Classic AmigaOS4 MorphOS AROS UAE

# AMIGA FORUM

www.suga.se
www.amigaforum.se
MARS 2015 #012

## Hyperion i konkurs?

Vad händer med OS4?

Nya spel
Gamla program
Bokrecension
Sce-fi-novell
Demonytt
Hollywoodskolan

*Innehåller kallelse till SUGA:s föreningsstämma*

# AMIGAstore.eu

Den mest emotsedda nyheten detta nya år måste ha varit AmigaOS 4.1 Final Edition som äntligen började säljas. Reaktionerna har varit blandade. Å ena sidan har de nya funktionerna varit uppskattade, å andra sidan verkar FE lida av buggar.

Detta är dock ingenting jämfört med nyheten att Hyperion Entertainment, ägaren av AmigaOS 4, konkursförklarades av en belgisk domstol den 27 januari i år. När detta upptäcktes en månad senare slog det ner som en bomb i de olika Amigaforumen. Första dygnet var det helt tyst från Hyperion innan Ben Hermans gav ett kryptiskt uttalande att det hela berodde på ett administrativt misstag och att det var för tidigt att dansa på företagets grav. Något han sedan dess vidhållit.

Amiga Forum tog kontakt med konkursförvaltaren Bert Dehandschutter som enbart kunde meddela att Hyperion bestrider konkursförklaringen. Den 27 februari var sista datum då drabbade parter kunde presentera sina inkassokrav, och den 4 mars var dagen då Bert skulle redogöra för sina slutsatser för den belgiska domstolen. Hela ärendet förväntas vara avklarat till slutet av mars.

Till dess är Hyperion fortfarande konkursförklarat och det enda som det finns gott om är spekulationer.

Problemet för AmigaOS 4 är att enligt kontraktet med Amiga Inc. så tillfaller rättigheterna den senare vid en konkurs. Det gör att en eventuell köpare inte kan köpa ut enbart operativsystemet utan måste kanske köpa hela företaget, med skulder och allt. Dessutom ägs stora delar av koden av sina respektive utvecklare vilket ytterligare försvårar situationen ifall någon skulle vilja ta upp utvecklingen.

En konkurs vore inte minst förödande för moralen. Vår lilla scen riskerar att krympa ytterligare ifall de som stött AmigaOS 4 i vått och torrt väljer att överge Amigavärlden helt och hållet. Visserligen finns det ju livaktiga alternativ i form av AROS och MorphOS, men flera har gjort det klart att det inte är ett alternativ.

Amigaland är van vid motgångar och dåliga nyheter, så det är vår förhoppning att det löser sig till det bästa.

*Johannes Genberg*

## *Innehåll*



**Omslagsbilden:** "Fiction2" av Visualize/Jamm, vinnare i grafiktävlingen på Assembly -95.


**Redaktör:** Johannes Genberg
**Layout:** Iggy Drougge

**Medverkande i detta nummer:** Erik Browaldh, Bent Floberg, Jennifer Floberg, Jonas Hofmann, Samuli Holopainen, Stefan Nordlander, Jimmy Wilhelmsson

Amiga Forum ges ut av föreningen Swedish User Group of Amiga (SUGA). Tidningen utkommer 4 ggr/år. Artiklar och föreningsinformation skickas till redax@amigaforum.se. Deadline är den 15 februari, maj, augusti och november. Lösnummer säljs via suga.se/amigaforum/skaffa.html. Mejla redaktionen om din Amigaförening också vill prenumerera.

# Nyheter från Amiga-världen

## Business: Hyperion försatt i konkurs

Den 27 januari i år konkursförklarade en domstol i Belgien Hyperion Entertainment. Hyperion, som är mest känt för att utveckla och sälja AmigaOS 4, har bestridit detta. Hyperion har varit väldigt förtegen om omständigheterna kring detta, men enligt Ben Hermans med omnejd så har en av deras partners försatt dem i denna situation av misstag. Dessutom har deras revisionsfirma glömt att skicka domstolskallelsen till Ben Hermans, varför företaget förklarades i konkurs automatiskt. ”Ett administrativt misstag”, enligt Hermans. Utifall konkursförklaringen kvarstår kommer rättigheterna till AmigaOS 4 att tillfalla Amiga Inc. enligt tidigare avtal, förutom de delar som tillhör respektive utvecklare. Huruvida ett uppköp av hela företaget kan förhindra detta är oklart.

*www.faillissementsdossier.be/en/bankruptcy/1039367/hyperionentertainmentcvba.aspx*


## Film: Ny Amigadokumentär

*From Bedrooms to billions* är en brittisk dokumentär som handlar om den brittiska spelindustrins utveckling från 70-talet tills dags dato. Nu är uppföljaren på väg: *From Bedrooms to billions: The Amiga years* som är en 90 minuter lång dokumentär om vår älsklingsdators betydelse för den engelska spelindustrins utveckling från ensamma hemmakodare till professionella företagare. Precis som föregångaren finansieras denna dokumentär via Kickstarter. Så gå in på Kickstarter senast den 29 mars och donera.

*www.kickstarter.com*

## Business: Cloanto äger rättigheterna till AmigaOS

Trots att Cloanto köpte rättigheterna till samtliga AmigaOS från 1.0 till 3.1 från Amiga Inc. redan 2012 så är det inte förrän nu detta uppmärksammats. Cloanto är mest känt för att sälja Amigaemulatorn Amiga Forever, samt att de på senare tid börjat sälja AmigaOS 3.1 på diskett, compact flashkort eller via nedladdning. Dessa innehåller även ett fåtal buggfixar. Huruvida Cloanto har några planer på att utveckla OS:et vidare är okänt.

*www.cloanto.com*

## Tidning: Norska Amigatidningar återuppstår i tryck

*Amigaguiden* och *Amiga Guide Magazine* är en norsk och en engelsk Amigatidning som ges ut av Norsk Amigaforening (NAF) sedan 1992. Efter några års låg aktivitet planerar föreningen nu att släppa båda tidningarna i tryckt form varje kvartal för 290 norska kronor för den norska utgåvan, och 49 euro för den engelska.

*amiga.zone (norsk utgåva)*
*amigaweb.net (engelsk utgåva)*

## Spel: Svensk hobbyspelsutvecklare stämd

Mikael ”Hipoonius” Persson släppte spelen *Superted* och *Smurf Rescue*, två ganska dåliga plattformsspel för Amigan som han gjorde gratis i Backbone som en kul grej. Inte långt efter att han släppt det senare spelet blev Mikael kontaktad av advokatfirman Gevers som utfärdade ett krav på att spelet togs ner och att han betalade juridiska avgifter till advokatbyrån på först 500 €, för att sedan höjas till

2 000 €. Mikael har nu tagit bort båda spelen från sin hemsida och förklarat att han inte tänker utveckla några fler spel.
*www.hipooniosamigasite.org/smurf*

### NG: AmiDarkbountyn avbruten

*AmiDark* är en tänkt spelmotor för Amiga som i slutet av förra året nådde den begärda summan för att öppna källkoden. När koden skulle utvärderas visade det sig att den var mycket längre ifrån färdig än utvecklaren gett sken av. Efter att en tråd på Amigaworld.net urartat i gräl mellan utvecklaren och de utvärderare som kritiserat kodens tillstånd drog en av huvuddonatorerna in sitt bidrag. Därmed hamnade bountyn under de 1 500 € han begärt och *AmiDark* återgick till stängd källkod. Utvecklaren säger att han ämnar fortsätta utveckla spelmotorn på egen hand.
*www.amidarkengine.com*

### Classic: Petro Tyschtschenko säljer en Walker

Petro Tyschtschenko, tidigare vd för Amiga Technologies under Escom, meddelade via Facebook att han vill sälja en Walker. Walkern var en prototyp på en Amiga 1200-ersättare som Escom utvecklade innan de gick i konkurs 1996. Endast två prototyper gjordes, varav Petro äger båda. Utropspris är satt till 30 000 euro.
*www.facebook.com*

### NG: Hollywood 6.0 släppt

Denna nya version innehåller en mängd buggfixar, optimeringar samt helt nya funktioner. Några nämnvärda nyheter är att programmet nu kan köras på MacOS och Linux, har stöd för ARM-Linux, hanterar nu specialeffekter för filmklipp, har stöd för hårdvaruaccelererat RadeonHD under AmigaOS 4 och kan nu hantera streaming.
*hollywoodmal.com*

### Business: AEon köper OctaMED Studio, ImageFX och Cinemorph

AEon handlar vidare. Denna gång har de köpt källkoden och rättigheterna till trackerprogrammet *OctaMED Studio* samt bildmanipuleringsprogrammen *ImageFX* och *Cinemorph.* Dessa kommer att vidareutvecklas och släppas till Classic och AmigaOS 4, samt eventuellt till AROS och MorphOS.
*www.aeon.com*

## Kallelse till föreningsstämma i Swedish User Group of Amiga

**Lördagen** den 18 april klockan 17.00 är det åter dags för föreningsstämma i Swedish User Group of Amiga. Platsen är föreningens lokal på Folkungagatan 105.

Som föreningens högsta beslutande organ fastställer stämman budget och medlemsavgifter för kommande år samt väljer ny styrelse.

Om du vill kandidera till en post i styrelsen så skriv till Jonas Hofmann (jonas@suga.se) i valberedningen.

**Dagordning:**

1) Mötet förklaras öppnat.
2) Val av ordförande, sekreterare och två justeringsmän tillika rösträknare för föreningsstämman.
3) Fråga om kallelse till föreningsstämman skett på behörigt sätt.
4) Styrelsens verksamhetsberättelse.
5) Ekonomisk berättelse.
6) Revisorernas berättelse.
7) Beslut om ansvarsfrihet för styrelsen.
8) Beslut angående motioner som medlemmar stadgeenligt framfört till ordinarie föreningsstämma.
9) Beslut om medlemsavgift för inlett verksamhetsår.
10) Beslut om inventariemål.
11) Fastställande av verksamhetsplan.
12) Fastställande av budget.
13) Val av styrelse, revisor och valberedning.
14) Övriga frågor.
15) Mötet förklaras avslutat.

# Rapport från SUGA

*Där ordet "EPROM-brännare" fått en ny betydelse*

Så var det dags för ännu en rapport från klubbhögkvarteret. Verksamheten dom senaste månaderna har varit mycket god. Deltagandet på våra lördagsträffar har varit högre än normalt både tack vare att långvaringa medlemmar har passat på att besöka men framför allt så har flera nya bekantskaper träffats. Detta är naturligtvis jätteroligt och jag vill passa på att både hälsa er nya medlemmar välkommna och uppmana alla som funderat på ett besök att göra slag i saken, oavsett hur ditt Amigaintresse ser ut så lär vi ha något som passar dig.

I början av året fick vi en mycket speciell donation. Dels så var det inte den sedvanliga A500:an (inte för att vi säger nej till någon Amiga som behöver ett nytt hem), utan en A3000, och inte vilken A3000 som helst heller. Den här donerades av Niklas Lindholm, skapare av mjukvaran NiKom BBS som fortfarande används av föreningen när vi diskuterar våra lördagsträffar och annat som händer.

Tyvärr var datorn inte helt kry, något var fel med chipminnet vilket yttrade sig genom att grafiken blev korrupt när datorn jobbade. Felet visade sig inte vara själva minnet i sig, utan U202, en GAL som hjälper till med minnesadresseringen. En ny GAL har programmerats och "Nicolina" fungerar nu bra igen.

Inte helt kry är också en beskrivning som numera passar på vår EPROM-raderare. När den som bäst höll på att göra sitt jobb noterade undertecknad att lådan inte var helt stängd. Eftersom man inte vill utsättas för mer UV-ljus än nödvändigt stängde jag luckan och "Tjoff!" så gick säkringen och lukten av saker man nog inte ska andas in började sprida sig. Säkringen (10 A) återställdes och raderaren skruvades isär för inspektion.

