modifiziert, mußte diesen ja behalten. Die 76 Gadgets werden doppelt verwendet, d.h. das der Inhalt ständig verändert wird.

Ich entwikelte eine Routine, die den tollen Nebeneffekt hat, das eine UNDO Funktion vorhanden ist. Diese muß direkt nach der Veränderung genutzt werden. (Bei Miete das Miete Gadget anklicken.....)

Jetzt muß noch dafür gesorgt werden, daß die GUI auf das Fenster geht. Sollte diese nicht drauf gehen, muß eine Warnung erzeugt werden.

1.3 wbbu

Ich habe BIING_UP! mit OS3.0 68020 und 2MB ChipRam programmiert und getestet.

Ich hatte bereits diverse Propleme mit CopIo und RPLSTR bei kleineren Systemen. Frank Wille, der für mich Enforcer Tests macht (habe keine MMU) hat ein höheres System, und kann mir bei den Proplemen unter OS2.0 68000 nicht helfen.

Also muß ich zweideutig werden:

Theoretische min. Konf.

OS2.0 68000

Garantierte min. Konf.

OS3.0 68020

Sollte Die GUI nicht auf den Bildschirm passen, machen Sie folgendes:
Screen vergrößern (sys:prefs/ScreenMode) oder den Screen auf HiRes
(640*256) und Topatz 8 schalten.

Getestet auch auf meiner Neuen Konfig: A1200 6MB RAM 68030.
BIING_UP! läuft ohne Enforcer oder Mungwall hits. Ich habe daher
auch keine Bedenken, das es auf einer kleineren Konfiguration läuft.

1.4 wbibu

Hm, eigentlich nicht schwer.

Sie haben die ersten 76 Gadgets für die Einträge. Gut, die Abkürzungen
sind etwas merkwürdig, aber für den BIING Spieler ausreichend.

Die Integer-Gadgets sind im Modus Einfügen und können mit der
TAB-Taste umgeschaltet werden. Im gegensatz zu meinen anderen
Programmen muß hier nicht ein IDCMP_GADGETUP (also kein Return oder
TAB) erfolgen, da ich dies für die Integer sowieso nicht abfrage. (Ich
werde mich bemühen, es für zukünftige Programme auch so zu machen.)

Die Restlichen Gadgets:

ABOUT ;-}

HELP ME Geschwätz. Danach benötigen sie Ärztliche Hilfe.

BIING UP! Datei schreiben

MIETE | KOSTEN :

gr. Lager [Miete] [Kosten]

So kannst Du die Miete und die Kosten einer Einheit verändern. Das
erste Gadget gibt die Miete wieder, das andere die Kosten.

Sollte Sie z.B. versehentlich mit DEF M. ihre mit Mühe erarbeiteten
werte löschen, ein weitere Druck auf Miete | Kosten hilft.

DIES SOLLTE ABER DIREKT NACH DEM MISTAKE PASSIEREN, DAS SONST DIE
WERTE VERLOREN GEHEN. DAS GILT AUCH, BEIM WECHSELN IN PARZ | --- !!!!

MAX WERT: 65535 (da Miete sowie die Kosten als Wort gespeichert werden

müssen. Es geht nichts kaputt, sondern der Wert entspricht nicht dem angegebenen. 65536 würde dann 0 entsprechen 65537 1 (WortSchleife)

Parz | --- :

Im Prinzip das gleiche wie Miete | Kosten, nur das Sie die Miete Die nicht benötigten Gadgets, die immer den Wert Null haben, habe ich gesperrt, so daß ein Wechseln mit TAB verbessert wird.

MAX WERT 255 (256 = 0 ; 257 = 1... (ByteSchleife))

DEF M.

Hiermit werden die Werte eingetragen, die Original drinn stehen sollten. (Im Miete | Kosten Modus können Sie dies mit dem Miete | Kosten Gadget rückgängig machen. Im Parz | --- können Sie dies mit dem Parz | ---)

CLR M

Setzt alle 76 Gadgets auf Null

CLR P

Setzt alle 38 Gadgets auf Null

DEF M.

Setzt die Werte so, das beim Ankauf - Verkauf der größt möglichste gewinn erzielt wird. HIER!

ESC - TASTE beendet das Programm wie ein Druck auf das ClodeGadget.

1.5 tzbbu

1. (schlechte !) Möglichkeit

Sie sollten die Miete auf 65535 und die Kosten auf 0 setzen. Zudem die Parzellen auf 0.

Das hatt den vorteil, das Sie Ihre Gebäude für weniger als 'nen Appel und 'n Ei bekommen.

Beim Verkaufen wird die Miete als erlös verwendet.