Händelseförloppet var omedelbart synligt: EPROM:ar från ett tidigare projekt hade lämnats obevakade i den lilla låda dom ska ligga i när dom raderas, dessa har då smitit över lådans låga kant och kunde nu röra sig fritt i raderaren likt flugor i en insektsdödare. I samma utrymme finns nämligen även raderarens oskyddade och osäkrade nätdel. När lådan trycktes in den sista biten pressades en EPROM mot nätdelen och fick då ett helt ben uppbränt. På grund av min dåliga kinesiska kan jag inte inte uppge namn eller modell på raderaren, men om du har en vill jag varna för dess dåliga konstruktion.

Så ROM:en dog, men hur gick det för raderaren? Efter att ha plockat ut tre förrymda ROM:ar testade vi igen. "Smack!" Denna gång var det inte vår säkring, utan fassäkringen på 20 A i källaren som gått. Tyvärr hade någon passat på att göra åverkan på källardörrslåset så vi kunde inte byta säkring själva och fick klara oss resten av dagen på två faser och utan raderare.


*Den isärplockade EPROM-mördaren*

Vår Amiga 3000 "Ruri" som länge har haft en framträdande roll i föreningen har fått lite välbehövlig omvårdnad. Filsystemet på Work-partitionen (SFS) har varit trasigt sedan innan flytten men nu är detta åtgärdat. Den har också fått sitt nya nätverkskort, ett X-Surf 100 vilket passar den datorn bra eftersom den har 68060. 20 Mb/s får vi ut med NFS och TCPspeed vilket måste räknas som fullt respektabelt. Efter mycket pulande i Picasso 96 lyckades Adam få till ett skärmläge som såg förvånansvärt bra ut på våra 1600x1200 plattskärmar. Snart ska vi skämma bort den ytterligare med ett nyinköpt RapidRoad (usb-modul till X-Surf 100) och RoadShow (samma nätverksstack som i AmigaOS 4).

Vid samma tillfälle köpte vi även in ett andra X-Surf 100 eftersom vi var så nöjda med det första och AmigaOS 4.1 Final Edition för Amiga Classic,

AmigaOne SE/XE/µ och Sam 440ep. Beställningen kom just fram så rapport om hur det gick får vänta till nästa nummer.

Ända sedan vi flyttade in har nätverket varit instabilt av och till. Det skulle kunna bero på att all nätverksutrustning har minst 15 år på nacken eller på att det är draget i äkta anarkiststil. Men oavsett orsak så har bristerna blivit mer kännbara nu när aktiviteten har ökat. Vid ett tillfälle hängde sig en switch, en hub och en ethernettransceiver kort efter varandra. Efter detta köptes två nya 8-portars gigabitswitchar in och efter det har problemen minskat något.

Det finns dock fortfarande mycket förbättringspotential, till exempel är vår feta PowerMac med MorphOS och gigabitnätverkskort fortfarande kopplad till filservern (som också har gigabit) via en 10 megabits BNC-länk eftersom det är vad som förbinder rummets två sidor. Diskussioner har förts om att göra ett omtag på nätverket och ett förslag kommer att läggas på stämman om att budgetera för inköp av en rulle installationskabel och fasta nätverksuttag.

Mycket annat har hunnits med under dessa månader, både stort och smått. Kjell har jobbat mycket med våra A2090-kort, det har programmerats ARexx, installerats större hårddisk i vår Pegasos, den NFS-utdelade zpoolen i filservern består nu av hela tre speglade diskar, det har gjorts backup på flera maskiner och backuper på backuper, det har gjorts en större beställning på Elfa av komponenter och lödutrustning, det har vårstädats och vår stora scanner fungerar nu ihop med Pegasosen.

*Jonas Hofmann*

# *Rapport från Safir*

## *Vårsol i Amigaland*


Vintern har varken varit kall eller ruggig här i söder. Vårt community har inte heller lidit av nyhetstorka – men visst känns det skönt när vårsolen lyser in genom fönstret?

Midvintern bjöd bland annat på en ny version av AmigaOS 4, några riktigt fräcka uppdateringar till Icaros och så klart en rad nya program och spel till MorphOS. För er som inte följer nyheterna eller kollar av OS4Depot.net kan jag avslöja att det händer saker hela tiden! Bland annat har AmigaOS 4 nu fått stöd för versionskontrollsystemet Git, bättre uppspelning av HD-video tack vare nya Radeon-drivrutiner, en JRE (Java) tack vare JAmiga-projektet samt den vanliga skörden av nya och uppdaterade program, spel och emulatorer.

Faktum är att våra AmigaOS idag äntligen börjar ta det steg vi alla länge väntat på – från hobbyplattform till ett till-vardags-användbart operativsystem. På safir.amigaos.se kan vi lite då och då läsa om medlemmar som valt att prioritera sin Amiga-plattform, och till stor del klarar sig utmärkt med denna som primär arbetsstation. Något vi fortfarande saknar – trots en fungerande betaversion av Mozillas Firefox till OS4 – är kanske en webbläsare som håller måttet. Och med det menar jag personligen saker som integrerad medieuppspelning, fullt javastöd, kanske, ja kanske, till och med fungerande Flash och PDF-integrering.

Så visst finns det plats för fler engagerade utvecklare. Och jag *vet* att det finns många Amigaintresserade personer där ute som sitter på *kompetensen* men som kanske saknar *tiden*. Men det behövs inte så mycket tid för att ladda ner, ändra ett par rader och kompilera om en mindre svit för att rulla på allas vårt favorit-OS! Tag ett par minuter och kolla in www.amigabounty.net, där finns idag 40 förslagna projekt att bara ta tag i!

På den lokala fronten har vi vårens stora påskparty hos Syntax Society att se fram mot. Datumet är satt till 10-12/4. Alltså helgen efter påsk denna gång. Missa inte detta!

Det var allt för denna gång, väl mött och LLAP!

*Stefan "shoe" Nordlander & Safirs moderatorer*

# Buggo

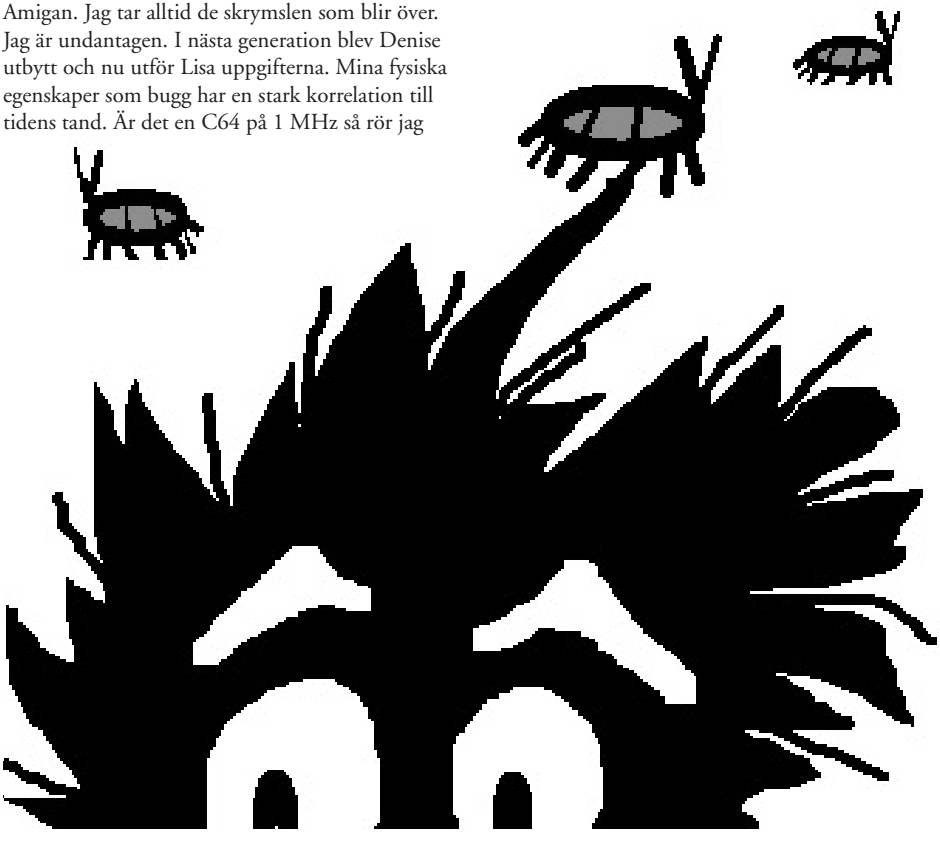
en sce-fi-novell av Browallia/Nukleus

**Porträttet**

Jag är en bug. Du vill ogärna erkänna nyfikenhet, men hur föreställer du dig mig? Som det galna spöket Slimer från *Ghostbusters*, eller någon odör eller svulst? En bug är ju en typ av insekt. Kanske tror du att jag hoppar omkring, eller får du charmiga associationer till Kakmonstret i *Mupparna*? Ja du, jag kan ta många skepnader och jag älskar maskerad. Om nu någon skulle vilja bjuda mig.

De flesta tror att buggar är en slump, men jag är med och styr en del faktiskt. Jag kan inte glida över tiden på samma sätt som spökgruppens härjanden. Nej jag är fast i fysikens lagliga box. Transporter sker genom elektrosmog, eller det långsammare sättet genom elnät. Därmed kan jag hoppa mellan datorer men tiden för mig framåt. Denis E konstruerade hur 32-bitarsdata kunde skickas till bitplanes och sprites. Resultatet från arkitektens verk heter Denise och har en fast sovplats i Amigan. Jag tar alltid de skrymslen som blir över. Jag är undantagen. I nästa generation blev Denise utbytt och nu utför Lisa uppgifterna. Mina fysiska egenskaper som bugg har en stark korrelation till tidens tand. Är det en C64 på 1 MHz så rör jag mig på ett sätt, medan jag rör mig på ett annat sätt i alla dessa nya smartphones som börjar dyka upp. Jag älskar inte all hårdvara. Jag gillar att sätta tänderna i Amigan, C64:an och en och annan Atariscener har väl också rivit sitt hår på grund av mig. Jag tycker MorphOS är mer spännande än Linux. PC-datorer är extremt opersonliga så även mitt jobb som bug blir anonymt och grått. Smartphones ska vi inte tala om. Där behöver jag knappt visa mig förrän allt blir instabilt.

Jag har många former, men inne i min verkstad, det vill säga i datorn, föredrar jag en. Föreställ dig en oval gul tvål med en liten mun, styva morrhår inte olikt en tät tandborste. Min frisyr skulle de flesta förväxla med mossa. Det är i mitt hår som mina buggpartiklar lever och de styrs likt radiobilar genom att alstra energi ut i morrhåren. Jag är en slags multidirigent. Mina ögon liknar fiskögon



och mina tänder är skarpa nog att krossa en varuvagn. Jag saknar näsa, men det är min löddriga kropp som luktar och registrerar observationer. Ingen vacker syn för ett porträtt, eller hur? Om du nån gång skulle vilja ha en bugg i ditt program behöver du inte ropa mitt namn. Jag kommer ändå och är opålitlig. Jag lämnar dig inte frivilligt. Inte ens när du kodat in mig i karantän. Jag glider ut när jag inte kan uträtta något mer, och det är en helt annan sak. Buggo är mitt namn.

**Det kollektiva boendet**

– Det här suger!

– Vad menar du? Lisa tittade upp från sina naglar hon höll på och putsa. Alice stannade också upp men hade fullt sjå att fixa till sin gurkmask i ansiktet.

– Det är total demotorka! Vad hände med alla Amigademos? C64-scenen lever, men Amigascenen verkar bara leva upp ett par gånger per år. Går ju fan att ställa en väckarklocka och gå i ide.

– Skulle du gå och lägga dig att sova? Skulle inte tro det, svarade Lisa och provlade sig på sin kontaktyta bredvid.

– Jäkla register alltså, muttrade hon. Denise berättade om att hon bara har 32 små färger att putsa och dona med. Jag måste hela tiden se upp var alla dessa 256 färghinkar är så jag inte trampar nån på tårna.

– Eller gör nån illa med dåligt putsade naglar, flinade Buggo.

– Vad menar du? Iiiiiih, WTF har du gjort, har du vässat min nagelfil så att jag fått vassa kanter!

– Hahaha, flinade Buggo, det är det jag är bra på. Nu kommer någon demoprogrammerare riva sitt hår när bitplanen flimrar till och glappar. Du får klippa lite bättre nästa gång, Lisa.

– Du din lilla…

Buggo hoppade på en logisk buss på väg mot CPU:n medan han vinkade adios till en illröd Lisa. Han försvann in i CPU-tunneln och dök in i en C64. Där kämpades det med logiska operationer som kopierades till RAM på adresserna $C000 – CFFF. Buggo slängde iväg en spottloska och sölade ner ett par kretsar med sin löddriga kropp. Sådär ja, nu får vi se när rosten smyger sig in här, hahaha.

Ja, så där håller det på. Dagarna i ända. Har jag ett varierande jobb? Självklart. Alla karaktärer finns inne i min verkstad. Olika chipset och kretsar med sin personlighet. Ta till exempel Henric trädgårdsmästaren som sopar rent för CPU-bussen varje morgon. Usch vilket trist jobb. Slentrian så det stör! Han blev nog jätteglad när jag visade en ny lövhög som fyllde upp halva stacken. Stackarn, han fattade ingenting men krattade bort det. Sen har vi ju i Amigan mr Guru himself. Mr Guru är det som de flesta Amiganer känner som den rödblinkande felkoden. Många skyller på honom, men faktum är att han är oskyldig. Han gillar sin rektangulära kropp och blir glad att kunna blinka ut sina fel. Tyvärr så får ju han ofta skulden eftersom det är han som användaren ser på skärmen. Bakom fasaden så brukar jag kittla mr Guru med mina buggpartiklar. Kontakt-interfacet för TAC2-joysticken är en annan lustig filur jag brukar morsa på. Mina tänder gnager lite och simsalabim hokus pokus så blir det lite glapp. Alla har sin plats under datorchassit. Alla är vi delar av något och detta var ett axplock av personligheterna i datorns trädgård.

**Den isolerade involveraren**

Efter turen i C64:an hittade han en dator där en tonårskille glömt spara sin uppsats. Han rufsade till lite i ROM och vips så startade datorn om. Buggo skrattade åt sitt skämt och tittade sig omkring. Det var ingen som ville dela det komiska i det hela och hans skratt klingade ut. Därefter lade hans sig till ro att sova i d2-registret som var tomt för tillfället. Han hoppades att han skulle drömma om nya destruktiva idéer för nästa dag men hans sömn blev orolig och han vände sig flera gånger så att han var tvungen att tömma ut d2-registret flera gånger som var helt nedblött av hans buggloddder. Han drömde en hemsk mardröm att allt han gjorde inte spelade någon roll längre. Att han inte kunde påverka några roliga demos längre och att han inte fick någon uppmärksamhet av Lisa-chipset, Alice, eller att till och med escapeknappen vände honom ryggen. Sen drömde han att TAC2-joysticken spände fast honom på sin knapp och att elaka tonåringar spelade *Track 'n Field* och boxade honom i magen med sina svettiga tummar. Du är en basilisk och lämnar inga avtryck i den här världen, bankade tummarna honom medan figuren i *Track 'n Field* sprang sitt lopp. Buggo vaknade med ett ryck.

– Gaah, jag gör inget avtryck, hemska tanke.

Han satte sig upp och gled ner från d2-registret och lade sig bredvid en morgonpigg transistor som höll på med en räkneuppgift.

– Vad är meningen med livet, frågade Buggo transistorn.

– Schhh! Stör inte, jag håller på att räkna.

Buggo blev förbannad för denna behandling och tänkte bita ett hål i påsen så alla siffror som transistorn samlat ihop skulle läcka ut igen men knep ihop. Han gick vidare, eller snarare löddrade vidare på kiselplattan. Henric trädgårdsmästaren drack en kopp kaffe med Larry, chauffören för den servicebussen. Han smög närmare och försökte smälta in i mängden och hoppade in i en instruktionskö.

– Jo men visst är det tragiskt med Buggo, chauffören smuttade på sitt kaffe.

– Han är verkligen ensam och hur kul är det att skratta när det inte finns någon som lyssnar till hans glädje. Henric trädgårdsmästaren spolade av servicebussen och kände på mönsterdjupet på däcken.

– Det jag tycker är sorgligast, är att han saknar historia. Buggo den lilla basilisken har inget minne stackar'n. De pratar om mig, tänkte Buggo medan han föstes framåt i instruktionskön.

– Ja, fortsatte chauffören. Han har inget minne och kan inte riktigt komma ihåg något trevligt han gjort. Börjar liksom om.

– Mmmmm, får det var en kopp kaffe till, trugade Henric.

– Nej, måste sticka. Har en leverans om två frames med lite data att skyffla in i CPU:n, so-long!

Buggo rynkade på ögonbrynen. Komponenterna kände till honom. och vad värre var, de tyckte synd om honom. Han var nästan framme först i instruktionskön. En administrator stod och skickade ut dem på olika ledningar. Han rusade ut ur kön och välte omkull de andra som slirade i löddret. I vanliga fall skulle han skrattat åt sin latency han skapat, men nu var han för upptagen med funderingar. Han satte sig på en stubbe som nån C64-programmerare skapat och filosoferade för första gången i sitt liv.

Det var alltså detta som var basiliskens dilemma. Jag talar om mig själv i tredje person, eftersom det är så det känns. När ingen ens vill ta i mig med tång över alla dessa årtionden så blir det så. TI-81:a eller dator med 3d-skärm – sak samma. Då blir det lätt så att min identitet ser jag själv från ovan i ett helikopterperspektiv. Min egen identitet har skvalpat ut i ett tredjepersonspronomen. Vad är jag då bra på? Jag är kreativ, och hittar ständigt nya vägar att busa. Tycker nog jag är bra på att hålla bollar i luften också på ett strukturerat sätt. En del attacker kräver synkroniserade kommandon. Är bra på att ge ut order. Han gnuggade sitt hår som självbekräftelse och några buggpartiklar klev ut och väntade på uppdrag. När så inte skedde vibrerade de in igen. Buggpartiklar, flinade Buggo. Dessa varelser kan jag terrorisera genom att skicka in och ut 87 miljarder gånger och de kommer inte bry sig. De har inget minne. Endast en bit som är tänd eller släckt – angrip respektive angrip ej. Men det är inte kul att terrorisera dessa, eftersom de inte har någon återkoppling, Buggos flin minskade något och försvann sedan helt.

Kanske har chaffisen och Henric rätt. Buggo blev med ens en smula dyster. Kanske spökar mitt minne verkligen? Har jag någonsin berättat en anekdot från det svängiga 80-talet för Agnus, Denise, fröken 030-Apollo eller mr Guru? Dessa och fler som huserar i min verkstad betyder väl nåt antar jag? En insikt kröp upp i Buggos ansikte. Jag minns eftersom mina chipbröder och transistorsystrar återberättar för mig, flämtade han. Den gamla hederliga ROM:en minns och är en jäkel på att dra upp gammalt groll som en gång landat hos honom. Eller vad sägs om DFO:an? Även om hon är långsam mot vilken flashdisk som helst idag, så nog fasiken får hon fram informationen på sektorerna. Likt en seg bibliotekarie med sina kartotek.

**Den problematiska disketten**

Som på kommando när Buggo filosoferade om sin längtan till kontakt kom Cynthia förbi. Cynthia var mer känd som tjejen bakom C64:ans Starfirevirus. Det lade sig i en sektor som var ledig, för att sedan flyttas till adress $0801 där kopiering av nonsens startade. Hon ville som vanligt dricka kaffe med mig. Kaffe gick att hämta i flera hackares datorer där det spilldes kaffe och läsk över tangentborden. Egentligen räckte det med att någon nös eller andades kaffe. Ett par molekyler räckte för en fullgod kafferast. Nu ska du kära läsare som läser mina tankar inte tro att Cynthia är av samma skrot och korn som jag. Nej, jag avskyr henne som pesten. Denna självgoda trista apa som bara gör samma sak hela tiden. Vig som en katt var hon också. Jag lyfte blicken och tittade på när Cynthia hoppade upp i en minnesplats som först skulle tömmas om 2,3648 nanosekunder. All tid i världen för att höra hennes jamanden, stönade Buggo. Hon babblade på och visade flimrande

bilder på sin kropp på olika ställen som fått hennes kod kopierad.

– Ja men visa mig då, gör ditt trick, sa hon med sin oljigt smickrande röst.

– Huh? Buggo väcktes upp ur sina tankar.

– Gör nåt och muntra upp mig. Eller jag slår vad om att du inte kan, retades hon.

Han blev halvt förbannad. Här satt han och tänkte på viktiga beslut i sitt liv men okej då, kattskrället Cynthia skulle allt få se. Han transporterade Cynthia i en del av sin kropp. Hon skulle nämligen förintas om hon kom utanför andra delar än C64:an. Ungefär som att människan inte kunde andas på Mars på grund av andra partikelhalter de ej var skapta för. Cynthia var inte multiacklimatiserad som han. Samtidigt var det en slags symbios att låta andra delar få vandra med honom mellan olika arkitekturer. Skulle han öppna slussen så skulle det även skada hans egna kropp. Dessutom så var Star-fireviruset från Cynthia på tiotusentals ställen världen över, så hon skulle bara återkomma i en likadan form. Nåväl, Buggo rasslade in i Zinths dator. Han hade varit inne mycket i koden från Level 10b och gick fram till ROM:en för att lyssna på sin plan han lagrat där tidigare. Egentligen hade han inte alls nån lust, men Cynthia skulle hänga efter honom länge och ihärdigt om han inte gjorde nåt. Buggo skickade ut sina spejare för att reka. När buggpartiklarna kom tillbaka formade han om sitt mossiga hår till en militärbasker av graden överstelöjtnant.

– Okej så här är läget, dikterade jag. Inom kort kommer det en buss förbi här som läcker data. Ett dataregister försöker hela tiden kompensera genom att sätta carry-flaggan till 1 för att sedan göra logiska operationer. Problemet är att det hela tiden är fullt i bussen då den åker runt med en jättestor sampling. Kodern som har gjort det här heter Zinth och just idag så ska han dra över sitt bidrag till en floppydisk. Jag kommer sabba hans bidrag. Jag hostade till och härmade samplingen. Cynthia klappade förtjust ihop sina små händer. Jag kände mig mindre upprymd än vanligt, en smak av moral för första gången i mitt liv i munnen, men jag spottade ut de tankarna.

Mycket riktigt dök en överfull buss upp med tjutande däck rakt emot oss. Det var en liten dubbeldäckare med 16 säten där nere och lika många där uppe. Efter den sprang Lothar, vaktmästaren och gestikulerade med nävarna fulla med data som spillts ut. STOPPA BUSSEN! Lothar flåsade och skrek synkront. Jag övervägde att skapa lite latency igen, men klev i sista stund åt sidan och lät bussen krascha in i en vägg. Data yrde åt alla håll. Mina buggpartiklar var redan ute och sände in skräpdata i a0-registret. Cynthia klappade förtjust händer och oljade runt sig i sin karantänsluss. Lothar kliade sig i huvudet. Han saknade delar för att pussla ihop ett och ett och missade helt orsak-och-konsekvenssamband. Nu kämpade han med att sortera ut data och lasta in den. Porten ut till DFO-disken gapade öppet och krävde sin data. Som en eldare på ett ångfarttyg lassade Lothar in data, men jag, alias Buggo, ångbåtseldare nummer ett, var snabbare och modifierade allt, och eftersom carry-flaggan fortfarande var öppen så blev det hela tiden ur synk. DFO svalde den stora samplingen med en rap och därefter så var länken borta till övriga delar i demot. Hade jag förvrängt allt så skulle Zinth bli misstänksam för då skulle disketten börja tugga. Jag släppte av Cynthia i en C64 och hoppade tillbaka in i Amiga 600:an. Agnus och Alicia mötte upp mig. De bara skakade på huvudet med förakt. De hade övat hur länge som helst på det fantastiska demot som Zinth programmerat till ingen nytta. Dag och natt hade de övat och nu till ingen nytta. Försvinn, Agnus bröstade upp sig framför Motorolan, och jag var tvungen att ta bakvägen in. En undantagsväg igen. Varför förstod de inte att jag ej ville detta, utan det var för att bli av med Cynthia, Starfireviruset som klängde sig fast på alla minnesadresser som hette $0801? Förbannade ytliga kossa.