Gut, da Du es auf 65535 gesezt hast, wird der erlös größer 60000 sein. Wenn Sie also 16 mal ein Gebäude Kaufen und wieder Verkaufen, sind Sie breits Millionär.

Sie sollten Ihre Kosten erst mal so lassen. Die Gebäude nicht auf einmal Kaufen, sondern nach und nach. Effektiv ist es, wenn diese einzeln im Käseblatt erscheinen.

Die Preise langsam runtersetzen. Aggresiev Werbung betreiben.

Schläger nur mit Extrem hohen Ausbildungspunkten einstellen. (6 sind zu wenig).

Nur das Beste Kaufen, sie haben das Geld.

Keine Gebäude Mieten. Wenn doch, sollten Sie diese zuerst abstoßen bevor Sie die Miete hochsetzen. Sie sind sonst schneller Pankrott als gewollt. (Bei meinem ersten Experiment was ich so über 800000 verschuldet).

T H E B E S T W A Y I S:

Du lääßt alles wie es ist, und änderst legedlich für Wartezimmer Standart folgende Werte. Miete auf 65535 und Kosten auf 0.

Sobald Du an diese Einheit rannkommst, kaufst Du eine und verkaüfst es. Der Gewinn beträgt 60K Lümmel. Mache es solange, bis Du genug hast. Das sollte as Startkapital reichen.

Dann setzt Du die Werte wieder auf die Originalwerte. Immer Luxus gebäude Kaufen. Du hast ja genug Kohle...

Der Vorteil: Hast Du obenstehendes getan, rafft der Computer alles zusammen. Die zweite: Der Computer kauft sich eventuell diese Einheit, da er diese aber ← Sowieso benötigt, stellt es kein Proplem für Dich dar. Der Computer verhält sich ansonsten wie Normal. Er Mietet nach und nach. Du kannst dann sofort auf MAX ausrrüsten.... Bei Mög.1. hatte der Rechner sofort alle Gebäude.

Also, 1 Minute Zeit für Kaufen/Verkaufen nehmen und dann geht es besser.

P.s: Sollte das mit den Ändern nicht glappen; BiPro zum Spielstart benutzen ;-{|

1.6 sonst

Ich habe immer ein Offenes Ohr für MANIPULATIONEN. Also, schickt mir die Informationen, am Besten das Game gleich mit. (Nur so kann ich Testen.)

Wer bei BIING noch Stellen in BiPro weiß, als her damit.

(Wer hat BIING! zu verkaufen ? Ich möchte es endlich mal Spielen. Am besten die AGA version.

Also schreib an:

Michael Roth
Hiesbach 4

35410 Hungen Langd.

(Das File namens FORMULAR würde es erleichtern. Ich verwende es auch, um anderen Programmierern zu schreiben....)

1.7 history

Ich hatte die Möglichkeit Biing! zu spielen. Ich überlebte aber nie den ersten Tag. Ich wollte zuviel auf einmal. Mir fehlte auch die Anleitung, um besser bescheid zu wissen. Ich nahm mir die BiPro datei vor und auch Gebäudenamen. Relativ schnell waren die Parzellen gefunden. Die Miete ließ sich auch nicht lange bitten. Die Kosten für den Kauf fand ich erst nach dem ersten Entwurf des GUI, was das Projekt erschweren sollte.... ;*<

Ich Experimentierte ein wenig mit den Werten herum. Mit der Zeit wurde es aber Nervig. Zudem wollte ich nicht einfach schreiben guckt euch mal BiPro an. Eine GUI (Graphical User Interface ~ Benutzeroberfläche) musste her. Es sollte möglich sein, sich den Aktuellen Stand komfortabel anzusehen. Aber ein Monitor für BiPro wollte Ich nicht Bauen. Es sollte Möglich sein die Miete und den Platzverbrauch zu Regeln.

Ich beauftragte einen Freund, das File BiPro auf Regelmäßigkeiten zu untersuchen. Gut, er tat seine Arbeit. Leider durfte ich alles prüfen und Teilweise korregieren, was das Projekt verzögerte.

Ich begann, die GUI zu Programmieren. Schnell stellte ich fest, daß ich zuviele Gadgets brauche. Ich hatte befürchtet, das ich nicht alle Gadgets auf den Bildschirm bekomme. Das hätte bedeutet, das ich ein Hauptfenster und drei Unterfenster (oder so) hätte bauen müssen. Ich entschloß mich aber, erstmal zu sehen, wieviele ich draufbekahm. Als ich die ersten 10 Gadgets fertig hatte (Sie werden immerhin Relativ zu vorhergehenden und relativ zum benutzen Font BERECHNET) dachte ich bereits daran, alle 76 Gadgets in das Fenster zu Quetschen. Als die 76 Gadgets fertig waren, begann ich damit, den Restplatz mit Gadgets vollzubraten. (Sie müssten auch Symetrisch sein.) Ich berechnete als ersets die Linke Ecke. Die Höhe und TopEdge hatte ich von einen der Integer Gadgets (rechts gegenüber) Kopiert. Das einzige was jezt Fehlte, war die Breite. Diese sollte natürlich auch Berechnet werden.

Als ich die Daten meines Freundes überprüfen wollte, fand ich auch die Kosten. Dies machte es nötig, die Daten völlig zu überarbeiten. Als dies fertig war, kopierte ich mir eine Routine von RPLSTR, um die Binäre zahlen zu Convertieren. (Leider ist diese nicht so besonders, aber systemkonform. Brauche eine schnelle Systemkonforme Lösung.) Ich schrieb Sie ein wenig um, und testete, ob Sie Funktioniert.

Ich Programmierete dann auch einen Wechsel, zu Mite|Kosten und Parz|---
.

Als dies getestet war, schrieb ich eine Dos-Routine, die die Aufgabe hatte, den Benötigten Bereich von BiPro zu Kopieren. Das gesammte File ist mir zu groß. Als diese Funktionierte mußte eine Routine her, die die Converter Routine mit den Werten fütterte. Dazu brauchte ich nun die Offsets, die ich mehrmals überprüft hatte.

Das Funktionierte. Ich Wechselte mehrmals. Dabei fielen mir nach einigen Tests folgende sachen auf:

Die Mausezeiger könnte gewechselt werden. Wer eine Eingabe macht, und Wechselt verliert seine Eingabe Für die Parzellen sind 50 % der Gadgets ungenutzt, und können mit TAB amgespungen werden.

Als erstes kümmerte ich mich um die Gadgets bei Parz. Ich habe einfach jedes einzelne, ungenutzte Gadget gesperrt. Dan kahn der Mausezeiger drann. Viel Arbeit war dies nicht. In GoldEd einfach RPLSTR zugeladen und Routinen Kopiert. Diese musste etwas angepasst werden, Jo!

Das mit dem Wechsel sollte mich über lange Zeit beschäftigen. Hier knapperte ich zwei Tage, bis ich eine Routine hatte, die das Wechseln gewärt, und ein Einfaches ins File pusseln ermöglicht.

Gut, die Erste Routine tat Ihre Arbeit. Ich freute mich und begann mit der Routine, die den Krempel ins File schreibt. Ich dachte, ich müste nur die Gadgets auserten... Aber halt! Ich habe ja 76 der Gadgets doppelt benutzt. Die Daten müssen also beim Wechseln irgentwo Gescpeichert werden.

Nach einer Weile hatte ich eine Routine raus, die das so Regelte: Als erstes wird das was im Speicher steht in den Zwischenbuffer Kopiert. Dann wird der Inhalt der Gadgets in die entsprechenden stellen des Files Kopiert. (Je nach wechsel.). Dann wird der Zwischenbuffer für die Gadgets ausgewertet.

Im Prinzip ging es nur Darum, die Daten der Gadgets vor dem löschen zu retten.

Bevor diese Routine funktionierte. waren einige Stunden vergangen. Die Heavy Metal Tapes gingen aus, in der Quitsche lief nichts, was man hätte nebenbei laufen lassen können. Ich saß bis in die Nacht, die Routine musste fertig werden. Ich muß jezt schon schwer überlegen, was die Routine genau macht... Gut, dies war geschafft.

Jezt kahn sie, die Stelle, die über meine Laune bestimmen sollte. Der erste ernsthafte Testlauf kahn.

Nach einem Testaufwand von etwa 3 Tape's viel mir ein, das die Bildschirmeinstellungen sehr unterschiedlich sein können. Bei RPLSTR leiß sich dies noch einfach lösen, aber hier nicht mehr.

Ich testete, ob die Gui überhaupt draufpasst. Ich hätte es einfach machen können, das Fenster einfach mit ChangeWindowBox groß ziehen dann einen Directen vergleich und bei Fehlschlag schließen.

NEIN hier nicht! Da ich die benötigten Daten für ChangeWindowBox eh berechnen mußte, hatte ich einfach vor ChangeWindowBox eine Unteroutine eingebaut, die die Daten des Screens (sc_Width & sc_Height) im Bezug auf wd_Width & wd Height prüft.

Sollte auch nur 1 Pixel fehlen, sollte Die Routine die Arbeit verweigern. Beim ersten nicht hatte ich einen BUG, da der SP nicht stimmte. Ich bin mit BSR eingesprungen und mit b(ne|eq) raus. Ich hatte den bsr korregiert, als ich später den gesammten SP am Anfang des Programmes rettete. So ermögliche ich mir einen einfachen Austieg.