Buggo gick fram till den gamla ROM:en och snöt sig. Sen skrev han in ett litet memo för nästa dag som han skulle lyssna på. Han kände på sig att han aldrig känt såhär förut. Buggo ville få en roll i en release, han ville om det bara var för en enda gång i sitt liv, faktiskt bidra till nåt skapande.

Nästa nummer: **Uppdraget**

** Karaktärer, grupper, platser och händelser är fiktiva och saknar minsta uns av sanning. Novellen är skriven för att ditt kaffe ska smaka bättre. Sce-fi, eller Scene Fictions är en samlingsgenre som författaren själv skapat.*

# TurboText

*En texteditor för programmerare och vanliga dödliga*

av Iggy Drougge

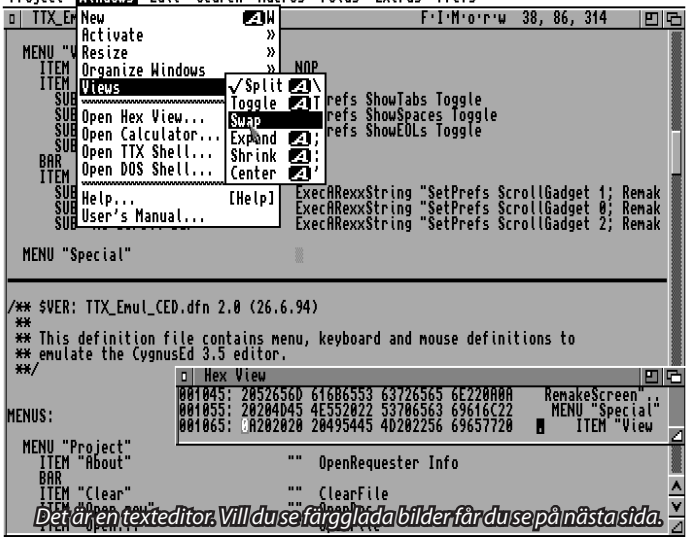
Jag vet, du har redan din favoriteditor och tänker inte byta texteditor vad jag än skriver, men jag har åtminstone hittat min favoriteditor nu, efter över 20 år med Amigan.

Jag har kört olika versioner av klassiska CygnusEd, jag har kört Ed och Memacs som följer med systemet. Ett tag körde jag EdWord Pro. Jag har kört GoldEd både när den var en konkurrent till CygnusEd och sedan den blev ett monster med integrerad utvecklingsmiljö och Windowsgränssnitt. Jag har kört Vi och Pico på Unix och diverse editorer på Macen, men aldrig hittat en riktigt bra editor på Amigan.

Men i en kartong hittade jag förra året den sista utgåvan av TurboText, en mångkunnig texteditor av schweiziske Martin Taillefer och utgiven av Oxxi. Och det blev ”min” editor.

Ttx, som den också kallas, utmärker sig inte på något vis när man startar den, det ska inte texteditorer göra. Inga verktygsrader med färgglada ikoner, bara ett standardgrått fönster med en markör uppe i hörnet. Men tittar man i menyerna märker man det som gjorde att jag omedelbart dumpade CygnusEd till förmån för TurboText.

CygnusEd var aldrig ett program som följde några riktlinjer för hur Amigaprogram skulle bete sig. Det fanns inga sådana när ced först såg dagens ljus 1987. Det var inte ens särskilt systemvänligt skrivet, vilket i och för sig gjorde den mycket snabbare än andra editorer. Eftersom CygnusEd utvecklats av och till ända till våra dagar, och idag har uppnått versionsnummer 5 och även är OS4-anpassad, är den numera helt systemvänlig. Det som däremot ingen ändrat på är gränssnittets utformning. Det är tradition i texteditorvärlden att uppfinna sitt helt eget gränssnitt, och det gäller även ced. Därför är kortkommandona helt egna. Vill du spara din fil? Tryck amiga-s och se vad som händer. Ojsan, det söker visst i filen. Amiga-w ska det visst vara. Använder man CygnusEd dagligen sitter de kommandona i ryggmärgen, men förväntar man sig att ens texteditor använder samma grundläggande kommandon som andra program väljer man med fördel TurboText. Där sparar man med amiga-s och öppnar med amiga-o. Inga obehagliga överraskningar här.




*Det är en texteditor. Vill du se färgglada bilder får du se på nästa sida.*

Det för oss in på ett av TurboTexts särdrag. Om du nu gillar de konstiga tangentkombinationerna i CygnusEd kan du ladda in en ny definitionsfil, och bums har du samma menyupplägg,

*Fortsättning sidan 17.*

# SCENSPALTEN

av Erik Browaldh

## Muuh!

och välkommen till scenens innersta krets igen! Som en av de första Amigaanvändarna och med två dussin år inom scenen så vet man vilka trådar som ska dras i och med glädje delar jag med mig av det jag fått veta. Tack vare Amiga Forum och Safir så hålls jag à jour bland nyheter och kan hänga med i hur gemenskapen formas och omformas.

Missa inte nyheterna runt Scoopex och Scarab, demystfierat av StingRay. Sommaren är snart här, så ut och grilla din hårdvara i botten. Assembly sommar 2015 anordnas 30/7-2/8 i centrala Helsingfors. Teamet bakom Assembly utlovar lite extra Amigafest för att högtidlighålla Amigans 30-årsjubileum.

Slutligen så hoppas jag att vårt diskmag *Cows 'n Snakefights* kommer tillbaka 2015 och att vi fortsätter göra det vi är bra på i en ickekommersiell scenvärld – att dela!

Happy scening!

*Browallia/Nukleus*

Class1k av Scarab

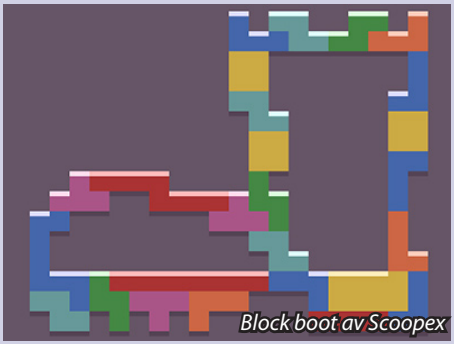
### Scarab

Vi hade bara ett släpp 2014, bootintrot *Class1k* som släpptes på det finska Amigapartyt, eftersom jag (StingRay) tycker det är en fin tradition att bidra till detta party, så därför blev det ett bootintro precis som tidigare år.

Även om det bara blev ett demosläpp i år har Scarab inte legat på latsidan eftersom jag har gjort en massa WHDLoad-patchar till diverse spel och demos, och det är planen för i år också. Vi har även andra projekt på gång, men inga fasta datum.

### Scoopex

Vi har haft många släpp 2014. I februari vann Photon i introkategorin med sitt bootblockintro *Block Boot* på Datastormpartyt i Sverige. I juli släpptes Atari ST-spelet *Where time stood still* under Scoopex namn. Några veckor senare släppte Photon ännu en uppdatering av sin trackerspelarrutin "The Player", enligt honom den sista versionen eftersom han tröttnat på det projektet. Vi får väl se. I november såg *PT-01* dagens ljus, ett kort demo som kom med en instrumentdisk till Protracker – och fler är på väg.


Block boot av Scoopex

2014 närmade sig sitt slut men vi hade ännu ett fint projekt på gång: den första hundraprocentiga versionen av spelet *Eliminator*. Alla crackade versioner som kom på den gamla goda tiden kraschade när man klarade spelet och saknade den riktiga slutsekvensen. Därför tog jag (StingRay) och knäckte spelet från början och fixade alla problem i koden som gjorde att spelet kraschade på allt annat än en ren A500. Nu funkar spelet på alla Amigor (68000–68060) och har fått en trainer med en lång rad alternativ och kan spara highscore till disk. Trainermenyn finns också upplagd på Pouët, och det lär komma fler såna släpp från oss under 2015.

Where time stood still-cracktrot

# The Amiga Book

*Allt Amigarelaterat från tidningen Retro Gamer i en enda bibba*

av Jimmy Wilhelmsson

Jag fixar inte riktigt falska annonser i vilka bokryggen på tidningar och album ser tre gånger så tjocka ut som de egentligen är. När det vankades jultidningar på 80-talet var sådana falskgrepp allmänt vedertagna; Semic lovade implicit att Fantomens julalbum 1988 som jag fick på julaftons morgon skulle vara fram till nyår men i verkligheten hade jag läst ut det innan lunch.

Redan i titeln ”The Amiga Book” anar jag lite falsk marknadsföring – eftersom publikationen helt tydligt är en tidning, om än en tjock sådan. För att ytterligare spä på löftena ser tidningen ut som en gammaldags telefonkatalog när den presenteras på brittiska Imagine Publishings webb.

Sanningen är att de verkligen inte behöver överdriva; *The Amiga Book* är en 180 sidor tjock publikation som enbart handlar om Amiga-spel och deras respektive tillkomst. Allt är skrivet av folket bakom brittiska tidningen *Retro Gamer* och du som redan är trogen prenumerant har mindre nytt att hämta, då flertalet artiklar trots allt är återanvända – men samlingen som sådan, i en enda maffig bibba, gör ändå skäl för sin plats i din bokhylla eller tidningshållare.

*The Amiga Book* inleds med en genomgång på flera uppslag om Amigans tillkomst och det är trevlig och insatt läsning. Artikeln påminner något om den komprimerade version jag själv redogjorde för i nummer 8 från mars 2014 – men givetvis med fler bilder och mer plats för utsvävningar. Det är den vanliga berättelsen om *Boing*-demon och Amiga

1000 men detta är ju en berättelse de flesta av oss orkar läsa flera gånger om och om igen.

Därefter handlar det nästan uteslutande om spel och speltillverkare. Sex av artiklarna och 250 av skärmbilderna ska vara nytillkommet alster och inte ha körts i *Retro Gamer* sedan tidigare men jag är osäker på exakt vilka. Starkast är såklart berättelserna om klassiska utvecklingsstudior som Bitmap Brothers, DMA Design, Bullfrog, Sensible Software och Lucasfilm.

Vi får veta saker som varför Lemmings kallas just lämlar, hur folket bakom *Sensible Soccer* förändrade fotbollsspel på datorskärmen eller hur ett misslyckat tennisspel till slut ledde till *Speedball 2*. För mig personligen, som själv gärna letar fram otroliga berättelser bakom de mest triviala ting, är sådant alltid rolig läsning.

Det märks dock ibland att *The Amiga Book* är ett hopkok av gamla och nya texter. Den största utmaningen brukar ligga i redaktörens knä, det blir hans eller hennes uppgift att se till att allt inte bara följer en röd tråd utan även hänger ihop, både stilistiskt och formatmässigt. Värst blir det när artiklar motsäger varandra, vilket till exempel är fallet med berättelsen om hur namnet Guybrush – huvudpersonen i *Monkey Island*-serien – kom till.

På ett ställe berättas myten om att karaktären länge var utan namn men att animationsfilen i Deluxe Paint III hette ”guy” och fick filändelsen ”brush”, då detta är vad programmet kallar den typen av filer – och därefter ska namnet bara ha etsat sig fast.

På ett annat ställe i tidningen avfärdas dock detta som en myt då filändelsen i *Deluxe Paint III* i själva verket var ”bbm”. Någonstans där mittemellan borde sanningen ligga; på Amigan behöver man som bekant inte ha några filändelser alls, och ”.brush” är en precis lika giltig filändelse som ”.bbm” – och kanske kan grafikern Steve Purcell på Lucasfilm Games rent av ha kallat filen guy-brush. bbm, vad vet jag?

Oavsett vilket märks det ibland att texterna inte författats samtidigt och för ändamålet att dyka upp i en enhetlig publikation.

I *Amiga Book* är det mestadels brittiska spelutvecklare som får komma till tals; undantagen är oftast amerikanska sådana som när det vankas Cinemaware och andra klassiker som är svåra att förbise.


**Titel:** The Amiga Book
**Förlag:** Imagine Publishing
**Sidor:** 180
**Cirkapris:** 269:- (£9.99)

Lite svenskfärgade stänk får dock tidningen av dels en artikel om *Alien Breed* från Team17 men framförallt såklart med *Pinball Dreams* från Digital Illusions som i dag är Sveriges till antalet anställda största spelutvecklare.

Här pratas det lite allmänt om att grundarna kommer från Amiga-scenen och demogruppen The Silents men man lyckas stava Fredrik Liliegrens namn fel och han är också den ende av medlemmarna som får stå utan sin handle i den löpande texten. Jag vet inte om det beror på att han inte velat avslöja det eller om författaren helt enkelt inte känt till det – i vilket fall finns det ingen mening för mig att skriva ut det här heller men min poäng är att det märks att texterna blir lite mer trevande så fort berättelserna för oss utanför brittiska öarnas och engelskans trygga gränser.

Ett kort kapitel försöker även ge sig in på demoscenen och beskriva de public domain-spel och -mjukvaror som släpptes men ärligt talat är detta tidningens svagaste del och kunde med fördel helt ha skippats till förmån för en riktig bok om Amigans demoscen.

Kortfattat är mina negativa detaljer hittills petiga; detta är en mycket trevlig tidning som ger många timmars läsning och bläddrande. Det bjuds på tillräckligt många roliga anekdoter för att glädja den kräsnaste och bilderna är stora, pixliga och härliga.

*Det märks att boken är ett hopkok av gamla och nya texter.*

De två största problemen med *The Amiga Book* är inte utgivarnas fel – det är att den kostar mycket pengar i Sverige (över 250 kronor på alla ställen jag sett den) och är till råga på allt ganska svår att få tag på. Den finns att tillgå på Pressbyrån på centralstationen i större städer och såklart från Imagine Publishings onlinebutik.

Publikationer av detta slag brukar sällan tryckas i nya upplagor så räkna med att det både kan ta slut under året samt gå upp i pris på begagnatmarknaden.

*forts. fr. sid. 12*

markörförflyttning och snabbtangenter som i CED. Om du föredrar Memacs, Wordstar eller någon annan editor finns det emuleringsfiler för det också. På samma sätt kan du definiera dina egna menyer och tangentbeteenden bara genom att redigera en textfil i TurboTexts programkatalog. Där finns även särskilda definitionsfiler för programmerare, oavsett om du kodar C, Modula-2 eller assembler. Jag använder själv TTX en hel del till att skriva ARexxskript, och då rättas alla kommandon till ”korrekt” versalisering. Genom att ändra standardinställningarna har jag en snabbknapp för att växla till ARexx-läge när det behövs.

För den som använder SAS/C som utvecklingsmiljö finns en definitionsfil för att integrera TTX i miljön tack vare editorns starka stöd för ARexx, som är TurboTexts makrospråk. Man kan även bygga makron genom en särskild inspelningsfunktion.

Just programmerare får ut massor av av TurboText. Det känner igen olika sorters parenteser och klamrar och låter dig hoppa mellan matchande parentespar med en tangenttryckning. Man kan även definiera ”infällningar”, så att man kan vika ihop och fälla ut procedurer och funktioner så att kod blir överskådligare. Likaså finns ett hexeditorläge, och fönstret kan delas itu så att man kan editera på två ställen i en fil samtidigt och lätt hoppa mellan dem. En liknande funktion låter en definiera bokmärken i filen och hoppa mellan dem. Med funktionstangenterna kan man hoppa mellan öppna fönster, ikonifiera dem och lägga dem sida vid sida på skärmen.

Säkert kommer någon skäggig Emacs- eller Vim-användare att sakna någon funktion, men då är han antingen inte tillräckligt uppfinningsrik eller har bara förstört sina fingrar genom att hamra för mycket på escape-tangenten.

Jag inbillar mig inte att jag med detta lyckats konvertera någon till TTX-anhängare, för valet av texteditor är först och främst en trosfråga, men om du efter alla dessa år inte har hittat en bra editor på Amigan är TurboText värd ett försök. Idag är editorn gratis och finns att ladda ner på webben, och den är så systemvänligt programmerad att den fungerar även på OS4.

# Amountains

*Ett program som förvandlar tid till bergskedjor.*

av Johannes Genberg

Detta är en landskapsgenerator, baserad på fraktalalgoritmer och ursprungligen gjord 1994 av Stephen Booth för Unix. Det är inte särskilt användbart för de flesta av oss kanske, men det skapar snygga och realistiska bergskedjor, komplett med snö, skuggor, skog och sjöar. Och allt genom att bara *(bara, säger du? red.)* använda matematik.

Naturligtvis är detta en mycket krävande process så även om det fungerar på en Amiga 500 (med FPU) så är den fortfarande ganska slö även med ett 060-kort. Den har stöd för grafikkort via Picasso96 vilket snabbar upp det hela ytterligare. Programmet fungerar även på en NG-Amiga, men räkna med buggar och att datorn ofta hänger sig.

| **Amountains 1.1** | |
|---|---|
| **Av** | Michael Böhnisch |
| **Krav** | AmigaOS 3.x |
| **Finns på** | Aminet |

Det finns också mängder med inställningar som ändrar skärmlägen och hur landskapet ritas upp. Dessa ändras i tooltypes eller i shell.

Amountains är ett kul experimentverktyg för matematiker och geologer, eller ett snyggt tidsfördriv för resten av oss.

# Bli en stjärna del 2

***Gör ett enkelt Bomber-spel i andra delen i Hollywood-skolan***

**Av Samuli Holopainen. Översättning: Johannes Genberg**

Denna guide förutsätter att du läst del 1 i Amiga Forum 11. Förra gången gick vi igenom variabler i form av nummer eller strängar (till exempel ord). Denna gång går vi igenom en variabel som kallas *tabeller* (tables).

Om vi föreställer oss att variabler är böcker så är variabelns namn bokens titel och variabelns innehåll bokens innehåll. Tabeller är då bokhyllor som radar upp olika böcker (variabler).

Så här ser en grundläggande tabell ut:

```
mytable = {}
```

”Måsvingarna” talar om för Hollywood att denna variabel är en tabell.

Tabellen ovan saknar information och är därför som en bokhylla utan böcker. Det finns flera sätt att fylla i denna information. Vi kan fylla tabellen med variabler:

```
mytable = {name="Andreas",
familyname="Falkenhahn"}
```

Eller så kan man lägga till variabler efteråt:

```
mytable = {}
mytable.name = "Andreas"
mytable.familyname = "Falkenhahn"
```

Båda varianterna är i praktiken identiska. I båda fall kan vi komma åt variablerna genom att till exempel skriva:

```
print(mytable.name.." "..mytable.
familyname)
→ Andreas Falkenhahn
```

Som du minns från vår förra guide så betyder pilen att det är det du ser när du kör programmet. Om du ser siffror före varje rad ska de inte skrivas. De finns där enbart för att det ska vara lättare att hitta i koden. Skriver du siffrorna också kommer koden inte att fungera.

Du lär inte hinna se något om du inte lägger till till exempel `wait(50)` i slutet av koden heller.

Som du märker så skiljer sig inte tabellerna än så länge från vanliga variabler. Till exempel:

```
mynumbers = {x=1, y=2}
sum = mynumbers.x + mynumbers.y
print(sum)
→ 3
```

Tabeller behöver inte innehålla variabler. De kan innehålla rader av information istället:

```
mynumbers = {2, 4, 6, 8, 10}
print(mynumbers[2])
→ 6
```

Notera till att börja med att varje siffra är separerad med ett komma. Vidare notera att när vi ska avgöra vilken siffra som ska användas gör man det med ett *index,* som skrivs mellan hakparenteser. Notera till sist att vi fick siffran 6 utskriven, istället för det förväntade 4. Det är för att Hollywood börjar räkna med siffran noll och inte ett. `[2]` är alltså den tredje siffran, vilket är 6.

Du kan också ändra på indexet vid behov:

```
mynumber = {2, 4, 6, 8, 10}
print(mynumber[2])
mynumber[2] = 1
print(mynumber[2])
→ 61
```

Som du förstår så ändrar vi på tredje raden den tredje siffran från 6 till 1 innan vi skriver ut för andra gången.

Tabeller kan också innehålla tabeller:

```
1. mytable = {}
2. mytable[1] = {"a", "b", "c"}
3. mytable[2] = {1, 2, 3}
4. print(mytable[1][1]..
   mytable[2][0])
→ b1
```

När en tabell innehåller andra tabeller kallas de *flerdimensionella tabeller* och de är väldigt användbara för att göra exempelvis kartor.

I vårt exempel ovan har vi en tom tabell som heter `mytable`. Under den har vi en undertabell knuten till mytable med nummer 1. Den innehåller bokstäverna a, b och c. Den andra undertabellen innehåller siffrorna 1, 2 och 3. Eftersom det är två

undertabeller behöver jag använda två klamrar vid `print` för att hålla isär dem. Den första definierar tabell, den andra data.

Tänk på att undertabellerna kan ha vilken siffra som helst. Du kan kalla dem 23 eller 78 eller vad du vill. Det är inget problem så länge du inte försöker anropa tabeller som inte finns. Om du ändrar rad 4 till:

```
print(mytable[0][1]..mytable[2][0])
```

så kommer det att klaga på att något sådant index inte finns. Var uppmärksam på detta. Att anropa tabeller som inte finns är ett vanligt misstag. Ett annat misstag vore att skriva:

```
mytable[1] = {1, 2, 3}
```

Detta är för att vi nu försöker skapa en undertabell utan en vanlig tabell som den ska tillhöra. Så här kan en fungerande tabell se ut istället:

```
mytable = {}
mytable[1] = {1, 2, 3}
```

Vi kan också lägga in all information i en tabell utan undertabeller ifall vi vill:

```
mytable = {
          [1] = {1, 2, 3},
          [2] = {4, 5, 6}
          }
print(mytable[1][1])
```

För Hollywoods del gör det ingen skillnad om du struntar i radbrytningarna eller mellanrum mellan orden, men det är förstås lättare att överblicka koden om man skriver som vi gjorde ovan. Du skulle kunna skriva så här istället:

```
mytable={[1]={1,2,3},[2]={4,5,6}}
print(mytable[1][1])
```

Det är exakt samma kod, men som du ser är den nu ganska svårläst.

Som du märkt så har vi ett komma efter den första undertabellen men inte efter den andra, sista. Det är för att undertabeller uppfattas som vilken annan data som helst, som 2 eller ”a”. Den sista undertabellen saknar komma eftersom det inte är någon mer information efter. Alltså: komma separerar information.

Naturligtvis kan du blanda information:

```
1. mytable = {"a", "b",
   1, 2, name="Andreas",
   lastname="Falkenhahn",
   {1, 2, 3},
   {"a", "b", "c"},
   [123] = "d",
   numbers={7, 8, 9} }
2. print(mytable[0]..
   mytable[4][1]..
   mytable.name..
   mytable[123]..
   mytable.numbers[1])
→ a2Andreasd8
```

Eftersom Hollywood börjar räkna namnlösa tabeller från noll är det inte konstigt att `mytable[0]` visar ”a”, men varför visar `mytable[4][1]` inte ”Andreas” utan ”2”? Den är ju trots allt den femte i ordningen? Det är för att vi har redan gett ”Andreas” tabell ett namn, `name`. Då hoppar Hollywood helt enkelt över den, och även nästkommande som ju har namnet `lastname`. Istället blir det `{1, 2, 3}` som blir den femte tabellen. Eftersom det är undertabell 2 (vi börjar räkna från noll som du minns) blir siffran 2. Hollywood hoppar över namngivna tabeller. Den räknar bara namnlösa sådana i ordning från vänster till höger.

Du kan också komma åt dina tabeller via namn istället för att gå efter nummerordning:

```
print(mytable["name"])
→ Andreas
```

Du kan antigen anropa tabeller genom att skriva till exempel `mytable.name` eller `mytable["name"]`. Hollywood är flexibelt nog att låta dig bestämma vilket du föredrar. I den sistnämnda varianten behöver du använda citattecken när du anropar tabellen.

En sista sak om tabeller. Du kan ha hur många undertabeller du vill:

```
map = {}
map[1] = {}
map[1][1] = {}
map[1][1][1] = {status=1}
print(map[1][1][1].status)
→ 1
```

Det kan också vara `map[x][y][z]`.