Gut, die Routine musste besser werden. Wenn nur ein Pixel fehlt... Der Anwender bekommt gesagt, wieviel Platz benötigt wird, und wieviel da ist. Soll er doch selbst entscheiden.

Dann bekam ich das nächste Problem: BiPro liegt ja werweiswo, aber ich weiß es nicht. Also muß ich es mir sagen lassen. Sie habe ich UNElegant ;-)) mit einem vor dem GUI Aufbau via FileRequester gemacht.

Ich war so begeistert von Biing_Up!, das ich es für den Versand fertig machte. Ich hatte sogar eine WB Routine drin. Aber: Da ich die Shell als mein Zuhause ansehe (für die WB habe ich zu wenig Speicher zudem geht in der Shell mehr) auch nur dort getestet. Nach dem Versand testete ich einen WB-Lauf und stellte fest, das es keine ERRORS gibt.

Auf dem PC können Programme einfach so beenden. Auf dem Amiga kommt so etwas nicht in Frage. Also mußte der Fehler in ein Requester gequetscht werden. O.K.

Wie gesagt bin ich CLI'ler und ärgerte mich über das FileRequester. Also einfach ReadArgs verwenden und Parameter auswerten. So kann Biing_UP! besser angepasst werden. Wenn ich das mit den ToolTypes rausbekomme, wird es einfacher. Dann wird ein ToolType eingetragen, und der Pfad aus diesem gelesen.

Das letzte, was ich an Biing_Up! tat, war eine kleine Überarbeitung der Texte.

Tjo, in Biing_UP! hat sich noch etwas getan. Also, ich habe Routinen eingebaut, die das Arbeiten auf der WB erleichtern. So ist es jetzt möglich, das Icon von BiPro hineinzuziehen, ohne das es Probleme gibt.

Neue Routine: Sie ermöglicht es, den Offset zu ändern. Ich kann ja schließlich nicht garantieren, das der meiner Version überall stimmt.

Es ist jetzt möglich, über das Zoom Gadget das Fenster zu verkleinern. Diese Werte werden ermittelt, und sollten bei jedem Font noch sinnig sein.

1.8 offset

Ich beschreibe jetzt zwei Möglichkeiten den Offset zu bestimmen.

Also:

Im Prinzip sucht sich BiPro nach dem unten erklärten Weg den Offset selber. Aber, sollte mal nicht Kli_nik-Grund_ein_heit\$\$\$\$\$\$ passen, ist es Ihre Aufgabe nach etwas vergleichbaren zu suchen. Vielleicht heißt es dann: Klinik_Grundeinheit-\$\$\$\$ oder so. Unten wird erklärt, wie Sie sich den Offset bestimmen. Ich habe dazu das mir zur Verfügung stehende BiPro file verwendet.

1. mit HexEditor (z.B. DFM2)

Suchen Sie mit einem HexEditor nach:

Kli_nik-Grund_ein_heit\$\$\$\$\$\$\$\$\$.

Hinter den 8 \$ ist die Adresse, die ich für Biing_up! benötige.

2 mit Type

```

53BA0: 26790006 1D727000 30390006 217CC0FC      &y...rp.09...!|Àü
53BB0: 0064D7C0 7000102A 002A122A 00284A01      .d$\times$Àp...*.*.*(J.
53BC0: 6706B233 000067B6 4CDF7FFF 4E754B6C      g.$^2$3..g¶Lß..NuKl
53BD0: 695F6E69 6B2D4772 756E645F 65696E5F      i_nik-Grund_ein_
53BE0: 68656974 24242424 24242424 003201F4      heit$$$$$$$$$.2.ô
53BF0: 00000000 03FF0000 01010004 00050000      .....
53C00: 00000000 00000000 00004B6C 695F6E69      .....Kli_ni
53C10: 6B2D4175 665F6E61 685F6D65 2C205374      k-Auf_nah_me, St
53C20: 616E5F64 61726424 002F01D6 00000000      an_dard$./.Ö....
53c30: 01010002 02010001 00020003 00110001      .....

```

O.K!

Folgende Zeile ist von Interesse.

```
53BE0: 68656974 24242424 24242424 003201F4      heit$$$$$$$$$.2.ô
```

das h liegt bei mir an der Adresse \$53BE0 (343008)
 eit\$\$\$\$\$\$\$\$\$ sind 11 Bytes dazu. Also 343019. Da ich sagte, 1 Byte hinter
 dem letzten \$ bedeutet noch 1 dazu. Das entspricht 343020.

Ich muß anmerken, das diese Methode Speicherfressend ist und ggf. FileSplitter
 (Wie Copio) verwednet werden müssen.

Bipro wird Hexadezimal ausgegeben etwa 2.5MB groß! (Filegröße.)