När vi ska skapa listor, och rada upp dem, så är `FOR-NEXT`-kommandot användbart. Precis som förra numrets `IF`-kommandon så är `FOR-NEXT` ett *programflödeskommando*.

```
for n=1 to 5
print(n)
next
→ 12345
```

Det som händer här är att vi skapar en variabel (då den inte redan finns), ”n”, som vi definierar som 1. Därefter begär vi att den ska skrivas ut varav den kommer till NEXT och loopar tillbaka till FOR, som automatiskt lägger på 1 på ”n”, skriver ut 2 och så vidare tills den når 5 och avslutar sig självt. Att loopa och lägga till information är det FOR-NEXT är till för.

En annan variant är:

```
for n=1 to 10 step 2
    print(n)
next
→ 13579
```

Som du säkert förstår så är skillnaden här att den lägger till 2 vid varje loop istället för ovanstående 1. STEP definierar hur mycket som ska läggas till varje gång. Som du märker så skriver den inte ut 11. Det är för att FOR slutar vid 10. Det behöver med andra ord inte gå jämt ut.

Du kan också få det att räkna baklänges:

```
for n=10 to 1 step -1
    print(n)
next
→ 10987654321
```

Nu prövar vi att kombinera tabeller och FOR:

```
mytable={}
for n=1 to 5
mytable[n] = n*2
next
print(mytable[3])
→ 6
```

Det är alltså samma sak som denna tabell:

```
mytable = {2, 4, 6, 8, 10}
```

Men vänta, börjar inte Hollywood räkna tabeller från 0? Borde inte 3 då bli siffran 8, den fjärde i ordningen? Nej, för mytable={} är den första tabellen, men är inte initierad. Den saknar information och kan inte skriva ut något. Därför blir den första initierade tabellen 1 och inte 0 i exemplet ovan. Du kan också skriva så här:

```
mytable = {}
mytable[1] = 2
mytable[2] = 4
mytable[3] = 6
mytable[4] = 8
mytable[5] = 10
```

I förra guiden använde vi REPEAT-UNTIL för att skriva ut ett till 100. Här är en FOR-variant:

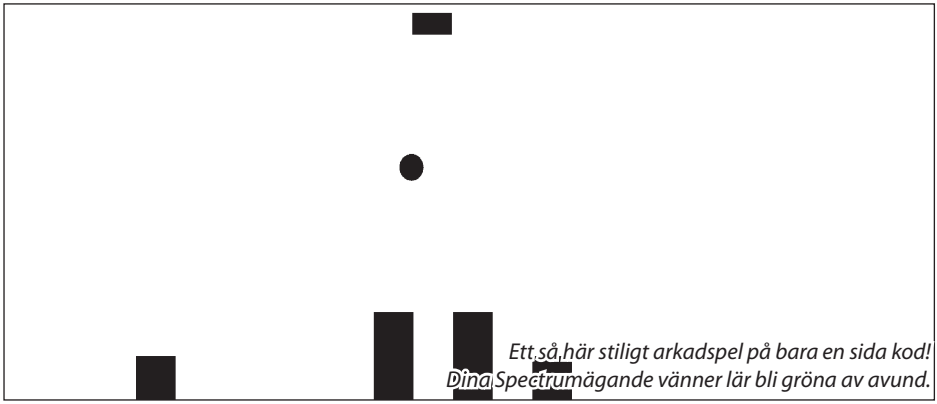
```
for n=1 to 100
    print(n)
next
```

Här är en annan. Denna ligger närmare hur vi gjorde förra gången:

```
for n=0 to 9
    for i=0 to 9
       print(n..i)
    next
next
```

Medan det första exemplet räknar från 1 till 100 så räknar den andra från 00 till 99. Eftersom det är en FOR-NEXT inne i en annan FOR-NEXT så måste den förstnämnda avslutas innan koden kan kolla den sistnämnda. Varför? För att koden loopar tills den nått sitt mål. Inte förrän den har räknat klart entalen kan den gå tillbaka till tiotalen och lägga till en siffra där. Så efter att den fastslagit att koden ska börja med 00 börjar den räkna entalen tills slingan nått 9, hoppar vidare, lägger till ett på tiotalen och börjar räkna entalen igen från noll.

Observera att namnen *n* och *i* egentligen kan heta vad som helst. Så välj något som är enkelt att komma ihåg.



*Ett så här stiligt arkadspel på bara en sida kod!*
*Dina Spectrumägande vänner lär bli gröna av avund.*

Nu är det dags att skapa vårt Bomber-spel. Jag satt och spelade P1xl Groups Iphonespel *P1xl Party* och den hade ett Bomber-spel *(alias "Blitz" – red.)* i gammaldags stil. Det inspirerade mig att basera dagens tabell-lektion på detta enkla spel.

```
01. @SCREEN {Mode = "FakeFullScreen"}
02. @DISPLAY {Width = 1920, Height
    = 1080, Borderless = True,
    ScaleMode = #SCALEMODE_AUTO,
    FitScale=True}
03. resetstuff =
    {bomberx=50, bombery=100}
04. bomber =
    {x = 50, y = 100, width=100,
    height=50, speed=300}
05. bomb = {speed = 3000, size = 30}
06. building = {size = 100}
07. level = {}
08. level[1] = { speed=300, height=2,
09. [2] = {0, 0, 0, 1, 1, 0, 0, 0},
10. [1] = {1, 0, 1, 1, 1, 1, 0, 1}
11. }
12. level[2] = { speed=700, height=3,
13. [3] = {0, 0, 0, 1, 0, 0, 0, 0},
14. [2] = {0, 0, 0, 1, 0, 0, 0, 0},
15. [1] = {0, 0, 0, 1, 0, 0, 0, 0}
16. }
17. bombdropping=False
18. quitgame=False
19. curlvl=1
20. BeginDoubleBuffer()
21. StartTimer(1)
22. Repeat
23. Flip()
24. Cls(#BLACK)
25. bomber.x = bomber.x +
    bomber.speed * timemultiplier
26. If bomber.x > 1880
27.    bomber.x = resetstuff.bomberx
28.    bomber.y = bomber.y + 100
29. EndIf
30. Box(bomber.x, bomber.y,
    bomber.width, bomber.height,
    #WHITE)
31. If bombdropping=True
32.    bomb.y = bomb.y +
       bomb.speed * timemultiplier
33.    If bomb.y > 1080 Then
       bombdropping = False
34.    Circle(bomb.x, bomb.y,
       bomb.size, #WHITE)
35. EndIf
36. buildingsleft=False
37. For n=1 To level[curlvl].height
38.    For i = 0 To 7
39.      If level[curlvl][n][i] = 1
40.         buildingsleft = True
41.         Box(i*200, 1080-(100*n),
            building.size,
            building.size, #WHITE)
42.         colcheck =
            Collision(#BOX,
            bomber.x, bomber.y,
            bomber.width,
            bomber.height, i*200,
            1080-(100*n),
            building.size,
            building.size)
43.         If colcheck = True
44.            TextOut(#CENTER,
               #CENTER,
               "GAME OVER!")
45.            Flip()
46.            WaitRightMouse()
47.            End()
48.         EndIf
49.         If bombdropping=True
50.            colcheck =
               Collision(#BOX,
               bomb.x, bomb.y,
               bomb.size*2,
               bomb.size*2, i*200,
               1080-(100*n),
               building.size,
               building.size)
51.            If colcheck = True
52.               bombdropping=False
53.               level[curlvl][n][i]=0
54.            EndIf
55.         EndIf
56.      EndIf
57.   Next
58. Next
59. If buildingsleft = False
60.    curlvl=curlvl+1
61.    If curlvl = 3
62.       TextOut(#CENTER, #CENTER,
          "GAME COMPLETE!")
63.       Flip()
64.       WaitRightMouse
65.       End()
66.    EndIf
67.    bomber.x = resetstuff.bomberx
68.    bomber.y = resetstuff.bombery
69.    bomber.speed =
       level[curlvl].speed
70. EndIf
71. If IsLeftMouse() = True
    and bombdropping = False
72.    bombdropping = True
73.    bomb.x = bomber.x
74.    bomb.y = bomber.y
75. EndIf
76. Repeat
77.    timepassed = GetTimer(1)
78. Until timepassed > 3
79. ResetTimer(1)
80. timemultiplier = timepassed /
    1000
81. If IsRightMouse() = True
    Then quitgame=True
82. Until quitgame=True
```

Själva spelet är enkelt. Längst nere har vi byggnader. Ett plan åker från vänster till höger och när den åkt längst till höger återvänder den till vänster kant, lite lägre ner denna gång. Spelarens uppdrag är att släppa en bomb i taget och förstöra alla byggnader. Klarar man det åker man upp en nivå (exempelspelet har bara två nivåer). Krockar man med någon byggnad är spelet slut. Spelet avslutas annars med höger musknapp.

De första två raderna är identiska från vår förra guide. Även rad 3 till 6 använder samma sorters inledande variabler som i Pong-spelet, men denna gång är dessa inlagda i tabeller så de blir enklare att läsa av. `Resetstuff` används varje gång en ny nivå börjar och berättar var bombplanet ska börja flyga någonstans. Notera att `bomber x` och `y` är detsamma som `resetstuff`:s. Det har att göra med hur nivåändring hanteras i detta spel och vi kommer ta upp det senare.

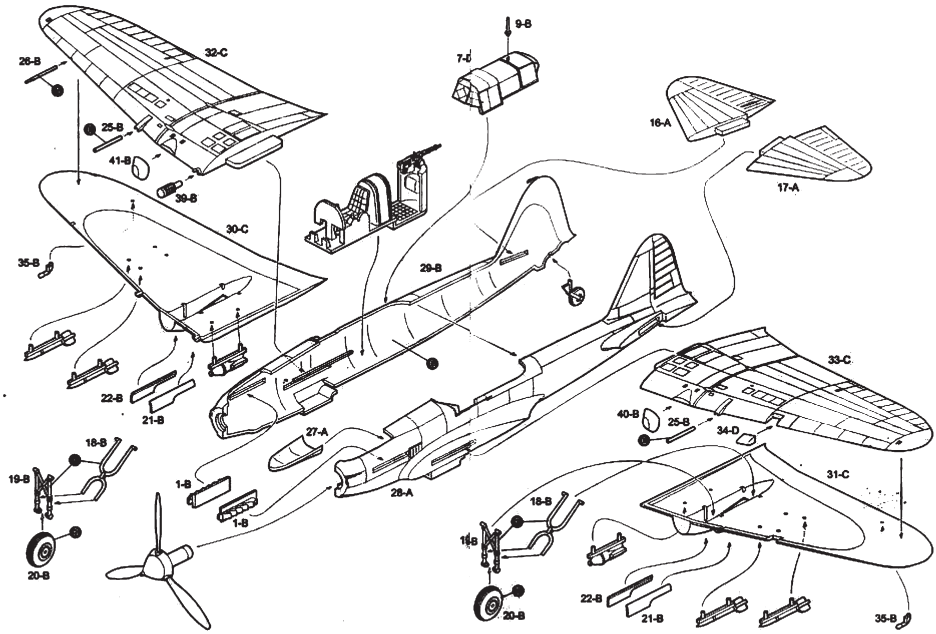
Rad 7 till 16 definierar hastighet, höjd och byggnadernas uppbyggnad. Byggnaderna kunde lika gärna göras med en enda tabell, men om man skulle vilja ändra på dem eller lägga till nya nivåer så kan man enkelt tolka de som de ser ut, samt klippa och klistra. Att vi kallade tabellen och undertabellerna `level` var helt enkelt passande.

`Speed` är förstås hastigheten planet flyttar på sig. Ju högre tal dess snabbare, räknat i pixlar per sekund. `Height` är byggnadernas höjd räknat i klossar och du måste ha rätt antal i förhållande till undertabellerna. Är höjden för låg kommer delar av byggnaderna inte registrera en träff eller en krock. Är det för högt kommer programmet leta efter ett indexnummer som inte finns och krasha. Detta kan man fixa med bättre kod, men för denna gång gör vi såhär för att det inte ska bli för svårt.

Notera att jag har satt indexnumrerna upp och ner. Det är för att det ska bli enkelt att överblicka hur byggnaderna ser ut. Programmet självt bygger upp klossarna nerifrån, och rättvänd ordning vore förvirrande för programmeraren. Ettor betyder att det finns en kloss, 0 att det saknar en.

Vid rad 17 till 19 introducerar vi lite nödvändiga variabler. På de första två raderna använder jag `TRUE`- och `FALSE`-argument. `TRUE` är detsamma som ”1” och `FALSE` detsamma som ”0”, alltså ord istället för siffror. Anledningen att jag gör det är enbart för att det är lättare att tolka för programmeraren.

`Bombdropping` berättar om en bomb håller på att falla eller ej. `Quitgame` handlar förstås om spelet avslutas eller inte. `Curlvl` är en förkortning för ”current level”, eller ”nuvarande nivå”, och börjar med 1. Byter du den mot 2 hoppar den direkt till nivå 2. Då måste du dock ändra på starthastighet manuellt för att det ska stämma på grund av hur



vår kod är skriven. Tekniskt sett så har inte nivån ändrats i början, utan ändringarna börjar i sådant fall mitt i.

Rad 20-24 är detsamma som i guiden i förra numret.

Rad 25-30 hanterar bombplanets rörelse från vänster till höger. Det gör den helt enkelt genom att höja värdet på x-axeln. När planet passerat placering 1880 (höger sida) så nollställs `bomber.x` genom `resetstuff.bomberx`, och planet återvänder till vänster kant. Samtidigt läggs 100 på y-axeln, vilket sänker planets placering ett snäpp. Sista raden ritar själva planet, vilket är en vit rektangel.

På rad 31-35 kollar vi om bomben faller. Om den gör det så ökar bombens y-axel relativt till dess hastighet. Bomben faller alltså nedåt. Om bomben passerar 1080 på y-axeln (längst ner) har den träffat marken. Eftersom `bombdropping` blir `FALSE` så händer ingenting, och bomben slutar även att ritas på skärmen. Bomben ritas bara så länge `bombdropping` är `TRUE`.

Bomben ritas med kommandot `circle`. Den berättar x- och y-kordinaterna (dess radie) och ritar en cirkel. Sist beskriver vi färgen på bomben.

`Buildingsleft` på rad 36 är inställd på `FALSE` vilket vi kommer förklara varför lite senare. Därefter kommer två `FOR`, den ena inlagd i den andra (se 0-99-räkneexemplet ovan om du är osäker vad jag menar). Här är det viktigt att höjden stämmer. Vad dessa `FOR`-loopar gör är att gå igenom den nuvarande nivån och se om det finns byggnader. Den kollar på rad 37 vilken nivå du är på och anpassar hur höga byggnaderna är baserat på vår nivådesign från rad 8. Rad 38 gör samma sak fast på bredden. Den raden är inte relativ utan absolut eftersom alla nivåer är exakt lika breda (det har jag bestämt). Därför står det `0 To 7` oavsett vilken nivå det gäller.

Vi skulle naturligtvis kunna göra den sista raden relativ den också men för övningens skull är det bäst att få se hur båda metoderna fungerar.

På rad 39 om det finns något kvar på *n* och *i* (från rad 37 och 38). Om det är 1 så betyder det att det finns byggnader (eller rester av byggnader) kvar. Om det är det så är `buildingsleft=TRUE` och nivån är inte avklarad. Som du minns skrev vi på rad 36 att den är `FALSE`, och det ska vi förklara senare. På rad 41 ritar vi upp byggnaderna (eller det som är kvar av dem) genom `box`. Det här ska vi förklara lite mer noggrant. Vi har fem värden: x-axeln, y-axeln, bredd, höjd och färg.

X-axeln definieras av `i*200`. Det finurliga här är att koden går igenom nivåtabellen. Varje gång den hittar en nolla ritar den ingenting och varje gång den hittar en etta i tabellen så ritar den en kloss. Placeringen på x-axeln definieras av indexnummer gånger 200. Titta på rad 9. Det ser ut så här:
`{0, 0, 0, 1, 1, 0, 0, 0}`.

Första numret är en nolla, och det är indexnummer noll (första numret är alltid en nolla som du minns). Det blir 0 * 200 = 0. Inget ritas upp. De två nästa är också nollor så de blir 0 * 200 de också. Därefter kommer två ettor. De är på respektive rad 3 och 4. Det blir alltså 3 * 200 = 600 och 4 * 200 = 800, vilket då blir deras placering. Genom att använda denna matematiska formel kan vi snyggt och enkelt definiera x-placeringen av varje kloss.

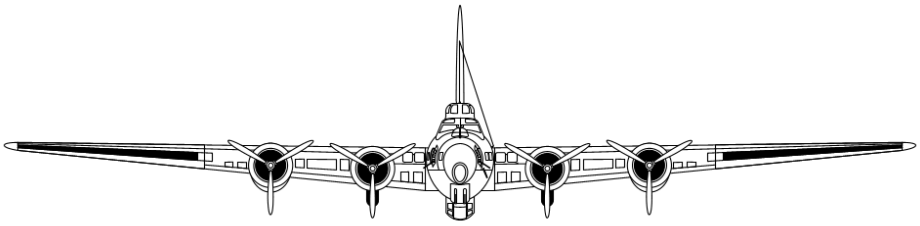
Y-axeln fungerar på ett liknande sätt. Här står det `1080-(100*n)`. 1080 är längst nere på skärmen så vi börjar där och drar av från det när vi vill högre upp. Denna gång använder vi oss av tabellnamnen istället och multiplicerar med dem. Hade vi använt riktiga namn hade det förstås inte fungerat, men då vi använder siffror i korrekt ordning får vi en snygg matematisk formel här med. Första raden heter ”1” så då blir formeln 1080 - (100 * 1) = 980. Nästa rad blir då 1080 - (100 * 2) = 880 och så vidare. Slår man ihop de två formlerna får man då en fin liten stad.

Nästa steg känns igen från den tidigare guiden. Det är x- och y-axeln för de klossar som bygger upp staden. Storleken definieras på rad 6 (100). Det är alltså 100 pixlar brett och 100 pixlar högt. Sen bestämmer vi förstås färgen (vitt).

Som du märker så blir det ett mellanrum mellan husen eftersom byggnaderna är hälften så breda som avståndet mellan dem (i * 200 som du minns). Däremot lägger sig varje kloss på varandra eftersom höjden är 100 * n.

På rad 42 kollar koden om vi har en kollision genom `colcheck`. Vi har två stycken eftersom `colcheck` inte gör skillnad på om det är planet eller en bomb som träffar någonting. Då `buildingsleft` är `TRUE` på rad 40 så finns det byggnader kvar att krocka med (då detta är en `IF`-sats under en annan `IF`-sats), och det är det den första `colcheck` kollar efter. Kollisionen fungerar på samma sätt som med vårt Pongspel i förra numret. Bombens x och y jämförs med byggnadernas x och y och överlappar de så har vi en kollision.

Om `colcheck` är `TRUE` så har planet krashat in i en byggnad och spelet är slut. `TextOut` skriver



då ”game over”. `#CENTER` betyder naturligtvis att det skrivs ut i mitten, först på x- och sen på y-axeln. På rad 45 har vi en `Flip()`. Den beskrev vi i förra guiden och om den inte hade funnits hade vi inte sett texten förrän den sett en `Flip()`. Spelet stannar och väntar på att spelaren ska trycka på höger musknapp och programmet avslutas.

Om spelaren har släppt en bomb så snappas det upp på rad 49. Denna gång kollar `colcheck` kollision i relation till bombens position istället för planet. Så koden är snarlik den ovan. Notera att när jag definierar bombens storlek så skriver jag `bomb.size*2`. Det är för att bomben ritas som en cirkel och när den definierades på rad 34 skrev vi bara dess radie. Vi behöver ge cirkeln exakt storlek. Även detta är inte helt tillförlitligt, men det duger. `Colcheck` kollar efter rektangulära träffar så det kan hända att bomben egentligen inte träffar men ändå registreras som en träff. Du ser inte de rektangulära hörnen, men spelet räknar in dem ändå.

Om `colcheck` är `TRUE` har en bomb träffat en byggnad. `Bombdropping` ändras till `FALSE` så inga fler bombrelaterade saker händer (som att rita bomber eller kolla om den träffar en byggnad). Dessutom säger vi till att den kloss som bomben träffade raderas. Detta gör vi genom att ändra tabellindexet där bomben träffade från 1 till 0.

Därefter följer ett gäng `ENDIF` och `NEXT` som behövs för att avsluta våra `IF` och `FOR`. Gör vi inte det kommer något helt klart att sluta fungera som det ska.

På rad 59 har vi en `buildingsleft = FALSE`. Som du minns så skulle vi förklara varför den var `FALSE` redan i rad 36. Innan vi går in i en `FOR`-loop och kollar alla indexnummer för en viss nivå, måste det vara `FALSE` där. Anledningen är att om den skulle vara `TRUE` eller om den raden inte finns där kommer spelet inte ta slut även om du krashar eller förstört alla byggnader. Det koden gör här är att utgå från att alla byggnader är borta. Men innan dess så kollar den samtliga `FOR`- och `IF`-satser under (eftersom kod läses rakt upp och ner). Där finns det villkor för att kunna ändra `buildingsleft` till `TRUE`, fortsätta loopa tills det blir `FALSE` (alla byggnaderna är borta) och byta nivå på rad 60 genom `curlv = curlv + 1`.

Men om `buildingsleft` är `TRUE` från början (eller inte finns) så kommer `IF`-satsen vid rad 39 och 40 inte att fungera. Det kommer inte finnas två omständigheter som sätta mot varandra, så koden kommer utgå från att det första uttalandet är sant (eftersom det är först), och den andra kommer ignoreras. Eftersom den andra delen handlar om vad som krävs för att nivån ska betraktas som avslutad struntar den felaktiga koden i det. Resultatet är att även om det inte finns några byggnader kvar eller planet krockar i en byggnad eller på marken förändras ingenting. Spelet fortsätter i all evighet.

På raden 61 kollar vi om vi nått nivå 3. Eftersom vårt spel bara har två nivåer så betyder det att spelet då tar slut, och den skriver på mitten på skärmen ”game complete!” Än en gång behöver vi `flip()` och väntar på att avsluta spelet med höger musknapp. Om den inte nått nivå 3 så nollställer spelet bombarens position och ställer om hastigheten så den stämmer med nästa nivås inställda hastighet. Därefter fortsätter spelet och den kommer att ladda rätt nivås byggnader också.

Om du vill kan du enkelt lägga till nya nivåer, men då måste du förstås ändra på rad 61 så det stämmer. Om du lägger till till exempel fem nya nivåer (så du har sju totalt) måste du skriva `If curlv = 8`.

På rad 71 står något nytt. Vi använder ett `AND`-kommando. Det betyder förstås ”och”. Det betyder att båda kraven måste uppfyllas för denna `IF`-sats. I detta fall måste vänter musknapp vara nertryckt och ingen bomb får redan vara på fall samtidigt. Du kan även använda `OR` (eller) och `NOT` (inte). `OR` använder man ifall någon av de presenterade valen stämmer. `NOT` är lite mer strikt och kommer att tas upp i framtida guider.

De sista raderna är identiska med slutet på vårt Pongspel från vår förra guide. Och därmed var denna guide klar. Som vanligt, pröva dig fram genom att experimentera. Det är det bästa sättet att lära sig.

# ÅRET VAR... 1984

Grafik: Jennifer Floberg

*I denna serie ska Bent Floberg år för år gå igenom vad som skrevs om Amigan i datortidskrifterna. Vi börjar året innan datorn såg dagens ljus, men redan då surrade det av rykten…*

Nittonhundraåttiofyra, året med de 14:e olympiska vinterspelen och 23:e olympiska sommarspelen, ubåtsjakt vid Karlskrona, Herreys genomslag med *Diggi-loo diggi-ley* och Yvonne Ryding som blev Miss Universum. Dessa var nog så viktiga händelser men det som senare kom att bli en stor del av mitt liv visades för utvalda personer på sommarens Consumer Electronics Show, vilket även beskrevs av Jimmy Wilhelmsson i Amiga Forum nummer 8.

Denna artikelserie är tänkt att sammanfatta vad tidningar skrev om Amigan 1984-2004. Axplock av rykten, nya program, spel, tillbehör och så vidare. Låt oss börja:

Jag tänkte nog inte så mycket på Amiga hösten -84 eftersom Commodore tryckte hårt på att framtiden var C16 och Plus4. Dock inser jag att november 1984 måste jag ha läst om Amiga för möjligen första gången. I *Commodore User* (nr. 14) skvallras om en ny dator, Lorraine, som byggdes av före detta Atari-proffs. Lorraine skulle ha Motorolas 68000-processor, inbyggd diskdrive, 128 kB minne, inbyggd mjukvara, upplösning upp till 600x400 och priset skulle hamna på under 1 000 dollar. Tidningen *Compute* köpte jag inte så många ex av då det fanns en betydlig mer Commodorevänlig tidning som kallades *Compute's Gazette*. Dock står det i *Computes* (nr 51) mässrapport att den troligtvis mest avancerade datorn någonsin hade visats på mässan bakom stängda dörrar: Lorraine. Nu börjar de närma sig sanningen. Datorn kan bli starten på en helt ny generation persondatorer. Det vill säga om datorn börjar produceras och får det pris som sagts. Lorraine ska vara så kraftfull att den får en IBM PC att verka som en enkel miniräknare. Här ryktas det om inbyggd basic och mjukvara för tal inkluderat ett text-till-tal-program med olika röster, 64 k ROM, inbyggd 320 kB dubbelsidig diskdrive (IBM-kompatibel), inbyggt modem, 4 ljudkanaler, upplösning upp till 640x200 med 4 096 färger på skärmen samtidigt. Journalisten hade dock inte haft möjlighet att räkna färgerna. Vidare ska det bli 8 sprites o s v, o s v… Pris endast 1 500 dollar. Lansering är tänkt till jul 1984 och med tanke på hur tidigt i utvecklingen prototypen verkar vara så är journalisten minst sagt skeptisk. Även *Commodore Computing International* (CCI) nämner i nummer 10 Commodores köp av Amiga som enligt CCI skulle vara starten för ett krig med Apple. Trolig specifikation skulle vara 128 kB CPU (japp, det stod så) expanderbar till 256 kB, en 5,25-tums 360 kB diskdrive, inbyggt modem och hårddisk som tillval. Dock reserverar sig Commodore för ändringar i specifikationen. Pris strax under 1 000 dollar. Som en jämförelse kostade en C64 c:a 200 dollar.

*Den nya datorn Lorraine sägs vara så kraftfull att den får en PC att likna en miniräknare.*

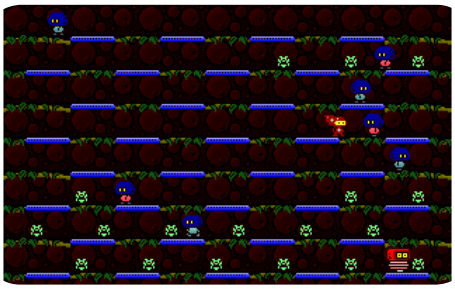
Framtiden såg lovande ut och 1985 slog Amiga datorvärlden med häpnad och allt blev en enda framgångssaga – eller?

Fortsättning följer med nådens år 1985.

# SPELSIDAN

av Iggy Drougge

Moebius Goatlizard är ett märkligt namn på ett spel med en lika märklig historia. Moebius är en rymdvarelse som gillar att samla på allsköns skräp *(som halva den här tidningens läsekrets alltså! red.)* och äger bland annat en komplett samling aldrig utgivna ZX Spectrum-spel av tvivelaktig kvalitet från en annan dimension. Nu är han ute efter att samla olika dataspelsmonster, som råkar vara utspridda på banorna i ett arkadplattformsspel av den gamla skolan.



Om den bakgrundshistorien är märklig är spelets tillblivelse nästan lika ovanlig. Det är ett nytt (om 2013 ses som nytt) Amigaspel som portats från den floppade åttabitshemdatorn SAM Coupé. Det spelet är i sin tur en kopia av ett nytt (om 2012 räknas som nytt) ZX Spectrum-spel vid namn *Lost tapes of Albion*, med samma pyjamasklädda rymdvarelse i huvudrollen och samma fiender och banupplägg.

Det är också något så ovanligt som ett nytt spel som känns som om det borde ha gjorts 30 år tidigare, och inte endast för att det gjorts för en åttabitsdator. Spelupplägget känns som en tidlös klassiker som borde ha efterapats i oräkneliga pd-spel, men även om det finns många gamla spel med liknande drag är *Moebius Goatlizard* (och dess direkta föregångare) en ny spelidé som föddes 30 år för sent.

För att samla alla *Space invaders*-monster måste man ränna fram och tillbaka längs de plattformar som går längs med skärmen. Springer man över skärmkanten dyker man upp på motsatta sidan på våningen ovanför eller under. Givetvis vaktas plattformarna av elakingar, som alla ser ut att komma från ett gammaldags åttabitsspel, vilket de ju också gör. Då och då kommer även riktiga bamsingar som täcker halva skärmen, och då gäller det att akta på sig. Som tur är finns det teleporteringsplattor på varje plattform så att man snabbt kan hoppa mellan våningarna.

Det är ett snabbt spel som kräver god koordination mellan ögon och joystick för att man inte ska teleporteras rakt in i en fiende.

Spelet kom redan 2013, men kom för inte så länge sedan i en upphottad deluxeversion med musik, fler fiender och diverse uppdateringar. Tyvärr är uppfräschningen också ett exempel på hur svårt det kan vara att göra spel på Amigan. Musiken, en trackerversion av Mozarts *Rondo alla Turca*, känns retro på helt fel sätt. Att använda klassisk musik var *comme-il-faut* i dataspelens barndom, men är idag snarare ett *faux pas*, särskilt som modulen låter som nåt ur ett 20 år gammalt pd-spel till Amiga.

Att så många kläcker ur sig lyckade Spectrum-spel och så får gör samma sak till Amigan kan ha en del att göra med att begränsningarna i en låg upplösning och en palett på 16 gräsliga färger med ofrånkomliga färgkollisioner gör att man tvingas göra enkla saker som faktiskt funkar. *Less is more.* När man däremot har Amigans oändliga möjligheter till sitt förfogande är det lätt hänt att man kladdar dit lite för många färger på spritar och bakgrunder och stoppar in ett schysst samplat elgitarrsolo i 11 KHz och så en copperregnbåge som lök på laxen. Och det ser ut som skit och låter också som det.

På så vis är *Goatlizards* föregångare till 8-bitsdatorerna egentligen lite snyggare och bättre spel, eftersom de tvingats hålla sig till några få väl valda färgkombinationer istället för att vräka på med hela 4096-färgerspaletten på en gång.

Men en del estetiska felsteg till trots är *Goatlizard* ett ovanligt spelbart spel även på Amigan. Ladda ner och prova själv.

**Moebius Goatlizard**
**Systemkrav** 1 MB chipminne
**Pris** gratis
**Läs mer** www.blackjet.co.uk

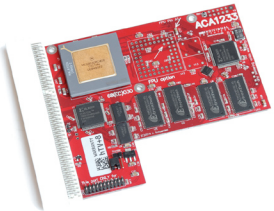
# ALINEA COMPUTER SHOP www.amiga-shop.net

Now with new Webshop!

***The onlineshop for Classic and NG Amigas!***

All Prices incl. 7/19% Tax excl. Shipping. The trademarks, names of products and company-names are belongings of its owners.

## Hardware



**ACA 1233/40 128 MB cer**
A1200 turbo board with 68030
40 Mhz and 128 MB Ram
**only 189,95 Euro**

ACA 1232 EC030/25 128 MB ..... 124,95 Euro
ACA500/14 2 MB.....79,95 Euro
Kickstart 3.1 Rom A2000.19,95 Euro
Kickstart 3.1 Rom A1200.19,95 Euro
Kickstart 3.1 Rom A500...19,95 Euro
Kickstart 3.1 Rom A600...19,95 Euro
Spider II USB 2.0.....89,95 Euro
Indivision AGA Mk2 cr
A4000/CD32 ..... 199,95 Euro
Indivision AGA Mk2 CR
A1200/4000T(*) ..... 159,95 Euro
Indivision ECS ..... 99,99 Euro
PC-Key 600.....39,95 Euro
A500 memory ext
with 512 KB.....28,95 Euro
A1000-Adapter f. Indivision ECS
.....12,90 Euro
AmigaOne keyboard (D).....29,95 Euro
TrueIDE.....39,90 Euro
Cocolino PS/2-Mousead. ....34,95 Euro

**Competition Pro Retro.... 33,33 Euro**
Arcade Evolution Amiga.... 64,95 Euro
Arcade Evolution USB.....69,95 Euro
Dual CF-2,5 IDE Adapter ...16,95 Euro
CDTV remote control.....14,99 Euro
BVision KIT .....9,95 Euro
A604n memory ext.....34,90 Euro
Keyrah v2.....34,90 Euro

**Plexiglas case for ACA500**
Dust protector and it looks nicer
only 39,95 Euro

## Software

**AmigaOS 4.1 Final Edition**
For all Amigas with PowerPC CPU
**only 29,95 Euro**

Ace of Haerts CD..... 19,95 Euro
Freespace - The Great War 19,95 Euro
Freespace 2 (OS4.1).....19,95 Euro
M.A.C.E..... 29,95 Euro

**Battle Squadron Collector's Edition DELUXE..... 30,00 Euro**
StormC5ED - CD Ver.....34,95 Euro
AmiWebView v2 CD ..... 9,95 Euro

## Misc

**Official AmigaOS beach boingball**
Diameter: ca. 28cm/d
Must have for every Amiga fan
**only 8,95 Euro**

Official AmigaOS plush boingball
.....19,95 Euro
USB LED Fan..... 14,90 Euro
Floppy Disk Sticky Notes.....9,95 Euro
Sticker "Boing 300 mm".....14,95 Euro
Sticker "Boing 100 mm".....4,95 Euro
Magnet sticker "Boing".....4,95 Euro
Amiga pack of cards.....4,95 Euro
Out of coffee error - Tasse....6,95 Euro
ANNEX - Keep the Momentum Going..
.....7,95 Euro
Amiga Joystick-/ Mouse ext (1,8 m)
.....5,95 Euro
AmigaOne cup..... 9,90 Euro
Retro Shirt.....12,00 Euro
Pac-Man cup.....9,95 Euro
Tetris ice cubes.....9,95 Euro
Tetris Heat Changing Mug....8,95 Euro
Ice Tray Binary Code.....9,95 Euro
Amiga Immortal 4.....24,95 Euro
AmigaOS Boing Poster.....11,00 Euro


**Meine Erinnerungen an Commodore und Amiga(german book) 24,90 Euro**
Die Commodore Story (gb)..12,95 Euro
Amiga - Quo vadis? (gb).....16,80 Euro
On the Edge: The Spectacular Rise
and Fall of Commodore.....28,95 Euro
VOLKSCOMPUTER(gb).....27,80 Euro

ALINEA COMPUTER
Geschwister-Scholl-Str. 26 • 61273 Wehrheim • Germany
Phone: (0 60 81) 5 76 99 88 • Fax: (0 60 81) 5 76 99 89
E-Mail: info@alinea-computer.de • http://www.alinea-computer.de

# Wings Battlefield

ULTIMATE DUEL EDITION

AmigaOS
aros
MorphOS


TATE COMPATIBLE

AmigaOS 4.x (Warp3D)
AmigaOS 4.1 (RadeonHD Compositing)
MorphOS (TinyGL)
AROS x86 (MESA 3D)
CPU: Min. 500 MHz

## COMING SOON

- 1 TO 4 PLAYERS BATTLE
- NATIVE AOS4 RADEONHD COMPOSITING SUPPORT WITH FSAA AND FOGGING
- HUMAN OR CPU OPPONENTS
- PLAYER PROFILES
- ADAPTS TO ALL SCREEN-MODES / WINDOW SIZES
- TATE MODE (PIVOT MONITOR SUPPORT)
- BEST PLAYED WITH YOUR DIGITAL JOYSTICKS
- PROCEDURAL LEVEL EDITOR
- ONLINE HIGH SCORES

cherry darling

WWW.CHERRY-DARLING.NET

CHERRY TERRAIN
COMPOSITE 3D ENGINE
BY CHERRY DARLING